

LAPORAN PENELITIAN

PROGRAM SELEKSI PENELITIAN BIDANG TEKNOLOGI INFORMASI DAN KOMUNIKASI

Kategori Penelitian:

INFRASTRUKTUR TIK

Kerangka Kerja Model Akses Terintegrasi Untuk Peningkatan Kualitas Layanan Akses Internet Di Lingkungan Jaringan Berkecepatan Rendah

DEPARTEMEN KOMUNIKASI DAN INFORMATIKA

Gedung Departemen Komunikasi Dan Informatika Lantai 5

Jl. Medan Merdeka Barat No. 9

Jakarta 10110

Judul Penelitian : Kerangka Kerja Model Akses Terintegrasi
Untuk Peningkatan Kualitas Layanan Akses Internet
Di Lingkungan Jaringan Berkecepatan Rendah

Program : Seleksi Penelitian Bidang Teknologi Informasi dan
Komunikasi

Kategori : Infrastruktur TIK

Peneliti : Ratna Wardani, S.Si., MT

Tempat/Tgl lahir : Padang, 18 Desember 1970

Alamat : Panggungan Kidul RT 04 RW 33 Trihanggo,
Gamping, Sleman, Yogyakarta 55291

Jenjang pendidikan
yang sedang dijalani : ~~S1~~ / ~~S2~~ / S3 * (Coret yang tidak perlu)

Universitas/Program studi : Universitas Gadjah Mada / Teknik Elektro

No. Induk Mahasiswa : 04/1453/PS

No. Telp/mobilephone : 08156804204

E-mail : ratna@uny.ac.id

Yogyakarta, 25 November 2007

Menyetujui
Pengelola S3 Teknik Elektro UGM

Peneliti

Dr. Ir. Sasongko Pramono Hadi, DEA
NIP. 130 815 059

Ratna Wardani, S.Si.,MT.
NIM. 04/1453/PS

ABSTRAK

Kualitas koneksi yang tidak handal (*low-quality connection*) merupakan suatu keadaan, ketika sumber daya yang disediakan oleh sistem tidak dapat memenuhi persyaratan QoS aplikasi, sehingga aplikasi tidak dapat beroperasi secara normal. Keterbatasan *bandwidth* menyebabkan koneksi yang lamban bahkan koneksi yang putus-sambung sehingga beberapa jenis aplikasi untuk akses Internet tidak dapat dilakukan.

Hingga saat ini, *World Wide Web* (WWW) dan teknologi *browser* yang terkait belum dapat menyediakan dukungan bagi akses Internet pada lingkungan jaringan berkecepatan rendah. Pada dasarnya, aplikasi Internet dikembangkan dengan asumsi jaringan memiliki kualitas koneksi yang baik. Dengan kondisi tersebut, aplikasi Internet pada dasarnya akan menghentikan operasi yang sedang dilakukan oleh pengguna jika sumber daya sistem tidak dapat memenuhi persyaratan sumber daya yang dibutuhkan oleh aplikasi. Berdasarkan kondisi tersebut, perlu dikembangkan suatu model akses yang lebih sesuai untuk lingkungan dengan kualitas koneksi yang tidak handal.

Penelitian ini mengembangkan konsep baru untuk mendefinisikan model akses terintegrasi untuk meningkatkan kualitas layanan akses Internet pada jaringan dengan kualitas koneksi yang rendah. Hasil yang dicapai dalam penelitian ini adalah:

1. Kerangka kerja model akses terintegrasi yang diperlukan sebagai dasar pengembangan model akses. Pengembangan kerangka kerja menggunakan pendekatan model akses dari perspektif pengguna melalui konsep *User-Oriented QoS*. Model akses seperti ini lebih sesuai untuk kondisi jaringan dengan kualitas koneksi yang tidak handal.

2. Spesifikasi model akses yang direpresentasikan dengan skema sebagai berikut :

$$S_i : \{ (S_n, f) \mid (a_i, [q_{exp}]) \mid (f \rightarrow S_i \vee \neg f \rightarrow S_i) \}$$

Dengan skema tersebut, pola akses dan parameter kualitas layanan yang diharapkan pengguna dapat dinyatakan dengan jelas dan mudah.

3. Rancangan *prototype browser* dibuat sebagai dasar implementasi kerangka kerja model akses terintegrasi.

Kerangka kerja yang dihasilkan dari penelitian ini menjadi dasar pengembangan model akses yang menyediakan suatu pola akses yang lebih dinamis dengan memberi kesempatan kepada pengguna untuk menyatakan kualitas layanan akses yang diharapkan dan menetapkan akses alternatif jika tingkat kualitas layanan yang diharapkan tidak dapat dipenuhi oleh sistem. Dengan demikian, model akses lebih sesuai digunakan untuk akses Internet pada kondisi jaringan dengan kualitas koneksi yang tidak handal.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala nikmat, rahmat, kesempatan dan hidayahNya bagi penulis untuk menyelesaikan penelitian yang berjudul *Kerangka Kerja Model Akses Terintegrasi untuk Peningkatan Kualitas Layanan Akses Internet di Lingkungan Jaringan Berkecepatan Rendah* dengan baik tanpa ada halangan apapun.

Penelitian ini dapat terlaksana atas biaya DIPA Badan Penelitian dan Pengembangan Sumber Daya Manusia tahun anggaran 2007 Nomor: 0164.0/059-06.0/-/2007 Departemen Komunikasi dan Informatika dan dukungan dari berbagai pihak. Penulis juga menyampaikan ucapan terima kasih dan penghargaan kepada yang terhormat:

1. Badan Penelitian dan Pengembangan Sumber Daya Manusia, Departemen Komunikasi dan Informatika.
2. Bapak Baringin Batubara, selaku Kepala Pusat Penelitian dan Pengembangan Pos dan Telekomunikasi.
3. Prof. Ir. F. Soesianto, B.Sc., Ph.D., selaku Promotor, Ir. Lukito Edi Nugroho, M.Sc., Ph.D., selaku Ko-Promotor I, Dr. Ahmad Ashari, M.Kom., selaku Ko-Promotor II yang telah memberikan bimbingan, arahan dan perbaikan untuk kesempurnaan penelitian.
4. Semua pihak yang telah membantu terlaksananya penelitian ini.

Penulis menyadari bahwa masih banyak hal-hal yang perlu dikaji lebih lanjut untuk pengembangan penelitian di bidang ini. Penulis berharap semoga penelitian ini dapat memperkaya khasanah ilmu pengetahuan di bidang teknologi informasi

Yogyakarta, 25 November 2007

Peneliti

Ratna Wardani

DAFTAR ISI

Lembar halaman Judul	i
Lembar Identitas dan Pengesahan	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I. PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah	2
C. Tujuan dan Sasaran	2
D. Ruang Lingkup Penelitian.....	3
E. Batas-batas Penelitian	5
BAB II. STUDI LITERATUR	8
A. Tinjauan Pustaka.....	8
1. Konsep Kualitas Layanan	8
2. Lapis Abstraksi Kualitas Layanan	10
3. Manajemen dan Mekanisme Pencapaian QoS	10
4. Kategori QoS	12
5. Persepsi Pengguna	13
B. Kerangka Pemikiran Konseptual	16
1. <i>Web Browser</i> sebagai <i>User Agent</i>	16
2. <i>Hypertext Transfer Protocol (HTTP)</i>	17
a. Model Hubungan HTTP	17
b. Format <i>Request</i>	18
c. Format <i>Response</i>	18
d. <i>HTTP Header</i>	19
BAB III. METODOLOGI PENELITIAN	25
A. Materi Penelitian	25
B. Cara Penelitian	26
BAB IV. HASIL PENELITIAN DAN PEMBAHASAN	28
A. Persyaratan <i>User_oriented QoS</i>	28

1. Parameter <i>User-oriented QoS</i>	28
2. Model Konseptual <i>User-oriented QoS</i>	31
B. Kerangka Kerja Model Akses Terintegrasi	33
C. Desain Arsitektur Kerangka Kerja	36
D. Rasionalisasi Kerangka Kerja	41
1. Studi kasus : Aplikasi <i>Browsing</i>	42
2. Alur Proses	43
3. Arsitektur Modul Perangkat Lunak	44
4. Uji Fungsionalitas Komponen	45
BAB V. KESIMPULAN DAN SARAN	47
A. Kesimpulan	47
B. Saran	47
DAFTAR PUSTAKA	48
LAMPIRAN.....	50

DAFTAR TABEL

Tabel 1. <i>Delay Audio</i> dan Pendengaran Manusia	14
Tabel 2. Batasan waktu untuk Transmisi Satu Arah.....	14
Tabel 3. Persepsi Pengguna terhadap <i>Jitter</i>	15
Tabel 4. Persepsi Pengguna terhadap <i>Error</i>	15
Tabel 5. Karakteristik dan parameter QoS	30
Tabel 6. <i>Mapping</i> parameter QoS pengguna ke QoS browser	43

DAFTAR GAMBAR

Gambar 1. Arsitektur Umum <i>Browser</i> dan <i>Web Server</i>	17
Gambar 2. Komponen dalam rantai <i>request/response</i> HTTP	18
Gambar 3. Diagram Transisi Model Akses.....	31
Gambar 4. Komponen Kerangka Kerja Model Akses	34
Gambar 5. Diagram Blok Fungsional Kerangka Kerja.....	35
Gambar 6. <i>Use Case Diagram</i> untuk Akses Pengguna	36
Gambar 7. <i>Use Case Diagram</i> untuk Sistem	36
Gambar 8. Pemetaan Awal Persyaratan Fungsionalitas	38
Gambar 9. <i>Class Diagram</i> Komponen	39
Gambar 10. <i>Class Diagram</i> parameter	39
Gambar 11. <i>Class Diagram</i> browser	40
Gambar 12. <i>Class Diagram</i> variable	40
Gambar 13. <i>Class Diagram</i> media	41
Gambar 14. Diagram Model Akses Pengguna	42
Gambar 15. Diagram Implementasi	44
Gambar 16. Tampilan Hasil Pengujian Akses Terpenuhi.....	45
Gambar 17. Tampilan Hasil Pengujian Akses Tidak Terpenuhi	46

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Sebagian besar negara sedang berkembang seperti Indonesia masih menghadapi masalah kualitas telekomunikasi yang rendah. Hal ini mengakibatkan keterbatasan dalam mengakses informasi. Kualitas layanan akses Internet berkecepatan tinggi dan handal baru bisa dirasakan oleh sebagian kecil masyarakat, karena biaya yang mahal. Dengan perkembangan teknologi komputer dan tersedianya jaringan telepon, Internet menjadi media yang sangat potensial dalam menyediakan akses ke berbagai sumber informasi secara elektronik. Sayangnya, keterbatasan *bandwidth* masih menjadi satu kendala. *Bandwidth* yang rendah menyebabkan akses yang begitu lambat. Misalnya, butuh waktu yang cukup lama untuk mengambil (*download*) suatu informasi yang diinginkan ketika pengguna melakukan akses *browsing*.

Pada saat yang bersamaan, perkembangan dunia komunikasi mengarah pada sistem berbasis *web*. Layanan kepada pelanggan dilakukan melalui *website*, tidak lagi melalui telepon. Demikian juga untuk pertukaran dokumen, pengambilan formulir aplikasi pendidikan, berbelanja dan berita. Infrastruktur jaringan yang tidak handal membatasi pengguna untuk mengakses berbagai fasilitas yang disediakan oleh Internet. Meskipun solusi infrastruktur jaringan dengan biaya terjangkau sudah banyak dikembangkan, namun masih terdapat kendala berkaitan dengan perangkat lunak aplikasi yang digunakan untuk mengakses Internet. *Browser*, *FTP*, *email client* dan aplikasi lainnya tidak sesuai dengan koneksi Internet yang tidak handal.

Hingga saat ini, *World Wide Web* (WWW) dan teknologi *browser* yang terkait belum dapat menyediakan dukungan bagi akses Internet pada lingkungan jaringan berkecepatan rendah. Pada dasarnya, aplikasi Internet dikembangkan dengan asumsi jaringan memiliki kualitas koneksi yang baik. Dengan kondisi tersebut, aplikasi Internet dapat menghentikan operasi yang sedang dilakukan oleh pengguna jika sumber daya sistem tidak dapat memenuhi persyaratan sumber daya yang dibutuhkan oleh aplikasi. Misalnya, pada situs berita yang memiliki konten gambar yang cukup banyak, *loading image* akan sangat lambat pada kecepatan koneksi 28 Kbps (Prevost, 2001). Bagi pengguna Internet dengan koneksi melalui *bandwidth* rendah, layanan terhadap akses ini hampir tidak dimungkinkan. Keadaan seperti ini harus diterima

oleh pengguna tanpa memiliki kesempatan untuk dapat mengalihkan ke model akses lain yang dimungkinkan.

Rendahnya kualitas telekomunikasi menjadi suatu masalah yang harus mendapat perhatian cukup serius. Masih banyak masyarakat yang berada pada lingkungan dengan kualitas koneksi Internet yang tidak memadai. Kondisi tersebut menghalangi masyarakat pengguna Internet untuk memanfaatkan potensi Internet sebagai sumber informasi dan sebagai sarana komunikasi global. Untuk itu perlu disediakan suatu model akses bagi semua orang yang memiliki keterbatasan terhadap akses Internet.

B. Rumusan Masalah

Kualitas koneksi Internet yang tidak handal (*low-quality Internet connection*) menyatakan suatu keadaan, ketika sumber daya yang disediakan oleh sistem tidak dapat memenuhi persyaratan kualitas layanan atau QoS aplikasi, sehingga aplikasi tidak dapat beroperasi secara normal. Kualitas layanan untuk akses Internet pada situasi jaringan dengan kualitas koneksi yang rendah (*low-quality Internet connection*) belum dapat terpenuhi melalui model akses yang tersedia saat ini. Pertama, perilaku akses yang tertentu dan sangat tergantung pada aplikasi yang digunakan. Kedua, berkaitan dengan belum tersedianya model akses terintegrasi di lingkungan aplikasi yang digunakan untuk melakukan akses Internet. Artinya, dalam melakukan akses pengguna sangat dibatasi oleh lingkungan aplikasi yang digunakannya.

Pada kondisi jaringan dengan kualitas koneksi yang rendah dan keterbatasan *bandwidth*, model akses yang ada saat ini tidak sesuai dengan infrastruktur yang tersedia. Untuk itu penelitian ini diarahkan untuk mengembangkan kerangka kerja model akses yang lebih sesuai pada kondisi jaringan dengan kualitas koneksi yang rendah. Pada model akses yang akan dikembangkan, harus tersedia fasilitas bagi pengguna untuk menyatakan pola akses yang diinginkan. Pola akses yang dinyatakan oleh pengguna kemudian ditransformasikan ke aras yang lebih rendah untuk proses selanjutnya.

C. Tujuan dan Sasaran

Penelitian ini mengembangkan kerangka kerja model akses terintegrasi untuk mengatasi masalah kualitas layanan akses Internet pada kondisi jaringan yang tidak handal. Kerangka kerja ini memuat kerangka pemodelan akses Internet yang

menyediakan spesifikasi persyaratan kualitas layanan dan mekanisme untuk memenuhi pencapaian kualitas layanan akses Internet pengguna. Melalui kerangka kerja ini, tersedia layanan yang optimal dalam melakukan akses Internet pada jaringan dengan kualitas koneksi yang rendah sehingga keterbatasan infrastruktur jaringan tidak lagi menjadi kendala bagi pengguna Internet dalam memanfaatkan potensi Internet.

Secara rinci, tujuan penelitian adalah:

- a. mengkaji konsep QoS serta mengembangkan konsep *user-oriented QoS* dan persyaratannya guna menemukan kerangka dasar bagi model akses terintegrasi;
- b. menemukan suatu model akses yang dapat menunjukkan relasi antara *user-oriented QoS* dengan QoS tradisional pada lapisan yang lebih rendah; dan
- c. mengembangkan kerangka kerja untuk penetapan QoS yang mengimplementasikan model akses terintegrasi guna meningkatkan aspek ketergunaan melalui fleksibilitas layanan bagi pengguna dalam mengakses Internet pada jaringan dengan kualitas koneksi yang rendah.

Kerangka kerja model akses terintegrasi yang dikembangkan dalam penelitian ini diharapkan dapat menjadi dasar pengembangan arsitektur layanan Internet maupun untuk melengkapi arsitektur yang sudah ada melalui penyediaan mekanisme untuk menyatakan persyaratan kualitas layanan dari perspektif pengguna.

Sasaran yang ingin dicapai di akhir penelitian ini adalah pengembangan model akses Internet yang sesuai untuk akses Internet pada lingkungan jaringan dengan kualitas koneksi yang kurang handal. Melalui model akses ini disediakan mekanisme spesifikasi akses yang dapat menyesuaikan dengan pola akses pengguna. Dengan model ini diharapkan pengguna tetap dapat memanfaatkan potensi Internet sebagai sumber informasi maupun sebagai media komunikasi meskipun pada lingkungan jaringan dengan kualitas koneksi yang tidak handal.

D. Ruang Lingkup Penelitian

Penelitian ini menggunakan pendekatan kualitas layanan subyektif (*User-Oriented QoS*) untuk mengatasi masalah dalam pengaksesan Internet pada kondisi jaringan yang kurang handal (*low-quality Internet connection*). Pada konteks penelitian ini, *User-Oriented QoS* mengacu pada persyaratan kualitas layanan pengguna yang didefinisikan berdasarkan persepsi dan ekspektasi pengguna terhadap

nilai QoS yang diterimanya. Melalui pendekatan ini, pemenuhan persyaratan kualitas layanan pengguna dilakukan melalui penyediaan suatu model akses yang dapat menggabungkan beberapa pola akses pengguna seperti *email*, *browsing* dan *file transfer* ke dalam suatu model akses. Melalui model ini pengguna dapat menggunakan aplikasi yang berbeda dalam mengekspresikan pola akses yang diharapkan. Pada kondisi jaringan dengan kualitas koneksi yang tidak handal, pendekatan ini diharapkan dapat meningkatkan kepuasan pengguna terhadap layanan Internet dan meningkatkan aspek ketergunaan bagi pengguna.

Pemilihan konsep *User-Oriented QoS* sebagai pendekatan didasarkan pada lima hal berikut:

1. Melalui konsep *User-Oriented QoS*, pengembangan model akses didasarkan pada persepsi pengguna sehingga tidak tergantung pada teknologi tertentu untuk validasinya ;
2. Nilai QoS bersifat relatif, ini berarti bahwa spesifikasi QoS pengguna yang satu tidak “lebih baik” dari spesifikasi QoS pengguna lainnya ;
3. Penyediaan ukuran persyaratan maksimum dan minimum QoS aplikasi yang masih dapat diterima oleh pengguna ;
4. Dapat mengubah pola akses yang statis menjadi pola akses dinamis untuk mengakomodasi ketidakhandalan jaringan yang menyebabkan waktu respons yang panjang, ketidaktersediaan layanan maupun kegagalan operasi aplikasi ; dan
5. Menyediakan strategi alternatif yang konsisten dengan kondisi waktu respons yang tidak diharapkan.

Pada konsep *User-Oriented QoS*, pengguna diberi kesempatan untuk melakukan spesifikasi kualitas layanan dengan mudah melalui beberapa parameter yang dipahami oleh pengguna. Pada lapis pengguna disediakan suatu antarmuka (*interface*) agar pengguna dapat menetapkan spesifikasi akses yang diinginkan beserta persyaratan kualitas layanan yang terkait dengan akses yang dilakukannya. Untuk mengimplementasikan model akses berbasis *User-Oriented QoS*, perlu dikembangkan kerangka kerja sebagai kerangka arsitektural model akses yang menyediakan mekanisme spesifikasi QoS berdasarkan representasi kualitas layanan pengguna.

Spesifikasi QoS dinyatakan dalam parameter-parameter yang berbeda untuk setiap tingkat abstraksi. Untuk itu kerangka kerja yang dikembangkan selain memiliki

mekanisme untuk menerima parameter QoS pada lapisan pengguna, tetapi juga harus memiliki mekanisme untuk mentransformasikan nilai-nilai QoS yang dispesifikasi oleh pengguna ke parameter aplikasi. Translasi parameter QoS pengguna ke parameter aplikasi merupakan satu hal yang tidak dapat diabaikan, karena merupakan bagian mekanisme pencapaian QoS pengguna. Dalam hal ini perlu dievaluasi mekanisme spesifikasi QoS yang ada saat ini sebagai dasar pengembangan kerangka kerja guna mendukung pemenuhan kualitas layanan berorientasi pengguna (*User-Oriented QoS*).

Mekanisme penetapan QoS disediakan untuk merealisasikan pola akses yang ditetapkan pengguna berdasarkan nilai-nilai parameter QoS yang didefinisikan sebelumnya. Diharapkan kerangka kerja dapat mengintegrasikan beberapa model akses yang saat ini masih bersifat parsial, sehingga memungkinkan akses yang berbeda dilakukan menggunakan satu model akses yang terintegrasi. Dengan demikian, pengguna memiliki fleksibilitas dalam mengontrol perilaku aksesnya.

E. Batas-batas Penelitian

Penelitian ini mengkaji beberapa aspek yang dapat digunakan sebagai pendekatan untuk mengatasi masalah akses pada jaringan dengan kualitas koneksi yang tidak handal. Pertama, pola akses pengguna yang bervariasi. Misalnya, pengguna tidak selalu mengharapkan respons seketika terhadap akses yang dilakukannya. Pada jenis akses tertentu misalnya ketika menggunakan layanan *web*, pengguna Internet dapat memberi toleransi waktu untuk respons yang diharapkan. Bahkan untuk layanan *email* tidak selalu diperlukan balasan (*reply*) bagi *email* yang dikirim. Tersedianya suatu pilihan untuk menentukan spesifikasi akses tertentu yang diinginkan akan lebih bermanfaat bagi pengguna ketika suatu akses gagal untuk dieksekusi, karena keterbatasan sumberdaya. Pada konteks ini, jika layanan *video conference* yang bersifat *real-time* tidak dimungkinkan, layanan *chatting* melalui IRC atau komunikasi *off-line* melalui grup diskusi dirasakan lebih bermanfaat bagi pengguna daripada gagal sama sekali.

Aspek kedua ditinjau dari sisi aplikasi. Aplikasi Internet sebagai salah satu sistem terdistribusi memiliki dua karakteristik yang berbeda (Katchabaw, 2002). Karakteristik pertama berkaitan dengan persyaratan kualitas layanan, atau sering disebut *Quality of Service* disingkat QoS. Aplikasi dengan persyaratan QoS yang tidak ketat masih dapat berfungsi secara benar meskipun persyaratan QoS tidak

terpenuhi. Sebaliknya, aplikasi dengan persyaratan QoS yang ketat hanya berfungsi secara benar jika persyaratan QoS terpenuhi. Aplikasi email misalnya, dapat dijalankan tanpa adanya jaminan QoS (*best-effort service*). *Web browsing* dan *chatting* hanya memerlukan *bandwidth* rendah dan bahkan cukup dapat bertoleransi terhadap beberapa kondisi kongesti jaringan. Akses *web* dapat dijalankan tanpa jaminan QoS, meskipun pada dasarnya tetap memerlukan jaminan berupa *delay* yang rendah dan *bandwidth* yang tinggi tergantung pada isi (*content*) yang diakses. Sebaliknya aplikasi *audio* dan *video* yang bersifat interaktif dan *real-time* diperlukan jaminan QoS yang dapat diprediksi (*predictable*) berupa *delay*, *jitter*, *loss control* yang cukup rendah, *bandwidth*, dan *reliability* yang tinggi. Karakteristik kedua berkaitan dengan sifat dinamis sistem terdistribusi. Persyaratan QoS dapat berbeda untuk pengguna yang berbeda pada aplikasi yang sama atau untuk pengguna yang sama pada waktu yang berbeda. Bahkan reliabilitas yang rendah juga cukup memadai untuk beberapa jenis aplikasi. Misalnya, informasi dalam aplikasi *video conference* harus ditransmisikan secara *real-time*, tetapi tidak selalu dengan kualitas suara dan gambar yang maksimal (Williams, 1992).

Aspek ketiga ditinjau dari sisi konsep kualitas layanan (QoS). QoS merupakan himpunan atribut-atribut yang dapat diterima pengguna dan dinyatakan dalam notasi yang dipahami pengguna melalui parameter-parameter yang dapat bersifat obyektif maupun subyektif (Bertino et al., 2004). Kualitas layanan obyektif berkaitan dengan parameter QoS sistem (*bandwidth*, *delay*, *jitter*, *packet loss*, dan lain-lain) yang dapat diukur secara kuantitatif dan diverifikasi nilainya. Berbeda dengan kualitas layanan obyektif, kualitas layanan subyektif berhubungan dengan kualitas layanan yang dirasakan oleh pengguna. Kualitas layanan subyektif lebih sulit untuk diukur karena pemahaman pengguna terhadap kualitas tidak selalu mengikuti solusi teknis yang memiliki dampak langsung terhadap kualitas layanan obyektif.

Parameter subyektif QoS didasarkan pada persepsi dan ekspektasi pengguna. Pada kondisi *low-quality connection*, aplikasi yang memiliki persyaratan QoS yang ketat gagal dieksekusi atau terputusnya koneksi menyebabkan suatu aplikasi tidak dapat melengkapi operasinya. Secara umum dapat dinyatakan bahwa persyaratan QoS aplikasi terhadap layanan sistem menjadi sulit dipenuhi pada kondisi *low-quality Internet connection*. Pada kondisi ini, parameter subyektif dapat dimanfaatkan untuk memenuhi persepsi dan ekspektasi pengguna terhadap QoS melalui pengalihan model akses jika suatu akses gagal dieksekusi dalam batasan waktu tertentu. Untuk itu

diperlukan suatu model akses yang memungkinkan pengguna tetap dapat melakukan akses Internet pada kondisi kualitas koneksi yang rendah melalui pemberian kesempatan kepada pengguna untuk menetapkan spesifikasi akses yang diinginkan.

BAB II

STUDI LITERATUR

A. Tinjauan Pustaka

Kualitas layanan atau QoS merupakan suatu konsep yang awalnya dikembangkan pada bidang jaringan. Konsep kualitas layanan menyediakan dan mengatur kriteria-kriteria teknis (seperti sumber daya dan kinerja jaringan) pada layer jaringan. Perkembangan teknologi komunikasi dan komputasi mendukung perluasan konsep kualitas layanan hingga pada layer pengguna. Konsep kualitas layanan tidak hanya menyatakan properti yang berkaitan dengan aspek aplikasi tertentu, tapi juga harus dapat merefleksikan nilai-nilai yang berorientasi pada pengguna dan dapat dinyatakan melalui ekspresi yang dipahami oleh pengguna.

Penelitian mengenai kualitas layanan yang dilakukan beberapa tahun terakhir telah memberikan beberapa hasil penting. Sebagian besar area penelitian difokuskan pada dukungan QoS untuk aplikasi multimedia terdistribusi, sistem operasi, transport layer, jaringan, dan bahasa formal. Dari sekian banyak penelitian, hanya sedikit yang memfokuskan pada isu-isu QoS pada lapis pengguna. Pada bagian berikut, diuraikan tentang terminologi yang terkait dengan kualitas layanan dan beberapa penelitian dalam rangka pengembangan kualitas layanan (QoS).

1. Konsep Kualitas Layanan

Kualitas layanan atau *Quality of Service* (QoS) merupakan suatu konsep yang awalnya dikembangkan pada lapisan sistem dan jaringan. Pada lapisan jaringan, konsep kualitas layanan dikembangkan guna menyediakan dan mengatur kriteria-kriteria teknis seperti sumber daya dan kinerja jaringan. Perkembangan teknologi komunikasi dan komputasi mendukung perluasan konsep kualitas layanan hingga pada lapisan pengguna. Konsep kualitas layanan tidak hanya menyatakan properti yang berkaitan dengan aspek aplikasi tertentu, tapi juga harus dapat merefleksikan nilai-nilai yang berorientasi pada pengguna dan dapat dinyatakan melalui ekspresi yang dipahami oleh pengguna.

Secara tradisional, QoS mengacu pada karakteristik tertentu suatu sistem. Karakteristik tersebut tidak dapat dikontrol oleh pengguna dan hanya menggambarkan aspek layanan yang disediakan oleh jaringan. Ray (2000) dalam penelitiannya mendefinisikan dua parameter yang digunakan untuk mengontrol QoS yaitu *delay* dan

throughput. Dua aspek *delay* yang berpengaruh terhadap QoS adalah *end-to-end delay* dan *jitter*. Tsalianis and Economides (2000) mendefinisikan kualitas layanan (QoS) sebagai himpunan karakteristik kuantitatif dan kualitatif sistem telekomunikasi yang penting untuk memenuhi persyaratan fungsionalitas aplikasi dan memenuhi tingkat kepuasan pengguna. Untuk itu jaringan harus menyediakan dukungan pencapaian tingkat QoS agar persyaratan QoS aplikasi dan pengguna terpenuhi. Cisco mendefinisikan QoS sebagai kemampuan suatu jaringan menyediakan layanan yang lebih baik melalui penyediaan prioritas berupa pemakaian *bandwidth*, pengaturan *jitter* dan *latency* serta perbaikan karakteristik *packet loss* (Cisco System, 2001). Venkateswaran (2002) mendefinisikan lima karakteristik unjuk-kerja jaringan yang mewakili parameter QoS, yaitu *Latency* atau *delay*, *Jitter*, *Throughput*, *Packet Loss* dan *Availability*. Beberapa definisi QoS telah mengizinkan spesifikasi beberapa persyaratan pengguna, namun belum sepenuhnya dapat didukung oleh jaringan yang mendasarinya karena belum adanya definisi yang jelas dan konsisten tentang keterkaitan QoS dengan persyaratan pengguna.

Ditinjau dari sisi infrastruktur jaringan, peningkatan QoS difokuskan pada sisi penyedia layanan (*provider*) dan analisis kinerja jaringan untuk pemenuhan kebutuhan kualitas suatu aplikasi. Dari sisi penyedia layanan, QoS dinyatakan sebagai tingkat kualitas yang diharapkan dapat ditawarkan oleh penyedia layanan kepada pengguna layanan (ETSI TR 00019, 2006). Beberapa karakteristik yang memenuhi syarat untuk kualitas layanan yaitu minimalisasi tunda pengiriman (*delay*), minimalisasi variasi tunda (*jitter*) dan konsistensi dari ketersediaan *throughput* (jumlah *bandwidth* yang tersedia untuk suatu aplikasi).

Berdasarkan beberapa definisi tersebut, istilah QoS secara umum lebih banyak digunakan untuk menyatakan kinerja sistem, daripada aspek operasional sistem seperti fungsionalitas. Konsep QoS dinyatakan melalui abstraksi pada lapisan sistem. Ini berarti bahwa nilai-nilai QoS yang diterima oleh pengguna didasarkan pada nilai-nilai obyektif yang ditawarkan oleh sistem seperti *bandwidth* yang tinggi, *delay*, *jitter* dan *data loss* yang rendah. Untuk itu mekanisme yang dikembangkan untuk pencapaian QoS didasarkan pada aspek teknologi jaringan yang tersedia melalui mekanisme kontrol *traffic-handling*.

Pada jaringan dengan kualitas koneksi tinggi tentu saja persyaratan QoS dapat terpenuhi, tetapi tidak pada jaringan dengan kualitas koneksi yang rendah. Untuk itu penelitian ini lebih difokuskan pada konsep kualitas layanan berdasar persepsi

pengguna, agar pemenuhan QoS pada jaringan dengan kualitas koneksi yang rendah tetap dapat terjamin. Pendekatan model akses berbasis persepsi pengguna dikembangkan agar pengguna memiliki kesempatan untuk mengekspresikan persyaratan QoS untuk layanan yang diterimanya melalui parameter QoS yang dipahaminya. Parameter-parameter tersebut dapat dikonfigurasi melalui aplikasi-aplikasi yang digunakan untuk melakukan akses Internet. Parameter-parameter itu selanjutnya dapat ditransformasikan ke parameter yang sesuai di sistem/jaringan.

2. Lapis Abstraksi Kualitas Layanan

Beberapa spesifikasi aplikasi memuat jaminan kualitas layanan. Agar dapat memenuhi jaminan ini, aplikasi memerlukan dukungan jaminan kualitas layanan dari infrastruktur sistem. Struktur ini mengimplikasikan bahwa kualitas layanan harus dapat dinyatakan pada lapis abstraksi yang berbeda.

El-Khatib (2005) dalam penelitiannya menggunakan model QoS yang terdiri dari tiga lapis konseptual, yaitu lapis pengguna, lapis aplikasi dan lapis sistem. Pada lapis abstraksi yang berbeda, parameter QoS dapat dinyatakan dan diukur secara berbeda. Nilai QoS pada lapis pengguna umumnya dikaitkan dengan *perceptual quality* yaitu tingkat kepuasan pengguna terhadap kualitas layanan yang diberikan oleh sistem. QoS pada lapis aplikasi dinyatakan melalui parameter yang berkaitan dengan *media control* seperti *frame rate*, resolusi, warna, kualitas suara, dan lain-lain. Sedangkan QoS pada lapis sistem dinyatakan melalui parameter yang berkaitan dengan *traffic control* seperti *throughput*, *bandwidth*, *loss rate*, *delay*, dan *jitter*. Dengan perbedaan abstraksi ini, kualitas layanan harus dapat diinterpretasikan dan ditranslasikan ke spesifikasi QoS pada lapis abstraksi yang berbeda.

3. Manajemen dan Mekanisme Pencapaian QoS

Manajemen QoS merupakan rangkaian proses yang dilakukan untuk melakukan pengawasan dan pengendalian sumber daya (jaringan, sistem, *middleware*, aplikasi dan lain-lain) yang terkait dengan penyediaan tingkat QoS yang memadai. Seluruh sumber daya harus dapat dikoordinasikan agar dapat memenuhi tingkat QoS yang sesuai dengan kebutuhan pengguna. Pada prinsipnya manajemen dan mekanisme pencapaian QoS secara hirarkis dapat ditinjau dari empat tingkatan, yaitu QoS pada level jaringan, QoS pada level sistem operasi, QoS pada level *middleware* dan QoS pada level aplikasi (Agedal, 2001).

Pada level jaringan, penelitian tentang QoS (Ray, 2000), (Venkateswaran, 2002) menekankan pada mekanisme jaringan dalam mengatur elemen-elemen jaringan, *hardware* maupun *software* untuk mendukung suatu layanan menggunakan teknologi ATM, *Integrated Service (IntServ)*, *Differentiated Service (DiffServ)* dan *Multi-Protocol Label Switching (MPLS)*. Manajemen QoS dilakukan melalui pencadangan sumberdaya, klasifikasi paket, pemisahan aliran lalu lintas, penjadwalan dan pengaturan kecepatan aliran paket yang masuk ke jaringan. Mekanisme pada level jaringan nantinya menjadi dasar bagi level di atasnya untuk merealisasikan jaminan QoS.

Pada level sistem operasi, mekanisme QoS digunakan untuk menjamin persyaratan QoS setiap sistem melalui penyediaan sumber daya lokal ke sistem. Beberapa sistem operasi seperti Eclipse (Pike, et al, 1995), 2K (Kon, et al, 2000), Nemesis (Leslie, et al, 1996) dikembangkan untuk dapat menyediakan jaminan QoS. Hal ini dimungkinkan karena sebagian besar sistem operasi tersebut merupakan bagian dari sistem terdistribusi.

Beberapa penelitian diarahkan guna pengembangan mekanisme yang menyediakan jaminan QoS pada level *middleware* yang terletak antara level aplikasi dan kernel sistem operasi. TAO yang berbasis *open-source* merupakan *middleware* yang menyediakan dukungan seperti *best-effort* bagi aplikasi (Schmidt, et al, 1998). Hesselman (2000) memfokuskan penelitian pada platform *middleware* untuk menetapkan *user-oriented QoS* pada data multimedia /*media streaming*. QoSME (*Quality of Service Management Environment*) dikembangkan dengan fitur untuk melakukan translasi persyaratan QoS aplikasi ke parameter QoS RSVP atau ATM di level jaringan (Wang, et al, 2000).

Pada level aplikasi, pencapaian QoS dilakukan melalui pengembangan berbagai bahasa spesifikasi QoS. QuO menyediakan QoS untuk perangkat lunak aplikasi terdistribusi yang tersusun atas objek (Pal, et al, 2001). QuO difokuskan pada spesifikasi, pengukuran dan pengontrolan aspek-aspek QoS suatu aplikasi serta menerapkan perilaku adaptif terhadap perubahan kondisi sistem. QML atau *QoS Modeling Language* (Frolund, 1998) merupakan pengembangan UML yang secara umum memungkinkan pendefinisian berbagai parameter QoS di berbagai lingkungan aplikasi. Asensio dan Villagra merekomendasikan suatu bahasa QoS yang mampu merepresentasikan seluruh informasi QoS ke level sistem. UML-Q merupakan perluasan dari UML (menggunakan UML profile) yang digunakan untuk

merepresentasikan informasi QoS yang berkaitan dengan sistem selama proses pengembangan sistem. Bahasa QoS yang didasarkan pada *framework* QoS ISO/ITU-T itu digunakan untuk melakukan spesifikasi QoS aplikasi berbasis objek terdistribusi. Campbell (1996) mengajukan arsitekur QoS disebut QoS-A yang menyediakan spesifikasi dan pencapaian properti kinerja yang diperlukan oleh aplikasi media kontinyu melalui jaringan ATM (*Asynchronous Transfer Mode*).

Beberapa bahasa spesifikasi tersebut dikembangkan guna mendukung spesifikasi kategori tertentu QoS seperti ketepatan waktu (*timeliness*) atau kehandalan (*reliability*), sedangkan bahasa spesifikasi lainnya memfokuskan pada spesifikasi umum QoS yang meliputi semua kategori QoS.

Kualitas layanan (QoS) merupakan salah satu area penelitian yang berkembang beberapa tahun terakhir. Secara tradisonal, istilah kualitas layanan lebih banyak mengarah pada karakteristik tertentu kinerja jaringan tanpa adanya pengaruh *user* (pengguna). Berdasarkan uraian di atas, berbagai perkembangan dan area yang menjadi fokus penelitian tentang kualitas layanan dirangkum dalam tabel berikut ini.

4. Kategori QoS

Kualitas layanan (QoS) dapat ditinjau dari beragam perspektif. Dari sisi sistem, QoS menyediakan mekanisme yang menjamin performans yang memadai dan diferensiasi layanan seperti minimal *delay* dan minimal *packet loss* untuk tipe aplikasi tertentu. Dari sisi pengguna, kepuasan pengguna menjadi dasar definisi QoS. Namun demikian, kualitas layanan berorientasi pengguna masih memiliki interpretasi yang beragam karena terdapat perbedaan yang cukup besar antara persepsi subyektif terhadap kualitas layanan antara pengguna yang satu dengan pengguna lainnya.

Untuk mengatasi perbedaan konsep QoS, *International Standards Organization* (ISO) mengajukan suatu kerangka kerja QoS sebagai standarisasi. OSI dan *International Telecommunication Union* (ITU) mengembangkan *Framework* QoS untuk penyediaan QoS melalui referensi arsitektur dan terminologi standar. *Framework* tersebut mencakup konsep, layanan dan mekanisme yang dapat diaplikasikan pada seluruh lapis OSI dan manajemen OSI.

Kerangka kerja QoS ISO mendefinisikan beberapa konsep dasar kualitas layanan, diantaranya adalah karakteristik QoS dan kategori QoS. Karakteristik QoS mewakili beberapa aspek QoS sistem, layanan maupun sumber daya yang dapat diidentifikasi dan dikuantisasi. *Time delay, throughput, availability* dan *precedence*

merupakan beberapa contoh karakteristik QoS. Karakteristik QoS berbeda dengan parameter QoS. Parameter QoS adalah suatu nilai kualitas layanan yang dilewatkan antar entitas. Kategori QoS merepresentasikan tipe persyaratan pengguna atau aplikasi. Tipe persyaratan yang berbeda dikelompokkan dalam kategori yang berbeda, sehingga membentuk himpunan karakteristik QoS. *ISO QoS Framework* mendefinisikan beberapa kategori dasar seperti *security*, *safety*, *timelines*, *capacity*, *integrity*, *coherence*, dan *reliability*.

Terkait dengan kategori QoS, ISO/IEC JTC1/SC21 (1998) mendefinisikan lima kategori QoS yang digunakan untuk menyatakan unsur pokok karakteristik QoS, yaitu:

- 1) *QoS Requirement*, menyatakan batasan-batasan kualitas layanan pengguna sistem atau komponennya yang dimiliki oleh sistem atau komponennya. Kategori ini dinyatakan melalui konsep Relasi QoS, yang akan dijelaskan pada bagian lain.
- 2) *QoS Capabilities*, menyatakan kemampuan aktual yang dimiliki komponen atau konfigurasi komponen dalam sistem yang berkaitan dengan kualitas layanan. *QoS Capabilities* dinyatakan melalui Relasi QoS.
- 3) *QoS Offers*, menyatakan kualitas layanan yang ditawarkan. Jika *QoS Capabilities* menyatakan kemampuan aktual suatu komponen, *QoS Offers* mengandung unsur ketidakpastian tentang kemampuan aktual suatu objek. Hal ini menyebabkan terjadinya ketidaksesuaian antara kemampuan aktual dengan kemampuan yang ditawarkan. *QoS Offers* juga menunjukkan bahwa seluruh kemampuan kualitas layanan dapat dipartisi sehingga hanya sebagian saja yang ditawarkan. *QoS Offers* dinyatakan melalui Relasi QoS.
- 4) *QoS Contracts*, merupakan hasil proses kesepakatan antar objek yang mendefinisikan himpunan kemampuan objek dalam suatu konfigurasi yang telah disepakati. Kategori ini dinyatakan melalui konsep Relasi QoS. *QoS Contracts* juga dapat dijalankan melalui mekanisme negosiasi yang dinamis.
- 5) *QoS Observation*, menyatakan nilai-nilai karakteristik kualitas layanan yang diobservasi melalui fungsi pengukuran.

5. Persepsi Pengguna

Persepsi pengguna merupakan salah satu faktor penting yang mempengaruhi ukuran kualitas layanan. Persepsi pengguna terhadap kualitas layanan dikaitkan

dengan ukuran subyektif kualitas layanan yang diterima, sedangkan persepsi kualitas layanan dari aspek teknologi didasarkan pada ukuran obyektif. Beberapa penelitian telah dilakukan terkait dengan aspek ukuran subyektif kualitas layanan. Bhatti et al. (2000) mengkaji efek delay terhadap persepsi kualitas layanan pengguna. Dalam hal ini, Bhatti et al. menyimpulkan bahwa toleransi pengguna terhadap delay dipengaruhi oleh faktor kontekstual (misalnya pola akses dan metode pengambilan halaman web) dan model konseptual pengguna terhadap cara kerja Internet. Ghinea et al. (1998) mengkaji efek *frame rate* terhadap persepsi kualitas pengguna. Dari penelitian ini Ghinea et al. menyimpulkan bahwa penurunan *frame rate* tidak mengurangi pemahaman dan persepsi pengguna terhadap content (El-Khatib, 2005).

Dari sisi pengguna, penentuan ukuran kualitas layanan aplikasi ditinjau dari persepsi pengguna terhadap *delay*, *jitter* dan *error* (Chen et al., 2003).

Persepsi Pengguna terhadap Delay

Persepsi pengguna terhadap *delay* memiliki pengaruh yang berbeda terhadap kualitas layanan untuk jenis aplikasi yang berbeda. Tabel 1 dan Tabel 2 menunjukkan persepsi pengguna terhadap *delay* pada aplikasi berbasis audio (Chen et al., 2003). Tabel 1 menunjukkan kaitan antara *delay* audio dengan pendengaran manusia yang diteliti oleh Fluckiger (1995). Tabel 2 menunjukkan batasan waktu transmisi satu arah yang distandarkan oleh ITU G.114 (1996).

Tabel 1. *Delay* Audio dan Pendengaran Manusia

<i>Delay</i> Audio (ms)	Pengaruh <i>delay</i> pada persepsi manusia terhadap suara
> 600	Suara tidak jelas dan sulit dimengerti
600	Suara masih sulit dipahami
250	Suara tidak jelas tapi masih dapat dimengerti
100	Audio dan suara asli hampir tidak dapat dibedakan
50	Tidak dapat dibedakan antara audio dengan suara asli

Tabel 2. Batasan waktu untuk Transmisi Satu Arah

Waktu transmisi satu arah (ms)	Pengaruh <i>delay</i> pada persepsi manusia terhadap suara
0 - 150	Dapat diterima oleh semua pengguna
150 - 400	Dapat diterima tapi masih terdapat gangguan
> 400	Tidak dapat diterima oleh pengguna

Persepsi Pengguna terhadap Jitter

Jitter merupakan parameter unjuk-kerja sistem yang digunakan untuk menyediakan kualitas layanan aplikasi berbasis suara dan gambar yang bersifat *real time*. Dibandingkan dengan semua tipe informasi, aplikasi suara yang bersifat *real time* merupakan jenis aplikasi yang paling sensitif terhadap *jitter*. Fluckiger (1995) meneliti tentang persepsi pengguna terhadap *jitter* (Tabel 3).

Tabel 3. Persepsi Pengguna terhadap *Jitter*

Jitter (ms)	Tipe Aplikasi
< 100	Suara dengan kualitas CD
400	Suara dengan kualitas telepon
20 - 30	Aplikasi multimedia
< 50	Aplikasi video dengan kualitas HDTV
100	Aplikasi video dengan kualitas <i>broadcast</i>
400	Aplikasi <i>videoconference</i>

Persepsi Pengguna terhadap Error

Pada konteks persepsi pengguna terhadap *error*, terdapat tiga hal yang harus dipertimbangkan agar persyaratan kualitas layanan aplikasi yang berkaitan dengan *error* masih dapat diterima oleh pengguna. Pertama, teknik transmisi ulang. Pada aplikasi berbasis teks, digunakan teknik transmisi-ulang untuk mengatasi *error* yang terjadi pada proses transmisi data. Namun teknik ini tidak sesuai jika diterapkan pada aplikasi audio dan video. Kedua, toleransi pengguna terhadap kesalahan transmisi. Pada dasarnya pada situasi tertentu, pengguna masih memiliki persepsi yang cukup baik terhadap kesalahan transmisi. Ketiga, aspek kompresi data. Tingkat *error* maupun *loss rate* pada data audio dan video terkompresi perlu dikontrol agar nilainya cukup rendah.

Tabel 4. Persepsi Pengguna terhadap *Error*

Error rate (ms)	Tipe Aplikasi
$< 10^{-3}$	Suara dengan kualitas CD tanpa kompresi
$< 10^{-4}$	Suara dengan kualitas CD terkompresi
$< 10^{-2}$	Suara dengan kualitas telepon
$< 10^6$	Aplikasi video dengan kualitas HDTV
10^5	Aplikasi video dengan kualitas <i>broadcast</i>
10^4	Aplikasi <i>videoconference</i>

B. Kerangka Pemikiran Konseptual

1. Web Browser Sebagai *User Agent*

Web browser dibuat dari keinginan untuk mengintegrasikan fungsi-fungsi akses jaringan yang sebelumnya terpisah ke dalam satu *interface* yang lebih sederhana. Diantaranya menampilkan *resource* yang dikodekan menggunakan format HTML, mendukung beberapa mekanisme *scripting* bahkan mendukung untuk *browser plug-ins* (*browser Plug-ins* adalah program tambahan yang memungkinkan untuk memainkan audio atau video)

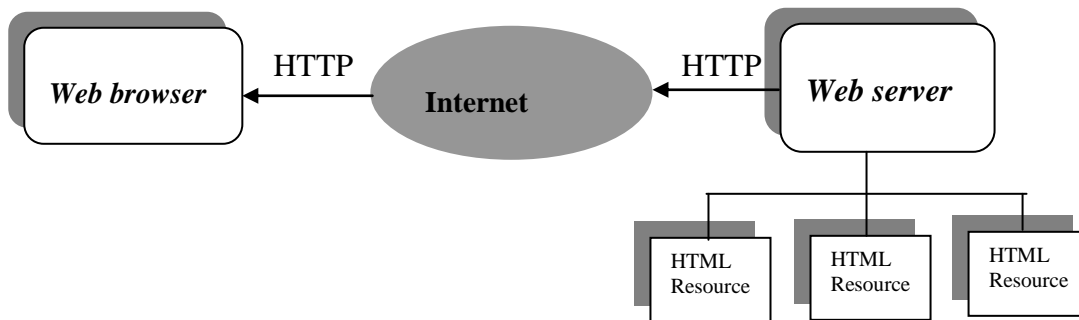
Web Browser adalah sebuah program aplikasi yang digunakan untuk mengakses situs-situs Web (*World Wide Web, WWW*). *Web browser* berperan sebagai media perantara antara pengguna dengan protokol pada tingkat aplikasi yang disebut *Hypertext Transfer Protokol (HTTP)*. *Web browser* bertindak sebagai agen antarmuka (*interface*) yang mewakili pengguna agar mudah dalam berkomunikasi memanfaatkan sintaks, metode dan informasi pada protokol HTTP. Pengguna hanya cukup menentukan suatu alamat situs web dalam format tertentu, maka *web browser* akan meneruskan aksesnya dalam bentuk permintaan (*request*) sesuai alamat tersebut dan menangkap tanggapan (*response*) yang diberikan oleh *Web server*.

Request dibuat menggunakan sebuah struktur penamaan yang telah disepakati, yang disebut dengan *Universal Resource Identifier (URI)*. Setiap *resource* yang tersedia pada jaringan Internet (dokumen HTML, gambar, video clip, program dan sebagainya)- mempunyai alamat unik yang telah kodekan dengan sebuah URI. Jenis skema penamaan URI yang sering digunakan adalah *Uniform Resource Locator (URL)* yang secara umum terdiri dari tiga bagian informasi, yaitu:

- Skema penamaan dari protokol atau mekanisme yang digunakan untuk mengakses *resource*.
- Nama dari sistem *hosting resource*
- Nama dari *resource* itu sendiri, dengan lokasi absolutnya pada sistem *host*.

Komunikasi antara *user* yang menggunakan *web browser* dan suatu situs Web mengikuti model *client-server* pada umumnya. Dalam hal ini *web browser* yang bertindak sebagai *client* membuat permintaan (*request*) untuk halaman web atau suatu *resource* dari suatu *web server*. Selanjutnya *server* dalam hal ini adalah *web server* membaca *request* dan mengidentifikasinya apakah harus memenuhi *request* tersebut dengan mengirim suatu obyek tertentu (*response*). Semua proses pertukaran informasi

antara *web browser* dan *server* terjadi pada suatu jaringan (*network*) yang umumnya adalah jaringan berskala besar (Internet).



Gambar 1. Arsitektur umum *Browser* dan *Web Server* (Friedrichs,1999)

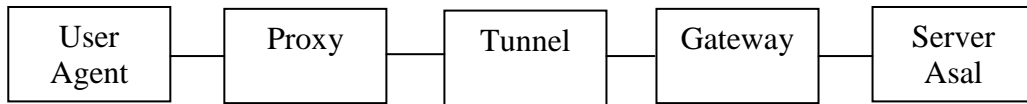
Mekanisme komunikasi WWW antara *client* (*web browser*) dan *server* (*web server*) ditangani oleh protokol HTTP. HTTP adalah protokol pada tingkat aplikasi untuk sistem informasi terdistribusi, kolaboratif, hypermedia. Protokol ini dikembangkan sebagai sebuah standar oleh suatu badan, yaitu : *Internet Engineering Task Force* (IETF) dari konsorsium yang bernama W3C dalam bentuk *Request for Comment* (RFC). Versi terbaru (saat penelitian ini dilaksanakan) yang telah direkomendasi sebagai standar adalah HTTP/1.1 yang tercantum dalam RFC2616.

2. Hypertext Transfer Protocol (HTTP)

a. Model hubungan HTTP

Request dan *response* dalam protokol HTTP disebut sebagai *request chain* dan *response chain*. Hubungan HTTP yang paling sederhana terdiri atas hubungan langsung antara *user agent* (*web browser*) dengan *server* asal.

Pada protokol HTTP terdapat tiga jenis hubungan dengan perantara : *proxy*, *gateway*, dan *tunnel*. *Proxy* bertindak sebagai agen penerus, menerima *request* dalam bentuk *Uniform Resource Identifier* (URI) absolut, mengubah format *request* dan mengirimkan *request* ke *server* yang ditunjukkan oleh URI. *Gateway* bertindak sebagai agen penerima dan menerjemahkan *request* ke protokol *server* yang dilayaninya. *Tunnel* bertindak sebagai titik *relay* antara dua hubungan HTTP tanpa mengubah *request* dan *response* HTTP. *Tunnel* digunakan jika komunikasi perlu melalui sebuah perantara dan perantara tersebut tidak mengetahui isi dari pesan dalam hubungan tersebut.



Gambar 2. Komponen dalam rantai request/response HTTP (Purbo,1998)

b. Format *Request*

Pesan *request* dari *client* ke *server* secara umum sebagai berikut :

```

Request = Method Request-URI HTTP-Version
        ( general-headers
          | request-headers
          | entity-headers)

[entity body]
  
```

Method menunjukkan metode apa yang hendak dilakukan pada *resource* yang ditunjuk oleh *Request-URI*. Ada beberapa metode yang didefinisikan oleh HTTP/1.1., Metode GET mengambil informasi apa saja dalam bentuk *entity* yang diidentifikasi oleh *Request-URI*. Metode HEAD sama dengan metode GET kecuali bahwa *server* harus tidak mengirimkan *entity* yang ditunjukkan oleh *Request-URI*. Metode POST digunakan untuk meminta *server* menempatkan *entity* yang dikirim bersama *request* sebagai subordinat dari *request-URI* yang dituju. Metode ini biasa digunakan dalam mengirimkan *form*. Metode PUT meminta *server* untuk menempatkan *entity request* sebagai *Request-URI*. Perbedaan anatar POST dan PUT adalah *Request-URI* dalam metode POST bertindak sebagai proses penerima data atau sebagai gateway, sementara dalam metode PUT, *Request-URI* adalah *entity* yang terdapat dalam *request*. Metode DELETE meminta *server* untuk menghapus URI yang dikirim *client*. *Client* tidak dapat menjamin *server* berhasil melaksanakan *request* yang diminta. Metode TRACE digunakan untuk meminta *loop-back* pesan. Metode TRACE ini dapat digunakan untuk diagnostik.

Request-URI menunjukkan *resource* yang hendak diakses melalui pesan *request*. *Request-URI* dapat berupa URI absolut, *path* absolut, atau tanda asteriks (*). *Request-URI* tergantung pada request itu sendiri. Tanda asteriks "*" berarti bahwa *request* tidak merujuk pada *resorce* tertentu, melainkan pada *server*. *Request-URI* asteriks

hanya boleh digunakan jika metode yang digunakan tidak harus merujuk pada sebuah *resource*. Contoh : `OPTIONS * HTTP/1.1`

Entity body merupakan data tambahan yang mungkin dikirimkan tergantung pada metode *request* yang digunakan. Metode *request* Post menempatkan data yang dibutuhkan pada bagian *entity body*.

c. Format *Response*

Pesan *response* dari *server* ke *client* setelah menerima *request* secara umum sebagai berikut :

```
Response= HTTP-Version Status-Code Status-Description
          ( general-headers
            | response-headers
            | entity-headers)

          [entity body]
```

Baris status berisi kode-status yang berupa kode tiga digit dan frasa-alasan, yaitu penjelasan singkat atas kode-status tersebut. Digit pertama kode-status menentukan kelas *response*. Protokol HTTP/1.1 mendefinisikan 5 nilai untuk digit pertama :

- 1xx : *Informational* – request diterima, dan proses berlanjut
- 2xx : *Success*- request diterima dan dimengerti
- 3xx : *Redirection* – request membutuhkan tindakan lebih lanjut.
- 4xx : *Client Error* – request mengandung sintaks yang salah
- 5xx : *Server Error* – server gagal melakukan tindakan sesuai request

Entity body pada pesan *response* merupakan dokumen HTML yang dibangkitkan dari *server* sesuai *request* dari *client*.

d. HTTP Header

HTTP Header terdiri atas *general header*, *request header*, *response header* dan *entity header*. *Header* pesan dapat terdiri atas beberapa baris, bergantung pada *field-field* yang perlu disertakan dalam *header* tersebut. *Request header* berkaitan dengan informasi permintaan maupun berkaitan dengan *client* itu sendiri. *Response Header* berguna memberikan informasi mengenai *server* serta mengenai akses lebih lanjut ke

URI. *General header* dipergunakan pada pesan *request* maupun pesan *response* yang berkaitan dengan informasi-informasi umum. Setiap pesan HTTP baik *request* maupun *response* dapat menyertakan isi pesan atau *entity* tergantung dari apakah pesan tersebut memungkinkan untuk membawa *entity*. *Entity* HTTP terdiri atas *header entity* dan *isi entity*. *Header entity* berisi informasi mengenai isi *entity* atau mengenai *resource* yang ditunjuk oleh *Request-URI*.

General header pada pesan HTTP *request* dan *response* adalah sebagai berikut :

```
General-header = Cache-Control
                | Connection
                | Date
                | Pragma
                | Trailer
                | Transfer-Encoding
                | Upgrade
                | Via
                | Warning
```

Field Cache-control memberikan aturan yang harus ditaati oleh seluruh mekanisme *chace* dalam rantai *request-response*.

Field Connection mengatur tipe hubungan HTTP, apakah akan menggunakan hubungan persistent atau tidak.

Field Date memberikan informasi mengenai waktu asal pesan.

Field Transfer-Encoding menentukan jenis transfer yang diberikan kepada isi pesan agar dapat sampai dengan aman ke *client*.

Field Upgrade Pada pesan HTTP digunakan untuk mengganti protokol yang hendak digunakan, field ini berguna sebagai mekanisme transisi dari protokol HTTP/1.1 ke protokol tipe lain.

Field Via Digunakan oleh proxy dan gateway untuk memberitahu jalur yang digunakan dalam sebuah rantai *request/response*.

Field Pragma Untuk memuat arahan (*directive*) implementasi tertentu yang mungkin diterapkan untuk penerima sepanjang rantai *request/response*.

- Field Trailer* Berisi nilai yang menunjukkan bahwa *field header* berada dalam sebuah trailer dari sebuah pesan yang disandikan dengan gumpalan kode pengiriman.
- Field Warning* Membawa informasi tambahan mengenai status atau perubahan bentuk dari sebuah pesan yang boleh di refleksikan dalam pesan tersebut.

Setelah baris-*request*, *client* mengirimkan *request header* yang berisi informasi mengenai *request* atau mengenai *client* itu sendiri ke *server*. *Header-header* tersebut adalah:

```
Header-request = Accept
                | Accept-Charset
                | Accept-Encoding
                | Accept-Language
                | From
                | Host
                | If-Modified-Since
                | If-Match
                | If-None-Match
                | If-Range
                | If-Unmodified-Since
                | Max-Forwards
                | Proxy-Authorization
                | Range
                | Referensi
                | User-Agent
```

- Accept* Menentukan tipe media yang dapat diterima sebagai *response* dari *server*
- Accept-Charset* Mengindikasikan set karakter yang dapat diterima sebagai *response*.
- Accept-Language* Memperkecil jenis bahasa yang lebih disukai untuk diterima oleh *client*.
- Host* Menentukan *host* dan *port* tempat *resource* hendak diambil. *Client* HTTP/1.1 harus menyertakan field ini dalam setiap *request*.

If-Modified-Since Digunakan bersama metode GET memberikan kondisi bahwa jika resource yang hendak diakses belum diubah sejak waktu yang ditentukan oleh *field* ini, maka *resource* tersebut tidak akan dikirim ke *server*, melainkan *server* memberikan *response* 304 (*not modified*) tanpa isi pesan.

User-Agent Berisi informasi mengenai *user agent* yang melakukan *request* ke *server*.

Setelah baris *response*, *server* HTTP mengirimkan *header-header response* ke *client*, *Header* ini memberikan informasi mengenai *server* serta mengenai akses lebih lanjut ke URI.

```
Header-response = Age
                  | Location
                  | Proxy-Authenticate
                  | Public
                  | Retry-After
                  | Server
                  | Vary
                  | Warning
                  | WWW-Authenticate
```

Age Perkiraan lama waktu sejak *response* atas URI diambil dari *server* asal. *Header* ini harus digunakan oleh *Chace* HTTP/1.1

Location Digunakan untuk mengarahkan *client* ke URL lain

Proxy-Authenticat Field ini memungkinkan *client* untuk mengirmkan identitasnya untuk menggunakan *proxy* yang membutuhakn autentikasi.

Public Memperlihatkan jenis metode yang didukung oleh *server*.

Retry-After Dapat digunakan bersama kode-status 503 untuk menunjukkan berapa lama layanan belum dapat diberikan.

Server Berisi informasi mengenai *server* yang menangani *request*.

Vary Digunakan *server* untuk mengirimkan sinyal bahwa *entity* yang dikirim adalah hasil negosiasi yang '*server driven*'. URI dengan *header* ini harus tidak *di-cache*.

Warning Digunakan untuk memberi informasi tambahan selain dari kode status.

WWW-Authenticate Dikirim beserta kode-status 401. Terdiri atas setidaknya satu *challenge* yang menentukan skema autentikasi dan parameter-parameternya.

Setiap pesan HTTP baik *request* maupun *response* dapat menyertakan isi pesan atau *entity* tergantung dari apakah pesan tersebut memungkinkan untuk membawa *entity*. *Entity* HTTP terdiri atas *header entity* dan isi *entity*. *Header entity* berisi informasi mengenai isi *entity* atau mengenai *resource* yang ditunjuk oleh *Request-URI*.

```
Header-entity = Allow
                | Content-Base
                | Content-Encoding
                | Content-Language
                | Content-Length
                | Content-Location
                | Content-MD5
                | Content-Range
                | Content-Type
                | Etag
                | Expires
                | Last-Modified
                | extension-header
```

Allow Menunjukkan metode yang didukung oleh resource yang ditunjuk oleh Request URI.

Content-Base Menentukan base URI yang digunakan untuk mengartikan URI relatif.

Content-Encoding Menentukan tipe media dan mekanisme decoding yang harus diberikan pada entity.

Content-Language Menentukan bahasa yang hendak digunakan.

Content-Length Menentukan panjang isi pesan dalam oktet.

Content-Location Menginformasikan lokasi resource dari isi pesan.

Content-MD5 Untuk memeriksa integritas isi entity

Content-Range Digunakan untuk mengirimkan sebagian dari seluruh entity.

<i>Content-Type</i>	Menentukan tipe media isi entity yang dikirim ke penerima.
<i>Etag</i>	Menentukan tag untuk entity yang bersangkutan. Digunakan untuk GET bersyarat.
<i>Expires</i>	Mengindikasikan waktu kadaluarsa entity. Cache harus mengambil ke server asal jika waktu kadaluarsa entity sudah terlewati.
<i>Last-Modified</i>	Mengindikasikan waktu terakhir perubahan entity.

BAB III

METODOLOGI PENELITIAN

A. Materi Penelitian

Fokus utama penelitian ini adalah pengembangan kerangka kerja model akses terintegrasi untuk kualitas layanan akses Internet pada jaringan yang kurang handal. Kerangka kerja yang dikembangkan terdiri atas spesifikasi persyaratan kualitas layanan pengguna dan arsitektur QoS untuk mengimplementasikan mekanisme QoS yang memenuhi kualitas layanan pengguna. Mekanisme QoS memuat fungsi-fungsi spesifikasi persyaratan kualitas layanan dan akses pengguna, *mapping* kualitas layanan dan manajemen kualitas layanan.

Untuk mendukung pengembangan kerangka kerja, materi penelitian mencakup tiga hal berikut:

1. Pengembangan konsep QoS dan persyaratannya guna menemukan kerangka dasar bagi model akses terintegrasi. Pada penelitian ini konsep QoS yang pada awalnya lebih banyak ditinjau dari sisi sistem, dikembangkan ke konsep QoS berorientasi pengguna. Kuantisasi QoS pada konsep ini menggunakan parameter subyektif yang berkaitan dengan persepsi dan ekspektasi pengguna terhadap kualitas layanan.
2. Penemuan model akses terintegrasi yang dapat menunjukkan relasi antara *user-oriented QoS* dengan QoS pada lapisan yang lebih rendah. QoS dari sudut pandang pengguna dinyatakan dalam parameter yang mengarah pada tingkat layanan yang dirasakan oleh pengguna. Sementara dari sisi sistem, QoS menyatakan tingkat kualitas yang disediakan oleh sistem kepada pengguna. Tingkat kualitas ini dinyatakan melalui nilai-nilai yang diberikan pada parameter QoS. Setiap layanan harus memiliki himpunan parameter QoS. Perbedaan spesifikasi QoS ini perlu diintegrasikan sehingga relasi antar parameter terdefinisi dengan jelas.
3. Pengembangan kerangka kerja yang mengimplementasikan model akses terintegrasi guna meningkatkan aspek ketergunaan melalui fleksibilitas layanan bagi pengguna dalam mengakses Internet pada jaringan dengan kualitas koneksi yang rendah. Kerangka kerja yang dimaksud merupakan susunan terstruktur dari

konsep dan relasinya yang menggambarkan QoS dan partisinya, relasi antara QoS pengguna dengan QoS aplikasi dan aspek-aspek lain yang berkaitan dengan QoS yang dinyatakan dalam suatu deskripsi. Untuk itu kerangka kerja harus memuat fungsi-fungsi untuk spesifikasi kualitas layanan pengguna, *mapping* parameter kualitas layanan pengguna ke parameter aplikasi dan mekanisme penetapan pola akses berorientasi pengguna.

B. Cara Penelitian

Tahapan yang ditempuh dalam penelitian ini mencakup langkah-langkah sebagai berikut:

1. Identifikasi Persyaratan *User-Oriented QoS*

Identifikasi persyaratan kualitas layanan pengguna dilakukan untuk mengembangkan model konseptual *User-Oriented QoS*. Identifikasi dilakukan melalui pendefinisian komponen-komponen yang digunakan untuk mengembangkan model *User-Oriented QoS*. Komponen pemodelan meliputi parameter-parameter yang dijadikan ukuran untuk kuantisasi kualitas layanan dari sisi pengguna, spesifikasi persyaratan QoS pengguna melalui perilaku akses dan mekanisme pencapaian QoS.

2. Pengembangan Model Konseptual *User-Oriented QoS*

Hasil identifikasi *user-oriented QoS* berupa parameter kualitas layanan dan spesifikasi perilaku pengguna dalam akses Internet, direpresentasikan menggunakan model berbasis keadaan (*state-based model*). Model ini akan menunjukkan perubahan keadaan (*state*) suatu akses secara sekuensial untuk setiap aksi yang dilakukan oleh pengguna. Aksi yang dilakukan pengguna didasarkan pada spesifikasi parameter yang digunakannya. Jika spesifikasi parameter (persyaratan QoS) terpenuhi, akan menghasilkan keadaan (*state*) akses yang diharapkan. Sebaliknya jika persyaratan QoS tidak terpenuhi keadaan akses akan berada pada keadaan yang lain.

Spesifikasil *User-Oriented QoS* dalam bentuk *state-based model* kemudian dikembangkan menjadi notasi dalam bentuk spesifikasi formal. Spesifikasi formal menyediakan deskripsi perilaku sistem yang memungkinkan untuk menyatakan aksi (*action*) dan perubahan suatu kondisi (*state transition*) menggunakan kombinasi notasi matematis dan pemrograman secara ringkas, jelas dan mudah

dipahami. Spesifikasi formal ini nantinya menjadi dasar pengembangan bahasa spesifikasi QoS.

3. Kerangka Kerja Model Akses Terintegrasi

Kerangka kerja model akses terintegrasi berorientasi pengguna menyediakan fungsi-fungsi yang diperlukan untuk mengimplementasikan model akses pengguna, seperti fungsi untuk spesifikasi QoS pengguna, mekanisme pola akses dinamis dan *Mapping QoS*.

Pengembangan kerangka kerja dilakukan melalui tahapan berikut:

1. Membuat desain arsitektur kerangka kerja yang menyediakan komponen-komponen fungsional untuk mendukung pengembangan model akses berorientasi pengguna.
2. Mengembangkan desain arsitektur kerangka kerja ke dalam model perangkat-lunak. Desain perangkat-lunak menggunakan *Enterprise Architecture* sebagai perangkat bantu (*tools*) pemodelan.
3. Pada tahap selanjutnya, komponen-komponen fungsional pada model perangkat-lunak diuji coba untuk memvalidasi dan mengevaluasi kerangka kerja yang diusulkan.

BAB IV

HASIL DAN PEMBAHASAN

A. Persyaratan *User-Oriented QoS*

Tahap analisis persyaratan kualitas layanan pengguna dilakukan untuk mengidentifikasi karakteristik, parameter-parameter dan domain nilai untuk parameter kualitas layanan pengguna. Pada penelitian ini, analisis terhadap persyaratan kualitas layanan pengguna menghasilkan 3 karakteristik kualitas layanan yang direpresentasikan melalui parameter kualitas layanan. Parameter kualitas layanan selanjutnya digunakan sebagai dasar pemodelan akses.

1. Parameter *User-Oriented QoS*

Pemberian nilai ke parameter kualitas layanan subyektif dilakukan oleh pengguna ketika melakukan akses Internet. Spesifikasi nilai parameter subyektif ini merepresentasikan tingkat kualitas layanan yang diharapkan pengguna. Model QoS subyektif pengguna dapat diidentifikasi melalui aspek-aspek QoS yang paling sering digunakan oleh pengguna sebagai ukuran kualitas layanan. Aplikasi *web browsing* misalnya, pengguna lebih memfokuskan pada aspek waktu akses dan *content* atau informasi yang diambil. Demikian juga untuk aplikasi *video streaming*, pengguna lebih memfokuskan pada aspek *image quality* maupun *audio quality*. Untuk itu, pemilihan parameter subyektif pada penelitian ini ditinjau melalui pendekatan persepsi (penilaian yang dinyatakan oleh pengguna terhadap layanan yang diterima), ekspektasi (nilai harapan yang dinyatakan pengguna pada layanan yang disediakan oleh sistem) dan performans (standar teknis suatu sistem yang menyatakan kehandalan sistem secara keseluruhan untuk menyediakan layanan yang dibutuhkan).

Jika ekspektasi pengguna lebih besar dari kualitas layanan yang diterima, maka nilai persepsi kurang dari satu ($P < 1$) dan tidak memenuhi persyaratan kualitas layanan pengguna. Jika ekspektasi pengguna sebanding dengan kualitas layanan yang diterima, maka nilai persepsi netral (relasi 1 : 1) dan masih memenuhi persyaratan kualitas layanan pengguna. Jika ekspektasi pengguna lebih kecil dari kualitas layanan yang diterima, nilai persepsi lebih dari satu ($P > 1$) dan memenuhi persyaratan kualitas layanan pengguna.

Berdasarkan evaluasi tersebut, konsep *User-oriented QoS* menekankan pada aspek pemenuhan persyaratan kualitas layanan yang didasarkan pada nilai subyektif

pengguna. Penelitian ini mendefinisikan beberapa parameter *User-oriented QoS* yang merepresentasikan kualitas layanan yang diharapkan pengguna, yaitu parameter yang berkaitan dengan aspek waktu akses, parameter yang berkaitan dengan aspek keberhasilan suatu permintaan atau akses dan parameter yang berkaitan dengan kesesuaian isi/informasi yang dihasilkan dari suatu permintaan atau akses .

Karakteristik Waktu (t)

Kecepatan merupakan atribut yang berkaitan dengan aspek temporal. Aspek temporal menyatakan ukuran waktu yang diperlukan untuk memperoleh *output* suatu proses. Dari atribut kecepatan, didefinisikan karakteristik waktu (**t**) untuk mengekspresikan persyaratan kualitas layanan pengguna. Karakteristik waktu dari sisi pengguna dispesifikasi melalui parameter *access_time* yang menyatakan batasan waktu tunggu yang masih dapat ditolerir oleh pengguna untuk menerima respons atas akses yang dilakukan.

Domain nilai untuk karakteristik waktu dinyatakan dalam bilangan numerik maupun domain yang bersifat kontinyu. Misalnya dalam melakukan akses *browsing*, pengguna dapat menerima QoS dengan *access_time* kurang dari 2 detik tetapi tidak dapat bertoleransi untuk *time_limit* yang lebih dari 15 detik.

Karakteristik Keberhasilan (s)

Kehandalan merupakan atribut yang mencirikan tingkat kepastian bahwa suatu fungsi telah dilakukan. Suatu sistem dikatakan handal jika sistem mampu menjalankan fungsi-fungsi yang dibentuk tanpa kegagalan dalam lingkungan tertentu pada suatu periode waktu yang ditetapkan. Kehandalan merupakan atribut yang bersifat sistemik, karena kehandalan sistem tergantung pada semua elemen yang berada dalam sistem tersebut (He, 2004). Berdasarkan atribut kehandalan, didefinisikan karakteristik keberhasilan (**s**) untuk mengekspresikan persyaratan kualitas layanan pengguna. Keberhasilan merupakan karakteristik yang ditinjau dari aspek derajat kepastian terpenuhinya suatu layanan. Karena itu domain nilai untuk karakteristik keberhasilan adalah 1 untuk *true* dan 0 untuk *false*.

Karakteristik keberhasilan dari sisi pengguna dispesifikasi melalui parameter *retry_amount*. Parameter ini menyatakan batasan perulangan proses selama akses yang dispesifikasi oleh pengguna masih gagal dipenuhi oleh sistem (karakteristik keberhasilan masih bernilai *false*). Misalnya dalam akses *browsing*, pengguna dapat

menyatakan parameter *retry_amount* = 3 selama akses yang dispesifikasi oleh pengguna masih gagal dieksekusi oleh sistem. Ini berarti bahwa jika akses *browsing* gagal dieksekusi, sistem mencoba untuk mengeksekusi-ulang permintaan akses pengguna dalam batasan yang telah didefinisikan pengguna.

Karakteristik Kesesuaian Isi (c)

Ketepatan merupakan atribut yang mencirikan tingkat kebenaran dari fungsi yang dilaksanakan. Berdasar atribut ini didefinisikan karakteristik kesesuaian isi (c), yaitu karakteristik yang berkaitan dengan ukuran ketepatan atau tingkat kesesuaian antara isi/*content* suatu *output* dengan fungsi atau *service* (layanan) yang dibentuk.

Karakteristik kesesuaian isi dispesifikasi melalui parameter-parameter *conten_match*, *image_quality*, *audio_quality* sesuai media akses yang digunakan untuk melakukan akses. Domain nilai untuk karakteristik kesesuaian isi bersifat diskrit, sehingga parameter *image_quality* dapat dinyatakan dengan nilai baik, sedang atau rendah.

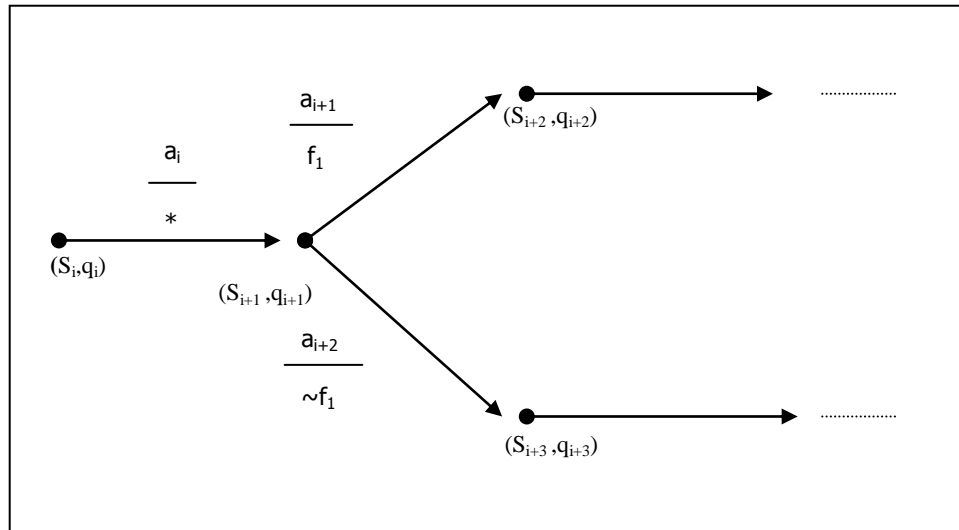
Karakteristik dan parameter *user-oriented QoS* yang didefinisiakan dalam penelitian ini dirangkum dalam Tabel 5.

Tabel 5. Karakteristik dan parameter QoS

Karakteristik	Deskripsi	Parameter
Waktu (t)	<ul style="list-style-type: none"> ▪ Karakteristik kualitas layanan dari aspek temporal ▪ Pengukuran : waktu yang diperlukan mulai proses request (<i>input</i>) sampai diterimanya response (<i>output</i>) 	<ul style="list-style-type: none"> ▪ <i>access_time</i>
Keberhasilan (s)	<ul style="list-style-type: none"> ▪ Karakteristik kualitas layanan dari aspek tingkat / derajat kepastian bahwa suatu fungsi (<i>service</i>) dapat dibentuk ▪ Pengukuran : probabilitas dari hasil eksekusi fungsi (<i>service</i>) yang dihasilkan 	<ul style="list-style-type: none"> ▪ <i>retry_amount</i>
Kesesuaian Isi (c)	<ul style="list-style-type: none"> ▪ Karakteristik kualitas layanan dari aspek tingkat kesesuaian hasil terhadap fungsi (<i>service</i>) yang dibentuk ▪ Pengukuran : kesesuaian isi / validitas data/informasi yang dihasilkan oleh fungsi (<i>service</i>) yang diproses 	<ul style="list-style-type: none"> ▪ <i>Content_match</i>, <i>media_quality</i>

2. Model Konseptual *User-oriented QoS*

Pendefinisian parameter kualitas layanan menjadi dasar perumusan model untuk melakukan spesifikasi persyaratan kualitas layanan. Untuk merepresentasikan spesifikasi kualitas layanan pengguna dalam melakukan akses Internet, digunakan model berbasis keadaan (*state-based model*).



Gambar 3. Diagram Model Akses Terintegrasi

Gambar 3 menunjukkan model konseptual spesifikasi akses pengguna. Jika S_i menyatakan keadaan awal (*initial-state*), S_{i+1} mewakili keadaan (*state*) setelah dilakukan aksi, a mewakili aksi yang mengubah satu *state* ke *state* berikutnya, kualitas layanan (q) yang dikarakteristik oleh parameter t , s dan c dan suatu situasi kondisional f diperhitungkan untuk suatu keadaan (*state*), maka suatu aksi a_i dengan kondisi f bernilai *true* (memenuhi persyaratan), akan menghasilkan keadaan (*state*) tertentu, sebaliknya aksi a_i dengan kondisi f bernilai *false* (tidak memenuhi persyaratan) akan menghasilkan keadaan (*state*) yang lain. Keadaan (*state*) yang lain ini bisa berupa akses alternatif yang didefinisikan pengguna. Gambar 4 merepresentasikan situasi kondisional ini.

Diagram keadaan untuk model akses yang diajukan kemudian dinyatakan melalui suatu skema yang terdiri atas 3 komponen, yaitu:

- 1 **action:** $(a_i, [q_{exp}])$ mewakili deskripsi aksi yang dilakukan oleh pengguna.

Aksi a_i akan memicu perubahan dari satu *state* ke *state* berikutnya.

a_i mewakili jenis akses yang dilakukan pengguna, berupa pernyataan-pernyataan (*primitive*) seperti *initial()*, *find(*, source)*, *send(*, dest)*, *receive(*, source)*, *cancel()* dan *loop(n)*.

q_{exp} mewakili spesifikasi kualitas layanan pengguna (t, s, c)

2 **precondition:** (S_n, f) mewakili *state* sebelum dilakukan aksi a_i .

S_n (dengan $n < i$) mewakili *state* sebelum dilakukan aksi a_i .

f mewakili kondisi Boolean yang menyebabkan aksi a_i dilaksanakan.

3 **postcondition:** $(f \rightarrow S_t \vee \neg f \rightarrow S_f)$ mewakili *state* setelah aksi a_i

S_t mewakili *state* yang dituju jika kondisi f bernilai *true*.

S_f mewakili *state* yang dituju jika kondisi f bernilai *false*.

f adalah kondisi Boolean yang nilainya didefinisikan :

$$f:: \begin{cases} true & \text{jika } q_{exp} \text{ terpenuhi} \\ false & \text{jika } q_{exp} \text{ tidak terpenuhi} \end{cases}$$

Dua konsep penting yang termuat dalam spesifikasi model akses terintegrasi ini adalah operasi (*operation*) dan keadaan (*state*). Operasi menyatakan aksi yang terjadi pada suatu keadaan dan kondisi yang memungkinkan perubahan dari keadaan satu ke keadaan lainnya (*transition*). Operasi dikaitkan dengan dua kondisi: *precondition* yang mendefinisikan lingkungan tempat operasi khusus dapat terjadi dan *postcondition* suatu operasi untuk menentukan apa yang terjadi pada saat suatu operasi telah menyelesaikan aksinya.

Skema operator untuk operasi suatu akses dapat dinyatakan dengan:

$$\text{Op: } \{ \text{precondition: } (S_n, f) \mid (\text{action: } (a_i, [q_{exp}]) \mid \text{postcondition: } (f \rightarrow S_t \vee \neg f \rightarrow S_f) \}$$

Format diatas dapat disederhanakan menjadi :

$$\text{Op: } \{ (S_n, f) \mid (a_i, [q_{exp}]) \mid (f \rightarrow S_t \vee \neg f \rightarrow S_f) \}$$

State digunakan untuk mendefinisikan properti kualitas layanan. Dalam konteks ini, atribut dinyatakan dengan q (parameter kualitas layanan). Suatu keadaan dapat mengalami transisi dari satu keadaan ke keadaan lainnya tergantung pada operasi yang dikenakan padanya. Keadaan (*state*) suatu akses dan operasi yang dikenakan terhadap *state* tersebut dapat dinyatakan :

$$S_i : \{ (S_n, f) \mid (a_i, [q_{exp}]) \mid (f \rightarrow S_t \vee \neg f \rightarrow S_f) \}$$

Pada suatu *state* S_i , *precondition* (S_n, f) merupakan persyaratan (*guard*) untuk terjadinya *action* $(a_i, [q_{exp}])$ dan *postcondition* $(f \rightarrow S_t \vee \neg f \rightarrow S_f)$ menunjukkan

state selanjutnya yang akan terjadi setelah aksi selesai dilaksanakan. Ekspresi pada *action* ($\mathbf{a}_i, [\mathbf{q}_{exp}]$) dievaluasi dari arah kiri ke kanan, artinya parameter \mathbf{q}_{exp} dievaluasi setelah aksi \mathbf{a}_i selesai dilaksanakan. Demikian juga untuk parameter-parameter \mathbf{q}_{exp} . *Initial state* dan *end state* dinyatakan melalui *precondition* dan *postcondition* bernilai Nil.

Dari skema yang telah didefinisikan, *state transition* untuk suatu akses dapat mengikuti pola yang paling sederhana sebagai berikut:

$$\begin{aligned} S_0: & \{ (\text{nil}) \mid (\text{initial}()) \mid (S_1, \mathbf{q}_{exp}) \} \\ S_1: & \{ (S_0, \text{initial}()) \mid (\mathbf{a}_1, [\mathbf{q}_{exp}]) \mid (f_1 \rightarrow S_2 \vee \neg f_1 \rightarrow S_3) \} \\ S_2: & \{ (S_1, f_1) \mid (\mathbf{a}_2, [\mathbf{q}_{exp}]) \mid (\text{nil}) \} \\ S_3: & \{ (S_1, \neg f_1) \mid (\mathbf{a}_3, [\mathbf{q}_{exp}]) \mid (\text{nil}) \} \end{aligned}$$

Pola *state transition* di atas dapat dijelaskan sebagai berikut. S_0 menunjukkan keadaan awal (*initial state*) yang ditandai dengan komponen *precondition* yang bernilai Nil. Pada keadaan S_1 pengguna menetapkan akses yang ingin dilakukan beserta spesifikasi kualitas layanan yang diharapkan dan *postcondition* yang menunjukkan *state* yang dimungkinkan setelah aksi ($\mathbf{a}_i, [\mathbf{q}_{exp}]$) dijalankan. S_2 mewakili keadaan apabila kondisi yang ditetapkan pada S_1 terpenuhi (akses utama), sedangkan S_3 mewakili keadaan bila kondisi yang disyaratkan pada S_1 tidak terpenuhi (akses alternatif).

Untuk pola akses yang lain, *state transition* dapat dikembangkan menjadi :

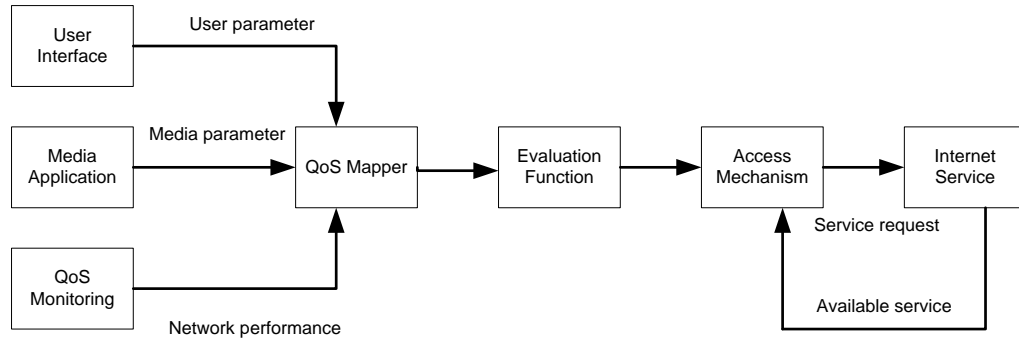
$$\begin{aligned} S_0: & \{ (\text{nil}) \mid (\text{initial}()) \mid (S_1, \mathbf{q}_{real}) \} \\ S_1: & \{ (S_0, \text{initial}()) \mid (\mathbf{a}_1, [\mathbf{q}_{exp}]) \mid (f_1 \rightarrow S_2 \vee \neg f_1 \rightarrow S_3) \} \\ S_2: & \{ (S_1, f_1) \mid (\mathbf{a}_2, [\mathbf{q}_{exp}]) \mid (\text{nil}) \} \\ S_3: & \{ (S_1, \neg f_1) \mid (\mathbf{a}_3, [\mathbf{q}_{exp}]) \mid (f_2 \rightarrow S_4 \vee \neg f_2 \rightarrow S_5) \} \\ S_4: & \{ (S_2, f_2) \mid (\mathbf{a}_4, [\mathbf{q}_{exp}]) \mid (\text{nil}) \} \\ S_5: & \{ (S_3, \neg f_2) \mid (\mathbf{a}_5, [\mathbf{q}_{exp}]) \mid (\text{nil}) \} \end{aligned}$$

B. Kerangka Kerja Model Akses Terintegrasi

Pengembangan kerangka kerja model akses terintegrasi dilakukan untuk mendefinisikan komponen-komponen yang menyusun model akses dan fungsionalitasnya. Proses pengembangan kerangka kerja model akses mencakup tahapan sebagai berikut:

1. Identifikasi awal komponen-komponen penyusun kerangka kerja model akses.

Hasil identifikasi awal komponen penyusun kerangka kerja model akses dapat dilihat pada Gambar 2.



Gambar 4. Komponen Kerangka Kerja Model Akses

2. Spesifikasi interaksi antar komponen

Spesifikasi interaksi antar komponen harus memenuhi persyaratan:

- Pengguna harus dapat menyatakan persyaratan kualitas layanan dan pola aksesnya melalui komponen *User Interface*
- Komponen *QoS Mapper* mentransformasikan spesifikasi QoS pengguna ke himpunan persyaratan QoS jaringan
- Komponen *Evaluation Function* melakukan pengecekan terhadap kesesuaian persyaratan QoS pengguna dengan ketersediaan sumber daya
- Berdasarkan hasil evaluasi, komponen *Access Mechanism* menetapkan layanan yang akan diberikan kepada pengguna

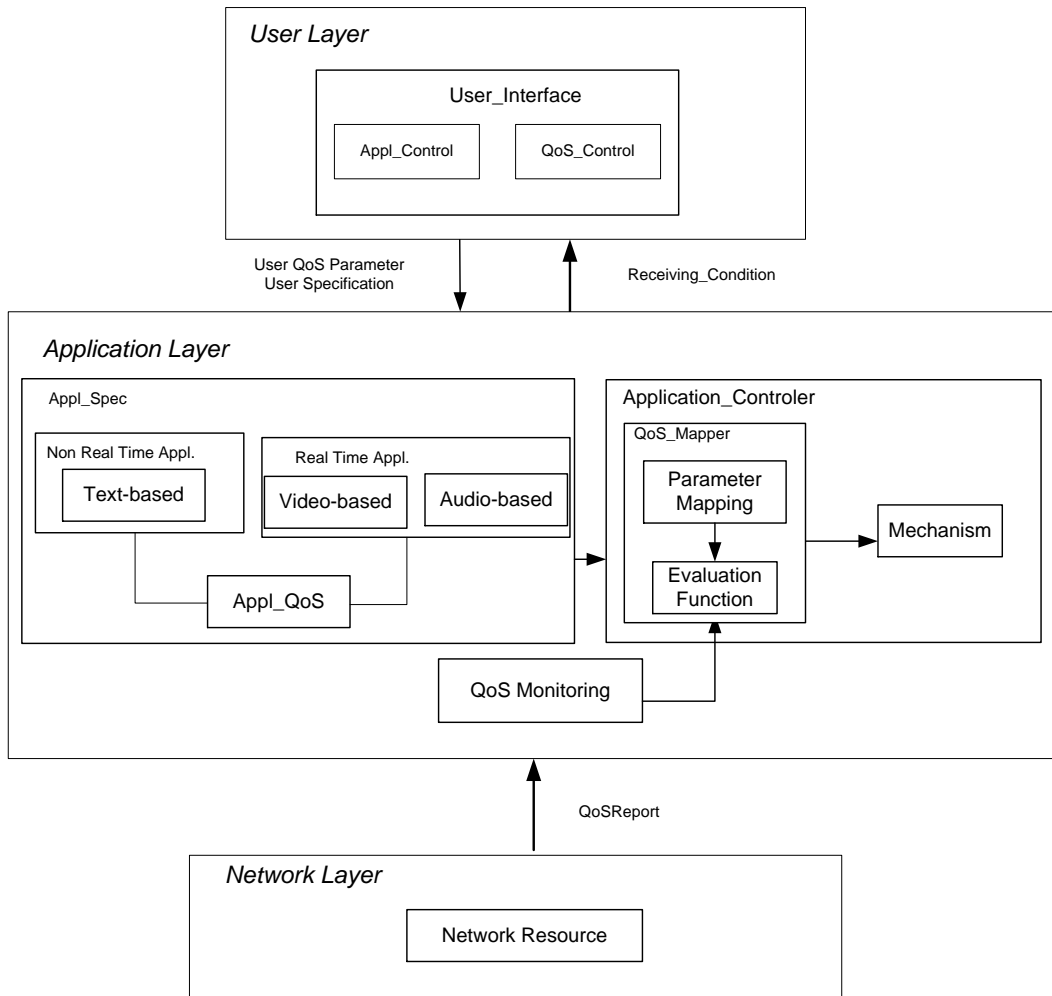
3. Identifikasi fungsionalitas setiap komponen

Komponen penyusun kerangka kerja model akses memiliki fungsi sebagai berikut:

- *User Interface* menampung preferensi pengguna berupa pola akses dan parameter QoS yang terkait dengan media aplikasi yang digunakan
- *Media Application* mengelola tipe media dan parameter yang terkait dengan tipe media aplikasi yang digunakan
- *QoS Monitoring* melakukan pengecekan ketersediaan sumber daya
- *Evaluation Function* melakukan perbandingan nilai-nilai QoS yang dispesifikasi oleh pengguna dengan nilai aktual QoS jaringan

- *Access Mechanism* menetapkan layanan yang sesuai bagi pengguna berdasarkan pola akses, parameter QoS serta ketersediaan sumberdaya

Hasil pengembangan kerangka kerja model akses terintegrasi digambarkan melalui diagram blok fungsional (Gambar 5).



Gambar 5. Diagram Blok Fungsional Kerangka Kerja

Pada *User Layer*, terdapat blok fungsional yang disebut *User_Interface*. *User_Interface* berfungsi sebagai antarmuka untuk menerima dan memproses spesifikasi akses pengguna berkaitan dengan media aplikasi dan kualitas layanan yang diharapkan.

Pada *Application Layer*, terdapat 2 blok fungsional yaitu *Appl_Spec* dan *Application_Controller*. Bagian ini berfungsi melakukan konfigurasi aplikasi agar dapat menyediakan layanan secara optimal berdasarkan spesifikasi kualitas layanan pengguna dan menyesuaikan kondisi sumberdaya jaringan. *Appl_Spec* merupakan

komponen spesifikasi yang menyediakan fungsi untuk melakukan konfigurasi aplikasi berdasarkan tipe media aplikasi yang dinyatakan pengguna melalui *User_Interface*. Setiap permintaan terhadap suatu aplikasi dapat melibatkan sejumlah media seperti *browser*, *audio* maupun *video* yang dibatasi oleh persyaratan kualitas layanan.

Implikasi dari spesifikasi kualitas layanan pengguna adalah nilai parameter kualitas layanan yang dinyatakan pengguna harus diinterpretasikan ke parameter kualitas layanan yang dipahami di lapis abstraksi lebih rendah. Untuk itu fungsi *Mapping* dikembangkan untuk mentranslasikan nilai parameter kualitas layanan pengguna ke himpunan parameter kualitas layanan aplikasi dan jaringan. *Application_Controller* memuat fungsi *QoS Mapper* yang terdiri dari *Parameter Mapping* dan *Evaluation Function*. Selain *QoS Mapper*, *Application_Controller* juga memuat fungsi *Mechanism* untuk menetapkan layanan yang dapat disediakan oleh sistem berdasarkan preferensi pengguna. Fungsi *QoS Monitoring* digunakan untuk mengambil informasi kualitas layanan yang disediakan (*QoS offer*) oleh lapis jaringan. Nilai-nilai kualitas layanan jaringan yang dihasilkan kemudian dievaluasi menggunakan *Evaluation Function* untuk mengetahui apakah kualitas layanan jaringan yang ditawarkan memungkinkan aplikasi beroperasi secara optimal.

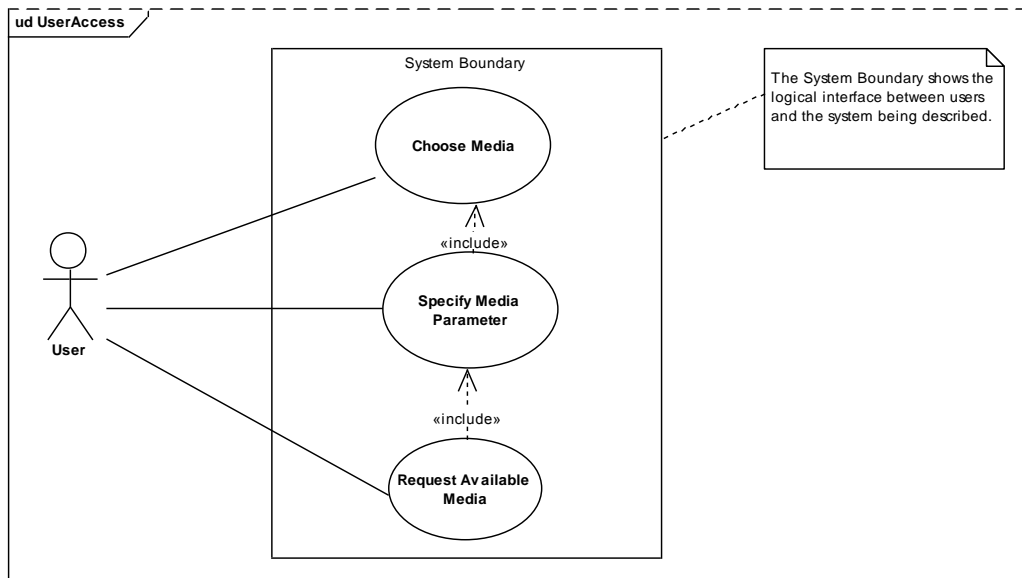
Ketika menerima *request* dari *User Layer*, *Application_Controller* mengaktifkan fungsi *Parameter Mapping* dan *Evaluation Function* untuk mentranslasikan parameter kualitas layanan pengguna ke parameter kualitas layanan aplikasi. Berdasarkan proses *mapping* dan *Evaluation Function*, *Application Control* akan menentukan mekanisme yang akan digunakan untuk memenuhi persyaratan kualitas layanan pengguna. Pemilihan mekanisme oleh *Application Control* didasarkan pada informasi yang dihasilkan oleh *QoS Monitoring*, *Appl_QoS* dan persyaratan kualitas layanan pengguna. Dalam kerangka kerja model akses terintegrasi ini, fungsi *Monitoring QoS* merupakan fungsi tambahan yang digunakan untuk memantau kualitas layanan aktual jaringan.

C. Desain Arsitektur Kerangka Kerja

Desain arsitektur model akses digunakan sebagai dasar untuk pengembangan modul-modul komponen yang menyusun kerangka kerja model akses. Terdapat 3 tahapan proses dalam pengembangan modul-modul komponen, yaitu penentuan persyaratan, model analisis dan model desain.

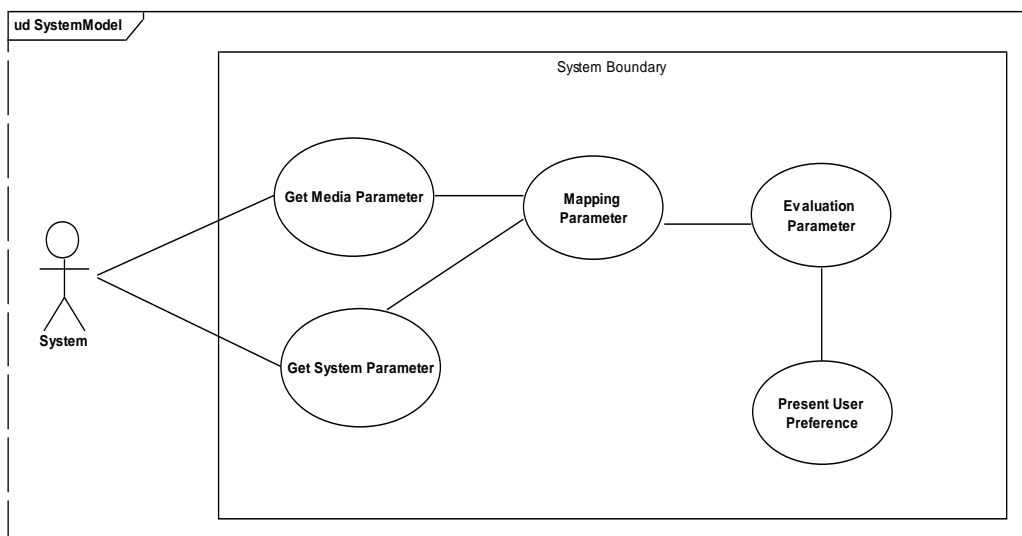
1. Penentuan Persyaratan

Penentuan persyaratan bertujuan mengidentifikasi persyaratan fungsionalitas sistem. Persyaratan fungsionalitas sistem digambarkan menggunakan diagram *Use Case*.



Gambar 6. *Use Case Diagram* untuk Akses Pengguna

Gambar 6 merupakan diagram *Use Case* yang menggambarkan komponen fungsionalitas dari sisi pengguna. Diperlukan 3 komponen fungsionalitas yang terkait dengan spesifikasi akses pengguna, yaitu pemilihan media aplikasi (*ChosenMedia*), penetapan parameter kualitas layanan sesuai media aplikasi yang dipilih (*SpecifyMediaParameter*) dan pengiriman permintaan terhadap ketersediaan aplikasi yang sesuai preferensi pengguna (*RequestAvailableMedia*).

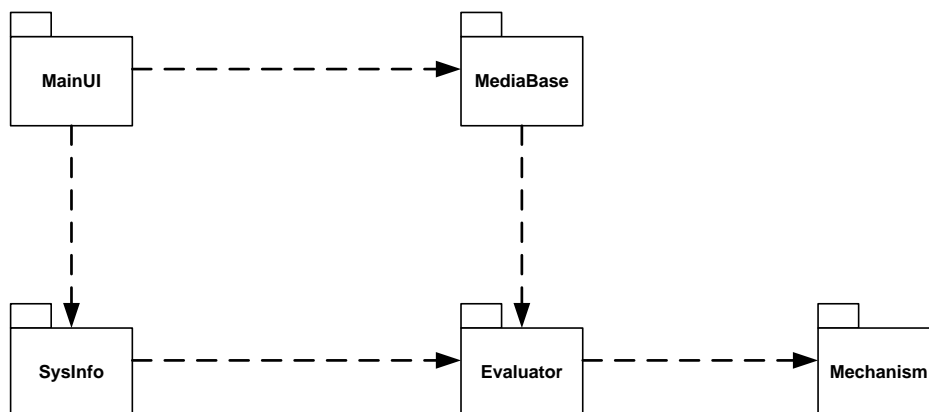


Gambar 7. *Use Case Diagram* untuk Sistem

Gambar 7 merupakan diagram *Use Case* yang menggambarkan komponen fungsionalitas dari sisi sistem. Diperlukan 5 komponen fungsionalitas, yaitu *GetMediaParameter*, *GetSystemParameter*, *MappingParameter*, *EvaluationFunction* dan *PresentUserPreference*. Berdasarkan preferensi pengguna, sistem akan membandingkan spesifikasi kualitas layanan pengguna dengan spesifikasi kualitas layanan aplikasi. Dalam hal ini, komponen *GetMediaParameter* digunakan untuk mengakses kualitas layanan aplikasi. Untuk penetapan mekanisme layanan yang dapat diberikan kepada pengguna, sistem juga akan memperhitungkan kondisi aktual jaringan. Dalam hal ini, komponen *GetSystemParameter* digunakan untuk melakukan monitor terhadap kondisi aktual jaringan. Komponen *MappingParameter* berfungsi mentranslasikan parameter kualitas layanan pada level abstraksi yang berbeda. Komponen *EvaluationFunction* berfungsi membandingkan nilai-nilai parameter kualitas layanan pengguna dengan nilai parameter kualitas layanan aplikasi dan sistem. Hasil evaluasi akan menetapkan mekanisme yang sesuai untuk memenuhi permintaan pengguna (*PresentUserPreference*).

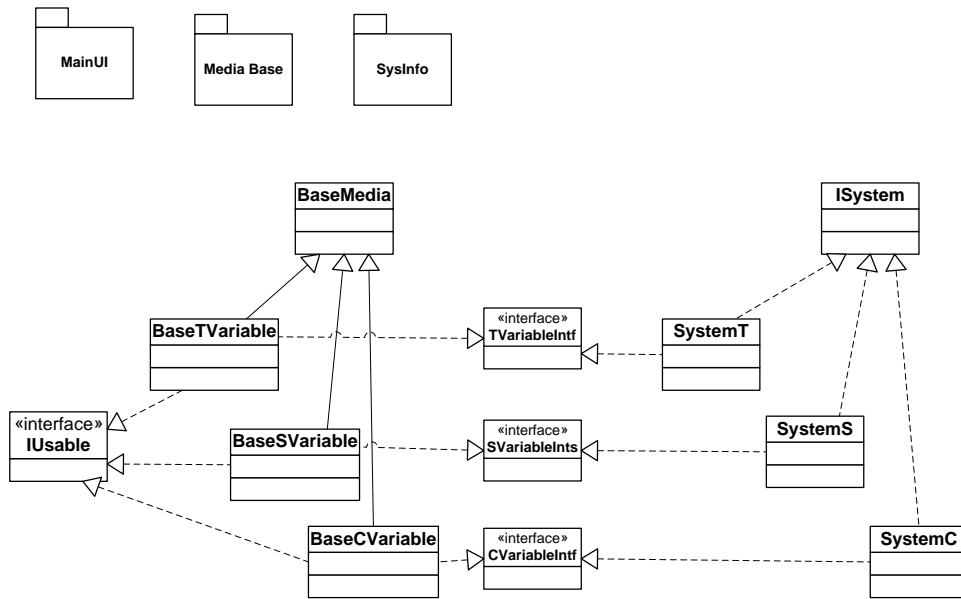
2. Model Analisis

Digunakan untuk melakukan pemetaan awal terhadap persyaratan fungsionalitas sistem. Tahapan ini dijelaskan pada Gambar 8. Kerangka kerja model akses dipetakan menjadi komponen-komponen *MainUI*, *MediaBase*, *SysInfo*, *Evaluator* dan *Mechanism*. Masing-masing komponen mewakili fungsionalitas sistem.



Gambar 8. Pemetaan Awal Persyaratan Fungsionalitas

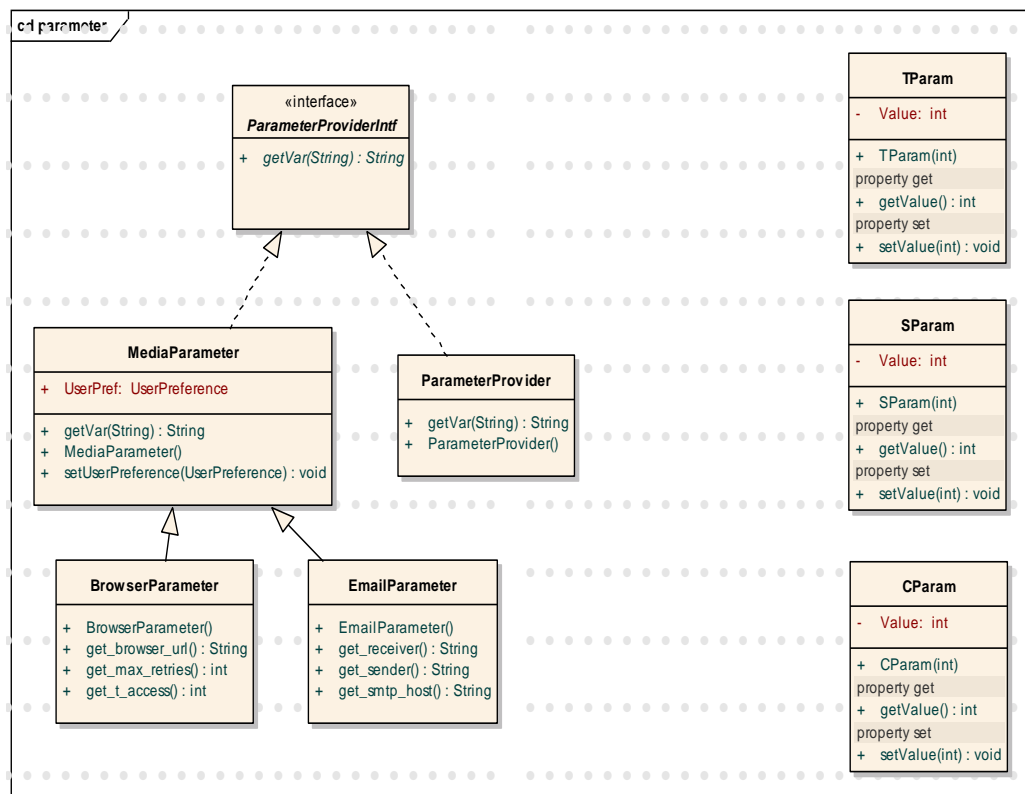
Pada tahap pemetaan awal, setiap komponen selanjutnya dikembangkan dalam *class-class* tertentu, seperti terlihat pada Gambar 9.



Gambar 9. Class Diagram Komponen

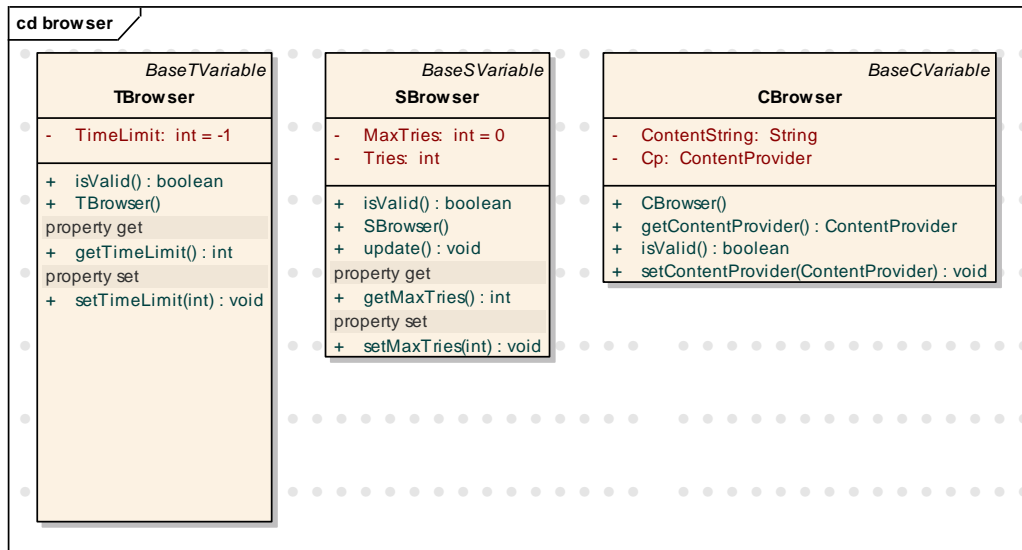
3. Model Desain

Digunakan untuk mengimplementasikan model akses yang telah didesain. Pada tahap ini didefinisikan *class-class* yang mewakili fungsionalitas setiap komponen.



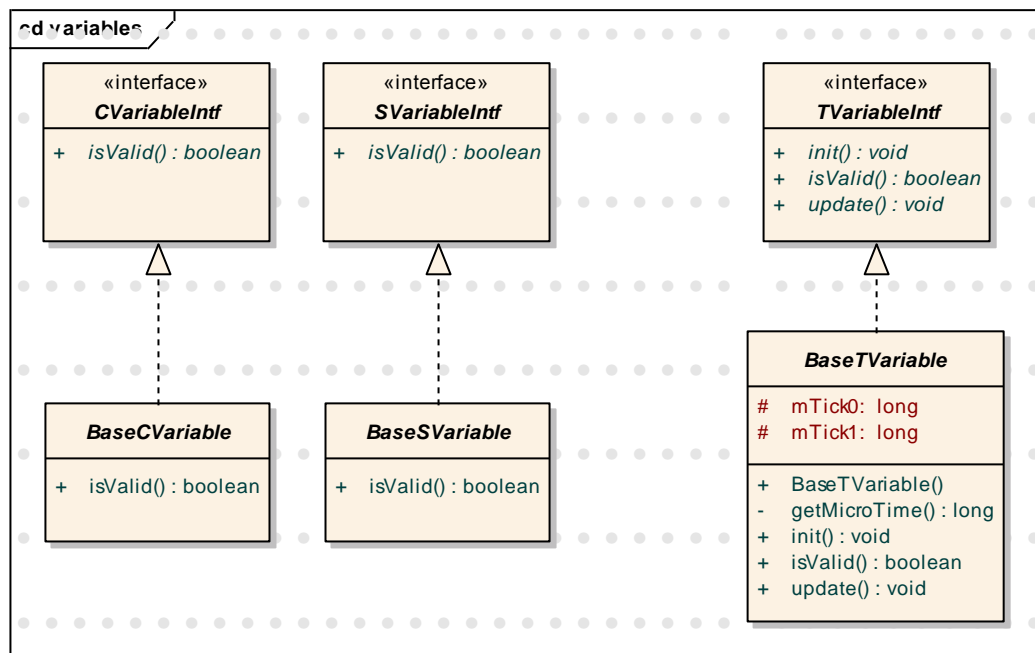
Gambar 10. Class Diagram parameter

Komponen *MediaBase*, mendefinisikan *class-class* yang bersesuaian dengan jenis aplikasi seperti *browser* dan *email*. Pada komponen ini terdapat sub *class* *MediaParameter* (Gambar 10) yang digunakan untuk mendefinisikan parameter-parameter yang terkait dengan berbagai jenis aplikasi.



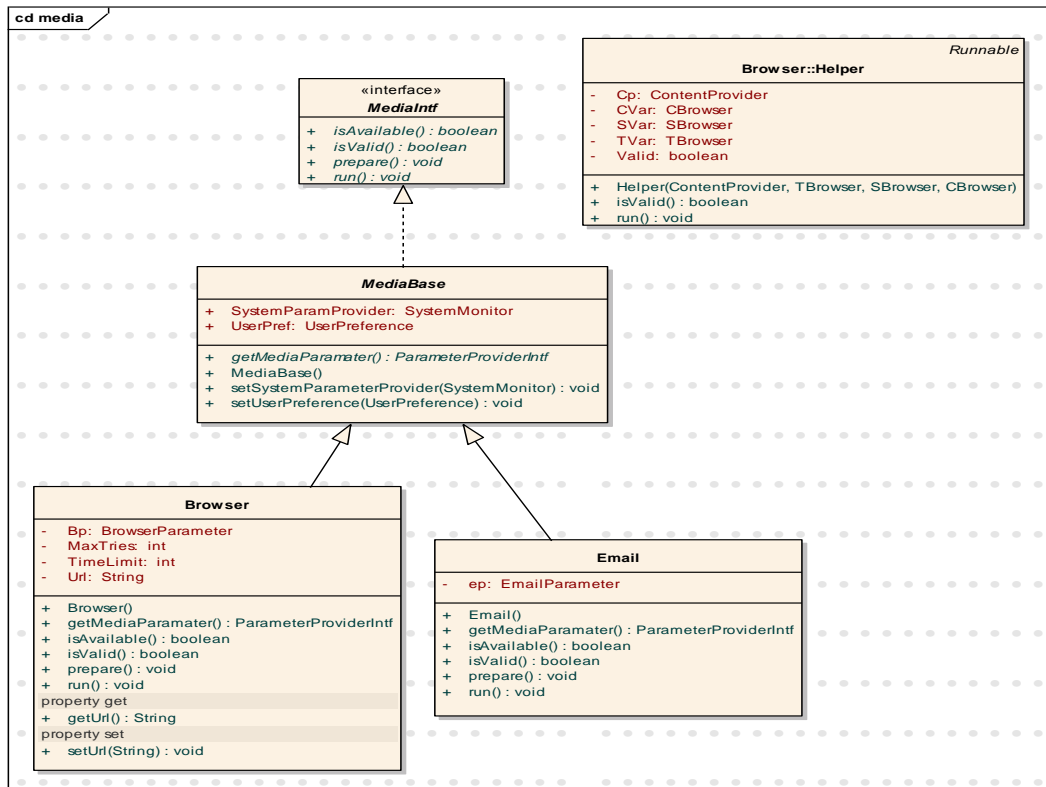
Gambar 11. *Class Diagram* browser

Gambar 11 merepresentasikan *class diagram browser*, yang didefinisikan di dalam komponen *MediaBase*. *Class* ini memuat tiga parameter kualitas layanan yang terkait dengan media aplikasi *browser* seperti *TimeLimit* mewakili karakteristik waktu, *MaxTries* mewakili karakteristik keberhasilan dan *ContentString* mewakili karakteristik kesesuaian isi.



Gambar 12. *Class Diagram* variable

Gambar 12 merepresentasikan *class diagram variable*. *Class variable* memiliki metode *isValid()* untuk mengevaluasi nilai parameter kualitas layanan dan memberikan nilai *true* jika nilai parameter kualitas layanan terpenuhi dan *false* jika sebaliknya.



Gambar 13. *Class Diagram* media

Gambar 13 menjelaskan *class diagram* media. *Class Browser* dan *class Email* merupakan dua dari media aplikasi yang digunakan. Kedua class ini diturunkan dari *class MediaBase* sehingga memiliki properti dan metode yang dimiliki oleh *class MediaBase*. Disamping itu, *class Browser* memiliki properti yang berkaitan dengan parameter kualitas layanan aplikasi *browsing* seperti *TimeLimit* untuk parameter t, *MaxTries* untuk parameter s dan *MinMatch* untuk parameter c.

D. Rasionalisasi Kerangka Kerja

Untuk menjelaskan konsep yang mendasari pengembangan kerangka kerja, digunakan suatu skenario yang mewakili pola akses pengguna dalam melakukan akses Internet. Konsep terintegrasi terlihat pada model yang menyediakan opsi bagi

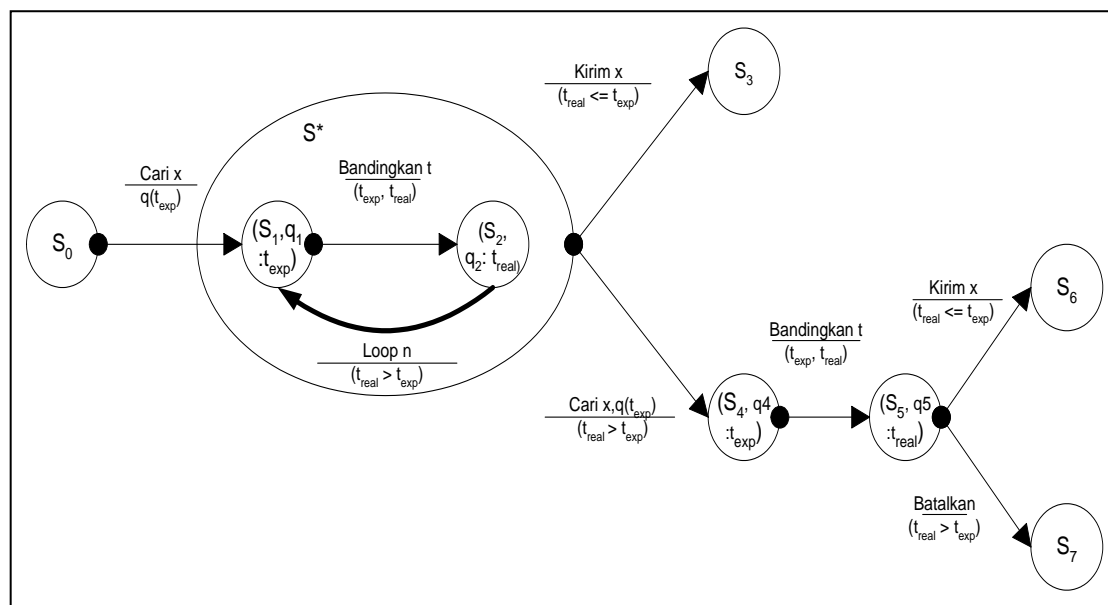
pengguna untuk menyatakan akses alternatif jika respons yang diharapkan tidak terpenuhi.

1. Studi Kasus : Aplikasi *Browsing*

Skenario ini menampilkan suatu pola akses untuk melakukan *browsing* dengan parameter t dan mengirimkan hasilnya ke suatu alamat email. Jika persyaratan kualitas layanan tidak terpenuhi, pengguna akan meminta sistem untuk mengulangi proses beberapa kali (n). Jika masih gagal, alihkan pencarian ke situs lain. Spesifikasi akses dinyatakan sebagai berikut:

- 1) Cari materi tentang "Distributed Databases" pada alamat situs <http://elearning.uny.ac.id/course/> dengan ekspektasi $q(t \leq 4)$.
- 2) Jika $q(t)$ terpenuhi, kirim x ke ratna@uny.ac.id
- 3) Jika $q(t)$ tidak terpenuhi, lakukan proses pencarian ulang $q(s)$ selama 3 kali
- 4) Jika masih gagal, cari di <http://www.e-learningcenter.com/freecourses.htm> dan kirim x ke alamat yang ditentukan jika q terpenuhi.
- 5) Jika q tidak terpenuhi, maka batalkan akses.

Berdasarkan model dan skema spesifikasi akses yang telah dibuat, skenario tersebut dapat diimplementasikan sebagai berikut. Gambar 14 merepresentasikan model akses menggunakan *state transition diagram*.



Gambar 14. Diagram Model Akses Pengguna

Skema spesifikasi akses pengguna dinyatakan sebagai:

$$S_0 : \{ (\text{nil}) \mid \text{initial}() \mid (S_1, q_{\text{exp}}) \}$$

$S^* : \{ (S_0, \text{initial}()) \mid \text{Loop}(n) \mid (S_4, t_{\text{exp}}) \}$
 [Loop(n):
 $S_1 : \{ (S_2, t_{2\text{real}} > t_{\text{exp}}) \mid \text{find}(\text{"distributed Databases"} ,$
 <http://elearning.uny.ac.id/course/>), $q(t_{\text{exp}} \leq 4) \mid ((S_2, t_{\text{real}})) \}$
 $S_2 : \{ (S_1, (t_{\text{exp}})) \mid \text{compare}(t_{\text{exp}}, t_{2\text{real}}) \mid ((t_{2\text{real}} \leq t_{\text{exp}}) \rightarrow S_3 \vee (t_{2\text{real}} >$
 $t_{\text{exp}}) \rightarrow S_1) \}$]
 $S_3 : \{ (S_2, (t_{2\text{real}} \leq t_{\text{exp}})) \mid \text{send}(\text{"Distributed Databases"} , \text{ratna@uny.ac.id}) \mid$
 (nil) }
 $S_4 : \{ (S_2, (t_{2\text{real}} > t_{\text{exp}})) \mid \text{find}(\text{"Distributed Databases"} ,$
 <http://www.e-learningcenter.com/freecourses.htm>), $q(t_{\text{exp}} \leq 4) \mid (S_5, t_{\text{real}}) \}$

 $S_5 : \{ (S_4, (t_{\text{exp}})) \mid \text{compare}(t_{\text{exp}}, t_{5\text{real}}) \mid ((t_{5\text{real}} \leq t_{\text{exp}}) \rightarrow S_6$
 $\vee (t_{5\text{real}} > t_{\text{exp}}) \rightarrow S_7) \}$
 $S_6 : \{ (S_5, (t_{5\text{real}} \leq t_{\text{exp}})) \mid \text{send}(x, \text{ratna@uny.ac.id}) \mid (\text{nil}) \}$
 $S_7 : \{ (S_5, (t_{5\text{real}} > t_{\text{exp}})) \mid \text{Cancel}() \mid (\text{nil}) \}$

2. Alur Proses

Berdasarkan kerangka kerja, spesifikasi akses pengguna mengikuti alur proses sebagai berikut:

1. Spesifikasi akses pengguna diproses pada *class UserPreference*. Spesifikasi akses pengguna mencakup alamat URL sesuai topik yang dicari dan preferensi pengguna terhadap kualitas layanan.
2. Aktivasi parameter aplikasi (*BrowserParameter*) dan aktivasi variabel-variabel yang terkait dengan parameter aplikasi (*TBrowser*, *SBrowser* dan *CBrowser*)
3. Modul *QoS_Mapper* melakukan translasi parameter kualitas layanan pengguna dan aplikasi serta mengevaluasi nilai-nilai parameter yang dispesifikasi pengguna.
 - a. *Mapping*:

Mapping parameter disajikan pada Tabel 6.

Tabel 6. *Mapping* parameter QoS pengguna ke QoS browser

QoS Pengguna	QoS Browser
Access_time	$Access_time = resp_transfer_time - req_transfer_time$
Retry_amount	Max_retry
Content	$Content_length = response_length$ atau $code_status = 2xx$

b. *Evaluasi function:*

Fungsi *isValid()* memberikan nilai *true* jika persyaratan kualitas layanan pengguna terpenuhi dan nilai *false* untuk persyaratan yang tidak terpenuhi.

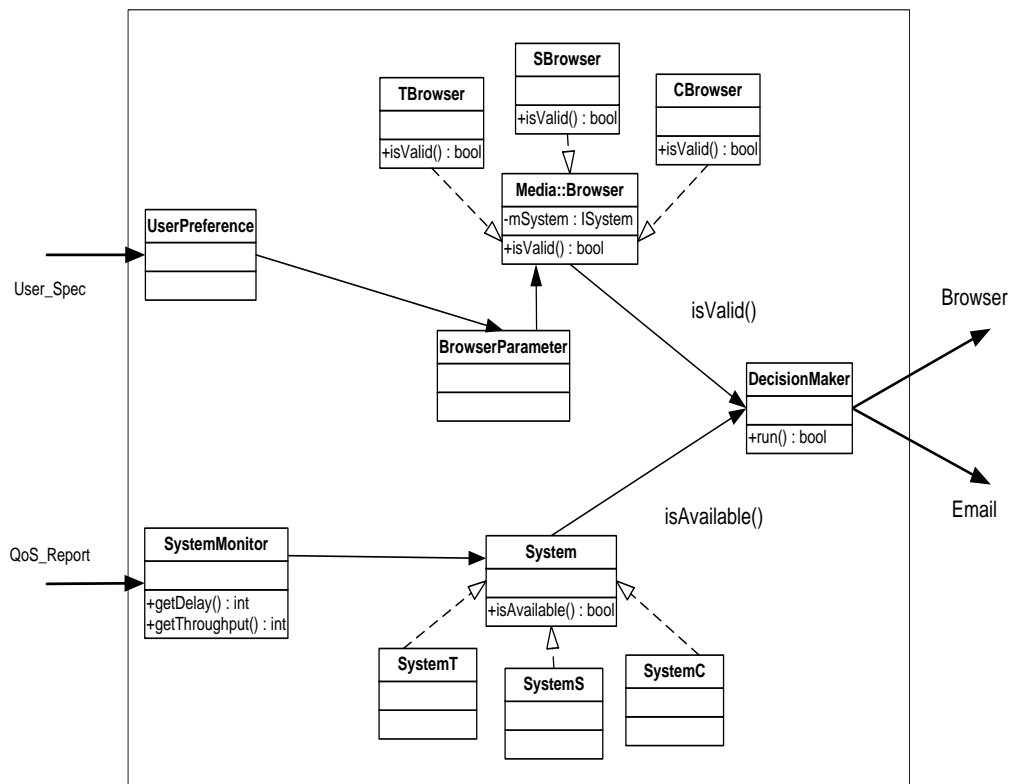
4. Aktivasi *SystemMonitor* dan fungsi *isAvailable()* memberikan nilai *true* jika ketersediaan *resource* memenuhi persyaratan minimal aplikasi.

5. Fungsi *DecisionMaker*:

- Jika *resource* tersedia (fungsi *isAvailable()* bernilai *true*) dan persyaratan QoS pengguna terpenuhi (fungsi *isValid()* bernilai *true*), aplikasi dapat dijalankan.
- Jika *resource* tidak tersedia, permintaan pengguna dialihkan ke akses alternatif sesuai spesifikasi awal.

3. Arsitektur Modul Perangkat Lunak

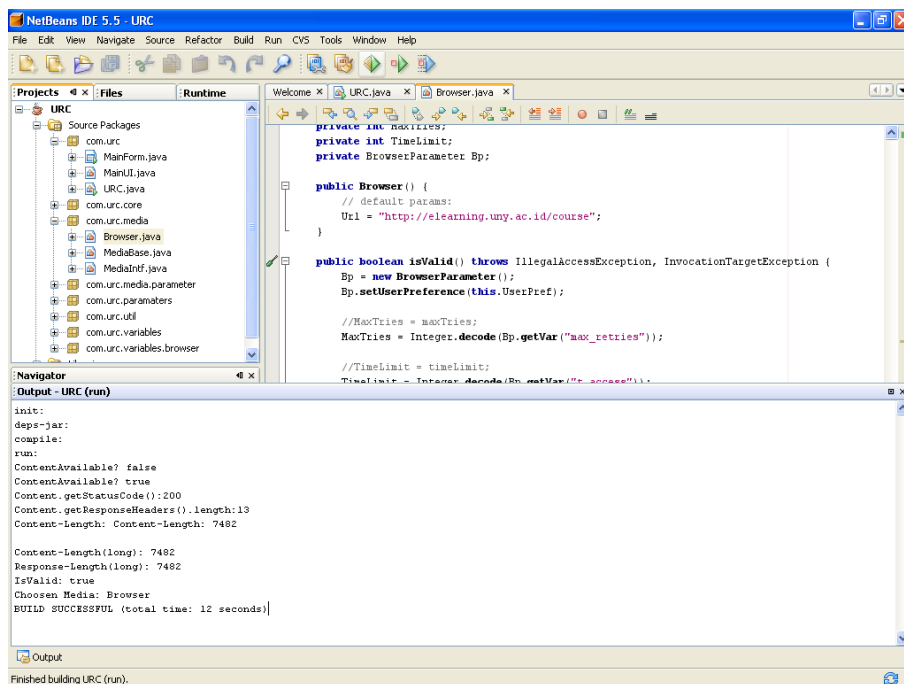
Sesuai desain arsitektur yang dibuat, implementasi kerangka kerja untuk skenario aplikasi *browsing* dapat dilihat pada Gambar 15. Implementasi untuk jenis aplikasi yang lain memiliki alur proses yang sama, sehingga model bersifat generik.



Gambar 15. Diagram Implementasi

4. Uji Fungsionalitas Komponen

Skenario uji coba dilakukan untuk mengevaluasi fungsionalitas dari proses-proses utama kerangka kerja model akses. Tampilan program yang menunjukkan hasil pengujian jika spesifikasi akses pengguna dapat dipenuhi oleh sistem dapat dilihat pada Gambar 16 dan Gambar 17. Pada Gambar 16, spesifikasi akses pengguna yang dinyatakan melalui parameter $t = 4000$ ms terpenuhi setelah terjadi pengulangan akses sebanyak dua kali ($s = 2$).



```
NetBeans IDE 5.5 - URC
File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Projects: URC
  Source Packages
  com.urc
  MainForm.java
  MainUI.java
  URC.java
  com.urc.core
  com.urc.media
  Browser.java
  MediaBase.java
  MediaInf.java
  com.urc.media.parameter
  com.urc.parameters
  com.urc.util
  com.urc.variables
  com.urc.variables.browser

Runtime: Welcome x URC.java x Browser.java x
private int maxTries;
private int TimeLimit;
private BrowserParameter Bp;

public Browser() {
  // default params:
  Url = "http://elearning.umy.ac.id/course/";
}

public boolean isValid() throws IllegalAccessException, InvocationTargetException {
  Bp = new BrowserParameter();
  Bp.setUserPreference(this.UserPref);

  //MaxTries = maxTries;
  MaxTries = Integer.decode(Bp.getVar("max_retries"));

  //TimeLimit = timeLimit;
  TimeLimit = Integer.decode(Bp.getVar("t_response"));
}

Navigator:
Output - URC (run)
init:
deps-jar:
compile:
run:
Content-Available? false
Content-Available? true
Content.getStatusCode(): 200
Content.getResponseHeaders().length: 13
Content-Length: Content-Length: 7482

Content-Length(long): 7482
Response-Length(long): 7482
IsValid: true
Chosen Media: Browser
BUILD SUCCESSFUL (total time: 12 seconds)

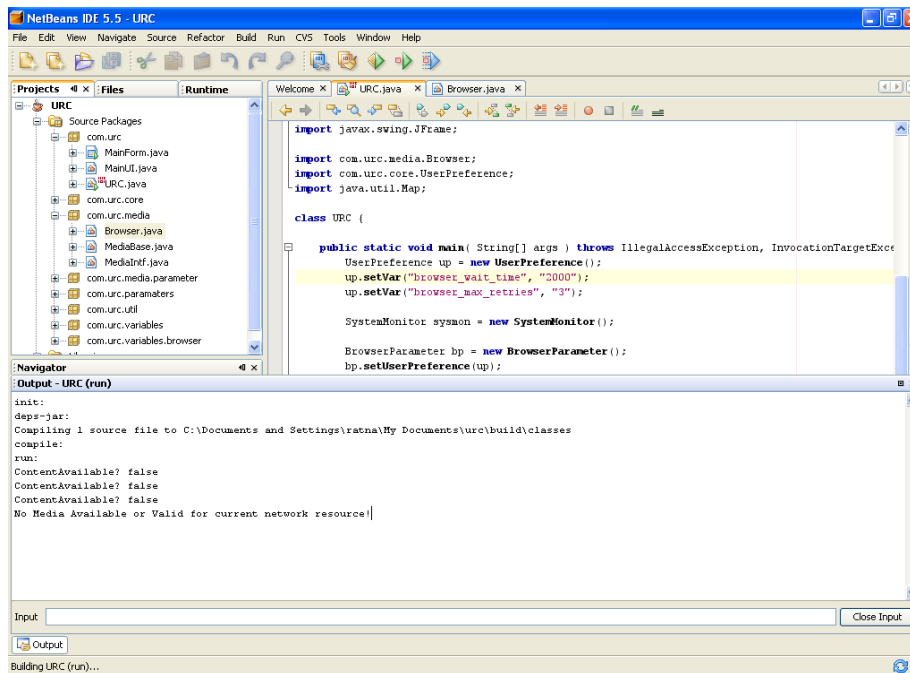
Output
Finished building URC (run).
```

Gambar 16. Tampilan Hasil Pengujian untuk Spesifikasi Akses Terpenuhi

Pada pengujian ini, karakteristik kesesuaian isi dievaluasi melalui perbandingan aspek *Content-Length* dengan *Response_Length*. Dari hasil uji coba, terlihat bahwa *Content-Length* dan *Response_Length* memiliki nilai yang sama, yaitu 7482. Ini berarti bahwa panjang isi pesan yang dikirim sesuai. *Code_status* bernilai 2xx berarti bahwa *request* pengguna diterima dan dimengerti.

Gambar 17 merepresentasikan pengujian dengan parameter $t = 2000$ ms dan $s = 3$ ($\text{max_tries} = 3$). Hasil pengujian menunjukkan spesifikasi akses pengguna tidak terpenuhi meskipun proses telah diulang sebanyak tiga kali. Kegagalan disebabkan karena salah satu dari *method isValid()* atau *method isAvailable()* memberikan nilai *false*. Ini berarti bahwa spesifikasi waktu respons yang diharapkan oleh pengguna tidak dapat dipenuhi oleh sistem. Dalam kondisi ini, sistem mengalihkan akses

pengguna ke akses alternatif yang telah didefinisikan oleh pengguna ketika menyatakan spesifikasi pola aksesnya.



Gambar 17. Tampilan Hasil Pengujian untuk Spesifikasi Akses Tidak Terpenuhi

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Beberapa hal yang dapat disimpulkan dari penelitian ini adalah:

4. Pencapaian tingkat kualitas layanan untuk akses Internet pada jaringan berkecepatan rendah dapat didekati dari perspektif pengguna melalui penyediaan model akses yang sesuai untuk kondisi jaringan dengan kualitas koneksi yang tidak handal.
5. Penelitian ini menghasilkan:
 - a. Kerangka kerja model akses terintegrasi yang diperlukan sebagai dasar pengembangan model akses
 - b. Spesifikasi model akses yang direpresentasikan dengan skema sebagai berikut :

$$S_i : \{ (S_n, f) \mid (a_i, [q_{exp}]) \mid (f \rightarrow S_t \vee \neg f \rightarrow S_t) \}$$

Dengan skema tersebut, pola akses dan parameter kualitas layanan yang diharapkan pengguna dapat dinyatakan dengan jelas dan mudah.

6. Model akses terintegrasi dapat digunakan sebagai satu alternatif dalam melakukan diferensiasi layanan guna mengatasi masalah pemenuhan kualitas layanan akses Internet pada jaringan yang memiliki kualitas koneksi jaringan yang tidak handal.

B. Saran

Desain *prototype browser* yang dihasilkan dari penelitian ini perlu diimplementasikan lebih lanjut melalui pengembangan model *user interface* sebagai *proxy server* yang berada dalam *web browser*. Penggunaan *proxy server* bertujuan agar ketika kondisi jaringan tidak memungkinkan untuk melakukan akses secara *on-line*, maka permintaan akses pengguna dapat dilakukan secara *off-line*. Diharapkan model *browser* yang dihasilkan lebih sesuai digunakan untuk akses Internet pada kondisi jaringan dengan kualitas koneksi yang tidak handal.

DAFTAR PUSTAKA

- Aagedal, J. Ø.**, Maret 2001, *Quality of Service Support in Development of Distributed Systems*, PhD thesis, Department of Informatics, Faculty of Mathematics and Natural Sciences. Oslo: University of Oslo.
- Alfano, M.**, *A Quality of Service Management Architecture (QoSMA) : a preliminary study*, Desember 1995, International Computer Science Institute Berkeley, California.
- Angin, O., Campbell, A. T., Cheok, L-T., Liao, Raymond, R-F., Lim, K-S., Nahrstedt, K.**, May 1997, *Report on the 5th IFIP International Workshop on Quality of Service (IWQOS'97)*, Center for Telecommunications Research Columbia University, New York City.
URL : <http://comet.ctr.columbia.edu/iwqos97/>
- Asensio, J.I., Villagra, V.A.**, *A UML for QoS Management Information Specification in Distributed Object-based Application*.
- Babulak, E.**, *The IT Network Quality of Service Provision Analysis in Light of The User's Perception and Expectation*.
URL : <http://www.soc.staffs.ac.uk>
- Bertino, E., A. K. Elmagarmid, M. S. Hacid**, 2004, *A Logical Approach Quality of Service Specification in Video Database*, Multimedia Tools and Applications, 23, 75-101.
- Braden, C. and Shenker.**, Juni 1994, *Integrated Service in the Internet Architecture : an Overview*, RFC 1633.
- Campbell, A., Aurrecochea, C., Hauw, L.**, *A Review of QoS Architectures*, Center for Telecommunication Research, Columbia University, New York.
- Chen, Y., T. Farley, N. Ye**, 2003, *QoS Requirements of Network Applications on the Internet*, Department of Industrial Engineering, Arizona State University, Tempe, AZ, USA.
- Cisco System. Cisco IOS**, 2001, *Quality of Service Networking*, Cisco Press.,
URL:<http://www.cisco.com>
- El-Khatib, K. M.**, 2005, *A QoS Content Adaptation Framework for Nomadic user*, Ph.D. Thesis, School of Information Technology and Engineering University of Ottawa, Ontario, Canada.
- ETSI TR 00019 v1.1.5**, 2006, User Group; List of Definition and Abbreviations.
- Friedrichs, J., Jubin, H., dan Team Jalapeno**, 1999, *Java Thin-Client Programming for a Network Computing Environment*, Prentice Hall, New Jersey

- Frølund, S. and J. Koistinen**, 1998b, *Quality-of-Service Specification in Distributed Object Systems*, *Distributed Systems Engineering Journal*, 5, 179-202.
- Hansen, Gil.**, 1997, *Quality of Service (QoS)*, Object Service and Consulting, Inc.
- Hesselman, C., Widya, I., van Halteren, A., Nieuwenhuis, B.**, 2000, *Middleware Support for Media Streaming Establishment Driven by User-Oriented QoS Requirement*.
[URL:http://www.home.cs.utwente.nl/~halteren/publications/IDMS_paper_camera_ready.pdf](http://www.home.cs.utwente.nl/~halteren/publications/IDMS_paper_camera_ready.pdf)
- ISO/IEC JTC1/SC21/WG7**, *Basic reference model of Open Distributed Processing – Part 1: Overview*, ISO/IEC DIS 10746-1.
- Katchabaw, M.J.**, Juni 2002, *Quality of Service Resource Management*, Ph.D Thesis, The University of Western Ontario, London, Ontario.
- Pal, P., R. S. Loyall, J. Zinky, R. Shapiro, J. Megqueir**, 2001, *Using QDL to Specify QoS Aware Distributed (QuO) Application Configuration*, BBN Technologies, Cambridge.
- Prevost, J.**, 2001, *A Reliable Low-Bandwidth Email-Based Communication Protocol*, Master's Thesis, Massachusetts Institute of Technology, 2001.
- Purbo, O. W., Basalamah, A., Fahmi, I., Thamrin, A. H.**, Juli 2000, *TCP/IP*, Elex Media Komputindo, Jakarta
- Raisanen, V.**, Mei 2001, *Quality of Service & Voice-over-IP*, Dr. Dobb's Journal.
[URL:http://www.ddj.com](http://www.ddj.com)
- Ray, G.**, Juli 2000, *Quality of Service in Data Networks : Products*.
[URL:http://www.ohio-state/~jain/cis788-99/QoS_products/index.html](http://www.ohio-state/~jain/cis788-99/QoS_products/index.html)
- Schmidt., D., S. Lavine, M.**, April 1998, *The Design of the TAO Real-Time Object Request Broker*, *Computer Communications Journal*, 21 (4).
- Tsalianis, A. and A. A. Economides**, 2000, *QoS Standards for Distributed Multimedia Application*, Proceeding IEEE Communications Quality & Reliability, International Workshop, 13-17.
- Venkateswaran, R.**, Maret 2002, *Network QoS and IP Telephony*.
- Vogel, A., Kerhevre, B., Bochmann, G. V., Gecsei, J.**, November 1994, *Distributed Multimedia Applications and Quality of Service : A Survey*, Proceedings of the 1994 Centre for Advanced Studies Conference, Toronto, Canada.
- Wang, P., Yemini. Y., Florissi, D., Florissi, Y.**, Agustus 2000, *QOSME : Toward QoS Managemenet and Guarantees*, World Computer Congress - International Conference on Communication Technologies, Beijing, China.