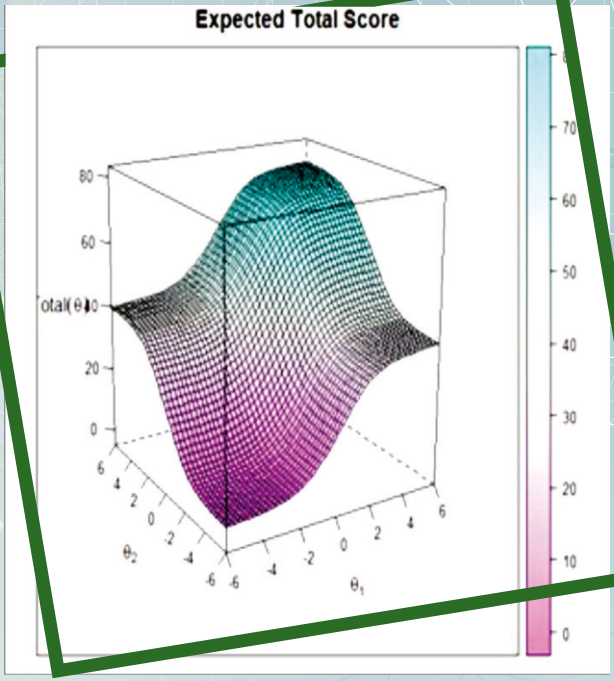
Analisis Instrumen Penelitian **dengan Teori Tes Klasik dan Modern** Menggunakan **Program R**



Hasan Djidu ~ Raoda Ismail ~ Nur Anisyah Rachmaningtyas ~ Sumin
Okky Riswanda Imawan ~ Suhariyono ~ Koryna Aviory ~ Eko Wahyunato Prihono
Devi Dwi Kurniawan ~ Johan Syahbrudin ~ Nurdin ~ Yudince Marinding ~ Firmansyah

Prof. Samsul Hadi, M.Pd., MT. ~ Prof. Heri Retnawati, M.Pd.

Editor: Prof. Dr. Samsul Hadi, M.T. | Prof. Dr. Heri Retnawati, M.Pd.

IKAPI IKATAN PENERBIT INDONESIA
Appti
UNY PRESS

ANALISIS INSTRUMEN PENELITIAN DENGAN TEORI TES KLASIK DAN MODERN MENGGUNAKAN PROGRAM R

Penulis:

Hasan Djidu, M.Pd.
Raoda Ismail, M.Pd.
Nur Anisyah Rachmaningtyas, M.Pd.
Sumin, M.Si.
Okky Riswandha Imawan, M.Pd.
Suhariyono, M.Pd.
Koryna Aviory, M.Pd.
Eko Wahyunanto Prihono, M.Pd.
Devi Dwi Kurniawan, M.Pd.
Johan Syahbrudin, M.Pd.
Nurdin, MA.
Yudince Marinding, M.Pd.
Firmansyah, M.Pd.

Prof. Samsul Hadi, M.Pd., MT.
Prof. Heri Retnawati, M.Pd.

Editor:

Prof. Samsul Hadi, M.Pd., MT.
Prof. Heri Retnawati, M.Pd.

UNDANG-UNDANG REPUBLIK INDONESIA
NOMOR 28 TAHUN 2014
TENTANG HAK CIPTA

Pasal 2

Undang-Undang ini berlaku terhadap:

a. semua Ciptaan dan produk Hak Terkait warga negara, penduduk, dan badan hukum Indonesia;
b. semua Ciptaan dan produk Hak Terkait bukan warga negara Indonesia, bukan penduduk Indonesia, dan bukan badan hukum Indonesia yang untuk pertama kali dilakukan Pengumuman di Indonesia;
c. semua Ciptaan dan/atau produk Hak Terkait dan pengguna Ciptaan dan/atau produk Hak Terkait bukan warga negara Indonesia, bukan penduduk Indonesia, dan bukan badan hukum Indonesia dengan ketentuan:
    1. negaranya mempunyai perjanjian bilateral dengan negara Republik Indonesia mengenai pelindungan Hak Cipta dan Hak Terkait; atau
    2. negaranya dan negara Republik Indonesia merupakan pihak atau peserta dalam perjanjian multilateral yang sama mengenai pelindungan Hak Cipta dan Hak Terkait.

BAB XVII
KETENTUAN PIDANA

Pasal 112

Setiap Orang yang dengan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 7 ayat (3) dan/atau Pasal 52 untuk Penggunaan Secara Komersial, dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau pidana denda paling banyak Rp300.000.000,00 (tiga ratus juta rupiah).

(1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
(2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
(3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
(4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

ANALISIS INSTRUMEN PENELITIAN DENGAN TEORI TES KLASIK DAN MODERN MENGGUNAKAN PROGRAM R

Penulis:

Hasan Djidu, M.Pd.
Raoda Ismail, M.Pd.
Nur Anisyah Rachmaningtyas, M.Pd.
Sumin, M.Si.
Okky Riswandha Imawan, M.Pd.
Suharyono, M.Pd.
Koryna Aviory, M.Pd.
Eko Wahyunanto Prihono, M.Pd.
Devi Dwi Kurniawan, M.Pd.
Johan Syahbrudin, M.Pd.
Nurdin, MA.
Yudince Marinding, M.Pd.
Firmansyah, M.Pd.

Prof. Samsul Hadi, M.Pd., MT.
Prof. Heri Retnawati, M.Pd.

Editor:

Prof. Samsul Hadi, M.Pd., MT.
Prof. Heri Retnawati, M.Pd.

UNY PRESS

2022

ANALISIS INSTRUMEN PENELITIAN DENGAN TEORI TES KLASIK DAN MODERN MENGGUNAKAN PROGRAM R

Penulis:
Hasan Djidu, Raoda Ismail, Sumin, Nur Anisyah Rachmaningtyas, Okky Riswanda Imawan, Suharyono, Koryna Aviory, Eko Wahyunanto Prihono, Devi Dwi Kurniawan, Johan Syahbrudin, Nurdin, Yudince Marinding, Firmansyah, Samsul Hadi, & Heri Retnawati.

Editor: Prof. Samsul Hadi, M.Pd., MT. & Prof. Heri Retnawati, M.Pd.

ISBN: 978-602-498-415-1
Edisi Pertama, Juli 2022

Diterbitkan dan dicetak oleh:
UNY Press
Jl. Gejayan, Gg. Alamanda, Komplek Fakultas Teknik UNY
Kampus UNY Karangmalang Yogyakarta 55281
Telp: 0274 – 589346
Mail: unypress.yogyakarta@gmail.com

Anggota Ikatan Penerbit Indonesia (IKAPI)
Anggota Asosiasi Penerbit Perguruan Tinggi Indonesia (APPTI)

Penyunting Bahasa : Raoda Ismail
Desain Sampul : Ibnu Qayyim Rabbani
Tata Letak : Sumin
Isi di luar tanggung jawab percetakan

Hasan Djidu, dkk.

ANALISIS INSTRUMEN PENELITIAN DENGAN TEORI TES KLASIK DAN MODERN MENGGUNAKAN PROGRAM R
-Ed.1, Cet.1.- Yogyakarta: UNY Press 2022
vi+ 250 hlm; 15 x 23 cm
ISBN: 978-602-498-415-1
1. Analisis Instrumen Penelitian dengan Teori Tes Klasik dan Modern Menggunakan Program R

Pengantar Penulis

Segala puji hanya milik Allah SWT, yang telah melimpahkan rahmat, karunia, dan pertolongan-Nya sehingga kami dapat menyelesaikan penyusunan buku Analisis Instrumen Penelitian dengan Teori Tes Klasik dan Modern menggunakan Program R. Program R merupakan *software* yang sangat *powerfull*, lengkap, gratis, dan menawarkan banyak kemudahan kepada pengguna untuk dengan berbagai paket analisis. Buku ini mencoba menghadirkan proses analisis instrumen penelitian dengan pendekatan *Classical Test Theory* (CTT) dan *Item Response Theory* (IRT) dengan contoh-contoh praktis dan aplikatif menggunakan program R. Materi-materi dalam buku ini disusun dalam bahasa yang sederhana sehingga dapat lebih mudah untuk dipahami oleh para pengguna yang berasal dari kalangan pemula (non statistikawan) disertai dengan pembahasan contoh-contoh aplikatif.

Adapun topik yang dibahas dalam buku ini meliputi: pengenalan Program R, proses instalasi, dan perintah-perintah dasar dalam Program R, membangkitkan data, CTT, IRT unidimensi penskoran dikotomi dan politomi, serta IRT multidimensi penskoran dikotomi dan politomi beserta cara menginterpretasikan hasil analisis data menggunakan program R. Semoga buku ini dapat menjadi referensi pengantar bagi mahasiswa dan dosen dalam belajar dan mempelajari CTT dan IRT untuk menganalisis instrumen penelitian.

Kami merasa bahwa buku ini masih memiliki kelemahan dan kekurangan. Kritik dan gagasan yang membangun dari pembaca sangat kami harapkan, agar menjadi bahan penyempurnaan buku ini di masa depan. Akhir kata, kami selaku tim penulis menghaturkan terima kasih kepada semua pihak yang telah membantu dan mendukung penyusunan buku ini, khususnya para pengajar dan mahasiswa Program Studi S3 PEP Universitas Negeri Yogyakarta.

Yogyakarta, Juli 2022

Penulis

Daftar Isi

Chapter 1
Pendahuluan

Oleh: Nurdin, Sumin, & Samsul Hadi

1.1 Sejarah Perkembangan Program R

R merupakan bahasa pemrograman dan sistem perangkat lunak untuk komputasi dan grafik. R terdiri dari bahasa ditambah lingkungan run-time dengan grafik, debugger, akses ke fungsi sistem tertentu, dan mampu menjalankan program yang tersimpan dalam file skrip. Pada tahun 1992 di Selandia Baru, nama R diambil dari penemu program ini, yaitu R Ross Ihaka dan Robert Gentleman di University of Auckland.

Program R pertama kali diimplementasikan pada awal 1990-an oleh Robert Gentleman dan Ross Ihaka, keduanya anggota fakultas di University of Auckland. Robert dan Ross mendirikan R sebagai proyek sumber terbuka pada tahun 1995. Sejak tahun 1997, R Core Group telah mengelola proyek R. Dan pada Februari 2000, R 1.0.0 dirilis. Bahasa R dimodelkan secara dekat pada Bahasa S untuk Komputasi Statistik yang disusun oleh John Chambers, Rick Becker, Trevor Hastie, Allan Wilks, dan lainnya di Bell Labs pada pertengahan 1970-an dan tersedia untuk umum pada awal 1980-an. Program R bersifat terbuka sehingga semua orang dapat berkontribusi untuk melakukan pengembangan-pengembangan bagian tertentu. Karena itulah perkembangan program R ini sangat pesat dan semakin luas penggunanya (Hafner, 2019).

R adalah bahasa pemrograman yang sangat powerful dan banyak digunakan di dunia data science dan big data. Pada mulanya, R adalah bahasa dan *environment* yang dibangun untuk komputasi statistik dan grafik, namun karena popularitasnya yang sangat besar, sekarang R memiliki ketentuan yang cukup untuk mengimplementasikan pembelajaran mesin dan algoritma Deep Learning dengan cara yang cepat dan efisien. Keistimewaan program R yang mengusung konsep Open Source dan gratis bagi semua pengguna di seluruh dunia, telah menjadikan software ini berkembang pesat. Sejak awal mula diluncurkan hingga buku ini diterbitkan, telah dikembangkan sebanyak 18.311 paket analisis dalam berbagai bidang analisis data statistik, big

data, data sains dan psikometri. Sejak pertengahan 1997 pengembang program R memiliki tim Inti sebagai orang-orang yang berperan aktif dan sangat berjasa dalam membangun dan mengembangkan program R, dengan hak akses dapat menulis ke kode sumber R, yaitu terdiri dari 27 orang dan ribuan kontributor dari seluruh dunia (https://cran.r-project.org/web/packages/).

1.2 Instalasi Program R

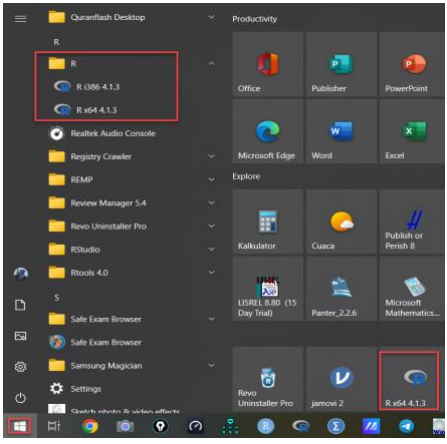
Program R gratis dan pemasangannya sangat mudah. Bagi anda yang baru pertama sekali menggunakan R, silahkan lakukan instalasi mengikuti langkah-langkah berikut ini.

```
https://cran.r-project.org
#lalu Pilih (klik) sesuai dengan device yang anda gunakan
Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
Download R for macOS
Download R for Windows
install R for the first time.
Download R-4.2.0 for Windows (79 megabytes, 64 bit)
```

Selanjutnya anda hanya perlu membuka file yang sudah diunduh (download) dan lakukan pemasangan sesuai dengan petunjuk yang tersedia. Jika pemasangan telah selesai, anda dapat menjalankan R console yang telah terpasang didesktop atau program file sesuai dengan versi yang ada pasang, dengan logo sebagai berikut.

Gambar 1.1 Logo R Console

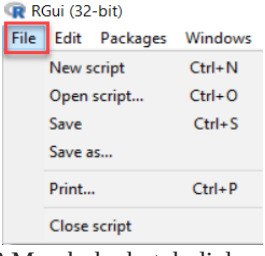
Kita dapat memulai R console di Windows 10 dengan cara mencari logo R di Start menu atau Desktop Windows, sehingga muncul tampilan seperti Gambar 1.2.



Gambar 1.2 Posisi R Console di Start Menu Windows 10.

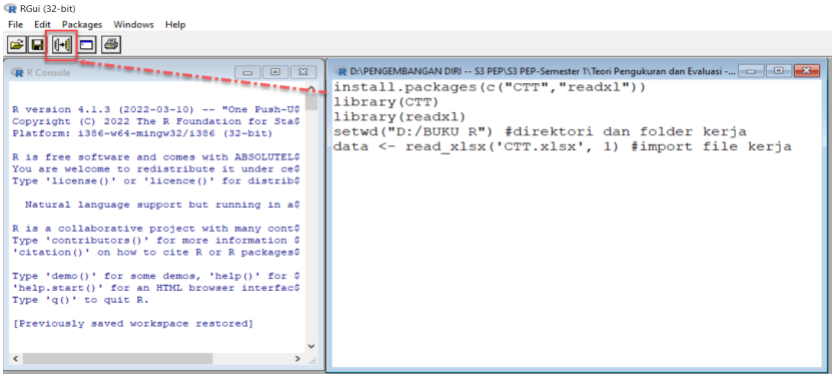
Untuk memulai penggunaan R anda dapat membuka melalui icon R di start menu windows (Gambar 1.2) atau desktop, kemudian setelah interface R studio telah terbuka, lanjutnya dengan membuka New script, untuk memasukkan syntax atau perintah analisis data di R Console.

```
File - new script (Ctrl N)
```



Gambar 1.3 Membuka kotak dialog New Script

Interface Program R terdiri R console, R Editor, dan R Graphic. Ketikan bahasa atau perintah pada menu utama R console, lalu klik icon  run sehingga muncul R *Graphic* seperti pada Gambar 1.



Gambar 1.4 Tampilan Antar Muka *R console*, *R Editor* dan *R Graphic*.

Syntax awal yang harus dipersiapkan sebelum analisis lebih lanjut menggunakan R Console ataua menginstall packages melalui syntax berikut ini:

```
install.packages (c("nama_paket1","nama_paket2"....),
```

kemudian kita harus panggil paket analisis yang telah diinstall menggunakan syntax berikut ini:

```
library(nama_paket)
```

kemudian dilanjutkan dengan mengatur direktori kerja dengan syntax berikut:

```
setwd("D:/nama_folder")
```

dilanjutkan dengan import data dengan ekstensi .xlsx atau .csv dengan syntax berikut: `data <- read_xlsx("nama file.xlsx")`

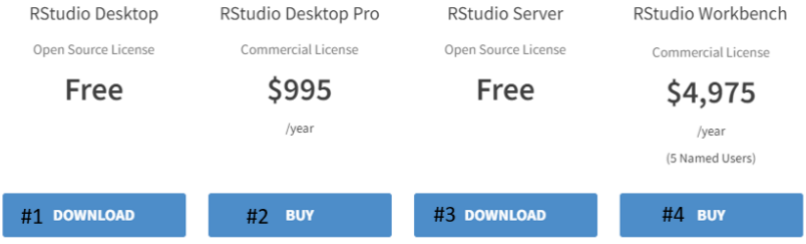
1.3 R-Studio

Seperti program-program lain yang sudah sering kita gunakan, Program R juga mengalami perkembangan yang sangat cepat. Banyak pengguna R yang tiba-tiba sangat expert ketika sudah menggunakan R tipe baru yang dikenal sebagai R-Studio. Keunggulan dari R-Studio dibandingkan dengan R-gui (versi lawas) terletak pada fasilitas-fasilitas yang memudahkan penggunanya. Berbagai syntax tidak lagi harus diketik manual, tetapi semua sudah disediakan oleh R-Studio. Seperti sebuah studio, R-Studio menyediakan semua kebutuhan yang Anda ingin kerjakan bersama R.

Untuk memulai R-Studio, kita harus melakukan instalasi R-Studio. Langkah-langkah umum instalasi R-Studio adalah:

https://www.rstudio.com/products/rstudio/download/

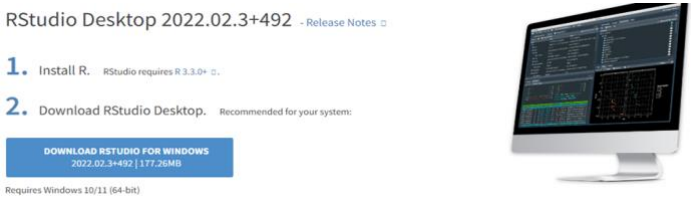
Setelah itu Anda akan diberikan empat pilihan yang bisa diunduh:



Gambar 1.5. Website R Studio (tempat untuk mendownload R studio)

Apabila kita membutuhkan **R** untuk menyelesaikan pekerjaan pribadi, maka silahkan klik pilihan #1 **DOWNLOAD.**

https://www.rstudio.com/products/rstudio/download/#download

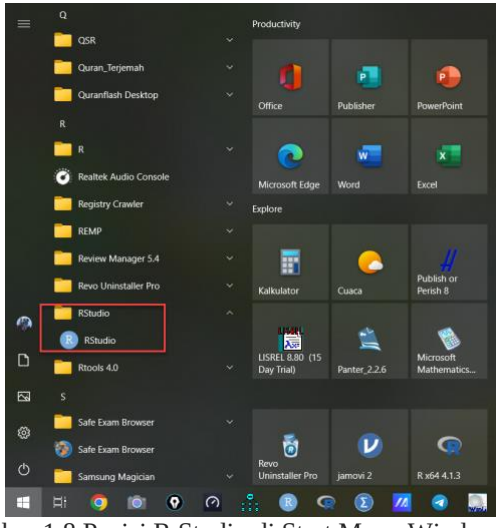


Gambar 1.6. Website unduhan R-studio untuk windows

Selanjutnya klik Download *R Studio For Windows*, lalu bukalah file (exe) yang diunduh (biasanya file ini disimpan di folder download) dan melakukan instalasi sesuai dengan petunjuk yang diberikan. Setelah instalasi *R-Studio* selesai, maka R-Studio bisa langsung dijalankan. Untuk menjalankan R Studio, silahkan cari di program file atau desktop dengan cara mengklik 2 kali logo R Studio sebagai berikut:

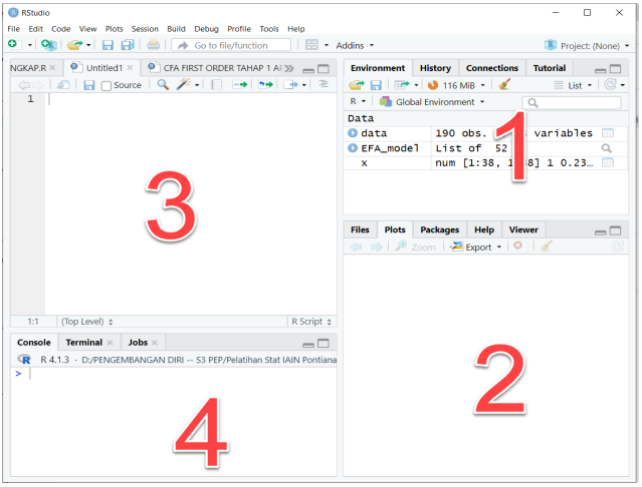
Gambar 1.7 Logo R Studio

Kita dapat memulai R Studio di Windows 10 dengan cara mencari logo R di Start menu atau Desktop Windows, sehingga muncul tampilan seperti Gambar 1.8.



Gambar 1.8 Posisi R Studio di Start Menu Windows 10.

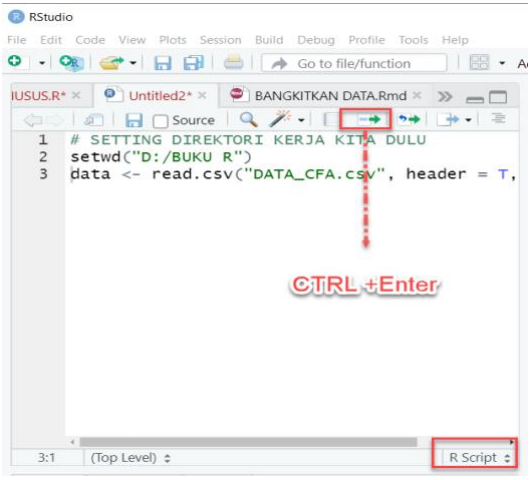
Setelah kita jalankan dengan mengklik dua kali logo R-Studio, maka kita akan dihadapkan pada antar muka software yang terdiri dari 4 kolom dengan fungsi yang berbeda sebagai berikut.



Gambar 1.9. Antar Muka *R-Studio*

File skrip anda, yang merupakan file teks biasa dengan akhiran ".R". Misalnya, *NamaFileFavoritAnda.R*. File ini berisi kode Anda.

1. Semua objek yang telah kita definisikan di environment.
2. Membantuk file dan plot.
3. Konsol atau R Script, yang memungkinkan anda mengetik perintah (syntax) dan menampilkan hasil analisis. Untuk menjalankan syntax dapat dilakukan dengan mengklik tanda panah biru di sudut kanan atas kolom R Script atau menekan tombol ***Ctrl+Enter*** di keyboard komputer anda.

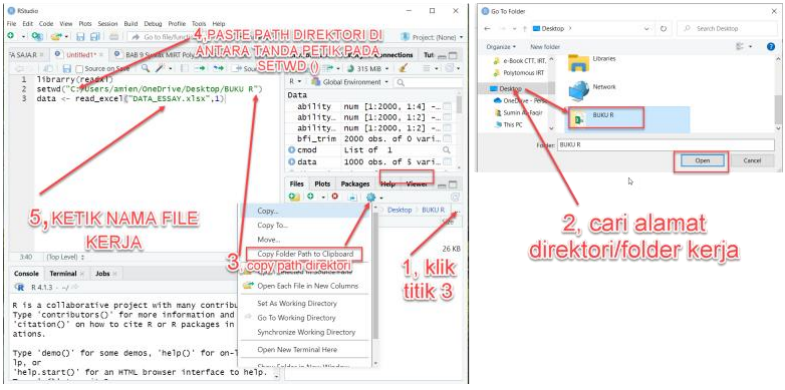


Gambar 1.10 Cara menjalankan syntax R Studio

1.4 Manajemen Direktori dan File Kerja Program R

Hal penting yang perlu diperhatikan sebelum mulai bekerja dengan program R adalah pengaturan direktori dan file kerja, terutama bagi para pengguna awal. Ketika kita bekerja menggunakan sistem operasi windows pertama-tama kita harus mengenal Windows Explorer, sebagai beranda kedua setelah Windows Desktop. Ada beberapa cara untuk menjalankan Windows Explorer, yang paling sederhana adalah dengan cara menekan klik kanan mouse pada start menu Windows yang ada di sudut kiri bawah pada Windows 10, kemudian klik file explorer, maka anda akan dihadapkan pada interface Window Explorer yang berisi Drive standar yaitu C, D, E atau F.

Pada R studio kita dapat mengatur direktori dengan memperhatikan langkah-langkah sebagaimana tertera pada gamber berikut ini.



Gambar 1.11 Manajemen Dikrektori R Studio.

Jika benar maka seharusnya anda mendapatkan jalur (path) folder, misalnya pada contoh ini, kami menyimpan file di Desktop Windows dengan Folder Utama Bernama BUKU R, yang berisi file kerja dan syntax R, path atau alamat foldernya adalah: *C:/Users/amien/OneDrive/Desktop/BUKU R*. Nama file kerja adalah: DATA_ESSAY.xlsx. Tetapi, pada komputer anda tentu tidak sama, tergantung dimana anda menyimpan file kerja, misalnya di drive D yang terdiri dari 3 folder, yaitu: Data Penting, Mahasiswa, folder inti adalah Skripsi dan nama file adalah: Data Skripsi.xlsx, maka pengaturan direktori kerjanya di R studio adalah:

```
setwd (“D:/Data Penting/Mahasiswa/Skripsi”)
data <- read_xlsx(“Data Skripsi.xlsx”).
```

1.5 Beberapa istilah pada program R

Direktori kerja: Direktori file tempat Anda bekerja. Perintah yang berguna: dengan getwd() anda mendapatkan lokasi direktori kerja Anda saat ini dan setwd() memungkinkan Anda menyetel lokasi baru untuknya. Ini sangat berguna ketika Anda ingin memuat atau menyimpan beberapa file. Anda cukup mengatur direktori kerja Anda secara global dan semua file dapat dimuat atau akan disimpan ke dalam direktori ini.

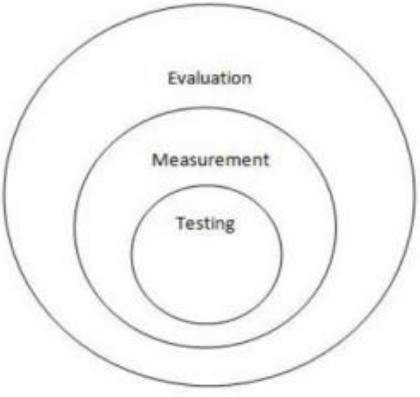
Workspace: Ini adalah file tersembunyi (disimpan di direktori kerja), di mana semua objek yang Anda gunakan (misalnya, data, matriks, vektor, variabel, fungsi, dan lain-lain.) disimpan. Perintah

yang berguna: ls() menampilkan semua elemen di ruang kerja kita saat ini dan rm(list=ls()) menghapus semua elemen di ruang kerja kita saat ini. Dimungkinkan juga untuk menghapus hanya beberapa objek dengan rm(object1, object2)

Catatan: Penting untuk diketahui, bahwa *R-Studio* tidak bisa dijalankan tanpa menginstal R-console terlebih dahulu, karena itu, jika Xanda ingin menggunakan *R Studio*, diwajibkan untuk menginstal *R-console* terlebih dahulu pada subbab 1.2.

1.6 Program R dan Pengukuran

Evaluasi, pengukuran, dan tes adalah aktivitas yang mempunyai keterkaitan antara satu dengan yang lainnya dalam dunia pendidikan. Tes merupakan alat pengukuran yang akan menjadi bahan evaluasi. Hubungan ketiga hal tersebut digambarkan dalam sebuah model oleh Lynch.



Gambar 1.12. Model Evaluasi, pengukuran dan tes (Lynch, 2001)

Umumnya, pengukuran berkaitan dengan penetapan data kuantitatif dengan menggunakan satu atau lebih instrumen seperti tes atau skala penilaian. Ketika dikontekstualisasikan dalam pendidikan, pengukuran dapat disebut sebagai proses yang digunakan untuk mengumpulkan tingkat kompetensi individu dalam hal numerik. Dengan kata lain, pengukuran dilakukan untuk mengukur tingkat pengetahuan atau keterampilan yang diperoleh seorang pelajar (Adom et al., 2020).

James M. Bradfield menekankan bahwa pengukuran adalah proses pemberian simbol pada dimensi suatu fenomena untuk mengkarakterisasi status fenomena setepat mungkin (Tripathi & Kumar, 2018). Ini berarti bahwa pengukuran memerlukan penundukan suatu variabel atau fenomena ke beberapa tolok ukur yang tepat dan dapat diukur. Scriven (1991) telah menolak bahwa pengukuran dilakukan untuk menentukan besaran suatu besaran. Penentuan ini biasanya dilakukan pada skala uji yang direferensikan kriteria atau pada skala numerik berkelanjutan. Instrumen pengukuran ini dapat mengambil berbagai bentuk seperti kuesioner, tes, atau perangkat apa pun. Scriven lebih lanjut mencatat: "Pengukuran adalah komponen umum dan terkadang besar dari evaluasi standar, tetapi bagian yang sangat kecil dari logikanya, yaitu, dari pembenaran untuk kesimpulan evaluatif."

Hasil pengukuran dengan berbagai alat tes (alat ukur) memerlukan pengolahan untuk mendapatkan simbol-simbol numerik tertentu. Aplikasi pengolah data hasil pengukuran yang saat ini mudah digunakan dan tersedia secara gratis adalah Program ***R***. Buku ini menjabarkan secara detail bagaimana ***R*** membantu anda dalam melakukan pengolahan data hasil pengukuran secara cepat, tepat dan mudah.

1.7 Paket Analisis (R Packages)

Ada banyak paket R yang berguna untuk berbagai aspek analisis data. Beberapa paket yang dapat membantu kita membuat plot yang indah dan yang mungkin ingin kita gunakan adalah paket *ggplot2* dan kumpulan paket-paket *rapiverse*. Kita dapat menginstal paket-paket penting sesuai keperluan dengan perintah ***install.packages('packagename')***. Untuk menggunakan jenis paket ***library(package)*** . Paket harus diinstal hanya sekali, tetapi Anda harus memuat paket dengan perpustakaan (paket) pada saat akan digunakan. Berikut ini adalah paket penting dalam analisis teori tes klasik, teori respons butir, *exploratory factor analysis*, *confirmatory factor analysis*, serta paket-paket yang digunakan untuk menganalisis atribut psikometri lainnya:

```
install.packages(c("CTT","psych","psychometric","Hmisc","readr
","nFactors","factoMineR", "lavaan", "semTools", "semPlot",
"readxl", "mirt"))
library(CTT) #Syntax teori tes klasik
library (psych) #Syntax atribut psikometri
library(psychometric) #Syntax atribut psikometri
library(Hmisc) #Analisis data, grafik tingkat tinggi, operasi
utilitas
library(readr) #Syntax untuk membaca data regular seperti
format .csv dan .txt
library(nFactors) #Syntax pelengkap analisis faktor
eksploratori untuk uji bartlett dan KMO
library(FactoMineR) #Syntax analisis faktor eksploratori
multivariat
library(readxl) #Syntax untuk membaca file dengan eksetensi
*xlsx
library(mirt) #Syntax item response theory
```

Referensi

Adom, D., Mensah, J. A., & Dake, D. A. (2020). Test, measurement, and evaluation: Understanding and use of the concepts in education. *International Journal of Evaluation and Research in Education (IJERE)*, *Vol. 9, No*, p. 109~119. https://doi.org/10.11591/ijere.v9i1.20457

Chalmers, P. (2022). *mirt: Multidimensional Item Response Theory*. https://cran.r-project.org/package=mirt

Epskamp, S. (2022). *semPlot: Path Diagrams and Visual Analysis of Various SEM Packages' Output*. https://github.com/SachaEpskamp/semPlot

Hafner, S. (2019). *An Introduction to R for Beginners*.

Harrell Jr., F. E. (2022). *Hmisc: Harrell Miscellaneous*. https://hbiostat.org/R/Hmisc/

Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2022). *semTools: Useful Tools for Structural Equation Modeling*. https://github.com/simsem/semTools/wiki

Lynch, B. K. (2001). Rethinking assessment from a critical perspective. *Language Testing*, *Vol.18 No.*, 351–372.

R Core Team. (2022a). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*.

R Core Team. (2022b). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/

Revelle, W. (2022). *psych: Procedures for Psychological, Psychometric, and Personality Research*. https://personality-project.org/r/psych/

Rosseel, Y. (2012). {lavaan}: An {R} Package for Structural Equation Modeling. *Journal of Statistical Software*, *48*(2), 1–36. https://doi.org/10.18637/jss.v048.i02

Rosseel, Y., Jorgensen, T. D., & Rockwood, N. (2022). *lavaan: Latent Variable Analysis*. https://lavaan.ugent.be

Scriven, M. (1991). *Evaluation Thesaurus* . *Thousand Oaks* ((4th ed.)). CA: Sage.

Tripathi, R., & Kumar, A. (2018). "Importance and Improvements in the Teaching-Learning process through Effective Evaluation Methodologies,." *ESSENCE Int. J. Env. Rehab. Conserv*, *Vol. 9, no*, 7–16.

Wickham, H., & Bryan, J. (2022). *readxl: Read Excel Files*. https://cran.r-project.org/package=readxl

Wickham, H., Hester, J., & Bryan, J. (2022). *readr: Read Rectangular Text Data*. https://cran.r-project.org/package=readr

Willse, J. T. (2018). *CTT: Classical Test Theory Functions*. https://cran.r-project.org/package=CTT

Chapter 2
Simulasi Data Dengan R

Oleh: Yudince Marinding & Sumin

R memiliki beberapa fungsi bawaan untuk pengambilan sampel dari berbagai jenis distribusi. Ini membuatnya sangat mudah untuk membuat data simulasi. Kita dapat mengakses fungsi-fungsi ini dengan mengetikkan "distribusi" ke bantuan R. Karena semua contoh yang melibatkan data simulasi juga akan melibatkan pengambilan sampel dari distribusi, ada baiknya meluangkan waktu untuk pengambilan sampel dari distribusi yang berbeda.

Ketika ingin melakukan simulasi analisis data, seseorang membutuhkan alat untuk dapat membangkitkan data sesuai patokan tertentu, misalnya dengan mean dan standart deviasi tertentu. Pembangkitan data dapat dilakukan menggunakan program R. Data yang dibangkitkan dapat berbentuk distribusi normal, gamma, poison, dan lain sebagainya. Sebagai contoh, distribusi normal adalah salah satu distribusi peluang kontinu yang grafiknya berupa kurva berbentuk genta/lonceng yang simetris, dengan parameter rata-rata dan simpangan baku (Tandililing & Ismail, 2021).

2.1 Membangkitkan Bilangan Acak (distribusi seragam)

Program R dapat membangkitkan data bilangan acak dengan jumlah tertentu. Pada subbab ini contoh awal yang disajikan adalah membangkitkan data dengan jumlah (n) tertentu dengan sebarang rata-rata dan standar deviasi.

R menghasilkan daftar angka acak antara nilai minimum dan maksimum. Salah satu definisi bilangan acak adalah gagasan tentang distribusi seragam. Pada distribusi seragam, semua angka yang mungkin dalam rentang memiliki peluang kemunculan yang sama. Kode berikut mengambil sampel 20 angka dari kisaran antara 0 dan 1, menggunakan fungsi runif. Sintaks untuk runif adalah:

```
round(runif(20, min=0, max=1),3)
[1] 0.106 0.832 0.312 0.705 0.945 0.728 0.250 0.679 0.188 0.283
[11] 0.869 0.335 0.503 0.296 0.530 0.956 0.650 0.797 0.114 0.602
```

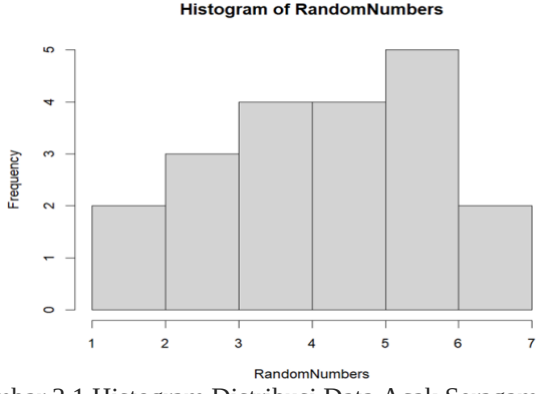
dimana; n adalah jumlah sampel. Min dan max adalah batas bawah dan atas dari distribusi. Berikut adalah contoh lain pengunaan fungsi runif.

```
RandomNumbers <- round(runif(20, min=1, max=7),3)
RandomNumbers
[1] 5.032 1.996 4.687 3.823 6.049 4.502 4.215 4.370 4.349 2.379
[11] 2.375 3.761 1.401 6.956 1.130 3.528 3.829 2.511 3.387 4.304
```

Syntax di atas berfungsi untuk mengambil sampel 20 angka antara 1 dan 7 dan menempatkannya ke dalam variabel bernama RandomNumbers. Jika kita mengubah nilai dalam fungsi, kita dapat membuat sampel angka dengan ukuran apa pun, dan membuat rentang ukuran apa pun.

Anda juga dapat dengan mudah untuk melihat angka-angka ini secara grafis. Misalnya, Anda mungkin melihat apakah sampel Anda benar-benar terlihat seperti distribusi seragam. Cara untuk melakukannya antara lain dengan membuat histogram yang menunjukkan frekuensi setiap angka menurut ukuran bin yang berbeda (misalnya, frekuensi angka antara 0 dan 10, 11 dan 20, dan seterusnya). Jika sampel adalah distribusi seragam, maka batang histogram harus kira-kira sama tingginya (misalnya, datar dan seragam). Histogram dapat dibuat menggunakan fungsi hist.

```
RandomNumbers <- round(runif(20, min=1, max=7),3)
hist(RandomNumbers)
```



Gambar 2.1 Histogram Distribusi Data Acak Seragam, n=20

Perhatikan bahwa beberapa batang grafik histogram menunjukkan nilai yang lebih tinggi dari yang lain, tetapi secara keseluruhan distribusinya relatif datar dan seragam. Fakta bahwa distribusi tidak seragam sempurna menunjukkan bahwa distribusi sampel tidak selalu terlihat persis seperti distribusi induk. Variabilitas dalam sampel ini disebabkan oleh kebetulan. Efek kebetulan dapat dieliminasi dengan cara menggunakan sampel yang lebih besar. Misalnya, kode di atas dapat dimodifikasi untuk mengambil sampel 10.000 angka daripada 100 angka. Sampel yang lebih besar ini seharusnya menghasilkan histogram yang menunjukkan distribusi yang lebih datar dan seragam. Jika kita gunakan sampel besar, misal: n=10000, maka akan menghasilkan beberapa variabilitas dalam ketinggian batang histogram, tetapi secara keseluruhan sampel yang lebih besar ini menunjukan bahwa data hasil simulasi memiliki distribusi yang lebih datar dan seragam. Sekali lagi, ini berarti bahwa setiap angka dalam distribusi memiliki peluang kemunculan yang sama. Semua distribusi lain yang tercakup di sini melibatkan distribusi yang tidak seragam, di mana setiap nilai dalam populasi memiliki kemungkinan kemunculan yang berbeda.

2.2 Membangkitkan Data Berdistribusi Normal

Distribusi normal (atau kurva lonceng) dapat dijadikan sampel dengan menggunakan fungsi rnorm. Sintaks untuk rnorm adalah:

```
Data_normal <- rnorm(100, mean = 0, sd = 1)
round(Data_normal,3)
  [1]  1.285  0.946  0.150  0.859 -2.017 -1.215  1.045  0.378
  [9] -0.136 -0.831 -2.115 -2.214  1.438  0.433 -0.176 -0.208
 [17]  0.681 -0.038 -1.218 -0.975  0.678 -0.617 -0.347 -0.622
 [25] -1.365 -0.456  0.253 -0.029  0.765  0.868  1.228  0.620
 [33] -0.972  0.776 -0.455  0.576 -0.621 -1.923 -0.125 -0.576
 [41] -0.442  0.836 -1.340 -0.385  0.596 -0.269  1.269  0.793
 [49]  0.094 -0.078  0.352 -1.413 -0.801 -0.821  0.388 -0.372
 [57] -1.375 -2.040 -0.013  0.972 -0.897  0.429  0.268  0.788
 [65]  1.032  0.758 -0.141  0.664 -0.884 -0.360 -0.489 -0.110
 [73] -0.282 -0.254  0.784 -0.076 -0.319 -0.421 -1.239  0.830
 [81]  0.171  0.719 -0.252  0.170 -1.324  1.518  0.195 -0.736
 [89] -0.727  0.422  0.783 -0.693  1.321 -0.866 -0.219  0.288
 [97] -0.329 -1.804 -0.407  0.615
```

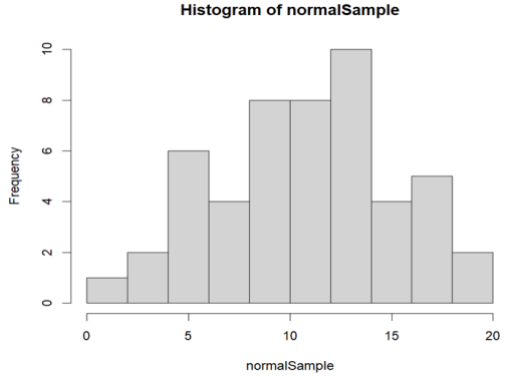
n adalah jumlah sampel, mean adalah pusat distribusi, dan sd adalah standar deviasi dari distribusi. Mari kita coba dengan mengambil

100 sampel dari distribusi normal dengan mean = 100, dan sd = 50, dan plot hasilnya dalam histogram.

```
normalSample <- rnorm(100, mean = 100, sd = 50)
round(normalSample,3)
```

```
 [1] 12.920 12.848 13.674 14.387 10.684 11.725 13.920 16.723 10.004
[10]  1.703  6.568 15.406 13.860  3.976  8.480  5.489 10.044 16.355
[19] 19.084  7.924 13.393 18.710  9.708  9.461 14.897  8.697  9.560
[28]  6.820 11.977  4.726 11.245 16.381  6.852 17.040 12.459  4.509
[37]  4.443  8.710  9.604 14.360 11.114  5.925 13.902 12.910  5.722
[46]  3.378 16.384 10.560 12.027  9.562
```

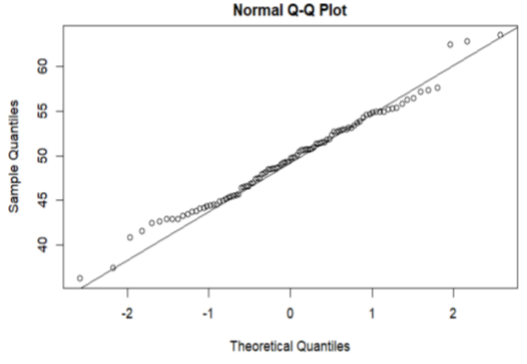
```
hist(normalSample)
```



Gambar 2.1 Histogram Distribusi Data Acak Normal, n=100

Histogram di atas menunjukkan kurva seperti lonceng yang khas dari distribusi normal. Distribusi data yang dibangkitkan dapat pula ditampilkan dalam bentuk Q-Q Plot dengan perintah sebagai berikut.

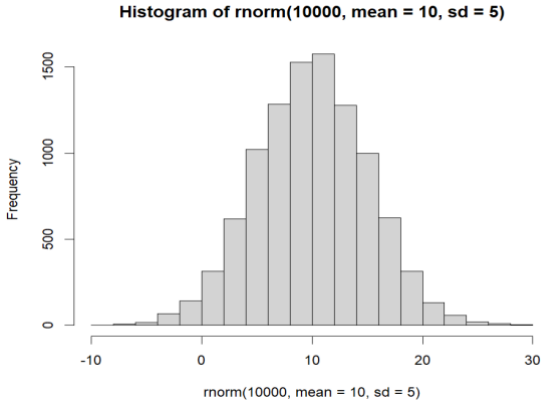
```
qqnorm(normalSample)
qqline(normalSample)
```



Gambar 2.2. Q-Q Plot Data normalSample

Jika kita mengambil sampel yang jauh lebih besar dari 10.000 angka, kurva seperti lonceng akan menjadi lebih jelas. Perintah yang digunakan masih rnorm seperti membangkitkan data berdistirbusi normal sebelumnya. Jumlah data yang diinginkan diubah menjadi 1000 seperti berikut.

```
hist(rnorm(10000, mean = 10, sd = 5))
```



Gambar 2.3 Histogram Distribusi Data Acak Normal, n=10.000

2.3 Membangkitkan Data Berdistribusi Poisson

Distribusi Poisson biasanya digunakan untuk memodelkan jumlah peristiwa yang diharapkan untuk suatu proses. Kita mengetahui tingkat rata-rata di mana peristiwa terjadi selama satuan waktu tertentu. Model Poisson sering digunakan untuk regresi Poisson, regresi logistik, dan fungsi massa probabilitas Poisson.

Distribusi Poisson umumnya digunakan dalam industri dan sains. Contoh klasik dari distribusi Poisson adalah jumlah tentara Prusia yang secara tidak sengaja terbunuh oleh tendangan kuda, karena menjadi contoh pertama dari aplikasi distribusi Poisson ke kumpulan data besar dunia nyata. Sepuluh korps tentara diamati selama 20 tahun, dengan total 200 pengamatan, dan 122 tentara terbunuh oleh tendangan kuda selama periode waktu itu. Pertanyaannya adalah berapa banyak kematian yang diharapkan selama periode satu tahun, yang ternyata dimodelkan dengan sangat baik oleh distribusi variabel acak Poisson.

Beberapa contoh lain misalnya: (1) jumlah rata-rata kegagalan peralatan per hari untuk perusahaan logistik jumlah rata-rata pelanggan yang tiba di pengecer; (2) jumlah pengunjung ke situs web; (3) jumlah panggilan telepon masuk; dan (4) jumlah keluhan pelanggan.

Data berdistribusi Poison dapat dibangkitkan dengan perintah sebagai berikut.

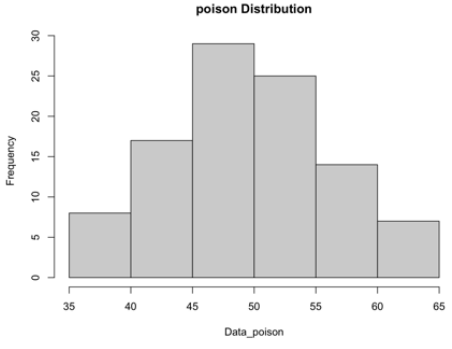
```
#banyaknya data = 100, mean=lamda=50
Data_poison<-rpois(100,50)
```

Setelah didefinisikan, data yang dibangkitkan dapat ditampilan dengan menulis kembali nama data yang elah didefinsikan sebagai berikut.

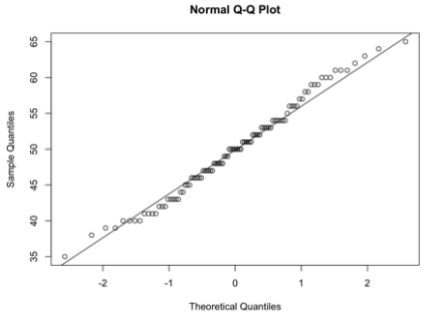
```
Data_poison
  [1] 61 60 40 55 61 52 51 53 47 48 44 40 50 54 37 43 42 54 50
 [20] 48 45 61 70 55 54 46 52 55 42 45 55 42 41 51 46 46 47 60
 [39] 52 55 48 33 46 29 49 49 48 42 53 50 40 53 46 53 43 43 56
 [58] 53 50 53 45 55 47 60 50 57 48 49 49 38 44 53 66 34 52 42
 [77] 44 36 53 48 46 50 49 54 36 52 48 49 48 48 52 53 56 50 53
 [96] 48 47 62 51 49
```

Seperti pada distribusi data yang telah dibangkitkan sebelumnya, data distribusi Poisson yang sudah dibangkitkan dapat ditampilkan histogram dan Q-Q Plot nya dengan perintah sebagai berikut.

```
hist(Data_poison, main = "poison Distribution")
qqnorm(Data_poison)
qqline(Data_poison)
```



Gambar 2.4 Histogram Distribusi Poison, n=100



Gambar 2.5 Histogram dan Q-Q Plot Distribusi Poisson, n=100

2.4 Membangkitkan Data Berdistribusi Gamma

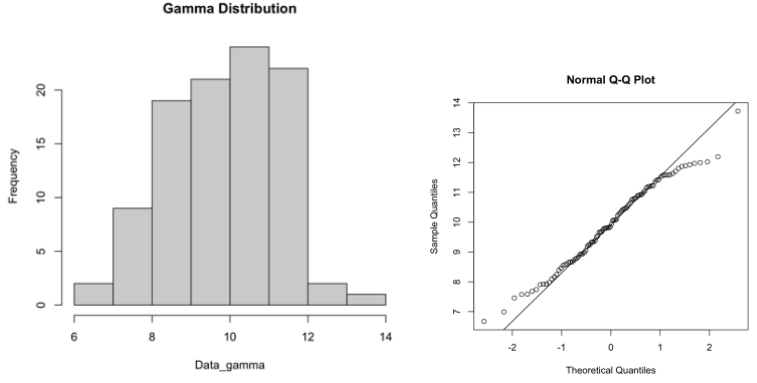
Berikut ini sintaks untuk membangkitkan datanya pada R Studio.

```
#membangkitkan data berdistribusi gamma
#sebanyak 100 data, aplha=50, dan beta=5
Data_gamma<-rgamma(100,50,5)
Data_gamma
  [1]  9.789114  9.241388  8.932983 11.803693  8.589609 10.549599
  [7]  9.944328  8.094237 11.575907 11.527349  8.987212  8.392635
 [13] 11.891729  9.390308 10.995110 10.819087  7.738859  8.457498
 [19]  6.672555 11.635076 11.415229 10.481808 11.586719 11.362252
 [25]  9.335067 11.866461 11.428366 10.599875 11.220192  9.809800
 [31]  8.835048 10.468083 10.226708  7.923283  8.918193 11.995307
 [37]  8.940256  9.232320  7.574738 13.720628 10.887709  7.978836
 [43] 11.970606  9.848558 10.093515  9.184513  9.345409  8.715967
 [49]  7.904543  8.653362 10.059559  7.460132 10.432306 10.063668
 [55] 10.277641 10.752801  8.790413 10.925429  8.764682  9.675386
 [61]  8.661751 11.703284  9.818335 11.225016  9.504648  9.052029
 [67]  8.168854  8.549111 12.196287 10.071137  9.796582 10.921414
 [73]  9.659483 10.655289  7.924224 11.572218  7.682708  9.813068
 [79]  9.736809 11.925692 11.196990  6.991946 10.779568  8.663481
 [85] 10.423337 10.791498  7.585827 11.048283 10.314007 11.576884
 [91] 11.152324 12.020344  9.321873  9.668007  8.244929 10.895787
 [97]  8.574378 11.190038  9.542484 10.368267
```

Diperoleh daat sebanyak 100 dengan alpha=50, dan beta=5. Rata-rata data yang berdistribusi Gamma yang sudah diperoleh dapat dihitung dengan rumus alpha/ beta = 10. Nilai rerata yang dihitung berdasarkan alpha dan beta dapat dibuktikan dari data yang diperoleh dengan menggunakan perintah sebagai berikut.

```
mean(Data_gamma)
[1] 9.927418
```

Terlihat bahwa data bangkitan yang diperoleh menghasilkan rerata = 10 sebagaimana jika dihitung dari alpha dan beta. Histriogram dan QQ Plot Data_gamma dapat pula ditampilkan sebagaimana distribusi data sebelumnya.



Gambar 2.6 Histogram dan Q-Q Plot Data_Gamma (alpha=50, beta=5)

Referensi

http://personality-project.org/r/book/
https://bookdown.org/rdpeng/rprogdatascience/simulation.html
https://rdrr.io/cran/catIrt/man/simIrt.html
R Core Team. (2022a). *R: A language and environment for statistical computing. R Foundation for Statistical Computing.*
R Core Team. (2022b). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/
Tandililing, P., Ismail, R. (2021). Pengantar Statistika Terapan. Purwokerto: Pena Persada.

Chapter 3
Teori Tes Klasik dengan R

Oleh: Johan Syahbrudin, Devi Dwi Kurniawan, & Samsul Hadi

3.1 Sekilas tentang Teori Tes Klasik

Pada bab ini, kami memperkenalkan pengukuran pendidikan dan psikometri menggunakan kerangka teori tes klasik (CTT) yang diaplikasikan melalui Program R. Pembahasan dimulai dengan mendefinisikan pengukuran, tes, skala pengukuran, validitas, dan reliabilitas dalam konteks CTT. Setelah pengenalan singkat, menyajikan beberapa contoh bagaimana menghitung banyak statistik berbasis CTT menggunakan program R.

Pengukuran merupakan kuantifikasi konstruk dengan menetapkan nilai numerik atau label kualitatif berdasarkan seperangkat aturan, prinsip, atau operasi. Misalnya, seorang guru matematika, ingin mengukur pengetahuan aljabar siswanya. Untuk menyelesaikan tugas ini, guru memutuskan untuk merancang dan menggunakan tes matematika yang mengukur pengetahuan aljabar siswanya. Selama pengembangan tes matematikanya, guru menghadapi beberapa tantangan pengukuran: Konsep apa dalam aljabar yang harus diketahui siswanya? Berapa banyak pertanyaan yang harus digunakannya? Berapa banyak pertanyaan yang harus saya sertakan per konsep, dan apa format pertanyaannya? Apakah perlu menulis pertanyaan baru atau dapatkah menggunakan pertanyaan dari tes yang sudah ada sebelumnya?

Instrumen pengukuran seperti tes, skala, survei, dan kuesioner adalah alat yang digunakan guru, dosen, psikolog, praktisi, atau peneliti, untuk mengukur konstruk, sifat, atau domain yang diminati. Penyusunan instrumen pengukuran dengan tujuan tertentu, seperti mengembangkan tes matematika untuk mengukur pengetahuan aljabar. Sebuah konstruksi, sifat, atau domain adalah konsep teoritis, seperti kemampuan membaca, kecerdasan, atau fungsi eksekutif. Dalam pendidikan dan psikologi, biasanya tidak dapat mengukur konstruksi secara langsung karena bersifat laten (tidak teramati). Untuk mengukur

konstruk laten, mengembangkan dan menggunakan item dalam instrumen pengukuran. Item ini biasanya disebut sebagai "variabel manifes" yang memberikan definisi operasional dari konstruk laten yang diukur. Konstruksi laten kadang-kadang didefinisikan secara luas, seperti kecerdasan, atau mungkin lebih sempit, seperti pengetahuan tentang fungsi trigonometri dalam matematika. Perbedaan antara istilah seperti tes, instrumen, dan survey tidak penting dalam kerangka dan konsep pengukuran yang akan dijelaskan dapat digeneralisasi untuk semua jenis alat pengukuran.

3.2 Karakteristik Teori Tes Klasik

Teori tes klasik (CTT), juga disebut teori skor murni, adalah kerangka kerja pengukuran yang memungkinkan untuk memahami, memanipulasi, dan menginterpretasikan hasil yang diukur. Premis CTT adalah bahwa setiap pengukuran mengandung kesalahan. Model CTT menguraikan skor yang diamati dari instrumen pengukuran menjadi komponen skor benar dan salah. Secara matematis, CTT dinyatakan sebagai:

$$X = T + E \tag{3.1}$$

Dalam model CTT, X adalah skor pengukuran/tes yang dapat diamati; T adalah pengukuran/nilai tes total (laten) yang sebenarnya; dan E adalah kesalahan acak. Ketika fokusnya adalah pada item individual dari pada skor total yang tidak dapat diamati. Untuk menggunakan model CTT, ada empat asumsi tambahan di luar bentuk umum yang disajikan dalam Persamaan berikut:

1. E (X) = T, nilai yang diharapkan dari skor yang diamati adalah skor benar,
2. Cov(T;E) = 0, skor benar dan kesalahan saling bebas,
3. Cov(E1;E2) = 0, kesalahan di seluruh bentuk tes saling bebas, dan
4. Cov(E1; T2) = 0, kesalahan pada satu bentuk tidak bebas pada skor sebenarnya pada bentuk lain.

Karena asumsi ini, model CTT dapat dinyatakan kembali sebagai jumlah sederhana dari komponen varians ortogonal (yaitu, tidak berkorelasi) seperti yang ditunjukkan di bawah ini:

$$\sigma_X^2 = \sigma_T^2 + \sigma_E^2 \tag{3.2}$$

Persamaan di atas menyatakan bahwa varians skor yang diamati adalah jumlah dari varians skor benar dan kesalahan. Dalam model ini, varians skor sebenarnya diasumsikan konstan (misalnya, tidak akan pernah berubah terlepas dari bentuk instrumen, waktu penilaian, dll.), sedangkan varians kesalahan berfluktuasi (misalnya, beberapa bentuk mungkin mengandung lebih banyak kesalahan. daripada yang lain). Kesalahan pengukuran dapat dibagi menjadi kesalahan acak (tidak dapat diprediksi dan tidak konsisten) dan sistematis (konstan dan dapat diprediksi).

3.3 Syarat Instrumen Tes yang baik

3.3.1 Validitas

Validitas mengacu pada sejauh mana kita mengukur apa yang ingin kita ukur. Validitas bertujuan untuk menyelidiki apakah variabel manifes adalah manifestasi sebenarnya dari konstruk target atau sesuatu yang lain. Validitas melibatkan pengumpulan bukti dan membangun kasus yang mendukung penggunaan instrumen pengukuran.

Bukti validitas mengambil banyak bentuk, dan mengukur bukti validitas cukup mudah dengan R. Bentuk umum dari bukti validitas adalah pendapat ahli. Pendapat para ahli dapat membantu mengomentari kesesuaian isi item, apakah instrumen tersebut cukup mengambil sampel semua konsep dalam konstruk, dan apakah item penting untuk mengukur target konstruk.

3.3.1.1 CVR Lawse

Salah satu cara untuk mengukur yang terakhir adalah dengan rasio validitas isi, CVR (Lawshe, 1975). CVR didefinisikan sebagai:

$$CVR = \frac{n_e - (N/2)}{N/2} \tag{3.3}$$

dimana n_e adalah jumlah ahli yang menganggap item tersebut penting dan N adalah jumlah ahli. Misalnya, membuat instrumen yang menanyakan kepada 20 ahli apakah menurut ahli item ini penting untuk mengukur. 17 orang setuju, maka CVR dapat dihitung menggunakan fungsi CRV pada R.

```
N <- 20
ne <- 17
cvr <- (ne-(N/2))/(N/2)
cvr
```

Hasil kalkulasi diperoleh CVR = 0,70 untuk item ini. Lawshe (1975) memberikan nilai ambang batas CVR yang diberikan sejumlah ahli. Untuk 20 ahli, minimum CVR adalah 0,42 dan akan menyimpulkan bahwa para ahli menganggap item ini sangat penting, dan kemungkinan mempertahankan item ini di instrumen.

Jika ingin menghitung sekaligus untuk banyak *rater* dan banyak butir, maka menggunakan sintaks berikut ini.

```
library(xlsx)
data <- read.xlsx('Lawshe.xlsx', 1, header= T)
data
```

```
#OUTPUT
Rater          Butir_1 Butir_2 Butir_3 Butir_4 Butir_5
Rater_1        1       1       3       2       1
Rater_2        2       3       2       3       2
Rater_3        1       1       2       3       1
Rater_4        2       3       1       2       3
Rater_5        2       3       1       2       1
Rater_6        1       3       1       2       1
Rater_7        1       3       1       3       1
```

```
data <- data[ , -1]
data3 <- ifelse(data == 3, 1, 0)
ne <- colSums(data3)
cvr <- data.frame((ne - (nrow(data3)/2))/(nrow(data3)/2))
colnames(cvr)<- 'cvr'
cvr
```

```
               CVR
Butir_1        -1.000000
Butir_2        0.4285714
Butir_3        -0.7142857
Butir_4        -0.1428571
Butir_5        -0.7142857
```

3.3.1.2 Indeks Aiken (V)

Validitas isi dapat juga dibuktikan menggunakan indeks Aiken. Rentang indeks Aiken (V) adalah 0 sampai dengan 1. Semakin tinggi indeks V, semakin tinggi validitas isi. Jika 1 butir dinilai oleh n rater, maka indeks V dapat dihitung dengan persamaan berikut.

$$V = \frac{\sum s}{[n(c-1)]} \quad (3.4)$$

dengan V adalah indeks Aiken untuk butir ke-i, s = penilaian rater (r) – nilai untuk kategori terendah (lo), c = nilai untuk kategori tertinggi (hi); dan *n* adalah jumlah *rater* yang menilai butir ke-i.

Apabila *m* butir dinilai oleh satu orang rater, maka persamaan 3.4 dapat diubah menjadi persamaan 3.5.

$$V = \frac{\sum s}{[m(c-1)]} \quad (3.5)$$

Contoh perhitungan indeks V pada program R berikut ini menggunakan data bangkitan dengan menggunakan fungsi runif. Data yang dibangkitkan terdiri dari 7 data, dengan nilai minimum = 1, dan nilai maksimum = 5. Pembaca mungkin akan mendapatkan data yang berbeda dari yang digunakan pada contoh ini, karena perintah runif menghasilkan angka acak. Setiap kali perintah runif dijalankan akan menghasilkan data yang berbeda.

```
lo <- 1 #Kategori terendah
hi <- 5 #Kategori tertinggi
data <- data.frame(round(runif(7, lo, hi)))
colnames(data) <- 'Butir_1'
data
```

```
  Butir_1
1       5
2       2
3       2
4       3
5       1
6       3
7       3
```

Selanjutnya, persamaan 3.4 digunakan untuk menghitung indeks Aiken untuk butir 1. Nilai *n*, dan *s* dihitung terlebih dahulu untuk memudahkan proses kalkulasi dengan perintah sebagai berikut. Jumlah baris (nrow) digunakan untuk menghitung banyaknya rater pada butir 1, dan diperoleh hasilnya adalah 7. Nilai s dihitung dengan mengurangkan semua data dengan 1 karena nilai kategori terendah adalah 1. Sedangkan c nilai untuk kategori tertinggi penilaian *rater*.

```
n <- nrow(data)
n
```

```
[1] 7
```

```
s <- data - lo
s
```

```
  Butir_1
1       4
2       1
3       1
4       2
5       0
6       2
7       2
```

```
sigma_s <- colSums(s)
sigma_s
```

```
Butir_1
     12
```

```
c <- hi
c
```

```
[1] 5
```

```
V <- data.frame(sigma_s/(n*(c-1)))
colnames(V) <- "Aiken's V"
V
```

```
        Aiken's V
Butir_1 0.4285714
```

Hasil perhitungan diperoleh indeks V Aiken untuk butir 1 sebesar 0.429. Nilai ini masih belum memenuhi kriteria minimum yang digunakan oleh Aiken untuk butir yang dinilai oleh 7 *rater* dengan jumlah kategori sebanyak 5. Pada Gambar 3.1 ditunjukkan nilai minimum indeks V yang dapat diterima untuk 7 *rater* dengan 5 kategori penilaian adalah 0.75.

TABLE 1

Right-Tail Probabilities (p) for Selected Values of the Validity Coefficient (V)

No. of Items (*m*) or Raters (*n*)	Number of Rating Categories (c) 2		3		4		5		6		7	
	V	p	V	p	V	p	V	p	V	p	V	p
2							1.00	.040	1.00	.028	1.00	.020
3							1.00	.008	1.00	.005	1.00	.003
3			1.00	.037	1.00	.016	.92	.032	.87	.046	.89	.029
4					1.00	.004	.94	.008	.95	.004	.92	.006
4			1.00	.012	.92	.020	.88	.024	.85	.027	.83	.029
5			1.00	.004	.93	.006	.90	.007	.88	.007	.87	.007
5	1.00	.031	.90	.025	.87	.021	.80	.040	.80	.032	.77	.047
6			.92	.010	.89	.007	.88	.005	.83	.010	.83	.008
6	1.00	.016	.83	.038	.78	.050	.79	.029	.77	.036	.75	.041
7			.93	.004	.86	.007	.82	.010	.83	.006	.81	.008
7	1.00	.008	.86	.016	.76	.045	.75	.041	.74	.038	.74	.036
8	1.00	.004	.88	.007	.83	.007	.81	[illegible]	.80	.007	.79	.007
8	.88	.035	.81	.024	.75	.040	.75	.030	[illegible]			
9	1.00	.002	.89	.003	.81	.007	.81	.006	[illegible]			
9	.89	.020	.78	.032	.74	.036	.72	.038	[illegible]			
10	1.00	.001	.85	.005	.80	.007	.78	.008	[illegible]			
10	.90	.001	.75	.040	.73	.032	.70	.047	.70	.039	.68	.048

Indeks Aiken minimum yang disarankan

Gambar 3.1 Tabel rujukan untuk memeriksa Indeks Aiken (V) 7 rater, 5 kategori (Aiken, 1985)

Kalkulasi Indeks Aiken (V) pada program R dapat pula dilakukan untuk beberapa butir. Contoh berikut menggunakan data yang telah disimpan pada direktori dengan nama “Aiken.csv”.

```
data <- read.csv('Aiken.csv', header = T, sep = ';')
data
```

```
    Rater Butir_1 Butir_2 Butir_3 Butir_4 Butir_5 Butir_6 Butir_7 Butir_8 Butir_9 Butir_10
1 Rater_1       5       1       2       1       3       1       2       2       3        4
2 Rater_2       3       3       4       4       4       4       4       1       3        2
3 Rater_3       2       1       1       4       1       3       3       5       3        3
4 Rater_4       4       1       2       3       3       3       5       5       3        3
5 Rater_5       2       4       2       2       1       4       5       3       1        2
```

Selanjutnya, seperti halnya perhitungan indeks Aiken untuk 1 butir oleh 7 rater (contoh pertama), proses perhitungan Indeks Aiken (V) untuk 10 butir ini dilakukan dengan persamaan 3.4. Pada program R, perhitungan diawali dengan mendefinisikan nilai n, c, dan menghitung s dari data.

```
data <- data[,-1]
n <- nrow(data)
s <- data - 1
sigma_s <- colSums(s)
c <- max(data)
```

Setelah seluruh komponen yang dibutuhkan didefinisikan/dihitung, Indeks V dapat dihitung dengan cara berikut.

```
V <- data.frame(sigma_s/(n*(c-1)))
colnames(V) <- "Aiken's V"
V
```

```
         Aiken's V
Butir_1       0.55
Butir_2       0.25
Butir_3       0.30
Butir_4       0.45
Butir_5       0.35
Butir_6       0.50
Butir_7       0.70
Butir_8       0.55
Butir_9       0.40
Butir_10      0.45
```

Berdasarkan hasil ini, Indeks V yang dihasilkan, tampak bahwa nilainya belum ada yang berada diatas ambang batas minimum 0.80 (lihat Gambar 3.2). Butir 7 adalah butir yang mendapatkan penilaian yang paling tinggi diantara seluruh butir lainnya. Sementara itu, Butir 2 adalah yang mendapatkan penilaian yang paling rendah.

TABLE 1

Right-Tail Probabilities (p) for Selected Values of the Validity Coefficient (V)

No. of Items (m) or Raters (n)	Number of Rating Categories (c) 2		3		4		5		6		7	
	V	p	V	p	V	p	V	p	V	p	V	p
2							1.00	.040	1.00	.028	1.00	.020
3							1.00	.008	1.00	.005	1.00	.003
3				.037	1.00	.016	.92	.032	.87	.046	.89	.029
4					1.00	.004	.94	.008	.95	.004	.92	.006
4			1.00	.012	.93	.020	.88	.024	.85	.027	.83	.029
5			1.00	.004	.93	.006	.90	.007	.88	.007	.87	.007
5	1.00	.031	.90	.025	.87	.021	.80	.040	.80	.032	.77	.047
6			.92	.010	.89	.007	.88	.005	.83	.010	.83	.008
6	1.00	.016	.83	.038	.78	.050	.79	.029	.77	.036	.75	.041
7			.93	.004	.86	.007	.82	.010	.83	.006	.81	.008
7	1.00	.008	.86	.016	.76	.045	.75	.041	.74	.038	.74	.036
8	1.00	.004	.88	.007	.83	.007	.81	.008	.80	.007	.79	.007
8	.88	.035	.81	.024	.75	.040	.75	.030	.72	.039	.71	.047
9	1.00	.002	.89	.003	.81	.007	.81	.006	.78	.009	.78	.007
9	.89	.020	.78	.032	.74	.036	.72	.038	.71	.039	.70	.040
10	1.00	.001	.85	.005	.80	.007	.78	.008	.76	.009	.75	.010
10	.90	.001	.75	.040	.73	.032	.70	.047	.70	.039	.68	.048
11	.91	.006	.82	.007	.79	.007	.77	.006	.75	.010	.74	.009
11	.82	.033	.73	.048	.73	.029	.70	.035	.69	.038	.68	.041
12	.92	.003	.79	.010	.78	.006	.75	.009	.73	.010	.74	.008
12	.83	.019	.75	.025	.69	.046	.69	.041	.68	.038	.67	.049
13	.92	.002	.81	.005	.77	.006	.75	.006	.74	.007	.72	.010
13	.77	.046	.73	.030	.69	.041	.67	.048	.68	.037	.67	.041

Indeks Aiken minimum yang disarankan

Gambar 3.2 Tabel rujukan untuk memeriksa Indeks Aiken (V) 5 rater, 5 kategori (Aiken, 1985)

3.3.1.3 Validitas Kriteria

Bentuk lain dari bukti validitas mengukur sejauh mana skor tes berhubungan dengan beberapa kriteria eksternal (bukti validitas terkait kriteria). Dukungan statistik untuk bukti validitas ini mungkin melibatkan penghitungan korelasi sederhana atau penggunaan regresi. Kembali ke kumpulan data minat, berharap bahwa tes kosakata (*vocab*) akan dikorelasikan dengan penilaian yang mengukur pemahaman bacaan (*reading*) dan penyelesaian kalimat (*sentcomp*). Dengan demikian, kita dapat menggunakan fungsi cor untuk menghitung korelasi Pearson antara variabel-variabel ini.

```
cor(interest[, c("vocab", "reading", "sentcomp")])
             Vocab  reading  sentcomp
vocab    1.0000000 0.8030912 0.8132765
reading  0.8030912 1.0000000 0.7252155
sentcomp 0.8132765 0.7252155 1.0000000
```

Korelasi Pearson antara kosakata (*vocab*) dan membaca (*reading*) adalah 0,803, sedangkan korelasi antara kosakata (*vocab*) dan kalimat (*sentcomp*) adalah 0,813. Ini akan mewakili bukti validitas bersamaan jika tes kosakata diberikan pada saat penilaian pembacaan dan penyelesaian kalimat. Jika tes kosakata mendahului tes membaca dan

penyelesaian kalimat, maka korelasi Pearson mewakili bukti yang mendukung validitas prediktif.

Asumsikan bahwa mengukur minat seseorang untuk menjadi guru menggunakan ukuran kepribadian dari dominasi sosial (socdom) dan ingin memahami jika ada kemampuan prediktif tambahan dalam memberikan penilaian pemahaman bacaan di luar ukuran kepribadian saja. Mencoba menilai apakah pemahaman bacaan memiliki validitas tambahan, dan itu dapat dinilai dalam kerangka regresi menggunakan regresi bertahap. Melakukan ini dengan memasang dua model regresi linier. Model pertama (mod_lama) meregresi guru (teacher) di dominasi sosial (socdom), sedangkan model kedua (mod_new) meregresi guru (teacher) di dominasi sosial (socdom) dan membaca (reading). Sebuah model regresi linier dapat diperkirakan menggunakan fungsi lm di R. Fungsi ini memerlukan penentuan rumus regresi dengan variabel terikat (guru) diprediksi oleh satu atau lebih variabel bebas (dominasi sosial dan membaca), dan nama kumpulan data yang mencakup variabel tersebut minat (interest).

```
mod_old <- lm(teacher ~ socdom, interest)
mod_new <- lm(teacher ~ socdom, + reading, interest)
```

Model yang dihasilkan model baru (mod_new) menyertakan variabel dependen dan independen yang sama yang digunakan dalam model lama (mod_old) sementara juga menyertakan variabel pembacaan. Untuk menguji kontribusi membaca di luar socdom dalam memprediksi guru, dapat mengekstrak perubahan nilai R-kuadrat (R^2) antara kedua model. Selain itu, dapat membandingkan model secara statistik menggunakan fungsi anova di R. Tes ini menguji apakah perubahan R^2 antara kedua model signifikan secara statistik (yaitu, $R^2 > 0$). Jika perubahan R^2 signifikan secara statistik, maka dapat disimpulkan bahwa membaca menjelaskan sejumlah variasi yang signifikan dalam variabel dependen guru di luar dominasi sosial (socdom).

```
summary(mod_new)$r.squared – summary(mod_old)$r.squared
anova(mod_old, mod_new)
[1] 0.09125979
Analysis of Variance Table
Model 1: teacher ~ socdom
```

```
Model 2: teacher ~ socdom + reading
  Res.Df    RSS  Df  Sum of Sq       F     Pr(>F)
1    248 244.98
2    247 221.03   1     23.951  26.765  4.854e-07  ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Dari *output* di atas, terlihat bahwa penilaian pemahaman bacaan memiliki validitas tambahan di luar ukuran dominasi sosial saja ($p < 0.001$) dan itu menjelaskan sekitar 9% lebih banyak variabilitas minat dalam profesi guru.

3.3.2 Reliabilitas

Alat ukur yang yang mempunyai reliabilitas yang baik akan memberikan hasil pengukuran yang stabil (Lawrence, 1994) dan konsisten (Mehrens & Lehmann, 1973). Tes dikatakan reliabel apabila skor amatan mempunyai korelasi yang tinggi dengan skor sebenarnya (Allen & Yen, 1979). Semakin besar reliabilitas suatu instrumen, maka akan semakin kecil tingkat kesalahan pengukuran, begitu pula sebaliknya, sehingga reliabilitas juga berkaitan dengan kesalahan pengukuran.

Estimasi merupkan roses penghitungan reliabilitas. Sehingga dalam mengestimasi reliabilitas tes dapat ditempuh dengan beberapa cara, yaitu reliabilitas konstruk, reliabilitas komposit, reliabilitas interrater, konsistensi internal, konsistensi eksternal, dan estimasi reliabilitas dengan menggunakan teori generalisabilitas.

3.3.2.1 Konsistensi Eksternal

a. Metode Tes Ulang (Test-Retest-Method)

Pada metode ini, pengukuran perlu dilakukan dua kali ulangan. Dapat dilakukan oleh orang yang sama atau berbeda. Pada pengukuran kedua, kondisi yang diukur harus berada pada kondisi yang sama dengan pengukuran pertama. Sehingga kedua hasil pengukuran tersebut dapat dikorelasikan sehingga menghasilkan reliabilitas skor dari perangkat pengukuran. Metode ini cocok digunakan apabila kondisi dari subjek yang diberikan pengukuran stabil atau tidak terjadi perubahan pada saat kedua pengukuran dilakukan, sedangkan keadaan

responden tidak statis atau selalu berkembang, sehingga teknik ini kurang tepat digunakan untuk penelitian psikologi.

Karena kita ingin mengkorelasikan hasil pengukuran yang pertama (X) dan yang kedua (Y). Maka untuk mengestimasi reliabilitasnya yaitu dengan menghitung koefisien korelasi linear (r_i) antara skor pengukuran X dan Y melalui persamaan:

$$r_i = \frac{n\Sigma X_i Y_i \; - \; \Sigma X_i \Sigma Y_i}{\sqrt{(n\,\Sigma X_i^2 - (\Sigma X_i\,)^2)\;(n\Sigma {Y_i}^2 - (\Sigma Y_i)^2)}} \qquad (3.6)$$

Sedangkan estimasi reliabilitas dengan teknik tes-retes menggunakan program R dapat dilakukan melalui sintaks berikut.

```
data<-read.csv("datareliabilitas-eks.csv",header=T,sep=";")
X <- data[-c(2)] #X terletak pada kolom 1, sehingga kolom 2
dikeluarkan (exclude)
Y <- data[-c(1)] #Y terletak pada kolom 2, sehingga kolom 1
dikeluarkan (exclude)
Reliabilitas <- c(cor(X,Y))
Reliabilitas
[1] 0.9400379
```

```
#Menghitung Standard Error of Measurement (SEM)
SEM <-
  function(X){
    X <- data.matrix(X)
    nilaiSEM <- sd(rowSums(X)) * sqrt(1 - Reliabilitas)
    return(list("Nilai SEM" = nilaiSEM))}
SEM(data)
[1] 2.018574
```

Dari *output* tersebut diperoleh koefisien reliabilitas sebesar 0,9400379, dan nilai tersebut dalam kategori tinggi (terkait kriteria reliabilitas, batasnya bisa saja berbeda antar pendapat ahli, tergantung referensi yang kita gunakan). Sedangkan nilai *standard error of measurement* atau kesalahan pengukurannya sebesar 2,018574. Semakin kecil kesalahan pengukurannya akan semakin bagus. Untuk memprediksikan rentang skor sebenarnya (*true score*, τ) yang diperoleh respoden digunakan hasil interpretasi dari SEM, di mana *true score* dari hasil penggukuran dituliskan dengan:

$$X - SEM < \tau < X + SEM$$

Skor tes yang baik adalah yang memiliki true score tinggi dan kesalahan pengukuran yang kecil. SEM sangat penting digunakan untuk kalibrasi item, penurunan SEM pasca kalibrasi item, menunjukan adanya peningkatan kualitas item.

b. Metode Equivalen

Untuk mengestimasi koefisien reliabilitas pada metode ini, diperlukan dua instrumen yang paralel. Dua buah instrumen dapat dikatakan paralel atau *equivalent* apabila kedua instrumen tersebut mempunyai kesamaan tingkat kesukaran, tujuan, dan susunan, walaupun butir-butir soalnya berbeda.

Adapun langkah-langkah dalam mengestimasi reliabilitas dengan metode ini yaitu: (a) menyiapkan dua buah instrumen yang paralel; (b) menentukan subjek untuk ujicoba instrumen; (c) melaksanakan pengukuran dengan mengadministrasikan instrumen; (d) memberikan skor pada setiap jawaban responden terhadap kedua instrumen tersebut; (e) menghitung koefisien korelasi dari kedua instrumen tersebut.

Estimasi reliabilitas dengan teknik ini sama dengan teknik tes-retes dengan cara mengkorelasikan hasil pengukuran instrumen paket pertama (X) dan yang kedua (Y), sehingga persamaan dan sintaks menggunakan program R nya pun sama seperti pada metode tes ulang, yang membedakan adalah teknik pengambilan datanya.

c. Metode Belah Dua (*Split Half Method*)

Metode ini hanya membutuhkan satu kali pengumpulan data dimana dalam satu instrumen dikerjakan satu kali oleh sejumlah subjek, dan reliabilitas skor perangkat pengukuran dapat diestimasi dengan cara membagi dua butir-butir pada instrumen. Proses pembagian dapat berdasar pada nomor ganjil-genap dari instrumen, atau separuh pertama dan kedua, dengan menggunakan nomor acak, atau tanpa menggunakan pola tertentu. Skor dari setiap teknik pembagian tersebut kemudian dikorelasikan. Adapun asumsi yang harus diperhatikan yaitu belahan pertama dan belahan kedua harus mengukur konstruk yang sama, banyaknya butir dari setiap belahan harus dapat dibandingkan, atau minimal jumlahnya hampir sama.

Pada pembuktian instrumen dengan cara ini salah satu caranya dengan rumus Spearman-Brown:

$$r_i = \frac{2r_b}{1 + r_b} \tag{3.7}$$

dengan $r_b = \frac{n\sum XY - \sum X \sum Y}{\sqrt{(n\sum X^2 - (\sum X)^2)\cdot(n\sum Y^2 - (\sum Y)^2)}}$

di mana r_i: koefisien reliabilitas skor instrumen; r_b: koefisien korelasi antara dua belahan instrumen; N: banyaknya responden; X: hasil pengukuran belahan pertama; dan Y: hasil pengukuran belahan kedua. Untuk mengestimasi reliabilitas dengan metode ini menggunakan program R, pertama-tama kita membagi dua data yang telah didapat, kemudian kita simpan datanya seperti berikut.

Kemudian kita tuliskan sintaksnya pada program R seperti berikut.

```
data<-read.csv("datareliabilitas-inc.csv",header=T,sep=";")
X <- data[-c(2)] #X terletak pada kolom 1, sehingga kolom 2
dikeluarkan (exclude)
Y <- data[-c(1)] #Y terletak pada kolom 2, sehingga kolom 1
dikeluarkan (exclude)
rb <- c(cor(X,Y))
Reliabilitas <- (2*rb)/(1+rb)
Reliabilitas
[1]0.9690923
```

```
#Menghitung Standard Error of Measurement (SEM)
SEM <-  function(X){X <- data.matrix(X)
nilaiSEM <- sd(rowSums(X)) * sqrt(1 - Reliabilitas)
            return(list("Nilai SEM" = nilaiSEM)) }
SEM(data)
[1] 1.449237
```

Dari output tersebut diperoleh koefisien reliabilitas sebesar 0,9690923, dan nilai tersebut dalam kategori tinggi (terkait kriteria reliabilitas, batasnya bisa saja berbeda antar pendapat ahli, tergantung referensi yang kita gunakan). Sedangkan nilai *standard error of measurement* atau kesalahan pengukurannya sebesar 1,449237.

d. Alpha dari Cronbach

Untuk mengestimasi reliabilitas instrumen yang mempunyai skala politomi, atau soal bentuk uraian dapat menggunakan Aplha Cronbach. Adapun rumus Alpha sebagai berikut.

$$\alpha = \left(\frac{k}{k-1}\right)\left(1 - \frac{\sum \sigma_i^2}{\sigma_t^2}\right) \quad (3.8)$$

Dengan α: koefisien reliabilitas instrumen; k: banyaknya butir pertanyaan dalam instrumen; $\sum \sigma_i^2$: jumlah varians butir instrumen; σ_t^2: varians skor total. Sedangkan estimasi dengan program R menggunakan sintaks sebagai berikut.

```
data <- read.csv("datareliabilitas-kom.csv",
                 header=T,sep=";")
#fungsi
cronbachs.alpha <-
  function(X)
    {
    X <- data.matrix(data)
    n <- ncol(X) #Jumlah item
    k <- nrow(X) #Jumlah responden
#Cronbachs alpha
alpha <- (n/(n - 1))*(1 - sum(apply(X, 2,
         var))/var(rowSums(X)))
return(list("Reliabilitas (Crombach's alpha)" = alpha,
            "Jumlah Item" = n,
            "Jumlah Responden" = k))}
cronbachs.alpha(data)
[1] 0.8486906
```

```
#Menghitung Standard Error of Measurement (SEM)
SEM <- function(X){X <- data.matrix(X)
        nilaiSEM <- sd(rowSums(X)) * sqrt(1 –
        cronbachs.alpha(X) [[1]]) return(list
        ("Nilai SEM" = nilaiSEM))
  }
SEM(data)
[1] 3.425343
```

Bedasarkan output yang diperoleh koefisien reliabilitas sebesar 0,8486906, dan nilai tersebut dalam kategori tinggi (terkait kriteria reliabilitas, batasnya bisa saja berbeda antar pendapat ahli, tergantung referensi yang kita gunakan). Sedangkan nilai *standard error of measurement* atau kesalahan pengukurannya sebesar 3,425343.

d. Kuder-Richardson

Metode lain untuk mengestimasi reliabilitas dengan reliabilitas komposit yaitu dengan mengaplikasikan formula Kuder dan Richardson (KR). Terdapat dua jenis formula KR, yaitu Kuder Richardson formula 20 (KR-20) dan Kuder Richardson formula 21 (KR-21). Untuk instrumen dengan skornya tiap butirnya 1 dan 0, dan

juga skala politomi, misalnya angket (skala Likert 1-2-3-4-5) atau soal bentuk uraian dapat menggunakan Formula KR-21. Untuk menganalisis butir dikotomi dapat menggunakan Formula KR-20. Instrumen dengan penskoran dikotomi, misal ya-tidak, benar-salah, 1-0, dan lain-lain. Adapun rumus KR-20 sebagai berikut.

$$r_{ii} = \frac{k}{(k-1)}\left(\frac{s_t^2 - \sum p_i q_i}{s_t^2}\right) \qquad (3.9)$$

Dengan r_{ii}: reliabilitas skor instrumen; k: banyaknya butir pertanyaan atau banyaknya soal; s_t^2: varians skor total; p_i: proporsi subjek yang menjawab dengan benar pada suatu butir (proporsi subjek yang mendapat skor 1) yang dihitung dengan rumus $p_i = \frac{\text{banyaknya subjek yang skornya 1}}{N}$; dan q_i: proporsi subjek yang menjawab salah ($q_i = 1 - p_i$).

Estimasi reliabilitas dengan rumus KR-20 pada program R menggunakan sintaks berikut.

```
data <- read.csv("datareliabilitas-kr20.csv",
                 header=T,sep=";")
#formula 20
KR20 <- function(X)
    {
    X <- data.matrix(X)
    k <- ncol(X) #Jumlah item
    #Person total score variances
    SX <- var(rowSums(X))
    #item means
    IM <- colMeans(X)
    return(((k/(k - 1))*((SX - sum(IM*(1 - IM)))/SX)))}
KR20(data)
[1] 0.8611331
```

```
#Menghitung Standard Error of Measurement (SEM)
SEM <- function(X){
    X <- data.matrix(X)
    nilaiSEM <- sd(rowSums(X)) * sqrt(1 - KR20(X)[[1]])
    return(list("Nilai SEM" = nilaiSEM))}
SEM(data)
[1] 1.183904
```

Berdasarkan output tersebut diperoleh koefisien reliabilitas sebesar 0,8611331, dan nilai tersebut dalam kategori tinggi (terkait kriteria reliabilitas, batasnya bisa saja berbeda antar pendapat ahli, tergantung referensi yang kita gunakan). Sedangkan nilai *standard error of measurement* atau kesalahan pengukurannya sebesar 1,183904.

3.3.2.2 Reliabilitas Kasus Khusus (Relibalitas komposit)

Jika instrumen yang digunakan membangun suatu konstruk yang sama namun terdiri dari butir yang berbeda-beda, maka untuk mengestimasi reliabilitasnya dapat menggunakan rumus reliabilitas komposit. Reliabilitas komposit seringkali disebut juga dengan reliabilitas konstrak. Jöreskog (1971) mengusulkan komposit reliabilitas (CR) yang kemudian dikembangkan oleh McDonald (2013), koefisien reliabilitas komposit untuk konstrak dapat didefinisikan melalui formula berikut ini.

$$\rho c = \frac{\left(\sum_{k=1}^{K_j} \lambda_{jk}\right)^2}{\left(\sum_{k=1}^{K_j} \lambda_{jk}\right)^2 + \sum_{k=1}^{K_j} \delta_{jk}} \tag{3.10}$$

dimana:

ρc koefisien reliabilitas komposit

K_j adalah jumlah dari indicator pada konstrak ξ_j

λ_{jk} adalah faktor *loading*

δ_{jk} adalah varians eror indikator ke K, $k = 1,2,\dots,K)$ dari konstrak ξ_j dengan; $\delta_{\text{jk}} = \sum_{k=1}^{K_j} 1 - \lambda_{jk}^2$

Reliabilitas komposit dapat di estimasi melalui analisis Confirmatory Factor Analysis. Pada program R dapat menggunakan package lavaan, dan beberapa sytax tambahan sebagai berikut.

```
library(lavaan)
library(Hmisc)
library(readxl)
#Set direktor dan memanggil data
setwd("C:/Users/amien/OneDrive/Desktop")
data <- read_xlsx("DATA_CFA.xlsx",1)
#Membuat model
model <- "
      Literasi_digital=~ b1+b2+b3+b4+b5
      "
#uji model dengan cfa
uji <- cfa(model, data = data)
summary(uji, fit.measures=TRUE, standardized = TRUE,
        rsquare = TRUE)

#Hitung AVE, CR equivalent (Estmasi Reliabiltias dan Buktikan
Validitas )
reliability(uji)
```

```
########Catatan: Omega = Convergent validity (cut off>0,7), AVE =
discrimninant validity (cut off>0,5), Alpha=konst. Internal (cut
of>0,7)
#cek cross loading
inspect(uji, what = "std")
dev.off()
```

#Ouput Reliabiltias

```
> reliability
       Literasi_digital
alpha         0.946
omega         0.947
omega2        0.947
omega3        0.947
avevar        0.781
```

Pada output reliabiltas menggunakan paket *Hmisc* terdapat 3 buah omega, yaitu; Omega, Omega 2 dan Omega 3, semuanya menunjukkan reliabilitas komposit, dan avevar menunjukkan rata-rata varians extraxted (AVE). Kita fokus pada satu ukuran saja, yaitu; reliabilitas komposit McDonald yang ditunjukkan dengan koefisien Omega 3. Ambang batas (threshold) sebuah konstrak memiliki reliabilitas komposit yang baik apabila koefisien omega >0,7.

3.3 Analisis Item Berdasarkan CTT Menggunakan R

Analisis item dalam pendekatan klasik sering bergantung pada dua statistik untuk mengevaluasi item tunggal: *nilai-P* dan *koefisien korelasi poin biserial*. Nilai-P mewakili proporsi peserta ujian yang merespons sesuai dengan kunci jawaban, dan biasanya disebut sebagai kesulitan item. Koefisien korelasi poin biserial adalah korelasi item dengan semua item lainnya. Nilai tersebut memberikan indeks kekuatan pembeda item (indeks daya beda), yang biasanya disebut sebagai diskriminasi item. Untuk menerapkan analisis item berbasis tes klasik (CTT) menggunakan R, kita siapkan file hasil rekap nilai (respons) siswa, dan kunci jawabannya. Pada kasus ini ada 5 opsi jawaban (A, B, D, E) yang kami tempatkan pada Sheet 1 lembar kerja excel, dan kunci jawaban di sheet 2 yang diletakan secara horizontal pada baris pertama. Selanjutnya, kita lakukan analisis item program R melalui syntax sebagai berikut:

```
library(psych)
library(CTT)
library(tidyverse)
library(readxl)
library(psychometric)
#Mengatur Direktori dan memanggil file kerja
setwd ("D:/BUKU R")
jawaban <- read_excel("DATA CTT.xlsx",1) #jawaban siswa berada di
sheet 1 lembar kerja excel
#Importing test key
kunci <- read_excel("DATA CTT.xlsx",2) #kunci jawaban berada di sheet
2 lembar kerja excel
kunci <- as.matrix(kunci)
```

Total skor per subjek dapat kita hitung menggunakan syntax berikut ini.

```
#Total skor persubjek
skor <- as.data.frame(scoring$score)
```

Output total skor tes dalam contoh kasus ini dari hasil perhitungan R menggunakan syntax di atas adalah sebagai berikut:

```
> skor
    scoring$score
P1              9
P2              7
P3              8
P4             10
P5              8
P6             10
P7             10
P8             10
P9              4
P10             5
P11             8
P12             8
P13            10
P14             6
P15             9
P16            10
P17             2
P18             9
P19             1
P20             0
P21             8
P22             9
P23             6
P24            10
P25             7
P26             9
P27             0
P28            10
P29             5
P30             2
```

```
P31            10
P32            10
P33             4
P34            10
P35             9
```

Menggunakan syntax CTT berikut ini kita dapat menghasilkan daya beda butir (pbis).

```
#Item analysis
Item_Analysis <- itemAnalysis(skor, hardFlag=.25, pBisFlag=.15, )
Item_Analysis$itemReport
```

Ringkasan output analisis item berbasis CTT menggunakan program R adalah sebagai berikut:`

```
> Item_Analysis$itemReport
itemName itemMean pBis   bis     alphaIfDeleted
V1       0.771   0.750   1.041   0.891
V2       0.829   0.796   1.179   0.890
V3       0.857   0.470   0.729   0.907
V4       0.714   0.660   0.878   0.897
V5       0.829   0.649   0.961   0.898
V6       0.714   0.814   1.082   0.887
V7       0.743   0.714   0.967   0.893
V8       0.800   0.740   1.057   0.892
V9       0.657   0.599   0.773   0.902
V10      0.314   0.495   0.648   0.908
```

3.3.1 Daya pembeda Butir

Indeks diskriminasi item adalah ukuran seberapa baik suatu barang mampu membedakan antara peserta ujian yang berpengetahuan luas dan mereka yang tidak, atau antara master dan non-master. Kisaran yang mungkin dari indeks diskriminasi adalah -1.0 hingga 1.0; namun, jika suatu item memiliki diskriminasi di bawah 0,0, itu menunjukkan masalah Kriteria daya beda ditetapkan berdasarkan koefisien korelasi poin biserial (**pbis**), (dapat diterima/baik jika pbis 0,4-1,0, diterima dan perlu diperbaiki jika Rpbis 0,30-0,39, diperbaiki jika Rpbis 0,2-0,29, soal dibuang jika pbis 0,00-0,19). Ketika suatu item mendiskriminasi secara negatif, secara keseluruhan peserta ujian yang paling berpengetahuan mendapatkan item yang salah dan peserta ujian yang paling tidak berpengetahuan mendapatkan item tersebut dengan benar. Suatu item yang mengukur sesuatu selain dari apa yang diukur oleh tes ditandai dengan indeks diskriminasi yang bernilai negatif. Atau

dapat diartikan bahwa item tersebut memiliki kunci jawaban yang salah.

3.3.2 Analisis Distraktor (efektivitas pengecoh)

Pada analisis distraktor peserta ujian dibagi menjadi tiga tingkat kemampuan bawah, menengah dan atas berdasarkan total nilai tes mereka. Proporsi peserta ujian yang menandai setiap opsi di masing-masing dari tiga tingkat kemampuan dibandingkan. Dalam tingkat kemampuan yang lebih rendah, kita berharap untuk melihat proporsi penguji yang lebih kecil memilih opsi yang benar dan sebagian besar dari mereka menandai opsi atau pengalih perhatian yang salah.

Idealnya, pengalih perhatian yang baik akan menarik proporsi peserta ujian yang hampir sama. Pengalih perhatian yang tidak menarik atau menarik sebagian kecil peserta ujian relatif terhadap distraktor lain harus dipertimbangkan untuk direvisi. Pada tingkat kemampuan yang lebih tinggi, kita akan berharap untuk melihat bahwa mayoritas peserta ujian memilih opsi yang benar. Jika distraktor lebih menarik daripada opsi yang benar untuk peserta ujian tingkat kemampuan yang lebih tinggi, maka itu harus dihilangkan atau direvisi. Kriteria pengecoh yang baik adalah jika opsi jawaban selain kunci (pengecoh) dipilih oleh testee dengan probabilitas P>0,05, artinya pengecoh telah berfungsi dengan baik, atau dipilih oleh minimal 1 testee. Kita dapat menggunakan syntax berikut ini untuk menghasilkan efektivitas pengecoh.

```
#Efektivitas Pengecoh
distractor.analysis(jawaban,kunci,p.table=TRUE,write.csv="Peng
ecoh.csv")
```

Hasil analisis efektivitas pengecoh dapat dilihat pada output program R sebagai berikut:

```
$butir01
        score.level
response lower middle upper
      *A 0.385  1.000 1.000
       B 0.154  0.000 0.000
       C 0.154  0.000 0.000
       D 0.154  0.000 0.000
       E 0.154  0.000 0.000
$butir02
        score.level
response lower middle upper
```

```
       A 0.077  0.000 0.000
       B 0.154  0.000 0.000
      *C 0.538  1.000 1.000
       D 0.154  0.000 0.000
       E 0.077  0.000 0.000
$butir03
        score.level
response lower middle upper
       A 0.000  0.091 0.000
      *B 0.692  0.909 1.000
       C 0.154  0.000 0.000
       D 0.077  0.000 0.000
       E 0.077  0.000 0.000
$butir04
        score.level
response lower middle upper
       A 0.231  0.000 0.000
      *B 0.308  0.909 1.000
       C 0.077  0.091 0.000
       D 0.231  0.000 0.000
       E 0.154  0.000 0.000
$butir05
        score.level
response lower middle upper
       A 0.077  0.000 0.000
       B 0.154  0.000 0.000
       C 0.154  0.000 0.000
       D 0.077  0.000 0.000
      *E 0.538  1.000 1.000
$butir06
        score.level
response lower middle upper
       A 0.154  0.000 0.000
       B 0.231  0.000 0.000
       C 0.154  0.000 0.000
      *D 0.231  1.000 1.000
       E 0.231  0.000 0.000
$butir07
        score.level
response lower middle upper
       A 0.077  0.000 0.000
       B 0.231  0.000 0.000
      *C 0.308  1.000 1.000
       D 0.231  0.000 0.000
       E 0.154  0.000 0.000
$butir08
        score.level
response lower middle upper
       A 0.154  0.000 0.000
       B 0.154  0.000 0.000
       C 0.154  0.000 0.000
      *D 0.462  1.000 1.000
       E 0.077  0.000 0.000
 $butir09
        score.level
response lower middle upper
      *A 0.308  0.727 1.000
       B 0.231  0.091 0.000
       C 0.154  0.000 0.000
```

```
       D 0.077  0.182 0.000
       E 0.231  0.000 0.000
$butir10
        score.level
response lower middle upper
       A 0.385  0.182 0.000
       B 0.154  0.364 0.000
       C 0.077  0.455 0.000
       D 0.385  0.000 0.000
      *E 0.000  0.000 1.000
```

3.3.3 Tingkat Kesulitan Butir

Kesulitan item dari suatu item adalah proporsi siswa yang menjawab item tes tertentu dengan benar. Indeks ini berguna dalam menilai apakah sesuai dengan tingkat siswa yang mengikuti tes. Kisaran indeks kesulitan item yang diinginkan adalah antara 0,3 hingga 0,7, sedangkan angka mendekati 0 atau 1 menawarkan sedikit informasi tentang pengukuran tingkat konstruk siswa. Namun, cut-off ekstrem untuk kesulitan item dapat berlaku untuk pengukuran yang dirancang untuk kelompok ekstrem. Sebuah item tergolong sulit jika probabilitas (P) menjawab benar pada item tertentu berada pada rentang 0,00-0,3, sedang 0,31-0,70, mudah 0,71-1,00). Tingkat kesukaran butir dihasilkan dari syntax efektivitas pengecoh (distractor) pada bagian 3.3.2.

```
Items Kunci*      lower    middle  upper
1       A       0.385   1.000   1.000
2       C       0.538   1.000   1.000
3       B       0.692   0.909   1.000
4       B       0.308   0.909   1.000
5       E       0.538   1.000   1.000
6       D       0.231   1.000   1.000
7       C       0.308   1.000   1.000
8       D       0.462   1.000   1.000
9       A       0.308   0.727   1.000
10      E       0.000   0.000   1.000
```

**jawaban*
sesuai kunci, diringkas dari hasil analisis per butir

Berdasarkan output tingkat kesukaran butir, dapat diketahui bahwa sebagian besar item memiliki tingkat kesukaran sedang (item 1 sampai dengan item 8), item 9 tergolong mudah dan item 10 tergolong sulit.

3.3.4 Kesalahan Pengukuran

Kesalahan standar pengukuran (SEm) adalah ukuran seberapa banyak nilai tes terukur yang tersebar di sekitar true score. Kesalahan pengukuran standar didasarkan pada nilai alfa, hasil perhitungan SEm menggunakan R adalah sebagai berikut:

```
#Standar Error Measurement (SEm)
coef.alpha <- alpha(skor)
s <- SD(skor)
SEm <- SE.Meas(s,coef.alpha)
data.frame(SEm)
```

Output Standar Error Pengukuran (SEm) yang dihasilkan dari syntax di atas adalah sebagai berikut:

```
Items    SEM
V1       0.131
V2       0.117
V3       0.109
V4       0.140
V5       0.117
V6       0.140
V7       0.136
V8       0.124
V9       0.148
V10      0.144
```

Skor tes yang baik adalah yang memiliki true score tinggi dan kesalahan pengukuran yang kecil. SEm sangat penting digunakan untuk kalibrasi item, penurunan SEm pasca kalibrasi item, menunjukan adanya peningkatan kualitas item.

Referensi

Aiken, L. R. (1985). Three coefficients for analyzing the reliability and validity of ratings. *Educational and Psychological Measurement*, *45*(1), 131–142. https://doi.org/10.1177/0013164485451012

Allen, M. J. & Yen, W. M. (1979). *Introduction to measurement theory*. Monterey, CA: Brooks/Cole Publishing Company.

Christopher D. Desjardins & Okan Bulut. (2018). Handbook of Educational Measurement and Psychometrics Using R. New York : CRC Press

Faraway, J. J. (2014). Linear models with R. Boca Raton, FL: CRC Press.

Lawshe, C. H. (1975). A quantitative approach to content validity. Personnel Psychology, 28 (4), 563–575.

R Core Team. (2022a). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*.

R Core Team. (2022b). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/

Revelle, W. (2022). *psych: Procedures for Psychological, Psychometric, and Personality Research*. https://personality-project.org/r/psych/

Wickham, H., & Bryan, J. (2022). *readxl: Read Excel Files*. https://cran.r-project.org/package=readxl

Willse, J. T. (2018). *CTT: Classical Test Theory Functions*. https://cran.r-project.org/package=CTT

Wongvorachan, T. (2020). Classical Test Theory practice.

Chapter 4
Analisis Faktor Eksploratori

Oleh: Eko Wahyunanto Prihono, & Heri Retnawati

4.1 Sejarah Analisis Faktor

Spearman adalah orang yang pertama kali mengenalkan analisis faktor pada tahun 1904, kemudian dikembangkan oleh Thurstone tahun 1974, Thomson tahun 1951, Lawley tahun 1940 dan 1941 dan lainnya (Cudeck & MacCallum, 2007). Meskipun demikian, terdapat pendapat bahwa ide terkait dengan analisis faktor ditemukan pada karya sebelumnya oleh Karl Pearson (Person, 1901). Awalnya Spearman dan sebagian besar pengikutnya tertarik untuk mengukur kemampuan manusia, khususnya kemampuan berupa kecerdasan umum. Tidak ada tujuan lain untuk mengembangkan metode umum analisis multivariat yang kemudian menjadi analisis faktor. Analisis faktor tidak dapat dikembangkan di antara teknik statistik multivariat, karena hampir seluruhnya dikembangkan dalam disiplin ilmu psikologi. Oleh karena itu, pengembangannya disesuaikan pada kebutuhan pengukuran kemampuan psikologis pada khususnya.

Pada awalnya analisis ini menemui banyak kontroversi, namun seiring dengan berkembangnya teknologi berupa komputer dan berkembangnya analisis data menggunakan program statistik melalui aplikasi praktis menyebabkan analisis faktor ini menjadi sangat bermanfaat. Pencapaian Spearman (1904) menjelaskan bahwa korelasi dalam gabungan variabel yang dapat diamati dengan hipotesis membuktikan faktor tersebut memiliki ketergantungan pada faktor umum. Saat ini analisis faktor digunakan dari berbagai penelitian yang bertujuan untuk menentukan faktor-faktor bersama pada setiap disiplin ilmu baik ekonomi, sosial, hukum, psikologi dan lain-lain.

Pada tahun 1904, belum banyak teori statistik untuk membantu Spearman. Padahal di tahun tersebut korelasi dirasa telah menjadi bidang penelitian yang penting. Penemuan korelasi *product-moment* diikuti oleh persamaan korelasi parsial. Korelasi parsial mengasumsikan bahwa hubungan antar variabel adalah linier. Hal ini

karena korelasi *product-moment* merupakan ukuran korelasi linier. Melihat edisi sebelumnya dari Pengantar Teori Statistik (Yule, 1922), kita dapat melihat seberapa jelas korelasi parsial pada persamaan awal. Setelah itu, fokus bergeser ke regresi berganda (lihat regresi berganda). Ini memberikan cara alternatif untuk mempelajari fenomena yang sama.

Analisis faktor merupakan suatu *interdependence technique*, dengan tujuan utama yaitu mendefinisikan struktur yang terletak di antara variabel-variabel. Analisis faktor digunakan untuk menganalisis struktur dari hubungan intern atau korelasi di antara sejumlah besar variabel dengan menerangkan korelasi yang baik antara variabel, yang diasumsikan untuk merepresentasikan dimensi-dimensi dalam data. Analisis faktor digunakan untuk mengelompokkan beberapa variabel yang memiliki karakteristik yang sama untuk dikumpulkan dalam satu faktor, sehingga beberapa atribut yang mempengaruhi satu komponen variabel dapat diringkas menjadi beberapa faktor utama yang mempunyai jumlah yang lebih sedikit.

Secara khusus, metode analisis faktor sering digunakan dalam berbagai kegiatan pengukuran. Mengingat banyaknya analisis data yang menggunakan analisis faktor, maka dirasa penting penggunaan analisis faktor yang benar dan tepat. Sehingga dapat menghasilkan laporan suatu penelitian dengan benar (Goretzko et al., 2021; Watkins, 2018).

Terdapat dua cara dalam melakukan analisis faktor, yakni analisis faktor eksploratori dan analisis faktor konfirmatori (Price, 2016). Analisis faktor eksploratori digunakan untuk membangun teori sedangkan analisis faktor konfirmatori digunakan untuk menguji teori. Pada bab ini penulis akan fokus mengulas pada analisis faktor eksploratori.

4.2 Analisis Faktor Eksploratori (*Eksploratory Factor Analysis*)

Eksploratory Factor Analysis (EFA) merupakan salah satu teknik analisis faktor, di mana beberapa faktor yang akan terbentuk berupa variabel laten yang belum dapat ditentukan sebelum analisis dilakukan. Pada EFA terbentuknya suatu faktor baru bersifat acak, yang kemudian

dapat diinterpretasikan sesuai dengan konstruk yang terbentuk. Kovarians antar variabel diuji dalam analisis faktor dan bertujuan untuk mendapatkan variabel laten yang lebih sedikit daripada jumlah indikator (Watkins, 2018).

Dalam EFA sebelumnya seorang peneliti belum memiliki teori atau hipotesis yang menyusun struktur faktor-faktornya yang akan terbentuk, sehingga dengan kata lain pada analisis faktor eksploratori merupakan salah satu cara membangun atau mengembangkan teori baru. Teknik ini digunakan untuk mereduksi data dari variabel asal menjadi variabel baru. Proses analisis ini akan menghasilkan korelasi antar variabel baru, sehingga pada akhirnya dapat dibuat menjadi satu atau lebih kumpulan variabel laten yang jumlahnya kurang dari jumlah variabel awal yang tidak saling berhubungan.

EFA juga digunakan untuk mendeteksi dan mengakses sumber laten dari variasi atau kovariansi dalam suatu pengukuran (Karl & Dag, 1994). EFA bersifat mengeksplor data empiris yang bertujuan untuk mengidentifikasi karakteristik dan korelasi antar variabel tanpa menentukan model pada data. Digunakan pendekatan eksploratif dengan cara mengamati besarnya muatan faktor guna melihat berapa banyak faktor yang diperlukan untuk mendeskripsikan korelasi di antara seperangkat indikator. EFA juga bertujuan untuk menemukan struktur laten dari variabel laten dengan mengidentifikasi faktor-faktor yang mempunyai karakteristik yang sama dan dimensi-dimensi tersembunyi yang dapat mempengaruhi variabel observasi (Park et al., 2002).

Hasil kajian tersebut perlu dibuktikan melalui pengujian secara empiris untuk membuktikan apakah faktor-faktor tersebut merupakan sebuah konstruk yang belum terakomodir dalam konsep-konsep yang ada sehingga faktor tersebut dapat melengkapi pemahaman konsep terhadap variabel baru. Berdasarkan hal-hal tersebut, maka suatu penelitian dilakukan untuk menguji hipotesis-hipotesis mengenai eksistensi konstruk dalam variabel-variabel baru sebagai konstruk yang independen. Studi yang dilakukan diperlukan untuk menegaskan apakah konstruk dari variabel baru tersebut merupakan konstruk yang independen ataukah sebaliknya

4.2.1 Asumsi EFA

Salah satu tahap awal dalam melakukan pengujian EFA adalah terpenuhinya asumsi analisis. Asumsi yang harus dipenuhi meliputi: hubungan antar variabel harus linier, variabel diasumsikan memiliki distribusi normal multivariat, dan kumpulan data tidak terdiri dari outlier multivariat (Tabachnick et al., 2007).

4.2.2 Ukuran Sampel

Terdapat berbagai pendapat untuk menentukan ukuran sampel minimum dalam pengujian EFA. Setidaknya harus ada empat atau lima individu per indikator (Floyd & Widaman, 1995), sedangkan (Gorsuch, 1988) mengatakan bahwa harus ada lima individu. Namun, dia menekankan bahwa ukuran sampel tidak boleh kurang dari 200. Selanjutnya (Streiner, 1994) berpendapat setidaknya ada lima individu per indikator seperti (Gorsuch, 1988) tetapi dia menunjukkan bahwa ukuran sampel tidak boleh kurang dari 100. Jika ukuran sampel kurang dari 100, harus ada 10 individu per indikator. Selain itu, (Comrey, 1988) menyatakan bahwa ukuran sampel untuk 200 sudah cukup dalam banyak kasus tetapi menekankan bahwa ini terjadi jika item dalam skala tidak melebihi 40.

(Guadagnoli & Velicer, 1988) melaporkan bahwa pemuatan faktor item lebih besar dari 0,80 stabil, bahkan jika ukuran sampel kurang dari 50, terlepas dari jumlah variabel. Terdapat pendapat bahwa ketika data biner berdistribusi normal, ukuran sampel 50 sudah cukup bila ada 12 variabel per faktor (Pearson, Robert Henry & Mundform, 2010), sedangkan (de Winter et al., 2009) menyatakan bahwa ukuran sampel di bawah 50 sudah cukup. Mereka menekankan bahwa untuk konstruksi unidimensional, jika beban faktor adalah 0,8 dan ada 24 indikator, ukuran sampel enam sudah cukup.

4.2.3 Pemilihan Variabel

Pemilihan variabel dilakukan dengan melihat korelasi yang kuat antara item dan akan dimasukan dalam analisis faktor. Sebaliknya item dengan korelasi yang lemah akan dikeluarkan dari analisis. Hal penting dalam analisis ini adalah ukuran hubungan atau korelasi antar item yang disusun berdasarkan suatu indikator awal. Hal ini dilakukan untuk mengidentifikasi hubungan dalam sekumpulan item tersebut. Salah satu

cara untuk mengetahui korelasi antar item dapat menggunakan *Measure of Sampling Adequacy* (MSA) dan *Kaiser-Meyer-Olkin* (KMO) *measure of sampling adequacy and Bartlett test of sphericity*. MSA digunakan untuk mengetahui apakah item sudah mencukupi untuk dianalisis lebih lanjut atau tidak. Lebih lanjut, nilai MSA digunakan untuk melihat korelasi antar item. Nilai MSA dapat dilihat dari perolehan anti *image correlation*. Apabila terdapat item awal yang mempunyai nilai MSA < 0.5, maka item tersebut dikeluarkan dari proses analisis. Setelah diperoleh item-item yang sesuai dengan indikator (MSA > 0,5) maka tahap selanjutnya yaitu melakukan uji kecukupan sampel.

Uji kecukupan sampel dapat dilihat dari besarnya indeks *Kaiser-Meyer-Olkin* (KMO) *Measure of Sampling Adequacy*. Nilai KMO yang diterima antara 0,5 sampai 1 menunjukkan bahwa analisis faktor tepat digunakan. Lebih lanjut, untuk mengetahui apakah matriks korelasi yang terbentuk itu berbentuk matriks identitas atau bukan peneliti menggunakan *Uji Bartlett*. Hal ini karena keterkaitan antar variabel sangat diperlukan dalam analisis faktor untuk melihat hubungan beberapa variabel untuk dikumpulkan dalam satu faktor. Jika matriks korelasi yang dihasilkan merupakan matriks identitas, maka tidak terdapat korelasi atau hubungan antar variabel, hal ini mengakibatkan analisis faktor tidak dapat dilakukan.

4.2.4 Pembentukan Faktor

Pemilihan Metode Ekstraksi Faktor

Terdapat beberapa metode ekstraksi faktor dalam EFA, diantaranya: principal axis factoring (PAF), principal components analysis (PCA), minimum residual (minRes), maximum likelihood (ML), alfa factoring, weighted least squares (WLS), dan unweighted least squares (ULS). Tentunya setiap metode memiliki kelebihan dan kelemahan jika dibandingkan satu sama lain, tetapi untuk data yang tidak berdistribusi normal, metode PAF secara umum menawarkan hasil yang lebih baik (Fabrigar & Wegener, 2011), sedangkan (Costello & Osborne, 2005)menyatakan bahwa jika data berdistribusi normal, maka maximum likelihood direkomendasikan. Tetapi jika asumsi

normalitas tidak berlaku, metode PAF direkomendasikan untuk sebagian besar kasus.

Salah satu pendekatan analisis faktor dapat dilakukan melalui *exploratory factor analysis* dengan metode *principal component analysis* (Elliot et al., 1999). Metode analisis faktor *principal component* bertujuan untuk mengetahui struktur yang mendasari item-item awal dalam analisis dan melakukan penyederhanaan struktur sekumpulan item awal tersebut melalui reduksi data. Prosedur matematis untuk mencari struktur kovariansi matriks $\sum$ dapat dilakukan dengan menggunakan matriks dekomposisi spektral. Koefisien faktor pada metode principal component dapat dilihat dari besarnya faktor loading yang terbentuk. Pendekatan yang dapat digunakan, yaitu dengan melihat besarnya nilai eigen.

4.2.5 Kriteria Penentuan Faktor

Untuk menentukan banyaknya faktor dapat dilakukan dengan melihat nilai eigen, dan *scree plot*. Jumlah variasi yang berhubungan pada suatu faktor ditunjukkan oleh nilai eigen. Faktor yang mempunyai nilai eigen ≥ 1 akan dipertahankan dan faktor yang mempunyai nilai eigen < 1 tidak akan diikutsertakan dalam (Gorsuch, 1983; Supranto, 2004). Jumlah kumulatif variasi yang telah dicapai dapat dijadikan sebagai jumlah faktor yang diambil. Jika nilai kumulatif persentase variansinya sudah memadai, maka ekstraksi faktor dapat dihentikan. Grafik yang merepresentasikan relasi antara faktor dengan nilai eigennya disebut dengan *Scree plot*. Kriteria ini ditentukan dengan membuat plot nilai eigen terhadap banyaknya faktor yang akan diekstraksi. Banyaknya faktor (*m*) diplotkan pada arah horizontal sedangkan nilai eigen diplotkan pada arah vertikal. Penurunan (*slope*) plot nilai eigen tersebut menentukan banyaknya faktor pada kriteria. Titik penghentian ekstraksi jumlah faktor pada saat *scree plot* mulai landai dan nilai eigen berada pada rentang lebih dari satu dan kurang dari satu. Titik tersebut menunjukan jumlah faktor yang dapat diekstraksi.

4.2.6 Rotasi Faktor

Proses rotasi dilakukan untuk menyederhanakan faktor dan meningkatkan kemampuan interpretasinya. Metode rotasi orthogonal

dan oblique, dalam analisis faktor, terus dikembangkan oleh banyak peneliti. Rotasi yang dilakukan dengan mempertahankan sumbu secara tegak lurus satu dengan yang lainnya disebut dengan rotasi orthogonal. Jika rotasi ini dilakukan, maka setiap faktor akan independen terhadap faktor lain dikarenakan sumbu yang terbentuk saling tegak lurus. Rotasi ini diaplikasikan untuk mereduksi jumlah variabel tanpa mempertimbangkan seberapa berartinya faktor yang diekstraksi, dengan kata lain rotasi ortogonal mengasumsikan bahwa faktor-faktor tidak berkorelasi. Rotasi yang tidak mempertahankan sumbu tegak lurus adalah rotasi oblique. Jika rotasi ini dilakukan, maka korelasi antar faktor masih diperhitungkan karena sumbu faktor tidak saling tegak lurus satu dengan yang lainnya. Rotasi ini diaplikasikan guna memperoleh jumlah faktor yang cukup berarti secara teori, dengan mengasumsikan bahwa faktor-faktor berkorelasi dan memungkinkan korelasi antar faktor (Costello & Osborne, 2005).

Pada metode rotasi orthogonal dapat dilihat berdasarkan hasil pengukuran varimax. Analisis varimax fokus untuk menyederhanakan kolom matriks faktor. Penyederhanaan dapat dilakukan secara maksimal jika hanya ada nilai 0 dan 1 dalam satu kolom. Sehingga terjadi kecenderungan untuk menghasilkan beberapa nilai *loading factor* yang tinggi (mendekati -1 atau +1) dan beberapa nilai *loading factor* mendekati 0 pada masing-masing kolom matriks (Gorsuch, 1983).

Ketika korelasi antara faktor dan variabel bernilai +1 atau -1, maka logika interpretasi akan lebih mudah, karena hal tersebut menginformasikan adanya asosiasi yang sempurna yang sifatnya positif atau negatif. Nilai 0 mengindikasikan adanya asosiasi yang sangat kurang. Teknik ini mencoba menghasilkan nilai *loading factor* yang besar. Jika dibandingkan dengan metode quartimax, maka struktur yang dihasilkan ini jauh lebih sederhana.

4.2.7 Penamaan Faktor

Dengan memperhatikan hal-hal mendasar dan cukup mewakili sifat-sifat dari item-item awal yang terkumpul dalam satu faktor, maka penamaan faktor dapat dilakukan. Adapun langkah yang dapat ditempuh yaitu dengan menerapkan generalisasi terhadap item-item

awal. Hal tersebut dilihat dari nilai *loading factor* yang diperoleh dari setiap item dengan membandingkan nilai *loading factor* dari variabel di dalam faktor yang terkonstruk. Penentuan signifikansi faktor loading dilakukan dengan menggunakan level signifikansi (α) 0,05 untuk mengidentifikasi *loading factor* yang signifikan berdasarkan ukuran sampelnya (Gorsuch, 1983). Sebuah item memiliki *loading facto* yang baik apabila item tersebut memiliki nilai di atas 0,3 (Watson, 2017). Dengan demikian, untuk mempermudah visualisasi, nilai *loading facto* yang ditampilkan hanya yang memiliki nilai di atas 0,3.

4.3 Penerapan EFA menggunakan Program R

Eksploratory Factor Analysis (EFA) sering digunakan dalam ilmu pendidikan dan sosial seiring dengan kemajuan teknologi. Selain pentingnya hasil analisis dan pertimbangan dalam penggunaan analisis faktor, *software* analisis yang dipakai dalam proses menganalisis juga penting bagi peneliti. Analisis faktor dapat dilakukan melalui beberapa *software* yang berbeda. Namun, perangkat lunak dapat berbeda dalam hal ekstraksi faktor yang diizinkan, rotasi faktor, atau metode korelasi. Oleh karena itu, pada *chapter* ini akan dijelaskan bagaimana melakukan EFA dengan bantuan software R. *Software* R dapat digunakan karena memberikan fleksibilitas kepada peneliti dan gratis (Team, 2013).

Terdapat beberapa paket untuk melakukan pengujian EFA di R. Dalam *chapter* ini akan mengeksplorasi sejumlah fungsi yang terkait dengan analisis faktor yang tersedia dalam paket dasar R serta beberapa fungsi yang sangat berguna pada *library* "*psych*" yang dikembangkan oleh (Revelle & Revelle, 2022). Untuk mengakses fungsi -fungsi ini, dapat dilakukan install dan jalankan *library* "*psych*", "*readxl*" terlebih dahulu, kemudian atur direktori kerja dan panggil file kerja dengan beberapa syntax berikut ini.

```
library(psych)
library(readxl)
setwd("D:/Prihono/Buku R")
data <- read_xlsx('Data EFA.xlsx', 1)
KMO <- KMO(data)
```

4.4 Interpretasi Hasil Analisis EFA dengan R

Uji asumsi kecukupan sampel di dalam analisis faktor eksploratori dapat dinilai menggunakan statistik Kaiser Meyer Olkin test (KMO test), dengan ketentuan jika nilai Measure of Sampling Adequacy (MS) > 0,5, artinya; asumsi jumlah sampling minimum pada analisis EFA terpenuhi. Menggunakan syntax program R berikut ini, kita dapat menghasilkan nilai MSA untuk semua item, dan MSA tiap item.

```
KMO <- KMO(data)
MSA_all ='names<-'(KMO$MSA, "Overal All MSA")
MSA_item = 'names<-'(KMO$MSAi, "MSA Per Item")
MSA_all
data.frame(MSA_item)
```

MSA seluruh item dan MSA peritem pada kasus ini adalah sebagai berikut.

```
> MSA_all
Overal All MSA
     0.7014245
> data.frame(MSA_item)
    MSA_item
1  0.7543914
2  0.8362477
3  0.5662306
4  0.8038340
5  0.7788690
6  0.7561291
7  0.7490624
8  0.5094402
9  0.6508755
10 0.6399759
11 0.3977960
12 0.7097983
13 0.7613063
14 0.7240632
15 0.5111248
16 0.8526004
17 0.6424264
18 0.6681084
19 0.7089239
20 0.8174460
21 0.7864983
22 0.8116893
23 0.5309520
24 0.6825491
25 0.6606801
26 0.6299935
27 0.6843912
```

```
28 0.6952667
29 0.6972151
30 0.5877247
31 0.6914560
32 0.5058976
33 0.5865347
34 0.4789424
35 0.8192469
```

Berdasarkan nilai MSA terlihat bahwa keseluruan istrumen memiliki nilai MSA sebesar 0,701 (MSA>0,5) dan Sebagian besar item memiliki nilai MSA >0,5 artinya asumsi kecukupan sampel dalam analisis EFA pada kasus ini terpenuhi. Untuk item dengan nilai MSA <0,5 seperti pada Item 11 dan Item 34 dapat di drop (dikeluarkan) apabila akan dilakukan analisis lanjut. Akan tetapi ketika item tersebut di keluarkan, perlu memperhatikan keterwakilan dari variabel atau indikator yang digunakan apakah terdapat keterwakilan item atau tidak. Jika tidak ada, maka item dapat dipertahankan untuk analisis lanjut.

Analisis EFA mensyaratkan korelasi yang signifikan antar item pengukuran, agar item-item tersebut dapat dikelompokkan pada faktor tertentu. Pengujian korelasi antar item menggunakan Uji Bartlett's, jika *Chi Square* signifikan (P<0,05), bermakna bahwa item saling berkorelasi signifikan. Menggunakan dua baris syntax berikut ini, kita dapat menghasilkan statistik uji bartlet

```
bartlet = cortest.bartlett(x, nrow(data))
data.frame(bartlet)
     chisq      p.value  df
1    1349.057   0.000    595
```

Berdasarkan nilai uji Bartlett's yang dihasilkan oleh program R, menunjukkan nilai P<0,05, hal ini dapat diinterpretasikan bahwa terdapat korelasi yang signifikan antar item. Selanjutnya dilihat nilai loading factor yang diperoleh dari rotasi varimax. Nilai loading factor yang diterima adalah jika memiliki nilai di atas 0,3 (Watson, 2017).

Selanjutnya, menggunakan syntax berikut ini kita dapat menghasilkan loading faktor tiap komponen (tiap faktor).

```
component <- fa(x, nfactors = nkomp, rotate='Promax') # rotate
diganti
print(component, cut=0.03, digits = 3)
Standardized loadings (pattern matrix) based upon correlation matrix
```

```
           MR1    MR3    MR2     h2    u2  com
Item 1   0.093  0.426 -0.096 0.2247 0.775 1.20
Item 2   0.400  0.279  0.032 0.3175 0.682 1.80
Item 3  -0.193  0.491  0.064 0.2185 0.782 1.34
Item 4   0.227  0.386  0.065 0.2680 0.732 1.69
Item 5   0.489  0.084  0.133 0.2915 0.708 1.21
Item 6   0.541  0.069  0.139 0.3415 0.658 1.17
Item 7   0.090  0.429 -0.163 0.2408 0.759 1.38
Item 8                0.448 0.2022 0.798 1.01
Item 9  -0.135  0.256  0.501 0.3208 0.679 1.66
Item 10  0.117  0.271        0.1101 0.890 1.37
Item 11 -0.234  0.403  0.112 0.1670 0.833 1.78
Item 12  0.389  0.049        0.1674 0.833 1.04
Item 13  0.352  0.304  0.179 0.3265 0.674 2.47
Item 14  0.485  0.073 -0.071 0.2712 0.729 1.09
Item 15 -0.098  0.326  0.056 0.0979 0.902 1.24
Item 16  0.709        -0.096 0.5164 0.484 1.04
Item 17  0.304 -0.047  0.575 0.4084 0.592 1.53
Item 18  0.067  0.424        0.2041 0.796 1.05
Item 19 -0.071  0.401  0.083 0.1555 0.844 1.15
Item 20         0.586 -0.242 0.3972 0.603 1.33
Item 21  0.323  0.269 -0.284 0.3151 0.685 2.93
Item 22  0.332  0.404 -0.408 0.5267 0.473 2.91
Item 23  0.272 -0.032  0.063 0.0725 0.927 1.13
Item 24  0.170  0.314  0.089 0.1746 0.825 1.73
Item 25 -0.238  0.515        0.2355 0.764 1.41
Item 26  0.460 -0.125  0.060 0.1889 0.811 1.18
Item 27  0.290  0.066  0.263 0.1700 0.830 2.09
Item 28  0.544 -0.106  0.461 0.4683 0.532 2.03
Item 29  0.286  0.342 -0.272 0.3371 0.663 2.88
Item 30  0.467 -0.127 -0.139 0.2146 0.785 1.33
Item 31 -0.134 -0.154  0.374 0.1934 0.807 1.62
Item 32 -0.129  0.129 -0.286 0.1000 0.900 1.83
Item 33 -0.244  0.175 -0.319 0.1560 0.844 2.48
Item 34 -0.132         0.362 0.1493 0.851 1.26
Item 35 -0.649  0.243  0.035 0.3718 0.628 1.28

                       MR1   MR3   MR2
SS loadings          3.867 2.985 2.069
Proportion Var       0.110 0.085 0.059
Cumulative Var       0.110 0.196 0.255
Proportion Explained 0.433 0.335 0.232
Cumulative Proportion 0.433 0.768 1.000

With factor correlations of
       MR1   MR3    MR2
MR1  1.000 0.352 -0.014
MR3  0.352 1.000  0.035
MR2 -0.014 0.035  1.000
```

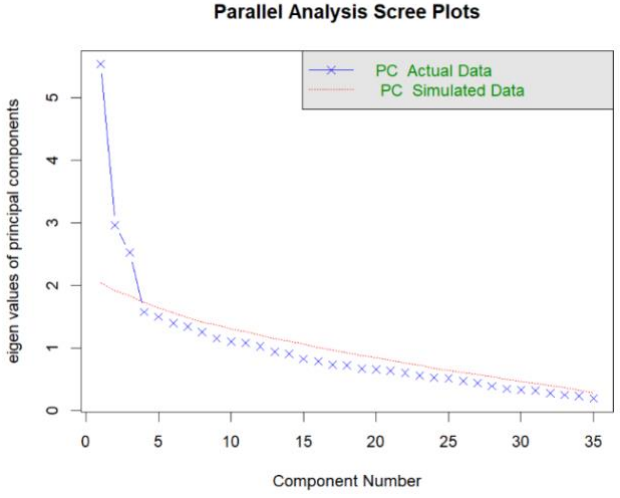
Nilai *loading factor* yang diterima adalah jika memiliki nilai di atas 0,3 (>0,3) (Watson, 2017). Namun demikian terdapat item yang memiliki nilai loading faktor <0,3 sehingga item – item tersebut dapat

didrop atau di eliminasi untuk analisis lanjutan. Misal pada Item 10, Item 23, Item 27, Item 32, Item 33, dan Item 35 (loading faktor < 0,3). Berdasarkan hasil analisis dari 35 item diperoleh 3 (tiga) faktor. Faktor 1 (MR1) berisi 11 buah item (Item 2, Item 5, Item 6, Item 12, Item 13, Item 14, Item 16, Item 21, Item 26, Item 28, dan Item 30); Faktor 2 (MR3) berisi 13 buah item (I Item 1, Item 3, Item 4, Item 7, Item 11, Item 15, Item 18, Item 19, Item 20, Item 22, Item 24, Item 25, dan Item 29); dan Faktor 3 (MR2) berisi 5 buah item (Item 8, Item 9, Item 17, Item 31, dan Item 34). Sehingga dapat disimpulkan dari 35 item pada isntrumen yang disusun terdapat 29 item yang layak untuk digunakan dalam analisis lanjutan dan terbentuk 3 faktor/komponen.

Selanjutnya untuk mengetahui relasi antara nilai eigen dengan faktornya dapat dilihat dari *Scree plot*. *Scree plot* menunjukkan banyaknya faktor yang akan diekstraksi. Kita dapat memproduksi *scree plot* menggunakan syntax berikut ini.

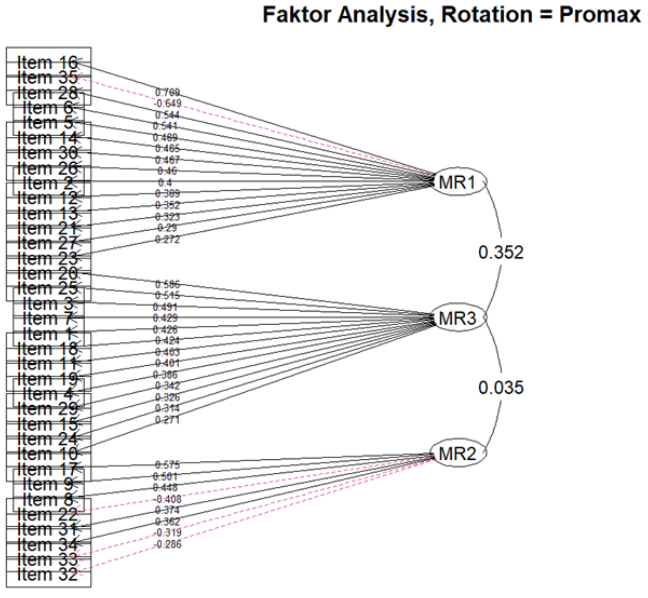
```
y <- fa1.parallel(x,
n.obs=nrow(data),fm="minres",fa="pc",nfactors=1)
```

Scree plot yang dihasilkan dari program R disajikan pada Gambar 4.1 sebagai berikut.



Gambar 4.1 Scree plot EFA

Berdasarkan Gambar 4.1 diketahui bahwa terdapat 3 (tiga) faktor yang paling dominan mengukur suatu variabel yang dianalisis. Dengan kata lain, item-item dapat dikelompokkan ke dalam 3 faktor/dimensi dengan nilai eigen di atas 1. Namun hal hal yang perlu diperhatikan jika eigen value pada faktor 1 memiliki nilai 2 kali lipat atau lebih dari faktor dua dan seterusnya, itu bermakna bahwa instrumen adalah satu dimensi. Untuk itu, perlu kita verifikasi atau perjelas, jumlah dimensi pengukuran menggunakan diagram jalur (*path diagram*) yang dihasilkan oleh analisis faktor. Diagram jalur EFA yang diperoleh menggunakan *software* R disajikan pada Gambar 4. 2.



Gambar 4.2 Diagram Jalur EFA

Berdasarkan diagram jalur pada Gambar 4.2 tampak ada 3 faktor yang terbentuk, namun ada beberapa item yang masih memiliki *loading faktor* <0,3. Pada kasus ini kita tidak mengamputasi item (*item droping*), yang semestinya dalam kasus sungguhan harus diamputasi atau dikalibrasi.

Kita juga dapat mendeteksi jumlah faktor atau jumlah komponen secara otomatis menggunakan sintak berikut ini.

```
## autodetect komponen
y <- fa.parallel(x,
n.obs=nrow(data),fm="minres",fa="pc",nfactors=1)
nkomp <- y$ncomp
nkomp <- ifelse(nkomp == 0, 1, nkomp)
nkomp
```

```
[1] 3
```

Jumlah faktor atau komponen yang dihasilkan dalam kasus ini sebanyak 3 (tiga) faktor.

Referensi

Comrey, A. L. (1988). Factor-analytic methods of scale development in personality and clinical psychology. *Journal of Consulting and Clinical Psychology*, *56*(5), 754.

Costello, A. B., & Osborne, J. (2005). Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research, and Evaluation*, *10*(1), 7.

Cudeck, R., & MacCallum, R. C. (2007). *Factor analysis at 100: Historical developments and future directions*. Routledge.

de Winter, J. C. F., Dodou, D., & Wieringa, P. A. (2009). Exploratory factor analysis with small sample sizes. *Multivariate Behavioral Research*, *44*(2), 147–181.

Elliot, A. J., McGregor, H. A., & Gable, S. (1999). Achievement goals, study strategies, and exam performance: a mediational analysis. *Journal of Educational Psychology*, *91*(3), 549.

Fabrigar, L. R., & Wegener, D. T. (2011). *Exploratory factor analysis*. Oxford University Press.

Floyd, F. J., & Widaman, K. F. (1995). Factor analysis in the development and refinement of clinical assessment instruments. *Psychological Assessment*, *7*(3), 286.

Goretzko, D., Pham, T. T. H., & Bühner, M. (2021). Exploratory factor analysis: Current use, methodological developments and recommendations for good practice. *Current Psychology*, *40*(7), 3510–3521.

Gorsuch, R. L. (1983). *Factor analysis (2nd ed.)*. Hillsdale, NJ: Erlbaum.

Gorsuch, Richard L. (1988). Exploratory factor analysis. In *Handbook of multivariate experimental psychology* (pp. 231–258). Springer.

Guadagnoli, E., & Velicer, W. F. (1988). Relation of sample size to the stability of component patterns. *Psychological Bulletin*, *103*(2), 265.

Karl, J., & Dag, S. (1994). Structural equation modeling with the SIMPLIS command language. *Scientific Software International*.

Park, H. S., Dailey, R., & Lemus, D. (2002). The use of exploratory factor analysis and principal components analysis in communication research. *Human Communication Research*, *28*(4), 562–577.

Pearson, Robert Henry & Mundform, D. J. (2010). Recommended sample size for conducting exploratory factor analysis on dichotomous data. *Journal of Modern Applied Statistical Methods, 9*((2)), 359–368. https://doi.org/10.22237/ jmasm/1288584240

Person, K. (1901). *On Lines and Planes of Closest Fit to System of Points in Space. Philiosophical Magazine, 2, 559-572.* ed.

Price, L. R. (2016). *Psychometric methods: Theory into practice*. Guilford Publications.

R Core Team. (2022a). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*.

R Core Team. (2022b). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/

Revelle, W., & Revelle, M. W. (2022). Package 'psych.' *The Comprehensive R Archive Network*, *337*, 338.

Streiner, D. L. (1994). Figuring out factors: the use and misuse of factor analysis. *The Canadian Journal of Psychiatry*, *39*(3), 135–140.

Supranto, J. (2004). Analisis Multivariat Arti dan Interpretasi. *Jakarta: Rineka Cipta*.

Tabachnick, B. G., Fidell, L. S., & Ullman, J. B. (2007). *Using multivariate statistics* (Vol. 5). pearson Boston, MA.

Team, R. C. (2013). *R: A language and environment for statistical computing*.

Watkins, M. W. (2018). Exploratory factor analysis: A guide to best practice. *Journal of Black Psychology*, *44*(3), 219–246.

Watson, J. C. (2017). Establishing evidence for internal structure using exploratory factor analysis. *Measurement and Evaluation in Counseling and Development*, *50*(4), 232–238.

Yule, G. U. (1922). *An introduction to the theory of statistics*. C. Griffin, limited.

Chapter 5
Pemodelan Rasch

Oleh: Suhariyono & Samsul Hadi

Georg Rasch adalah seorang psikometri Denmark yang memperkenalkan teori dan pendekatan pengukuran ilmu sosial dalam teks klasiknya yang berjudul Model Probabilistik untuk Beberapa Tes Kecerdasan dan Pencapaian (Rasch, 1960). Pendekatan pengukuran ini melibatkan transformasi data respons item ordinal, seperti data yang dikumpulkan dalam penilaian pendidikan pilihan ganda pemahaman siswa sekolah menengah tentang desain teknik (Alemdar et al., 2017), sebuah survei yang dirancang untuk mengukur *self-asessment* untuk membuat keputusan karir (Nam et al., 2011), atau skala diagnostik yang digunakan untuk mengidentifikasi individu dengan depresi (Shea et al., 2009). Sekarang disebut teori pengukuran Rasch, pendekatan ini didasarkan pada prinsip dan persyaratan yang mencerminkan pengukuran dalam ilmu fisika.

Rasch menggunakan istilah objektivitas spesifik (Rasch, 1977) untuk menggambarkan pentingnya mengidentifikasi situasi spesifik dimana persyaratan untuk pengukuran invarian didekati. Dalam menekankan invarian, Rasch mencatat bahwa interpretasi yang bermakna dan penggunaan instrumen pengukuran ilmu sosial tidak mungkin kecuali invarian didekati.

Selain memberikan informasi yang berguna tentang kepatuhan terhadap sifat pengukuran mendasar, model Rasch memiliki beberapa fitur teoritis dan praktis lainnya yang telah berkontribusi pada popularitasnya yang meluas di seluruh disiplin ilmu dalam ilmu sosial, perilaku, dan kesehatan.

Untuk membantu peneliti praktis memanfaatkan fitur yang berguna ini, buku kami memberikan gambaran umum tentang beberapa model utama dalam keluarga model Rasch, menawarkan panduan dasar tentang estimasi model menggunakan paket R yang tersedia, dan memberikan saran dan saran untuk menafsirkan hasil. dari analisis.

5.1 Model Rasch Dikotomi

Model Rasch dikotomi (Rasch, 1960) adalah model paling sederhana dalam keluarga model Rasch (Wright & Mok, 2004). Itu dirancang untuk digunakan dengan data ordinal yang diberi skor dalam dua kategori (biasanya 0 atau 1). Model Rasch dikotomis menggunakan skor jumlah dari tanggapan ordinal ini untuk menghitung perkiraan tingkat interval yang mewakili lokasi orang (yaitu, kemampuan orang atau pencapaian orang) dan lokasi item (yaitu, kesulitan untuk memberikan tanggapan yang benar atau positif) pada skala linier yang mewakili variabel laten (log-odds atau skala "logit"). Perbedaan antara lokasi orang dan item dapat digunakan untuk menghitung probabilitas respons yang benar atau positif (x = 1), daripada respons yang salah atau negatif (x = 0).

Persamaan untuk model Rasch dikotomus dapat dinyatakan dalam bentuk log-odds sebagai berikut:

$$ln\left[\frac{\emptyset ni1}{\emptyset ni0}\right] = \theta n - \delta i \tag{5.1}$$

Model Rasch memprediksi probabilitas orang n pada item i memberikan jawaban yang benar atau positif (x = 1), daripada jawaban yang salah atau negatif (x = 0), lokasi orang yang diberikan (yaitu, kemampuan, pencapaian, n) dan item lokasi (yaitu, kesulitan, i), seperti yang dinyatakan pada skala logit.

$$P_i(\theta) = \frac{e^{(\theta - b_i)}}{1 + e^{(\theta - b_i)}} \tag{5.2}$$

5.2 Asumsi model Rasch

Estimasi yang dihitung dengan menggunakan model Rasch dikotomis hanya dapat diinterpretasikan secara bermakna jika terdapat bukti bahwa data tersebut mendekati persyaratan untuk model tersebut. Spesifikasi (persyaratan) pemanfaatan model Rasch dikotomi adalah sebagai berikut:

- Unidimensi : Perangkat hanya mengukur satu dimensi dominan saja
- Independensi lokal : Setelah mengontrol variabel laten, tidak ada hubungan substansial antara respons terhadap item individual
- Intervensi Person: Lokasi item tidak bergantung pada (yaitu, independen dari) orang yang tanggapannya digunakan untuk memperkirakannya
- Invariansi Item : Lokasi orang tidak bergantung pada (yaitu, independen dari) item yang digunakan untuk memperkirakannya

Bukti bahwa data mendekati persyaratan ini memberikan dukungan untuk interpretasi yang berarti dan penggunaan estimasi item dan orang pada skala logit sebagai indikator lokasi item dan orang pada variabel laten. Dalam praktiknya, banyak analis mengevaluasi beberapa atau semua persyaratan ini menggunakan berbagai indikator model-data yang cocok untuk faset dalam model Rasch (dalam hal ini, item dan orang). Dalam bab ini, kami menyediakan beberapa kode dasar untuk menghitung beberapa indeks kecocokan berbasis residual yang populer untuk item dan orang.

5.3 Rasch Dikotomi Menggunakan Package TAM

Kami akan menggunakan paket *"Test Analysis Modules"*, atau "TAM" (Robitzsch et al., n.d.) untuk menjalankan analisis model Rasch dikotomi dalam bab ini. Meskipun dimungkinkan untuk menggunakan paket R lain untuk melakukan analisis model Rasch dikotomus.

Paket TAM menerapkan estimasi kemungkinan maksimum marginal (MMLE) untuk memperkirakan model Rasch dikotomi. Harap ingat pendekatan estimasi ini saat membandingkan hasil antara TAM dan paket R lain atau program perangkat lunak yang menggunakan teknik estimasi lain, seperti ltm (Rizopoulos, 2006), mirt (Osteen, 2010), eRm (Rizopoulos, 2006) selain itu juga banyak sekali paket paket yang mendukung IRT model rasch di dalam R antara lain readxl (Wickham & Jennifer Bryan, 2022), sirt (Robitzsch, 2020), psych (Revelle, 2018)

```
setwd("~/buku R")
library("stats4")
```

```
library("lattice")
library("CDM")
library("mvtnorm")
library("TAM")
library(readxl)
library("WrightMap")
respon<- read_excel("transreas.xlsx")
respon
```

Tabel 5.1 Data Respons Peserta

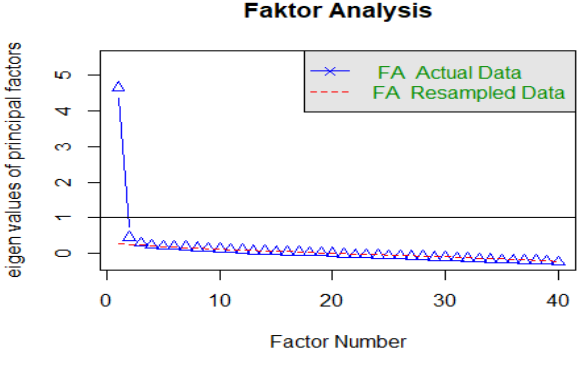
	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11
1	1	1	1	1	0	1	0	1	1	1	0
2	1	1	1	1	1	1	0	1	1	1	0
3	1	1	0	0	0	1	1	1	1	1	0
4	0	1	1	0	1	1	0	1	1	1	0
5	0	1	0	1	0	1	1	1	1	1	0
6	0	1	0	0	0	1	1	1	1	1	0
7	1	1	0	1	1	1	1	1	1	1	0
8	0	1	0	1	1	1	1	1	1	1	0
9	0	1	1	1	1	1	1	1	1	1	0
10	1	1	1	1	0	1	1	1	1	1	0

5.4 Uji Asumsi Rasch

5.4.1 Unidimensi

Unidimensi dibuktikan dengan menggunakan analisis faktor. Untuk sintak nya dapat di lihat di bawah ini. Dari gambar hasil analisis terlihat jelas bahwa hanya satu faktor yang dominan, hal ini bisa dikatakan bahwa data unidimensi dan dapat dilanjutkan untuk analisis dengan Rasch model (Rasch, 1960).

```
Faktor <- fa.parallel(respon,fm = "minres",fa = "fa", sim=F,
main="Faktor Analysis")
```



Gambar 5.1 Grafik hasil analisis faktor

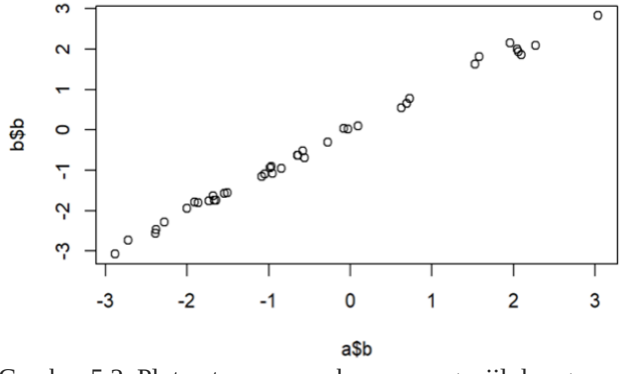
Dari hasil analisis factor terlihat jelas bahwa hanya satu faktor yang dominan, hal ini bisa dikatakan bahwa data unidimensi dan dapat dilanjutkan untuk analisis dengan Rasch model (Rasch, 1960).

5.4.2 Invariansi Parameter

a. Invariansi Parameter Person

Untuk invariansi parameter person data dibagi menjadi dua bagian yaitu data butir ganjil dan data butir genap. Untuk sintaknya dapat di buat seperti di bawah ini.

```
# Kemampuan peserta ganjil
p_ganjil <- seq(1, dim(respon)[1], 2)
d_ganjil <- respon[p_ganjil, ]
# invarian parameter
uji <- mirt(data = d_ganjil, model = 1,  itemtype = 'Rasch')
koef_ganjil <- coef(uji, simplify=TRUE, IRTpars = T)
koef_ganjil$items
# Kemampuan peserta genap
p_genap <- seq(2, dim(respon)[1], 2)
d_genap <- respon[p_genap, ]
# invarian parameter
uji <- mirt(data = d_genap, model = 1,  itemtype = 'Rasch')
koef_genap <- coef(uji, simplify=TRUE, IRTpars = T)
koef_genap$items
# plot
a <- data.frame(koef_ganjil$items)
b <- data.frame(koef_genap$items)
plot(a$b, b$b)
```



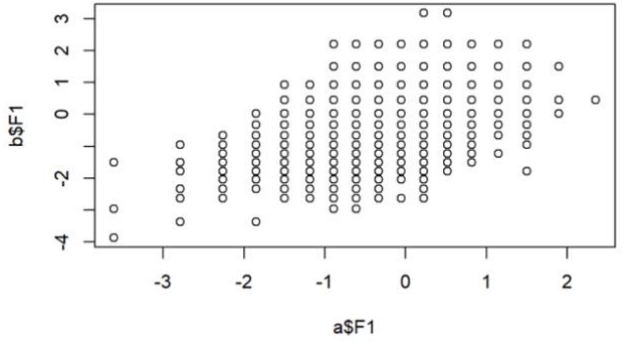
Gambar 5.2. Plot antara person bernomor ganjil dan genap

Hasil plot menunjukkan bahwa terjadi regresi antara kemampuan (*ability person*) bernomor genap dan bernomor ganjil. Hal ini mengungkapkan bahwa terjadi invariansi parameter person (Hambleton & Swaminathan, 1985b).

b. Invariansi Parameter Butir

Untuk invariansi parameter butir data dibagi menjadi dua bagian yaitu data person ganjil dan genap. Untuk sintaknya dapat di buat seperti di bawah ini. Hasil plot menunjukkan bahwa terjadi regresi antara tingkat kesulitan item bernomor genap dan bernomor ganjil. Hal ini mengungkapkan bahwa terjadi invariansi parameter item (Osteen, 2010).

```
# Tk kesulitan butir ganjil
b_ganjil <- seq(1, dim(transreas)[2], 2)
data_bgj <- transreas[ , b_ganjil]
head(data_bgj)
uji <- mirt(data = data_bgj, model = 1,  itemtype = 'Rasch')
ability_mle_bgj <- fscores(uji, method = "ML")
ability_mle_bgn <- fscores(uji, method = "ML")
# plot
a <- data.frame(ability_mle_bgj)
b <- data.frame(ability_mle_bgn)
plot(a$F1,b$F1)
```



Gambar 5.3 Invariansi Parameter Butir

Hasil plot menunjukkan bahwa terjadi regresi antara tingkat kesulitan item bernomor genap dan bernomor ganjil. Hal ini mengungkapkan bahwa terjadi invariansi parameter butir (Osteen, 2010).

1. Indenpendensi Lokal

```
mod <- sirt::rasch.mml2(transreas)
beta <- mod$item$b
mod.wle <- sirt::wle.rasch(dat= transreas , b = beta)

## WLE Reliability= 0.689

eta <- mod.wle$theta
q3 <- sirt::Q3(dat = transreas, theta = eta , b = beta)
```

Tabel 5.2 Q3 Deskriptif

M	SD	Min	10%	25%	50%	75%	90%	Max
-0.024	0.058	-0.195	-0.099	-0.057	-0.028	0.009	0.054	0.132

Interpretasi konvensional: korelasi harus mendekati nol. Nilai yang besar merupakan bukti adanya masalah dengan skala, tetapi karena kita tidak mengetahui distribusi asimtotik, kita harus mengandalkan aturan praktis untuk memutuskan kapan harus menolak model fit (Christensen et al., 2017).

5.5 Rasch Dikotomous Menggunakan R

```
mod1 <- tam(transreas)
summary(transreas)
mod1$xsi
```

Tabel 5.3 Hasil estimasi

xsi.index	xsi.label	est
1	b1	-0,895
2	b2	-1,833
3	b3	-0,022
4	b4	-1,555
5	b5	1,692
6	b6	-2,974
7	b7	0,094
8	b8	-2,285
9	b9	-2,470
10	b10	-1,844
11	b11	1,992
12	b12	-2,730
13	b13	2,927
14	b14	0,590
15	b15	2,053
16	b16	2,022
17	b17	-0,626
18	b18	-1,121

xsi.index	xsi.label	est
19	b19	-1,746
20	b20	-1,703
21	b21	-1,068
22	b22	-2,423
23	b23	0,673
24	b24	-1,657
25	b25	1,577
26	b26	-1,688
27	b27	-0,010
28	b28	-1,974
29	b29	-1,014
30	b30	2,178
31	b31	0,755
32	b32	-1,116
33	b33	-1,536
34	b34	-0,941
35	b35	-0,956
36	b36	-0,635
37	b37	-0,635
38	b38	1,971
39	b39	-0,555
40	b40	-0,292

5.5.1 Tingkat kesulitan item

```
itemDiff<-mod1$xsi$xsi
itemDiff
```

Tabel 5.4 Tingkat Kesulitan Item

[1]	-0,895	-1,833	-0,022	-1,555	1,692
[6]	-2,974	0,094	-2,285	-2,470	-1,844
[11]	1,992	-2,730	2,927	0,590	2,053
[16]	2,022	-0,626	-1,121	-1,746	-1,703
[21]	-1,068	-2,423	0,673	-1,657	1,577
[26]	-1,688	-0,010	-1,974	-1,014	2,178
[31]	0,755	-1,116	-1,536	-0,941	-0,956
[36]	-0,635	-0,635	1,971	-0,555	-0,292

Data di atas merupakan data hasil output perhitungan tingkat kesulitan item dengan paket TAM terlihat bahwa item tersulit mempunyai nilai sebesar 2,927 dan merupakan item nomor 13 dan item termudah mempunyai nilai sebesar -2,422 dan merupakan item nomor 22.

5.5.2 Kemampuan Person

Untuk mencari kemampuan person dapat digunakan sintak seperti di bawah ini.

```
abil<-tam.wle(mod1)
abil
personAbility<-abil$theta
personAbility
mean(personAbility)
sd(personAbility)
summary(personAbility)
```

Tabel 5.5 Preview Parameter Person

[1]	**0,659**	**0,847**	**1,253**	**0,141**	**0,307**	**0,307**
[7]	1,045	1,474	0,847	1,474	1,045	0,479
[13]	0,659	1,045	0,479	0,847	1,045	0,141
[19]	0,659	1,253	0,659	0,479	0,479	0,659
[25]	0,479	-0,019	-0,174	0,659	0,847	0,307
[31]	0,847	0,479	0,479	0,479	0,847	0,479
[37]	0,141	1,045	0,307	0,479	0,659	0,659
[43]	0,307	0,307	0,307	1,045	1,045	0,659
[49]	0,141	0,479	0,659	0,479	0,659	0,659
[55]	0,141	0,479	1,045	0,141	0,307	0,847
[61]	0,659	0,307	0,659	-0,174	0,141	0,659
[67]	1,045	0,479	0,847	0,307	0,659	0,141
[73]	0,659	0,847	1,045	0,307	0,847	0,659
[79]	-0,174	0,659	1,045	0,307	0,847	0,479
[85]	0,141	0,307	0,479	0,307	0,847	0,307
[91]	1,045	1,045	0,659	0,141	0,479	1,045
[97]	0,479	0,307	-0,174	1,253	0,659	0,659
[103]	0,847	1,045	0,659	0,141	0,847	0,141

Tabel 5.6 Ringkasan Person Ability

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-3,623	-0,473	0,141	-0,006	0,479	1,710

Dari perhitungan diperoleh bahwa rerata kemampuan (tetha) pada skala – 0,006.

5.5.3 Item fit

```
item.fit <- TAM::tam.fit( mod1 )
summary(item.fit)
item.fit
```

Tabel 5.7 Item Fit

parameter	Outfit	Outfit_t	Outfit_p	Infit	Infit_t	Infit_p
b1	1,161	7,241	0,000	1,121	5,499	0,000
b2	0,844	-4,025	0,000	0,934	-1,642	0,101
b3	1,010	0,806	0,421	1,008	0,651	0,515
b4	1,069	1,992	0,046	1,033	0,972	0,331
b5	1,389	9,471	0,000	1,096	2,539	0,011
b6	0,820	-2,364	0,018	0,968	-0,373	0,709
b7	1,031	2,569	0,010	1,025	2,055	0,040
b8	0,902	-1,869	0,062	0,969	-0,556	0,578
b9	0,767	-4,194	0,000	0,941	-0,989	0,323
b10	0,816	-4,767	0,000	0,913	-2,168	0,030
b11	1,443	8,678	0,000	1,091	1,972	0,049
b12	0,852	-2,205	0,027	0,963	-0,514	0,607
b13	1,291	3,372	0,001	1,026	0,353	0,724
b14	1,217	13,166	0,000	1,156	9,605	0,000
b15	1,467	8,738	0,000	1,083	1,747	0,081
b16	1,641	11,748	0,000	1,118	2,498	0,012
b17	0,904	-5,771	0,000	0,919	-4,817	0,000
b18	0,811	-8,052	0,000	0,876	-5,164	0,000
b19	0,901	-2,639	0,008	0,949	-1,332	0,183
b20	0,931	-1,864	0,062	0,964	-0,952	0,341
b21	0,991	-0,385	0,700	0,993	-0,303	0,762
b22	0,709	-5,541	0,000	0,909	-1,590	0,112
b23	1,123	7,158	0,000	1,086	5,068	0,000
b24	0,806	-5,706	0,000	0,903	-2,756	0,006
b25	1,216	5,971	0,000	1,079	2,294	0,022
b26	0,817	-5,241	0,000	0,912	-2,433	0,015
b27	1,015	1,218	0,223	1,012	1,029	0,303
b28	0,875	-2,922	0,003	0,940	-1,352	0,176
b29	0,963	-1,611	0,107	0,973	-1,179	0,238
b30	1,392	6,898	0,000	1,067	1,316	0,188
b31	1,226	11,961	0,000	1,134	7,291	0,000
b32	0,871	-5,412	0,000	0,912	-3,624	0,000
b33	0,828	-5,425	0,000	0,907	-2,850	0,004
b34	0,993	-0,325	0,745	0,993	-0,305	0,760
b35	0,927	-3,352	0,001	0,946	-2,451	0,014
b36	0,907	-5,535	0,000	0,925	-4,461	0,000
b37	0,970	-1,731	0,083	0,974	-1,535	0,125
b38	1,404	8,100	0,000	1,083	1,832	0,067
b39	0,883	-7,504	0,000	0,899	-6,401	0,000
b40	0,989	-0,831	0,406	0,988	-0,906	0,365

Osteen (2010) memberikan *rule of thumb* untuk menilai implikasi kecocokan model terhadap pengukuran rasch yaitu nilai outfit antara $0{,}5 \leq \text{MNSQ} \leq 1{,}5$. Sehingga dalam hasil tersebut, diperoleh informasi bahwa ada satu item yang tidak fit yaitu item 16 karena mempunyai nilai 1,641.

5.5.4 Person fit

```
person.fit <- tam.personfit(mod1)
person.fit
summary(person.fit
```

Tabel 5.8 Preview person fit

	outfitPerson	outfitPerson_t	infitPerson	infitPerson_t
1	0,382	-1,917	0,556	-2,060
2	0,494	-1,275	0,714	-1,150
3	0,751	-0,317	0,732	-1,000
4	0,882	-0,271	1,004	0,089
5	0,462	-1,863	0,586	-2,063
6	0,776	-0,593	0,880	-0,468
7	0,521	-1,035	0,770	-0,855
8	0,702	-0,337	0,955	-0,072
9	0,761	-0,443	0,995	0,065
10	0,447	-0,942	0,781	-0,765
11	1,021	0,207	1,058	0,304
12	0,684	-0,843	0,716	-1,240
13	0,664	-0,817	0,847	-0,568
14	0,434	-1,326	0,526	-2,092
15	0,747	-0,629	0,948	-0,144

Tabel 5.9 Ringkasan dari person fit

outfitPerson	outfitPerson_t	infitPerson	infitPerson_t
Min, :0,248	Min, :-2,652	Min, :0,392	Min, :-3,089
1st Qu,:0,604	1st Qu,:-1,141	1st Qu,:0,723	1st Qu,:-1,250
Median :0,869	Median :-0,281	Median :0,923	Median :-0,3055
Mean :0,997	Mean :-0,098	Mean :0,955	Mean :-0,1484
3rd Qu,:1,253	3rd Qu,: 0,804	3rd Qu,:1,152	3rd Qu,: 0,8274
Max, :4,142	Max, : 4,947	Max, :2,088	Max, : 5,4827

Dari hasil perhitungan dapat dilihat bahwa kemampuan person berada dalam rentang teta sebesar 0,248 hingga 4,142.

5.6. Rasch Dikotomus Menggunakan *Package* MIRT

Untuk menggunakan paket mirt Langkah Langkah analisis dapat di lakukan seperti di bawah ini.

```
setwd("~/buku R")
library(mirt)
LSAT <- read_excel("transreas.xlsx")
head(LSAT)
results.rasch <- mirt(data=LSAT, model=1, itemtype="Rasch",
SE=TRUE,verbose=FALSE)
coef.rasch <- coef(results.rasch, IRTpars=TRUE, simplify=TRUE)
items.rasch <- as.data.frame(coef.rasch$items)
print(items.rasch
```

Tabel 5.10 Hasil perhitungan rasch dengan paket mirt

	a	b	g	u
b1	1	-0,895	0	1
b2	1	-1,833	0	1
b3	1	-0,022	0	1
b4	1	-1,555	0	1
b5	1	1,692	0	1
b6	1	-2,974	0	1
b7	1	0,094	0	1
b8	1	-2,284	0	1
b9	1	-2,470	0	1
b10	1	-1,844	0	1
b11	1	1,992	0	1
b12	1	-2,729	0	1
b13	1	2,927	0	1
b14	1	0,590	0	1
b15	1	2,053	0	1
b16	1	2,022	0	1
b17	1	-0,626	0	1
b18	1	-1,121	0	1
b19	1	-1,746	0	1
b20	1	-1,702	0	1
b21	1	-1,068	0	1
b22	1	-2,422	0	1
b23	1	0,673	0	1
b24	1	-1,657	0	1
b25	1	1,577	0	1
b26	1	-1,688	0	1
b27	1	-0,010	0	1
b28	1	-1,974	0	1
b29	1	-1,014	0	1
b30	1	2,177	0	1
b31	1	0,754	0	1
b32	1	-1,115	0	1
b33	1	-1,536	0	1
b34	1	-0,941	0	1
b35	1	-0,956	0	1
b36	1	-0,635	0	1
b37	1	-0,635	0	1
b38	1	1,971	0	1
b39	1	-0,555	0	1
b40	1	-0,291	0	1

Hasil perhitungan paket *mirt* berupa nilai a,b,g dan u di mana a merupakan daya beda sedang b merupakan tingkat kesulitan g merupakan tebakan semu dan u merupakan model disini karena model Rasch maka nilai u adalah 1. Karena ini model rasch maka nilai dari a semua 1 sedangkan nilai g semua nol karena nilai g akan muncul jika kita menggunakan 3PLItemfit.

```
itemfit(results.rasch, fit_stats = "S_X2")
```

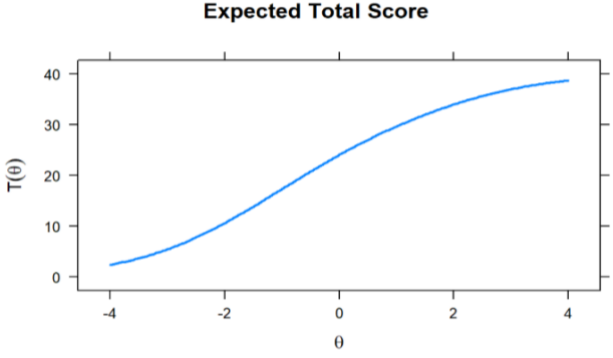
Tabel 5.11 Hasil perhitungan item fit dengan paket mirt

	item	S_X2	df.S_X2	RMSEA.S_X2	p.S_X2
1	b1	194,431	21	0,063	0,000
2	b2	70,404	22	0,032	0,000
3	b3	22,800	21	0,006	0,355
4	b4	44,468	21	0,023	0,002
5	b5	326,356	19	0,088	0,000
6	b6	28,369	21	0,013	0,130
7	b7	57,520	21	0,029	0,000
8	b8	26,135	22	0,009	0,246
9	b9	64,964	21	0,032	0,000
10	b10	75,924	22	0,034	0,000
11	b11	301,324	18	0,087	0,000
12	b12	28,802	21	0,013	0,119
13	b13	68,483	15	0,041	0,000
14	b14	332,264	20	0,086	0,000
15	b15	349,844	18	0,094	0,000
16	b16	607,789	18	0,125	0,000
17	b17	112,555	21	0,046	0,000
18	b18	186,028	21	0,061	0,000
19	b19	31,419	21	0,015	0,067
20	b20	30,432	21	0,015	0,084
21	b21	30,698	21	0,015	0,079
22	b22	91,708	21	0,040	0,000
23	b23	110,804	20	0,046	0,000
24	b24	99,612	21	0,042	0,000
25	b25	149,906	19	0,057	0,000
26	b26	88,579	21	0,039	0,000
27	b27	35,525	21	0,018	0,025
28	b28	41,788	22	0,021	0,007
29	b29	32,974	21	0,016	0,047
30	b30	200,280	18	0,069	0,000
31	b31	296,932	20	0,081	0,000
32	b32	106,432	21	0,044	0,000
33	b33	101,167	21	0,043	0,000
34	b34	19,818	21	0,000	0,533
35	b35	43,686	21	0,023	0,003
36	b36	104,384	21	0,043	0,000
37	b37	39,333	21	0,020	0,009
38	b38	273,466	18	0,082	0,000
39	b39	140,509	21	0,052	0,000
40	b40	35,383	21	0,018	0,026

Dari hasil menunjukkan bahwa ada item yang fit dan ada item yang tidak fit dimana item yang fit yang mempunyai nilai RMSEA lebih kecil dari 0,08 (Anderson & Gerbing, 1988) dan nilai p value lebih kecil dari 0,05 (Hulland, 1999).

5.6.1 Plot expected total score

```
plot(results.rasch, type = 'score', theta_lim = c(-4,4),
lwd=2) #expected total score
```

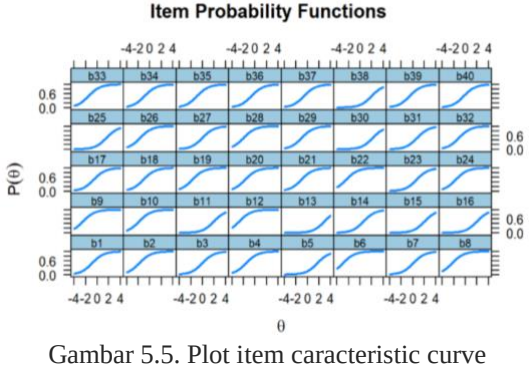


Gambar 5.4 Plot total skor harapan

Plot *expected total score* adalah gambaran total skore teta dan tingkat kesulitan. Dimana sumbu x merupakan tingkat kesulitan dan sumbu y merupakan total skor yang di peroleh peserta tes.

5.6.2 Plot Item Caracteristic Curve

```
plot(results.rasch, type = 'trace', theta_lim = c(-4,4),
lwd=2) #Item Characteristic Curves
```

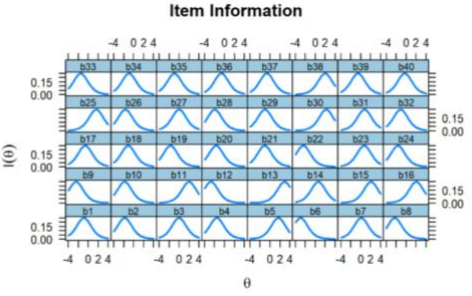


Gambar 5.5. Plot item caracteristic curve

Gambar di atas merupakan gambar kurva karakteristik semua item mulai dari item nomor 1 hingga nomor 40. Kurva karakteristik merupakan gambaran atau visualisasi dari item soal dan nilai tetha atau kemampuan peserta tes.

5.6.3 Item Information Function

```
plot(results.rasch, type = 'infotrace', theta_lim = c(-
4,4), lwd=2) #Item Information Function
```



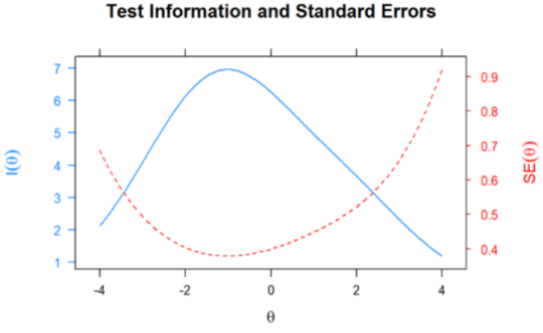
Gambar 5.6 Grafik *Item Information Function*

Gambar di atas merupakan gambar fungsi informasi setiap item. Fungsi informasi ini merupakan informasi dimana item tersebut memberikan gambaran tentang kecocokan item dengan kemampuan peserta test.

5.6.4 *Test Information Fuction*

Informasi butir tes dapat dilihat dari fungsi informasi tes (*test information function, TIF*) yang dihasilkan dari syntax R berikkut ini.

```
plot(results.rasch, type = 'infoSE', theta_lim = c(-4,4),
lwd=2) #Test Information Function + Standard Error
```



Gambar 5.7. Fungsi informasi tes

Dari hasil gambar *Test Information Function* menunjukkan bahwa tes ini sangat cocok untuk peserta dengan theta – 3,9 hingga 2,3. Di lihat dari perpotongan antara nilai I (θ) dengan nilai SEM (θ).

5.7 Analisis Butir Model Rasch Dikotomi Menggunakan Paket eRm

```
setwd("~/buku R")
library(eRm)
library(readxl)
transreas <- read_excel("transreas.xlsx")
transreas
results.rasch <- RM(transreas)
results.rasch
coef(results.rasch)
```

5.7.1 Item Parameter

Tabel 5.12 Hasil paket eRm

	b1	b2	b3	b4	b5	b6
Estimate	0,389	-1,338	0,485	-1,057	2,172	-2,493
Std.Err	0,049	0,062	0,045	0,057	0,058	0,094
	b7	b8	b9	b10	b11	b12
Estimate	0,600	-1,795	-1,983	-1,349	2,467	-22,456
Std.Err	0,045	0,072	0,077	0,062	0,064	0,085
	b13	b14	b15	b16	b17	b18
Estimat	33,894	1,090	2,526	2,496	-0,119	-0,617
Std.Err	0,091	0,047	0,065	0,065	0,047	0,051
	b19	b20	b21	b22	b23	b24
Estimate	-1,250	-1,206	-0,564	-1,935	1,172	-1,159
Std.Err	0,060	0,059	0,051	0,076	0,047	0,059
	b25	b26	b27	b28	b29	b30
Estimate-	2,059	0,497	-1,481	-0,509	2,649	1,252
Std.Err	0,056	0,059	0,045	0,065	0,050	0,068
	b31	b32	b33	b34	b35	b36
Estimate	1.191	-0,612	-1,037	-0,436	-0,451	-0,128
Std.Err	0,047	0,051	0,057	0,050	0,050	0,047
	b37	b38	b39	b40		
Estimate	-0,128	2,446	-0,047	0,216		
Std.Err	0,047	0,063	0,047	0,046		

Dari hasil *output* di dapat bahwa item termudah adalah item nomor 12 dengan nilai sebesar -3,389 dan item tersulit adalah item nomor 6 dengan nilai sebesar 2,492. Karena keduanya berada dalam luar rentang antara -2 < b < +2.

5.7.2 Person parameter

```
pres.rasch <- person.parameter(results.rasch)
pres.rasch
```

Tabel 5.13 Hasil perhitungan person parameter

Raw Score	Estimate	Std.Error
3	-3,257	0,627
5	-2,624	0,512
7	-2,162	0,455
8	-1,964	0,436
9	-1,780	0,422
10	-1,607	0,410
11	-1,442	0,401
12	-1,284	0,394
13	-1,131	0,389
14	-0,982	0,385
15	-0,835	0,382
16	-0,690	0,380
17	-0,546	0,379
18	-0,402	0,379
19	-0,258	0,380
20	-0,113	0,382
21	0,035	0,385
22	0,184	0,389
23	0,337	0,393
24	0,494	0,399
25	0,656	0,405
26	0,823	0,413
27	0,997	0,421
28	1,177	0,430
29	1,367	0,440
30	1,566	0,452
31	1,776	0,465
32	1,999	0,481
33	2,239	0,499

Dari hasil output menunjukkan bahwa kemampuan (tetha) bergerak dari -3, 257 hingga 2,239

5.7.3 Item Fit

```
itemfit(pres.rasch)
```

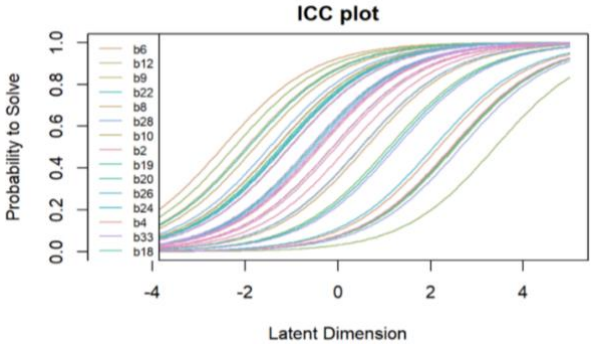
Tabel 5.14 Item fit paket eRm

	Chisq	df	p-value	Outfit MSQ	Infit MSQ	Outfit t	Infit t	Discrim
b1	2.543,889	2101	0,000	1,210	1,177	7,046	7,469	-0,003
b2	1.603,116	2101	1,000	0,763	0,887	-4,763	-2,931	0,428
b3	2.092,354	2101	0,549	0,995	0,994	-0,256	-0,410	0,271
b4	2.244,832	2101	0,015	1,068	1,040	1,485	1,184	0,171
b5	3.198,918	2101	0,000	1,522	1,072	8,750	2,041	-0,112
b6	1.541,787	2101	1,000	0,733	0,916	-2,761	-1,100	0,288
b7	2.161,732	2101	0,174	1,028	1,017	1,578	1,276	0,216
b8	1.769,229	2101	1,000	0,842	0,934	-2,323	-1,272	0,300
b9	1.371,912	2101	1,000	0,653	0,885	-4,978	-2,043	0,398
b10	1.549,196	2101	1,000	0,737	0,858	-5,297	-3,701	0,476
b11	3.339,284	2101	0,000	1,589	1,059	8,103	1,383	-0,130
b12	1.616,495	2101	1,000	0,769	0,911	-2,702	-1,348	0,303
b13	2.833,932	2101	0,000	1,348	0,952	3,042	-0,620	-0,017
b14	2.749,715	2101	0,000	1,308	1,199	11,783	12,013	-0,098
b15	3.422,034	2101	0,000	1,628	1,047	8,267	1,071	-0,135
b16	4.087,708	2101	0,000	1,945	1,100	11,834	2,262	-0,251
b17	1.785,221	2101	1,000	0,849	0,877	-6,958	-6,796	0,470
b18	1.516,580	2101	1,000	0,721	0,815	-9,297	-7,638	0,589
b19	1.757,018	2101	1,000	0,836	0,911	-3,378	-2,396	0,388
b20	1.835,501	2101	1,000	0,873	0,933	-2,643	-1,851	0,355
b21	2.025,870	2101	0,878	0,964	0,984	-1,137	-0,618	0,285
b22	1.237,107	2101	1,000	0,589	0,832	-6,257	-3,152	0,492
b23	2.469,675	2101	0,000	1,175	1,101	6,535	5,952	0,041
b24	1.498,301	2101	1,000	0,713	0,842	-6,604	-4,633	0,519
b25	2.652,857	2101	0,000	1,262	1,059	5,059	1,840	-0,016
b26	1.526,892	2101	1,000	0,726	0,862	-6,124	-3,928	0,482
b27	2.120,799	2101	0,376	1,009	1,004	0,514	0,309	0,246
b28	1.702,064	2101	1,000	0,810	0,895	-3,422	-2,480	0,392
b29	1.951,936	2101	0,991	0,929	0,958	-2,375	-1,755	0,339
b30	3.143,142	2101	0,000	1,495	1,021	6,262	0,445	-0,072
b31	2.770,885	2101	0,000	1,318	1,154	10,735	8,471	-0,072
b32	1.674,756	2101	1,000	0,797	0,864	-6,600	-5,514	0,494
b33	1.550,851	2101	1,000	0,738	0,848	-6,467	-4,778	0,510
b34	2.027,576	2101	0,872	0,965	0,984	-1,225	-0,684	0,297
b35	1.851,704	2101	1,000	0,881	0,918	-4,220	-3,613	0,406
b36	1.795,412	2101	1,000	0,854	0,884	-6,677	-6,351	0,457
b37	1.974,666	2101	0,976	0,939	0,955	-2,684	-2,388	0,341
b38	3.231,785	2101	0,000	1,537	1,051	7,583	1,204	-0,113
b39	1.724,727	2101	1,000	0,821	0,850	-8,824	-8,775	0,517
b40	2.045,042	2101	0,805	0,973	0,976	-1,469	-1,563	0,303

Dari hasil perhitungan dapat dilihat bahwa item ada yang fit dan ada yang tidak fit dimana item yang fit dapat dilihat dari nilai Outfit

MNSQ antara 0,5 <MNSQ<1,5 dan Outfit Z-standard antara -2,0<ZSTD<2,0 (Boone et al., 2013).

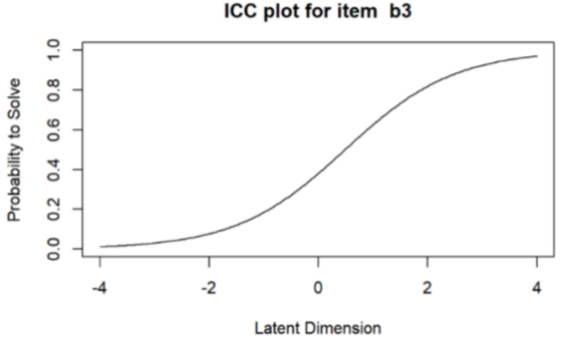
```
plotjointICC(results.rasch, xlim = c(-5, 5))
```



Gambar 5.8 Kurva karakteristik seluruh butir

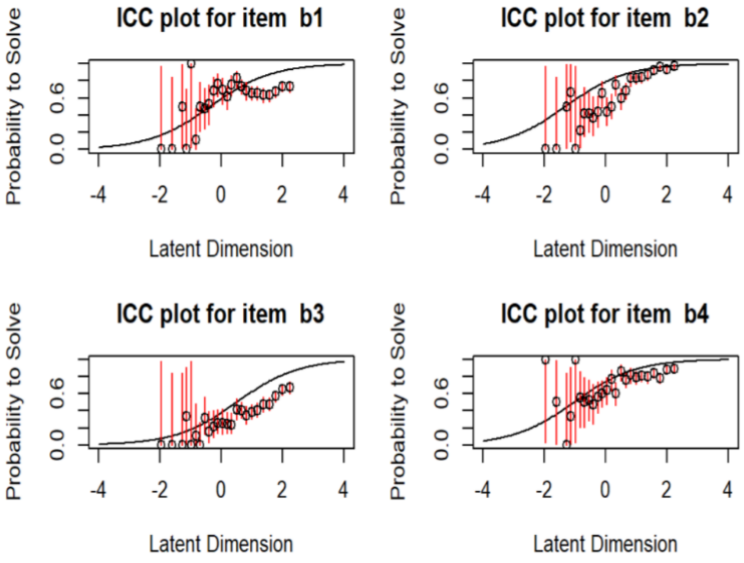
Gambar di atas merupakan gambar kurva karasteristik dari semua item. Kurva karakteristik merupakan gambaran atau visualisasi dari nilai tingkat kesukaran dengan kemampuan peserta tes.

```
plotICC(results.rasch, i = 3)
```



Gambar 5.8 Kurva karakteristik butir 3

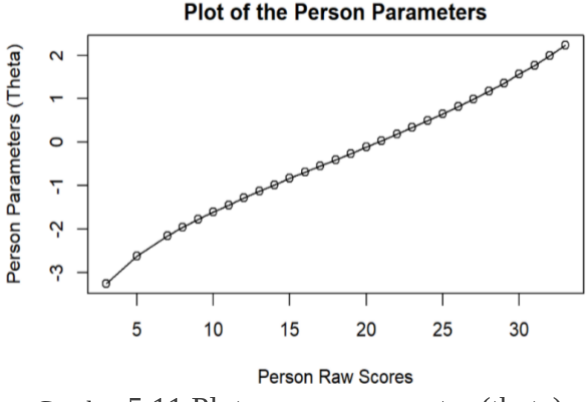
```
plotICC(results.rasch, item.subset = 1:4, ask = F, empICC =
list("raw"),
empCI = list(lty = "solid"))
```



Gambar 5.10 Kurva karakteristik Butir 1-4.

Dari gambar di atas dapat dilihat bahwa hubungan antara person parameter (tetha) dengan grafik ICC dimana garis – garis merah merupakan standar errornya.

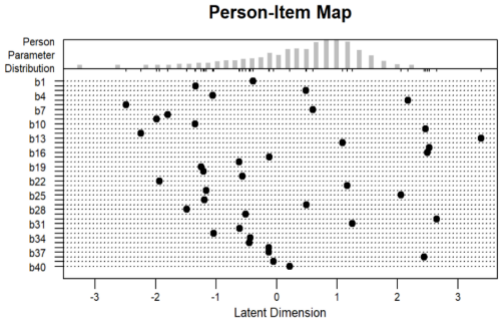
```
plot(pres.rasch)
```



Gambar 5.11 Plot person parameter (theta)

Dari gambar dapat dilihat bahwa person parameter (tetha) bergerak dari -3 hingga 2 dan gambar tersebut merupakan hubungan antara theta dengan raw skore (skor yang di peroleh).

```
plotPImap(results.rasch)
```



Gambar 5.12 Peta Item Personal

Gambar di atas menunjukkan hubungan antara distribusi person parameter (tetha) dengan item.

5.8 Rasch Dikotomus Menggunakan Package ltm

```
setwd("~/buku R")
library(ltm)
library(readxl)
transreas <- read_excel("transreas.xlsx")
transreas
model<-rasch(transreas)
model
```

Tabel 5.15 Hasil perhitungan dengan paket ltm

Dffclt,b1	**Dffclt,b2**	**Dffclt,b3**	**Dffclt,b4**	**Dffclt,b5**	**Dffclt,b6**
-1,438	-2,944	-0,035	-2,499	2,719	-4,778
Dffclt,b7	Dffclt,b8	Dffclt,b9	Dffclt,b10	Dffclt,b11	Dffclt,b12
0,151	-3,670	-3,968	-2,963	3,201	-4,385
Dffclt,b13	Dffclt,b14	Dffclt,b15	Dffclt,b16	Dffclt,b17	Dffclt,b18
4,704	0,949	3,299	3,250	-1,005	-1,801
Dffclt,b19	Dffclt,b20	Dffclt,b21	Dffclt,b22	Dffclt,b23	Dffclt,b24
-2,805	-2,735	-1,717	-3,892	1,082	-2,662
Dffclt,b25	Dffclt,b26	Dffclt,b27	Dffclt,b28	Dffclt,b29	Dffclt,b30
2,534	-2,713	-0,016	-3,172	-1,630	3,499
Dffclt,b31	Dffclt,b32	Dffclt,b33	Dffclt,b34	Dffclt,b35	Dffclt,b36
1,212	-1,792	-2,467	-1,512	-1,535	-1,022
Dffclt,b37	Dffclt,b38	Dffclt,b39	Dffclt,b40	Dscrmn	
-1,019	3,167	-0,891	-0,468	0,622	

Dari hasil perhitungan menggunakan paket ltm diperoleh bahwa item termudah adalah item nomor 6 dengan nilai sebesar -4,778 dan item tersulit adalah item nomor 13 dengan nilai sebesar 4,704.

5.8.1 Hitung kecocokan model

```
GoF.rasch(model)
## Bootstrap Goodness-of-Fit using Pearson chi-squared
## Call:
## rasch(data = transreas, IRT.param = TRUE)
## Tobs: 4.582269e+15
## data-sets: 50
## p-value: 0.02
```

Nilai p value 0,02 dan lebih kecil dari 0,05 berarti model fit (Hulland, 1999)

5.8.2 Item Difficulty Level

```
coef(model,prob=TRUE, order=TRUE)
```

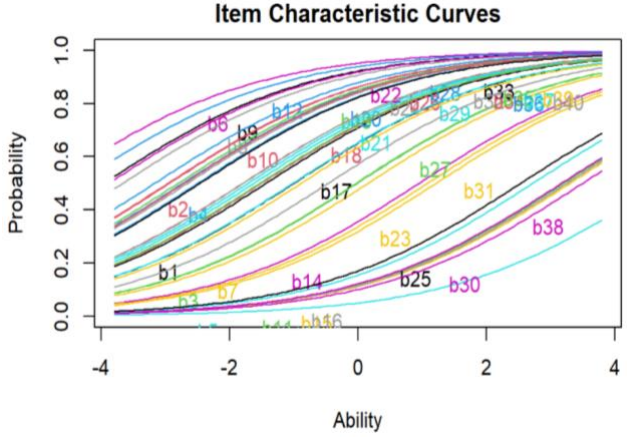
Tabel 5.16 Tingkat kesulitan item paket ltm

	Dffclt	Dscrmn	P(x=1|z=0)
b6	-4,778	0,622	0,951
b12	-4,385	0,622	0,939
b9	-3,968	0,622	0,922
b22	-3,892	0,622	0,918
b8	-3,670	0,622	0,908
b28	-3,172	0,622	0,878
b10	-2,963	0,622	0,863
b2	-2,944	0,622	0,862
b19	-2,805	0,622	0,851
b20	-2,735	0,622	0,846
b26	-2,713	0,622	0,844
b24	-2,662	0,622	0,840
b4	-2,499	0,622	0,826
b33	-2,467	0,622	0,823
b18	-1,801	0,622	0,754
b32	-1,792	0,622	0,753
b21	-1,717	0,622	0,744
b29	-1,630	0,622	0,734
b35	-1,535	0,622	0,722
b34	-1,512	0,622	0,719
b1	-1,438	0,622	0,710
b36	-1,022	0,622	0,654
b37	-1,019	0,622	0,653
b17	-1,005	0,622	0,652
b39	-0,891	0,622	0,635
b40	-0,468	0,622	0,572
b3	-0,035	0,622	0,505
b27	-0,016	0,622	0,502
b7	0,151	0,622	0,476
b14	0,949	0,622	0,357

	Dffclt	Dscrmn	P(x=1|z=0)
b23	1,082	0,622	0,338
b31	1,212	0,622	0,320
b25	2,534	0,622	0,171
b5	2,719	0,622	0,155
b38	3,167	0,622	0,122
b11	3,201	0,622	0,120
b16	3,250	0,622	0,117
b15	3,299	0,622	0,114
b30	3,499	0,622	0,102
b13	4,704	0,622	0,051

Dari perhitungan rasch menggunakan paket ltm dihasilkan bahwa tingkat kesulitan item bergerak dari -4,777 hingga 4,704.

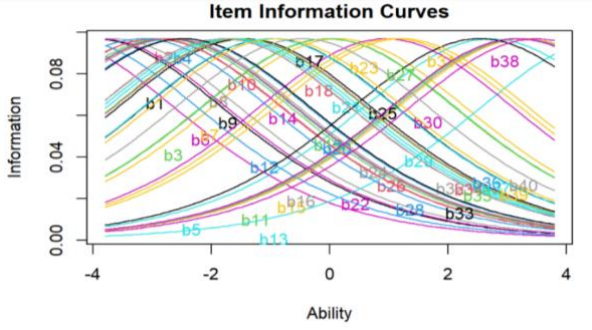
```
plot(model,type="ICC")
```



Gambar 5.13. Item characteristic curve

Pada grafik yang terletak di atas, Anda akan melihat bahwa ada sumbu. Nilai tingkat kemampuan *(Ability)* ditunjukkan pada sumbu horizontal (sumbu x), sedangkan kemungkinan untuk mengerjakan item secara tepat direpresentasikan pada sumbu vertikal (sumbu Y) dalam grafik ini (Probabilitas). Jika penggeser di paling kanan skala kesulitan soal digeser lebih jauh ke kanan, maka skor akan meningkat (Kemampuan).

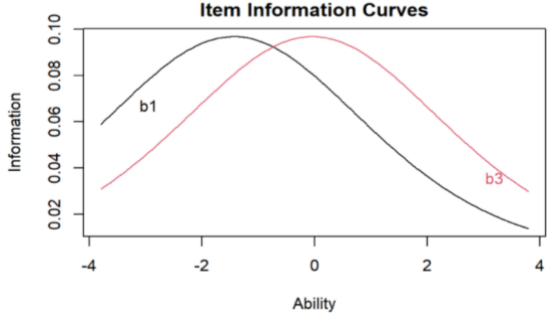
```
plot(model,type="IIC") # IIC untuk semua item
```



Gambar 5.14. Item Information curva

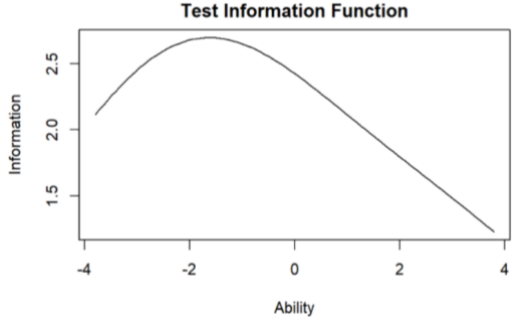
Gambar 5.14 merupkan gambar *Item Information Curve* yang merupakan gambaran atau visualisasi dari tingkat kesulitan item dengan posisi kemampuan peserta tes.

```
plot(model,type="IIC", items=c(1,3)) # item individual
```



Gambar 5.15. Item Information curve item 1 dan 3

```
plot(model,type="IIC", items=0) # area keseluruhan pengukuran item
```



Gambar 5.16. Test Information Curve

Dari hasil gambar Test Information Function menunjukkan bahwa tes ini sangat cocok untuk peserta dengan theta – 3,9 hingga 3,9. Di lihat dari garis permulaan yang mendekati nilai -4 hingga garis akhir yang mendekati angka 4.

References

Alemdar, M., Lingle, J. A., Wind, S. A., & Moore, R. A. (2017). Developing an engineering design process assessment using think-aloud interviews. *International Journal of Engineering Education*, *33*(1), 441–452.

Anderson, J. C., & Gerbing, D. W. (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin*, *103*(3), 411.

Boone, W. J., Staver, J. R., & Yale, M. S. (2013). *Rasch analysis in the human sciences*. Springer.

Christensen, K. B., Makransky, G., & Horton, M. (2017). Critical values for Yen's Q 3: Identification of local dependence in the Rasch model using residual correlations. *Applied Psychological Measurement*, *41*(3), 178–194.

Hambleton, R. K., & Swaminathan, H. (1985). *Item Response Thery Principles and Applications*. Kluwer Nijhoff Publishng.

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strategic Management Journal*, *20*(2), 195–204.

Nam, S. K., Yang, E., Lee, S. M., Lee, S. H., & Seol, H. (2011). A psychometric evaluation of the career decision self-efficacy scale with Korean students: A Rasch model approach. *Journal of Career Development*, *38*(2), 147–166.

Osteen, P. (2010). An Introduction to Using Multidimensional Item Response Theory to Assess Latent Factor Structures. *Journal of the Society for Social Work and Research*, *1*(2), 66–82. https://doi.org/10.5243/JSSWR.2010.6

Rasch, G. (1960). *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.*

Rasch, G. (1977). On specific objectivity an attempt at formalizing the request for generality and validity of scientific statements. *Danish Yearbook of Philosophy*, *14*(1), 58–94.

Revelle, W. (2018). *psych: procedures for personality and psychological research. Northwestern University, Evanston*.

Rizopoulos, D. (2006). Itm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1–25. https://doi.org/10.18637/jss.v017.i05

Robitzsch, A, Kiefer, T., & Wu, M. (n.d.). *TAM: Test Analysis Modules. 2020*.

Robitzsch, Alexander. (2020). sirt: Supplementary item response theory models. R package version 3.9-4. *Computer Software]. Retrieved from Https://CRAN. R-Project. Org/Package= Sirt*.

Shea, T. L., Tennant, A., & Pallant, J. F. (2009). Rasch model analysis of the Depression, Anxiety and Stress Scales (DASS). *BMC Psychiatry*, *9*(1), 1–10.

Wickham, H., & Jennifer Bryan. (2022). *readxl: Read Excel Files*. https://cran.r-project.org/package=readxl

Wright, B. D., & Mok, M. M. C. (2004). An overview of the family of Rasch measurement models. *Introduction to Rasch Measurement*, 1–24.

Chapter 6
IRT Unidimensi Penskoran Dikotomi

Oleh: Hasan Djidu & Heri Retnawati

Teori respons item atau *item response theory* (IRT) merupakan salah satu perkembangan paling berpengaruh di bidang pengukuran pendidikan dan psikologis. IRT menyediakan dasar untuk metode statistik yang digunakan dalam konteks seperti pengembangan tes, analisis item, penyetaraan, bank soal, dan pengujian adaptif terkomputerisasi atau biasa dikenal dengan CAT. Penerapannya juga meluas ke pengukuran berbagai konstruksi laten dalam berbagai disiplin ilmu.

Pada bab ini, akan diperkenalkan model IRT untuk penskoran item dikotomi. Bab ini diawali dengan deskripsi singkat mengenai kerangka dalam IRT dan perbandingannya dengan teori pendahulunya yakni CTT, kekuatan, kelemahan dan karakteristiknya masing-masing. Selanjutnya, pembaca akan disuguhkan: pengantar singkat model IRT unidimensional, model matematika, parameter-parameter dalam IRT unidimensi untuk membantu dalam interpretasi; dan terakhir akan ditampilkan contoh-contoh estimasi dalam pemodelan IRT unidimensi menggunakan program R.

Pemodelan IRT pada bab ini akan mencakup model Rasch, IRT 1-parameter, IRT 2-parameter, IRT 3-parameter, dan IRT 4-parameter. Ada banyak *packages* dapat digunakan dalam menjalankan pemodelan IRT pada R yang dapat ditemukan di CRAN, diantaranya ‘eRm’ (Hatzinger & Rusch, 2009; Mair et al., 2021; Mair & Hatzinger, 2007b, 2007a), ‘ltm’ (Rizopoulos, 2022), dan ‘mirt’ (Chalmers, 2012, 2021). Pada Bab ini, pembaca akan diberikan contoh pemodelan IRT dengan menggunakan *packages* ‘mirt’. Penulis memilih menggunakan mirt karena *packages* ini paling komprehensif dalam mengestimasi berbagai model IRT (Desjardins & Bulut, 2018).

6.1 Teori Respons Butir vs Teori Tes Klasik

Pada Bab 3 telah diperkenalkan model *classical test theory* (CTT), karakteristik dan bagaimana menghitung parameter-parameternya.

Sejak perkembangannya pada 1950-an dan 1960-an (Hambleton & Swaminathan, 1985a; Retnawati, 2014a), IRT telah menjadi metodologi statistik pilihan untuk analisis butir dan pengembangan tes. Keberhasilan IRT atas CTT pendahulunya terutama berasal dari fokus IRT pada komponen yang membentuk tes, yaitu item itu sendiri. Dengan memodelkan hasil pada tingkat item, bukan pada tingkat tes seperti pada CTT, IRT lebih kompleks tetapi juga lebih komprehensif dalam hal informasi yang diberikannya tentang kinerja tes. Diantara perbedaan mendasar antara CTT dan IRT antara lain sebagai berikut.

Pertama, IRT adalah model probabilitas dari respons *examinee* pada suatu item berdasarkan level pada atribut laten yang diukur. Probabilitas menjawab benar atau memilih, dimodelkan dengan fungsi monoton naik dari variabel/atribut laten yang diukur (dinotasikan dengan θ). Fungsi ini bergantung pada tingkat kesulitan, daya pembeda, dan tebakan. IRT tidak fokus pada skor amatan (seperti pada CTT), tetapi fokus pada hubungan antara item dengan atribut laten yang diukur.

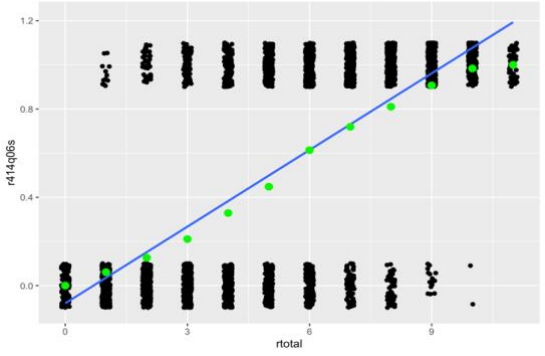
Kedua, atribut laten yang diestimasi dari model IRT diasumsikan independen atau tidak bergantung pada item pada tes dan karakteristik peserta tes. Sementara itu, skor yang diperoleh pada CTT sangat bergantung pada karakteristik butir yang digunakan dalam tes. Dengan kata lain, bila diberikan item-item yang sulit maka skor yang diperoleh akan lebih rendah apabila diberikan item yang sulit. IRT, dalam hal ini menekankan bahwa variasi karakteristik item tidak akan mempengaruhi hasil estimasi atribut laten yang diukur. Dengan asumsi ini, IRT lebih mudah digunakan untuk membandingkan performa *examinee* dengan perangkat tes yang berbeda, dibandingkan dengan CTT.

Ketiga, IRT menggunakan asumsi bahwa parameter item bersifat invarian antara kelompok peserta tes yang berbeda dan antara pengujian yang satu dengan pengujian berikutnya. Asumsi ini menekankan bahwa karakteristik *examinee* tidak mempengaruhi estimasi parameter butir pada model IRT. Hasil estimasi parameter item pada model IRT relatif sama meskipun diujikan/diberikan kepada berbagai kelompok yang berbeda (misalnya etnis, gender, dan sebagainya).

Keempat, model pada IRT tidak didasarkan pada skor total seperti pada CTT. Karena CTT menggunakan skor total, tes yang diberikan hanya menghasilkan satu perkiraan keandalan (reliabilitas) dan satu perkiraan kesalahan pengukuran atau *standard error measurement* (SEM). Pada CTT, SEM ini dianggap tidak berubah untuk semua orang yang mengikuti tes. SEM terlihat dalam skor dianggap sama, terlepas dari tingkat pada konstruk. Kondisi ini akan menjadi masalah, terutama jika item tes tidak sesuai dengan tingkat kemampuan sekelompok orang tertentu. Misalnya, tes mengukur kemampuan matematika yang dipelajari oleh siswa di kelas VII. Tes diberikan kepada sekelompok siswa yang baru mulai belajar di kelas VII, dan kelompok lain yang sudah menyelesaikan VII. Hasil estimasi reliabilitas tes dari respons yang diberikan kepada siswa yang sudah mempelajari materi tes akan lebih baik dibandingkan dengan estimasi dengan menggunakan respons siswa yang belum belajar materi dalam tes. Oleh karena reliabilitas dalam CTT adalah tunggal, maka kesalahan pengukuran yang dihasilkan pada dua kelompok tersebut dianggap sama, yang pada kenyataannya tentu akan berbeda. Reliabilitas dan SEM pada CTT dianggap konstan dan tidak bergantung pada konstruk. Pada IRT, presisi akan berkurang ketika ada ketidaksesuaian antara karakteristik examinee dan tingkat kesulitan item. Dengan demikian, SEM di IRT dapat bervariasi sesuai dengan kemampuan *examinee* dan karakteristik item yang diberikan.

Kelima, IRT dan CTT memiliki perbedaan dalam bentuk kurva yang menunjukkan hubungan antara skor item dan skor konstruk. Daya beda CTT dimodelkan dalam kurva linear sederhana antara keduanya, sedangkan IRT memodelkan hubungan keduanya dalam bentuk kurva lengkung (tidak linear). Daya beda untuk suatu item dapat divisualisasikan dalam diagram pencar (*scatter plot*), dengan skor konstruk pada sumbu x - dan skor item pada sumbu y. Diskriminasi item positif yang kuat akan ditunjukkan dengan titik-titik untuk skor yang salah berkumpul di bagian bawah skala, dan titik-titik untuk skor yang benar berkumpul di bagian atas. Sebuah garis yang melewati titik-titik ini kemudian akan memiliki kemiringan positif. Karena daya beda pada CTT didasarkan pada korelasi, maka kurva yang ditampilkan selalu dalam bentuk garis lurus dengan kemiringan konstan (garis

berwarna biru pada Gambar 6.1). Sementara itu, IRT hubungan skor item dengan konstruk dimodelkan dengan kurva logistik dengan bentuk yang melengkung (titik-titik berwarna hijau pada Gambar 6.1). titik-titik hijau menunjukkan probabilitas



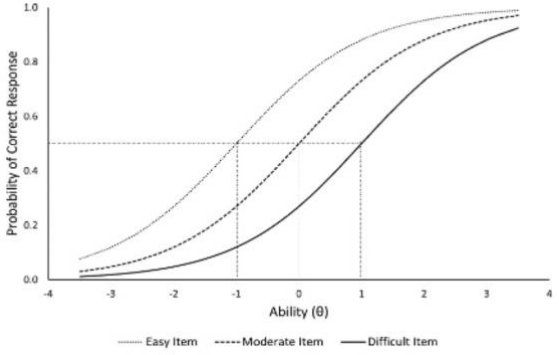
Gambar 6.1 *Scatter Plot* hubungan skor total (sumbu x) dan skor item (sumbu y).

Garis mewakili hubungan antara konstruk dan skor item untuk CTT (lurus) dan IRT (melengkung). Keterbatasan utama pada IRT adalah bahwa ini adalah model ini lebih kompleks dan membutuhkan sampel orang yang jauh lebih besar daripada yang dibutuhkan dalam CTT. Pada CTT minimal yang direkomendasikan adalah 100 peserta ujian untuk melakukan analisis butir soal (lihat Bab 3), sedangkan pada IRT diperlukan peserta yang lebih banyak 500 atau 1000 peserta ujian untuk mendapatkan hasil yang stabil, tergantung pada kompleksitas model yang dipilih (Paek et al., 2021).

6.2 Konsep Dasar IRT

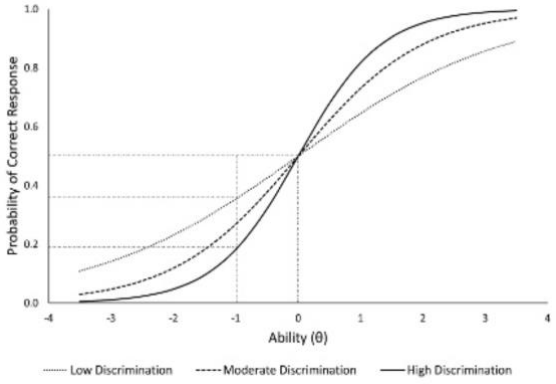
Karakteristik item pada IRT terdiri dari tingkat kesulitan (*difficulties*), daya pembeda (*discriminant*), tebakan (*guessing*). Tingkat kesulitan item adalah tingkat atribut laten minimum yang diperlukan untuk menjawab dengan benar pada item tersebut. Pada konteks pengukuran skala psikologis, istilah ‘menjawab benar’ berarti ‘memilih item’. Daya pembeda adalah kekuatan item dalam membedakan individu yang memiliki tingkat atribut laten yang rendah dengan individu yang memiliki tingkat atribut laten yang tinggi. Parameter tebakan adalah probabilitas menjawab benar oleh individu

yang memiliki tingkat atribut laten yang lebih rendah dari tingkat kesulitan item. Berdasarkan tiga parameter tersebut (kesulitan, daya pembeda, dan tebakan) dapat digambarkan kurva karakteristik item atau biasa dikenal dengan istilah item characteristic curve (ICC) yang menunjukkan probabilitas *examinee* menjawab benar (sumbu y) pada tingkat atribut laten (sumbu x).



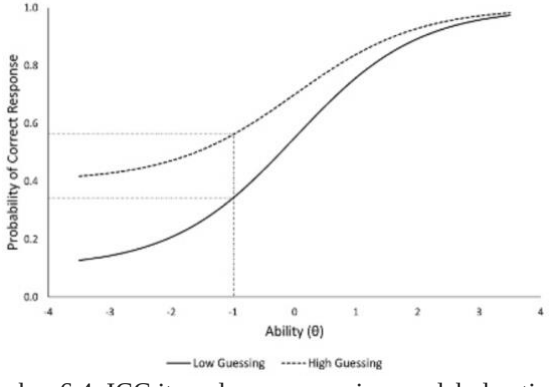
Gambar 6.2 ICC dari item dengan tingkat kesulitan rendah (*easy*), sedang (*moderate*) dan sulit (*difficult*)

Pada Gambar 6.2 diperlihatkan ICC dari dari item dengan daya beda yang sama, tetapi tingkat kesulitan yang berbeda. Untuk mencapai probabilitas menjawab benar sebesar 0,5, dibutuhkan tingkat atribut laten θ sebesar -1 (untuk item yang mudah), 0 (untuk item yang sedang), dan 1 (untuk item yang sulit). Nilai -1, 0, dan 1 menunjukkan tingkat kesulitan tiga item tersebut. Karena ICC adalah kurva probabilitas logistik, maka atribut laten juga berada pada skala logit yang nilainya dapat berada pada rentang $-\infty$ sampai ∞, tetapi umumnya mengumpul pada rentang -5 sampai $+5$, atau -4 sampai $+4$. Skala logit juga digunakan dalam menentukan tingkat kesulitan sebuah item.



Gambar 6.3. ICC dari item dengan daya pembeda rendah (*easy*), sedang (*moderate*) dan sulit (*difficult*)

Pada Gambar 6.3 ditampilkan ICC tiga item yang memiliki tingkat kesulitan dan tebakan (*guessing*) yang sama ($\theta = 0$), namun daya pembeda yang berbeda (rendah, sedang, dan tinggi). Daya pembeda menunjukkan kemiringan (*slope*) ICC pada titik yang menunjukan tingkat kesulitan item. ICC yang lebih curma menunjukkan daya pembeda item semakin baik dalam membedakan *examinee* berkemampuan tinggi dengan kemampuan rendah. Item dengan daya pembeda yang rendah lebih datar dibandingkan item dengan daya pembeda yang lebih tinggi.



Gambar 6.4. ICC item dengan guessing rendah dan tinggi

Pada Gambar 6.4 diperlihatkan ICC dari dua item yang memiliki tingkat tebakan (*guessing*) yang berbeda. Tingkat tebakan menentukan *intercept* (titik potong) kurva ICC pada sumbu y. Semakin tinggi tebakan (semakin tinggi lokasi titik potong dengan sumbu y), semakin tinggi probabilitas *examinee* berkemampuan rendah menjawab benar pada item tersebut. *Examinee* dengan tingkat atribut laten (dalam tes disebut 'kemampuan') $\theta = -1$, memiliki probabilitas hampir 60% untuk menjawab benar pada item yang memiliki tingkat tebakan yang tinggi (garis putus-putus). *Examinee* yang sama memiliki probabilitas sekitar 35% untuk menjawab benar pada item yang memiliki tingkat tebakan yang lebih rendah. Efek dari tebakan ini semakin kecil pada saat tingkat atribut laten (kemampuan) semakin tinggi.

Selain ICC, konsep penting lain dalam IRT adalah *îtem information function* (IIF). IIF menunjukan jumlah besarnya informasi untuk suatu item pada suatu interval tingkatan atribut laten. Semakin tinggi nilai IIF, maka semakin tinggi pula informasi yang tersedia untuk *examinee* dengan level atribut laten tertentu. IIF untuk item ke i ($i = 1,2,3, \dots, N$) pada satu level θ dinotasikan dengan $I_i(\theta)$. Karena item-item diasumsikan saling bebas (lokal independen), maka IIF dari sejumlah item dapat dijumlahkan untuk mendapatkan fungsi informasi test atau *test information function* (TIF) dengan persamaan berikut.

$$I(\theta) = \sum_{i=1}^{N} I_i(\theta), \qquad (6.1)$$

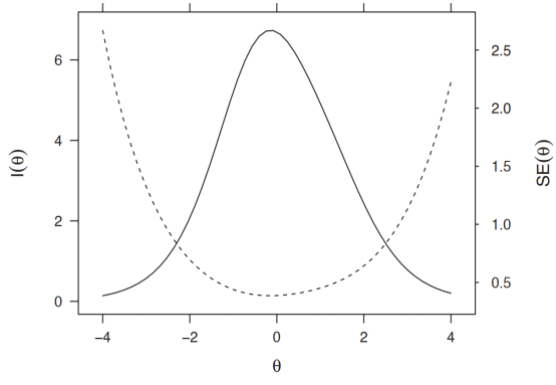
dimana $I(\theta)$ adalah total informasi tes untuk atribut laten θ. Dengan menggunakan TIF, dapat dihitung *conditional standard error of measurement* (cSEM), yang memberikan informasi tentang presisi dari seperangkat tes pada tingkat atribut laten (kemampuan) tertentu. Nilai cSEM dihitung dengan persamaan berikut.

$$cSEM(\theta) = \sqrt{\frac{1}{I(\theta)}}, \qquad (6.2)$$

TIF dan cSEM memiliki hubungan yang saling berkebalikan. Pada titik dimana TIF bernilai maksimum, cSEM bernilai minimum dan sebaliknya. Dengan kata lain, semakin tinggi TIF akan semakin rendah cSEM, sebaliknya semakin rendah TIF menunjukkan cSEM semakin tinggi. TIF yang tinggi menunjukkan presisi yang tinggi suatu tes dalam

mengestimasi level atau tingkat atribut laten yang diukur. Pada Gambar 6.5 terlihat bahwa tes akan menghasilkan estimasi dengan presisi yang paling baik jika diberikan kepada *examinee* yang memiliki θ disekitar 0. Dengan kata lain, sebuah tes akan menghasilkan cSEM yang berbeda antarindividu, bergantung pada level atau tingkat atribut laten/kemampuan (θ) yang dimiliki.

Misalnya, Andri, Malik, dan Sarah memiliki level/tingkat kemampuan yang berbeda. $\theta_{Andri} = -1{,}5$, $\theta_{Malik} = 0$, dan $\theta_{Sarah} = 3$. Berdasarkan TIF dan SEM pada Gambar 6.5, hasil estimasi atribut laten (kemampuan) yang dihasilkan oleh tes terhadap Andri memiliki cSEM yang lebih besar daripada estimasi terhadap kemampuan Malik. Meskipun demikian, estimasi kemampuan Andri dan Malik oleh tes menghasilkan TIF yang lebih tinggi dibandingkan dengan cSEM nya. Berbeda dengan estimasi kemampuan Sarah yang akan menghasilkan presisi yang kurang baik karena cSEM lebih besar nilainya daripada TIF.



Gambar 6.5. TIF (garis tidak terputus) dan cSEM (garis putus-putus) yang saling berkebalikan

6.3 Asumsi IRT

Hambleton dan Swaminathan (Hambleton & Swaminathan, 1985a) menyebutkan beberapa asumsi dalam teori respons butir. Beberapa asumsi yang paling utama yakni unidimensionalitas, independensi lokal, dan invariansi parameter (Desjardins & Bulut, 2018; Hambleton & Swaminathan, 1985a). Unidimensionalitas menunjukkan bahwa item atau tes hanya mengukur satu atribut laten

yang dominan. Metode untuk memeriksa pemenuhan asumsi unidimensionalitas antara lain *principal component analysis* (PCA), *eksploratory factor analysis* (EFA), *confirmatory factor analysis* (CFA), DIMTEST (Strout, 1990), dan DETECT (Zhang & Strout, 1999). Dari sejumlah metode tersebut, EFA dan CFA merupakan dua metode yang paling umum digunakan untuk menilai dimensionality (ulasan tentang EFA telah dibahas pada Chapter 4).

Independensi lokal mengasumsikan bahwa probabilitas untuk menjawab benar pada suatu item hanya dipengaruhi oleh atribut laten (kemampuan). Independensi lokal menunjukkan bahwa tidak ada hubungan antara item yang satu dengan item yang lain. Ketika asumsi unidimensionalitas telah terpenuhi, biasanya asumsi independensi lokal juga terpenuhi (Desjardins & Bulut, 2018; Hambleton & Swaminathan, 1985a). Salah satu uji yang paling umum digunakan dalam mendeteksi pemenuhan asumsi independensi lokal adalah Statistik Q3 Yen (1981).

Invariansi parameter, sebagaimana telah dijelaskan pada perbandingan CTT dan IRT, adalah asumsi bahwa parameter item tidak dipengaruhi oleh karakteristik peserta/*examinee*, dan sebaliknya karakteristik peserta/*examinee* tidak dipengaruhi oleh parameter item.

6.4 Model-Model IRT Unidimensional Item dengan Penskoran Dikotomi

Terdapat empat model IRT unidimensional untuk tes dengan penskoran dikotomi, yakni model 1-parameter logistik, 2-parameter logistik, 3-parameter logistik, dan 4-parameter logistik. Model-model tersebut memiliki perbedaan dalam hal asumsi terkait parameter yang digunakan yakni daya pembeda dan tebakan.

6.4.1 Model IRT 1-Parameter Logistik (1-PL)

Model paling sederhana pada IRT adalah model 1-parameter logistik (IRT 1PL). Secara matematis, IRT 1PL dapat dituliskan dengan persamaan berikut.

$$P(\theta_j, a, b_i) = \frac{exp\left(Da(\theta_j - b_i)\right)}{1 + exp\left(Da(\theta_j - b_i)\right)}, \qquad (6.3)$$

dimana θ_j adalah level atribut laten orang ke-j ($j = 1,2,3,\dots,J$), a adalah parameter daya pembeda item, b_i adalah parameter tingkat kesulitan pada item ke-i ($I = 1,2,3,\dots,I$), D adalah konstanta untuk menempatkan parameter pada model logistik pada model kurva normal ($D = 1{,}7$). Persamaan di atas menunjukkan bahwa probabilitas orang ke-j merespon/menjawab benar pada item ke-i ditentukan oleh tingkat atribut laten (kemampuan), daya pembeda, dan tingkat kesulitan item. Pada model IRT 1-PL, daya pembeda tidak memiliki indeks, yang menunjukkan bahwa besarnya daya pembeda untuk semua butir diasumsikan sama. Selain itu, pada model IRT 1-PL diasumsikan tidak ada efek tebakan yang mempengaruhi respons peserta.

Untuk mendemonstrasikan estimasi model IRT 1-PL, digunakan *dataset* "Data_5000x30-HD.csv" yang sudah disediakan penulis di Google Drive. Data ini adalah data simulasi yang dibuat dengan menggunakan WinGen (Han, 2007). Berikut ini adalah perintah untuk membaca data tersebut.

```
data <-
read.csv(sprintf("https://docs.google.com/uc?id=%s&export=down
load","1-LXQkC_-tSb_WBW2IdDr9NNFxJp1akJs"),header=T,sep=";")
data
```

RStudio: Notebook Output

b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23	b24	b25	b26	b27	b28	b29	b30
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	1	0	0	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	1	1	0
0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	1	0	0
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0

1–21 of 5,000 rows | 3–30 of 30 columns

Previous 1 2 3 4 5 6 … 48 Next

Gambar 6.6 Data_5000x30-HD.csv dari penyimpanan Google Drive

Karena data yang digunakan disimpan pada penyimpanan online, maka diperlukan koneksi internet untuk menggunakan data tersebut. Perintah di atas akan memunculkan data seperti pada Gambar 6.6. Data terdiri dari respons yang di skor dikotomi (0 1), dengan jumlah item (kolom) = 30, dan jumah respons (baris) = 5000. Jika format data

disimpan dengan model seperti pada Gambar 6.6, kita dapat menghitung jumlah item dan respons dengan menggunakan perintah ncol(data) dan nrow(data).

Estimasi parameter IRT 1-PL menggunakan packages ‘mirt’ (Chalmers, 2012b, 2021). Berikut ini sintaks yang digunakan untuk melakukan instalasi dan mengaktifkan library ‘mirt’ pada R. perintah “install.packages("mirt")” hanya digunakan saat pertama kali saja, setelahnya packages ini telah tersimpan sehingga dapat langsung mengaktifkan library.

```
install.packages(‘mirt’)
library(‘mirt’)
```

Sebelum melanjutkan pada penggunaan berbagai packages lainnya, kami ingin memperkenalkan kepada pembaca cara lain untuk mengaktifkan library dengan lebih efektif menggunakan Package Management Tool ‘pacman’. Packages ini yang berfungsi untuk menginstall dan mengaktifkan satu atau beberapa packages dalam satu perintah. Kita hanya perlu menginstall packages ini satu kali pada R, dan selanjutnya packages ini dipakai untuk instalasi packages yang diperlukan (jika packages belum terpasang di mesin R) sekaligus mengaktifkan library nya untuk melaksanakan proses analisis. Untuk memasang packages ‘pacman’ gunakan perintah install.packages.

```
Install.packages(‘pacman’)
```

Selanjutnya, untuk menggunakan packages ini kita cukup menuliskan packages apa saja yang akan digunakan dalam proses analisis. Sebagai contoh, apabila kita akan menggunakan packages mirt, writexl, ggplot2 dalam proses analisis, kita dapat meringkas perintah install.packages() dan library() dengan satu perintah berikut.

```
pacman::p_load(mirt,writexl,ggplot2)
```

Berikutnya, akan dilakukan estimasi parameter IRT dengan model 1-PL menggunakan packages ‘mirt’. Sebelum menggunakan fungsi mirt, terlebih dahulu didefinisikan model analisis yang akan digunakan. Perintah berikut perlu didefinisikan terlebih dahulu.

```
model1PL <- "F = 1 - 30
CONSTRAIN = (1 - 30, a1)"
```

Perintah pada baris pertama menunjukkan bahwa atribut laten tunggal yakni 'F' diukur oleh item 1 – 30 pada data. Selanjutnya, *constrain* pada baris kedua, digunakan untuk mendefinisikan bahwa daya pembeda pada semua item adalah sama. Selanjutnya, estimasi parameter dapat dilakukan dengan perintah.

```
fit_1PL <- mirt(data = data, model = model1PL, SE = TRUE)
```

Perintah di atas menghasilkan estimasi parameter dan disimpan dengan dengan nama 'fit_1PL'. Selanjutnya, untuk menyimpan dan menampilkan parameter butir dalam 1 data.frame digunakan fungsi coef. Parameter butir disimpan pada 'params_1PL' kemudian perintah pada baris kedua digunakan untuk menampilkan data dalam params_1PL yang telah disimpan.

```
params_1PL <- coef(fit_1PL, IRTpars = TRUE, simplify = TRUE)
params_1PL
```

Objek yang baru disimpan dengan nama 'params_1PL' berisi informasi paramater item hasil estimasi menggunakan model IRT 1-PL, rerata atribut laten (kemampuan), matriks varians-kovarian dari atribut laten, dan informasi tambahan berkaitan dengan proses estimasi. Kita dapat melihat struktur isi dari objek tersebut dengan perintah str(fit_1PL) namun karena informasi yang tersedia sangat banyak, kita tidak akan menampilkannya. Secara default, fungsi coef akan menghasilkan parameter d_i yakni tingkat kemudahan (esainess). Parameter ini ditransformasi menjadi parameter tingkat kesulitan b_i dengan perintah IRTpars = TRUE pada fungsi coef. Secara matematis hubungan d_i dan b_i diberikan pada persamaan berikut.

$$b_i = \frac{-d_i}{a1_i}, \tag{6.4}$$

dengan $a1_i$ adalah daya pembeda item. Selanjtunya, perintah simplify=TRUE (TRUE dapat juga diganti dengan T) digunakan untuk menyederhanakan output hasil estimasi parameter setiap butir dalam 1 dataframe. Jika tidak dituliskan, secara default perintah simplify=F (atau bisa diganti dengan False) dan hasil estimasi parameter butir ditampilkan dalam bentuk list yang panjang. Selanjutnya, karena fungsi coef menghasilkan sejumlah informasi lain selain parameter butir,

maka kita dapat menyimpan parameter butir yang sudah diestimasi dalam 1 dataframe menggunakan perintah berikut.

```
params_1PL <- params_1PL$items
head(params_1PL)
```

```
          a          b g u
b1 1.294743 -0.4269886 0 1
b2 1.294743  1.0082852 0 1
b3 1.294743  2.2272007 0 1
b4 1.294743  1.8630043 0 1
b5 1.294743  0.4871354 0 1
b6 1.294743  1.1113290 0 1
```

Kolom pertama pada output di atas adalah identitas item, yang bersesuaian dengan nama (head) setiap kolom pada data. Pada kolom kedua (a) menunjukkan data pembeda yang nilainya = 1,294743 untuk semua item, sebagaimana asumsi pada model IRT 1-PL. kolom ketiga (b) merupakan indeks tingkat kesulitan item yang nilainya menggunakan skala logit. Selanjutnya kolom keempat (g) menunjukkan guessing atau tebakan yang nilainya diasumsikan = 0 pada IRT 1-PL. Sedangkan kolom terakhir (u) menunjukkan asimtot atas kurva (pada model IRT 1-PL biasanya diabaikan karena asimtot kurva selalu berada pada dua titik, yakni 0 dan 1). Nilai u ini akan dideskripsikan pada penjelasan mengenai model IRT 3-PL dan IRT 4-PL. Output hasil estimasi parameter butir di atas dapat disimpan dalam bentuk file dengan menggunakan packages ‘writexl’ (Ooms, 2021). Prosedur instalasi dan mengaktifkan library ‘writexl’ dapat dilakukan dengan menambahkan pada perintah pacman::p_load() di atas. Untuk menyimpan hasil analisis dalam format xlsx digunakan perintah berikut.

```
write_xlsx(params_1PL,"/Users/hasandjidu/Documents/Buku-
R/Parameter1PL")
```

Fungsi write_xlsx() diisi dengan nama objek berupa data frame (params_1PL), kemudian diikuti lokasi penyimpanan (“Users/hasandjidu/….Buku-R/), dan bagian terakhir adalah nama file (Parameter1PL). perintah ini akan menyimpan params-1PL pada direktori yang ditentukan, dengan nama file “Parameter1PL”, dan dengan format *xlsx.

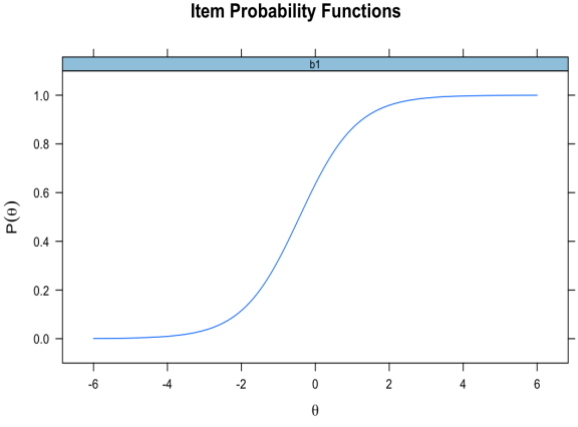
Selanjutnya, kita dapat pula melihat standar error dari estimasi masing-masing paramater item dengan menjalankan kembali fungsi coef() dengan menambahkan perintah printSE=T. Pada perintah di bawah ini, kita mengestimasi standar error semua item (baris pertama) kemudian pada baris kedua ditampilkan standar error untuk butir tertentu (butir 27)

```
se_1PL <- coef(fit_1PL, printSE = TRUE)
se_1PL$b27
```

```
            a1           d logit(g) logit(u)
par 1.29474279 -1.24595404     -999      999
SE  0.01619797  0.04042585       NA       NA
```

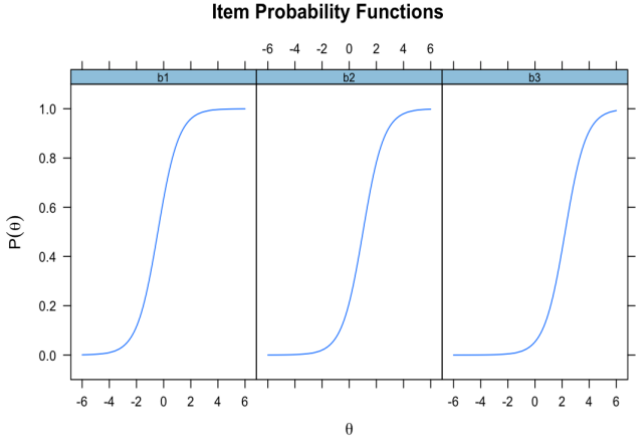
Selanjutnya, untuk menampilkan plot ICC dari item-item yang telah diestimasi parameternya, mirt menyediakan fungsi plot dan itemplot. Pada perintah berikut akan dihasilkan plot untuk item berdasarkan urutanya pada data baik secara terpisah, atau sekaligus. item = 1 pada baris pertama digunakan untuk menampilkan ICC untuk butir pertama (Gambar 6.7), sedangkan item = 1:3 pada baris kedua untuk menampilkan ICC untuk butir pertama sampai ketiga (Gambar 6.8).

```
plot(fit_1PL, type = "trace", which.items = 1)
```



Gambar 6.7. ICC item 1 model IRT 1-PL pada Data_5000x30-HD

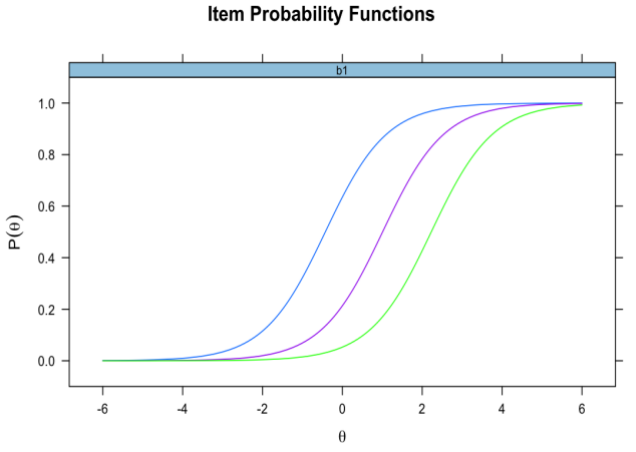
```
plot(fit_1PL, type = "trace", which.items = 1:3)
```



Gambar 6.8. ICC item 1,2, & 3 model IRT 1-PL pada Data_5000x30-HD

Karena model IRT 1-PL menghasilkan estimasi daya pembeda item yang sama (pada contoh ini =1.29474279), maka ICC dari 3 item pada Gambar 6.7 diperlihatkan dengan kurva yang memiliki kemiringan yang sama, namun berbeda pada titik dimana probabilitas = 0,5. Jika ketiga item ini ingin digambarkan dalam satu bidang seperti yang ilustrasi pada Gambar 6.2, maka kita dapat menggunakan packages 'latticeExtra' (Sarkar, 2008, 2021; Sarkar & Andrews, 2019) dengan mendefinisikan ICC masing-masing item terlebih dahulu.

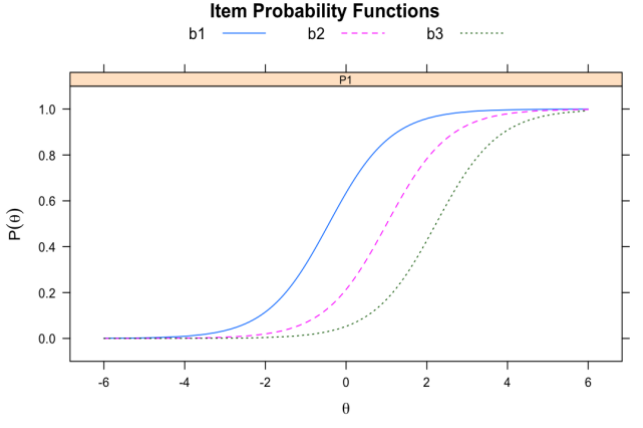
```
plot1 <- plot(fit_1PL, type = "trace", which.items = 1)
plot2 <- update(plot(fit_1PL, type = "trace", which.items =
2), col="purple")
plot3 <- update(plot(fit_1PL, type = "trace", which.items =
3), col="green")
plot1+plot2+plot3
```



Gambar 6.9. ICC item 1,2, & 3 model IRT 1-PL pada Data_5000x30-HD menggunakan packages fungsi plot dan packages 'latticeExtra'

Gabungan plot ICC dapat pula diperoleh dengan fungsi plot dengan mengatur facet_item=F sebagai berikut.

```
plot(fit_1PL, type = "trace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```

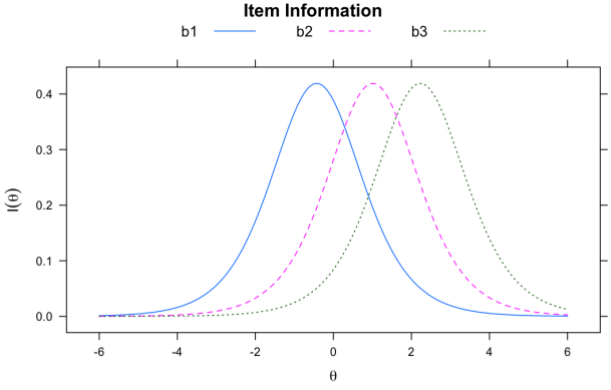


Gambar 6.10. ICC item 1,2, & 3 model IRT 1-PL pada Data_5000x30-HD menggunakan fungsi plot

Pada Gambar 6.9 dan Gambar 6.10 terlihat ICC untuk item ke-1 ditunjukkan dengan garis berwarna biru, ICC untuk item ke-2 ditunjukkan dengan garis berwarna ungu (purple), sedangkan ICC untuk item ke-3 ditunjukkan dengan garis berwarna hijau. Tiga ICC pada model IRT 1-PL ini terlihat memilliki bentuk yang sama. Perbedaan tingkat kesulitan ditunjukkan dengan posisi kurva, dimana posisi paling kanan menunjukkan item dengan tingkat kesulitan yang paling tinggi, sebaliknya posisi ICC paling kiri menunjukan tingkat kesulitan paling rendah.

Selanjutnya untuk mendapatkan kurva dari IIF dan TIF juga menggunakan fungsi plot dengan mengubah perintah pada type dari 'trace' menjadi 'infotrace'.

```
plot(fit_1PL, type = "infotrace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.11. Kurva IIF item 1, 2, & 3 model IRT 1-PL pada Data_5000x30-HD

Pada Gambar 6.11 terlihat tiga kurva IIF yang menunjukkan perbedaan lokasi θ dimana informasi item akan menghasilkan informasi yang optimum. Selanjutnya, untuk menampilkan kurva IIF yang dikombinasikan dengan SE dapat dilakukan dengan mengubah 'infotrace' menjadi 'infoSE' seperti yang ditampilkan pada Gambar 6.12. Perintah pada baris pertama di bawah ini hanya menampilkan IIF

dan SE untuk item 1, baris kedua menampilkan TIF gabungan item 1 sampai item 5, dan baris ketiga menampilkan TIF untuk seluruh item.

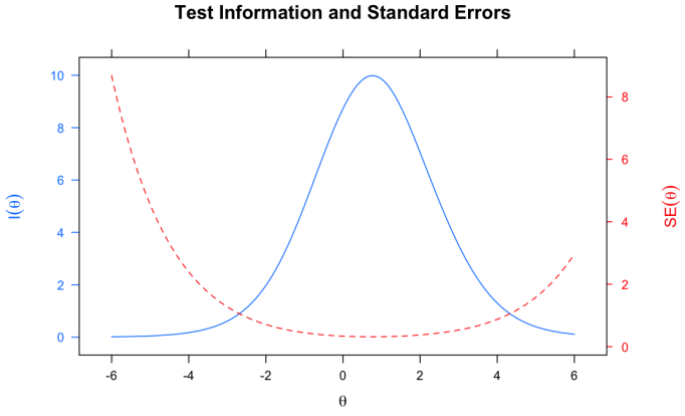
```
plot(fit_1PL, type = "infoSE", which.items = 1)
plot(fit_1PL, type = "infoSE", which.items = 1:5)
plot(fit_1PL, type = "infoSE")
```

Gambar 6.12. Kurva IIF dan SE untuk item 1 model IRT 1-PL pada Data_5000x30-HD

Gambar 6.13. Kurva TIF dan SE untuk gabungan item 1 sampai 5 model IRT 1-PL pada Data_5000x30-HD

Gambar 6.13 dan Gambar 6.14 memperlihatkan TIF yang merupakan gabungan dari IIF beberapa item. Gambar 6.13 adalah gabungan IIF untuk item 1 sampai item 5, sedangkan Gambar 6.15

menunjukan gabungan IIF untuk item 1 sampai 30. Gambar 6.12 sampai Gambar 6.14 ini memperlihatkan interval θ dimana item atau gabungan item akan menghasilkan informasi yang lebih tinggi dibandingkan *error* pengukuran. Sebagai contoh, pada Gambar 6.14 terlihat bahwa 30 item ini akan menghasilkan informasi yang lebih tinggi daripada error pengukuran jika item-item direspons oleh peserta yang memiliki θ dengan batas bawah mendekati -3 sampai batas atas mendekati +4.5.



Gambar 6.14. Kurva TIF dan SE untuk seluruh (30 item) model IRT 1-PL pada Data_5000x30-HD

6.4.2 Model Rasch

Model Rasch adalah kasus khusus pada model IRT-1PL dimana parameter daya pembeda item diatur = 1 untuk semua item. Konseksuensinya, persamaan yang menunjukkan probabilitas untuk merespons dengan benar pada item ke-i pada pemodelan IRT 1-PL dapat dituliskan menjadi model Rasch sebagai berikut.

$$P(Y_{ij} = 1|\theta_j, b_i) = \frac{\exp\left(D(\theta_j - b_i)\right)}{1 + \exp\left(D(\theta_j - b_i)\right)} \qquad (6.5)$$

dimana, semua komponen pada persamaan sama dengan model IRT 1-PL kecuali parameter a yang nilainya = 1 untuk semua item. Probabilitas untuk menjawab benar pada model Rasch hanya

dipengaruhi oleh tingkat kesulitan item. Oleh karena probabilitas hanya ditentukan oleh tingkat kesulitan item, maka apabila seorang peserta memiliki kemampuan yang tepat sama dengan tingkat kesulitan item, maka probabilitasnya menjawab benar adalah 50%.

Untuk mengilustrasikan estimasi parameter pada pemodelan Rasch, akan digunakan kembali data Data_5000x30-HD yang sudah disiapkan penulis pada GoogleDrive. Perintah pada R untuk estimasi parameter item dengan model Rasch lebih sederhana daripada yang telah dilakukan pada model IRT 1-PL, karena tidak perlu mendefinisikan *constrain* pada model analisis yang akan digunakan. Pada fungsi 'mirt' ditambahkan dengan perintah itemtype="Rasch" seperti berikut ini.

```
fit_Rasch <- mirt(data = data, model = 1, itemtype = "Rasch",
SE = TRUE)
```

Selanjutnya, seperti yang dilakukan pada model IRT 1-PL, digunakan digunakan fungsi coef untuk menampilkan hasil estimasi parameter butir dan disimpan pada 1 data.frame dengan nama 'params_Rasch.

```
params_Rasch <- coef(fit_Rasch, IRTpars = TRUE, simplify =
TRUE)
params_Rasch <- params_Rasch$items
head(params_Rasch)
```

```
   a          b g u
b1 1 -0.5548108 0 1
b2 1  1.3033465 0 1
b3 1  2.8813043 0 1
b4 1  2.4098243 0 1
b5 1  0.6286634 0 1
b6 1  1.4367491 0 1
```

Susunan output parameter butir yang dihasilkan sama dengan susun output parameter butir pada model IRT 1-PL. Perbedaannya terletak pada kolom daya pembeda (a) yang berisi nilai 1 untuk semua butir. Fungsi head() berfungsi untuk menampilkan enam baris pertama data.frame dari objek param_Rasch. Apabila ingin melihat atau membandingkan tingkat kesulitan item, maka perintah yang digunakan langsung menuliskan nama objeknya yakni param_Rasch. Dari data ini

bisa diinterpretasikan tingkat kesulitan butir dari yang paling sulit, paling mudah, dan seterusnya.

```
    a           b g u
b1  1 -0.55481075 0 1
b2  1  1.30334648 0 1
b3  1  2.88130430 0 1
b4  1  2.40982428 0 1
b5  1  0.62866342 0 1
b6  1  1.43674910 0 1
b7  1  0.89319986 0 1
b8  1  0.33263183 0 1
b9  1  3.54564305 0 1
b10 1  1.48212033 0 1
b11 1  1.10437751 0 1
b12 1  1.55228873 0 1
b13 1  1.05905505 0 1
b14 1  1.54944736 0 1
b15 1  1.00703734 0 1
b16 1 -0.90375946 0 1
b17 1 -0.54603016 0 1
b18 1  1.08839175 0 1
b19 1  1.38536299 0 1
b20 1  3.15773400 0 1
b21 1  0.95211477 0 1
b22 1 -0.50991435 0 1
b23 1  0.04172902 0 1
b24 1  0.07459290 0 1
b25 1 -0.74865804 0 1
b26 1  2.48292962 0 1
b27 1  1.24383609 0 1
b28 1 -0.15439116 0 1
b29 1  1.60250558 0 1
b30 1  0.96161522 0 1
```

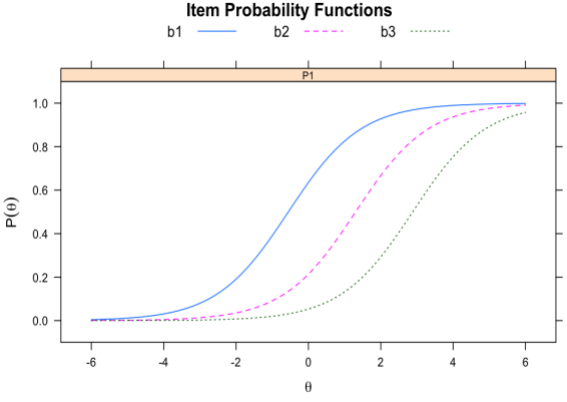
Output hasil estimasi parameter butir di atas dapat disimpan dalam bentuk file dengan menggunakan packages ‘writexl’ (Ooms, 2021) dengan perintah yang sama seperti yang sudah dilakukan pada model IRT 1-PL.

```
write_xlsx(params_Rasch,"/Users/hasandjidu/Documents/Buku-
R/ParameterRasch")
```

Selanjutnya, untuk menampilkan plot ICC dari item-item yang telah diestimasi parameternya, digunakan fungsi plot seperti yang sudah dijelaskan pada model IRT 1-PL. Berikut ini perintah untuk menampilkan gabungan plot ICC hasil estimasi dengan model Rasch untuk item 1 sampai 3. Butir 1, 2, dan 3 akan digunakan terus sebagai contoh pada model IRT 1-PL, Rasch, IRT 2-PL, IRT 3-PL, dan IRT 4-

PL agar terlihat perbandingan ICC antara model yang satu dengan lainnya.

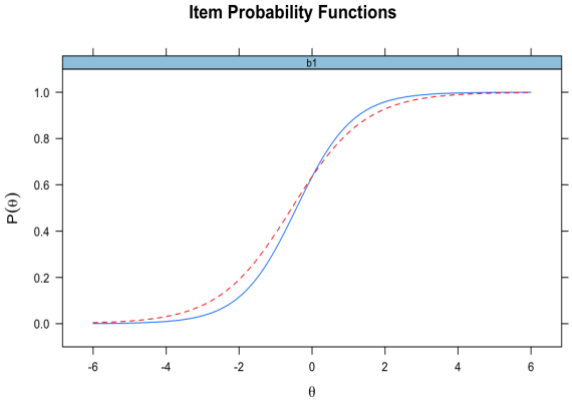
```
plot(fit_Rasch, type = "trace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.15. ICC item 1,2, & 3 model Rasch pada Data_5000x30-HD

Pada Gambar 6.15 terlihat ICC yang hampir sama dengan ICC yang dihasilkan oleh model IRT 1-PL pada butir 1, 2, 3. Akan tetapi, karena perbedaan slope ICC (daya pembeda) maka tingkat kesulitan item memiliki perbedaan. Pada Gambar 6.16 kami menggunakan packages 'latticeExtra' (Sarkar, 2008, 2021; Sarkar & Andrews, 2019) untuk membuat ICC item 1 hasil estimasi dengan model IRT 1-PL dan model Rasch. ICC tersebut memperlihatkan tingkat kesulitan item 1 hasil dari model IRT 1-PL sebesar -0.4269886, sedangkan hasil estimasi pemodelan Rasch diperoleh tingkat kesulitan sebesar -0. 55481075.

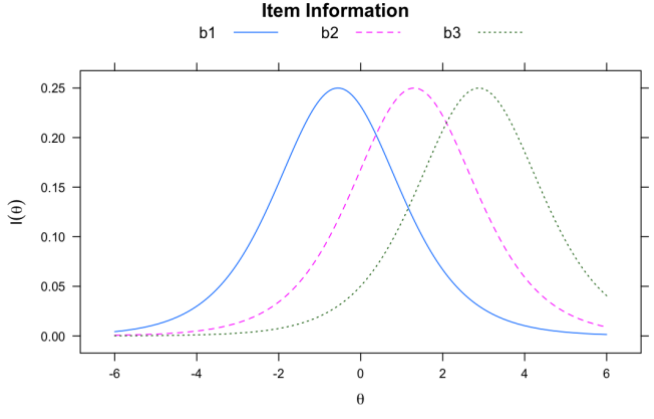
```
plot1 <- plot(fit_1PL, type = "trace", which.items=1)
plot2 <- update(plot(fit_Rasch, type = "trace",
         which.items = 1), col="red",
         par.settings = simpleTheme(lty = 2))
plot1+plot2
```



Gambar 6.16. ICC item 1 pada hasil estimasi model IRT 1-PL (garis biru) dan model Rasch (garis merah putus-putus) menggunakan Data_5000x30-HD

Selanjutnya, kurva IIF dan TIF digambar dengan menggunakan fungsi plot dengan mengubah perintah pada type dari 'trace' menjadi 'infotrace', dan akhir kurva gabungan IIF atau TIF dengan standar error diperoleh dengan mengunakan type= 'infoSE' pada fungsi plot.

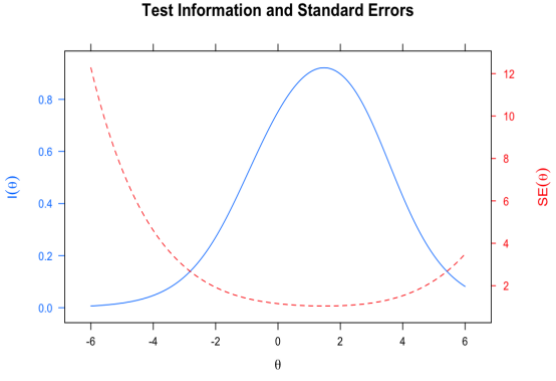
```
plot(fit_Rasch, type = "infotrace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.17. Kurva IIF item 1, 2, & 3 model Rasch pada Data_5000x30-HD

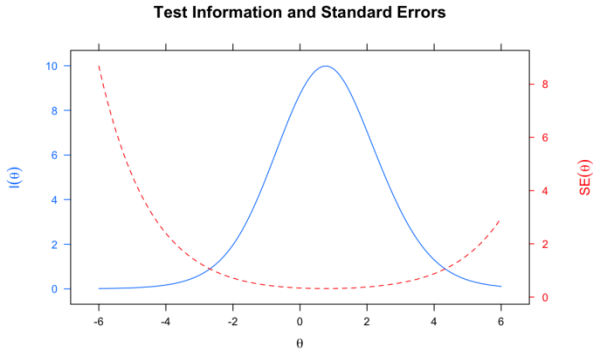
```
plot(fit_1PL, type = "infoSE", which.items = 1)
plot(fit_1PL, type = "infoSE", which.items = 1:5)
plot(fit_1PL, type = "infoSE")
```

Gambar 6.18. Kurva IIF dan SE untuk item 1 model Rasch pada Data_5000x30-HD



Gambar 6.19. Kurva TIF dan SE untuk gabungan item 1 sampai 5 model Rasch pada Data_5000x30-HD

Gambar 6.19 dan Gambar 6.20 memperlihatkan TIF yang merupakan gabungan dari IIF beberapa item. Gambar 6.19 adalah gabungan IIF untuk item 1 sampai item 5, sedangkan Gambar 6.20 menunjukan gabungan IIF untuk item 1 sampai 30. Gambar 6.19 sampai Gambar 6.20 ini memperlihatkan interval θ dimana item atau gabungan item akan menghasilkan informasi yang lebih tinggi dibandingkan error pengukuran.



Gambar 6.20. Kurva TIF dan SE untuk seluruh (30 item) model Rasch pada Data_5000x30-HD

6.4.3 Model IRT 2-Parameter Logistik (2-PL)

Model IRT 2-PL merupakan model yang lebih fleksibel dari model 1-PL dan Rasch. Pada model IRT 2-PL, parameter daya pembeda setiap item diestimasi berdasarkan data dan nilainya tidak seragam seperti pada model 1-PL dan Rasch. Secara matematis, IRT 2-PL dapat dituliskan dengan persamaan berikut.

$$P(Y_{ij} = 1|\theta_j, a_i, b_i) = \frac{\exp\left(Da_i(\theta_j - b_i)\right)}{1 + \exp\left(Da_i(\theta_j - b_i)\right)}, \qquad (6.6)$$

dimana θ_j, b_i, dan D adalah parameter yang sama dengan yang sudah dijelaskan pada model 1-PL dan Rassch. Pada model IRT 2-PL juga diasumsikan tidak ada efek tebakan sehingga parameter tebakan (guessing) hasil estimasi juga akan bernilai 0. Perbedaannya terletak pada parameter a yang kini memiliki indeks i yang artinya parameter a setiap butir tidak seragam.

Fungsi 'mirt' digunakan kembali pada Data_5000x30_3PL dengan mengubah perintah pada itemtype pada fungsi mirt menjadi "2PL". Hasil estimasi parameter butir menunjukkan nilai a yang bergam, sedangkan parameter g dan u masih bernilai sama dengan hasil estimasi pada model 1-PL dan Rasch.

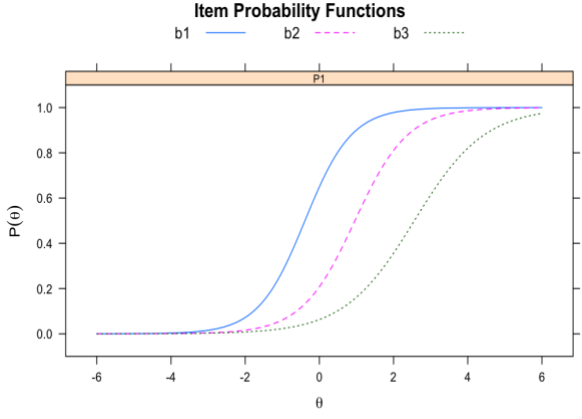
```
fit_2PL <- mirt(data = data, model = 1, itemtype="2PL", SE =
TRUE)
params_2PL <- coef(fit_2PL, IRTpars = T, simplify = T)
```

```
params_2PL <- params_2PL$items
head(params_2PL)
```

```
          a          b g u
b1 1.586838 -0.3890980 0 1
b2 1.396524  0.9577856 0 1
b3 1.054864  2.5600109 0 1
b4 1.172460  1.9854385 0 1
b5 1.112596  0.5299122 0 1
b6 1.199793  1.1600539 0 1
```

Plot ICC yang dihasilkan dari model 2-PL memiliki perbedaan antara item karen nilai slope yang beragam. Gambar 6.21 terlihat bahwa item 1, 2, dan 3 memiliki bentuk plot yang berbeda karena perbedaan slope ketiganya. Item 3 memiliki daya beda paling rendah diantara ketiganya. Terlihat dari slope ICC item 3 (ICC titik-titik hijau).

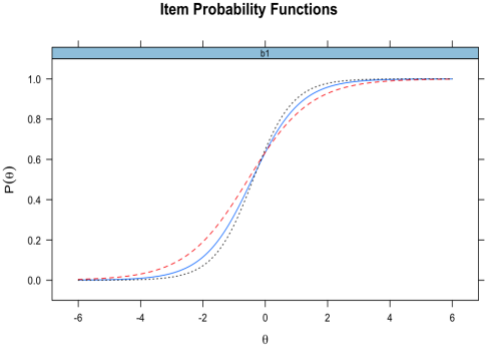
```
plot(fit_2PL, type = "trace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.21 ICC item 1, 2, & 3 model IRT 2-PL pada Data_5000x30_3PL

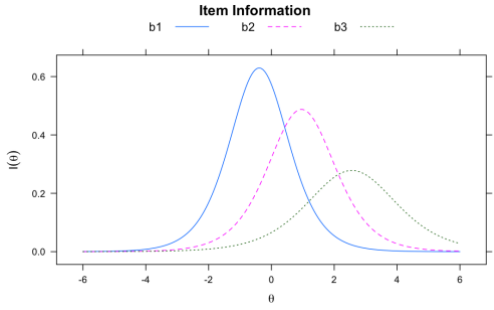
Perbedaan ICC hasil estimasi model IRT 1-PL, model Rasch, dan Model 2-PL bisa ditampilkan dengan menggabungkan hasil estimasinya menggunakan packages ‘latticeExtra’ (Sarkar, 2008, 2021; Sarkar & Andrews, 2019). Perintah yang digunakan sebagai berikut. Pada Gambar 6.22 terlihat 3 ICC item 1 hasil estimasi dengan model IRT 1-PL, model Rasch, dan model IRT 2-PL.

```
plot1 <- plot(fit_1PL, type = "trace",which.items = 1)
plot2 <- update(plot(fit_Rasch, type = "trace",
         which.items = 1), col="red",
         par.settings = simpleTheme(lty = 2))
plot3 <- update(plot(fit_2PL, type = "trace",
         which.items = 1), col="black",
         par.settings = simpleTheme(lty = 3))
plot1+plot2+plot3
```

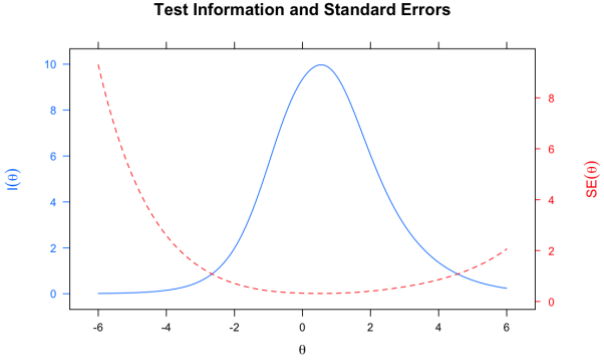


Gambar 6.22. ICC item 1 pada hasil estimasi model IRT 1-PL (garis biru), model Rasch (garis merah putus-putus), dan IRT 2-PL (garis biru) menggunakan Data_5000x30-HD.

Kurva IIF (Gamabr 6.23) dan TIF (Gambar 6.24) dihasilkan dengan perintah yang sama (plot) dengan mengubah data yang diolah oleh fungsi plot dengan fit_2PL. Gambar 6.23 memperlihatkan bahwa masuknya parameter a (daya pembeda) dalam estimasi yang dilakukan pada model IRT 2-PL mempengaruhi bentuk dan ukuran kurva IIF. Item 1, 2, dan 3 tidak hanya berbeda dalam hal interval dimana presisi pengukuran dapat tercapai, tetapi juga berbeda dalam hal besarnya informasi optimum yang dapat dihasilkan. Item 1 yang memiliki daya pembeda paling diantara ketiganya menghasilkan IIF paling tinggi, sedangkan item 3 dengan daya pembeda paling rendah diantara ketiganya menghasilkan nilai IIF yang paling rendah. Berbeda dengan Gambar 6.11 dan Gambar 6.17 (hasil estimasi dengan model IRT 1-PL dan Rasch) menunjukkan nilai IIF yang sama pada semua item.



Gambar 6.23. Kurva IIF item 1, 2, & 3 model IRT 2-PL pada Data_5000x30-HD



Gambar 6.24. Kurva TIF dan SE untuk seluruh (30 item) model IRT 2-PL pada Data_5000x30-HD

6.4.4 Model IRT 3-Parameter Logistik (3-PL)

Model IRT 3-PL adalah perluasan dari model IRT 2-PL yang juga memasukkan parameter a (daya pembeda) dan b (tingkat kesulitan item) kemudian ditambah dengan parameter c (tebakan/ *pseudo guessing*) pada model estimasinya. Secara matematis, IRT 3-PL dapat dituliskan dengan persamaan berikut.

$$P(\theta_j, a_i, b_i, c_i) = c_i + \frac{1 - c_i}{1 + exp\left(-Da_i(\theta_j - b_i)\right)}, \qquad (6.7)$$

dimana θ_j, a_i, b_i, dan D adalah parameter yang sama dengan yang sudah dijelaskan pada model 2-PL. Pada model IRT 3-PL diasumsikan bahwa probabilitas menjawab juga dipengaruhi oleh tebakan (*guessing*) sehingga nilai g yang dihasilkan pada estimasi menggunakan fungsi

‘mirt’ tidak bernilai 0. Nilai g menunjukkan parameter c pada model IRT 3-PL.

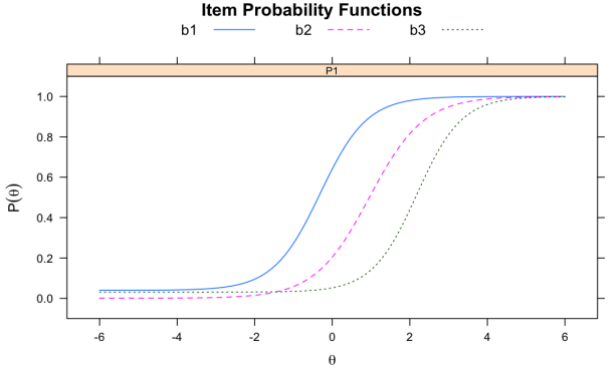
Data_5000x30-HD kembali digunakan untuk mengestimasi parameter butir menggunakan mirt, dengan memberikan perintah itemtype="3PL". Hasil estimasi parameter butir menunjukkan nilai a, b, dan c yang beragam, sedangkan nilai u =1 (masih bernilai sama dengan hasil estimasi pada model 2-PL).

```
fit_3PL <- mirt(data = data, model = 1, itemtype="3PL", SE = TRUE)
params_3PL <- coef(fit_3PL, IRTpars = T, simplify = T)
params_3PL <- params_3PL$items
head(params_3PL)
```

```
           a          b            g u
b1 1.655939 -0.3107806 0.0400178001 1
b2 1.425308  0.9567585 0.0009333005 1
b3 1.733362  2.1819660 0.0309135685 1
b4 1.477074  1.8785522 0.0225353954 1
b5 1.198692  0.5955865 0.0289111074 1
b6 1.377034  1.1816578 0.0275582839 1
```

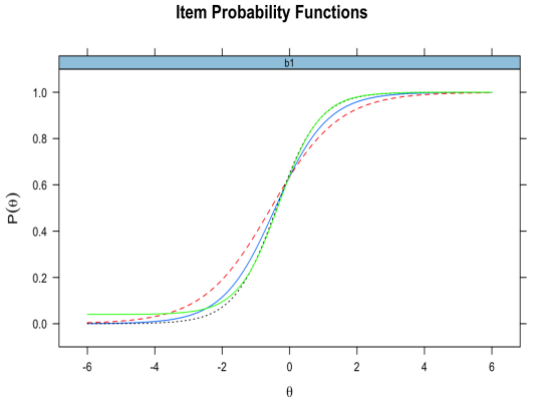
Plot ICC yang dihasilkan dari mode IRT 3-PL memiliki perbedaan antara item karena nilai *slope* (daya pembeda) dan *intercept* (tebakan semu) yang beragam. Gambar 6.25 memperlihatkan item 1 (garis wana biru) memiliki tingkat tebakan semu yang paling tinggi diantara ketiganya.

```
plot(fit_3PL, type = "trace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.25 ICC item 1, 2, & 3 model IRT 3-PL pada Data_5000x30-HD

```
Perbedaan ICC hasil estimasi model IRT 1-PL, model Rasch,
Model 2-PL, dan Model 3-PL kembali ditampilkan dengan
menggabungkan hasil estimasi menggunakan packages
'latticeExtra' (Sarkar, 2008, 2021; Sarkar & Andrews, 2019).
Perintah yang digunakan sebagai berikut.
plot1 <- plot(fit_1PL, type = "trace", which.items=1)
plot2 <- update(plot(fit_Rasch, type = "trace",
         which.items = 1), col="red", par.settings =
         simpleTheme(lty = 2))
plot3 <- update(plot(fit_2PL, type = "trace",
         which.items = 1), col="black", par.settings =
         simpleTheme(lty = 3))
plot4 <- update(plot(fit_3PL, type = "trace",
         which.items = 1), col="green", par.settings =
         simpleTheme(lty = 1))

plot1+plot2+plot3+plot4
```



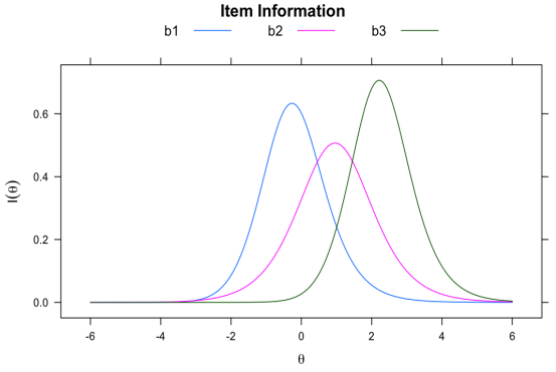
Gambar 6.26. ICC item 1 hasil estimasi model IRT 1-PL (garis biru) dan model Rasch (garis merah putus-putus), model 2-PL (garis hitam putus-putus), dan model 3-PL (garis hijau) menggunakan Data_5000x30-HD

ICC untuk item 1 pada Gambar 6.26 memperlihatkan perbedaan ICC antar model estimasi yang digunakan. Terlihat titip potong ICC terhadap sumbu probabilitas $\neq 0$. Semakin besar nilai c, maka semakin tinggi juga titik pada sumbu probabilitas yang dilewati oleh kurva ICC item1.

Kurva IIF dan TIF dihasilkan dengan perintah yang sama (plot) dengan mengubah data yang diolah oleh fungsi plot dengan fit_3PL. Gambar 6.27 terlihat bahwa masuknya parameter c (tebakan semu) dalam estimasi yang dilakukan pada model IRT 3-PL mempengaruhi

bentuk dan ukuran kurva IIF. Item 1, 2, dan 3 tidak hanya berbeda dalam hal interval dimana presisi pengukuran dapat tercapai, tetapi juga berbeda dalam hal besarnya informasi optimum yang dapat dihasilkan.

```
plot(fit_3PL, type = "infotrace", which.items = 1:3,
facet_items = FALSE, auto.key = list(points = F, lines = T,
columns = 3),
par.settings = simpleTheme(lty = 1:3))
```

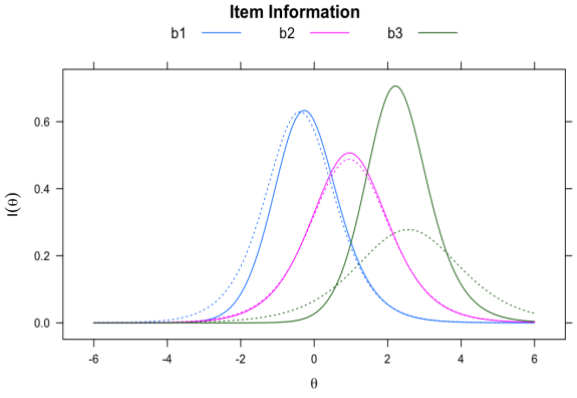


Gambar 6.27. Kurva IIF item 1, 2, & 3 model IRT 3-PL pada Data_5000x30-HD

Pada model IRT 2-PL item 1 menghasilkan nilai IIF yang paling optimum dan minimum pada item 3. Masuknya parameter *c* pada Gambar 6.27 justru mengakibatkan item 3 menghasilkan nilai IIF yang paling optimum. Hal ini mengindikasikan bahwa besarnya parameter c (tebakan semu) pada item 3 lebih kecil dibandingkan tebakan semu pada item 1. Pada Gambar 6.28 diperlihatkan Gabungan kurva IIF untuk item 1, 2, dan 3 hasil estimasi pada model IRT 2-PL (garis biru, ungu, dan hijau) dan model IRT 3-PL (garis putus-putus berwarna biru, ungu, dan hijau) yang dibuat dengan perintah sebagai berikut.

```
plot1 <- plot(fit_3PL, type = "infotrace", which.items = 1:3,
facet_items = FALSE, auto.key = list(points = F, lines = T,
columns = 3),
par.settings = simpleTheme(lty = 1))
plot2 <- update(plot(fit_2PL, type = "infotrace", which.items
= 1:3,
facet_items = FALSE, auto.key = list(points = F, lines = T,
columns = 3),
par.settings = simpleTheme(lty = 3)))
```

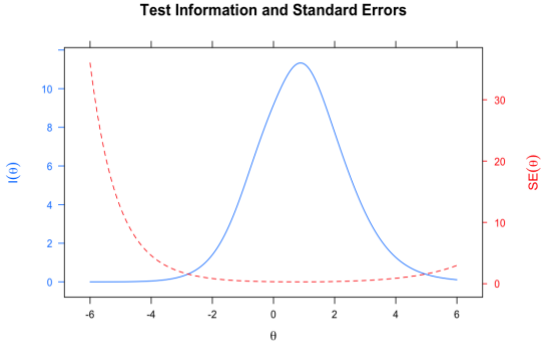
```
plot1+plot2
```



Gambar 6.28. Kurva IIF item 1, 2, & 3 hasil estimasi model IRT 3-PL (garis tidak terputus) dan model IRT 2-PL (garis putus-putus) pada Data_5000x30-HD

Gambar 6.28 menunjukkan perubahan kurva IIF hasil estimasi model IRT 2-PL dan model 3-PL. Selanjutnya, sama seperti model 1-PL, Rasch, dan model 2-PL, kurva TIF juga dapat ditampilkan secara bersama dengan kurva standar error pengukuran. Menggunakan perintah infoSE pada fungsi plot diperoleh kurva TIF seperti ditampilkan pada Gambar 6.28.

```
plot(fit_3PL, type = "infoSE")
```



Gambar 6.29. Kurva TIF dan SE untuk seluruh (30 item) model IRT 3-PL pada Data_5000x30-HD.

6.4.5 Model IRT 4-Parameter Logistik (4-PL)

Model IRT 4-PL diperkenalkan pada tahun 1981 sebagai kasus khusus untuk model IRT 3-PL. Model IRT 4-PL memungkinkan item-item memiliki dua asimtot, yakni asimtot bawah dan asimtot atas. Model IRT 4-PL memasukkan parameter a (daya pembeda) dan b (tingkat kesulitan item) dan c (tebakan semu), kemudian ditambah dengan parameter u, yakni parameter batas yang mencegah probabilitas merespons benar pada item mendekati 1 atau 100%. Secara mateamatis, model IRT 4-PL dituliskan sebagai berikut.

$$P(\theta_j, a_i, b_i, c_i, u_i) = c_i + \frac{u_i - c_i}{1 + exp\left(-D a_i(\theta_j - b_i)\right)}, \qquad (6.8)$$

dimana θ_j, a_i, b_i, c_i dan D adalah parameter yang sama dengan yang sudah dijelaskan pada model 2-PL. Model IRT 3-PL adalah model yang paling populer dalam mengestimasi atribut laten berupa prestasi. Sementara itu, model IRT 4-PL menurut Desjardins dan Bulut (Desjardins & Bulut, 2018) cocok digunakan untuk mengestimasi atribut-atribut non kognitif (misalnya: motivasi, minat, dan lainnya) karena probabilitas untuk mencapai level tertinggi pada atribut laten non kognitif tersebut tidak mencapai 1. Desjardins dan Bulut (Desjardins & Bulut, 2018) menyebutkan beberapa studi yang menjelaskan keuntungan menggunakan asimtot atas (u) ini dalam model IRT seperti Loken dan Rulison (2010), Osgood, et al. (2002), Reise dan Waller (2009), dan Tavares, et al. (2004)

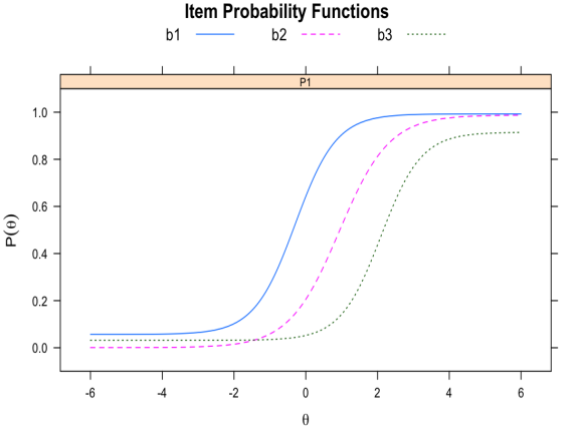
Seperti sebelumnya Data_5000x30-HD diestimasi menggunakan mirt, dengan memberikan perintah itemtype="4PL". Hasil estimasi parameter butir menunjukkan nilai a, b, c, dan u yang beragam.

```
params_4PL <- coef(fit_4PL, IRTpars = T, simplify = T)
params_4PL <- params_4PL$items
head(params_4PL)
```

```
          a          b           g         u
b1 1.745361 -0.2940075 0.0567453424 0.9927720
b2 1.436292  0.9331810 0.0006618066 0.9878258
b3 1.774842  2.0943478 0.0314242982 0.9149132
b4 1.450554  1.8816790 0.0212637418 0.9935002
b5 1.212050  0.5733401 0.0289469757 0.9907058
b6 1.369803  1.1707871 0.0262331643 0.9949473
```

Plot ICC yang dihasilkan dari mode IRT 3-PL memiliki perbedaan antara item karena nilai *slope* (daya pembeda) dan *intercept* (tebakan semu) yang beragam. Gambar 6.30 memperlihatkan item 3 (garis hijau putus-putus) memiliki tingkat asimtot atas paling rendah diantara ketiga item.

```
plot(fit_4PL, type = "trace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```

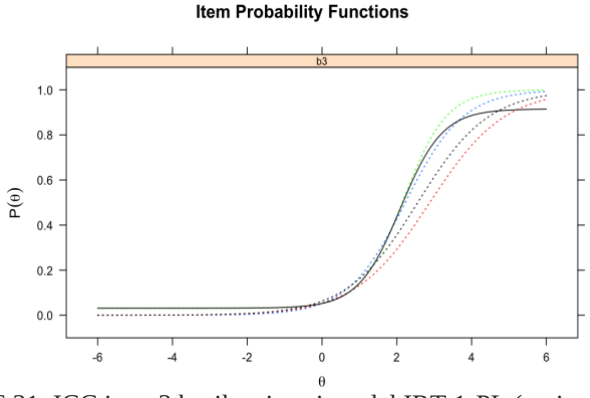


Gambar 6.30 ICC item 1, 2, & 3 model IRT 3-PL pada Data_5000x30-HD

Seperti sebelumnya, kita dapat melihat perbedaan ICC hasil estimasi model IRT 1-PL, model Rasch, Model 2-PL, Model 3-PL, dan Model 4-PL dengan menggabungkan ICC hasil estimasi menggunakan packages 'latticeExtra' (Sarkar, 2008, 2021; Sarkar & Andrews, 2019). Perintah yang digunakan sebagai berikut.

```
plot1 <- plot(fit_1PL, type = "trace", which.items = 3,
              par.settings = simpleTheme(lty = 3), col="blue")
plot2 <- update(plot(fit_Rasch, type = "trace", which.items
=3), col="red",
                par.settings = simpleTheme(lty = 3))
plot3 <- update(plot(fit_2PL, type = "trace", which.items =
3), col="black",
                par.settings = simpleTheme(lty = 3))
plot4 <- update(plot(fit_3PL, type = "trace", which.items =
3), col="green",
```

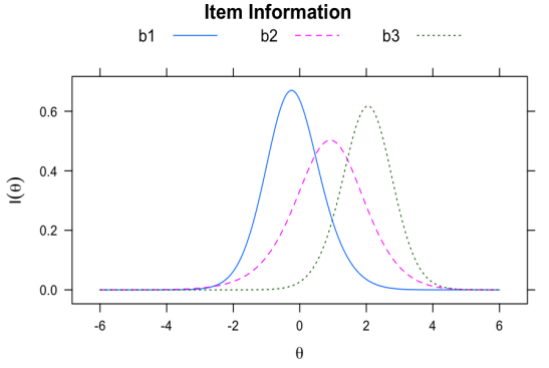
```
                par.settings = simpleTheme(lty = 3))
plot5 <- update(plot(fit_4PL, type = "trace", which.items =
3), col="black",
                par.settings = simpleTheme(lty = 1))
plot1+plot2+plot3+plot4+plot5
```



Gambar 6.31. ICC item 3 hasil estimasi model IRT 1-PL (garis putus-putus biru) dan model Rasch (garis putus-putus merah), model 2-PL (garis putus-putus hitam), model 3-PL (garis putus-putus hijau), dan model 4-PL (garis hitam) menggunakan Data_5000x30-HD

ICC untuk item 3 pada yang dihasilkan Gambar 6.31 merupakan hasil estimasi dengan lima model yang berbeda, yakni IRT 1-PL, Rasch, IRT 2-PL, IRT 3-PL, dan IRT 4-PL. Selanjutnya, kurva IIF dan TIF dihasilkan dengan perintah yang sama (plot) dengan mengubah data yang diolah oleh fungsi plot dengan fit_4PL. Gambar 6.32 menunjukan bahwa masuknya parameter u (asimtot atas) dalam estimasi yang dilakukan pada model IRT 4-PL mempengaruhi bentuk dan ukuran kurva IIF. Seperti yang ditunjukkan pada model 2-PL dan 3-PL, item 1, 2, dan 3 tidak hanya berbeda dalam hal interval dimana presisi pengukuran dapat tercapai, tetapi juga berbeda dalam hal besarnya informasi optimum yang dapat dihasilkan.

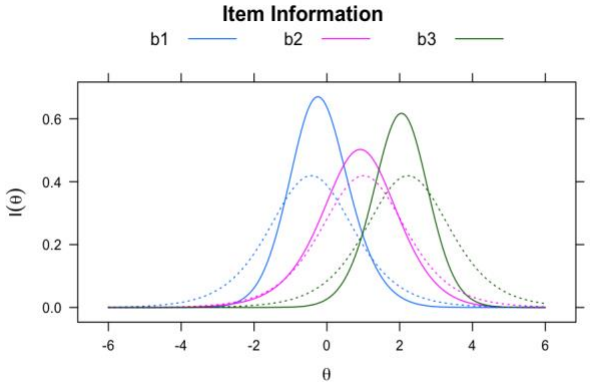
```
plot(fit_4PL, type = "infotrace", which.items = 1:3,
     facet_items = FALSE, auto.key = list(points = F,
     lines = T, columns = 3),
par.settings = simpleTheme(lty = 1:3))
```



Gambar 6.32. Kurva IIF item 1, 2, & 3 model IRT 4-PL pada Data_5000x30-HD

Pada sintaks berikut kita akan membandingkan kurva IIF dari item 1, 2, dan 3 berdasarkan hasil estimasi model IRT 1-PL dengan model IRT 4-PL. Perbedaan yang sangat terlihat dari kurva IIF adalah besarnya nilai optimum informasi yang dihasilkan antara model IRT 1-PL dengan model IRT 4-PL. Kendati demikian, interval θ yang bersesuaian dengan nilai IIF yang tinggi tidak berubah signifikan.

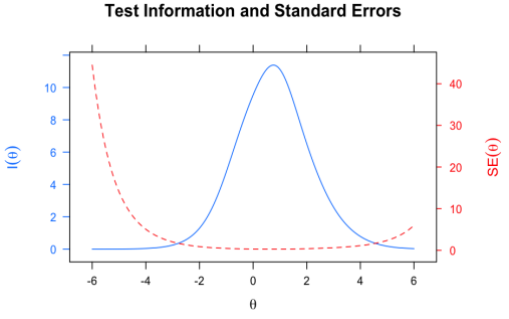
```
plot1 <- plot(fit_4PL, type = "infotrace", which.items = 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 1))

plot2 <- update(plot(fit_1PL, type = "infotrace", which.items
= 1:3,
facet_items = FALSE,
auto.key = list(points = F, lines = T, columns = 3),
par.settings = simpleTheme(lty = 3)))

plot1+plot2
```



Gambar 6.33. Kurva IIF item 1, 2, & 3 hasil estimasi model IRT 4-PL (garis tidak terputus) dan model IRT 1-PL (garis putus-putus) pada Data_5000x30-HD

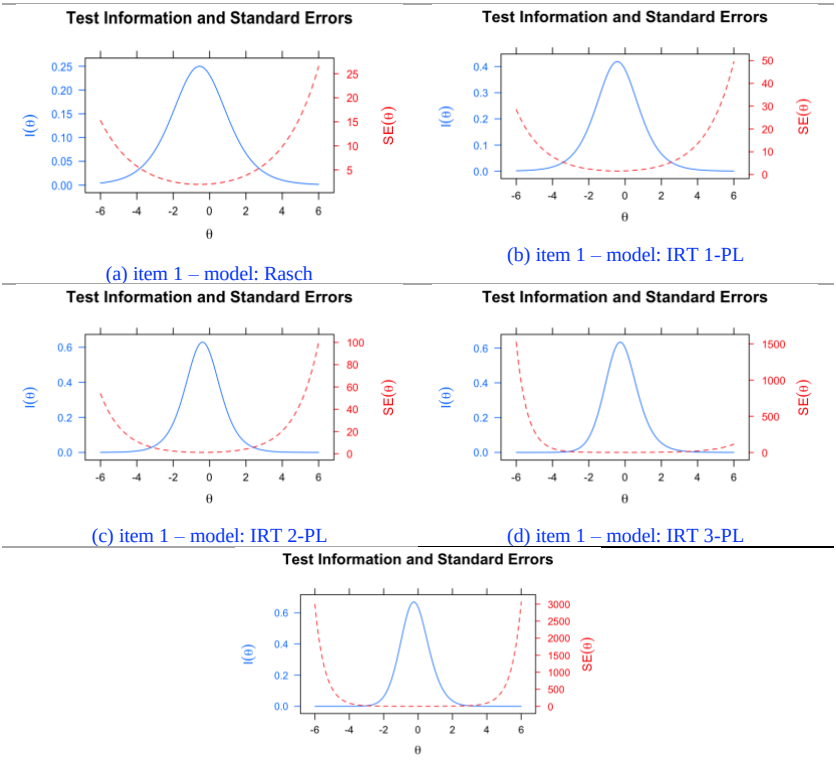
Gambar 6.34 menunjukkan kurva TIF hasil estimasi model IRT 3-PL. Hal menarik yang dapat dilihat dari kurva TIF dan IIF pada 5 pemodelan yang sudah ditampilkan adalah interval nilai SE yang justru semakin besar pada model yang menggunakan parameter yang lebih banyak. SE yang besar terutama ditunjukkan pada interval θ dimana informasi lebih rendah dari error.

```
plot(fit_3PL, type = "infoSE")
```



Gambar 6.34. Kurva TIF dan SE untuk seluruh (30 item) model IRT 3-PL pada Data_5000x30-HD

```
plot(fit_1PL, type = "infoSE",which.items=1)
plot(fit_Rasch, type = "infoSE",which.items=1)
plot(fit_2PL, type = "infoSE",which.items=1)
plot(fit_3PL, type = "infoSE", which.items=1)
plot(fit_4PL, type = "infoSE", which.items=1)
```



(a) item 1 – model: Rasch

(b) item 1 – model: IRT 1-PL

(c) item 1 – model: IRT 2-PL

(d) item 1 – model: IRT 3-PL

(e) item 1 – model:IRT 4-PL

Gambar 6.35. Kurva IIF dan SE untuk item 1 dengan lima jenis pemodelan pada Data_5000x30-HD

Gambar 6.35 menunjukkan bahwa semakin banyak parameter yang dimasukkan dalam model, semakin tinggi informasi yang dihasilkan dan semakin tinggi pula besarnya error pengukuran terutama pada interval θ di mana informasi lebih kecil dari SE. Sebagai contoh, nilai IIF hasil pemodelan Rasch (Gambar 6.35(a)) memiliki nilai IIF optimum yang kurang dari 0,5 dan error pada interval θ dari -6 sampai +6 adalah 25. Nilai IIF yang lebih tinggi ditunjukkan pada hasil pemodelan dengan jumlah parameter lebih banyak (Gambar 6.35(c),(d), dan (e)) dimana nilai IIF mencapai nilai optimum lebih dari 0.6, namun nilai SE pada interval -6 sampai +6 mencapai 3000 (model IRT 4-PL).

6.5 Estimasi θ Model IRT Unidimensi pada Penskoran Dikotomi

Setelah mengestimasi parameter item dengan model IRT tertentu, langkah selanjutnya adalah menggunakan parameter item untuk mengestimasi θ atau tingkat atribut laten/kemampuan peserta. Pada konteks pelaksanaan tes yang mengukur kemampuan kognitif, tingkat atribut laten θ yang diperoleh menunjukkan kemampuan. Untuk membahas bagian ini, kita akan menggunakan terminologi 'kemampuan' untuk menggambarkan level/tingkat atribut laten θ.

Dari respons peserta terhadap N item, kemampuan diestimasi dengan menghitung probabilitas peserta ke-j untuk menjawab benar $P(\theta_j, a_i, b_i, c_i)$, dan probabilitas menjawab salah yakni $Q = 1 - P(\theta_j, a_i, b_i, c_i)$ pada tiap item yang telah diestimasi parameternya (pada contoh ini menggunakan model IRT 3-PL). Selanjutnya digunakan fungsi Likelihood dengan mengalikan P atau Q pada N item yang direspons oleh peserta dengan persamaan Likelihood berikut ini.

$$L(\theta_j) = \prod_{i=1}^{N} P_i(\theta_j, a_i, b_i, c_i)^{x_i} \times Q_i(\theta_j, a_i, b_i, c_i)^{1-x_i}, \qquad (6.9)$$

dimana x_i adalah skor peserta ke-j pada item ke-i ($x_i = 1$ jika benar; $x_i = 0$ jika salah). Fungsi di atas adalah salah satu contoh fungsi Likelihood untuk satu pola respons tertentu. Desjardins dan Bulut (Desjardins & Bulut, 2018) menyebutkan 3 metode yang umum digunakan dalam estimasi atribut laten atau kemampuan.

- Maximum Likelihood Estimation (MLE): metode ini mencari θ yang menghasilkan nilai Likelihood paling optimal.
- Maximum a Posteriori (MAP) atau biasa dikenal dengan Bayesian modal: pada metode ini, fungsi Likelihood dikalikan dengan fungsi yang merepresentasikan distribusi populasi. MAP menghitung modus distribusi a posterior ini sebagai hasil akhir yang menggambarkan θ.
- Expected a Posteriori (EAP): merupakan salah satu jenis pendekatan MAP yang menggunakan mean dari distribusi a posterior untuk mendapatkan nilai θ.

Salah satu keunggulan dari metode Bayesian adalah kemampuannya dalam mengestimasi kemampuan pada pola respons yang benar semua atau salah semua. Pada metode MLE, kondisi tersebut tidak dapat ditangani karena nilai maksimum pada fungsi Likelihood tidak dapat diestimasi pada respons yang hanya benar atau salah saja. Pada *packages* mirt menyediakan fungsi untuk mengestimasi θ dengan metode MLE, MAP, dan EAP menggunakan fungsi fscores. Pada contoh berikut ini, akan kita gunakan kembali Data 5000x30-HD untuk mengestimasi θ dengan menggunakan hasil estimasi parameter butir dengan model IRT 3-PL.

```
latent_mle <- fscores(fit_3PL, method = "ML",
                      full.scores = TRUE, full.scores.SE =
TRUE)
latent_map <- fscores(fit_3PL, method = "MAP",
                      full.scores = TRUE, full.scores.SE =
TRUE)
latent_eap <- fscores(fit_3PL, method = "EAP",
                      full.scores = TRUE, full.scores.SE =
TRUE)
```

Hasil estimasi dengan metode MLE, MAP, dan EAP disimpan dengan nama masing-masing latent_mle, latent_map, dan latent_eap. untuk melihat hasil estimasi dapat digunakan fungsi head() pada masing-masing objek hasil estimasi.

```
head(latent_mle)
            F1     SE_F1
[1,] -1.3060349 0.5618675
[2,]  0.6106992 0.2956062
[3,] -0.6836921 0.3876538
[4,]  1.5003470 0.3182265
[5,]  0.9894194 0.2976850
[6,]       -Inf        NA
```

```
head(latent_map)
             F1     SE_F1
[1,] -1.0306933 0.4323109
[2,]  0.5613993 0.2847911
[3,] -0.5969463 0.3511864
[4,]  1.3651796 0.2973764
[5,]  0.9088471 0.2855400
[6,] -1.7845607 0.5474136
```

```
head(latent_eap)
             F1     SE_F1
[1,] -1.1140868 0.4547821
[2,]  0.5533318 0.2882759
```

```
[3,] -0.6390497 0.3627656
[4,]  1.3760368 0.3017219
[5,]  0.9065681 0.2900860
[6,] -1.9075729 0.5676012
```

Hasil estimasi tingkat kemampuan θ (atribut laten) ditunjukkan pada kolom F1 sedangkan kolom SE_F1 menunjukkan standar error. Kolom pertama menunjukkan urutan peserta. Hasil estimasi ini dapat digabung menjadi 1 data frame dengan mengambil nilai pada kolom F1 . Hasil penggabungan ini selanjutnya dapat disimpan dengan format *xlsx seperti yang sudah dilakukan pada hasil estimasi parameter butir. Perintah yang digunakan untuk menggabungkan hasil estimasi kemampuan adalah sebagai berikut.

```
latent <- data.frame(MLE = latent_mle[ ,1],
                     MAP = latent_map[ ,1],
                     EAP = latent_eap[ ,1])
head(latent)
```

```
         MLE        MAP        EAP
1 -1.3060349 -1.0306933 -1.1140868
2  0.6106992  0.5613993  0.5533318
3 -0.6836921 -0.5969463 -0.6390497
4  1.5003470  1.3651796  1.3760368
5  0.9894194  0.9088471  0.9065681
6       -Inf -1.7845607 -1.9075729
```

Pada output hasil analisis, diperlihatkan bahwa hasil estimasi dengan metode MLE pada contoh di atas menghasilkan nilai -inf untuk peserta ke-6 mengindikasikan bahwa pola respons peserta dengan urutan ke-6 pada data adalah salah semua (0). Apabila pola respons adalah benar semua (1) maka hasil estimasi MLE akan menghasilkan +inf. Sementara itu, pada metode MAP dan EAP, masih dapat mengestimasi θ untuk pola respons tersebut.

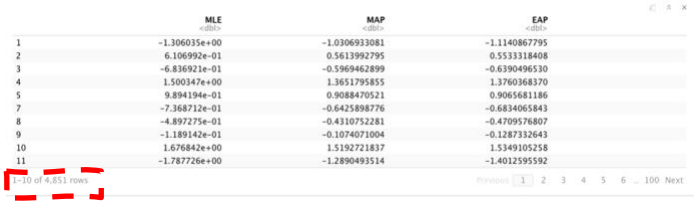
```
latent[c(6,984,388),]
```

```
     MLE       MAP       EAP
6   -Inf -1.784561 -1.907573
984  Inf  3.071852  3.154655
388  Inf  3.071852  3.154655
```

Pada contoh di atas, terlihat hasil estimasi MLE terhadap kemampuan peserta ke 6 adalah -inf karena responsnya semua 0, dan inf untuk hasil estimasi peserta ke 388 dan 984 yang mengindikasikan jawaban benar semua (1). Selanjutnya, hasil estimasi masing-masing

metode ini dapat ditampilkan statistik deskriptifnya dengan menggunakan fungsi summary dan apply. Namun sebelum dapat menghitung ringkasan statistiknya, hasil estimasi MLE harus dibersihkan dulu dari nilai -inf dan inf dengan fungsi is.finite.

```
latent_est <- latent[is.finite(latent$MLE), ]
```



	MLE <dbl>	MAP <dbl>	EAP <dbl>
1	-1.306035e+00	-1.0306933081	-1.1140867795
2	6.106992e-01	0.5613992795	0.5533318408
3	-6.836921e-01	-0.5969462899	-0.6390496530
4	1.500347e+00	1.3651795855	1.3760368370
5	9.894194e-01	0.9088470521	0.9065681186
7	-7.368712e-01	-0.6425898776	-0.6834065843
8	-4.897275e-01	-0.4310752281	-0.4709576807
9	-1.189142e-01	-0.1074071004	-0.1287332643
10	1.676842e+00	1.5192721837	1.5349105258
11	-1.787726e+00	-1.2890493514	-1.4012595592

1–10 of 4,651 rows

Previous 1 2 3 4 5 6 ... 100 Next

Dengan fungsi is.finite diperoleh data yang tersisa sebanyak 4581 respons. latent_est digunakan untuk menghitung statistik deskrpitif dari masing-masing metode. Fungsi apply digunakan dengan memasukkan nama dataframe yakni latent_est, kemudian angka 2 karena data akan diringkas berdasarkan kolom (1 jika anak diringkas berdasarkan baris), dan terakhir adalah metode yang dipakai (summary dan sd).

```
apply(latent_est, 2, summary)
apply(latent_est, 2, sd)
```

```
                MLE          MAP          EAP
Min.    -14.92145559 -1.75386401 -1.873743382
1st Qu.  -0.63137909 -0.55128502 -0.593384299
Median    0.02871796  0.02603568  0.002592781
Mean      0.01173587  0.08321656  0.054079036
3rd Qu.   0.72346256  0.66479158  0.657520016
Max.      3.91587363  2.82631553  2.894041911
      MLE       MAP       EAP
1.2813387 0.8537320 0.8857811
```

Rata-rata (mean) hasil estimasi kemampuan dari ketiga metode di atas tidak terlalu jauh berbeda. Namun jika dilihat standar deviasi untuk metode MLE adalah yang paling besar diantara ketiganya, sedangkan MAP, dan EAP menghasilkan sd yang tidak jauh berbeda. Parameter kemampuan hasil estimasi MLE, MAP, dan EAP dapat juga dicermati korelasinya dengan menggunakan fungsi cor.

```
cor(latent_est)
          MLE       MAP       EAP
MLE 1.0000000 0.8543419 0.8558475
```

```
MAP 0.8543419 1.0000000 0.9999177
EAP 0.8558475 0.9999177 1.0000000
```

Korelasi yang ditampilkan antar metode estimasi kemampuan menunjukkan bahwa korelasi yang tinggi yang berarti ada kesesuaian antara metode MLE, MAP, dan EAP. Selanjutnya, dihitung *root mean square deviation* (RMSD) yang menunjukkan perbedaan antara standar deviasi hasil estimasi MLE, MAP, dan EAP.

$$RMSD = \sqrt{\frac{\sum_{j=1}^{N}\left(\hat{\theta}_{1j} - \hat{\theta}_{2j}\right)^2}{N}}, \qquad (6.10)$$

dimana, $\hat{\theta}_{1j}$ dan $\hat{\theta}_{2j}$ adalah hasil estimasi kemampuan oleh metode 1 dan metode 2.

```
rmsd_MLE_MAP <- sqrt(sum(latent_est$MLE -
latent_est$MAP)^2/length(latent_est$MLE))
rmsd_MLE_MAP
[1] 4.978568
rmsd_MLE_EAP <- sqrt(sum(latent_est$MLE -
latent_est$EAP)^2/length(latent_est$MLE))
rmsd_MLE_EAP
[1] 2.949165
rmsd_MAP_EAP <- sqrt(sum(latent_est$MAP -
latent_est$EAP)^2/length(latent_est$MAP))
rmsd_MAP_EAP
[1] 2.029403
```

Hasil estimasi RMSD menunjukkan nilai yang cukup besar. Nilai RMSD terbesar ditemukan antara metode MLE dan MAP, kemudian MLE dengan EAP. Sedangkan EAP dan MAP menghasilkan nilai RMSD yang paling kecil. Akan tetapi, ini bukan ketentuan yang pasti anda temukan pada setiap hasil estimasi. Desjardins dan Bulut (2018) pada estimasi yang menggunakan data lain mendapatkan nilai RMSD justru antara MLE dan MAP. Selain itu, mereka mendapatkan nilai RMSD yang lebih kecil (0,03 – 0,09) dari yang diperoleh pada contoh ini.

6.6 Diagnosa Model

Pada IRT, diagnosa model dapat dilakukan pada tiga level, yakni item, person, dan model (goodness of fit model). Pada subbab berikut

akan dibahas diagnosa model yang diperiksa dengan menggunakan packages mirt.

6.6.1 Item Fit

Ada dua metode yang dapat diaplikasikan untuk memeriksa item fit, yakni secara statistik dan secara grafis. Desjardins dan Bulut (Desjardins & Bulut, 2018) menyebutkan statistik yang paling sering digunakan adalah *signed* Chi Square (χ^2) (Orlando & Thissen, 2000), χ^2 Bock (Bock, 1972), $Q1$ Yen (Yen, 1981), statistik $G2$ (McKinley & Mills, 1985), dan statistik infit dan outfit untuk pemodelan Rasch. Statistik χ^2 Bock, statistik $Q1$ Yen, dan statistik $G2$ bergantung pada perbandingan probabilitas empiris dengan probabilitas yang diprediksi. Pada metode-metode statistik tersebut, hipotesis nol yang diuji adalah bahwa model IRT yang dipilih cocok dengan data. Statistik χ^2 diberikan dengan persamaan matematika berikut.

$$\chi^2 = \sum_{j=1}^{J} N_j \frac{\left(O_j - E_j\right)^2}{E_j\left(1 - E_j\right)} \ , \qquad (6.11)$$

dimana j adalah indeks dari interval atribut laten (misalnya $\theta = -2$ sampai $\theta = -1.5$), O_j adalah probabilitas menjawab benar yang teramati (empirik) pada interval j, E_j adalah probabilitas menjawab benar yang diduga (teoritik) pada interval j berdasarkan model IRT yang fit, N adalah jumlah respons/examinee/peserta yang berada pada interval atribut laten j.

Pada statistik χ^2 Bock, setiap interval atribut laten memiliki ukuran yang sama, dan nilai tengah (median) pada setiap interval digunakan untuk menghitung proporsi benar. Sementara itu, statistik $Q1$ Yen memerlukan 10 interval atribut laten dengan frekuensi yang sama, dan probabilitas yang diestimasi adalah rerata (*mean*) dari probabilitas respons benar. Kedua statistik ini memiliki derajat bebas $J - m$, dimana m adalah jumlah parameter pada model IRT.

Adanya item yang secara signifikan tidak cocok dengan model akan menyebabkan bias dalam estimasi atribut laten. Oleh karenanya, Orlando dan Thissen (2000) mengusulkan agar mengelompokkan *examinee* berdasarkan *raw score*, tidak berdasarkan hasil estimasi atribut laten. Metode yang diusulkan oleh Orlando dan Thissen (2000)

ini disebut *signed* χ^2 atau statistik $S-X^2$ yang ditunjukkan oleh model matematika berikut.

$$S-X^2=\sum_{k}^{n-1} N_k \frac{(O_k-E_k)^2}{E_k(1-E_k)} \qquad (6.12)$$

dimana n adalah jumlah item pada satu perangkat instrumen, k adalah *raw score* (jumlah respons benar), dan elemen lainnya sama dengan yang digunakan pada persamaan χ^2. Adapun derajat bebas (df) pada statistik $S-X^2$ adalah $(n-1)-m$, dimana m adalah jumlah parameter pada model IRT.

Sementara itu, statistik $G2$ berdasarkan uji rasio Likelihood. Dengan elemen yang sama pada persamaan χ^2 dan $S-X^2$, statistik $G2$ dihitung dengan menggunakan persamaan berikut.

$$G^2=\sum_{j=1}^{J} N_j\left(O_j \log\frac{O_j}{E_j}+\left(1-O_j\right)\log\frac{1-O_j}{1-E_j}\right) \qquad (6.1.3)$$

Statistik $G2$ memiliki derajat bebas $(df) = J-m$. Seperti yang sudah dijelaskan di atas bahwa statistik χ^2 Bock, statistik $Q1$ Yen, dan statistik $G2$ menguji hipotesis nol yang sama yakni item memiliki kecocokan dengan model IRT. Oleh karena itu, item yang tidak signifikan menunjukkan item yang tidak fit dengan model IRT. Untuk melakukan pengujian statistik item fit pada R digunakan fungsi itemfit(). Pada contoh berikut akan dilakukan pemeriksaan item fit pada seluruh model yang sudah didemonstrasikan pada sub bab sebelumnya dengan menggunakan data Data_5000x30-HD.

```
itemfit_rasch <- itemfit(fit_Rasch, fit_stats =
                  c("S_X2", "G2"), impute = 10)
head(itemfit_rasch)
```

```
  item     G2 df.G2 RMSEA.G2  p.G2   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   b1 98.371     9    0.045 0.000 47.688      23      0.015  0.002
2   b2 40.255     9    0.026 0.000 49.087      25      0.014  0.003
3   b3 22.221     9    0.017 0.008 74.201      25      0.020  0.000
4   b4 31.981     9    0.023 0.000 41.741      25      0.012  0.019
5   b5  6.405     9    0.000 0.699 34.846      25      0.009  0.091
6   b6  9.445     9    0.003 0.397 33.658      25      0.008  0.115
```

Pada output hasil analisis di atas, diperlihatkan hasil uji itemfit dengan menggunakan statistik S_X2, dan G2 dan ditampilkan hasil uji untuk enam item pertama. Item 1, 2, 3, dan 4 menunjukkan ketidakcocokan dengan model Rasch berdasarkan nilai p pada statistik S_X2 (0.000) dan G2 (0.019). Sementara itu, item 5 dan item 6 menunjukkan kecocokan dengan model Rasch dengan nilai p pada kedua uji statistik yang lebih dari 0.05.

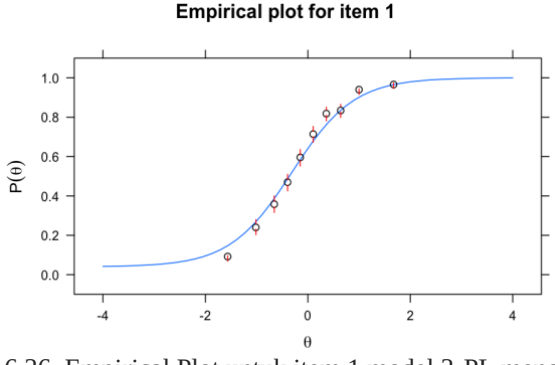
```
itemfit_3PL <- itemfit(fit_3PL, fit_stats = c("S_X2",
"G2"),impute = 10)
head(itemfit_3PL)
```

```
  item     G2 df.G2 RMSEA.G2  p.G2   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   b1 39.686     7    0.031 0.000 23.025      21      0.004  0.343
2   b2 19.308     7    0.019 0.007 45.631      24      0.013  0.005
3   b3 12.948     7    0.013 0.073 23.270      25      0.000  0.562
4   b4 22.451     7    0.021 0.002 31.076      25      0.007  0.187
5   b5 12.329     7    0.012 0.090 15.593      24      0.000  0.902
6   b6 12.106     7    0.012 0.097 25.767      24      0.004  0.365
```

Selanjutnya, jika dilakukan pengujian statistik pada model 3-PL diperoleh hasil yang sedikit berbeda pada enam item pertama. Statistik $G2$ menunjukkan item 3 cocok dengan model IRT 3-PL, sedangkan statistik S_X2 menunjukkan 5 item fit, dan hanya item 2 yang tidak memenuhi.

Fungsi itemfit dapat pula digunakan untuk menggambar plot emprik untuk item, yang berguna untuk mendeteksi item-item yang tidak fit. Pada contoh berikut diperlihatkan *empirical plot* untuk item 1 yang diperiksa kecocokannya dengan model IRT 3-PL.

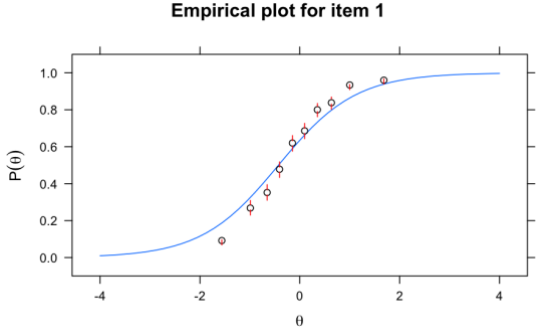
```
itemfit(fit_3PL, empirical.plot = 1)
```



Gambar 6.36. Empirical Plot untuk item 1 model 3-PL menggunakan Data_5000x30-HD

Gambar 6.36 menunjukkan probabilitas empiris dan teoritis untuk item 1 dengan menggunakan model IRT 3-PL. Tititik menunjukkan data empiris, sedangkan garis menunjukkan probabilitas teoritis. Semakin dekat titik-titik data empiris dengan garis teoritis, semakin baik kecocokannya dengan model. Jika dibandingkan, dengan model IRT 1-PL (Gambar 6.37) terlihat bahwa kecocokannya item 1 tidak lebih baik dibandingkan dengan model 3-PL.

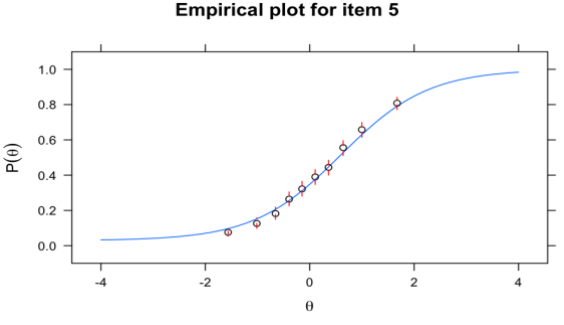
```
itemfit(fit_1PL, empirical.plot = 1)
```



Gambar 6.37. Empirical Plot untuk item 1 model 1-PL menggunakan Data_5000x30-HD

Hal yang sama juga terlihat jika dibandingkan itemfit untuk item 1 dengan item 5 menunjukkan perbedaan. Gambar 6.38 menunjukkan item 5 lebih baik kecocokannya dengan model 3-PL dibandingkan dengan item 1 karena sebaran titik pada item 5 lebih dekat dengan probabilitas empirisnya.

```
itemfit(fit_3PL, empirical.plot = 5)
```



Gambar 6.38. Empirical Plot untuk item 5 model 3-PL menggunakan Data_5000x30-HD

6.6.2 Person Fit

Person fit menunjukkan kecocokan antara pola respons peserta dengan model IRT yang dipilih. Person fit digunakan untuk menilai validitas model yang dipilih pada level *examinee* kebermaknaan hasil estimasi tingkat atirbut laten atau kemampuan. Pada subbab ini, akan digunakan standardized fit index (statistik Z_h) yang diajukan oleh Drasgow, et al. (1985). Statistik Z_h untuk *examinee* ke – j dengan tingkat atribut laten θ_j dapat dihitung dengan model matematika berikut.

$$Z_h = \sum \frac{[LogL|\theta_j - \sum E(LogL|\theta_j)]}{\sqrt{(\sum V(LogL|\theta_j))}}, \qquad (6.14)$$

dimana $LogL$ adalah nilai log-likelihood untuk pola respons *examinee* ke-j, $E(LogL|\theta_j)$ adalah rata-rata nilai log-likelihood untuk distribusi sampel dari log-likelihood kondisional pada θ_j, dan $V(\theta_j)$ adalah varians distribusi sampel daro nilai log-likelihood. Karena statistik Z_h mengikuti distribusi normal, maka nilai dugaan dari $Z_h$adalah nol (0) ketika pola respons sejalan dengan model IRT yang dipilih. Nilai negatif yang besar Z_h (misalnya $Z_h < -2$) mengindikasikan person misfit (pola respons tidak cocok dengan model). Nilai positif dan besar pada Z_h mengindikasikan bahwa likelihood untuk pola respons *examinee* lebih tinggi dari likelihood yang diprediksi berdasarkan pada model IRT yang dipilih. Kondisi ini juga menunjukkan ketidakcocokan pola respons dengan model IRT.

Pada contoh berikut, fungsi personfit pada *packages* mirt untuk menghitung statistik Z_h menggunakan data Data_5000x30-HD dengan menggunakan model IRT 3-PL. Fungsi personfit dapat dijalankan jika tidak ada *missing data* respons. Oleh karena itu, pertama dilakukan ekslusi respons yang memuat *missing data* menggunakan fungsi na.omit(data). Setelah mengeluarkan *missing data*, dilakukan analisis ulang dengan model IRT, dan dilanjutkan dengan pemeriksaan personfit.

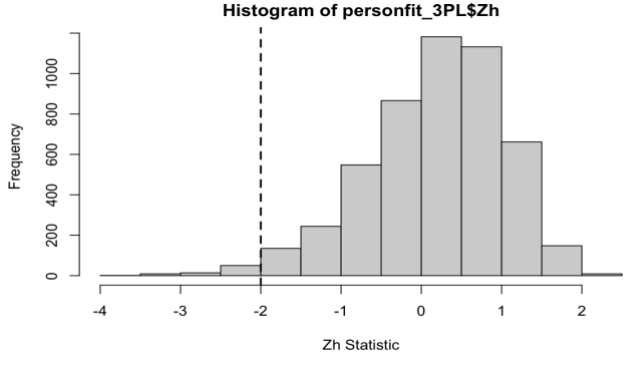
```
data_nomiss <- na.omit(data)
```

```
fit_3PL <- mirt(data = data_nomiss, model = 1,itemtype =
"3PL",
                SE = TRUE, verbose = FALSE)
personfit_3PL <- personfit(fit_3PL)
head(personfit_3PL)
```

```
     outfit    z.outfit     infit     z.infit         Zh
1 3.4590195  2.61937881 1.3828398  0.97219201 -1.5803310
2 1.0687431  0.33818130 1.1482503  0.89456622 -0.6902491
3 0.8949704 -0.04303122 0.9511926 -0.07600166  0.1448672
4 0.9268837 -0.08794066 1.0904635  0.51620523 -0.2614706
5 0.8076696 -0.66813321 0.9093274 -0.53984704  0.6180355
6 0.0597248 -1.81135240 0.0791824 -2.24067530  1.2984648
```

Pada data Data_5000x30-HD tidak ada *missing data* sehingga analisis data menggunakan selururuh raw data yabng berjumlah 5000 respons. Fungsi na.omit(data) di atas akan mengeluarkan data *missing* dan menyimpannya pada *dataframe* data_nomiss yang selanjutnya digunakan dalam estimasi parameter dan personfit. Pada output hasil analisis di atas, empat kolom pertama adalah nilai statistik outfit, z.outfit, infit, dan z.infit untuk masing-masing *person*. Kolom terakhir (Zh) digunakan untuk mengevaluasi personfit pada model yang dipilih (IRT 3-PL). Kriteria person fit model adalah jika Zh nilainya lebih dari -2. Untuk mendeteksi person fit ini dapat pula digunakan histogram (Gambar 6.39) dengan menambahkan garis batas dengan fungsi abline(v = -2, lwd = 2, lty = 2).

```
hist(personfit_3PL$Zh, xlab="Zh Statistic")
abline(v = -2, lwd = 2, lty = 2)
```



Gambar 6.39. Empirical Plot untuk item 5 model 3-PL menggunakan Data_5000x30-HD

```
misfits_3PL <- subset(personfit_3PL, Zh < -2)
rownames(misfits_3PL)
```

```
[1] "24"   "427"  "465"  "487"  "522"  "536"  "576"  "601"  "869"  "876"
 "1063" "1151" "1166" "1191" "1254" "1343" "1414" "1589" "1616" "1639"
[21] "1729" "1821" "1848" "1981" "2054" "2135" "2138" "2205" "2264" "2379"
      "2403" "2405" "2518" "2559" "2582" "2643" "2644" "2850" "3064" "3205"
[41] "3243" "3293" "3403" "3456" "3496" "3526" "3638" "3640" "3657" "3915"
       "3929" "3984" "4019" "4144" "4237" "4358" "4394" "4404" "4409" "4410"
[61] "4508" "4542" "4581" "4593" "4670" "4713" "4716" "4722" "4757" "4786"
       "4856" "4857" "4956" "4983"
```

Selanjutnya, pola respons yang tidak fit dengan model dapat ditampilkan dengan menggunakan fungsi subset(). Dari 5000 data respons, diperoleh 74 pola respons yang tidak fit dengan model 3PL.

```
nrow(misfits_3PL)
[1] 74
```

6.6 Pemilihan Model

Pemilihan model IRT dalam bergantung pada teori yang mendasari proses asesmen dan karakteristik data respons. Untuk menentukan model mana yang akan digunakan digunakan statistik berbasis Likelihood untuk membandingkan kriteria kecocokan model. Perbandingan model pada *packages* mirt dapat dilakukan dengan fungsi anova. Contoh berikut ini kita akan gunakan kembali hasil estimasi parameter butir yang sudah disimpan dengan fit_1PL, fit Rasch, fit_2PL, fit_3PL, dan fit_4PL. Perintah berikut akan menghasilka tabel ringkasan AIC, AICc, SABIC, BIC, loglik, X2, df, dan p untuk masing-masing model.

```
anova(fit_1PL, fit_Rasch,fit_2PL,fit_3PL,fit_4PL)
       AIC    SABIC       HQ      BIC    logLik      X2  df    p
1 148725.4 148829.0 148796.2 148927.5 -74331.71     NaN NaN  NaN
2 148725.4 148829.0 148796.2 148927.5 -74331.71   0.000   0 0.00
3 148315.8 148516.2 148452.9 148706.9 -74097.92 467.598  29 0.00
4 148252.0 148552.5 148457.5 148838.5 -74035.98 123.870  30 0.00
5 148297.0 148697.7 148571.1 149079.0 -74028.48  15.004  30 0.99
```

Model yang terbaik ditunjukkan oleh nilai AIC, BIC, SABIC yang paling kecil. Pada contoh hasil analisis terlihat bahwa model yang paling baik adalah IRT 3-PL berdasarkan nilai AIC yang terkecil diantara 4 model lain, yakni 148252. Jika melihat dari SABIC justru model IRT 2-PL yang memiliki SABIC terkecil yakni 14516.2. untuk melihat perbandingan kecocokan model, berikut ini dibandingkan kriteria kecocokan model dari model 1-PL dengan 2-PL, 2-PL dengan 3-PL, dan 3-PL dengan 4-PL.

```
anova(fit_1PL,fit_2PL)
Model 1: mirt(data = data, model = model1PL, SE = TRUE)
Model 2: mirt(data = data, model = 1, itemtype = "2PL", SE =
TRUE)
      AIC    SABIC       HQ      BIC    logLik      X2  df   p
1 148725.4 148829.0 148796.2 148927.5 -74331.71     NaN NaN NaN
2 148315.8 148516.2 148452.9 148706.9 -74097.92 467.598  29   0
```

```
anova(fit_2PL,fit_3PL)
Model 1: mirt(data = data, model = 1, itemtype = "2PL", SE =
TRUE)
Model 2: mirt(data = data, model = 1, itemtype = "3PL", SE =
TRUE)
      AIC    SABIC       HQ      BIC    logLik     X2  df   p
1 148315.8 148516.2 148452.9 148706.9 -74097.92    NaN NaN NaN
2 148252.0 148552.5 148457.5 148838.5 -74035.98 123.87  30   0
```

Perbandingan model 1-PL dan 2-PL menunjukkan bahwa model 2-PL memiliki derajat kecocokan model yang lebih baik dibandingkan model 1-PL dengan $\chi^2_{29} = 467.598$ dan $p < 0.05$. Selanjutnya, membandingkan model 2-PL dengan 3-PL, didapatkan hasil bahwa model paling cocok adalah 3-PL dengan $\chi^2_{30} = 1123.87$ dan $p < 0.05$. Selanjutnya, perbandingan model 3-PL dan model 4-PL menghasilkan $\chi^2_{30} = 15.004$ dan $p = 0.99 > 0.05$. Artinya, kecocokan model 3-PL tidak berbeda signifikan dengan model 4-PL.

```
anova(fit_3PL,fit_4PL)
Model 1: mirt(data = data, model = 1, itemtype = "3PL", SE =
TRUE)
Model 2: mirt(data = data, model = 1, itemtype = "4PL", SE =
TRUE)
     AIC    SABIC       HQ      BIC    logLik     X2  df    p
1 148252 148552.5 148457.5 148838.5 -74035.98    NaN NaN  NaN
2 148297 148697.7 148571.1 149079.0 -74028.48 15.004  30 0.99
```

Desjardins dan Bulut (Desjardins & Bulut, 2018) menyebutkan bahwa pada pengujian secara statistik, biasanya model yang memiliki parameter lebih banyak akan menghasilkan kecocokan model dengan data yang lebih baik dibandingkan yang lebih sedikit parameternya. Mereka menggarisbawahi bahwa pemilihan model seharusnya tidak hanya berdasarkan hasil uji statistik tersebut atau hanya berdasarkan asumsi teoritik penilaian. Mereka mendorong pembaca untuk mereview estimasi parameter item dan kecocokan model sebagai bagian dari pengembangan tes, bukan sebagai tahapan yang terpisah.

Referensi

Chalmers, R. P. (2012). mirt : A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, *48*(6), 1–29. https://doi.org/10.18637/jss.v048.i06

Chalmers, R. P. (2021). *mirt: Multidimensional Item Response Theory*. https://cran.r-project.org/package=mirt

Desjardins, C. D., & Bulut, O. (2018). *Handbook of Educational Measurement and Psychometrics Using R*. CRC Press.

Hambleton, R. K., & Swaminathan, H. (1985). *Item response theory: Principles and applications*. Nijhof'f Publishing.

Han, K. T. (2007). WinGen: Windows software that generates item response theory parameters and item responses. *Applied Psychological Measurement*, *31*(5), 457–459. https://doi.org/10.1177/0146621607299271

Hatzinger, R., & Rusch, T. (2009). IRT models with relaxed assumptions in eRm: A manual-like instruction. *Psychology Science Quarterly*, *51*(1).

Mair, P., & Hatzinger, R. (2007a). CML based estimation of extended Rasch models with the eRm package in R. *Psychology Science*, *49*(1).

Mair, P., & Hatzinger, R. (2007b). Extended Rasch modeling: The eRm package for the application of IRT models in R. *Journal of Statistical Software*, *20*(9). https://www.jstatsoft.org/v20/i09

Mair, P., Hatzinger, R., & Maier, M. J. (2021a). *eRm: Extended Rasch Modeling*. https://cran.r-project.org/package=eRm

Mair, P., Hatzinger, R., & Maier, M. J. (2021b). *eRm: Extended Rasch Modeling*. https://cran.r-project.org/package=eRm

Ooms, J. (2021). *writexl: Export Data Frames to Excel xlsx Format*. https://cran.r-project.org/package=writexl

Paek, I., Liang, X., & Lin, Z. (2021). Regarding Item Parameter Invariance for the Rasch and the 2-Parameter Logistic Models: An Investigation under Finite Non-Representative Sample Calibrations. *Measurement: Interdisciplinary Research and Perspectives*, *19*(1), 39–54. https://doi.org/10.1080/15366367.2020.1754703

R Core Team. (2022a). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*.

R Core Team. (2022b). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/

Retnawati, H. (2014). *Teori respon butir dan penerapannya: untuk peneliti, praktisi pengukuran dan pengujian, mahasiswa pascasarjana*. Parama Publishing.

Rizopoulos, D. (2022). *ltm: Latent Trait Models under IRT*. https://github.com/drizopoulos/ltm

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. Springer. http://lmdvr.r-forge.r-project.org

Sarkar, D. (2021). *lattice: Trellis Graphics for R*. http://lattice.r-forge.r-project.org/

Sarkar, D., & Andrews, F. (2019). *latticeExtra: Extra Graphical Utilities Based on Lattice*. http://latticeextra.r-forge.r-project.org/

Yen, W. M. (1981). Using Simulation Results to Choose a Latent Trait Model. *Applied Psychological Measurement, 5*(2), 245–262. https://doi.org/10.1177/014662168100500212

Chapter 7
IRT Unidimensi Penskoran Politomi

Oleh: Raoda Ismail, Nur Anisyah Rachmaningtyas, & Samsul Hadi

Pada pembahasan *Chapter* sebelumnya, telah diperkenalkan model Teori Respons Butir Unidimensional untuk butir yang dinilai secara dikotomus dengan dua kategori respons (misalnya benar atau salah, setuju atau tidak setuju, dan ya atau tidak). Pada saat butir terdiri dari lebih dari dua kategori respons, baik kategori ordinal maupun nominal, butir ini disebut sebagai butir dengan skor politomi atau disebut sebagai butir politomi. Nomor yang ditetapkan untuk tanggapan yang berbeda secara kategoris biasanya adalah 0, 1, 2, dan 3 (atau 1, 2, 3 dan 4) untuk empat pilihan atau set empat tingkat kategori dalam sebuah item. Chapter ini memperkenalkan aplikasi dari model IRT yang menangani tanggapan item dengan skor politomi.

Pada model respons butir yang dinilai secara dikotomi, fungsi respons butir (*Item Response Function*-IRF) adalah unit penting dari pemodelan respons item. Dalam pemodelan respons politomi, unit dasar pemodelan berada pada tingkat kategori, yaitu fungsi respons kategori (*Category Response Function*-CRF), atau fungsi probabilitas kategori. Sama seperti IRF dikotomis yang dapat ditampilkan secara grafis menggunakan kurva karakteristik item (*Item Characteristic Curve*-ICC), CRF politomi dapat ditampilkan secara grafis menggunakan kurva karakteristik kategori (*Category Characteristic Curve*-CCC), atau kurva probabilitas kategori (Paek & Cole, 2020).

Pada *Chapter* 7, dijelaskan model respons butir untuk butir yang diberi skor politomi. Model teori resposn butir untuk butir politomi dapat dilihat sebagai bentuk umum dari model teori respons butir untuk butir dengan skor dikotomi (misalnya model Rasch, model 1 PL, model 2 PL, atau model 3PL). Setara dengan Chapter 6, pada Chapter ini akan disajikan berbagai model teori respons butir yang dirancang untuk butir dengan skor politomi dan menunjukkan bagaimana memperkirakan model ini dalam R. Model Teori Respons Butir politomi yang dijelaskan adalah *Partial Credit Model* (Masters, 1982), *Generalized Partial Credit Model* (Muraki, 1992), *Rating Scale Model* (Andrich,

1978), *Graded Response Model* (Samejima, 1969), *Nominal Response Model* (Bock, 1972), dan *Nested Logit Model*-NLM (Suh & Bolt, 2010). Akan digunakan MIRT Package (Chalmers, 2012) untuk mendemonstrasikan estimasi model teori respons butir untuk butir yang diberi skor politomi. Peneliti lain juga telah mengusulkan kategorisasi serupa untuk model IRT politomi (De Ayala, 2013).

Model respons butir politomi dapat dikategorikan menjadi model respons butir nominal dan ordinal, tergantung pada asumsi karakteristik data (Retnawati, 2014: 32) .Dalam Chapter ini, akan dikategorikan model Teori Respons Butir politomi ke dalam tiga kelompok berdasarkan skala pengukurannya yaitu ordinal atau nominal, dan untuk melihat apakah model tersebut termasuk dalam keluarga model Rasch: (1) Model Rasch Polytomous untuk item ordinal; (2) Model non-Rasch Polytomous untuk item ordinal; dan (3) Model IRT Politomi untuk item nominal.

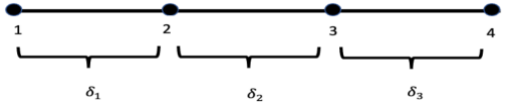
7.1 Model Rasch Polytomous untuk Item Ordinal

Partial Credit Model (PCM) dan Rating Scale Model (RSM) dianggap sebagai bentuk politomi dari model Rasch. Model-model ini mengasumsikan bahwa kategori yang berdekatan dalam item ordinal dengan skor politomi (misalnya, sangat setuju, setuju, tidak setuju, atau sangat tidak setuju) sebenarnya adalah dua kategori dikotomis, seperti dalam model IRT dikotomis. Oleh karena itu, model-model ini menggunakan logika yang sama dari model IRT dikotomi untuk memperkirakan probabilitas memperoleh kategori yang lebih tinggi (misalnya, sangat setuju daripada setuju) berdasarkan tingkat sifat laten responden. Meskipun menggunakan mekanisme yang sama untuk mengestimasi parameter item, PCM dan model skala penilaian berbeda satu sama lain dalam hal mengestimasi batas kategori (yaitu, ambang batas). Bagian berikut akan menjelaskan secara singkat model-model ini dan mendemonstrasikan bagaimana memperkirakannya menggunakan paket MIRT dalam R.

7.1.1 Partial Credit Model (PCM)

Partial Credit Model (PCM) atau model kredit parsial (Masters, 1982) adalah model respons item keluarga Rasch yang dapat

menangani respons item dengan lebih dari dua kategori. PCM adalah perpanjangan langsung dari model Rasch sederhana dikotomis. Model kredit parsial atau PCM yang juga dikenal sebagai model logit kategori yang berdekatan, adalah bentuk politomi dari model Rasch. PCM mengasumsikan bahwa item polytomous terdiri dari beberapa kategori. Model berfokus pada kategori yang berdekatan ketika memperkirakan ambang (yaitu, kesulitan) antara kategori respons. Mengingat bahwa suatu item memiliki K kategori respons yang dipesan, PCM memperkirakan ambang batas K−1 untuk item tersebut. Misalnya, jika item memiliki empat kategori respons (sangat tidak setuju, tidak setuju, setuju, atau sangat setuju), PCM akan memperkirakan tiga ambang ($\delta_1, \delta_2, \delta_3$) antara kategori yang berdekatan. Gambar 7.1 menunjukkan ambang batas untuk item empat kategori.



Gambar 7.1 Ambang Batas antara Empat Kategori Respons (Desjardins & Bulut, 2018)

Peluang diperolehnya poin X_i $(X_i = 0, 1, \dots, m_i)$ pada butir i untuk PCM dapat ditulis sebagai:

$$P(X_i|\theta, \delta_{ih}) = \frac{\exp\left[\sum_{h=0}^{X_i} \theta - \delta_{ih}\right]}{\sum_{k=0}^{m_i} \exp\left[\sum_{h=0}^{k} \theta - \delta_{ih}\right]} \qquad (7.1)$$

di mana θ adalah sifat laten, δ_{ih} adalah parameter langkah (juga dikenal sebagai kesulitan langkah) yang mewakili kesulitan relatif dalam memperoleh h poin di atas (h-1) poin, karena subskrip m_i menetapkan kategori respons maksimum untuk setiap item secara individual, PCM memungkinkan jumlah kategori respons berbeda antar item.

Perlu dicatat bahwa PCM tidak memerlukan ambang batas untuk mengikuti urutan yang sama dengan kategori respons. Karena PCM mempertimbangkan kategori yang berdekatan di setiap langkah, kategori respons yang berdekatan diperlakukan sebagai serangkaian item dikotomis, tetapi tanpa batasan urutan di luar kategori yang berdekatan. Melanjutkan contoh yang sama pada Gambar 7.1, ambang

batas antara kategori respons 1 dan 2 (yaitu, δ_1) dapat lebih besar dari ambang batas (δ_2) antara kategori respons 2 dan 3.

Fitur ini membuat PCM sangat fleksibel saat memperkirakan parameter item untuk skala penilaian dan survei tipe Likert karena terkadang responden lebih sering memilih kategori yang lebih tinggi dibandingkan kategori yang lebih rendah. Misalnya, asumsikan pertanyaan survei menanyakan seberapa sering seseorang menggunakan ponsel cerdasnya di siang hari, dan pilihan jawabannya tidak pernah, sangat jarang, kadang-kadang, dan sangat sering. Jika sebagian besar responden cenderung sangat sering menggunakan ponsel mereka di siang hari, memilih opsi respons kadang-kadang dan sangat sering lebih mungkin dipilih daripada tidak pernah atau sangat jarang. Dengan demikian, ambang batas antara kadang-kadang dan sangat sering dapat lebih kecil daripada ambang batas antara tidak pernah dan sangat jarang.

Tidak seperti item tipe Likert, item polytomous yang mengukur pencapaian atau bakat diharapkan mengikuti urutan yang sama dengan kategori respons. Misalnya, dalam masalah matematika dengan kemungkinan skor 0, 1, dan 2, memperoleh 2 poin akan lebih sulit daripada memperoleh 0 atau 1 poin. Oleh karena itu, kami mengharapkan ambang batas antara 1 dan 2 lebih besar dari ambang batas antara 0 dan 1. Bergantung pada konten item dan sifat laten yang diukur, pengguna harus memastikan bahwa urutan kategori respons dari PCM berfungsi seperti yang diharapkan.

Dalam contoh berikut, digunakan kumpulan data rse dari *hemp package* untuk mendemonstrasikan cara memasang PCM. Kumpulan data rse berasal dari skala harga diri Rosenberg yang mengukur harga diri individu. Skala tersebut terdiri dari 10 item atau pernyataan dengan empat pilihan jawaban: 1 = sangat tidak setuju, 2 = tidak setuju, 3 = setuju, dan 4 = sangat setuju. Kumpulan data rse terdiri dari sampel acak 1000 responden yang menyelesaikan semua item pada instrumen. Karena pernyataan pada butir 3, 5, 8, 9, dan 10 memiliki kata-kata negatif (mis., Q3: Saya cenderung merasa bahwa saya gagal), tanggapan terhadap butir-butir ini diberi kode terbalik (mis., 1 = sangat setuju, 2 = setuju, 3 = tidak setuju, 4 = sangat tidak setuju). Dengan demikian, skor yang lebih tinggi untuk semua item menunjukkan harga

diri yang lebih besar. Rincian lebih lanjut tentang kumpulan data rse dapat ditemukan di *hemp package*.

Sebelum memulai analisis, pertama-tama kita mengaktifkan *hemp package* dan mirt (Chalmers, 2012a) menggunakan fungsi *library*.

```
library(“hemp”)
library(“mirt”)
```

Seperti yang telah kita lakukan untuk model IRT dikotomis di Chapter 6, dimulai dengan mendefinisikan sifat laten menggunakan 10 variabel pertama dalam kumpulan data rse (rse[, 1:10]), yang sesuai dengan item dalam skala harga diri Rosenberg. Selanjutnya, kita mendefinisikan dan menyimpan model sebagai pcm_mod dan memperkirakannya menggunakan fungsi mirt. Meskipun kami menetapkan itemtype = "Rasch", fungsi mirt mengenali bahwa item mungkin memiliki lebih dari dua kategori respons, dan dengan demikian cocok dengan PCM. Dalam kumpulan data rse, semua item memiliki empat kategori respons. Namun, seperti yang disebutkan sebelumnya, jumlah kategori respons dapat bervariasi antar item saat PCM digunakan. Setelah model diestimasi, kemudian diekstrak hasilnya menggunakan fungsi coef dan menyimpannya sebagai pcm_params.

```
pcm_mod <- "selfesteem = 1 - 10"
pcm_fit <- mirt(data = rse[, 1:10], model = pcm_mod,
                itemtype = "Rasch", SE = TRUE)
pcm_params <- coef(pcm_fit, IRTpars = TRUE, simplify = TRUE)
```

Seperti yang telah disebutkan di Chapter 6, fungsi coef menyimpan parameter item dan informasi tambahan sebagai daftar. Jadi, dapat memilih parameter item dari daftar ini, menyimpannya sebagai bingkai data baru yang disebut pcm_items, dan kemudian mencetak parameter item.

```
pcm_items <- as.data.frame(pcm_params$items)
pcm_items
```

	a <dbl>	b1 <dbl>	b2 <dbl>	b3 <dbl>
Q1	1	-2.862010	-1.6407355	0.9905954
Q2	1	-3.054737	-2.1650857	1.0886321
Q3	1	-2.221950	-0.5364370	1.7623392
Q4	1	-3.350431	-1.4546799	1.8055787
Q5	1	-1.849256	-0.0803302	1.4718476
Q6	1	-2.183106	-0.1655968	2.0228575
Q7	1	-1.783392	0.1374231	2.3139519
Q8	1	-1.581559	0.8743338	1.9704969
Q9	1	-1.235833	1.2172079	2.0237022
Q10	1	-1.399323	0.6126267	1.1601504

Gambar 7.2 Parameter Item

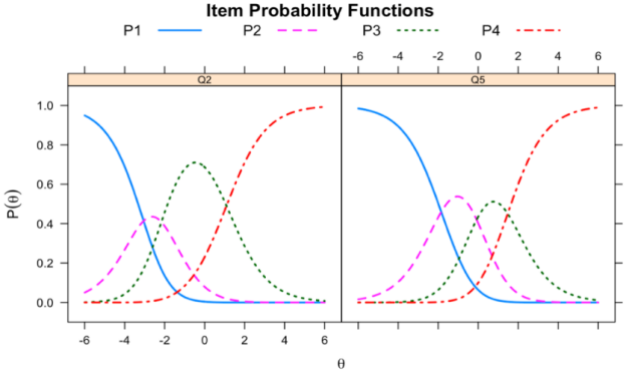
Pada *output*, nama item dalam kumpulan data rse dicetak sebagai nama baris. Kolom pertama menunjukkan parameter diskriminasi (a), yang sama dengan 1 untuk semua item karena PCM, mirip dengan model Rasch, membatasi parameter diskriminasi menjadi 1. Pada rangkaian kolom berikutnya (b1 hingga b3), kita melihat estimasi ambang (yaitu, langkah) parameter. Karena item dalam kumpulan data rse memiliki empat opsi respons, PCM memperkirakan tiga parameter ambang batas untuk setiap item. Kolom yang diberi label sebagai b1 hingga b3 pada output sesuai dengan parameter dalam Persamaan 7.1.

Selanjutnya, digunakan fungsi plot dalam paket mirt untuk memeriksa item secara visual. Tidak seperti model IRT dikotomi, model IRT politomi memiliki kurva karakteristik pilihan (*option characteristic curves* - OCC), yang dapat dianggap sebagai perpanjangan ICC untuk item politomi. Karena item memiliki lebih dari dua kategori respons, ada beberapa OCC yang diplot per item. Setiap kurva mewakili probabilitas memilih opsi respons tertentu sebagai fungsi dari sifat laten (θ).

```
plot(pcm_fit, type = "trace", which.items = c(2, 6),
     par.settings = simpleTheme(lty = 1:4, lwd = 2),
     auto.key = list(points = FALSE, lines = TRUE, columns =
4))
```

Saat memanggil fungsi plot, kita menyediakan model yang sesuai (pcm_fit) dan menentukan tipe plot yang ingin kita gunakan (type = "trace"). Untuk mendemonstrasikan bagaimana kita akan memplot OCC hanya untuk dua item, kita tentukan which.items = c(2, 6) untuk memplot Q2 dan Q6 dalam kumpulan data rse. Jika argumen ini dihapus

dari fungsi plot, secara default akan menggambar OCC untuk semua item dalam satu plot teralis. Selanjutnya, kita tentukan beberapa pengaturan tambahan untuk plot menggunakan argumen par.settings dan fungsi simpleTheme. Bisa menggunakan lty = 1:4 untuk meminta jenis garis yang berbeda untuk setiap OCC, dan untuk membuat kurva lebih tebal, kami menetapkan lwd = 2, yang berarti lebar garis. Terakhir, kami membuat legenda menggunakan auto.key. Legenda ini memiliki 4 kolom yang menunjukkan garis dan bukan titik untuk OCC kami. Pada Gambar 7.2, kategori respons diberi label sebagai P1 hingga P4. Untuk kedua item, OCC mengikuti urutan yang diharapkan sama seperti kategori respons. OCC untuk opsi respons pertama (P1) berada di paling kiri plot, sedangkan opsi respons terakhir (P4) terletak di paling kanan plot. OCC lain di tengah juga dipesan dengan benar.



Gambar 7.3 Kurva Karakteristik Opsi Untuk Item 2 dan 5 Untuk PCM.

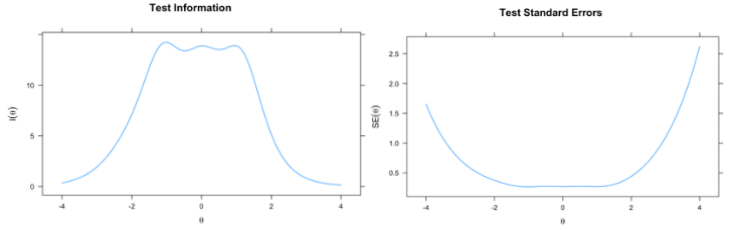
Selanjutnya, memplot IIF untuk menunjukkan jumlah informasi yang dijelaskan setiap item sebagai fungsi dari tingkat sifat laten mereka. Kami menggunakan fungsi plot lagi tetapi kali ini kami meminta IIF dengan mengatur type = "infotrace". Pada opsi par.settings, kita kembali mengubah lebar garis agar lebih terlihat. Gambar 7.4 menunjukkan plot IIF untuk item 2 dan 5 dalam kumpulan data rse.

```
plot(pcm_fit, type = "infotrace", which.items =
     c(2, 5),par.settings = simpleTheme(lwd = 2))
```

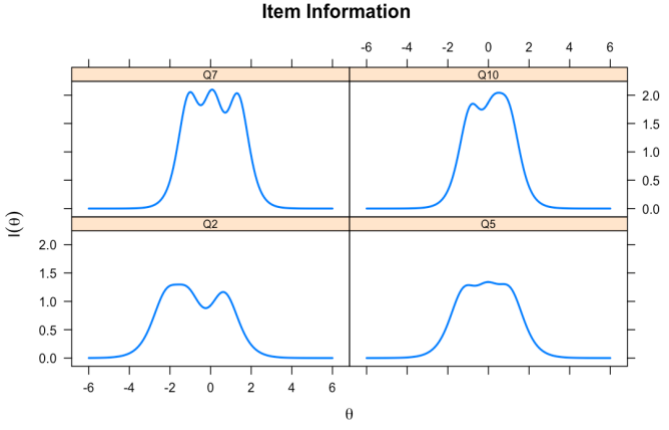
Selain IIF untuk item individual, kami juga dapat memplot jumlah total informasi (yaitu, *Test Information Function* — TIF) yang tersedia

dari item dan distribusi bersyarat dari *Standard Error Measurement* (cSEM) dalam data rse mengatur. Dalam contoh berikut, ditunjukkan cara memplot TIF dan cSEM pada rentang sifat laten -6 hingga 6. Gunakan type = "info" untuk plot TIF dan type = "SE" untuk plot cSEM.

```
plot(pcm_fit, type = "info", theta_lim = c(-4, 4))
plot(pcm_fit, type = "SE", theta_lim = c(-4, 4))
```



Gambar 7.4 Kurva Informasi Tes dan Standar Eror Tes



Gambar 7.5 Fungsi Informasi Item untuk Item 2, 5, 7, 10 Untuk PCM

7.1.2 Rating Scale Model (RSM)

Rating Scale Model (RSM; (Andrich, 1978)) adalah model IRT keluarga Rasch lainnya yang dapat menangani data respons butir yang dicetak secara politomi. RSM mensyaratkan bahwa semua item memiliki jumlah opsi, atau kategori yang sama, dan mengasumsikan bahwa parameter ambang batas yang berdekatan adalah berjarak sama, yaitu, berjarak sama, di semua item. Untuk item empat kategori yang

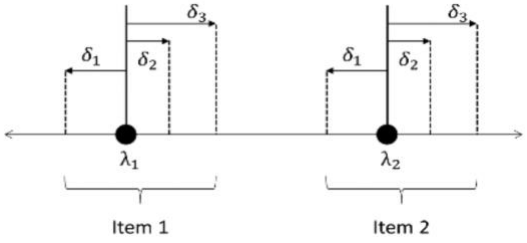
diberi skor atau nilai 0, 1, 2, dan 3, misalnya, jarak antara ambang pertama dan kedua sama di semua item, jarak antara ambang kedua dan ketiga sama di semua item , dll. Perhatikan, ini bukan batasan bahwa jarak antara kategori pertama dan kedua sama dengan jarak antara kategori kedua dan ketiga, dst., untuk satu item.

Rating Scale Model (Model Skala Penilaian) dapat dilihat sebagai bentuk terbatas dari PCM. Model skala penilaian dapat berfungsi dengan baik pada instrumen yang memiliki bentuk respon struktural yang sama atau format respons yang diasumsikan berfungsi dengan cara yang sama di semua item. Contoh umum dari model ini adalah instrumen yang mengandung skala Likert. Model skala penilaian mengasumsikan bahwa terdapat serangkaian ambang batas ordinal yang memisahkan kategori respons ordinal yang berdekatan satu sama lain. Setiap item mendapatkan parameter lokasi yang unik, tetapi perbedaan antara kategori respons di sekitar parameter lokasi dibatasi agar sama di semua item. Oleh karena itu, item dalam instrumen berbeda dalam hal lokasi keseluruhannya sementara penyebaran ambang kategori dalam item tetap sama.

Untuk Model Skala Penilaian, peluang terpilihnya kategori c (c = 0, 1, …, m) untuk butir i dapat ditulis sebagai:

$$(X_{ic}|\theta, \lambda_i, \delta_i, \dots, \delta_m) = \frac{\exp\left[\sum_{j=0}^{c}\left(\theta-(\lambda_i+\delta_j)\right)\right]}{\sum_{k=0}^{m_i}\exp\left[\sum_{j=0}^{c}\left(\theta-(\lambda_i+\delta_j)\right)\right]} \qquad (7.2)$$

Dimana λ_i adalah parameter lokasi untuk item ke-i dan $\delta_i, \dots, \delta_m$ adalah parameter ambang kategori. Perhatikan bahwa kategori item (yaitu c) tidak memiliki sebuah subskrip i, yang menyiaratkan bahwa semua item harus memiliki jumlah kategori yang sama, tidak seperti *Partial Credit Model*. Gambar 7.6 menunjukkan dua item skala penilaian pada kontinum sifat laten. Meskipun item 1 dan item 2 berbeda dalam hal lokasi keseluruhan (λ), penyebaran ambang kategori (δ) setara di kedua item.



Gambar 7.6 Dua Item Skala Penilaian Pada Kontinum Sifat Laten (Desjardins & Bulut, 2018)

Dalam contoh berikut, disesuaikan Model Skala Penilaian ke kumpulan data rse. Semua item dalam kumpulan data rse memiliki empat kategori, yang memungkinkan kami memperkirakan jumlah parameter ambang kategori yang sama untuk semua item. Kali ini ketika kami mendefinisikan model kami, kami menentukan itemtype = "rsm" untuk memperkirakan Model Skala Penilaian. Opsi tipe item = "rsm" membatasi parameter diskriminasi dan ambang kategori menjadi sama di semua item. Kami menyimpan hasil estimasi untuk Model Skala Penilaian sebagai rsm_fit, mengekstrak hasilnya menggunakan fungsi coef, kemudian menyimpan parameter item yang diperkirakan sebagai rsm_items, dan mencetaknya.

```
library("mirt")
rsm_mod <- "selfesteem = 1 - 10"
rsm_fit <- mirt(data = rse[,1:10], model = rsm_mod,
                itemtype = "rsm")
rsm_params <- coef(rsm_fit, simplify = TRUE)
rsm_items  <- as.data.frame(rsm_params$items)
rsm_items
```

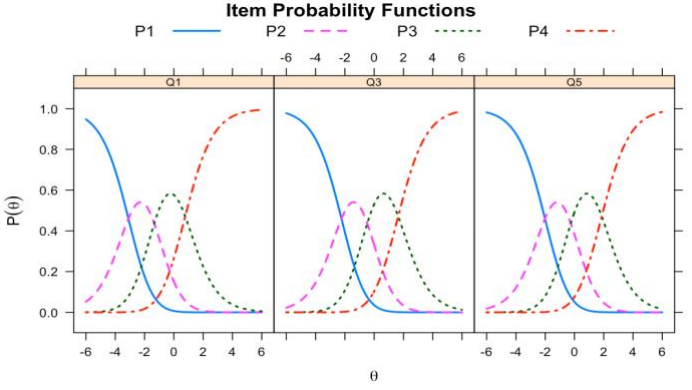
	a1 <dbl>	b1 <dbl>	b2 <dbl>	b3 <dbl>	c <dbl>
Q1	1	-3.101344	-1.339085	0.7753426	0.0000000
Q2	1	-3.101344	-1.339085	0.7753426	0.1813604
Q3	1	-3.101344	-1.339085	0.7753426	-0.8685444
Q4	1	-3.101344	-1.339085	0.7753426	-0.2943794
Q5	1	-3.101344	-1.339085	0.7753426	-1.0881627
Q6	1	-3.101344	-1.339085	0.7753426	-1.1163890
Q7	1	-3.101344	-1.339085	0.7753426	-1.4311156
Q8	1	-3.101344	-1.339085	0.7753426	-1.7382447
Q9	1	-3.101344	-1.339085	0.7753426	-2.0169977
Q10	1	-3.101344	-1.339085	0.7753426	-1.4522775

Gambar 7.7 Item Parameter

Pada output, a1 adalah parameter diskriminasi item, yang ditetapkan ke 1 untuk semua item karena RSM, sebagai bentuk politomi

dari model Rasch, mengharuskan semua item memiliki parameter diskriminasi yang sama. Kolom berikut (yaitu, d1 hingga d3) mewakili parameter ambang kategori, yang sama untuk semua item, dan c adalah parameter lokasi yang diperkirakan secara unik untuk setiap item dalam kumpulan data rse. Parameter lokasi yang diperkirakan menunjukkan bahwa item yang paling mudah adalah Q9 dan item yang paling sulit adalah Q2. Untuk melihat perbandingan visual dari kedua item ini, kami memplot OCC untuk item 2 dan 9. Kode untuk membuat plot ini sama seperti sebelumnya untuk PCM, kecuali jika ditentukan item mana yang akan diplot menggunakan which.items = c(1, 3, 5) sebagai gantinya.

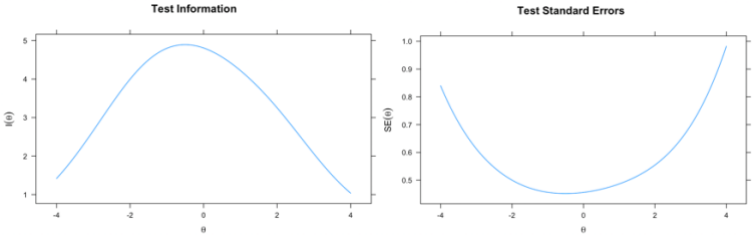
```
plot(rsm_fit, type = "trace", which.items =
     c(1, 3, 5), par.settings = simpleTheme(lty = 1:4,
     lwd = 2), auto.key = list(points = FALSE,
     lines = TRUE, columns = 4))
```



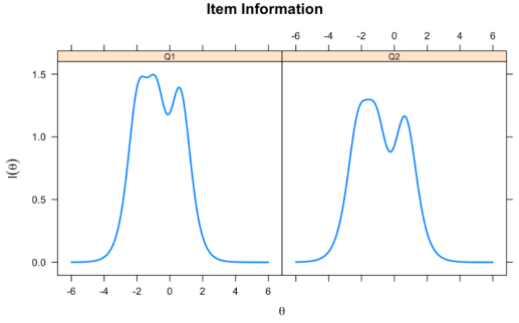
Gambar 7.8 Kurva Karakteristik Opsi Untuk Item 1, 3 dan 5 Untuk RSM.

Gambar 7.8 menunjukkan bahwa memilih kategori respons yang lebih tinggi (misalnya, 4 = sangat setuju atau 3 = setuju) untuk Q5 lebih mudah daripada memilih opsi respons yang sama untuk Q3. Kita juga dapat menggambar plot TIF dan cSEM untuk RSM dengan menentukan type = "info" untuk plot TIF dan type = "SE" untuk plot cSEM.

```
plot(rsm_fit, type = "info", theta_lim = c(-4, 4))
plot(rsm_fit, type = "SE", theta_lim = c(-4, 4))
```



Gambar 7.9 Kurva Informasi Tes dan Standar Eror Tes



Gambar 7.10 Fungsi Informasi Item untuk Item 1 dan 2 Untuk RSM

7.2 Model Non-Rasch Polytomous untuk Item Ordinal

Model kredit parsial umum (Muraki, 1992) dan model respons bertingkat (Samejima, 1969) dapat dilihat sebagai bentuk politomi dari model IRT dua parameter (2PL). Tidak seperti model Rasch polytomous, model kredit parsial umum dan model respons bertingkat mengasumsikan bahwa item bervariasi dalam hal tingkat diskriminasi mereka dalam membedakan peserta ujian atau responden dengan sifat laten rendah dan tinggi. Kedua model ini sedikit berbeda satu sama lain berdasarkan konseptualisasi kurva karakteristik opsi (De Ayala, 2013). Bagian berikut akan menjelaskan secara singkat model-model ini dan mendemonstrasikan bagaimana memperkirakannya menggunakan paket mirt (Chalmers, 2012a) dalam R.

7.2.1. Generalized Partial Credit Model (GPCM)

Model kredit parsial umum (GPCM; (Muraki, 1992) adalah perpanjangan dari PCM (dan karena itu perpanjangan dari RSM), di mana kemiringan item atau parameter diskriminasi diestimasi secara bebas. GPCM mirip dengan PCM mengenai bagaimana mengkonseptualisasikan kurva karakteristik opsi, sementara itu juga

mirip dengan model 2PL dalam parameter diskriminasi item yang dapat bervariasi di seluruh item. Alih-alih memperbaiki parameter diskriminasi item menjadi 1 untuk semua item, GPCM memperkirakan parameter diskriminasi item unik untuk setiap item. Menurut Muraki (Muraki, 1992), parameter diskriminasi item menunjukkan sejauh mana tanggapan kategoris bervariasi antara item sebagai perubahan sifat laten.

Probabilitas diperolehn dari GPCM poin X_{ik} (X_{ik} = 0, 1, . . . , mi) pada butir *i* untuk GPCM dapat ditulis dimana a_i adalah parameter diskriminasi untuk butir *i* dan suku sisanya sama dengan Persamaan 7.3. Mirip dengan PCM, ambang (δ_{ik}) tidak dibatasi dalam urutan yang sama dengan kategori respons.

$$(X_{ik}|\theta, a_i, \delta_{ik}) = \frac{\exp\left[\sum_{h=1}^{ik} a_i(\theta-\delta_{ih})\right]}{\sum_{k=0}^{m_i}\exp\left[\sum_{h=1}^{c} a_i(\theta-\delta_{ih})\right]} \quad (7.3)$$

Dalam contoh berikut, kami memperkirakan ambang kategori serta parameter diskriminasi item untuk item dalam kumpulan data rse menggunakan GPCM. Alih-alih itemtype = "Rasch", kali ini kita menggunakan itemtype = "gpcm" untuk memperkirakan parameter item berdasarkan GPCM.

```
library("mirt")
gpcm_mod <- "selfesteem = 1 - 10"
gpcm_fit <- mirt(data = rse[, 1:10], model = gpcm_mod,
                 itemtype = "gpcm", SE = TRUE)
gpcm_params <- coef(gpcm_fit, IRTpars = TRUE, simplify = TRUE)
gpcm_items <- gpcm_params$items
gpcm_items
```

```
Calculating information matrix...
           a         b1          b2        b3
Q1  1.8238891 -1.7202941 -0.95938961 0.5885070
Q2  1.7307657 -1.8564721 -1.28862013 0.6510270
Q3  1.8710897 -1.3231835 -0.31308686 1.0425363
Q4  1.2811061 -2.2085974 -1.00156932 1.2138474
Q5  1.4939839 -1.1502285 -0.04606415 0.9041157
Q6  2.7132868 -1.2057255 -0.09730049 1.1220286
Q7  2.3609630 -1.0210663  0.07245970 1.3106380
Q8  0.8945663 -1.1800257  0.81206980 1.2679593
Q9  1.4911761 -0.7707405  0.78235286 1.2329194
Q10 1.9005838 -0.8381950  0.33153091 0.7308140
```

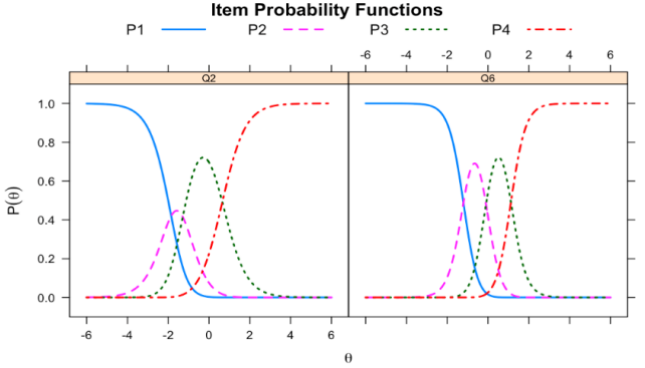
Gambar 7.11 Item Parameter

Berbeda dengan parameter item yang diestimasi untuk PCM, kolom pertama dari output dari GPCM (yaitu, diskriminasi item) tidak

tetap ke 1. Sebaliknya, setiap item memiliki parameter diskriminasi item yang unik. Output menunjukkan bahwa item dalam kumpulan data rse agak bervariasi dalam hal diskriminasi item. 3 kolom yang tersisa menunjukkan perkiraan parameter ambang kategori dari GPCM.

Untuk menunjukkan dampak dari berbagai parameter diskriminasi item di GPCM, kami memplot OCC untuk Q2 dan Q6. Tidak seperti pada Gambar 7.2, kita melihat bahwa item yang diestimasi dari GPCM tidak memiliki kemiringan yang sama (yaitu, diskriminasi). Gambar 7.11 menunjukkan bahwa kemiringan OCC untuk Q2 lebih curam daripada kemiringan OCC untuk Q6 sebagai akibat dari diskriminasi item yang lebih tinggi di Q8.

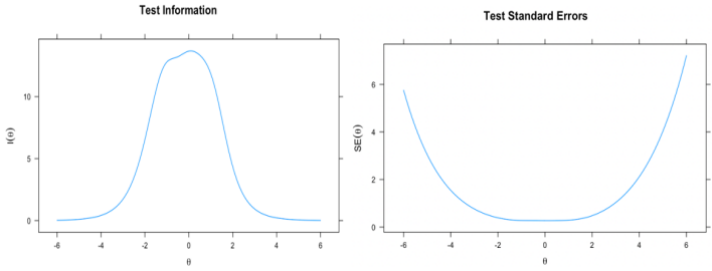
```
plot(gpcm_fit, type = "trace", which.items = c(2, 6),
     par.settings = simpleTheme(lty = 1:4, lwd = 2),
     auto.key=list(points = FALSE, lines = TRUE, columns = 4),
theta_lim = c(-4, 4))
```



Gambar 7.12 Kurva Karakteristik Opsi Untuk Item 2 dan 6 Untuk GPCM.

Seperti yang telah kami tunjukkan untuk PCM dan RSM, kami juga dapat menggambar plot TIF dan cSEM untuk GPCM dengan menentukan type = "info" untuk plot TIF dan type = "SE" untuk plot cSEM.

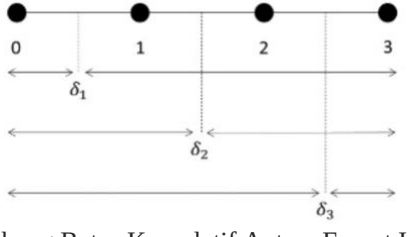
```
plot(gpcm_fit, type = "info", theta_lim = c(-4, 4))
plot(gpcm_fit, type = "SE", theta_lim = c(-4, 4))
```



Gambar 7.13 Kurva Informasi Tes dan Standar Eror Tes

7.2.2 Graded Response Model (GRM)

Model respons bertingkat (GRM; (Samejima, 1969)), awalnya diperkenalkan sebagai kasus homogen dari model respons bergradasi dalam arti bahwa bentuk fungsi respons kategori kumulatif adalah sama, dan tidak pernah saling bersilangan. Sebelumnya, PCM, RSM, dan GPCM memodelkan probabilitas merespons dalam kategori k dari item ke-i. GRM menggunakan respons kumulatif untuk memodelkan probabilitas merespons dalam kategori k atau lebih tinggi. GRM juga dikenal sebagai model logit kumulatif, adalah perpanjangan politomi dari model 2PL. GRM cocok untuk item dengan rangkaian respons dasar yang jelas. Model GRM probabilitas kategori respons yang diberikan atau lebih tinggi dengan mengikuti urutan yang sama dengan opsi respons. Dalam GRM, setiap kategori respons menyumbangkan beberapa informasi untuk kemungkinan seseorang memilih kategori respons tertentu. Untuk item dengan kategori respons berurutan K, GRM membuat item dikotomi K 1 dengan membagi kategori respons secara kumulatif. Masing-masing item dikotomi buatan ini memiliki parameter kesulitan yang unik tetapi memiliki parameter diskriminasi yang sama.



Gambar 7.14 Ambang Batas Kumulatif Antara Empat Kategori Respons (Desjardins & Bulut, 2018)

Misalnya, pertimbangkan item polytomous dengan empat kategori respons yang dipesan (X = 0, 1, 2, atau 3). Gambar 7.14 menunjukkan tiga ambang $\delta_1, \delta_2, \delta_3$ yang membagi empat kategori respons secara kumulatif. Artinya, setiap parameter ambang batas menunjukkan tingkat sifat laten yang diperlukan untuk memiliki peluang 50% untuk memilih kategori respons tertentu atau lebih tinggi.

Probabilitas perolehan poin X_i atau lebih $(X_i = 0, 1, \ldots, m_i)$ pada butir ke-*i* untuk GRM dapat ditulis sebagai berikut:

$$P^*\left(X_i \middle| \theta, a_i, \delta_{X_i}\right) = \frac{e^{a_i\left(\theta - \delta_{X_i}\right)}}{1 + e^{a_i\left(\theta - \delta_{X_i}\right)}} \quad (7.4)$$

di mana θ adalah sifat laten, a_i adalah parameter diskriminasi untuk item *i*, δ_{X_i} adalah lokasi batas kategori untuk kategori X_i (mirip dengan parameter ambang batas kategori pada model sebelumnya), dan $P^*\left(\theta, a_i, \delta_{X_i}\right)$ adalah probabilitas seseorang memperoleh skor Xi atau lebih tinggi (De Ayala, 2013). Seperti disebutkan sebelumnya, GRM membagi item polytomous menjadi serangkaian item dikotomis menggunakan probabilitas kumulatif. Artinya, item i terdiri dari item dikotomi mi yang memiliki parameter diskriminasi yang sama (a_i) tetapi memiliki parameter kesulitan yang unik (δ_{X_i}).

Dalam contoh berikut, kami memperkirakan parameter item untuk kumpulan data rse menggunakan GRM. Kami menggunakan itemtype = "graded" agar sesuai dengan GRM.

```
library("mirt")
grm_mod <- "selfesteem = 1-10"
grm_fit <- mirt(data = rse[, 1:10], model = grm_mod,
                itemtype = "graded", SE = TRUE)
grm_params <- coef(grm_fit, IRTpars = TRUE, simplify = TRUE)
```

Selanjutnya, kami menyimpan parameter item yang diperkirakan sebagai grm_items dan mencetaknya.

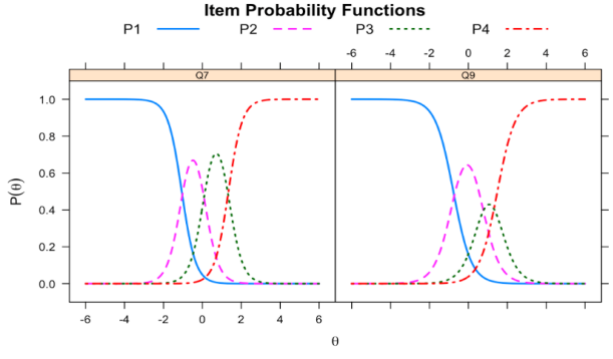
```
grm_items <- as.data.frame(grm_params$items)
grm_items
```

	a <dbl>	b1 <dbl>	b2 <dbl>	b3 <dbl>
Q1	2.324817	-1.9096256	-0.86459285	0.6212696
Q2	2.136179	-2.1468087	-1.15272898	0.6594750
Q3	2.435460	-1.3941348	-0.25807123	1.0778067
Q4	1.650816	-2.4179124	-0.86144982	1.2003060
Q5	2.172579	-1.2073811	-0.05750731	1.0257647
Q6	3.202812	-1.2249932	-0.08533037	1.1446882
Q7	2.810958	-1.0605179	0.09078729	1.3417869
Q8	1.420458	-1.2018548	0.51592382	1.7199115
Q9	2.163341	-0.7728069	0.63992017	1.4899442
Q10	2.645276	-0.8719370	0.23061101	0.9486129

Gambar 7.15 Parameter Item

Dalam output, a1 adalah parameter diskriminasi item dan kolom yang tersisa (yaitu, b1 hingga b3) mewakili lokasi batas kategori untuk item dalam kumpulan data rse. Mirip dengan output dari GPCM, setiap item memiliki parameter diskriminasi yang unik. Seperti GRPM, Q6 dan Q8 memiliki parameter diskriminasi tertinggi dan terendah dalam kumpulan data rse. Selanjutnya, menggambar plot OCC dan memeriksa kemiringan kurva karakteristik opsi untuk Q5 dan Q9, yang memiliki tingkat diskriminasi serupa tetapi berbeda berdasarkan lokasi batas kategorinya. Gambar 6.8 menunjukkan bahwa kemiringan OCC untuk Q5 dan Q9 sangat mirip, meskipun Q5 tampaknya lebih mudah daripada Q9.

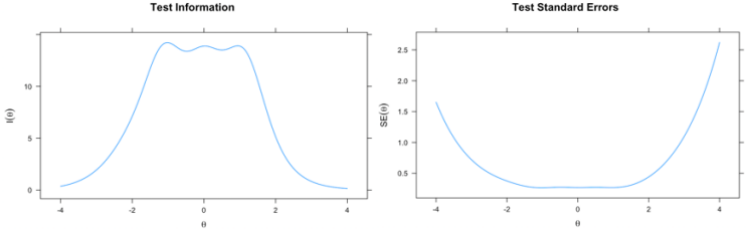
```
plot(grm_fit, type = "trace", which.items = c(5, 9),
     par.settings = simpleTheme(lty = 1:4,lwd = 2),
     auto.key = list(points = FALSE, lines = TRUE, columns =
4))
```



Gambar 7.16 Kurva Karakteristik Opsi Untuk Item 7 dan 9 Untuk GRM

Terakhir, seperti yang telah kita lakukan untuk model sebelumnya, kita dapat menggambar plot TIF dan cSEM untuk GRM dengan menggunakan type = "info" untuk plot TIF dan type = "SE" untuk plot cSEM.

```
plot(grm_fit, type = "info", theta_lim = c(-4, 4))
plot(grm_fit, type = "SE", theta_lim = c(-4, 4))
```



Gambar 7.17 Kurva Informasi Tes dan Standar Eror Tes

Selain GRM tradisional, fungsi mirt juga mampu mengestimasi model hybrid yang menggabungkan karakteristik tertentu dari RSM dan GRM menjadi satu model. Model ini disebut model respons bergradasi skala-peringkat dalam manual paket mirt. Untuk memperkirakan model hybrid ini, itemtype dalam fungsi mirt harus disetel ke itemtype = "grsm" atau itemtype = "grsmIRT". Rincian lebih lanjut tentang model respons berjenjang skala penilaian dapat ditemukan di Muraki (Muraki, 1992).

7.3 Model IRT Polytomous untuk Item Nominal

Pada bagian sebelumnya, telah dipelajari mengenai teori respon butir politomi dengan model butir ordinal. Berbeda halnya pada bagian ini, akan lebih memperdalam model politomi dengan data nominal. Secara khusus, kategori respons dari item politomi tidak memiliki hubungan ordinal, maka dari itu kategori respons yang diperoleh tidak dapat mengasumsikan data ordinal dari satu kategori respons ke kategori respons lainnya sebagai fungsi dari sifat laten. Jenis data terbagi menjadi dua, data diskret dan data kontinu. Data dengan jenis diskret terdiri dari data dengan skala nominal dan skala ordinal, sedangkan data dengan skala interval dan data rasio termasuk dalam data kontinu. Jenis data kontinu, mengarahkan pada asumsi bahwa data

terdistribusi secara normal, baik dilihat secara marginal maupun kondisional yang didasarkan dari seperangkat prediktor, kovariat, atau variabel independent. Selain itu, data dengan skala interval dan rasio, dapat menghitung statistik deskriptif seperti mean dan standar deviasi hingga menghasilkan plot pencar, plot batang, plot daun, serta histogram, hanya saja tidak sesuai dengan data berskala nominal dan ordinal (Desjardins & Bulut, 2018). Kecocokan data dengan skala nominal dan ordinal digunakan pada analisis data yang mengarah pada ketersediaan tabel kontingensi untuk menunjukkan jumlah atau proporsi responden yang mendukung respons tertentu, dan membuat diagram batang, plot titik, atau plot mosaik.

Model IRT politomi dengan data nominal secara langsung memperkirakan probabilitas memilih kategori respons tertentu terhadap semua kategori lain dalam item yang diberi skor nominal. Artinya, setiap item dibagi menjadi serangkaian item berdasarkan kategori nominal. Berdasarkan pengembangan model IRT politomi, model nominal diperoleh dari perluasan kasus Thurstone untuk pilihan biner yang mana dapat digeneralisasikan untuk pilihan pertama diantara tiga atau lebih alternatif pilihan (Thissen et al., 2012). Contohnya, dapat diasumsikan bahwa seorang peneliti memberikan survei kepada pelanggan perusahaan elektronik tentang produk teknologi dan hiburan. Salah satu pertanyaan dalam survei menanyakan kepada pelanggan produk mana yang memiliki kemungkinan besar akan mereka beli dalam enam bulan ke depan, misalnya laptop, smartphone maupun tablet. Jika peneliti menerapkan model IRT nominal untuk item ini, peneliti akan memiliki beberapa perbandingan biner untuk setiap produknya, seperti laptop versus smartphone atau tablet; smartphone versus laptop atau tablet; dan tablet versus laptop atau smartphone.

Namun demikian, model nominal digunakan secara luas dalam analisis butir soal dan penilaian tes. Model nominal dapat digunakan untuk tiga tujuan (Samejima, 2016), Pertama, sebagai analisis item dan metode penilaian untuk item yang memperoleh respons nominal murni. Kedua, untuk memberikan pemeriksaan empiris bahwa item yang diharapkan menghasilkan respons yang dipesan telah benar-benar melakukannya. Ketiga, untuk memberikan model tanggapan terhadap testlet. Testlet adalah set item yang dinilai sebagai satu unit yang

didalamnya terdiri dari beberapa item tes (Wainer et al., 2007). Hanya saja, seringkali kategori respons testlet yang terbentuk adalah pola respons terhadap item-item penyusunnya dan pola-pola itu jarang diurutkan. Nah, terdapat beberapa model IRT politomi dalam literatur untuk item nominal, bagian ini berfokus pada dua model, model respons nominal, dan model logit bersarang.

7.3.1 Nominal Response Model (NLM)

Untuk mempermudah dalam memperkuat pemahaman keterkaitan dengan IRT politomi lainnya, perlu diketahui bahwa RSM adalah submodel dari PCM, dimana PCM sendiri merupakan submodel dari GPCM. Sedangkan, RSM, PCM, dan GPCM merupakan submodel dari NRM dimana merupakan bentuk umum dari model yang dibagi dengan total. Model nominal untuk pertama kalinya dikembangkan oleh Bock pada tahun 1972 yang mengarah pada penggambaran secara umum untuk pertama kali teori respons butir dengan respons politomi. NRM atau *Nominal Response Model* juga dikenal dengan *Nominal Categories Model* (NCM) yang memiliki perbedaan dengan model politomi lainnya. Model respons nominal (NRM) cocok untuk item yang diberi skor politomi dengan kategori respons tertentu.

NRM menentukan probabilitas memilih salah satu kategori terhadap kategori yang tersisa, kecondongan pada pilihan tanpa ketentuan urutan yang telah diberikan. NRM memiliki respons jawaban yang tidak tersusun, meskipun hasil dari tanggapan sering dikodekan dengan numerik seperti 0,1,2,… . Sebenarnya, nilai-nilai pada tanggapan tidak mewakili semacam skor pada item, tetapi hanya indikasi nominal untuk kategori suatu respons saja. Selain itu, model ini juga bagian dari wujud tanggapan terhadap item pilihan ganda karena sulitnya menentukan urutan alternatif. Oleh karena itu, setiap kategori respons memiliki parameter diskriminasi dan kesulitannya sendiri. Ciri khas yang dimiliki oleh model ini adalah NRM dapat memodelkan data yang tidak berurutan dan NRM tidak dibangun pada konsep dikotomisasi yang menghasilkan Batasan kategori sehingga membentuk dasar model bertingkat. Sebenarnya pula, secara khsusu NRM merupakan perpanjangan model multinomial, efek tetap dan logit linier yang bercampur dengan IRT.

Sebelum mengarah pada bagaimana membedakan NRM dengan model IRT politomi lainnya, penting diketahui model matematisnya terlebih dahulu sebagai pijakan awal untuk memperdalam bagaimana cara kerja NRM. Peluang terpilihnya kategori respons ke-*k* (k = 0, 1, … , m) untuk butir *i* pada NRM (Desjardins & Bulut, 2018), dapat dituliskan :

$$P(X_{ik}|\theta, \boldsymbol{a}, \gamma) = \frac{e^{\gamma_{ik}+a_{ik}\theta}}{\sum_{k=1}^{m} e^{\gamma_{ik}+a_{ik}\theta}} \quad (7.5)$$

Keterangan :

a : vektor parameter butir untuk diskriminan hingga butir ke-i ($a = [a_{i1}; a_{i2}; \dots; a_{im}]$)

γ : vektor parameter butir untuk diskriminan hingga butir ke-i ($\gamma = [\gamma_{i1}; \gamma_{i2}; \dots; \gamma_{im}]$)

m : jumlah kategori respons pada item *i*

Pada model ini, kategori dengan estimasi kemiringan terbesar untuk suatu item akan membentuk kurva karakteristik kategori atau dikenal dengan *Category Characteristic Curve* (CCC) yang meningkat secara monoton di sepanjang sifat laten atau skala θ, dan sebaliknya. Dalam NRM, hanya mewakili parameter intersep dari persamaan linier. Baker (1992) menunjukkan perlu adanya transformasi mengenai bentuk linier dari persamaan dalam parameter IRT baik a maupun b. Pada kategori dengan kemiringan terkecil untuk suatu item selalu memiliki CCC yang menurun secara monoton di sepanjang skala θ.

Kategori dengan perkiraan nilai kemiringan antara yang terbesar dan terkecil mengambil bentuk kurva unimodal atau berbentuk lonceng, atau secara monoton meningkat (atau menurun) bentuk sepanjang skala θ. Parameter intersep (c_{ik}) pada NRM dapat dimaknai sebagai daya tarik dari kategori ke-*k* yang mana besarannya pada a_{ik} berguna untuk menginterpretasikan urutan kategori, misalnya $a_{i0} = 1, a_{i1} = 0, a_{i2} = -1$. Hal tersebut menunjukkan adanya perbedaan kategori, dimana untuk $k = 2$ termasuk kategori rendah, $k = 1$ termasuk kategori sedang atau tengan, dan $k = 0$ termasuk kategori tinggi. Namun, jika pengurutan kategori sama, $a_{ik} = a_{i(k-1)}$ tetapi $c_{ik} > c_{i(k-1)}$, maka kategori k pada CCC akan lebih tinggi daripasda k-1.

Pada model ini pula, terbagi menjadi dua batasan model mengenai parameter item untuk mempermudah memastikan identifiksi model. Bagian pertama adalah jumlah parameter diskriminan item sama dengan nol ($\sum_{k=1}^{m} a_{ik} = 0$) dan pada bagian kedua adalah jumlah dari perbandingan parameter kemiringan dan intersep item sama dengan nol $\sum_{k=1}^{m} \gamma_{ik} = 0$. Nah, dalam hal ini, sangat perlu diperhatikan bahwa parameter tingkat kesulitan item pada NRM tidak memiliki arti yang sama dengan parameter tingkat kesulitan item pada model IRT dikotomus 3PL maupun dengan model IRT politomi lainnya untuk item pada data ordinal. Mengapa demikian?

Parameter kesulitan item pada NRM memiliki kecenderungan respons untuk memilih kategori respons tertentu, contohnya seseorang akan lebih memilik smartphone daripada tablet maupun laptop. Semakin besar nilai a_k maka semakin kuat kategori respons *k* pada sifat laten yang diukur. Contoh mudahnya adalah konteks soal pilihan ganda dengan beberapa pilihan jawaban, misalnya saja A, B, C, D, dan E. Pilihan jawaban yang benar diharapkan memiliki parameter diskriminan positif dan tinggi, karena hal tersebut erat kaitannya dengan sifat laten yang dimiliki oleh item soalnya. Sedangkan parameter tingkat kesulitan item pada NRM yang dimaksud lebih pada rasio parameter kemiringan dan intersep yang ada pada CCC hasil analisis dari NRM.

Untuk mendemonstrasikan dan menunjukkan proses perhitungan estimasi model IRT polytomous dengan data nominal di R, berikut adalah contoh analisis yang menggunakan data hasil pengujian pada karakter siswa kelas V Sekolah Dasar dengan jumlah responden sebanyak 142 siswa. Terdapat 7 butir pilar karakter yang terdiri dari menghargai (*respect*), kejujuran (*honesty*), kerjasama (*cooperation*), tanggung jawab (*responsibility)*, kerendahan hati (*humility*), toleransi (*tolerance*), dan kasih sayang (*love*). Semua item diukur pada skala empat poin dengan kategori respons "1=sangat tidak setuju", "2=tidak setuju", "3=setuju" dan "4=sangat setuju". Berikut adalah hasil dari demonstrasi yang dilakukan menggunakan R dan syntax yang disusun sesuai dengan model respons nominal. Data yang digunakan dalam contoh analisis ini diasumsikan memiliki kecocokan model yang fit.

```
# data = data, panggil data dengan file Latihan_NRM.csv dalam
Link bernama data yang dibaca melalui perintah read.csv
dikarenakan format yang digunakan csv, dapat menyesuaikan
format file.
```

```
library(mirt)
# Baca data
data_NLM <- read.csv(sprintf(
"https://docs.google.com/uc?id=%s&export=download",
"1JWAO0ANIDCFB9Q1qo4Am35mGW7nv3O1i"),header=T,sep=";")
data
```

Penggunaan paket (mirt) difungsikan sebagai alat bantu pada pemanggilan data set yang akan digunakan dalam analisis butir NRM untuk hasil yang lebih stabil dibanding dengan paket lainnya (Paek & Cole, 2020). Selain menggunakan (mirt), dapat digunakan pula menggunakan paket (ltm), paket (eRm) dan paket MCMCpack. Ketiganya memiliki perbedaan masing-masing fungsi. Paket (ltm) dan paket (eRm) lebih dibatasi pada analisis model IRT dengan unidimensional secara efektif, sedangkan paket MCMCpack masih harus memerlukan pendukung estimasi Bayesian dan hanya untuk set respons butir dikotomus saja.

Sebelum melakukan analisis model IRT politomi dengan NRM, maka haruslah melakukan pemanggilan/baca data terlebih dahulu dan yakinkan bahwa paket yang akan digunakan telah tersedia dalam program R. Dalam hal ini menggunakan data yang berada dalam panggilan Link sedangkan penginstall-an paket dapat dilakukan dengan menggunakan install.packages("mirt"). Data dipanggil dengan perintah read.csv karena menyesuaikan file/berkas data yang ada, dan didukung dengan pembatas isian data didalamnya yakni tanda titik koma menggunakan perintah sep = ';'. Perlunya tambahan pengetahuan mengenai pemanggilan pada setiap langkah pada R.

data = data[, 1:7] ; data yang dipanggil Bernama "data" pada tahap sebelumnya pada saat pemanggilan data awal yang dipilih pada keseluruhan baris tetapi kolom khusus 1 hingga 7
model = nrm_mod; panggilan model nominal tentang agression yang dipanggil dalam data melalui 7 item

```
# itemtype = "nominal"; fungsi yang digunakan dalam
mengestimasi parameter item untuk model 2PL-NLM
# SE = TRUE; bermakna estimasi perhitungan kesalahan standar
pengukuran (standart error)
```

```
# Model Respon Nominal
# Menyeleksi data butir
nrm_mod <- "agression = 1 - 7"
nrm_fit <- mirt(data = data[, 1:7], model = nrm_mod,
                itemtype = "nominal", SE = TRUE)
nrm_params <- coef(nrm_fit, IRTpars = TRUE, simplify = TRUE)
nrm_items  <- as.data.frame(nrm_params$items)
nrm_items
```

Tabel 7.1 Output Respons Jawaban

	a1	**a2**	**a3**	**a4**
Respect	-5,1211180	1.6754894	1.4094236	2.0362049
Honesty	0.1561847	-0.9057709	-0.1681674	0.9177536
Cooperation	0.1918181	-0.1789653	-0.2575828	0.2447299
Humality	-0.1016822	-0.2544368	-0.2930203	0.6491393
Love	0.9174769	-1.7919888	-0.4660169	1.3405288
Responsibility	1,8137136	-2.5273797	-1.3263817	2.0400478
Tolerance	-0.1592274	-0.2622966	-0.3847473	0.8062713
	c1	**c2**	**c3**	**c4**
Respect	-10.2552775	2.38552791	4.5868023	3.28294725
Honesty	-0.8013619	-0.06196494	0.5289507	0.33437615
Cooperation	-0.6571438	0.25000386	0.8091115	-0.40197152
Humality	-1.2539825	0.11748049	1.0955333	0.04096877
Love	-1.3879751	-0.08990669	1.1216397	0.35624205
Responsibility	-1.7926513	-0.74655969	1.4853470	1.05386402
Tolerance	-0.9715030	0.29970974	0.8811017	-0.20930839

Pada output dari respons jawaban, menunjukkan a1, a2, a3 dan a4 merepresentasikan parameter daya beda dari keempat jawaban dari yang sangat tidak setuju, tidak setuju, setuju, hingga sangat setuju. Perolehan keempat respons yang tertinggi adalah a4 atau sangat setuju untuk semua item. Inilah temuan yang diharapkan pada pengukuran karakter siswa SD kelas V yang menunjukkan agresi yang lebih tinggi. Hal tersebut dapat diartikan bahwa keempat pilihan jawaban lebih mampu mengidentifikasi respons dari respons yang rendah hingga tinggi. Padahal keseluruhan parameter daya beda atau diskriminan tersebut dibatasi oleh nol.

Pada tahap selanjutnya, masuk pada analisis model respons nominal yang mana perintahnya menggunakan sum_constraints yang memiliki fungsi perintah untuk menjumlahkan parameter daya beda dan tingkat kesukaran item yang terbingkai dalam suatu syntax sum_constraints.

discrimination = round(rowSums(nrm_items[, 1:7]), 3) → memiliki makna dimana daya beda digabungkan dengan penjumlahan dari butir 1 hingga butir 7 yang tersusun dalam bingkai sum_constraints 3 desimal.
difficulty = round(rowSums(nrm_items[, 1:7]), 3)) → memiliki makna dimana tingkat kesukaran digabungkan dengan penjumlahan dari butir 1 hingga butir 7 yang tersusun dalam bingkai sum_constraints 3 desimal.

```
# Penggabungan Parameter
sum_constraints <- data.frame(discrimination =
                   round(rowSums(nrm_items[,1:7]),3),
                  difficulty=round(rowSums(nrm_items
                   [, 1:7]), 3))
sum_constraints
```

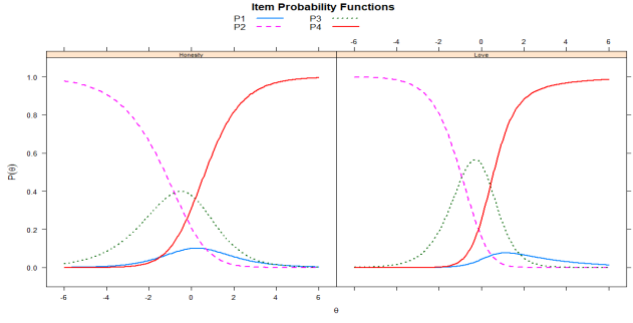
Tabel 7.2 Output Parameter Discrimination dan Difficulty

	discrimination	**difficulty**
Respect	-3.283	-3.283
Honesty	-0.334	-0.334
Cooperation	0.402	0.402
Humality	-0.041	-0.041
Love	-0.356	-0.356
Responsibility	-1.054	-1.054
Tolerance	0.209	0.209

Hasil analisis diatas, dapat dilihat apabila keseluruhan dari setiap parameter dijumlahkan menghasilkan nol sebagai hasil dari parameterisasi dari NRM. Masing masing parameter bergerak dalam jumlah yang berbeda beda. Tahap selanjutnya mengarah pada penggambaran plot.

type = "trace" ; pemanggilan jejak analisis sebelumnya yang dapat digunakan sebagai pendukung pembuatan plot pada tahap ini
which.items = c(2,5); pemanggilan butir yang ingin dianalisis dan dimunculkan plot yang diinginkan, dalam hal ini butir 2 dan 5.
auto.key = list(points = FALSE, lines = TRUE, columns = 3)) → serangkaian set syntax default yang dijadikan sebagai kunci pengaturan pada label teks atau parameter grafis, dalam hal ini columns = 3 diartikan bahwa kolom dibagi menjadi 3
par.settings = simpleTheme(lty = 1:3, lwd = 2) → membantu mengubah setting dari auto.key sesuai yang diinginkan

```
# Menggambar Plot
 plot(nrm_fit, type = "trace", which.items = c(2,5),
     par.settings = simpleTheme(lty = 1:3, lwd = 2),
     auto.key = list(points = FALSE, lines = TRUE, columns =
3)
```

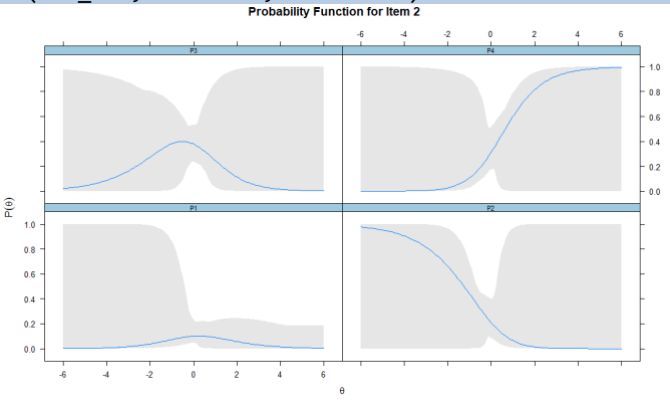


Gambar 7.18 Kurva karakteristik pada item pilihan yakni item Honesty dan Love dengan analisis model NRM

Pada respons P2 dan P4 menunjukkan agresi respons yang sama sama seimbang untuk kedua item, hanya saja pada P3 la yang lebih terlihat berbeda. Dimungkinkan pada agresi P3 responden membutuhkan agresi yang relative lebih tinggi pada item Honesty (2) daripada item Love (5). Selain itu, dapat pula kita melihat hasil plot untuk memastikan bahwa pilihan pada respons tertentu secara terpisah. Melalui gambar karakteristik item dibawah ini, secara detail diperoleh kurva setiap tingkatan respons dari P1, P2, P3 dan P4.

```
# Menggambar Plot
```

```
itemplot(nrm_fit, item = 2, CE = TRUE)
```



Gambar 7.19 Kurva karakteristik pada item Honesty dengan analisis model NRM

7.3.2 Nested Logit Model (NLM)

Model logit bersarang sangat cocok untuk tipe soal pilihan ganda dimana respons benar dan salah dimodelkan secara terpisah. Oleh karenanya, dapat dikatakan bahwa *Nested Logit Model* (NLM) dapat dianggap sebagai varian dari NRM yang sebenarnya gabungan dari model IRT dikotomus dengn NRM itu sendiri sehingga menghasilkan NLM yang tergolong pada IRT politomi. Namun berbeda dengan NRM, NLM ini mampu menyuguhkan potensi informasi melalui pengecoh pada item. Utama dalam analisis NLM adalah lebih pada pemanfaatan dan pengabaian informasi dari pengecoh item saat memodelkan respons benar dari pilihan (Suh Bolt 2010). Secara detail, NLM memiliki dua tipe level yang berbeda. Level 1, NLM lebih menggambarkan mengenai pilihan respons yang benar berbanding dengan pengecohnya. Lebel 2, mengarah pada pemodelan peluang yang terkait dengan pengecoh yang berlaku pada setiap item.

Ada beberapa varian atau bentuk NLM, hanya saja yang sering dijumpai adalah 3PL dengan versi NLM. Model ini memiliki special kasus mengenai adanya kendala tambahan pada diskriminan dan parameter pengecoh. Tipe 3PL untuk NLM secara matematis diwujudkan dengan formula sebagai berikut.

$$P(X_i = 0, D_{iv} = 1|\theta) = P(X_i = 0|\theta)P(D_{iv} = 1|X_i = 0, \theta)$$
$$= \left\{1 - \left[c_i + (1 - c_i)\frac{1}{1+e^{-(b_i+a_i\theta)}}\right]\right\}\left[\frac{e^{Z_{iv}(\theta)}}{\sum_{k=1}^{m} e^{Z_{ik}(\theta)}}\right] \qquad (7.6)$$

Dimana $P(\theta)$ adalah peluang peserta tes mampu menjawab item tes ke-*i* dengan jawaban salah dan memilih pengecoh kategori *v* dengan sifat laten, c_i adalah parameter tebakan butir *i*, b_i adalah parameter kesulitan butir *i*, a_i adalah parameter diskriminasi untuk butir *I*, sedangkan $Z_{ik}(\theta) = \gamma_{iv} + a_{iv}\theta$ menjadi pembilang dalam persamaan diatas.

Selain adanya 3PL NLM, ada pula 2 PL NLM yang dapat didefinisikan sebagai bentuk khusus dari 3PL NLM dimana untuk parameter tebakan semu pad item *i* (c_i) dibatasi menjadi 0. Sehingga, memunculkan formula baru untuk 2PL NLM sebagai berikut.

$$P(X_i = 0, D_{iv} = 1|\theta) = \left\{1 - \left[\frac{1}{1+e^{-(b_i+a_i\theta)}}\right]\right\}\left[\frac{e^{Z_{iv}(\theta)}}{\sum_{k=1}^{m} e^{Z_{ik}(\theta)}}\right] \qquad (7.7)$$

Berikut mendemonstrasikan cara menyesuaikan 2PL-NLM dan 3PL-NLM dengan menggunakan kumpulan data pilihan ganda yang berisi tanggapan item dari tes hipotetis yang terdiri dari 7 item pilihan ganda yang diberikan kepada 142 peserta ujian. Setiap item memiliki empat opsi respons (A, B, C, dan D) tetapi opsi respons diwakili secara numerik dalam kumpulan data (yaitu, A = 1; B = 2; C = 3; dan D = 4). Dalam contoh berikut, kami mempertimbangkan empat opsi respons sebagai kategori nominal, dan dengan demikian memberikan kunci jawaban untuk menentukan opsi respons yang benar untuk item tersebut.

Pada pengaplikasian NLM menggunakan R, akan dicoba dengan 2 model, yang pertama akan dicoba menggunakan itemtype = "2PLNRM" dalam fungsi mirt untuk memilih 2PL-NLM untuk memperkirakan parameter item, dan juga akan dicoba dengan itemtype = "3PLNRM", yang kemudian dapat dilihat perbandingan keduanya. Hasil analisis dapat diekstrak berupa parameter item yang diperkirakan, dapat pula menyimpannya ke twoplnlm_items, dan kemudian mencetak enam baris pertama menggunakan fungsi head. Data yang dipakai dalam analisis NLM ini sama dengan data yang digunakan analisis NRM, yakni menggunakan data Latihan_NLM.csv (yang berada dalam link) dimana hasilnya harus diawali dengan penentuan kunci jawaban dalam syntax. Sebelum melakukan analisis, perlu pemanggilan data terlebih dahulu, sama halnya analisis di bagian sebelumnya mengenai

NRM. Data yang digunakan dalam contoh analisis ini diasumsikan memiliki kecocokan model yang fit.

```
library(mirt)
# Baca data
data_NLM <- read.csv(sprintf(
"https://docs.google.com/uc?id=%s&export=download",
"1JWAO0ANIDCFB9Q1qo4Am35mGW7nv3O1i"),header=T,sep=";")
data_NLM
```

Melalui pemanggilan data dengan syntax diatas, menghasilkan outpur R seperti berikut. Hal ini harus dilakukan untuk memastikan data yang digunakan sesuai dengan keinginan dan tujuan analisis NLM, sehingga analisis dapat dilanjutkan hingga tahap akhir.

data = data, panggilan data dengan file Latihan_NRM.csv Bernama data yang dibaca melalui perintah read.csv dikarenakan format yang digunakan csv, dapat menyesuaikan format file.

Tahap selanjutnya adalah analisis menggunakan dengan itemtype = "2PLNRM", dan mempersiapkan kunci jawaban untuk 7 butir yang telah direspon oleh 142 responden. Pemanggilan dilakukan disesuaikan dengan panggilan data yang digunakan. Perlunya tambahan pengetahuan mengenai pemanggilan pada setiap langkah pada R.

data = data; data yang dipanggil Bernama “data” pada tahap sebelumnya pada saat pemanggilan data awal

model = twoplnlm_mod; panggilan twoplnlm_mod tentang ability yang dipanggil dalam data melalui 7 item yang telah terseleksi dengan kunci jawaban / key

itemtype = "2PLNRM"; fungsi yang digunakan dalam mengestimasi parameter item untuk model 2PL-NLM

SE = TRUE;bermakna estimasi perhitungan kesalahan standar pengukuran (standart error)

key = key; penggunaan kunci jawaban dengan panggilan “key”

```
# Model Logit Bersarang 2PL NLM
#menyeleksi data butir
key = c(3, 1, 3, 2, 1, 4, 1)
twoplnlm_mod <- "ability = 1 - 7"
twoplnlm_fit <- mirt(data = data_NLM,
                     model = twoplnlm_mod,
                     itemtype = "2PLNRM",
```

```
                        SE = TRUE, key = key)
twoplnlm_params <- coef(twoplnlm_fit, IRTpars = TRUE,
                        simplify = TRUE)
twoplnlm_params
twoplnlm_items <- as.data.frame(twoplnlm_params$items)
```

Tabel 7.3 Output Parameter Item

	a	b	g	u	a1	a2
Respect	-0.423	2.051	0	1	-6.244	2.975
Honesty	-0.010	-234.172	0	1	-0.928	-0.139
Cooperation	-0.276	-0.416	0	1	0.078	-0.270
Humality	-0.108	-13.002	0	1	-0.230	-0.377
Love	0.557	5.855	0	1	-1.634	-0.123
Responsibility	2.477	0.222	0	1	5.006	-3.135
Tolerance	0.078	33.072	0	1	-0.322	-0.457

	a3	c1	c2	c3
Respect	3.269	-8.606	3.851	4.756
Honesty	1.067	-0.327	0.280	0.047
Cooperation	0.191	-0.380	0.521	-0.142
Humality	0.607	-1.217	1.132	0.085
Love	1.757	-0.605	0.711	-0.106
Responsibility	-1.871	-3.252	0.486	2.766
Tolerance	0.780	-0.023	0.556	-0.533

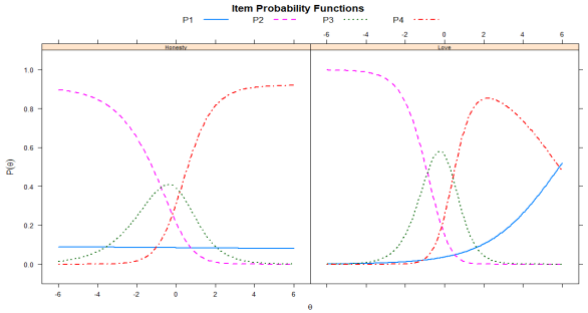
Dalam output, ditunjukkan terdapat a, b, g, dan u dimana keempatnya meliputi tingkat kesulitan, daya beda, tebakan semu (asimtotik bawah), dan kecerobohan (asimtotik atas) seperti yang telah dijelaskan sebelumnya. Lalu pada kolom selanjutnya ditunjukkan hasil a1, a2, a3, c1, c2, dan c3 yang berarti menunjukkan parameter NRM untuk ketiga pengecoh pada item pilihan ganda yang disajikan pada perangkat tes. Lalu mengapa nilai g dan u berada pada nilai g = 0 dan u = 1? Hal tersebut disebabkan berlakunya model 2PL yang berbasis NRM dimana model ini tidak melibarkan tebakan semu dan parameter kecerobohan pada analisis ini.

type = "trace"; pemanggilan jejak analisis sebelumnya yang dapat digunakan sebagai pendukung pembuatan plot pada tahap ini

which.items = c(1, 7); pemanggilan butir yang ingin dianalisis dan dimunculkan plot yang diinginkan, dalam hal ini butir 1 dan 7.

auto.key = list(points = FALSE, lines = TRUE, columns = 4)); serangkaian set syntax default yang dijadikan sebagai kunci pengaturan pada label teks atau parameter grafis, dalam hal ini columns = 4 diartikan bahwa kolom akan dibagi menjadi 4
par.settings = simpleTheme(lty = 1:4, lwd = 2); membantu mengubah setting dari auto.key sesuai yang diinginkan

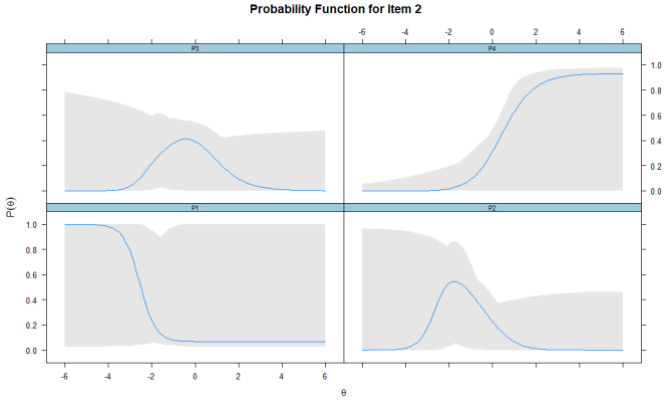
```
plot(twoplnlm_fit, type = "trace", which.items =
     c(2,5), par.settings = simpleTheme(lty = 1:4,
     lwd = 2), auto.key = list(points = FALSE,
     lines = TRUE, columns = 4))
```



Gambar 7.20 Kurva Karakteristik pada item pilihan yakni item 2 dan dengan analisis model 2PL NLM

Berdasarkan gambar diperoleh estimasi parameter dari 2PL NLM dimana terdapat kemiringan positif pada P4 kedua butir karena respons benar ditunjukkan pada garis tersebut. Pada kedua kurva, parameter tebakan dimulai dari 0 pada 2PL NLM sehingga kategori akan dimulai pada kisaran $P(\theta) = 0$. Temuan lainnya, diantara kedua butir tersebut untuk distractor yang paling masuk akal menurut responden kemampuan rendah ditunjukkan pada P4 atau pilihan D untuk item 2 dan pada P1 untuk pilihan A item 5. Selain itu, pada item 5, ada kemungkinan yang sangat tinggi untuk responden dengan kemampuan rendah memilih respons P1 dibanding pada item ke 1, yang mana pada kedua item untuk P1 merupakan pengecoh. Nah, mengingat pada butir 2 untuk P1 hampir menunjukkan kurva datar, maka dapat dikatakan bahwa P1 tersebut adalah pengecoh yang paling tidak masuk akal. Mengapa? Karena responden dengan kemampuan rendah ataupun tinggi memiliki pilihan jawaban yang sama yakni tidak mengarah pada

disktraktor tersebut. Melalui gambar karakteristik item dibawah ini, secara detail diperoleh kurva setiap tingkatan respons dari P1, P2, P3 dan P4.



Gambar 7.21 Kurva karakteristik pada item Honesty dengan model 2PL NLM

```
# Model Logit Bersarang 3PL NLM
threeplnlm_mod <- "ability = 1 - 7"
threeplnlm_fit <- mirt(data = data_NLM,
                  model = threeplnlm_mod, itemtype =
                  "3PLNRM", SE = TRUE, key = key)
threeplnlm_fit
threeplnlm_params <- coef(threeplnlm_fit, IRTpars =
                     TRUE, simplify = TRUE)
threeplnlm_params
threeplnlm_items <- as.data.frame
                    (threeplnlm_params$items)
```

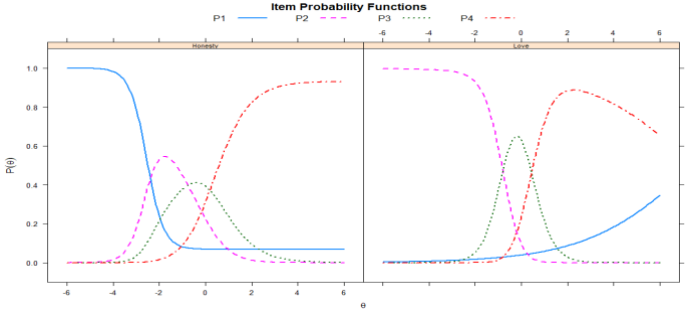
Sama halnya pada tahap sebelumnya 2PL NLM, pada analisis model ini dilakukan dengan langkah yang sama. Hanya aja pada tahapan ini berbedaannya pada itemtype = "3PLNRM". Hasil analisis dapat dilihat seperti berikut.

Tabel 7.4 Output Parameter Item

	a	b	g	u	a1
Respect	-0.337	1.756	0.159	1.000	-4.929
Honesty	-2.723	-2.554	0.068	1.000	-0.884
Cooperation	-1.980	-1.637	0.406	1.000	0.090
Humality	-1.059	-4.077	0.179	1.000	-0.277
Love	0.428	7.486	0.000	1.000	-2.312
Responsibility	3.246	0.330	0.070	1.000	4.621
Tolerance	-10.176	-2.317	0.059	1.000	-0.225

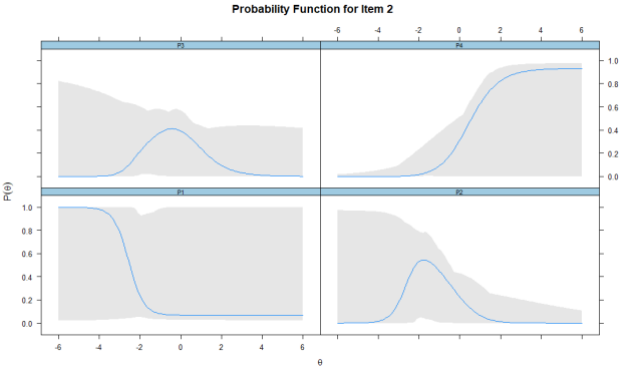
	a2	a3	c1	c2	c3
Respect	2.246	2.682	-7.130	3.115	4.014
Honesty	-0.156	1.040	-0.293	0.261	0.031
Cooperation	-0.199	0.109	-0.390	0.513	-0.123
Humality	-0.286	0.563	-1.244	1.126	0.118
Love	0.058	2.254	-0.920	0.957	-0.037
Responsibility	-3.107	-1.514	-2.412	-0.045	2.457
Tolerance	-0.468	0.693	-0.033	0.536	-0.503

```
plot(threeplnlm_fit, type = "trace", which.items=
     c(2,5), par.settings = simpleTheme(lty = 1:4,
     lwd = 2), auto.key = list(points = FALSE,
     lines=TRUE, columns=4))
```



Gambar 7.22 Kurva Karakteristik pada item pilihan yakni item 2 dan 5 dengan analisis model 3PL NLM

Pada output 3PL NM, menunjukkan adanya g sebagai estimassi parameter tebakan semu pada item yang berbeda dengan 2PL NLM, dimana pada 3PL NLM terdapat koefisien pada masing-masing itemnya. Hasilnya bahwa tebakan semu kurang berfungsi dimana koefisien g pada setiap item menunjukkan hasil yang sangat kecil. Oleh karenanya, setelah dibandingkan dengan 2PL NLM, maka perkiraan kesulitan item dan parameter diskriminan yang dibantu dengan tebakan semu lebih berarti dan lebih medetail dalam mengukur peluang yang terjadi pada responden saat menjawab soalan. Melalui gambar karakteristik item dibawah ini, secara detail diperoleh kurva setiap tingkatan respons dari P1, P2, P3 dan P4.



Gambar 7.23 Kurva karakteristik pada item Honesty dengan model 3PL NLM

Pada bagian ini telah diperkenalkan model IRT politomi dengan pemenuhan asumsi unidimensi diperoleh beberapa model analisis. Model-model tersebut terdiri dari PCM, GPCM, RSM, GRM, NRM dan NLM yang mana secara keseluruhan terinduk dalam dua jenis data yakni data nominal dan ordinal. Selain itu, analisis menggunakan R untuk menghasilkan output yang stabil menggunakan paket mirt yang mana lebih mampu memperkirakan model IRT jenis ini secara mendalam. Berdasarkan paparan yang telah diberikan, pembaca diharapkan mampu menggali lebih dalam mengenai R pada analisis IRT unidimensi data politomi untuk TAM package ataupun modifikasi syntax untuk lebih detail menggali informasi pada setiap jenis analisisnya. Paket tersebut dapat digunakan untuk mengestimasi model IRT politomi dengan sudut pandang yang berbeda dengan MIRT. Model IRT yang disajikan dalam bab ini cocok untuk pengujian unidimensional struktur yang terdiri dari item-item politomi dengan respons berurutan (ordinal) ataupun kategori nominal. Dalam bab berikutnya, akan lebih terfokus pada multidimensi dari model IRT baik dalam data dikotomus maupun politomi.

Referensi

Andrich, D. (1978). A rating formulation for ordered response categories. *Psychometrika*, *43*(4), 561–573. https://doi.org/10.1007/BF02293814

Bock, D. . (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, *37*(1), 29–51. https://doi.org/10.1007/BF02291411

Chalmers, R. . (2012). MIRT: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*(6), 1–29.

De Ayala, R. . (2013). *The theory and practice of item response theory*. The Guilford Press.

Desjardins, C. D., & Bulut, O. (2018). *Handbook of educational measurement and psychometrics using r*. Taylor & Francis Group.

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychome-Trika*, *47*(2), 149–174. https://doi.org/10.1007/BF02296272

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*(2), 159–176. https://doi.org/10.1177/014662169201600206

Paek, I., & Cole, K. (2020). *Using R for Item Response*.

Retnawati, H. (2014). *Teori Respons Butir dan Penerapannya*. Yogyakarta: Parama Publishing.

Samejima, F. (1969). *Estimation of latent ability using a response pattern of graded scores*. Psychometric Society.

Samejima, Fumiko. (2016). Graded response models. In *Handbook of Item Response Theory* (Vol. 1, pp. 95–107). https://doi.org/10.1201/9781315374512

Suh, Y., & Bolt, D. . (2010). Nested logit models for multiple-choice item response data. *Psychometrika*, *75*(3), 454–473.

Thissen, D., Cai, L., & Bock, R. D. (2012). The Nominal Categories Item Response model. In R. Ostini & M. Nering (Eds.), *Polytomous Item Response Theory Models* (Taylor & F, pp. 43–76). Routledge, Taylor & Francis Group. https://doi.org/10.4135/9781412985413

Wainer, H., Bradlow, E. T., & Wang, X. (2007). *Testlet Response Theory and Its Applications*.

Chapter 8
IRT Multidimensi Penskoran Dikotomi

Oleh: Koryna Aviory & Heri Retnawati

Model teori respons butir memiliki beberapa karakteristik, salah satunya adalah unidimensi. Model ini mengukur sifat atau konstruksi laten tunggal. Ketika butir mengukur dua atau lebih sifat laten, maka instrumen tersebut mempunyai karakteristik multidimensi. Estimasi parameter butir dan tingkat sifat laten tersebut diestimasi menggunakan teori respons butir multidimensi atau disebut MIRT (*Multidimensional Item respon Theory*). MIRT digunakan untuk mengestimasi butir yang melibatkan banyak sifat laten. MIRT tidak hanya memberikan skor domain yang andal, tetapi menghasilkan skor secara keseluruhan yang andal (Yao, 2010). Model ini menjelaskan kemungkinan respons menjawab benar untuk butir yang diberikan sebagai fungsi dari vektor sifat laten, bukan sifat laten tunggal. MIRT dapat digunakan untuk berbagai tujuan, misalnya

1. Memperluas tujuan IRT unidimensional untuk memberikan deskripsi tentang karakteristik butir dan bagaimana informasi butir digabungkan untuk memberikan deskripsi tentang kemampuan seseorang (Reckase, 2009b).
2. Memodelkan data respons butir ketika satu atau lebih butir mengukur beberapa sifat laten secara bersamaan (Chang & Wang, 2011).
3. Meningkatkan presisi pengukuran ketika ada tes individu atau subtes yang mengukur sifat laten yang berkorelasi (Wang et al., 2004; Wetzel & Hell, 2014).

Jumlah dimensi yang dipakai pada model tergantung interaksi antara butir dengan peserta tes yang diselaraskan dengan tujuan analisis. Pada model teori respons butir multidimensi, data dapat berupa butir skor dikotomus maupun politomi. Butir skor dikotomus diperoleh dari pertanyaan yang hanya memuat dua pilihan jawaban saja, misalkan benar dan salah. Contoh instrumen yang menghasilkan data skor

dikotomus adalah instrumen tes berbentuk soal pilihan ganda atau instrumen nontes yang hanya memiliki dua pilihan jawaban saja.

Data yang diperoleh kemudian disusun dalam sebuah matriks data, misalkan matriks D, dengan x_{ij} menyatakan elemen pada baris ke-i dan kolom ke-j. Elemen pada baris menyatakan butir soal, sedangkan elemen pada kolom menyatakan peserta tes. Ada beberapa asumsi yang menjadi pertimbangan ketika menyusun matriks data, diantaranya:

1. Asumsi kemonotonan
   Kemampuan peserta tes berbanding lurus dengan probabilitas peserta tes yang menjawab butir soal dengan benar.
2. Asumsi turunan fungsi
   Turunan fungsi terdefinisikan jika fungsi probabilitas menjawab benar bersifat smooth.
3. Asumsi independensi lokal
   Probabilitas kombinasi respons dapat ditentukan dengan hasil probabilitas respons individual pada saat probabilitas dihitung kondisional pada titik dalam ruang yang didefinisikan oleh konstruk hipotetik.

8.1 Model MIRT

Dalam teori respons butir multidimensi terdapat dua model berdasarkan adanya hubungan antara sifat-sifat laten, yaitu *compensatory* dan *noncompensatory* (Sijtsma & Junker, 2006). Model MIRT *compensatory* mendefiniskan probabilitas respons yang benar berdasarkan jumlah serangkaian sifat laten yang dibobot dengan parameter kemiringan yang berbeda (indeks deskriminan). Pada model *compensatory*, kemampuan tinggi diperbolehkan pada salah satu dimensi, sedangkan dimensi lainnya berkemampuan rendah. Kemampuan yang dimaksudkan merupakan probabilitas menjawab benar pada butir tes. Contoh model *compensatory* pada kasus dua dimensi, seorang peserta tes dengan kemampuan tinggi pada salah satu dimensi (misalnya, dimensi 1) dan berkemampuan rendah pada dimensi lainnya (misalnya dimensi 2), masih memiliki probabilitas tinggi untuk menjawab butir dengan benar karena kemampuan rendah pada dimensi 1 dikompensasikan dengan kemampuan tinggi pada dimensi 2.

Model *noncompensatory* mendefiniskan probabilitas respons menjawab benar berdasarkan perkalian probabilitas dengan masing-masing memiliki sifat laten yang berbeda. Sehingga, kelemahan dalam satu dimensi tidak dapat dikompensasikan dengan kekuatan pada dimensi lainnya. Misalnya pada kasus dengan dua dimensi, seorang *testee* dengan kemampuan rendah dimensi 1 dan berkemampuan tinggi pada dimensi 2, mempunyai peluang untuk tidak menjawab butir dengan benar karena untuk menjawab item tersebut diperlukan kemampuan keduanya, baik dimensi 1 maupun dimensi 2.

Model *compensatory* memiliki dua tipe model, yaitu model MIRT logistic dan model ogive normal. Dalam model logistik, rendahnya kemampuan dapat dikompensasikan pada dimensi lainnya. Hal ini merupakan karakteristik kombinasi linear, sehingga model tersebut dinamakan model MIRT logistik linear. Model MIRT logistic linear dapat ditulis sebagai berikut.

$$P_i(\theta j) = c_i + (1 - c_i)\frac{\exp\left(\sum_{m=1}^{k} a_{im}\theta'_{jm} + d_i\right)}{1 + \exp\left(\sum_{m=1}^{k} a_{im}\theta'_{jm} + d_i\right)} \qquad (8.1)$$

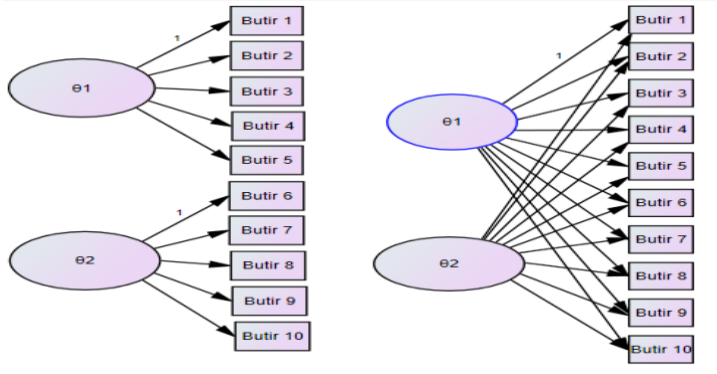
dimana $P_i(\theta_j)$ adalah probabilitas peserta ke-j menjawab benar butir ke-i, saat kemampuan peserta tes θ_j, a_{im} merupakan parameter diskriminan untuk butir ke-i pada dimensi ke-m, c_i adalah parameter *pseudo-guessing* butir ke-i, dan d_i merupakan tingkat kesulitan butir ke-i.

8.2 MIRT Between-Item dan Within-Item

Model MIRT berdasarkan struktur pengujiannya dikategorikan menjadi dua, yaitu *between-item* dan *within-item*. Dalam MIRT *between-Item*, setiap butir dikaitkan hanya dengan salah satu sifat laten. Model ini biasanya digunakan untuk memperkirakan subskor dari subtes yang mengukur sifat laten yang berkorelasi.

MIRT *within-item*, setiap butir dapat dikaitkan dengan dua atau lebih sifat laten yang diukur. Struktur pengujian ini dikenal sebagai struktur kompleks. Misalkan, suatu tes mengukur dua sifat laten (yaitu kecerdasan umum dan kemampuan kognitif), semua butir dikaitkan dengan kecerdasan umum dan setiap butir dikaitkan dengan kemampuan kognitif.

Ilustrasi MIRT *between-item* dan *within-item* untuk tes yang mengukur dua sifat laten dapat dilihat pada Gambar 8.1. Dalam model *between-item*, butir 1 sampai 5 mengukur sifat laten pertama (θ_1), sedangkan butir 6 sampai 10 mengukur sifat laten kedua (θ_2). Struktur pengujian menjadi multidimensi dimana setiap sifat laten didefiniskan oleh sekumpulan butir unidimensi sedangkan secara keseluruhan menjadi multidimensi. Dalam model *within-item*, butir dikaitkan dengan $\theta_1 \; dan \; \theta_2$ sehingga struktur menjadi kompleks. $\theta_1 \; dan \; \theta_2$ kemungkinan besar dikorelasikan. Nilai korelasi antara sifat laten menjadi lebih besar, sehingga hasil estimasi menjadi kurang dapat diandalkan.



Gambar 8.1 MIRT between-item dan within-item

8.3 MIRT Eksploratory dan Confirmatory

Struktur pengujian *eksploratory* dan *confirmatory* dilakukan ketika melakukan analisis data respons dengan MIRT. Jumlah dimensi yang diharapkan perlu ditentukan sebelum memperkirakan parameter butir dan tingkat sifat laten. Pendekatan MIRT *eksploratory* merupakan pendekatan untuk mencari hubungan antara sifat laten dan butir yang diturunkan dari data. Jika ada hipotesis mengenai struktur tes, maka jumlah dimensi dan hubungan antara butir dan dimensi perlu ditentukan (Reckase, 2009c). Pendekatan MIRT *confirmatory* dilakukan untuk melakukan pengujian berdasarkan teori.

8.4 Model MIRT Secara Umum

Model MIRT untuk data dikotomus yang akan dibahas pada Chapter 8 meliputi model 2PL, model Rasch, dan model IRT bifactor.

8.4.1 Model MIRT 2PL

MIRT 2PL (M2PL) merupakan perpanjangan dari model 2PL unidimensi. MIRT 2PL yang akan dibahas adalah model MIRT *compensatory* dimana tingkat kesulitan dan diskriminan dapat diestimasi dari tes multidimensi. Struktur pengujian dapat berupa *between-item* dan *within-item*. Model M2PL dapat dituliskan sebagai berikut.

$$P(\theta_j, a_i, d_i) = \frac{exp\,(a_i\theta_j' + d_i)}{1 + exp\,(a_i\theta_j' + d_i)} \qquad (8.2)$$

dimana θ_j merupakan kemampuan peserta tes ke-j pada butir ke-i $(\theta_{j1}, \theta_{j2}, \dots, \theta_{jM})$, a_i merupakan kemiringan (indeks deskriminan) untuk setiap butir ke-i $(a_i = (a_{i1}, a_{i2}, \dots, a_{iM}))$, dan d_i merupakan parameter intercept untuk setiap butir ke-i. Parameter intercept (d_i) tidak sama dengan tingkat kesulitan butir dalam IRT unidimensional, karena parameter ini tidak dapat dianggap sebagai indikator kesulitan butir (Reckase, 2009a).

Tingkat kesulitan diperoleh dengan melakukan transformasi pada parameter intercept. Transformasi yang dilakukan sebagai berikut.

$$B_i = \frac{-d_i}{\sqrt{\sum_{m=1}^{M} a_{im}^2}} \qquad (8.3)$$

dimana B_i merupakan parameter tingkat kesulitan multidimensi untuk butir ke-i, sering disebut sebagai MDIFF. Semakin tinggi nilai B_i, maka semakin sulit butir tersebut.

Parameter diskriminan butir dari model MIRT juga diperoleh dengan menggunakan transformasi. Transformasinya adalah

$$A_i = \sqrt{\sum_{m=1}^{M} a_{im}^2} \qquad (8.4)$$

dimana A_i adalah parameter diskriminan multidimensi yang juga disebut sebagai MDISC. Parameter diskriminan multidimensi hampir sama dengan parameter diskriminan unidimensi. Jika suatu butir hanya

mengukur satu sifat laten, maka hanya akan memiliki satu elemen bukan nol (misalnya, $a_i = (1{,}4; 0; 0)$ dalam tes tiga dimensi). Pada kasus ini, parameter diskriminan multidimensi akan sama dengan elemen bukan nol dari a_i.

Estimasi model M2PL dapat dilakukan dengan program R. Sebelum melakukan analisis, terlebih dahulu mengaktifkan *packages* hemp dan mirt dengan perintah library.

```
library (hemp)
library (mirt)
```

Instrumen tes berupa soal ujian (mimic) sebanyak 24 butir. Ujian tersebut diikuti oleh 2000 peserta tes. Butir tersebut dinilai secara dikotomi (1 = benar; 0 = salah) dan diberi nama butir 1, butir 2, …, butir 24. Struktur tes yang digunakan adalah multidimensi, dimana butir 1 sampai butir 6, butir 13 sampai butir 21, butir 23, dan butir 24 mengukur sifat laten pertama. Sedangkan butir 7 sampai butir 20, butir 22 sampai butir 24 mengukur sifat laten kedua.

Estimasi model menggunakan model M2PL. *Packages* yang digunakan adalah mirt. Perintah R untuk estimasi model IRT multidimensional sangat mirip dengan estimasi model IRT unidimensional. Perbedaan utamanya terletak pada struktur pengujian, IRT multidimensional melibatkan dua atau lebih sifat laten.

Struktur tes dua dimensi didefinsikan terlebih dahulu, yaitu F_1 dan F_2. Setiap butir dipisahkan berdasarkan sifat latennya dengan menggunakan rentang, misalnya sifat laten pertama (1-6, 13-21, 23-24), sifat laten kedua (13-20, 22-24). Kovarians antara dua sifat laten perlu didefnisikan. Jika kovarians antara F_1 dan F_2 bernilai nol, maka kedua sifat laten tersebut ortogonal. Fungsi MIRT akan memperbaiki varians dari sifat laten F_1 dan F_2 bernilai satu. Untuk mengestimasi varians dalam model, maka $COV = F_1 * F_2, F_1 * F_1, F_2 * F_2$. Selain itu, rata-rata sifat laten ditetapkan bernilai nol. Jika rata-rata perlu diestimasi, maka struktur ujinya menjadi $MEAN = F_1 * F_2$. Struktur pengujiannya disimpan sebagai m2pl_mod.

```
m2pl_mod <- 'F1 = 1-6, 13-21, 23-24
             F2 = 7-20, 22-24
             COV = F1*F2'
```

Dalam MIRT, model yang akan digunakan untuk mengestimasi 2 parameter adalah M2PL (model = m2pl_mod), data yang digunakan adalah mimic (data = mimic), tipe butir yang diestimasi dengan 2PL (itemtype = "2PL"). Algoritma dalam mengestimasi fungsi MIRT adalah method = "EM". Untuk model MIRT hingga tiga dimensi, algoritma EM dianggap efektif, tetapi algoritma MHRM direkomendasikan untuk struktur pengujian yang melibatkan lebih dari tiga dimensi. Struktur pengujian disimpan sebagai m2pl_fit, estimasi parameter menggunakan fungsi coef, kemudian disimpan sebagai m2pl_params.

```
m2pl_fit <- mirt (data = mimic, model = m2pl_mod, itemtype =
"2PL", method = "EM", SE = T)
m2pl_fit
m2pl_params <- coef(m2pl_fit, simplify = T)
```

Parameter butir yang diestimasi dapat dicetak langsung dengan menambahkan $item pada akhir m2pl_params.

```
head (m2pl_params$items)
        a1       a2 d          g u
item1 1.0451546  0  0.03009274 0 1
item2 0.9549557  0 -0.30599493 0 1
item3 1.0843917  0  0.23204938 0 1
item4 1.2282254  0  0.20808671 0 1
item5 0.9071266  0  0.16610451 0 1
item6 0.8660056  0  0.77031887 0 1
```

Output tersebut sangat mirip dengan output model IRT unidimensi. Akan tetapi, pada MIRT ada dua estimasi parameter diskriminan, yaitu a_1 *dan* a_2, untuk sifat laten pertama dan kedua. Pada output tersebut terlihat bahwa ada beberapa estimasi parameter diskriminan yang sama dengan nol. Hal ini disebabkan karena butir tersebut hanya mengukur satu sifat laten, pertama atau kedua, bukan keduanya. Misalnya, butir 1 memiliki nilai diskriminan 1,045 untuk a_1 dan nilai diskriminan 0 untuk a_2. Butir tersebut mengukur sifat laten pertama (F_1) tetapi tidak mengukur sifat laten kedua (F_2). Butir 13 memiliki nilai diskriminan 0,828 untuk a_1 dan nilai diskriminan 0,675 untuk a_2. Hal ini berarti butir 13 mengukur sifat laten pertama dan kedua. Kolom d merupakan estimasi parameter intersep untuk setiap butir. Sedangkan g dan u merupakan estimasi parameter asimtot bawah dan asimtot atas. Asimtot bawah nilai yang menyatakan probabilitas

menjawab benar *testee* ketika kemampuan yang dimilikinya sangat rendah pada keseluruhan dimensi. Parameter ini sama artinya dengan parameter *c* pada IRT unidimensi (Retnawati, 2014b). Dalam model M2PL, estimasi parameter asimtot bawah ditetapkan bernilai nol dan parameter asimtot atas ditetapkan bernilai satu untuk setiap butir.

Fungsi MDIFF dan MDISC merupakan fungsi dari *packages* mirt yang digunakan untuk mengubah parameter intercept dan diskriminan menjadi tingkat kesulitan butir multidimensi dan parameter diskriminan multidimensi berdasarkan persamaan 8.3 dan 8.4. Estimasi parameter menggunakan data frame dan disimpan sebagai m2pl_butir. Untuk mencetak enam butir pertama menggunakan fungsi head.

```
m2pl_butir <- data.frame (MDISC(m2pl_fit), MDIFF(m2pl_fit))
colnames (m2pl_butir) <- c("m2pl_mdisc", "m2pl_mdiff")
head (m2pl_butir)
m2pl_mdisc       m2pl_mdiff
item1   1.0451546       -0.02879262
item2   0.9549557       0.32042841
item3   1.0843917       -0.21399037
item4   1.2282254       -0.16942062
item5   0.9071266       -0.18311061
item6   0.8660056       -0.88950791
```

Selain estimasi parameter butir, dapat ditentukan estimasi matriks varians-kovarians dari dua sifat laten tersebut dengan menambahkan $covar pada akhir perintah m2pl_params. Outputnya menunjukkan matriks 2×2 dimana elemen diagonalnya adalah varians dari sifat laten (F_1 dan F_2) yang bernilai satu. Sedangkan estimasi kovarians dari dua sifat laten bernilai 0,588.

```
m2pl_params$cov
       F1     F2
F1 1.000000 0.588067
F2 0.588067 1.000000
```

Varians dari sifat laten bernilai satu, maka matriks varians-kovarians merupakan matriks korelasi dari sifat-sifat laten. Sehingga dapat dikatakan bahwa korelasi antara F_1 dan F_2 sebesar 0,588. Akan tetapi, jika varians F_1 dan F_2 diestimasi dalam model, rumus yang digunakan untuk mengubah estimasi kovarians menjadi estimasi koefisien korelasi adalah

$$r_{F_1,F_2} = \frac{cov(F_1, F_2)}{(S_{F_1})(S_{F_2})} = \frac{0{,}588}{\sqrt{1}\sqrt{1}} = 0{,}588 \qquad (8.5)$$

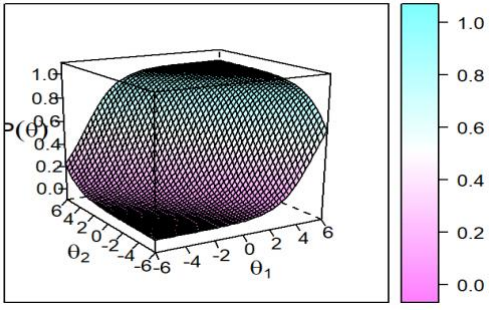
dimana $S_{F_1}\ dan\ S_{F_2}$ merupakan standar deviasi dari sifat laten F_1 dan F_2, dan cov (F_1 dan F_2) merupakan covarians antara dua sifat laten.

MIRT dapat digunakan untuk memeriksa karakteristik butir dan karakteristik tes. Model MIRT menghasilkan karakteristik butir yang tampak sebagai permukaan atau sering disebut sebagai permukaan respons butir (*Item Response Surface* atau *Item Characteristics Surface*). Plot kontur digunakan untuk menyampaikan informasi yang sama dengan plot permukaan. Plot kontur menunjukkan plot permukaan jika dilihat dari atas. Plot kontur lebih mudah untuk diinterpretasikan. Model MIRT merupakan model IRT yang melibatkan dua laten atau lebih, akan tetapi secara grafis ada beberapa keterbatasan dalam model ini, yaitu paket mirt pada R hanya mampu menangani model dua dimensi, fungsi grafis pada R hanya terbatas pada dua dimensi dan tiga dimensi. Permukaan respons butir akan sulit digambarkan jika dimensi yang diukur lebih dari dua.

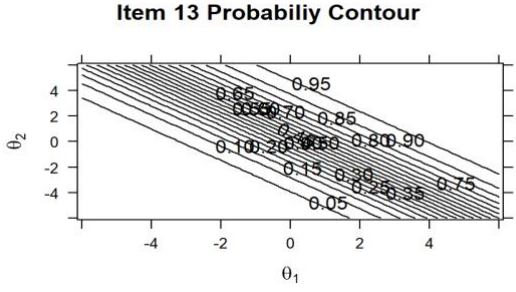
Butir 13 merupakan salah satu butir yang mengukur kedua sifat laten (F_1 dan F_2). Langkah pertama, membuat grafik karakteristik butir. Grafik yang disajikan berupa grafik tiga dimensi dengan sifat laten pertama berada pada sumbu x, sifat laten kedua berada pada sumbu y, dan probabilitas menjawab benar butir 13 pada sumbu z. Untuk membuat grafik tersebut menggunakan perintah type = "tracecontour".

```
itemplot (m2pl_fit, type = "trace", item = 13)
itemplot (m2pl_fit, type = "tracecontour", item = 13)
```

Gambar 8.2 dan 8.3 menunjukkan bagaimana probabilitas peserta tes menjawab benar butir 13 dari sifat laten pertama dan sifat laten kedua. Gambar 8.2 merupakan plot permukaan sedangkan Gambar 8.3 merupakan plot kontur. Model M2PL merupakan model *compensatory*, artinya kelemahan pada satu sifat laten dapat dikompensasi dengan kekuatan pada sifat laten lainnya. Misalkan $\theta_1 = 0\ dan\ \theta_2 = 2$, probabilitas menjawab benar butir 13 mendekati 0,80, walaupun $\theta_1 < \theta_2$.



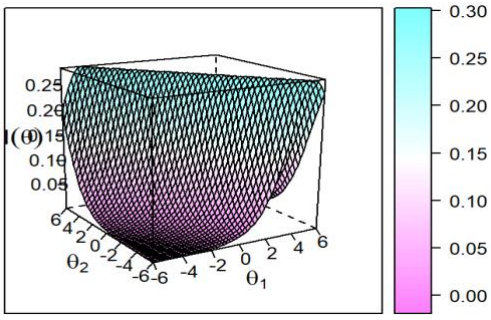
Gambar 8.2 Plot Permukaan Butir 13



Gambar 8.3 Plot Kontur Butir 13

Selain memeriksa karakteristik butir, dapat dilihat juga fungsi informasi yang diberikan oleh butir berdasarkan sifat latennya. Untuk membuat grafik tersebut digunakan perintah type = "info". Grafik yang dihasilkan mirip dengan plot permukaan karakteristik butir, akan tetapi sumbu *z* menunjukkan tingkat informasi butir berdasarkan probabilitas menjawab benar. Gambar 8.4 menunjukkan bahwa tingkat informasi butir yang tertinggi ketika kedua sifat laten sekitar nol, sedangkan tingkat informasi butir terendah ketika kedua sifat laten menjadi sangat rendah atau sangat tinggi.

```
itemplot (m2pl_fit, type ="info", item = 13)
```



Gambar 8.4 Grafik Fungsi Informasi Butir 13

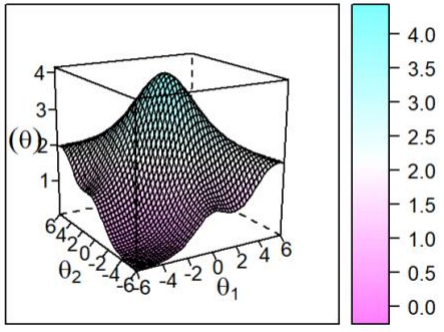
Grafik fungsi informasi butir juga dapat digambarkan sebagai plot kontur, sama seperti plot karakteristik butir. Selain itu, grafik fungsi informasi dapat digunakan untuk memeriksa skor yang diharapkan dan *standard error* setiap item. Perintah R untuk menggambar plot adalah

```
itemplot (m2pl_fit, type = “infocontour”, item = 13)
itemplot (m2pl_fit, type = “score”, item = 13)
itemplot (m2pl_fit, type = “SE”, item = 13)
```

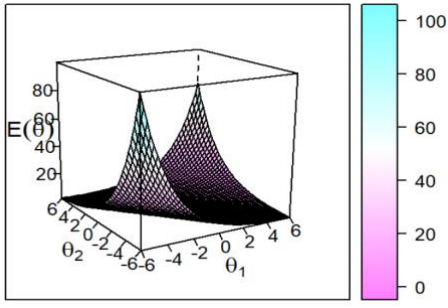
Grafik fungsi informasi juga dapat digunakan untuk menggambar plot pada tingkat tes. Plot ini merangkum fungsi informasi test (*Test Information Function* atau TIF) dan *Conditional Standard Error Measurement* (cSEM). cSEM adalah standar deviasi dari skor pengamatan peserta tes yang diharapkan dari pengukuran dengan skor benar yang tetap dan tidak berubah. Perintah berikut ini menunjukkan perintah untuk membuat plot TIF dan plot cSEM.

```
plot (m2pl_fit, type = ‘info’)
plot (m2pl_fit, type = ‘SE’)
```

Gambar 8.5 menunjukkan grafik TIF sedangkan Gambar 8.6 menunjukkan grafik cSEM.



Gambar 8.5 Grafik TIF



Gambar 8.6 Grafik cSEM

Model MIRT dapat mengestimasi tingkat sifat laten. Metode *Maksimum Aposteriori* (MAP) dan *Expected Aposteriori* (EAP) digunakan untuk mengestimasi tingkat sifat laten dari model M2PL. Estimasi *Maksimum Likelihood* (ML) dapat diperoleh dengan menuliskan metode = "ML". Estimasi ML kemungkinan gagal memberikan hasil ketika semua peserta tes menjawab salah atau semuanya benar. Masalah yang sama akan ditemukan ketika memperkirakan tingkat sifat laten dari MIRT. Oleh karena itu, metode yang digunakan untuk mengestimasi tingkat sifat laten adalah metode = "MAP" untuk estimasi MAP dan metode = "EAP" untuk estimasi EAP, dimana hasilnya disimpan sebagai m2pl_map dan m2pl_eap. Jika ingin mencetak pada baris pertama, maka menggunakan fungsi head.

```
m2pl_map <- fscores (m2pl_fit, metode = "MAP", full.scores =
T,  full.scores.SE = T)
```

```
head (m2pl_map)
```

```
        F1         F2       SE_F1     SE_F2
[1,]  0.4453246  0.0453828 0.4918731 0.4803416
[2,]  0.9311577  1.3492089 0.5424382 0.5421459
[3,]  1.1960052  0.5477967 0.5417752 0.5022786
[4,] -0.2446835 -0.4667777 0.4834888 0.4855015
[5,]  0.9780828 -0.1158884 0.5144185 0.4850905
[6,]  0.4324385  1.1331393 0.5072794 0.5155978
```

```
m2pl_eap <- fscores (m2pl_fit, metode = "EAP", full.scores =
T, full.scores.SE = T)
head (m2pl_eap)
```

```
       F1         F2          SE_F1     SE_F2
[1,]  0.4453246  0.0453828 0.4918731 0.4803416
[2,]  0.9311577  1.3492089 0.5424382 0.5421459
[3,]  1.1960052  0.5477967 0.5417752 0.5022786
[4,] -0.2446835 -0.4667777 0.4834888 0.4855015
[5,]  0.9780828 -0.1158884 0.5144185 0.4850905
[6,]  0.4324385  1.1331393 0.5072794 0.5155978
```

Hasil ouput menunjukkan bahwa F1 dan F2 merupakan estimasi sifat laten, sedangkan SE_F1 dan SE_F2 menunjukkan *standart error* untuk estimasi sifat laten. Estimasi sifat laten dengan metode MAP dan EAP dapat digabungkan, kemudian disimpan dalam m2pl_scores. Selain itu, korelasi diantara estimasi sifat laten juga dapat dihitung. Hasil perhitungan menunjukkan bahwa estimasi sifat laten dengan metode MAP dan EAP saling berkorelasi sebesar 0,789.

```
m2pl_scores <- data.frame (map1 = m2pl_map [,1], map2 =
m2pl_map [,2], eap1 = m2pl_eap [,1], eap2 = m2pl_eap [,2])
m2pl_scores
cor (m2pl_scores)
```

```
         map1      map2      eap1     eap2
map1    1.0000000 0.7890935 1.0000000 0.7890935
map2    0.7890935 1.0000000 0.7890935 1.0000000
eap1    1.0000000 0.7890935 1.0000000 0.7890935
eap2    0.7890935 1.0000000 0.7890935 1.0000000
```

Estimasi model MIRT dapat dilakukan lebih spesifik. Misalnya, beberapa butir dapat dibatasi untuk memiliki parameter kemiringan dan intersep yang sama, dengan asumsi bahwa butir yang diharapkan diantara sifat laten memiliki parameter kemiringan yang sama, maka dapat menggunakan CONSTRAIN dalam definisi model.

```
m2pl_mod_constrant <- 'F1 = 1-6, 13-21, 23-24
                       F2 = 7-20, 22-24
                       COV = F1*F2
```

```
                    CONSTRAIN = (1-6, 21, a1), (7
                    20, 22, a2)'PERLU OUTPUT
```

Dalam M2PL juga terdapat model *noncompensatory*. Untuk mengestimasi model tersebut, maka itemtype = "2PL" diganti dengan itemtype = "PC2PL". Model M2PL *noncompensatory* memiliki parameter kemiringan dan intersep yang terpisah untuk masing-masing sifat laten. Dalam model M2PL *noncompensatory* tidak ada kompensasi diantara sifat laten.

```
m2pl_fit <- mirt (data = mimic, model = m2pl_mod,
            itemtype = "PC2PL", SE = T)
```

Model M2PL tidak melibatkan tebakan apapun. Akan tetapi, jika jawaban peserta tes diduga dipengaruhi karena tebakan, maka model multidimensi 3PL dapat dipilih dengan itemtype = "3PL". Nilai asimtot dapat ditentukan untuk semua butir dengan fungsi mirt (misalnya nilai tebakan = 0,10).

8.4.2 Model Rasch Multidimensi

Model Rasch multidimensi mampu mengestimasi parameter butir dari tes yang mengukur beberapa sifat laten (Adams et al., 1997). Ketika $P(X_{ij}) = 1$ didefinisikan sebagai peluang peserta tes ke-i untuk menjawab dengan benar ditentukan oleh kombinasi kemampuan peserta tes ke-i (θ_i), dengan tingkat kesukaran butir ke-j (δ_j) (McCullagh & Nelder, 1990; Hoijtink & Vollema, 2003). Model Rasch multidimensi dapat ditulis sebagai berikut.

$$P_j(X_{ijk} = 1|\theta_j) = \frac{\exp \sum_{k=1}^{\kappa} b_{jk}(\theta_{ik} - \delta_j)}{1 + \exp \sum_{k=1}^{\kappa} \mathrm{b_{jk}}\,(\theta_j - \delta_j)} \quad (8.6)$$

Dimana $b_{jk} = 1$, merupakan konstanta diskriminasi butir (Verguts & De Boeck, 2000). Sehingga pada model Rasch, parameter yang dapat diestimasi adalah tingkat kesukaran dan kemampuan peserta tes.

Model Rasch multidimensi dapat digunakan untuk pengujian *between-item* dan *within-item* yang mengukur sifat laten. Korelasi diantara sifat laten berfungsi sebagai informasi tambahan. Estimasi kemampuan antar dimensi dipengaruhi oleh korelasi antar dimensi

(Briggs & Wilson, 2003). Contoh estimasi dengan model Rasch multidimensi dilakukan dengan menggunakan data yang telah didefiniskan pada model M2PL. Dalam Rasch multidimensi, perintah yang digunakan itemtype = “Rasch”.

```
rasch_mod <- 'F1 = 1-6, 13-21, 23-24
              F2 = 7-20, 22-24
              COV = F1*F2'
mrasch_fit <- mirt (data = mimic, model = mrasch_mod, itemtype
= "Rasch", SE = T)
mrasch_params <- coef(mrasch_fit, simplify = T)
```

Parameter yang diestimasi kemudian dicetak delapan baris pertama dengan perintah head. Output yang dihasilkan sangat mirip dengan output untuk M2PL. Akan tetapi, parameter diskriminasi setiap butir ditetapkan bernilai 1 dalam model Rasch multidimensi.

```
head (mrasch_params$items,8)
      a1 a2  d          g u
item1  1  0  0.02939241 0 1
item2  1  0 -0.29424319 0 1
item3  1  0  0.21640911 0 1
item4  1  0  0.18627632 0 1
item5  1  0  0.16313740 0 1
item6  1  0  0.76289725 0 1
item7  0  1  0.02322080 0 1
item8  0  1 -0.90994871 0 1
```

Parameter intersep dirubah menjadi parameter tingkat kesulitan butir dengan menggunakan perintah MDIFF. Jika dibandingkan dengan estimasi parameter intersep, hasil output estimasi parameter tingkat kesulitan menunjukkan bahwa hanya tanda parameter intercept pada berubah (untuk butir yang mengukur satu sifat laten). Hasil estimasi tingkat kesulitan menunjukkan bahwa butir yang paling sukar adalah butir 11, dengan tingkat kesulitan 1,024. Sedangkan butir yang paling mudah adalah butir 10 dengan nilai estimasi sebesar -0,889.

```
mrasch_mdiff <- MDIFF (mrasch_fit)
mrasch_mdiff
         MDIFF_1
item1  -0.02939241
item2   0.29424319
item3  -0.21640911
item4  -0.18627632
item5  -0.16313740
item6  -0.76289725
```

```
item7  -0.02322080
item8   0.90994871
```

Estimasi matriks varians-kovarians yang mengukur dua sifat laten dapat ditentukan. Estimasi varians untuk sifat laten pertama dan kedua sebesar 0,689 dan 0,739. Estimasi kovarians yang dihasilkan sebesar 0,344.

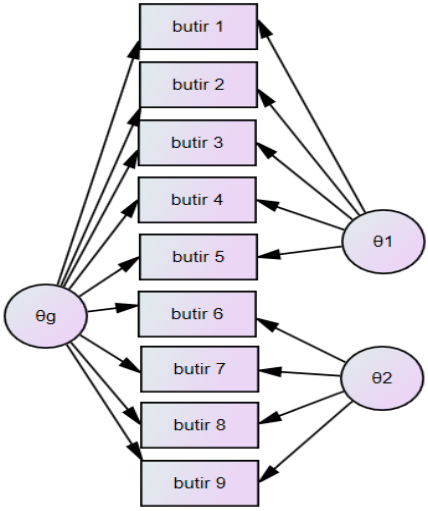
```
mrasch_params$cov
   F1     F2
F1 0.689 0.344
F2 0.344 0.739
```

Dalam model Rasch multidimensi juga dapat ditentukan karakteristik butir, fungsi informasi butir, tes informasi, dan *standarr error*. Langkah yang dilakukan sama seperti pada M2PL, yang berbeda hanya pada perintah modelnya saja.

```
Itemplot (mrasch_fit, type = "trace", item = 13)
Itemplot (mrasch_fit, type = "tracecontour", item = 13)
plot (mrasch_fit, type = "info", item = 13)
plot (mrasch_fit, ketik = "SE", item = 13)
plot (mrasch_fit, ketik = "score", item = 13)
```

8.4.3 Model IRT Bi-Factor

Model IRT bi-factor adalah bentuk khusus dari model MIRT untuk pengujian *within-item*. Jumlah sifat laten yang diukur pada model bi-faktor hanya dua (sifat laten umum dan sifat laten sekunder). Dalam IRT bi-faktor, semua butir dihubungkan dengan sifat laten umum dan setiap butir juga dihubungkan dengan sifat laten sekunder. Sifat laten umum menjelaskan variabilitas bersama di dalam butir, sedangkan sifat laten sekunder menjelaskan variabilitas secara khusus (unidimensi) di dalam butir (Chalmers, 2012c). Dalam identifikasi model bi-faktor, sifat laten umum dan sifat laten sekunder dibatasi agar tidak saling berkorelasi. Struktur bi-faktor dapat diterapkan pada berbagai model MIRT. Model IRT bi-faktor dapat digunakan untuk skor dikotomus maupun politomi.



Gambar 8.7 Model Bi-Faktor

Gambar 8.7 menunjukkan contoh struktur model bi-faktor, dimana butir 1 sampai butir 5 mengukur sifat laten pertama (θ_1), butir 6 sampai butir 10 mengukur sifat laten kedua (θ_2), dan semua butir juga dihubungkan dengan sifat laten umum (θ_g). Sifat laten pertama (θ_1), sifat laten kedua (θ_2), dan sifat laten umum (θ_g) tidak saling berkolerasi dalam model bi-faktor.

Simulasi model bi-faktor dalam *packages* mirt dapat dilakukan dengan beberapa langkah. Pertama, mendefiniskan sebuah vektor dimana vektor tersebut dapat ditentukan hubungan antara setiap butir dengan sifat laten umum dan sifat laten sekunder, misalnya sifat laten umum yang diukur adalah aritmatika, sifat laten sekundernya aljabar, geometri, dan kalkulus. Sifat laten umum membahas semua butir yang ada pada tes. Jumlah butir sebanyak 32. Panjang vektor merupakan jumlah butir. Data yang digunakan adalah dikotomus. Kedua, mengestimasi parameter.

```
data (SAT12)
data <- key2binary (SAT12,
       key = c
(1,4,5,2,3,1,2,1,3,1,2,4,2,1,5,3,4,4,1,4,3,3,4,1,3,5,1,3,1,5,4
,5))
```

```
specific <- c
(2,3,2,3,3,2,1,2,1,1,1,3,1,3,1,2,1,1,3,3,1,1,3,1,3,3,1,3,2,3,1
,2)
bifactor_fit <- bfactor (data, specific)
bifactor_params <- coef (bifactor_fit, simplify = T)
head (bifactor_params$items)
```

```
       a1        a2  a3       a4         d           g  u
Item.1 0.7849370  0  0.4374608 0.0000000 -1.0720733  0  1
Item.2 1.4894082  0  0.0000000 0.8153208  0.4708391  0  1
Item.3 1.1469831  0 -0.1531092 0.0000000 -1.1727331  0  1
Item.4 0.5261708  0  0.0000000 0.5811485 -0.5574675  0  1
Item.5 0.9674140  0  0.0000000 0.5150225  0.6282463  0  1
Item.6 1.1358511  0  0.5793347 0.0000000 -2.1261537  0  1
```

Hasil output menunjukkan bahwa empat kolom pertama (a_1, a_2, a_3, a_4) merupakan estimasi parameter kemiringan untuk sifat laten umum dan sifat laten sekunder (aljabar, geometri, dan kalkulus). Parameter kemiringan a_1 bernilai lebih besar dari nol untuk semua butir, sedangkan parameter kemiringan $a_2, a_3, dan\, a_4$ ada yang bernilai nol, artinya butir tersebut tidak terikat dengan sifat laten tertentu. Misalnya, butir 4 dikaitkan dengan sifat laten umum dan sifat laten sekunder ketiga (kalkulus). Oleh karena itu, estimasi kemiringan sifat laten utama sebesar 0,526 dan estimasi kemiringan sifat laten sekunder ketiga sebesar 0,581. Estimasi intercept untuk butir 4 sebesar -0,557.

Estimasi kemiringan dan intercept juga dirubah untuk mengestimasi diskriminan dan tingkat kesulitan pada multidimensi dengan menggunakan perintah MDISC dan MDIFF.

```
bifactor_items <- cbind (MDISC (bifactor_fit),
MDIFF(bifactor_fit))
colnames (bifactor_items) <- c("mdisc", "mdiff")
head (bifactor_items)
```

```
       mdisc       mdiff
Item.1 0.8986090  1.1930365
Item.2 1.6979649 -0.2772961
Item.3 1.1571572  1.0134605
Item.4 0.7839575  0.7110940
Item.5 1.0959644 -0.5732361
Item.6 1.2750633  1.6674888
```

Dalam model bi-factor, korelasi diantara sifat laten bernilai nol. Matriks kovariansi dari sifat laten dapat ditunjukkan dengan menambahkan perintah $cov pada estimasi parameternya. Hasil estimasi menunjukkan bahwa elemen diagonal bernilai satu, sedangkan elemen non-diagonal bernilai nol. Keempat sifat laten merupakan model orthogonal dimana tidak ada korelasi diantara sifat laten.

```
bifactor_params$cov
    G  S1  S2 S3
G   1   0   0   0
S1  0   1   0   0
S2  0   0   1   0
S3  0   0   0   1
```

Referensi

Adams, T., Bezner, J., & Steinhardt, M. (1997). The conceptualization and measurement of perceived wellness: Integrating balance across and within dimensions. *American Journal of Health Promotion*, *11*(3), 208–218. https://doi.org/10.4278/0890-1171-11.3.208

Briggs, D. C., & Wilson, M. (2003). Requests for reprints should be sent to An Introduction to Multidimensional Measurement using Rasch Models. *JOURNAL OF APPLIED MEASUREMENT*, *4*(1), 87–100.

Chalmers, R. P. (2012). Mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*. https://doi.org/10.18637/JSS.V048.I06

Chang, H.-H., & Wang, C. (2011). M.D. Reckase (2009) Multidimensional Item Response Theory (Statistics for Social and Behavioral Sciences). *Psychometrika*, *76*(3), 504–506. https://doi.org/10.1007/S11336-011-9212-X

Desjardins, C. D., & Bulut, O. (2018). *Handbook of Education Measurement and Psychometrics Using R*. CRC Press (Taylor & Francis Group).

Hoijtink, H., & Vollema, M. (2003). Contemporary Extensions of the Rasch Model. *Quality and Quantity*, *37*(3), 263–276. https://doi.org/10.1023/A:1024497124563

McCullagh, P., & Nelder, J. A. (1990). Generalized Linear Models, 2nd Edn. In *Chapman and Hall* (Second Edi). https://doi.org/10.2307/2347392

Reckase, M. D. (2009a). Estimation of Item and Person Parameters. *Multidimensional Item Response Theory*, 137–178. https://doi.org/10.1007/978-0-387-89976-3_6

Reckase, M. D. (2009b). Historical Background for Multidimensional Item Response Theory (MIRT). *Multidimensional Item Response Theory*, 57–77. https://doi.org/10.1007/978-0-387-89976-3_3

Reckase, M. D. (2009c). Multidimensional Item Response Theory Models. *Multidimensional Item Response Theory*, 79–112. https://doi.org/10.1007/978-0-387-89976-3_4

Retnawati, H. (2014). Teori respons butir dan penerapannya: Untuk peneliti, praktisi pengukuran dan pengujian, mahasiswa pascasarjana. *Nuha Medika.* www.nuhamedika.gu.ma

Sijtsma, K., & Junker, B. W. (2006). Item response theory: Past performance, present developments, and future expectations. *Behaviormetrika*, *33*(1), 75–102.

Verguts, T., & De Boeck, P. (2000). A Rasch model for detecting learning while solving an intelligence test. *Applied Psychological Measurement*, *24*(2), 151–162.

https://doi.org/10.1177/01466210022031589
Wang, W. C., Chen, P. H., & Cheng, Y. Y. (2004). Improving Measurement Precision of Test Batteries Using Multidimensional Item Response Models. *Psychological Methods*, *9*(1), 116–136. https://doi.org/10.1037/1082-989X.9.1.116
Wetzel, E., & Hell, B. (2014). Multidimensional Item Response Theory Models in Vocational Interest Measurement: An Illustration Using the AIST-R. *Journal of Psychoeducational Assessment*, *32*(4), 342–355. https://doi.org/10.1177/0734282913508244
Yao, L. (2010). Reporting valid and reliable overall scores and domain scores. *Journal of Educational Measurement*, *47*(3), 339–360. https://doi.org/10.1111/J.1745-3984.2010.00117.X

Chapter 9
IRT Multidimensi Penskoran Politomi

Oleh: Sumin, Firmansyah, & Samsul Hadi

Ada dua asumsi mendasar IRT unidimensional, yaitu; Independensi Lokal; Probabilitas menanggapi suatu item secara statistik tidak tergantung pada menanggapi item lain mana pun sementara dikondisikan pada kemampuan dalam tes (Hambleton & Swaminathan), dan Unidimensionalitas; Hanya ada satu kemampuan yang mendasari perbedaan respons individu terhadap item (Embretson & Reise, 2000).

Meskipun IRT unidimensional telah banyak digunakan dalam pengukuran pendidikan, penelitian menunjukkan bahwa asumsi unidimensional seringkali sulit dipenuhi dalam konteks dunia nyata (Ackerman, 1994; Reckase, 1985). Dalam situasi dunia nyata, item tes mungkin memerlukan lebih dari satu kemampuan/sifat atau konstruksi hipotesis untuk menyelesaikannya.

Ada beberapa alasan mengapa kita harus memperhatikan multidimensionalitas dalam pengukuran. Pertama, penting untuk mengukur apa yang ingin Anda ukur. Jika asumsi unidimensionalitas IRT (UIRT) dilanggar, maka estimasi parameter item akan menjadi bias, dan *standard error* yang terkait dengan estimasi kemampuan akan terlalu kecil. Kedua, uji keadilan (*fairness*). Multidimensionalitas dapat menyebabkan DIF, dan bias item. Beberapa tes memiliki desain yang membuat seseorang mengharapkan multidimensi. Beberapa tes dimodelkan sebagai unidimensional, tetapi hasilnya dilaporkan sebagai sub skor atau skor komposit. Beberapa dimensi mungkin berguna secara diagnostik.

Sama halnya dengan IRT dikotomus dan politomi unidimensi, IRT Politomi multidimensi juga memiliki beberapa pendekatan. IRT Politomi atau IRT untuk data ordinal dan nominal pada prinsipnya merupakan perluasan atau pengembangan model politomi unidimensi. Pendekatan IRT yang dapat digunakan untuk memodelkan item politomi multidimensi adalah (Chalmers, 2012; Desjardins & Bulut, 2018):

1. *Multidimensional Generalized Partial Credit Model* (MGPCM) dikembangkan oleh Muraki (1992); model teori respons item polytomous; sangat tidak setuju hingga sangat setuju (tipe Likert), dan format pilihan ganda, atau tes essay. Skor yang diberikan kepada peserta tes pada item dilambangkan dengan $k = 0, 1, 2 \ldots K_i$
2. *Multidimensional Partial Credit Model* (MPCM), dikembangkan oleh Masters (1982) model teori respons item dengan lebih dari dua kategori, yaitu beberapa pilihan respons; misalnya, sangat tidak setuju hingga sangat setuju (tipe Likert), dan format pilihan ganda, atau essay. Kunci dari fungsi model ini adalah spesifikasi dari matriks bobot W_{ilk}. Misalkan suatu item tes memiliki kategori skor $K_i = 3$, yaitu; 0, 1, dan 2, diasumsikan item tersebut sensitif terhadap perbedaan dua dimensi
3. *Multidimensional Graded Response Model* (MGRM) dikembangkan oleh Samejima (1972); model teori respons item polytomous; respons dikumpulkan dari item dengan kategori yang diurutkan secara eksplisit; misalnya, item lima kategori dengan kategori mulai dari sangat tidak setuju hingga sangat setuju untuk skala Thurstone, respons kategori $k = 0, 1, 2, 3, 4 \ldots m_i$
4. *Multidimensional Nominal Response Model* (MNRM) dikembangkan oleh Block (1972) dapat memodelkan kategori yang dinilai secara nominal (level pengukuran nominal), dapat digunakan untuk memodelkan respons terhadap item pilihan ganda atau item lain di mana salah satu kategori item diberi skor secara nominal. Dalam melakukannya, model ini menyediakan cara untuk memecahkan masalah lama dalam pengukuran pendidikan, yang secara bersamaan memperkirakan semua parameter dari alternatif jawaban item pilihan ganda.

Leventhal and Stone (2018) merangkum tiga multidimensi teori respons item (IRT) untuk skala Likert atau skala rating, yaitu: model respons nominal multidimensi (MNRM), model kredit parsial umum yang dimodifikasi (MGPCM), dan model IRTree. Menurut Jin & Chen (2020) perbedaan utama antara MNRM dan MGPCM adalah bahwa θ

dan γ yang diasumsikan sebagai compensatory dalam MNRM, sedangkan θ dan ω adalah non-compensatory dalam MGPCM.

9.1 *Multidimensional Generalized Partial Credit Model* (MGPCM)

Perluasan multidimensi dari model kredit parsial umum (GPCM) dirancang untuk menggambarkan interaksi orang dengan item yang diberi skor dengan lebih dari dua kategori. Skor maksimum untuk Item i diwakili oleh K_i. Agar konsisten dengan cara skor item dikotomis, skor terendah diasumsikan 0 dan terdapat kategori skor $K_i + 1$ secara keseluruhan. Skor yang diberikan kepada seseorang pada item tersebut diwakili oleh $k = 0, 1, \dots, K_i$. Representasi matematis model MGPC diberikan oleh persamaan berikut.

$$P(u_{ij} = k|\theta_j) = \frac{e^{k\, a_i\, \theta_j' - \sum_{u=0}^{k} \beta_{iu}}}{\sum_{v=0}^{K_i} e^{v\, a_i\, \theta_j' - \sum_{u=0}^{k} \beta_{iu}}} \tag{9.1}$$

di mana β_{iu} adalah parameter ambang batas untuk kategori skor u β_{i0} didefinisikan sebagai 0, dan semua simbol lain memiliki arti yang ditentukan sebelumnya. Representasi model yang diberikan di sini adalah sedikit variasi dari bentuk yang diberikan dalam Yao dan Schwarz (2006).

Ada dua perbedaan penting antara persamaan untuk model MGPC dan untuk model GPC yang diberikan dalam persamaan di atas. Pertama, model tidak menyertakan parameter kesulitan dan ambang yang terpisah. Kedua, karena θ adalah vektor dan β_s adalah skalar, maka tidak mungkin untuk mengurangkan parameter ambang dari θ. Sebaliknya, bentuk kemiringan/intersep dari model kredit parsial tergeneralisasi digunakan sebagai dasar dari generalisasi multidimensi, $a\theta + d$, tetapi dengan tanda intersep dibalik. Hasilnya adalah bahwa s tidak dapat dijelaskan dengan cara yang sama seperti parameter ambang batas dalam model versi UIRT. Hal ini akan dibahas lebih detail setelah menyajikan bentuk permukaan respons item.

9.2 *Multidimensional Partial Credit Model* (MPCM)

Ada sejumlah penyederhanaan versi multidimensi dari model kredit parsial umum yang memiliki sifat khusus dari model Rasch.

Artinya, mereka memiliki statistik yang cukup yang dapat diamati untuk parameter item dan orang. Kelderman dan Rijkes (1994) menyajikan bentuk umum untuk satu perluasan multidimensi dari model Rasch ke kasus butir tes polytomous. Model mereka disajikan rumus di bawah ini menggunakan simbol yang sedikit berbeda dari presentasi aslinya untuk memfasilitasi perbandingan dengan model lain yang disajikan dalam buku ini. Model yang sangat mirip disajikan oleh Adams et al. (1997).

$$P(u_{ij} = k|\theta_j) = \frac{e^{\sum_{\ell=1}^{m}(\theta_{j\ell} - b_{i\ell k})W_{i\ell k}}}{\sum_{r=1}^{K_i} e^{\sum_{\ell=1}^{m}(\theta_{j\ell} - b_{i\ell k})W_{i\ell r}}} \quad (9.2)$$

di mana b_{ilk} adalah parameter sulit untuk Item i pada dimensi l untuk kategori skor k, dan W_{ilk} adalah bobot penilaian yang telah ditentukan sebelumnya untuk Item i terkait dengan dimensi l dan kategori skor k. Simbol lainnya memiliki arti yang sama seperti pada persamaan sebelumnya. Kunci dari fungsi model ini adalah spesifikasi dari matriks bobot, W_{ilk}. Misalkan suatu item tes memiliki kategori $K_i = 3$, dengan skor 0,1 dan 2. Diasumsikan juga item tersebut peka terhadap perbedaan dua dimensi

9.3 *Multidimensional Graded Response Model* (MGRM)

Pendekatan lain untuk pemodelan multidimensi dari tanggapan terhadap item tes dengan lebih dari dua kategori skor disajikan oleh Muraki dan Carlson (1993). Model ini merupakan generalisasi dari model respons bergradasi unidimensional dan menggunakan fungsi respons yang memiliki bentuk ogive normal. Seperti versi unidimensional model ini, model multidimensi mengasumsikan bahwa keberhasilan penyelesaian tugas yang ditentukan oleh item tes memerlukan sejumlah langkah dan mencapai langkah k membutuhkan keberhasilan pada langkah k -1. Jenis model ini juga sesuai untuk skala penilaian di mana kategori penilaian memasukkan semua kategori sebelumnya. Contohnya adalah skala penilaian untuk jumlah waktu yang dihabiskan untuk sebuah proyek.

Jika kategori peringkat yang menunjukkan satu jam dihabiskan untuk sebuah proyek dipilih, berarti bahwa semua kategori peringkat yang menentukan kurang dari satu jam juga berlaku. Parameterisasi

model yang diberikan di sini menganggap skor terendah pada Item i adalah 0 dan skor tertinggi adalah m_i. Probabilitas menyelesaikan k atau lebih langkah diasumsikan meningkat secara monoton dengan peningkatan salah satu konstruksi hipotesis yang mendasari pengujian seperti yang diwakili oleh elemen vektor θ. Ini setara dengan mendikotomikan skala pada k dan mencetak k atau lebih tinggi sebagai 1 dan di bawah k sebagai 0 dan menyesuaikan model dikotomis dengan hasilnya. Probabilitas menyelesaikan k atau lebih langkah dimodelkan dengan model ogive normal dua parameter dengan parameter person didefinisikan sebagai kombinasi linier dari elemen-elemen dalam vektor θ yang dibobot dengan parameter diskriminasi.

Probabilitas menerima spesifik skor k, adalah perbedaan antara probabilitas berhasil melakukan pekerjaan untuk k langkah atau lebih dan berhasil melakukan pekerjaan untuk $k+1$ langkah atau lebih. Jika probabilitas memperoleh skor item k atau lebih tinggi pada tingkat θ tertentu adalah $P^*(u_{ij} = k|\theta_j)$, maka probabilitas seorang peserta ujian akan menerima skor k dapat didefinisikan dengan formula berikut:

$$P(u_{ij} = k|\theta_j) = P^*(u_{ij} = k|\theta_j) - P^*(u_{ij} = k+1|\theta_j) \quad (9.3)$$

dimana $P^*(u_{ij} = 0|\theta_j) = 1$ karena melakukan pekerjaan untuk Langkah 0 atau lebih adalah berisikan semua peserta ujian dan $P^*(u_{ij} = m_i + 1|\theta_j) = 0$ karena itu memungkinkan untuk mengerjakan pekerjaan yang mewakili lebih dari m_i kategori. Probabilitas terakhir didefinisikan sehingga probabilitas setiap skor dapat ditentukan dari persamaan di atas. Samejima (1969) melabeli istilah di sisi kanan ekspresi sebagai fungsi respons kategori kumulatif dan istilah di sisi kiri ekspresi sebagai fungsi respons kategori. Kurva ogive normal dari GRM diperoleh melalui persamaan berikut ini.

$$P(u_{ij} = k|\theta_j) = \frac{1}{\sqrt{2\pi}} \int_{a_i'\theta_j + d_{ik}+1}^{a_i'\theta_j + d_{ik}} e^{\frac{-t^2}{2}} dt, \quad (9.4)$$

dimana k adalah skor pada item 0, 1, 2, …m_i, $a_i$adalah sebuah vector parameter daya beda dan d_{ik} adalah sebuah parameter yang terkait

dengan kemudahan seseorang mencapai langkah ke-k dari item tertentu.

9.4 IRT Polytomous Multidimensi Menggunakan R

Studi kasus; Ujian essay mata kuliah metode statistika di sebuah perguruan tinggi, soal terdiri dari 20 item,dengan 250 peserta tes. Berdasarkan rubrik yang disusun, responden yang menjawab salah diberikan skor 0, jika siswa menjawab benar diberikan skor 1, 2, 3 dan 4 sesuai dengan tingkat kesukaran soal. Tabulasi nilai siswa disimpan pada Drive D dengan nama file "Data Essay.xlsx, tipe data adalah excel 2007 ke atas (.xlsx), atau "Data Essay.csv, tipe data adalah comma delimited (.csv).

9.4.1 Pemeriksaan Asumsi Unidimensi

Lankah pertama, kita install package analisis, sebagaimana telah dijelaskan Langkah-langkahnya pada Chapter 1, lanjutkan dengan mengaktifkan paket analisis *mirt* dan pendukunganya melalui perintah *Library ( )* pendukungnya.

Langkah kedua; Siapkan data format Excel dengan file extensi .csv separator tanda koma (,) atau .xlsx. Agar lebih mudah dan tidak perlu mengubah tipe file, sebaiknya gunakan ekstensi .xlsx

```
#Atur atau arahkan Direktori kerja dan panggil file kerja pada
R Studio dengan syntax sebagai berikut
setwd("D:/PENGEMBANGAN DIRI -- S3 PEP/S3-PEP-Semester
2/Program R/Buku R")
data <- read_xlsx("DATA_ESSAY.xlsx", 1) #angka 1 menunjukkan
sheet 1 pada worksheet excel
data.frame(data)
```

Langkah ketiga, lakukan uji dimensionalitas instrumen atau butir sebagai persyaratan untuk memilih model IRT multidimensi. Pada tahap pemeriksaan dimensi pengukuran, kita perlu memeriksa asumsi kecukupan sampel di dalam analisis faktor eksploratori dapat dinilai menggunakan statistik Kaiser Meyer Olkin test (KMO test), dengan ketentuan jika nilai Measure of Sampling Adequacy (MSA) > 0,5, artinya; asumsi jumlah sampling minimum pada analisis EFA terpenuhi. KMO dan MSA dapat dihasilkan melalui syntax program R sebagai berikut:

```
#Uji Dimensionalitas instrument dapat dilakukan menggunakan
EFA dengan metode parallel.
```

```
library(psych)
x <- cor(data)
KMO <- KMO(x)
MSA_all ='names<-'(KMO$MSA, "Overal All MSA")
MSA_item = 'names<-'(KMO$MSAi, "MSA Per Item")
MSA_all
data.frame(MSA_item)
```

Output R untuk Uji Kecukupan Sampel MIRT Polytomous

```
> MSA_all
Overal All MSA
     0.8336299
```

```
> data.frame(MSA_item)
 MSA_item
1  0.8575324
2  0.8618212
3  0.8378012
4  0.7568501
5  0.5673273
6  0.8305175
7  0.8156353
8  0.8268348
9  0.7888273
10 0.8426296
11 0.8422219
12 0.8423652
13 0.8628654
14 0.7673600
15 0.8639609
16 0.8692347
17 0.8489970
18 0.5699447
19 0.9031983
20 0.8262055
```

Berdasarkan output MSA, baik pada keseluruhan item pengukuran maupun tiap item menunjukkan nilai MSA>0,5, asumsi jumlah sampling minimum pada analisis EFA terpenuhi.

Langkah keempat, memeriksa korelasi antar item pengukuran dalam faktor yang sama menggunakan uji bartlet (Bartlet Tes) Asumsi yang kedua dalam analisis faktor eksploratori adalah korelasi antar sekelompok Item. Analisis EFA mensyaratkan korelasi yang signifikan diantara sekelompok item pengukuran agar item-item tersebut dapat dikelompokkan pada faktor tertentu (faktor yang sama).

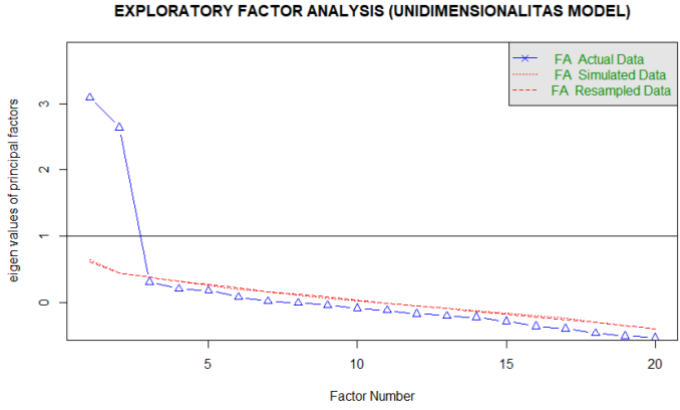
```
bartlet = cortest.bartlett(x, nrow(data))
data.frame(bartlet)
Output Bartlett Sphericity Test
     chisq       p.value  df
```

```
     1163.467    0.00000 190
```

Pengujian korelasi antar item menggunakan Uji Bartlett, jika probabilitas X^2 signifikan (P<0,05), bermakna bahwa item saling berkorelasi signifikan.

Langkah kelima, pemeriksaan dimensi secara visual menggunakan screeplot dan PCA plot. Berdasarkan Gambar Scree Plot, dapat diketahui secara visual inflation poin (perpotongan data dengan quantil ke 95) terjadi setelah titik ke-2, hal ini menunjukkan bahwa dimensi atau aspek yang terbentuk ada 2 buah. Dengan kata lain, item-item dapat dikelompokkan ke dalam 2 dimensi dengan nilai eigen di atas 1. Scree plot dapat dihasilkan dari syntax program R sebagai berikut:

```
EFA_model <- fa(r = data, nfactors = 2, rotate = "varimax", fm
= "mle")
print(EFA_model)
EFA_model$loadings
(plot <- princomp(data, cor = T))  # inappropriate
scree(data, factors = F, main="Scree plot")
library(nFactors)
ev <- eigen(cor(data)) # get eigenvalues
ap <- parallel(subject=nrow(data),var=ncol(data),
               rep=100,cent=.05)
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)
plotnScree(nS)
```

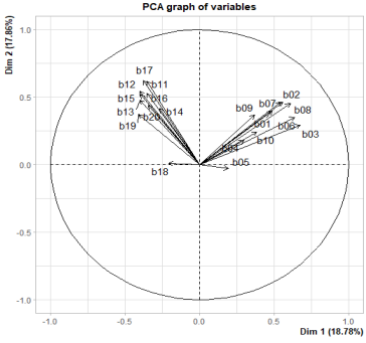


Gambar 9.1 Scree Plot Unidimensionalitas IRT Polytomous

Untuk memperkuat scree plot, kita dapat menampilkan secara visual dimensi pengukuran menggunakan grafik analisis komponen utama (PCA), jika persentase yang menunjukkan porsi pada dimensi 1

dan dimensi 2 hampir seimbang hal itu bermakna bahwa dimensi pengukuran adalah dua dimensi, dan sebaliknya jika porsi pada dimensi 1 mencapai 2 kali lipat atau lebih dari dimensi 1, walaupun eigenvalue lebih dari 1, maka dapat disimpulkan bahwa model pengukuran yang cocok adalah satu dimensi (unidimensi). Grafik PCA dapat diperoleh melalui syntax sebagai berikut:

```
library(FactoMineR)
result <- PCA(data) # graphs generated automatically
```



Gambar 9.2 Grafik PCA

Pada kasus ini tampak secara visual porsi dimensi 1 sebesar 18,78% sedangkan dimensi 2 sebesar 17,86% sehingga dapat disimpulkan bahwa model yang tepat adalah model dua dimensi (multidimensi).

Langkah keenam, menilai kriteria ketepatan model EFA. Kita juga dapat menilai tingkat ketepatan model EFA untuk memverifikasi jumlah dimensi pengukuran pada kasus ini menggunakan indices ketepatan model AIC, SABIC, HQ, BIC dan log likelihood. Kriteria tersebut dapat diperoleh menggunakan syntax berikut ini.

```
Library(mirt)
(mod1 <- mirt(data, 1)) #kita set 1 dimensi
coef(mod1)
summary(mod1)
residuals(mod1)
residuals(mod1, restype = "exp")
(mod2 <- mirt(data, 2)) #kita set 2 dimensi
```

Kita perhatikan output fit dimensi pada model IRT 2 dimensi menggunakan program R berikut ini.

```
# Output tingkat ketepatan EFA pada model dua dimensi.
Iteration: 104, Log-Lik: -6324.542, Max-Change: 0.00009
Call:
mirt(data = data, model = 2)
Full-information item factor analysis with 2 factor(s).
Converged within 1e-04 tolerance after 104 EM iterations.
mirt version: 1.36.1
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 31
Latent density type: Gaussian
Log-likelihood = -6324.542
Estimated parameters: 117
AIC = 12883.08
BIC = 13295.09; SABIC = 12924.19
G2 (1e+10) = 9888.35, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

Tampak nilai AIC, SABIC, HQ, BIC dan log likelihood pada dimensi 2 menunjukkan nilai yang lebih kecil jika dibandingkan dengan dimensi 1, walaupun probabilitas chi-square signifikan (tidak fit), namun kriteria-kriteria ketepatan model yang lain dapat direkomendasikan sebagai kriteria yang fit untuk memilih dua dimensi.

Langkah ketujuh, melakukan rotasi faktor. Untuk melihat faktor loading setelah dilakukan rotasi varimax menggunakan syntax berikut ini.

```
summary(mod2, rotate = "varimax", suppress = 0.25)
```

Output Loading Faktor setelah dirotasi menggunakan metode Varimax.

```
Rotation:  varimax
Rotated factor loadings:

       F1    F2     h2
b01    NA 0.644 0.4146
b02    NA 0.738 0.5468
b03    NA 0.750 0.5763
b04    NA 0.289 0.0846
b05    NA    NA 0.0280
b06    NA 0.681 0.4692
b07    NA 0.753 0.5695
b08    NA 0.762 0.5812
b09    NA 0.495 0.2470
b10    NA 0.419 0.1773
b11 0.710    NA 0.5100
b12 0.668    NA 0.4466
b13 0.582    NA 0.3405
```

```
b14 0.447    NA 0.1997
b15 0.748    NA 0.5619
b16 0.680    NA 0.4620
b17 0.748    NA 0.5604
b18    NA    NA 0.0353
b19 0.502    NA 0.2703
b20 0.528    NA 0.2793
Rotated SS loadings:  3.654 3.707
Factor correlations:
        F1      F2
F1      1       0
F2      0       1
```

Berdsasarkan output program R di atas, tampak pada dimensi 1 (F1) hanya item nomor 14 memiliki loading <0,5. Pada dimensi 2, item nomor 4, item nomor 5, item nomor 9 dan item nomor 10 memiliki loading faktor <0,5, item-item tersebut dapat dibuang atau dikalibrasi. Namun pada kasus ini, kita sengaja tidak membuang item-item tersebut, karena pada contoh ini kita tidak bermaksud melakukan kalibrasi atau pengembangan instrumen.

9.4.2 Analisis IRT Multidimensi Politomi

Langkah pertama, mengkonstruksi Model Multidimensi (tentukan faktor dan item pengukuran tiap faktor). Pembaca dapat menyesuikan variabel atau faktor yang digunakan, pada kasus ini menggunakan dua buah faktor dalam pembelajaran metode statistika, yaitu; deskripitif dan inferensial. Konstruksi model dapat dilakukan dengan menulis syntax program R berikut ini.

```
model_multidim <- 'Deskriptif = 1 - 10
          Inferensial = 11 - 20
          COV = Deskriptif * Inferensial'
```

Langkah kedua, memilih model IRT multidimensi politomi yang terbaik (model yang paling tepat). Pemilihan model IRT multidimensi politomi Sesuai Skala Data (Misal; Likert Gunakan GPCM, GRM dan polytomous Rasch model, dan data nominal gunakan MNRM). Jalankan syntax program berikut ini.

```
mgpcm_fit <- mirt(data = data, model = model_multidim,
                  itemtype = "gpcm", SE = TRUE)
mgrm_fit <- mirt(data = data, model = model_multidim,
                  itemtype = "graded", SE = TRUE)
#Keterangan: MPCM adalah bentuk lain dari Rasch Multidimensi
mpcm_fit <- mirt(data = data, model = model_multidim,
                   itemtype = "Rasch", SE = TRUE)
```

```
#Model Multidimensional Nominal Response Model (MNRM)
sebaiknya jangan di run jika data yang digunakan bukan tipe
nominal.

mnrm_fit <- mirt(data = data, model = model_multidim,
                 itemtype = "nominal", SE = TRUE)
```

Kita gunakan Kriteria Kecocokan Model untuk memilih model yang paling cocok dengan data. Jalankan syntax program R berikut ini.

```
data.frame (Model  = c("MGPCM","MGRM","MPCM","MNRM"),
            AIC = c(mgpcm_fit@Fit$AIC, mgrm_fit@Fit$AIC,
mpcm_fit@Fit$AIC, mnrm_fit@Fit$AIC),
            p.M2 = c(M2(mgpcm_fit)$p, M2(mgrm_fit)$p,
M2(mpcm_fit)$p, M2(mnrm_fit)$p),
            RMSEA = c(M2(mgpcm_fit)$RMSEA, M2(mgrm_fit)$RMSEA,
M2(mpcm_fit)$RMSEA, M2(mnrm_fit)$RMSEA),
            TLI = c(M2(mgpcm_fit)$TLI, M2(mgrm_fit)$TLI,
M2(mpcm_fit)$TLI, M2(mnrm_fit)$TLI),
            CFI = c(M2(mgpcm_fit)$CFI, M2(mgrm_fit)$CFI,
M2(mpcm_fit)$CFI, M2(mnrm_fit)$CFI))
```

Berdasarkan hasil analisis menggunakan software R studio pada syntax yang telah kita tampilkan sebelumnya, diperoleh hasil sebagai berikut.

```
        Model   AIC             p.M2    RMSEA   TLI     CFI
1       MGPCM   12894.000       0.159   0.023   0.987   0.989
2       MGRM    12870.420       0.108   0.026   0.984   0.986
3       MPCM    13145.700       0.266   0.017   0.993   0.993
4       MNRM    12930.340       0.515   0.000   1.002   1.000
```

Berdasarkan ouput kriteria kecocokan model di atas, dari 3 model yang diusulkan, semua model fit dengan data, namun model terbaik adalah Multidimensional Graded Response Model (MGRM) karena memiliki AIC terkecil, dengan probabilitas Chi-Square (p.M2) >0,05, RMSEA<0,08, TLI dan CFI>0,95. Selanjutnya, kita juga perlu menilai ketepatan model pada masing-masing item menggunakan beberapa model IRT Multidimensi. Secara default program R menyediakan statistik Chi-Square (S_X2), derajat bebas Chi-Square (df.S_X2), Root Mean Square Error Approximation (RMSEA), dan probabilitas Chi-Square (p.S_X2).

Langkah keempat, kita lakukan penilaian model berdasarkan parameter secara keseluruhan pada tiap model yang di jalankan.

```
mgpcm_parameters <- coef(mgpcm_fit, simplify = TRUE)
mgrm_parameters <- coef(mgrm_fit, simplify = TRUE)
```

```
mpcm_parameters <- coef(mpcm_fit, simplify = TRUE)
mnrm_parameters <- coef(mnrm_fit, simplify = TRUE)
round(head(mgpcm_parameters$items),3)
round(head(mgrm_parameters$items),3)
round(head(mpcm_parameters$items),3)
round(head(mnrm_parameters$items),3)
```

Ouput fit parameter model GMPCM, MGRM, MGPCM dan MNRM menggunakan R Studio dalam contoh ini menggunakan adalah sebagai berikut.

```
> round(head(mgpcm_parameters$items),3)
       a1 a2 ak0 ak1 ak2 ak3 d0     d1     d2     d3 ak4     d4
b01 0.957  0   0   1   2   3  0  2.021  1.543  0.718  NA     NA
b02 0.909  0   0   1   2   3  0  0.016 -0.471 -0.261   4 -1.996
b03 1.101  0   0   1   2   3  0 -2.036  1.835  2.344   4  1.492
b04 0.186  0   0   1   2   3  0 -0.053 -2.965 -0.882   4 -0.377
b05 0.083  0   0   1   2   3  0 -2.600 -2.400 -0.123  NA     NA
b06 0.840  0   0   1   2   3  0  1.336  0.714 -0.098   4  0.099
> round(head(mgrm_parameters$items),3)
       a1 a2    d1     d2     d3     d4
b01 1.426  0 2.762 -0.234 -1.899     NA
b02 1.830  0 1.121 -0.266 -1.108 -3.321
b03 1.876  0 2.317  2.237  0.590 -1.695
b04 0.513  0 0.777 -0.521 -0.594 -1.264
b05 0.208  0 0.048 -0.098 -0.277     NA
b06 1.588  0 2.369 -0.014 -1.075 -1.802
> round(head(mpcm_parameters$items),3)
    a1 a2 ak0 ak1 ak2 ak3 d0     d1     d2     d3 ak4     d4
b01  1  0   0   1   2   3  0  1.592  1.095  0.650  NA     NA
b02  1  0   0   1   2   3  0 -0.257 -0.753 -0.300   4 -1.524
b03  1  0   0   1   2   3  0 -2.776  0.690  1.145   4  0.690
b04  1  0   0   1   2   3  0  0.093 -2.804 -0.831   4 -0.560
b05  1  0   0   1   2   3  0 -2.454 -2.267 -0.161  NA     NA
b06  1  0   0   1   2   3  0  1.032  0.340 -0.329   4  0.225
> round(head(mnrm_parameters$items),3)
       a1 a2 ak0    ak1   ak2   ak3 d0     d1     d2     d3 ak4     d4
b01 0.966  0   0  1.136 2.216 3.000  0  2.155  1.646  0.940  NA     NA
b02 0.994  0   0  1.094 2.229 2.600  0  0.100 -0.497 -0.056   4 -2.213
b03 1.050  0   0  0.945 1.755 2.743  0 -2.178  1.557  2.099   4  1.133
b04 0.198  0   0  1.934 4.249 2.702  0 -0.021 -3.032 -0.839   4 -0.340
b05 0.081  0   0 -0.395 1.772 3.000  0 -2.611 -2.400 -0.123  NA     NA
b06 0.822  0   0  0.816 1.870 3.048  0  1.225  0.628 -0.237   4 -0.014
```

Langkah kelima, kita nilai model berdasarkan jumlah item yang cocok. Kita dapat memilih salah satu model yang paling cocok dari seluruh model yang dijalankan berdasarkan kriteria ketepatan model keseluruan, parameter, dan item fit.

Item fit menggunakan MGPCM, MGRM, MPCM dan MNRM dapat dapat kita peroleh dengan menjalankan syntax R berikut ini.

Output yang dihasilkan oleh R Studio pada penilaian parameter item MIRT Politomi (item fit) adalah sebagai berikut:

```
> itemfit(mgpcm_fit)[order(itemfit(mgpcm_fit)$p.S_X2),]
   item    S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
3   b03 119.583      71      0.052  0.000
11  b11  92.009      63      0.043  0.010
2   b02  85.756      67      0.034  0.061
7   b07  56.450      43      0.035  0.082
9   b09  91.240      74      0.031  0.085
13  b13  85.943      71      0.029  0.109
8   b08  59.716      48      0.031  0.120
17  b17  85.676      72      0.028  0.129
5   b05  40.086      32      0.032  0.154
6   b06  78.610      68      0.025  0.178
10  b10  81.473      71      0.024  0.186
14  b14  72.577      63      0.025  0.192
12  b12  63.528      57      0.021  0.257
20  b20  73.147      69      0.016  0.344
18  b18  67.155      66      0.008  0.437
1   b01  50.904      51      0.000  0.477
15  b15  31.473      32      0.000  0.493
19  b19  59.301      63      0.000  0.609
16  b16  42.088      46      0.000  0.637
4   b04  62.536      74      0.000  0.826
```

Kita juga dapat mengeluarkan item fit untuk IRT dengan model respons politomi berjenjang multidimensi (*graded respons model*, MGRM).

```
> itemfit(mgrm_fit)[order(itemfit(mgrm_fit)$p.S_X2),]
   item    S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
3   b03 118.313      71      0.052  0.000
11  b11  90.015      63      0.041  0.014
9   b09  95.366      75      0.033  0.056
2   b02  84.259      66      0.033  0.064
7   b07  57.592      44      0.035  0.082
13  b13  88.528      73      0.029  0.104
6   b06  83.138      70      0.027  0.135
17  b17  84.855      72      0.027  0.143
5   b05  40.070      32      0.032  0.155
10  b10  82.624      71      0.026  0.163
8   b08  56.366      48      0.026  0.191
14  b14  63.870      57      0.022  0.248
12  b12  58.764      56      0.014  0.375
20  b20  74.189      71      0.013  0.375
18  b18  67.698      66      0.010  0.419
1   b01  50.801      51      0.000  0.482
15  b15  26.780      28      0.000  0.530
19  b19  59.347      62      0.000  0.572
16  b16  42.166      45      0.000  0.593
4   b04  63.638      74      0.000  0.799
```

Selain MGPCM dan MGRM, melalui syntax R Studio, otomatis kita juga dapat mengeluarkan item fit untuk IRT multidimensi polytomous Rasch Polytomous (MPCM)

```
> itemfit(mpcm_fit)[order(itemfit(mpcm_fit)$p.S_X2),]
   item    S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
5   b05  65.957      29      0.072  0.000
3   b03 116.965      75      0.047  0.001
11  b11 101.034      64      0.048  0.002
18  b18  83.416      54      0.047  0.006
2   b02  93.666      66      0.041  0.014
8   b08  63.573      47      0.038  0.054
9   b09  93.343      73      0.033  0.055
10  b10  91.044      72      0.033  0.064
6   b06  87.793      70      0.032  0.074
17  b17  88.075      72      0.030  0.096
14  b14  34.010      25      0.038  0.108
13  b13  89.312      74      0.029  0.108
7   b07  53.636      45      0.028  0.177
12  b12  67.148      61      0.020  0.275
1   b01  61.516      56      0.020  0.285
20  b20  74.506      69      0.018  0.304
15  b15  34.830      32      0.019  0.335
19  b19  59.753      62      0.000  0.557
16  b16  41.758      45      0.000  0.610
4   b04  62.004      67      0.000  0.650
```

```
> itemfit(mnrm_fit)[order(itemfit(mnrm_fit)$p.S_X2),]
   item    S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
3   b03 121.932      70      0.055  0.000
11  b11  87.941      60      0.043  0.011
7   b07  57.475      40      0.042  0.036
2   b02  83.090      62      0.037  0.038
9   b09  91.226      70      0.035  0.045
17  b17  85.725      68      0.032  0.072
8   b08  60.435      46      0.035  0.075
14  b14  33.086      23      0.042  0.080
10  b10  83.346      68      0.030  0.100
6   b06  81.082      66      0.030  0.100
5   b05  40.105      30      0.037  0.103
13  b13  78.867      65      0.029  0.116
19  b19  63.561      58      0.020  0.287
12  b12  58.168      53      0.020  0.291
20  b20  70.655      66      0.017  0.325
15  b15  26.607      25      0.016  0.376
1   b01  50.849      49      0.012  0.401
18  b18  65.004      64      0.008  0.442
16  b16  40.060      42      0.000  0.556
4   b04  64.006      71      0.000  0.709
```

Berdasarkan *output* Program R di atas dapat diketahui bahwa pada model MGPCM dan MGRM ada 2 item yang tidak fit, yaitu; b03 dan b11, karena memiliki P<0,05. Pada rasch polytomous (MPCM) terdapat

5 item yang tidak fit, yaitu; b05, b03, b11, b18 dan b02. Berdasarkan beberapa kriteria pemilihan model pada di atas, maka kita dapat memutuskan bahwa model yang tepat untuk instrumen dalam contoh kasus ini adalah MGRM.

Langkah keenam, Menilai tingkat kesukaran butir multidimensi (MDIFF) dan daya pembeda multidimensi (MDISC). Dua deskripsi statistik yaitu; MDIFF dan MDISC, digunakan untuk memvisualisasikan karakteristik item dalam model MIRT. Untuk membandingkan pola pemilihan item di antara metode pemilihan item, indeks diskriminasi item (MDISC) selama perkembangan tes dicatat. Secara geometris, MDISC adalah kemiringan paling curam pada permukaan respons item.

Menurut Reise dan Yu (1990), tiga tingkat tingkat diskriminasi item ditentukan untuk GRM dan GPCM yang mencakup item dengan kualitas buruk, sedang, dan baik. Secara khusus, nilai diskriminasi item dipilih secara acak dari distribusi seragam U (0,44, 0,75) untuk item buruk, U (0,58, 0,98) untuk item sedang, dan U (0,75, 1,33) untuk item baik.

MDISC digunakan untuk menggambarkan daya pembeda multi-dimensi dari item tes, memiliki interpretasi yang tidak berbeda dengan parameter a dalam model teori respons item uni-dimensi yang menyatakan daya pembeda item tes sebagai arah dan jarak dalam ruang laten yang lengkap, sedangkan MDIFF digunakan untuk menggambarkan kesulitan multi-dimensi dari item tes, memiliki interpretasi yang tidak berbeda dengan parameter b dalam model teori respons item uni-dimensi yang menyatakan kesulitan item tes sebagai arah dan jarak dalam ruang laten yang lengkap (Reckase, 1985). MDISC dan MDIFF dapat dihasilkan dari syntax R berikut ini.

```
MDISC <- round(MDISC(mgrm_fit),3)
MIDFF <- round(MDIFF(mgrm_fit),3)
MDISC_MDIFF <- cbind(MDISC,MIDFF)
```

Output MDIFF dan MDISC MIRT Politomi menggunakan R studio adalah sebagai berikut.

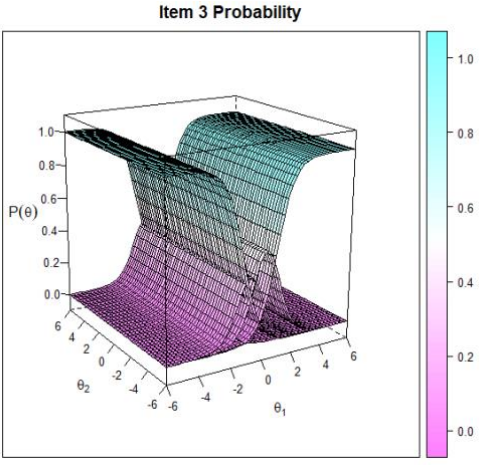
```
> MDISC_MDIFF
    MDISC MDIFF_1 MDIFF_2 MDIFF_3 MDIFF_4
b01 1.426  -1.936   0.164   1.331      NA
b02 1.830  -0.613   0.145   0.606   1.815
b03 1.876  -1.235  -1.193  -0.315   0.904
b04 0.513  -1.514   1.016   1.156   2.461
b05 0.208  -0.231   0.470   1.334      NA
b06 1.588  -1.491   0.009   0.677   1.135
b07 1.968   0.205   0.470   0.982   1.151
b08 1.993  -1.335   0.246   0.372   1.696
b09 0.962  -1.389  -1.337  -0.233   1.259
b10 0.793  -0.816   0.176   0.881   1.241
b11 1.713  -1.651  -0.804  -0.038   0.129
b12 1.529  -1.461   0.281   0.693   1.498
b13 1.210  -1.788  -1.699  -0.403   0.472
b14 0.842  -1.891  -0.995   1.024   2.122
b15 1.906  -1.440  -1.368  -1.111   1.564
b16 1.556  -1.863  -1.440   1.093   1.630
b17 1.936  -0.649   0.025   0.753   0.958
b18 0.188  -3.629  -2.156   0.698   2.906
b19 0.996  -0.868  -0.276   1.431   1.725
b20 1.067  -1.836   0.119   0.675   1.566
```

```
> round(mgrm_parameters$cov,3)
            Deskriptif Inferensial
Deskriptif       1.000      -0.032
Inferensial     -0.032       1.000
```

Kasus ini menggunakan soal *essay* dengan 4 kategori skor, yaitu; 0, 1, 2, 3 dan 4. Indeks pembeda multidimensi (MDISC) yang dihasil oleh program R bervariasi antara 0,188 sampai 1,968.

Langkah ketujuh, Menilai Item Plot (Peluang menjawab benar pada kemampuan tertentu, fungsi informasi tes, kesalahan standar tes, dan total skor yang diharapkan). Pada kasus ini kita gunakan item nomor 3 sebagai contoh untuk menggambar pola yang menghubungkan antara peluang peserta menjawab pada kemampuan theta. Menggunakan fungsi item plot dengan opsi type = "trace", pertama-tama kita membuat plot permukaan karakteristik item untuk item 3. Ini adalah plot tiga dimensi dengan sumbu berikut: sifat laten pertama pada sumbu x, sifat laten kedua pada sumbu y, dan probabilitas menjawab item 3 dengan benar pada sumbu z. Kemudian, kita membuat kontur item plot untuk item yang sama dengan mengubah tipe pada syntax menjadi type = "tracecontour".

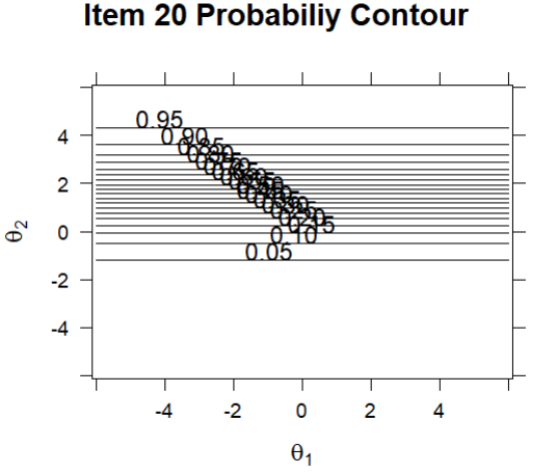
```
itemplot(mgrm_fit, type = "trace", item = 3)
```



Gambar 9.3 Kurva Probilitas Item 3

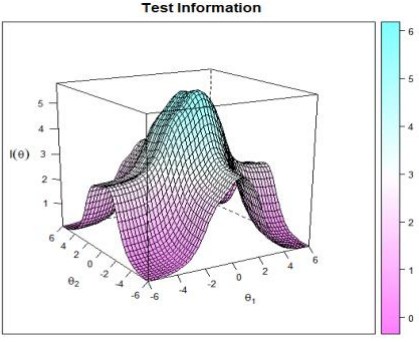
Hal ini menunjukkan bahwa ada beberapa permukaan di setiap plot permukaan item karena kedua item memiliki beberapa kategori respons. Setiap permukaan menampilkan kemungkinan memilih kategori respons tertentu dalam item. Grafik kontur probabilitas item dapat dihasilkan melalui syntax berikut ini.

```
itemplot(mgrm_fit, type = "tracecontour", item = 3)
```



Gambar 9.4 Grafik Kontur Probabilitas Item Nomor 20

Plot informasi item, dalam fungsi item plot yang dihasilkan mirip dengan plot permukaan karakteristik item, tetapi kali ini sumbu z menunjukkan tingkat informasi item alih-alih probabilitas keberhasilan item. Gambar 9.5 menunjukkan bahwa tingkat informasi item tertinggi ketika kedua sifat laten sekitar nol, sedangkan informasi item terendah ketika kedua sifat laten menjadi sangat rendah atau sangat tinggi.

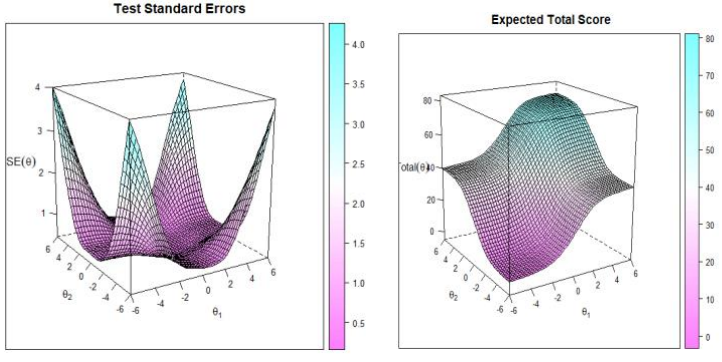


Gambar 9.5 Kurva Informasi Item pada Item Nomor 3

Selain plot tingkat item yang diringkas di atas, juga dimungkinkan untuk menggambar plot pada tingkat tes. Plot ini merangkum fungsi informasi pengujian dan kesalahan standar bersyarat pengukuran. Syntax berikut ini menunjukkan penggunaan fungsi plot untuk membuat plot TIF dan plot *conditional standard error measurement* (cSEM).

```
plot(mgrm_fit, type = "SE", item = 3)
plot(mgrm_fit, type = "score", item=3)
```

Ouput yang dihasilkan R studio adalah berupa grafik dua dimensi.



Gambar 9.6 dan Gambar 9.7 SEM dan Ekspektasi Skor Total

Grafik pertama (Gambar 9.6) menunjukkan hubungan antara kemampuan peserta tes pada item nomor 3 tampak secara visual bahwa testee dengan kemampuan antar -2 sampai 2 menghasilkan standar error yang lebih rendah, baik pada dimensi 1 maupun dimensi 2 (ditunjukkan dengan θ_1 dan θ_2 . Hal ini bermakna bahwa item soal tes nomor 3 cocok diterapkan pada seperta tes dengan kemampuan antara -2 sampai 2 (kemampuan tergolong sedang). Grafik skor total yang diharpkan (Gambar 9.7) menunjukkan skor total yang diharapkan (dalam skala 0-100) dapat dicapai oleh sebagian besar peserta tes untuk butir 3 pada dimensi kemampuan 1 (θ_1) maupun dimensi kemampuan 2 (θ_2) adalah 40 (dalam skala 100), sedangkan skor maksimum yang diperoleh peserta tes pada butir 3 adalah 80.

Langkah kedelapan, menilai peserta yang cocok dengan model yang dipilih. Gunakan syntax R berikut ini untuk menghasilan person fit.

```
test_person_fit <- round(personfit(mgrm_fit),3)
test_person_fit
person_fit <- data.frame(test_person_fit)
person_fit
```

Kita dapat mengetahui kecocokan model MGRM pada masing-masing peserta tes melalui person fit, kriteria fit adalah outfit, z.outfit, infit, z.infit dan Zh. Infit adalah inovasi dari Ben Wright (Rasch, 1993). Ben memperhatikan bahwa statistik fit statistik standar (yang sekarang kita sebut Outfit) sangat dipengaruhi oleh beberapa outlier (pengamatan yang sangat tidak terduga). Ben membutuhkan statistik fit yang lebih sensitif terhadap keseluruhan pola respons, jadi dia merancang Infit. Infit menimbang pengamatan dengan informasi statistiknya (varian model) yang lebih tinggi di tengah tes dan lebih rendah di titik ekstrem. Efeknya adalah membuat Infit tidak terlalu terpengaruh oleh outlier, dan lebih sensitif terhadap pola observasi inlying. Statistik Infit and Outfit Ben awalnya dihitung sebagai statistik rata-rata (yaitu, statistik chi-kuadrat dibagi dengan derajat kebebasannya).

```
> person_fit
    outfit z.outfit infit z.infit     Zh
1    0.384   -2.040 0.442  -2.605  0.329
2    0.844   -0.254 0.759  -0.864  1.427
3    1.176    0.613 1.246   0.984  0.198
4    0.815   -0.258 0.730  -0.953  1.093
5    0.788   -0.419 0.686  -1.137 -0.062
6    0.702   -0.682 0.746  -0.882  0.038
7    0.501   -0.895 0.692  -0.918  1.204
8    0.933   -0.146 1.023   0.175 -0.965
9    0.667   -0.954 0.741  -1.073  1.360
10   0.911   -0.123 0.786  -0.744 -0.387
...
```

Indeks person fit Zh dengan yang rendah (misalkan < -3) menunjukkan potensi pola respons yang menyimpang. Pada tabel di atas, peserta (testee) nomor 10 <-3, testee tersebut memiliki pola respons yang menyimpang. kelompok testee yang memiliki nilai Zh < -3 adalah person yang responnya tidak cocok dengan pemodelan MGRM pada contoh ini.

Langkah kesembilan, Menilai faktor skor (jika item merupakan skala Likert) atau kemampuan peserta tes (jika item adalah soal essay)

```
ability_map <- fscores(mgrm_fit, method = "MAP")
ability_eap <- fscores(mgrm_fit, method = "EAP")
ability <- cbind(ability_map, ability_eap)
ability1 <-round(data.frame(ability),3)
ability1
```

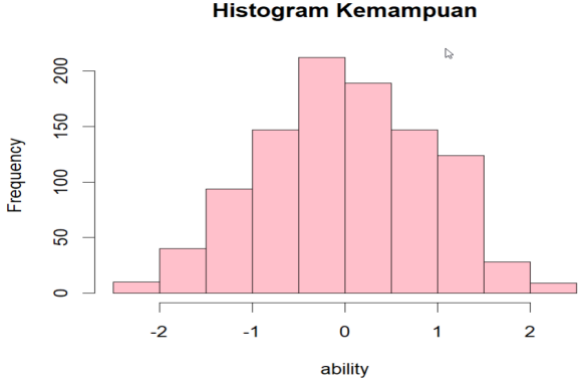
Untuk model MIRT, kita dapat mengikuti prosedur yang sama untuk memperkirakan tingkat sifat laten dari penilaian multidimensi. Kita menggunakan maksimum a posteriori (MAP) dalam contoh ini. Kita mengharapkan metode posteriori (EAP) untuk memperkirakan tingkat sifat laten dari model MGRM. Estimasi berbasis ML juga dapat diperoleh dengan menuliskan method= " ML" pada syntax. Namun, hal yang perlu diperhatikan adalah estimasi kemungkinan maksimum (ML) gagal memberikan hasil ketika semua respons item salah atau benar untuk peserta ujian, sebagai alternatifnya kita gunakan EAP dan MAP. Masalah yang sama tetap ada ketika memperkirakan tingkat sifat laten dari model MIRT.

```
> ability1
    Deskriptif Inferensial Deskriptif.1 Inferensial.1
1        1.359       0.772        1.372         0.771
2       -0.945       0.825       -0.966         0.822
3       -0.362       0.248       -0.368         0.256
4       -1.272       0.016       -1.294         0.012
5       -0.825       1.227       -0.841         1.240
6       -0.283       1.598       -0.288         1.628
7        2.140       1.274        2.212         1.287
8        0.017      -0.754        0.010        -0.750
9        1.155       0.507        1.162         0.513
10      -0.305       1.369       -0.320         1.388
...
```

Kita juga dapat mengetahui distribusi kemampuan peserta tes secara visual melalui histogram ability yang dikeluarkan oleh Program R melalui syntax berikut ini.

```
hist.default(ability, main = "Histogram Kemampuan", col =
"yellow")
## End(Not run)
```

Sebaran kemampuan peserta dapat dilihat pada grafik histogram, gambar histogram yang relatif membentuk lonceng, dan tidak terlalu menceng kekiri atau kekanan menunjukkan bahwa distribusi kemampuan peserta menyebar mengikuti distribusi normal. distribusi normal bivariat dengan berbagai mean dan varians/kovarians dianggap menggambarkan kemampuan sebenarnya dari peserta ujian. Distribusi kemampuan yang berbeda berarti bahwa formulir tes diberikan kepada populasi yang berbeda. Berdasarkan histogram kemampuan peserta tes pada Gambar 9.8, secara visual dapat kita ketahui bahwa sebagian besar peserta memiliki kemampuan yang menyebar disekitar -1 dan +1.



Gambar 9.8 Histogram Kemampuan Peserta

Referensi

Ackerman, T. A. (1994). Using multidimensional item response theory to understand what items and tests are measuring. *Applied Measurement in Education*, *7*(4), 255–278.

Adams, T., Bezner, J., & Steinhardt, M. (1997). The conceptualization and measurement of perceived wellness: Integrating balance across and within dimensions. *American Journal of Health Promotion*, *11*(3), 208–218. https://doi.org/10.4278/0890-1171-11.3.208

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists. Maheah*. New Jersey: Lawrence Erlbaum Associates, Publishers.

Jin, K.-Y., & Chen, H.-F. (2020). MIMIC approach to assessing differential item functioning with control of extreme response style. *Behavior Research Methods*, *52*(1), 23–35. https://doi.org/10.3758/s13428-019-01198-1

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychome-Trika*, *47*(2), 149–174. https://doi.org/10.1007/BF02296272

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*(2), 159–176. https://doi.org/10.1177/014662169201600206

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests.* ERIC.

Reckase, M. D. (1985). *Models for Multidimensional Tests and Hierarchically Structured Training Materials.* American coll testing program iowa city ia test development div.

Chapter 10
Equating

Oleh: Okky Riswandha Imawan & Heri Retnawati

Equating Tes Secara Klasik

Ketika mengoneksikan skor tes yang memiliki beberapa bentuk yang dikonstruk dengan spesifikasi yang sama, Angoff menerapkan istilah penyetaraan (*equiting*) (Brennan & Kolen, 2004). Merujuk pada pendapat Mislevy & Linn yang menyatakan bahwa equating merupakan usaha untuk menghubungkan skor yang berbeda dalam suatu asesmen. Sementara itu, menurut Brennan & Kolen (2004), skor tes dihubungkan dengan menggunakan kalibrasi, di mana antara tes memiliki kesamaan konstruk yang diukur, meskipun reliabilitas atau tingkat kesulitannya memiliki perbedaan.

Menyamakan adalah prosedur statistik yang biasa digunakan dalam program pengujian di mana administrasi di lebih dari satu kesempatan dan lebih dari satu kelompok peserta ujian dapat menyebabkan *overexposure* butir, mengancam keamanan tes. Tujuan menyamakan adalah untuk menyesuaikan perbedaan-perbedaan ini dalam kesulitan di seluruh bentuk tes alternatif, sehingga menghasilkan yang sebanding skala skor (Albano, 2016).

Equating dilakukan karena kekhawatiran tentang keamanan tes, yaitu masalah seputar paparan item. Selain itu dalam *equating*, prosedur statistik yang dikenal sebagai persamaan, biasanya digunakan untuk melakukan penyesuaian nilai tes dari bentuk tes yang berbeda. Menyamakan serta menyesuaikan variasi dalam kesulitan bentuk tes dan memungkinkan skor pada berbagai bentuk tes untuk dipertukarkan dan lebih sebanding (Desjardins & Bulut, 2017).

Menempatkan skor dari dua tes pada skala yang sama adalah tujuan dari penyetaraan (Retnawati, 2014). Skor dari dua instrumen atau perangkat tes dapat disetarakan dengan memperhatikan kondisi tes yang mengukur kemampuan yang berbeda tidak dapat disetarakan. Lebih tegas lagi, Hambleton, Swaminathan & Rogers (1991) menyatakan bahwa skor dari tes-tes yang reliabilitasnya tidak sama

tidak dapat disetarakan, akan tetapi skor pada tes-tes yang bervariasi tingkat kesukarannya dapat disetarakan.

Adapun Lord (Hambleton dan Swaminathan, 1985) menjelaskan bahwa terdapat 4 prinsip dasar penyetaraan, sebagai berikut: (1) Prinsip kesetaraan (*equity*), yaitu setiap kelompok peserta tes dengan kemampuan yang sama, kondisi distribusi frekuensi skor pada tes Y setelah ditransformasi sama dengan distribusi frekuensi skor pada tes X, (2) Prinsip invariansi populasi, yang berarti hubungan penyetaraan pada transformasi tidak lagi memperhatikan kelompok populasi yang digunakan, (3) Prinsip simetri, artinya penyetaraan dapat dilakukan dua arah, tanpa memperhatikan pemberian label, (4) Prinsip unidimensi, yaitu perangkat tes yang disetarakan harus mengukur kemampuan yang sama.

Paket R (Albano, 2016) berisi fungsi untuk penautan skor-observasi dan menyamakan di bawah grup tunggal, grup setara, dan grup tidak setara dengan jangkar uji dan desain kovariat. Selanjutnya perlu ditekankan kepada pembaca bahwa *Chapter* ini hanya membahas tentang equating tes secara klasik saja.

Berikut ini sintaks untuk membaca library untuk melakukan equating pada R Studio. Library (equate) untuk dapat melakukan equating pada R Studio dan library (xlsx) untuk dapat membaca file excel yang berupa xlsx yang akan digunakan sebagai sumber data paket instrument yang akan diequating, jika sumber data berbentuk lain (bukan berupa file xlsx) maka perlu menggunakan library yang lain tergantung jenis datanya, misalnya csv untuk perangkat MacBook, dan sebagainya.

```
#membaca library
library(equate)
library(xlsx)
```

Berikut ini sintaks untuk membaca data hasil tes paket a yang akan di equating. Pada contoh ini data berupa file xlsx dan data peket a terdapat pada sheet 1, sehingga terdapat keterangan “1” setelah nama file excelnya.

```
#membaca data paket a
a <- read.xlsx('PAKET TES.xlsx', 1, header = T)
a <- data.frame(a)
```

```
a
```

Berikut ini hasil membaca data paket a. Data yang dibutuhkan hanya berupa sampel, misalkan pada contoh ini terdiri dari 58 sampel atau responden. Selanjutnya jumlah butir instrument beserta skornya juga dibutuhkan, misalkan pad contoh ini ada 10 butir instrumen. Terakhir, dibutuhkan pula skor total untuk setiap sampel.

Tabel 10.1 Data Paket A

Sampel	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_10	Skor Total
Siswa_1	1	1	1	1	0	0	1	0	1	1	7
Siswa_2	1	1	1	0	0	0	1	1	0	0	5
Siswa_3	1	1	1	0	0	1	1	0	0	1	6
Siswa_4	0	1	0	1	1	0	1	0	0	1	5
Siswa_5	1	0	0	1	1	1	1	0	1	1	7
Siswa_6	0	1	1	1	1	1	1	1	0	0	7
Siswa_7	0	0	0	0	1	1	0	0	0	1	3
Siswa_8	1	1	1	1	0	0	1	0	1	0	6
Siswa_9	0	0	0	0	0	0	0	0	1	0	1
Siswa_10	1	1	1	1	1	0	0	1	0	1	7
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
Siswa_58	0	1	0	0	0	1	0	0	0	1	3

Sementara itu berikut ini hasil membaca data ra (frekuensi skor paket a). Data ra ini hanya membutuhkan frekuensi skor dari sampel dan skor sampel terdapat pada kolom 12 pada file excel, sehingga sintaksnya sebagai berikut.

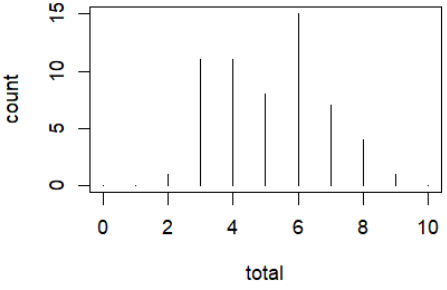
```
ra <- freqtab(a[,12], scales = 0:10)
ra
```

Tabel 10.2 Data ra

total	count
0	0
1	0
2	1
3	11
4	11
5	8
6	15
7	7
8	4
9	1
10	0

Pada Tabel di atas, dapat diketahui bahwa sampel yang mendapatkan skor 0 tidak ada, skor 1 juga tidak ada, skor 2 terdapat 1 sampel, skor 3 terdapat 11 sampel, dan seterusnya. Berikut ini sintaks dan hasil plot data ra, yang hasilnya menyerupai distribusi normal.

```
plot(ra)
```



Gambar 10.1 hasil plot ra

Selanjutnya sintaks untuk membaca data hasil tes paket b yang akan di*equating*, serta sintaks untuk mendapatkan rb dan plotnya. Sekali lagi, pada contoh ini data berupa file xlsx, dan untuk paket b misalkan berada pada sheet ke 2, sehingga harus dituliskan "2" setelah nama file xlsx.

```
#membaca data paket b
b <- read.xlsx('PAKET TES.xlsx', 2, header = T)
b <- data.frame(b)
b
```

Berikut ini hasil membaca data paket b. Secara keseluruhan data paket b hampir sama dengan paket a, yaitu terdiri dari 10 butir. Misalkan kedua paket instrument direspon oleh sampel yang berbeda, misalkan paket b dikerjakan oleh 60 sampel yang berbeda dengan paket a (dalam contoh ini paket a dikerjakan oleh 58 sampel).

Tabel 10.3 hasil membaca data paket b

Sampel	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_10	Skor Total
Siswa_1	0	1	1	1	0	0	1	0	1	0	5
Siswa_2	0	0	1	0	1	0	1	0	1	0	4
Siswa_3	0	0	1	1	1	1	1	0	0	0	5
Siswa_4	1	0	0	0	1	1	1	1	0	1	6
Siswa_5	0	0	0	0	1	0	1	0	0	0	2
Siswa_6	1	1	1	1	0	0	0	0	1	1	6
Siswa_7	1	0	1	1	1	0	0	0	0	1	5
Siswa_8	0	0	1	0	0	0	0	0	0	0	1
Siswa_9	1	1	1	0	1	1	1	1	0	1	8

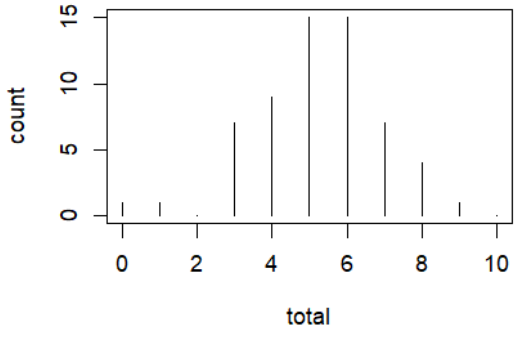
```
Siswa_1
   0      0    1    0    0    0    0    1    1    1    0       4
   .      .    .    .    .    .    .    .    .    .    .       .
   .      .    .    .    .    .    .    .    .    .    .       .
   .      .    .    .    .    .    .    .    .    .    .       .
Siswa_6
   0      0    1    0    1    0    1    1    1    1    1       7
```

```
rb <- freqtab(b[,12], scales = 0:10)
rb
```

Tabel 10.4 hasil membaca data rb

total	count
0	1
1	1
2	0
3	7
4	9
5	15
6	15
7	7
8	4
9	1
10	0

```
plot(rb)
```



Gambar 10.2 hasil plot rb

10.1 *Equating Equivalent Groups Design*

Identitas dan fungsi rata-rata adalah varian terbatas dari fungsi linier. Fungsi linier, pada gilirannya, adalah kasus khusus dari fungsi linier umum (Albano, 2015). Semua fungsi ini berbeda berdasarkan penyesuaian yang dibuat untuk menempatkan satu bentuk ke skala yang sama dari bentuk lain (Desjardins & Bulut, 2017). Pada desain grup ekuivalen, dua tes yang akan disetarakan diberikan kepada dua grup

yang setara, yang dipilih secara random dari populasi sama, di mana kedua grup mempunyai tingkat kemampuan yang sama.

Menurut Retnawati (2014), kelebihan dari equivalent group design adalah lebih simple dalam penggunaan dan dapat mengeliminasi pengaruh latihan dan kepenatan. Kelemahan desain ini adalah adanya bias yang dihasilkan dari proses penyetaraan, karena grup-grup tersebut distribusi kemampuannya belum tentu sama. Untuk mengurangi bias yang mungkin muncul terkait dengan sampel, maka secara umum desain ini memerlukan ukuran sampel yang besar. Berikut ini 6 contoh *equating* klasik menggunakan *random groups design* atau yang biasa disebut *equivalent groups design.*

Berikut ini sintaks dalam R Studio untuk melakukan equating dengan berbagai tipe yang tersedia dalam paket. Pembaca harus menge-run req1 sampai req6 jika ingin menggunakan semua tipe equating yang tersedia. Akan tetapi jika yang dibutuhkan hanya tipe equating tertentu saja, maka tidak perlu untuk menge-*run* semuanya.

```
#contoh equating klasik menggunakan 6 tipe dari "random groups
design/equivalent groups design", tanpa anchor/butir bersama
req1 <- equate(ra, rb, type = "identity")
req2 <- equate(ra, rb, type = "mean")
req3 <- equate(ra, rb, type = "linear")
req4 <- equate(ra, rb, type = "general linear")
req5 <- equate(ra, rb, type = "equipercentile")
req6 <- equate(ra, rb, type = "circle-arc")
```

10.1.1 *Equating* tipe "*Identity*" dan metode "*None*"

Fungsi identitas tidak akan membuat penyesuaian dan mengasumsikan bahwa X dan Y memiliki properti skala yang sama (yaitu, properti distribusi). X sudah pada skala yang sama dengan Y karena memiliki sifat yang identik dengan Y(Desjardins & Bulut, 2017). Fungsi linier sesuai ketika kesulitan bentuk tes berubah secara linier di seluruh skor skala, dengan konstanta b dan laju perubahan a, untuk lebih detailnya pembaca dapat mengakses Albano (2016). Berikut ini sintaks untuk melakukan *equating* nya.

```
#hasil equating tipe "identity" dan metode "none"

req1 <- equate(ra, rb, type = "identity")
req1
```

```
str(req1)
detail_req1 <- req1$concordance
detail_req1
plot(detail_req1$scale, detail_req1$yx, type = 'l')
```

Berikut ini output str(req1) yang berisikan hasil *equating* tipe *Identity*.

```
$ name        : chr "Identity Equating: ra to rb"
$ type        : chr "identity"
$ method      : chr "none"
$ design      : chr "equivalent groups"
$ x           : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y           : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  3 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ se   : num [1:11] 0 0 0 0 0 0 0 0 0 0 ...
$ internal    : logi TRUE
$ lts         : logi FALSE
$ coefficients: Named num [1:6] 0 1 5 5 10 10
 ..- attr(*, "names")= chr [1:6] "intercept" "slope" "cx" "cy" ...
$ points      :'data.frame':   2 obs. of  3 variables:
 ..$ low : num [1:2] 0 0
 ..$ mid : num [1:2] 5 5
 ..$ high: num [1:2] 10 10
$ weights     : Named num [1:4] 0 0 0 0
 ..- attr(*, "names")= chr [1:4] "wax" "way" "wbx" "wby"
- attr(*, "class")= chr "equate"
```

Gambar 10.3 hasil str(req1) yang merupakan *equating* tipe *Identity*

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan dari Pembaca. Berikut ini hasil req1, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
Identity Equating: ra to rb

Design: equivalent groups

Summary Statistics:
   mean   sd  skew kurt min max  n
x  5.16 1.66  0.16 2.09   2   9 58
y  5.22 1.70 -0.42 3.54   0   9 60
yx 5.16 1.66  0.16 2.09   2   9 58

Coefficients:
intercept     slope        cx        cy        sx        sy
        0         1         5         5        10        10
```

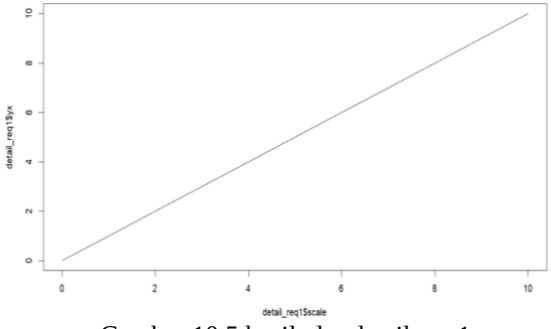
Gambar 10.4 hasil req1

Berikut ini hasil detail_req1, yang memuat scale, yx, dan se. Output menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan paket a dan b adalah sama, artinya paket a dan b setara. Kolom scale menunjukkan skor peserta tes pada paket a, sedangkan kolom yx menunjukkan skor peserta pada paket tes b.

Tabel 10.5 hasil detail_req1

scale	yx	se
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0
7	7	0
8	8	0
9	9	0
10	10	0

Berikut ini hasil plot detail_req1 yang menunjukkan hubungan antara data paket a dan paket b yang linear karena skor peserta ketika mengerjakan paket a sama dengan skor peserta ketika mengerjakan paket b, artinya paket a setara dengan paket b.



Gambar 10.5 hasil plot detail_req1

10.1.2 Equating tipe "Mean" dan metode "None"

Fungsi rata-rata menyesuaikan skor dengan perbedaan rata-rata antara X dan Y. Jika X lebih sulit dari Y, maka selisih ini ditambahkan ke skor pada X; jika X lebih mudah dari Y, maka perbedaan ini dikurangi dari skor pada X (Desjardins & Bulut, 2017). Berikut ini sintaks untuk melakukan *equating* nya.

```
#hasil equating tipe "mean" dan metode "none"
req2 <- equate(ra, rb, type = "mean")
```

Berikut ini str(req2) yang merupakan hasil *equating* tipe *Mean.*

```
str(req2)
```

```
$ name        : chr "Mean Equating: ra to rb"
$ type        : chr "mean"
$ method      : chr "none"
$ design      : chr "equivalent groups"
$ x           : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y           : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  2 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] 0.0615 1.0615 2.0615 3.0615 4.0615 ...
$ internal    : logi TRUE
$ lts         : logi FALSE
$ coefficients: Named num [1:6] 0.0615 1 5 5 10 ...
 ..- attr(*, "names")= chr [1:6] "intercept" "slope" "cx" "cy" ...
$ points      :'data.frame':   2 obs. of  3 variables:
 ..$ low : num [1:2] 0 0
 ..$ mid : num [1:2] 5 5
 ..$ high: num [1:2] 10 10
$ weights     : Named num [1:4] 0 0 1 1
 ..- attr(*, "names")= chr [1:4] "wax" "way" "wbx" "wby"
- attr(*, "class")= chr "equate"
```

Gambar 10.6 hasil str(req2) yang merupakan *equating* tipe *Mean*

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan dari Pembaca. Berikut ini hasil req2, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
req2
```

```
Mean Equating: ra to rb

Design: equivalent groups

Summary Statistics:
   mean   sd  skew kurt  min  max  n
x  5.16 1.66  0.16 2.09 2.00 9.00 58
y  5.22 1.70 -0.42 3.54 0.00 9.00 60
yx 5.22 1.66  0.16 2.09 2.06 9.06 58

Coefficients:
intercept     slope        cx        cy        sx        sy
   0.0615    1.0000    5.0000    5.0000   10.0000   10.0000
```

Gambar 10.7 hasil req2

```
detail_req2 <- req2$concordance
detail_req2
```

Berikut ini hasil detail_req2, yang memuat scale dan yx. Output menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan

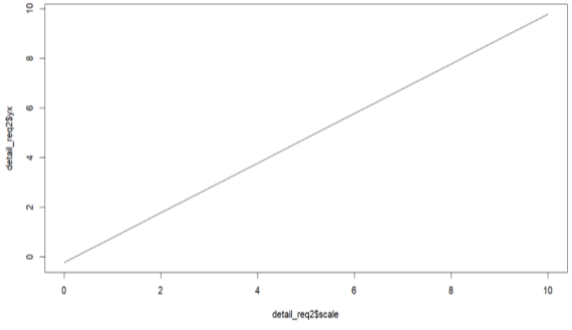
paket a dan b adalah sedikit sekali perbedaannya (tidak signifikan), artinya paket a dan b cukup setara. Peserta tes yang mendapat skor 0 pada paket a setara dengan skor 0.06149425 pada paket b. Selanjutnya peserta tes yang mendapat skor 1 pada paket a setara dengan skor 1.06149425 pada pake b. Kesimpulannya adalah paket b cukup setara dengan paket a. Rumus skornya menjadi sebagai berikut:
skor paket b = skor paket a + 0,06149425 atau Y = X + 0,06149425

Tabel 10.6 hasil detail_req2

scale	yx
0	0.06149425
1	1.06149425
2	2.06149425
3	3.06149425
4	4.06149425
5	5.06149425
6	6.06149425
7	7.06149425
8	8.06149425
9	9.06149425
10	10.06149425

Berikut ini hasil plot detail_req2 yang menunjukkan hubungan antara data pekt a dan paket b yang cukup setara.

```
plot(detail_req2$scale, detail_req2$yx, type = 'l')
```



Gambar 10.8 hasil plot detail_req2

10.1.3 Equating tipe "Linear" dan metode "None"

Fungsi linier menyesuaikan skor berdasarkan rata-rata dan standar penyimpangan bentuk tes X dan Y. Skor disamakan sedemikian rupa sehingga jika skor suatu tes standar deviasi di atas rata-rata pada bentuk X, skor yang disamakan pada Y akan menjadi satu standar deviasi di

atas rata-rata (Desjardins & Bulut, 2017). Berikut ini sintaks untuk melakukan *equating* nya.

```
#hasil equating tipe "linear" dan metode "none"
req3 <- equate(ra, rb, type = "linear")
str(req3)
```

Berikut ini *output* str(req3) yang merupakan hasil *equating* tipe *Linear.*

```
$ name        : chr "Linear Equating: ra to rb"
$ type        : chr "linear"
$ method      : chr "none"
$ design      : chr "equivalent groups"
$ x           : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y           : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  3 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] -0.0487 0.9727 1.994 3.0154 4.0368 ...
 ..$ se   : num [1:11] 0.558 0.402 0.279 0.187 0.127 ...
$ internal    : logi TRUE
$ lts         : logi FALSE
$ coefficients: Named num [1:6] -0.0487 1.0214 5 5 10 ...
 ..- attr(*, "names")= chr [1:6] "intercept" "slope" "cx" "cy" ...
$ points      :'data.frame':   2 obs. of  3 variables:
 ..$ low : num [1:2] 0 0
 ..$ mid : num [1:2] 5 5
 ..$ high: num [1:2] 10 10
$ weights     : Named num [1:4] 1 1 1 1
 ..- attr(*, "names")= chr [1:4] "wax" "way" "wbx" "wby"
- attr(*, "class")= chr "equate"
```

Gambar 10.9 hasil str(req3) yang merupakan *equating* tipe *Linear*

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan dari Pembaca. Berikut ini hasil req3, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
req3
```

```
Linear Equating: ra to rb

Design: equivalent groups

Summary Statistics:
   mean   sd  skew kurt  min  max  n
x  5.16 1.66  0.16 2.09 2.00 9.00 58
y  5.22 1.70 -0.42 3.54 0.00 9.00 60
yx 5.22 1.70  0.16 2.09 1.99 9.14 58

Coefficients:
intercept     slope        cx        cy        sx        sy
  -0.0487    1.0214    5.0000    5.0000   10.0000   10.0000
```

Gambar 10.10 hasil req3

Berikut ini hasil detail_req3, yang memuat scale, yx, dan se. Output menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan paket a dan b adalah sedikit berbeda (tidak signifikan perbedaannya). Peserta tes yang mendapat skor 0 pada paket a setara dengan skor -0.04872003 pada paket b. Selanjutnya peserta tes yang mendapat skor 1 pada paket a setara dengan skor 0.97265933 pada pake b. Selanjutnya peserta tes yang mendapat skor 2 pada paket a setara dengan skor 1.99403869 pada pake b. Hal ini (perbandingan skor 0 sampai 2 pada paket a) menunjukkan bahwa paket b sedikit lebih sulit dibandingkan paket a, akan tetapi hal ini hanya berlaku sampai di skor 2 pada paket a, setelah itu untuk skor 3 pada paket a sampai skor 10, hasilnya menunjukkan kebalikannya yaitu paket b sedikit lebih mudah dibandingkan paket a, karena skor di paket b sedikit lebih besar dibandingkan skor yang di paket a.

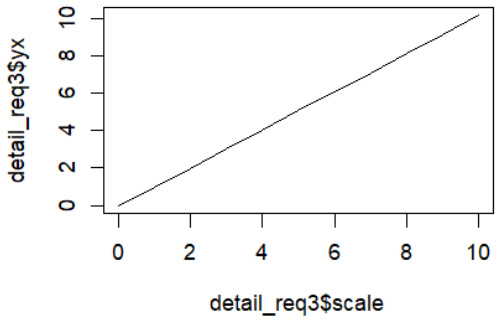
```
detail_req3 <- req3$concordance
detail_req3
```

Tabel 10.7 hasil detail_req3

scale	yx	se
0	-0.04872003	0.5578767
1	0.97265933	0.4024651
2	1.99403869	0.2789016
3	3.01541805	0.1871862
4	4.03679741	0.1273190
5	5.05817677	0.0992999
6	6.07955613	0.1031289
7	7.10093549	0.1388061
8	8.12231484	0.2063314
9	9.14369420	0.3057048
10	10.16507356	0.4369264

Berikut ini hasil plot details req3 yang menunjukkan hubungan antara data paket a dan paket b yang hasilnya tidak selinear tipe-tipe equating yang telah dicontohkan sebelumnya.

```
plot(detail_req3$scale, detail_req3$yx, type = 'l')
```



Gambar 10.11 hasil plot detail_req3

10.1.4 Equating tipe "General Linear" dan metode "None"

Berikut ini sintaks untuk melakukan equating tipe general linear dan metode none yang langkah-langkahnya serupa dengan contoh-contoh sebelumnya, hanya saja berbeda di kodenya yaitu kali ini dimisalkan menggunakan req4.

```
#hasil equating tipe "general linear" dan metode "none"
req4 <- equate(ra, rb, type = "general linear")
str(req4)
```

Berikut ini hasil str(req4) yang merupakan *equating* tipe *General Linear.*

```
$ name        : chr "General Linear Equating: ra to rb"
$ type        : chr "general linear"
$ method      : chr "none"
$ design      : chr "equivalent groups"
$ x           : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y           : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  2 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
$ internal    : logi TRUE
$ lts         : logi FALSE
$ coefficients: Named num [1:6] 0 1 5 5 10 10
 ..- attr(*, "names")= chr [1:6] "intercept" "slope" "cx" "cy" ...
$ points      :'data.frame':   2 obs. of  3 variables:
 ..$ low : num [1:2] 0 0
 ..$ mid : num [1:2] 5 5
 ..$ high: num [1:2] 10 10
$ weights     : Named num [1:4] 0 0 0 0
 ..- attr(*, "names")= chr [1:4] "wax" "way" "wbx" "wby"
```

Gambar 10.12 hasil str(req4) yang merupakan *equating* tipe *General Linear*

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan dari Pembaca. Berikut ini hasil req4, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
req4
```

```
General Linear Equating: ra to rb

Design: equivalent groups

Summary Statistics:
   mean   sd  skew kurt min max  n
x  5.16 1.66  0.16 2.09   2   9 58
y  5.22 1.70 -0.42 3.54   0   9 60
yx 5.16 1.66  0.16 2.09   2   9 58

Coefficients:
intercept     slope        cx        cy        sx        sy
        0         1         5         5        10        10
```

Gambar 10.13 hasil req4

Berikut ini hasil detail_req4, yang memuat scale dan yx. *Output* menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan paket a dan b adalah sama, artinya paket a dan b setara.

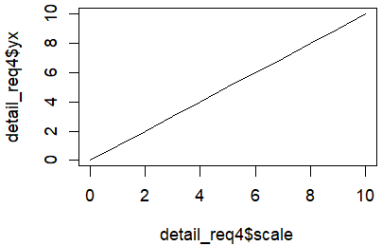
```
detail_req4 <- req4$concordance
detail_req4
```

Tabel 10.8 hasil detail_req4

scale	yx
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Berikut ini hasil plot detail_req4 yang menunjukkan hubungan antara data paket a dan paket b yang linear, karena kemampuan peserta ketika mengerjakan paket a maupun paket b hasilnya sama.

```
plot(detail_req4$scale, detail_req4$yx, type = 'l')
```



Gambar 10.14 hasil plot detail_req4

10.1.5 Equating tipe "Equipercentile" dan metode "None"

Berikut ini sintaks untuk melakukan equating tipe equipercentile dan metode none yang langkah-langkahnya serupa dengan contoh-contoh sebelumnya, hanya saja berbeda di kodenya yaitu kali ini dimisalkan menggunakan req5.

```
#hasil equating tipe "equipercentile" dan metode "none"
req5 <- equate(ra, rb, type = "equipercentile")
str(req5)
```

Berikut ini hasil str(req5) yang merupakan equating tipe *Equipercentile*.

```
$ name        : chr "Equipercentile Equating: ra to rb"
$ type        : chr "equipercentile"
$ method      : chr "none"
$ design      : chr "equivalent groups"
$ x           : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y           : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  3 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] -0.5 -0.5 0.0172 3.1749 4.5069 ...
 ..$ se   : num [1:11] 0 0 0.891 0.465 0.337 ...
$ points      :'data.frame':   2 obs. of  2 variables:
 ..$ low : num [1:2] 0 0
 ..$ high: num [1:2] 10 10
$ smoothmethod: chr "none"
- attr(*, "class")= chr "equate"
```

Gambar 10.15 hasil str(req5) yang merupakan *equating* tipe *Equipercentile*

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan

dari pembaca. Berikut ini hasil req5, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
req5
```

```
Equipercentile Equating: ra to rb

Design: equivalent groups

Smoothing Method: none

Summary Statistics:
   mean   sd  skew kurt  min  max  n
x  5.16 1.66  0.16 2.09 2.00 9.00 58
y  5.22 1.70 -0.42 3.54 0.00 9.00 60
yx 5.25 1.64 -0.30 3.46 0.02 8.98 58
```

Gambar 10.16 hasil req5

Berikut ini hasil detail_req5, yang memuat scale, yx, dan se. Output menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan paket a dan b adalah cukup berbeda, artinya paket a dan b tidak setara. Peserta tes yang mendapat skor 0 pada paket a setara dengan skor -0,5 pada paket b. Akan tetapi, selanjutnya peserta tes yang mendapat skor 1 pada paket a juga setara dengan skor -0,5 pada pake b. Sementara itu peserta yang mendapat skor 10 pada paket a setara dengan skor 10,5 pada paket b. Kesimpulannya adalah tidak bisa ditentukan paket manakah yang lebih sulit, karena konversi skor berbeda dari skor yang satu dengan skor yang lainnya antara kedua paket. Hasil ini menunjukkan bahwa jika dua paket instrument terbukti setara ketika menggunakan tipe equating lainnya belum tentu hasilnya serupa dengan equating tipe equipercentil ini. Pembaca lah yang dapat menentukan tipe equating mana yang akan digunakan sesuai kebutuhan.

```
detail_req5 <- req5$concordance
detail_req5
```

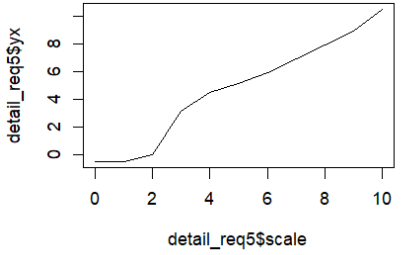
Tabel 10.9 hasil detail_req5

scale	yx	se
0	-0.50000000	0.0000000
1	-0.50000000	0.0000000
2	0.01724138	0.8908110
3	3.17487685	0.4653525
4	4.50689655	0.3374122
5	5.16206897	0.3465134
6	5.9551724	0.3233333

7	6.95812808	0.5254729
8	7.97413793	0.5584580
9	8.98275862	0.8908110
10	10.50000000	0.0000000

Berikut ini hasil plot details q5 yang menunjukkan hubungan antara data paket a dan paket b yang tidak linear, sangat berbeda dengan hasil yang ditunjukkan oleh tipe-tipe equating yang telah dibahas sebelumnya, padahal data yang digunakan sama.

```
plot(detail_req5$scale, detail_req5$yx, type = 'l')
```



Gambar 10.17 hasil plot detail_req5

10.1.6 Equating tipe "Circle-arc" dan metode "None"

Berikut ini sintaks untuk melakukan equating tipe circle-arc dan metode none yang langkah-langkahnya serupa dengan contoh-contoh sebelumnya, hanya saja berbeda di kodenya yaitu kali ini dimisalkan menggunakan req6.

```
#hasil equating tipe "circle-arc" dan metode "none"
req6 <- equate(ra, rb, type = "circle-arc")
str(req6)
```

Berikut ini hasil str(req6) yang merupakan equating tipe *Circle-arc*.

```
$ name       : chr "Circle-Arc Equating: ra to rb"
$ type       : chr "circle-arc"
$ method     : chr "none"
$ design     : chr "equivalent groups"
$ x          : 'freqtab' int [1:11(1d)] 0 0 1 11 11 8 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ y          : 'freqtab' int [1:11(1d)] 1 1 0 7 9 15 15 7 4 1 ...
 ..- attr(*, "dimnames")=List of 1
 .. ..$ total: chr [1:11] "0" "1" "2" "3" ...
 ..- attr(*, "design")= chr "eg"
$ concordance :'data.frame':   11 obs. of  2 variables:
 ..$ scale: num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
 ..$ yx   : num [1:11] 0 1.02 2.04 3.05 4.06 ...
$ simple     : logi TRUE
$ coefficients: Named num [1:5] 0 1 5 -203 203
 ..- attr(*, "names")= chr [1:5] "intercept" "slope" "xcenter" "ycenter" ...
$ points     :'data.frame':   3 obs. of  3 variables:
 ..$ low : num [1:3] 0 0 0
 ..$ mid : num [1:3] 5.1552 5.2167 0.0615
 ..$ high: num [1:3] 10 10 0
- attr(*, "class")= chr "equate"
```

Gambar 10.18 hasil str(req6) yang merupakan equating tipe Circle-arc

Pada gambar output di atas, terdapat banyak keterangan yang dapat dilihat dari hasil equating yang dapat digunakan seusai kebutuhan dari Pembaca. Berikut ini hasil req6, yang menunjukkan mean, sd, skew, kurt, min, max, dan n.

```
req6
```

```
Circle-Arc Equating: ra to rb

Design: equivalent groups

Summary Statistics:
   mean   sd  skew kurt  min  max  n
x  5.16 1.66  0.16 2.09 2.00 9.00 58
y  5.22 1.70 -0.42 3.54 0.00 9.00 60
yx 5.21 1.66  0.15 2.08 2.04 9.02 58

Coefficients:
intercept     slope   xcenter   ycenter         r
   0.0000    1.0000    5.0000 -203.0445  203.1061
```

Gambar 10.19 hasil req6

Berikut ini hasil detail_req6, yang memuat scale dan yx. Output menunjukkan bahwa nilai/kemampuan peserta ketika dites dengan paket a dan b adalah hanya sedikit berbeda (tidak signifikan perbedaannya), artinya paket a dan b cukup setara. Peserta tes yang mendapat skor 0 pada paket a setara dengan skor 0 pada paket b. Selanjutnya peserta tes yang mendapat skor 1 pada paket a setara dengan skor 1.022161 pada pake b. Akan tetapi ketika skor 10 pada paket a kembali setara dengan paket b yaitu skornya 10. Kesimpulannya adalah paket b cukup setara dengan paket a.

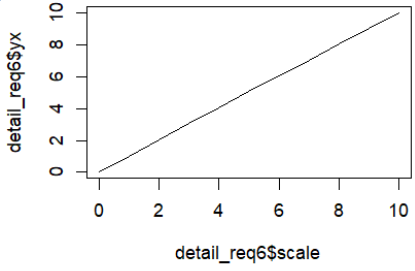
```
detail_req6 <- req6$concordance
detail_req6
```

Tabel 10.10 hasil detail_req6

scale	yx
0	0.000000
1	1.022161
2	2.039396
3	3.051706
4	4.059092
5	5.061554
6	6.059092
7	7.051706
8	8.039396
9	9.022161
10	10.000000

Berikut ini hasil plot details req 6 yang menunjukkan hubungan antara data paket a dan paket b yang cukup linear, karena skor peserta pada paket a cukup setara dengan paket b.

```
plot(detail_req6$scale, detail_req6$yx, type = 'l')
```



Gambar 10.20 hasil plot detail_req6

10.2 Perbandingan Skor Equating Klasik Equivalent Groups Design

Hasil equating berbagai tipe perlu dibandingkan untuk mngetahui mana yang sesuai dengan kebutuhan pengguna. Enam tipe equating yang termasuk equivalen group design telah dijelaskan langkah-langkah penggunaannya pada pembahasan sebelumnya. Berikut ini sintaks untuk membandingkan hasil equating klasik untuk *equivalent groups design*.

```
round(cbind(xscale = 0:10,
            "identity" = req1$conc$yx,
             "mean" = req2$conc$yx,
             "linear" = req3$conc$yx,
             "general linear" = req4$conc$yx,
             "equipercentile" = req5$conc$yx,
```

```
"circle-arc" = req6$conc$yx),2)
```

Berikut ini perbandingan hasil *equiting*nya yang menunjukkan bahwa hanya tipe equipercentil yang hasilnya cukup mencolok menggambarkan ketidaksetaraan antara skor peserta pada paket a dan paket b. Selain tipe equiting equipercentile menunjukkan bahwa skor peserta pada paket a setara dengan paket b, karena tidak ada perbedaan sama sekali atau ada perbedaan namun tidak signifikan. Selanjutnya, tergantung pada Pembaca akan menggunakan tipe equiting yang mana sesuai dengan kebutuhannya.

xscale	identity	mean linear	general linear	equipercentile	circle-arc
0	0	0.06	-0.05	0	-0.50	0.00
1	1	1.06	0.97	1	-0.50	1.02
2	2	2.06	1.99	2	0.02	2.04
3	3	3.06	3.02	3	3.17	3.05
4	4	4.06	4.04	4	4.51	4.06
5	5	5.06	5.06	5	5.16	5.06
6	6	6.06	6.08	6	5.96	6.06
7	7	7.06	7.10	7	6.96	7.05
8	8	8.06	8.12	8	7.97	8.04
9	9	9.06	9.14	9	8.98	9.02
10	10	10.06	10.17	10	10.50	10.00

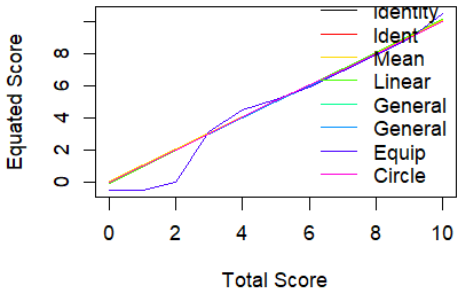
Gambar 10.21 Perbandingan hasil *equating equivalent groups design*

10.3 Perbandingan Plot 6 Teknik Equating Klasik Equivalent Groups Design

Perbedaan skor peserta tes antara beberapa tipe equiting yang termasuk equivalen groups design telah dipaparkan pada pembahasan sebelumnya. Berikut ini sintaks untuk membandingkan plot hasil equating klasik untuk *equivalent groups design*.

```
plot(req1, req2, req3, req4, req4, req5, req6)
```

Berikut ini perbandingan hasil plot *equiting*nya yang menunjukkan bahwa tipe equipercentil (warna biru tua) menghasilkan grafik yang paling berbeda dengan grafik tipe equiting lainnya. Equiting tipe equipercentil menghasilkan grafik yang tidak linear, berbeda dengan grafik tipe equiting lainnya yang cenderung linear. Hal ini menunjukkan bahwa berdasarkan hasil equiting tipe equipercentil, paket a tidak setara dengan paket b.



Gambar 10.22 perbandingan plot hasil *equating equivalent groups design*

10.4 Equating Nonequivalent Groups Design

Ketika populasi target tunggal tidak dapat diasumsikan, desain kelompok yang tidak setara digunakan dan metode persamaan harus ditentukan. Umumnya, metode ini bekerja dengan menghubungkan skor total pada paket A dan paket B melalui skor pada set butir umum yang muncul pada kedua bentuk (yaitu, skor jangkar umum) dan penciptaan populasi sintetis berbobot. Metode-metode ini berbeda dalam cara mereka menggambarkan dan memperkirakan bentuk hubungan ini. Metode Tucker, bobot nominal, dan skor sejati Levine mengandalkan berbagai bentuk regresi untuk melakukan hal ini, sedangkan estimasi frekuensi dan Braun/Holland tidak. Metode penyamaan berantai adalah satu-satunya metode yang tidak secara eksplisit melibatkan pembuatan populasi sintetik (Desjardins & Bulut, 2017).

Menurut Retnawati (2014), pada desain non-equivalent groups design, dua tes yang akan disetarakan diberikan kepada dua grup yang tidak ekuivalen. Butir yang merupakan bagian dari butir-tes paket A dan juga paket B disebut butir bersama (*anchor item*). Keunggulan menggunakan *anchor item* dalam *equiting* adalah setiap grup hanya memperoleh satu paket tes, skor pada butir bersama ikut diperhitungkan dalam perhitungan skor total tes, dan waktu pelaksanaan tes lebih singkat. Berikut ini 3 contoh equating klasik menggunakan *nonequivalent groups design*.

Berikut ini sintaks untuk membaca data paket a dan b yang dibutuhkan khusus untuk *nonequivalent groups design*, yang sedikit

berbeda dengan sebelumnya, karena adanya butir *anchor*. Berbeda dengan "equivalent group design" yang hanya membutuhkan data sampai pada skor total setiap peserta/sampel, pada "nonequivalent groups design" ini data yang dibutuhkan lebih banyak. Tambahan data yang dibutuhkan yaitu keterangan butir mana saja yang merupakan anchor dan jumlah butir anchor yang benar.

```
#membaca data paket a dan b untuk kepentingan "nonequivalent
groups design"

a <- read.xlsx('PAKET TES.xlsx', 1, header = T)
a <- data.frame(a)
a
b <- read.xlsx('PAKET TES.xlsx', 2, header = T)
b <- data.frame(b)
b

#memanggil bagian dari data paket a dan b untuk kepentingan
"nonequivalent groups design"

na <- freqtab(a[,12:13], scales = list(0:10, 0:2))
nb <- freqtab(b[,12:13], scales = list(0:10, 0:2))
na
nb
```

Misalkan pada file xlsx di kolom 12 merupakan skor total setiap peserta, kemudian misalkan terdapat 2 butir anchor yaitu butir nomor 9 dan 10, selanjutnya kolom 13 berisikan jumlah butir anchor yang benar untuk setiap peserta. Artinya, jika peserta 1 misalnya hanya benar butir anchor nomor 9 dan salah di butir anchor nomor 10 maka skornya di kolom 13 adalah 1. Contoh lainnya misalnya peserta nomor 2 menjawab benar butir anchor nomor 9 dan 10 maka skornya di kolom 13 adalah 2. Conoth lainnya lagi misalnya peserta 3 salah menjawab butir anchor 9 dan 10 maka skornya di kolom 13 adalah 0. Berikut ini hasil na (frekuensi skor dan jumlah *anchor* yang benar untuk paket a).

total	anchor	count
0	0	0
1	0	0
2	0	1
3	0	8
4	0	3
5	0	2
6	0	2
7	0	1
8	0	0
9	0	0
10	0	0
0	1	0
1	1	0
2	1	0
3	1	2
4	1	7
5	1	6
6	1	9
7	1	5
8	1	1
9	1	1
10	1	0
0	2	0
1	2	0
2	2	0
3	2	1
4	2	1
5	2	0
6	2	4
7	2	1
8	2	3
9	2	0
10	2	0

Gambar 10.23 hasil na

Hasil analisis di atas menunjukkan bahwa untuk paket a terdapat 3 kategori yaitu dengan total jawaban anchor 0 (11 baris dari skor 0 sampai 10), total jawaban anchor 1(11 baris dari skor 0 sampai 10), dan total jawaban anchor 2 (11 baris dari skor 0 sampai 10), sehingga total 33 baris data ke bawah. Demikian pula untuk data paket b, berikut ini hasil na (frekuensi skor dan jumlah *anchor* yang benar untuk paket b)

total	anchor	count
0	0	1
1	0	1
2	0	0
3	0	4
4	0	3
5	0	3
6	0	1
7	0	0
8	0	0
9	0	0
10	0	0
0	1	0
1	1	0
2	1	0
3	1	2
4	1	5
5	1	12
6	1	8
7	1	3
8	1	2
9	1	0
10	1	0
0	2	0
1	2	0
2	2	0
3	2	1
4	2	1
5	2	0
6	2	6
7	2	4
8	2	2
9	2	1
10	2	0

Gambar 10.24 hasil nb

Setelah berhasil membaca data excel dan memanggil bagian data yang dibutuhkan dari paket a dan b seperti pada pembahsan sebelumnya, selanjutnya dilakukan proses equatingnya. Berikut ini sintaks untuk contoh *equating* klasik menggunakan 3 tipe dari "*nonequivalent groups design*". Pembaca harus menge-run neq1 sampai neq3 jika ingin menggunakan semua tipe equating yang tersedia. Akan tetapi jika yang dibutuhkan hanya tipe equating tertentu saja, maka tidak perlu untuk menge-run semuanya.

```
#contoh equating klasik menggunakan 3 tipe dari "nonequivalent
groups design", dengan anchor/butir bersama
neq1 <- equate(na, nb, type = "linear", method = "tuck",
               ws = 1)
neq2 <- equate(na, nb, type = "equip", method = "freq",
               ws = 1)
neq3 <- equate(na, nb, type = "linear", method = "braun",
               ws = 1)
```

10.4.1 Equating tipe "Linear" dan metode "Tucker"

Berikut ini sintaks untuk untuk *equating* kombinasi antara tipe *Linear* dengan metode *Tucker.*

```
#hasil equating tipe "linear" dan metode "tucker"

neq1 <- equate(na, nb, type = "linear", method = "tuck",
               ws = 1)
neq1
detail_neq1 <- neq1$concordance
detail_neq1
plot(detail_neq1$scale, detail_neq1$yx, type = 'l')
```

Proses equating menggunakan R Studio untuk equating non equivalent groups design ini sebenarnya sama saja langkah-langkahnya dengan pembahasan-pembahasan sebelumnya, hanya kodenya yang berbeda yaitu neq1. Untuk kepentingan penyederhanaan tampilan maka akan ditampilkan salah satu output terpentingya saja. Berikut ini hasil *equating* kombinasi antara tipe *Linear* dengan metode *Tucker* yang menunjukkan bahwa paket a cukup setara dengan paket b (tidak signifikan perbedaan skornya), dilihat dari skor pada scale, yx, dan se.

scale	yx	se.n	se.g
0	-0.2401142	0.7127703	0.7100815
1	0.7760715	0.5988906	0.5993560
2	1.7922571	0.4902663	0.4936407
3	2.8084428	0.3912990	0.3969591
4	3.8246284	0.3113382	0.3176691
5	4.8408141	0.2679754	0.2714636
6	5.8569997	0.2788421	0.2755233
7	6.8731854	0.3387591	0.3279869
8	7.8893710	0.4275808	0.4107061
9	8.9055567	0.5309949	0.5091417
10	9.9217424	0.6419881	0.6158026

Gambar 10.25 detail_neq1

10.4.2 Equating tipe "Equipercentil" dan metode "Frequency Estimation"

Berikut ini sintaks untuk untuk *equating* kombinasi antara tipe *Equipercentile* dengan metode *Frequency Estimation*.

```
#hasil equating tipe "equipercentil" dan metode "frequency
estimation"
```

```
neq2 <- equate(na, nb, type = "equip", method = "freq", ws =
1)
neq2
detail_neq2 <- neq2$concordance
detail_neq2
plot(detail_neq1$scale, detail_neq2$yx, type = 'l')
```

Proses equating menggunakan R Studio untuk equating non equivalent groups design ini sebenarnya sama saja langkah-langkahnya dengan pembahasan-pembahasan sebelumnya, hanya kodenya yang berbeda yaitu neq3. Untuk kepentingan penyederhanaan tampilan maka akan ditampilkan salah satu output terpentingya saja. Berikut ini hasil *equating* kombinasi antara tipe *Equipercentil* dengan metode *Frequency Estimation* yang menunjukkan bahwa paket a tidak setara dengan paket b (cukup signifikan perbedaan skornya), dilihat dari skor pada scale, yx, dan se.

scale	yx
0	-0.5000000
1	-0.5000000
2	-0.1176471
3	2.9958069
4	4.2473032
5	4.9576891
6	5.7349656
7	6.6813084
8	7.7866242
9	8.7500000
10	10.5000000

Gambar 10.26 detail_neq2

10.4.3 Equating tipe "Linear" dan metode "Braun/Holland"

Berikut ini sintaks untuk untuk equating kombinasi antara tipe *Linear* dengan metode *Braun/Holland*.

```
#hasil equating tipe "linear" dan metode "Braun/Holland"
neq3 <- equate(na, nb, type = "linear", method = "braun",
               ws = 1)
neq3
detail_neq3 <- neq3$concordance
detail_neq3
plot(detail_neq3$scale, detail_neq3$yx, type = 'l')
```

Proses equating menggunakan R Studio untuk equating non equivalent groups design ini sebenarnya sama saja langkah-langkahnya dengan pembahasan-pembahasan sebelumnya, hanya kodenya yang berbeda yaitu neq2. Untuk kepentingan penyederhanaan tampilan maka akan ditampilkan salah satu output terpentingya saja. Berikut hasil equating kombinasi antara tipe *Linear* dengan metode *Braun/Holland* yang menunjukkan bahwa paket a setara dengan paket b (tidak signifikan perbedaan skornya), dilihat dari skor pada scale dan yx.

scale	yx
0	-0.3548574
1	0.6832424
2	1.7213421
3	2.7594419
4	3.7975416
5	4.8356414
6	5.8737411
7	6.9118408
8	7.9499406
9	8.9880403
10	10.0261401

Gambar 10.27 detail_neq3

10.5 Perbandingan Skor Hasil Equating Klasik dengan Desain Nonequivalent Groups

Hasil equating berbagai tipe perlu dibandingkan untuk mngetahui mana yang sesuai dengan kebutuhan pengguna. Tiga kombinasi tipe equating yang termasuk nonequivalen group design telah dijelaskan langkah-langkah penggunaannya pada pembahasan sebelumnya. Berikut ini sintaks Perbandingan Skor Hasil *Equating* Klasik *Nonequivalent groups design*.

```
# Perbandingan Skor Hasil Equating Klasik Nonequivalent groups
design
round(cbind(xscale = 0:10, tucker = neq1$conc$yx,
       fe = neq2$conc$yx, braun = neq3$conc$yx), 2)
```

Berikut ini Perbandingan Skor Hasil *Equating* Klasik *Nonequivalent groups design*. Perbandingan hasil *equiting*nya yang menunjukkan bahwa hanya hasil *equating* kombinasi antara tipe *Equipercentil* dengan metode *Frequency Estimation* yang hasilnya cukup mencolok menggambarkan ketidaksetaraan antara skor peserta pada paket a dan paket b. Selain tipe equiting *Equipercentil* dengan

metode *Frequency Estimation* menunjukkan bahwa skor peserta pada paket a setara dengan paket b, karena tidak ada perbedaan sama sekali atau ada perbedaan namun tidak signifikan. Seperti telah disarankan sebelumnya, tergantung pada Pembaca akan menggunakan tipe equiting yang mana sesuai dengan kebutuhannya.

xscale	tucker	fe	braun
0	-0.24	-0.50	-0.35
1	0.78	-0.50	0.68
2	1.79	-0.12	1.72
3	2.81	3.00	2.76
4	3.82	4.25	3.80
5	4.84	4.96	4.84
6	5.86	5.73	5.87
7	6.87	6.68	6.91
8	7.89	7.79	7.95
9	8.91	8.75	8.99
10	9.92	10.50	10.03

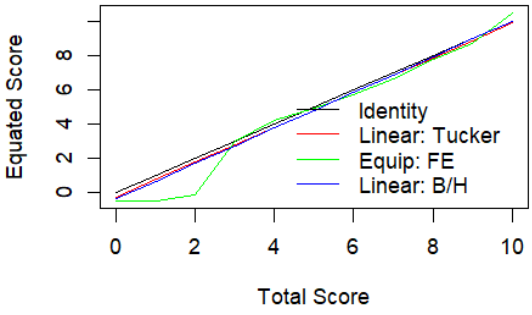
Gambar 10.28 Perbandingan Skor Hasil *Equating* Klasik *Nonequivalent groups design*

10.6 Perbandingan Plot 3 Teknik Equating Klasik dengan Desain Nonequivalent Groups

Perbedaan skor peserta tes antara beberapa tipe equiting yang termasuk equivalen groups design telah dipaparkan pada pembahasan sebelumnya. Berikut ini sintaks Perbandingan plot Skor Hasil *Equating* Klasik *Nonequivalent groups design*.

```
#perbandingan plot 3 teknik equating klasik nonequivalent groups design
plot(neq1, neq2, neq3)
```

Berikut ini plot Perbandingan plot Skor Hasil *Equating* Klasik *Nonequivalent groups design*. Perbandingan hasil plot *equiting*nya yang menunjukkan bahwa tipe equiting *Equipercentil* dengan metode *Frequency Estimation* (warna hijau muda) menghasilkan grafik yang paling berbeda dengan grafik tipe equiting lainnya. Equiting tipe equiting *Equipercentil* dengan metode *Frequency Estimation* menghasilkan grafik yang tidak linear, berbeda dengan grafik tipe equiting lainnya yang cenderung linear. Hal ini menunjukkan bahwa berdasarkan hasil equiting tipe equiting *Equipercentil* dengan metode *Frequency Estimation*, paket a tidak setara dengan paket b.



Gambar 10.29 Plot Perbandingan Skor Hasil *Equating* Klasik *Nonequivalent groups design.*

Referensi

Albano A (2015). *"A General Linear Method for Equating with Small Samples."* Journal of Educational Measurement, 52(1), 55–69. doi:10.1111/jedm.12062.

Albano, A. D. (2016). *equate: An R package for observed-score linking and equating.* Journal of Statistical Software, 74 (8), 1–36. doi: 10.18637/ jss.v074.i08

Brennan, R.L., dan Kolen, M.J. (2004). *Concordance Between ACT and ITED Scores From Different Popolation*. Jurnal Applied Psichological Measurement, Vol 28. No. 4, July 2004, p. 219-226

Desjardins, C.D., & Bulut, O. (2017). *Handbook of Educational Measurement and Psychometrics Using R (1st ed.)*. Chapman and Hall/CRC. https://doi.org/10.1201/b20498

Hambleton, R.K. & Swaminathan, H. (1985). *Item response theory*. Boston, MA: Kluwer Inc.

Hambleton, R.K., Swaminathan, H., & Rogers, H.J. (1991). *Fundamental of item response theory*. Newbury Park, CA: Sage Publication Inc.

Kolen, M.J. dan Brennan, R.L. (2004). *Test Equating : Methods and Practices*. New York : Springer.

Retnawati, H. (2014). *Teori Respons Butir dan Penerapannya*. Yogyakarta: Parama Publishing.

Analisis Instrumen Penelitian **dengan Teori Tes Klasik dan Modern** Menggunakan **Program R**

Program R merupakan *software* yang sangat *powerfull*, lengkap, gratis, dan menawarkan banyak kemudahan kepada pengguna dengan berbagai paket analisis. Buku ini mencoba menghadirkan proses analisis instrumen penelitian dengan pendekatan *Classical Test Theory* (CTT) dan *Item Response Theory* (IRT) dengan contoh-contoh praktis dan aplikatif menggunakan program R. Materi-materi dalam buku ini disusun dalam bahasa yang sederhana sehingga dapat lebih mudah untuk dipahami oleh para pengguna yang berasal dari kalangan pemula (non statistikawan) disertai dengan pembahasan contoh-contoh aplikatif.

Adapun topik yang dibahas dalam buku ini meliputi: pengenalan Program R, proses instalasi, dan perintah-perintah dasar dalam Program R, membangkitkan data, CTT, IRT unidimensi penskoran dikotomus dan politomus, serta IRT multidimensi penskoran dikotomus dan politomus beserta cara menginterpretasikan hasil analisis data menggunakan program R. Semoga buku ini dapat menjadi referensi pengantar bagi mahasiswa dan dosen dalam belajar dan mempelajari CTT dan IRT untuk menganalisis instrumen penelitian.

ISBN 978-602-498-415-1

UNY PRESS

Jl. H. Affandi (jl. Gejayan), Gg. Alamanda,
Kompleks FT-UNY, Kampus Karangmalang, Yogyakarta,
Kode Pos: 55281, Telp. (0274)589346,
unypress.yogyakarta@gmail.com