

Algoritma dan Struktur Data

Algoritma Pemrograman

Ponco Wali Pranoto, M.Pd
Kelas B TE

Created : Bekti W, M.Pd



Program

- Program: sederetan perintah-perintah yang harus dikerjakan oleh komputer untuk menyelesaikan masalah.
- 3 level bahasa pemrograman:
 1. Bahasa tingkat rendah
 2. Bahasa tingkat menengah
 3. Bahasa tingkat tinggi

Bahasa Tingkat Rendah

- Bahasa mesin
- Berisi: kode-kode mesin yg hanya dapat diinterpretasikan langsung oleh mesin komputer.
- Berupa kode numerik 0 dan 1
- Microcode: sekumpulan instruksi dalam bahasa mesin
- (+) : Eksekusi cepat
- (-) : Sulit dipelajari manusia

Bahasa Tingkat Menengah

- Bahasa Assembly
- Bahasa simbol dari bahasa mesin
- Contoh: ADD, SUB, dll
- Macro instruksi: sekumpulan kode dalam bahasa assembly
- (+) : Eksekusi cepat, masih dapat dipelajari daripada bahasa mesin, file kecil
- (-) : Tetap sulit dipelajari, program sangat panjang

Bahasa Tingkat Tinggi

- The 3rd Generation Programming Language
- Lebih dekat dengan bahasa manusia
- Memberi banyak fasilitas kemudahan dalam pembuatan program, mis.: variabel, tipe data, konstanta, struktur kontrol, loop, fungsi, prosedur, dll
- Contoh: Pascal, Basic, C++, Java
- (+) : Mudah dipelajari, mendekati permasalahan yang akan dipecahkan, kode program pendek
- (-) : Eksekusi lambat, tidak dapat dilakukan langsung oleh komputer (membutuhkan translator)

Translator

- Translator: penerjemah dari bahasa tingkat tinggi ke bahasa tingkat rendah.
- Assembler merupakan penerjemah bahasa Assembly ke bahasa mesin.

Interpreter

- Perintah diterjemahkan baris demi baris jadi program tidak dianalisis seluruhnya dulu tapi bersamaan dengan jalannya program.
- Interpreter adalah suatu program komputer yang mampu menerjemahkan program dari bahasa tingkat tinggi yang dimengerti oleh manusia dan langsung menjalankan program tersebut.
- (+) : mudah bagi user, debugging cepat
- (-) : eksekusi program lambat, tidak langsung menjadi program executable
- Contoh: Basic, List

Compiler

- Seluruh program diterjemahkan.
- Semua perintah (pascal, C++) dirubah dalam bentuk exe atau bahasa assembly.

Paradigma Bahasa Pemrograman



Algoritma

- Algoritma: sederetan langkah-langkah logis yang disusun secara sistematis untuk memecahkan suatu masalah.
- Disebut Logis karena setiap langkah bisa diketahui dengan pasti.
- Algoritma lebih merupakan alur pemikiran untuk menyelesaikan suatu pekerjaan atau suatu masalah.

Syarat Algoritma

- Algoritma harus tidak ambigu
- Algoritma harus tepat
- Algoritma harus pasti
- Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
- Algoritma memiliki nol atau lebih masukan,
- Algoritma memiliki satu atau lebih keluaran.
- Algoritma harus efektif

Belajar Memprogram dan Belajar Bahasa Pemrograman

- Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu yang mudah dibaca dan dipahami.
- Belajar bahasa pemrograman adalah belajar memakai suatu bahasa, aturan tata bahasanya, instruksi-instruksinya, tata cara pengoperasian compiler-nya untuk membuat program yang ditulis dalam bahasa itu saja.

Notasi Algoritma

- Penulisan algoritma tidak tergantung dari spesifikasi bahasa pemrograman dan komputer yang mengeksekusinya.
- Notasi algoritma bukan notasi bahasa pemrograman tetapi dapat diterjemahkan ke dalam berbagai bahasa pemrograman.

Notasi Algoritma

I. Uraian kalimat deskriptif (narasi)

Contoh:

Algoritma Kelulusan_mhs

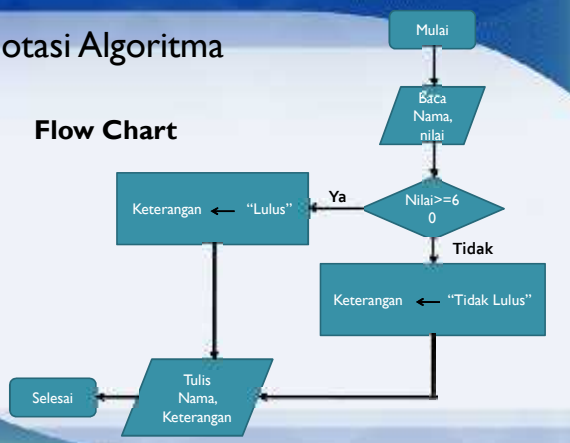
Diberikan nama dan nilai mahasiswa, jika nilai tersebut lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus jika nilai lebih kecil dari 60 maka dinyatakan tidak lulus.

DESKRIPSI :

1. baca nama dan nilai mahasiswa.
2. jika nilai ≥ 60 maka
3. keterangan \leftarrow lulus
4. tetapi jika
5. keterangan \leftarrow tidak lulus.
6. tulis nama dan keterangan

Notasi Algoritma

2. Flow Chart



Notasi Algoritma

3. Pseudo Code

Ada 3 bagian: Judul, Deklarasi, Deskripsi.

Algoritma kelulusan

Deklarasi

nama, keterangan : string

nilai : integer

Deskripsi

read (nama, nilai)

if nilai ≥ 60 then

keterangan \leftarrow 'lulus'

else

keterangan \leftarrow 'tidak lulus'

write(nama, keterangan)

Aturan Pseudo Code

- **Judul algoritma**
Bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) tentang algoritma tersebut. Nama sebaiknya singkat dan menggambarkan apa yang dilakukan oleh algoritma tersebut.
- **Deklarasi**
Bagian untuk mendefinisikan atau mendeklarasikan semua apa yang digunakan atau dibutuhkan dalam pemrograman.
- **Deskripsi**
Bagian ini berisi uraian langkah-langkah penyelesaian masalah.

Operator Aritmetik

- /, *, div, mod \rightarrow level tinggi
- +, - \rightarrow level rendah
- Mod dan div hanya untuk bilangan bulat!

• Contoh :

• $5 - 2 + 1 = ?$

• $5 - 2 * 3 = ?$

• $(6 + 3 * 2) / 6 - 2 * 3 = ?$

Tipe Data

- Bilangan bulat (Shortint , Integer, Longint, Byte, Word)
- Boolean (Boolean, ByteBool , WordBool, LongBool)
- Bilangan real (Real, Single, Double, Extended, Comp)
- Karakter
- String

Latihan

Buatlah notasi algoritma untuk :

- a. Menghitung luas dan keliling lingkaran dengan memasukkan nilai jari-jari
- b. Mengidentifikasi suatu bilangan apakah bilangan tersebut ganjil atau genap