

**PERANCANGAN DAN PEMBUATAN APLIKASI
SISTEM KATALOG PERPUSTAKAAN
BERBASIS J2ME**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh :

MUHAMMAD RIZA FAHRUDDIN
NIM. 0410633055

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2009

**PERANCANGAN DAN PEMBUATAN APLIKASI
SISTEM KATALOG PERPUSTAKAAN
BERBASIS J2ME**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

MUHAMMAD RIZA FAHRUDDIN
NIM. 0410633055

Disahkan dan disetujui oleh dosen pembimbing :

Pembimbing I

Pembimbing II

Arief Andy Soebroto, ST., M.Kom

NIP. 132 231 567

Herman Tolle, ST., MT

NIP. 132 283 206

**PERANCANGAN DAN PEMBUATAN APLIKASI
SISTEM KATALOG PERPUSTAKAAN
BERBASIS J2ME**

Disusun oleh :

MUHAMMAD RIZA FAHRUDDIN

NIM. 0410633055

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 13 Agustus 2009

Majelis Penguji :

Ir. Muhammad Aswin, MT

NIP. 131 879 045

Suprpto, ST, MT

NIP. 132 149 320

Ir. Heru Nurwarsito, M.Kom

NIP. 131 879 033

Mengetahui :

Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom

NIP. 131 879 033

KATA PENGANTAR

Segala puja dan puji syukur kehadirat Allah SWT karena dengan rahmat, berkah, karunia serta ridho-NYA penyusunan skripsi ini dengan judul “Perancangan dan Pembuatan Aplikasi Sistem Katalog Perpustakaan Berbasis J2ME” dapat diselesaikan. Penulis menyadari bahwa kajian ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Keluarga Besar Bapak Drs. H. M Shofi Ghufron untuk seluruh do’a dan dukungannya yang telah diberikan kepada Ananda selama studi hingga terselesaikannya skripsi ini.
2. Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
3. Rudy Yuwono, ST., MSc selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Arief Andy Soebroto, ST., M.Kom selaku dosen pembimbing pada penyusunan skripsi ini.
5. Herman Tolle, ST., MT selaku dosen pembimbing pada penyusunan skripsi ini.
6. Bapak dan Ibu dosen serta karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
7. Teman, sahabat, dan saudaraku yang telah bersama menjalani hari dengan suka dan duka. Terima kasih atas pelajaran hidup yang sangat berharga, semua kenangan tidak akan bisa terlupakan.
8. Serta semua pihak yang tak dapat disebutkan satu persatu yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Kesempurnaan hanya milik Allah SWT. Segala kritik dan saran yang sifatnya membangun dari pembaca tentang isi skripsi ini akan diterima dengan senang hati.

Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, Agustus 2009

Penyusun



DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	vi
DAFTAR GAMBAR	viii
ABSTRAK	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1 Katalog Perpustakaan.....	6
2.2 Teknologi Java.....	6
2.2.1 <i>Java2 Standard Edition (J2SE)</i>	7
2.2.2 <i>Java 2 Micro Edition (J2ME)</i>	7
2.2.2.1 J2ME Configuration.....	8
2.2.2.2 J2ME Profile.....	9
2.2.2.3 MIDlet.....	11
2.2.2.4 Koneksi HTTP pada J2ME.....	13
2.2.2.5 J2ME dan WAP.....	14
2.3 PHP.....	15
2.3.1 Konsep Kerja PHP.....	16
2.3.2 <i>Script PHP</i>	17
2.3.3 Style Kode PHP.....	17
2.3.4 Variabel pada PHP.....	19
2.3.5 Kelebihan PHP.....	19
2.4 Basis Data.....	20
2.4.1 MySQL.....	20



2.4.1.1	Keistimewaan MySQL.....	21
2.4.2	SQL (Structured Query Language).....	22
BAB III	METODE PENELITIAN	24
3.1	Studi Literatur.....	24
3.2	Perancangan.....	25
3.2.1	Analisis Kebutuhan (<i>Requirement Analysis</i>).....	25
3.2.2	Perancangan (<i>Design</i>).....	25
3.3	Implementasi.....	25
3.4	Pengujian Perangkat Lunak.....	26
3.5	Pengambilan Kesimpulan dan Saran.....	26
BAB IV	PERANCANGAN PERANGKAT LUNAK	27
4.1	Analisis Kebutuhan (<i>Requirements Analysis</i>).....	27
4.1.1	Identifikasi Aktor.....	27
4.1.2	Daftar Kebutuhan.....	28
4.1.3	Diagram <i>Use Case</i>	29
4.1.4	Skenario <i>Use Case</i>	29
4.2	Perancangan Perangkat Lunak.....	37
4.2.1	Perancangan Umum.....	38
4.2.2	Perancangan Detail.....	40
4.2.2.1	Diagram Kelas (<i>Class Diagram</i>).....	40
4.2.2.2	Diagram Sekuensial (<i>Sequence Diagram</i>).....	44
4.2.3	Perancangan <i>User Interface</i>	46
4.2.3.1	Perancangan Menu Utama Guest.....	46
4.2.3.2	Perancangan Menu Utama Member.....	47
4.2.3.3	Perancangan <i>Form Login</i>	47
4.2.3.4	Perancangan <i>Form Catalog</i>	47
4.2.3.5	Perancangan <i>Form Register</i>	48
4.2.3.6	Perancangan <i>Form Forgot Password</i>	49
4.2.3.7	Perancangan Form Account Password.....	49
BAB V	IMPLEMENTASI	50
5.1	Spesifikasi Sistem.....	50
5.1.1	Spesifikasi Perangkat Keras.....	50

5.1.2	Spesifikasi Perangkat Lunak.....	50
5.2	Batasan-Batasan Implementasi	51
5.3	Implementasi Class pada File Program.....	51
5.4	Implementasi Algoritma.....	53
5.4.1	Implementasi Algoritma Pencarian Koleksi	53
5.5	Implementasi Antarmuka Aplikasi	62
5.5.1	Implementasi Antarmuka Menu Utama Guest.....	62
5.5.2	Implementasi Antarmuka Menu Utama Member	63
5.5.3	Implementasi Antarmuka Login	64
5.5.4	Implementasi Antarmuka Register.....	65
5.5.5	Implementasi Antarmuka Forgot Password.....	65
5.5.6	Implementasi Antarmuka Catalog.....	66
5.5.7	Implementasi Antarmuka Account Password.....	68
5.5.8	Implementasi Antarmuka Property System	69
BAB VI	PENGUJIAN PERANGKAT LUNAK.....	70
6.1	Pengujian.....	70
6.1.1	Pengujian Unit.....	70
6.1.2	Pengujian Integrasi.....	76
6.1.3	Pengujian Validasi	77
6.1.3.1	Kasus Uji Validasi	78
6.1.3.2	Hasil Pengujian Validasi.....	80
6.1.4	Pengujian Performansi Koneksi.....	81
6.1.4.1	Pengujian Waktu Akses <i>Query</i>	81
6.1.4.2	Pengujian Performansi <i>Web Sever</i>	88
BAB VII	PENUTUP.....	90
7.1	Kesimpulan	90
7.2	Saran.....	90
	DAFTAR PUSTAKA.....	92
	LAMPIRAN.....	93

DAFTAR TABEL

Tabel 2.1 Perbandingan antara CLDC dan CDC	9
Tabel 2.2 Perbandingan MIDP 1.0 dan MIDP 2.0	10
Tabel 2.3 Atribut MIDlet	12
Tabel 4.1 Deskripsi Aktor	27
Tabel 4.2 Daftar kebutuhan fungsional dan non fungsional	28
Tabel 4.3 Skenario <i>use case</i> Membuat Akun Member	30
Tabel 4.4 Skenario <i>use case</i> Login	31
Tabel 4.5 Skenario <i>use case</i> Lupa Password	32
Tabel 4.6 Skenario <i>use case</i> Melihat Daftar Koleksi	33
Tabel 4.7 Skenario <i>use case</i> Melihat Detail Koleksi	35
Tabel 4.8 Skenario <i>use case</i> Mengubah Password Akun Member	35
Tabel 4.9 Skenario <i>use case</i> Melihat Petunjuk Penggunaan Aplikasi	36
Tabel 4.10 Skenario <i>use case</i> Logout	37
Tabel 5.1 Spesifikasi perangkat keras	50
Tabel 5.2 Spesifikasi perangkat lunak komputer	51
Tabel 5.3 Implementasi <i>class</i> pada kode program *.java dan *.php.....	51
Tabel 5.4 Penjelasan menu pada antarmuka Menu Utama Guest.....	63
Tabel 5.5 Penjelasan menu pada antarmuka Menu Utama Member.....	64
Tabel 5.6 Penjelasan menu pada antarmuka Login.....	64
Tabel 5.7 Penjelasan menu pada antarmuka Register.....	65
Tabel 5.8 Penjelasan menu pada antarmuka Forgot Password	66
Tabel 5.9 Penjelasan menu pada antarmuka Catalog.....	67
Tabel 5.10 Penjelasan menu pada antarmuka Search Result	67
Tabel 5.11 Penjelasan menu pada antarmuka Catalog Detail	68
Tabel 5.12 Penjelasan menu pada antarmuka Account Password	69
Tabel 5.13 Penjelasan menu pada antarmuka Property System.....	69
Tabel 6.1 <i>Test case</i> untuk pengujian unit operasi connectHttp()	72
Tabel 6.2 <i>Test case</i> untuk pengujian unit operasi getData()	73
Tabel 6.3 <i>Test case</i> untuk pengujian unit operasi selectCollection().....	75
Tabel 6.4 <i>Test case</i> untuk pengujian integrasi operasi eventCatalog().....	77



Tabel 6.5 Kasus Uji Membuat Akun Member	78
Tabel 6.6 Kasus Uji Login	78
Tabel 6.7 Kasus Uji Lupa Password	79
Tabel 6.8 Kasus Uji Melihat Daftar Koleksi.....	79
Tabel 6.9 Kasus Uji Melihat Detail Koleksi	79
Tabel 6.10 Kasus Uji Mengubah Password Akun Member	80
Tabel 6.11 Kasus Uji Logout	80
Tabel 6.12 Hasil pengujian validasi	80
Tabel 6.13 Hasil pengujian waktu akses <i>query</i> terhadap tabel anggota_rbte	81
Tabel 6.14 Hasil pengujian waktu akses <i>query</i> terhadap tabel pengarangkoleksi	82
Tabel 6.15 Hasil pengujian waktu akses <i>query</i> terhadap tabel paketkoleksi	83
Tabel 6.16 Hasil pengujian waktu akses <i>query</i> terhadap tabel pengarang.....	83
Tabel 6.17 Hasil pengujian waktu akses <i>query</i> terhadap tabel penerbit	84
Tabel 6.18 Hasil pengujian waktu akses <i>query</i> terhadap tabel paket.....	85
Tabel 6.19 Hasil pengujian waktu akses <i>query</i> terhadap tabel jeniskoleksi	85
Tabel 6.20 Hasil pengujian waktu akses <i>query</i> terhadap tabel koleksi.....	86
Tabel 6.21 Tabel rata-rata pengujian waktu akses <i>query</i>	87
Tabel 6.22 Spesifikasi perangkat keras untuk <i>server</i> Aplikasi Katalog Perpustakaan berbasis J2ME.....	88



DAFTAR GAMBAR

Gambar 2.1 Arsitektur J2ME	8
Gambar 2.2 Posisi CLDC pada J2ME	9
Gambar 2.3 Daur hidup MIDlet	11
Gambar 2.4 Mekanisme HTTP	13
Gambar 2.5 Skema konsep kerja PHP	16
Gambar 2.6 Hierarki database	20
Gambar 4.1 <i>Use Case Diagram</i> Sistem Katalog Perpustakaan	29
Gambar 4.2 <i>Package Diagram</i> Sistem Katalog Perpustakaan	38
Gambar 4.3 Relasi antar <i>class</i>	39
Gambar 4.4 <i>Class diagram</i> anggota package gui	41
Gambar 4.5 <i>Class diagram</i> anggota package control	42
Gambar 4.6 <i>Class diagram</i> anggota package midlet	42
Gambar 4.7 <i>Class diagram</i> anggota package model di client	43
Gambar 4.8 <i>Class diagram</i> anggota package model di server	44
Gambar 4.9 Diagram sekuensial Melihat Daftar Koleksi	45
Gambar 4.10 Diagram sekuensial Melihat Detail Koleksi	46
Gambar 4.11 Desain Menu Utama Guest	46
Gambar 4.12 Desain Menu Utama Member	47
Gambar 4.13 Desain Form Login	47
Gambar 4.14 Desain Form Catalog	48
Gambar 4.15 Desain Form Register	48
Gambar 4.16 Desain Form Forgot Password	49
Gambar 4.17 Desain Form Account Password	49
Gambar 5.1 Diagram alir pencarian koleksi untuk eventCatalog()	54
Gambar 5.2 Algoritma pencarian koleksi untuk eventCatalog()	55
Gambar 5.3 Diagram alir pencarian koleksi untuk connectHttp()	56
Gambar 5.4 Algoritma pencarian koleksi untuk connectHttp()	57
Gambar 5.5 Diagram alir pencarian koleksi untuk getData()	58
Gambar 5.6 Algoritma pencarian koleksi untuk getData()	59
Gambar 5.7 Diagram alir pencarian koleksi untuk selectCollection()	60



Gambar 5.8 Algoritma pencarian koleksi untuk selectCollection().....	61
Gambar 5.9 Antarmuka aplikasi Menu Utama Guest	63
Gambar 5.10 Antarmuka aplikasi Menu Utama Member.....	63
Gambar 5.11 Antarmuka aplikasi Login.....	64
Gambar 5.12 Antarmuka aplikasi Register	65
Gambar 5.13 Antarmuka aplikasi Forgot Password	66
Gambar 5.14 Antarmuka aplikasi Catalog.....	66
Gambar 5.15 Antarmuka aplikasi Search Result	67
Gambar 5.16 Antarmuka aplikasi Catalog Detail.....	68
Gambar 5.17 Antarmuka aplikasi Account Password.....	68
Gambar 5.18 Antarmuka aplikasi Property System.....	69
Gambar 6.1 Pemodelan algoritma connectHttp() ke dalam flow graph.....	71
Gambar 6.2 Pemodelan algoritma getData() ke dalam flow graph.....	72
Gambar 6.3 Pemodelan algoritma selectCollection() ke dalam flow graph	74
Gambar 6.4 Pemodelan algoritma eventCatalog() ke dalam flow graph.....	76
Gambar 6.5 Grafik hubungan antara data entry dan rata-rata kecepatan waktu akses <i>query</i>	87
Gambar 6.6 Grafik linieritas kecepatan waktu akses <i>query</i>	88
Gambar 6.7 Pengujian performansi <i>Web Server</i> Apache HTTP.....	89



ABSTRAK

MUHAMMAD RIZA FAHRUDDIN. 2009. : Perancangan dan Pembuatan Aplikasi Sistem Katalog Perpustakaan Berbasis J2ME. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Arief Andy Soebroto, ST., M.Kom dan Herman Tolle, ST., MT

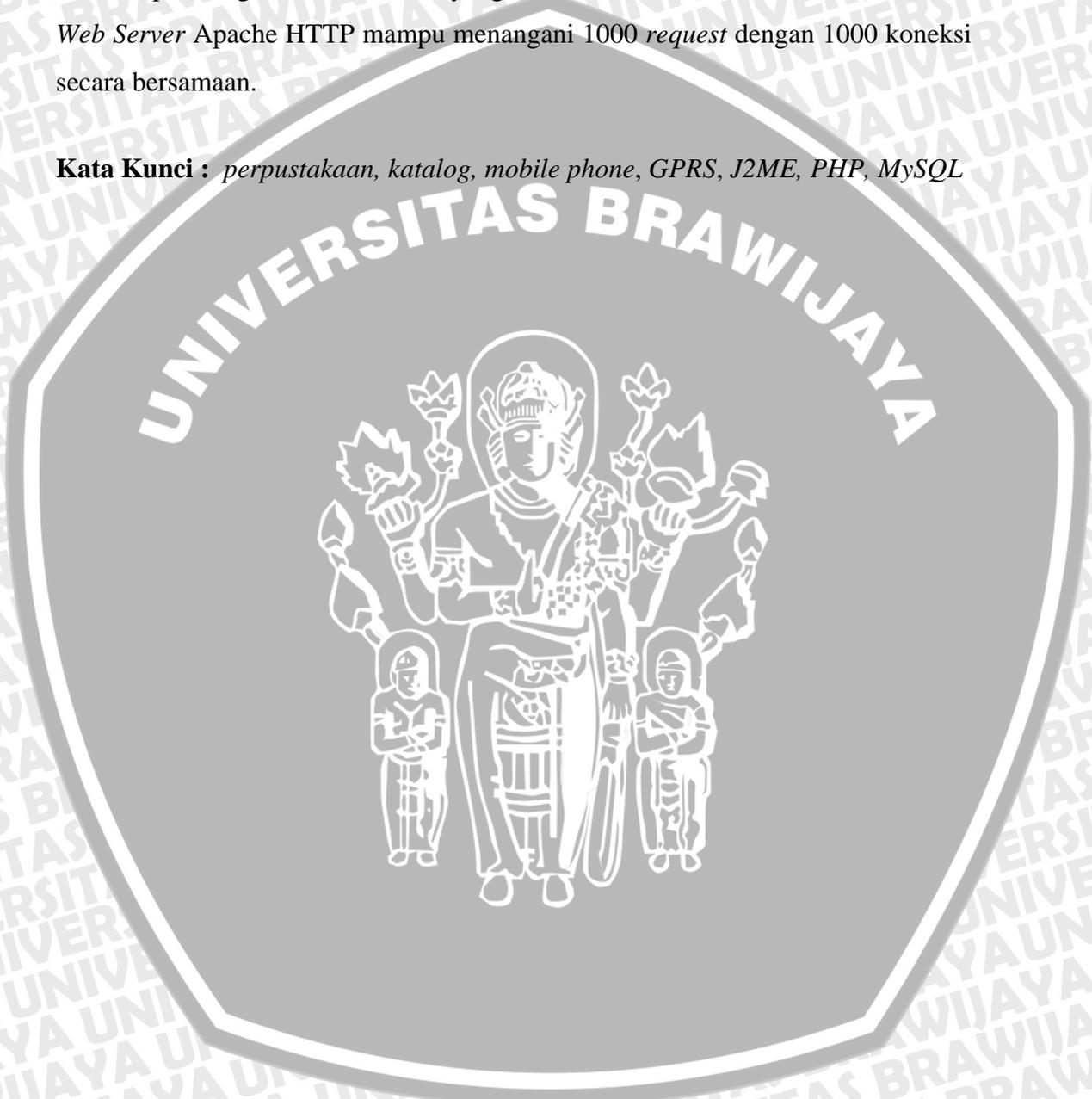
Pentingnya peranan perpustakaan bagi sistem pendidikan mendorong berkembangnya teknologi informasi di bidang perangkat lunak. Kumpulan bahan pustaka di dalam perpustakaan yang berjumlah ratusan bahkan ribuan menyebabkan pengguna mengalami kesulitan sehingga diperlukan suatu perangkat lunak untuk membantu proses pencarian katalog perpustakaan. Perangkat lunak dikembangkan dengan teknologi *Java2 Micro Edition* (J2ME), PHP, dan MySQL. J2ME digunakan untuk menangani tampilan atau antarmuka *user* dan melakukan koneksi ke *Web Server* untuk mengirim dan menerima data dari basis data. PHP digunakan untuk melakukan komunikasi dengan basis data. MySQL digunakan untuk menyimpan data yang berhubungan dengan bahan pustaka di Ruang Baca Teknik Elektro.

Pada skripsi ini memfokuskan pada pengembangan antarmuka *user* dan komunikasi ke basis data. Basis data yang digunakan mengadopsi basis data Ruang Baca Teknik Elektro yang sudah ada. Pengembangan antarmuka *user* di sisi *client* menggunakan pustaka standar yang tersedia dalam J2ME CLDC 1.0 dan MIDP 2.0. Format data *server* yang digunakan adalah format data sederhana (*plain text*) sehingga dapat mempercepat transfer data dari *server* ke *client*. Data di simpan dalam sebuah *String* dan dipisahkan dengan karakter *delimiter* “?”. Aplikasi perangkat lunak yang dikembangkan memberikan kemudahan pada penggunaannya dalam mencari bahan pustaka sesuai dengan keinginan pengguna dan menampilkan informasi bahan pustaka secara lebih detail. Perangkat lunak ini dapat digunakan oleh pengguna melalui *mobile phone* yang terkoneksi dengan GPRS kapanpun dan dimanapun mereka berada serta pengguna tidak tergantung pada sebuah komputer yang terkoneksi dengan internet atau intranet.

Perangkat lunak diuji dengan menggunakan pengujian unit, pengujian integrasi, pengujian validasi, pengujian waktu akses *query*, dan pengujian

performansi *web server*. Hasil pengujian waktu akses *query* menunjukkan waktu rata-rata terlama yang dibutuhkan adalah 0.2389 ms dengan jumlah data 1600. Linieritas kecepatan waktu akses *query* antar data *entry* mendekati 0.02 ms. Hasil pengujian performansi *web server* menunjukkan kinerja Apache HTTP menangani 100 sampai dengan 2000 koneksi yang dilakukan oleh *client* secara bersamaan. *Web Server* Apache HTTP mampu menangani 1000 *request* dengan 1000 koneksi secara bersamaan.

Kata Kunci : *perpustakaan, katalog, mobile phone, GPRS, J2ME, PHP, MySQL*



BAB I PENDAHULUAN

1.1 Latar Belakang

Perpustakaan memiliki peranan penting bagi sistem pendidikan dalam suatu lembaga. Perpustakaan mempunyai fungsi sebagai sumber ilmu pengetahuan dan teknologi serta sebagai penunjang keberhasilan proses belajar mengajar. Pada awalnya proses pencarian koleksi bahan pustaka yang ada di perpustakaan dilakukan dengan cara tradisional. Artinya, mencari bahan pustaka tersebut satu persatu diantara kumpulan buku yang berjumlah ratusan bahkan ribuan. Dalam perkembangannya, informasi katalog perpustakaan ditampilkan dalam media elektronik melalui internet atau intranet karena kemudahannya dalam mendapatkan dan memperbarui informasi.

Telepon seluler (ponsel) biasa dikenal dengan sebutan *handphone* (disingkat HP) merupakan salah satu teknologi yang sangat pesat perkembangannya. Telepon seluler berfungsi sebagai alat komunikasi dan menjadi alat untuk melakukan komunikasi yang lainnya seperti *email*, *internet* dan lain sebagainya.

Salah satu alternatif pengaksesan informasi melalui media elektronik adalah mengembangkan aplikasi sistem katalog perpustakaan berbasis *mobile*. Aplikasi yang akan dibuat ini adalah aplikasi perangkat lunak yang memberikan kemudahan pada penggunaanya dalam berinteraksi dan mencari informasi daftar semua buku yang ada di Ruang Baca Teknik Elektro Universitas Brawijaya. Pengguna dapat mengakses aplikasi ini melalui telepon selular kapanpun dan dimanapun mereka berada serta pengguna tidak lagi tergantung pada sebuah komputer yang terkoneksi dengan internet atau intranet.

Pengembangan sebuah aplikasi sistem katalog perpustakaan diperlukan basis data dan bahasa pemrograman. Basis data digunakan untuk menyimpan data-data yang digunakan dalam membangun sistem katalog perpustakaan. Bahasa pemrograman untuk membuat perangkat lunak sistem katalog perpustakaan dan antarmuka *user* untuk memudahkan pengguna dalam mencari dan melihat hasil

pencarian. Basis data dibuat dengan menggunakan MySQL. MySQL mempunyai kelebihan diantaranya memiliki kecepatan yang tinggi dalam menangani sebuah *query*, mudah digunakan, dapat menampung lebih dari 50 juta *record* dan juga sifatnya yang *open source* serta dapat digunakan secara gratis[PRAd-03:3]. Bahasa pemrograman yang dipakai adalah J2ME pada sisi *client* dan PHP pada sisi *server*. J2ME untuk menangani tampilan atau antarmuka *user* sedangkan PHP untuk komunikasi dengan basis data. PHP mempunyai kelebihan diantaranya banyak *web-server* yang telah mendukung dengan konfigurasi yang relatif mudah, tidak melakukan sebuah kompilasi dalam penggunaannya, dan dapat digunakan secara gratis[WIK-08]. Pada skripsi ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman Java yang berjalan pada perangkat *mobile phone* yang dikenal dengan teknologi *Java2 Micro Edition*. Pemilihan Java didasarkan pada keunggulannya yaitu terintegrasi dengan baik sehingga J2ME bisa terhubung dengan *back-end* server dengan mudah. Java merupakan salah satu bahasa pemrograman berorientasi objek sehingga mempermudah dan mempercepat pengembangan sistem, serta multiplatform[MAR-06:10].

1.2 Rumusan Masalah

Dari latar belakang masalah yang sudah dipaparkan di atas maka rumusan masalah pada skripsi ini difokuskan pada :

1. Analisis kebutuhan dari *user* sebagai acuan untuk membangun sistem
2. Perancangan sistem katalog perpustakaan sebagai penghubung antara analisis kebutuhan dan implementasi
3. Implementasi aplikasi sistem Katalog Perpustakaan dengan menggunakan teknologi J2ME pada sisi *client*, PHP pada sisi *server* dan MySQL sebagai *database server*
4. Pengujian sistem dan kebutuhan *user*

1.3 Batasan Masalah

Batasan masalah yang perlu diajukan dalam pengembangan aplikasi perangkat lunak skripsi ini, antara lain:

1. Aplikasi ini dibangun hanya untuk menyediakan layanan pencarian informasi katalog yang ada di RBTE dengan informasi yang tidak jauh beda dengan web RBTE yang ada saat ini dengan layanan utama sebagai berikut:
 - Mendaftar sebagai anggota
 - Login
 - Melihat daftar koleksi
 - Melihat detail koleksi
 - Mengubah profile
 - Mengubah password
 - Logout
2. Informasi yang ditampilkan bukan keseluruhan isi bahan pustaka melainkan hanya list atau daftar bahan pustaka (judul bahan pustaka, nama pengarang, edisi, cetakan, kota terbit, penerbit, tahun terbit, subjek bahasan, ISBN dan lain-lain) yang ada di RBTE
3. Pembahasan difokuskan pada sisi perangkat lunaknya bukan di sisi perangkat keras yang digunakan
4. Pembahasan difokuskan pada pengembangan antarmuka *user* dan komunikasi antara *client* dengan *server* tanpa membahas mengenai perancangan basis data
5. Menggunakan basis data yang sudah ada dalam sistem basis data Ruang Baca Teknik Elektro Universitas Brawijaya
6. Teknologi yang digunakan pada sisi *client* adalah *Java2 Micro Edition (J2ME)*, sedangkan pada sisi *server* menggunakan PHP sebagai bahasa *server side* untuk komunikasi data dari dan ke *server*
7. Koneksi *client* dengan *server* menggunakan protokol HTTP
8. Perangkat keras yang dapat digunakan untuk perangkat lunak ini adalah telepon selular dengan dukungan java, dalam hal ini penulis menggunakan telepon selular dengan dukungan Java MIDP 2.0 dan konfigurasi CLDC sebagai standar karena saat ini masih banyak telepon selular yang mendukung fitur tersebut

9. Tools yang digunakan untuk mengembangkan perangkat lunak ini adalah NetBeans IDE 6.5 yang dilengkapi dengan J2ME WTK 2.5.2 yang dijalankan pada sistem operasi Windows XP Professional
10. Sistem katalog perpustakaan berbasis *mobile* ini hanya berfungsi sebagai pendukung Sistem Informasi Perpustakaan berbasis web yang sudah ada di Teknik Elektro, sehingga keberadaan Sistem Informasi Perpustakaan dalam bentuk *web based* tetap diperlukan untuk mengolah data

1.4 Tujuan

Tujuan akhir dari skripsi ini adalah membangun aplikasi sistem katalog perpustakaan dengan menggunakan teknologi *Java2 Micro Edition (J2ME)*.

1.5 Manfaat

Aplikasi pada skripsi ini dapat dimanfaatkan oleh mahasiswa, dosen, dan lain sebagainya untuk memberikan kemudahan dalam mendapatkan informasi katalog buku atau sejenisnya yang ada di Ruang Baca Teknik Elektro Universitas Brawijaya kapanpun dan dimanapun mereka berada.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

Bab I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, manfaat serta sistematika penulisan.

Bab II Tinjauan Pustaka

Membahas kajian pustaka dan dasar teori dasar yang digunakan untuk menunjang penulisan skripsi ini. Kajian pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan skripsi ini, meliputi penerapan teknologi J2ME untuk pembuatan sistem katalog perpustakaan berbasis *mobile*. Dasar teori yang diperlukan berdasar kajian pustaka untuk penyusunan skripsi ini adalah teknologi Java, J2ME, koneksi HTTP pada J2ME, PHP dan MySQL.

Bab III Metode Penelitian

Memuat metode yang digunakan dalam penulisan yang terdiri dari studi literatur, perancangan dan implementasi, pengujian, serta pengambilan kesimpulan dan saran.

Bab IV Perancangan Perangkat Lunak

Membahas analisis kebutuhan dan perancangan yang sesuai dengan teori yang ada.

Bab V Implementasi

Pada tahap ini dilakukan implementasi untuk merealisasikan perancangan yang telah dibuat sebelumnya.

Bab VI Pengujian Perangkat Lunak

Membahas strategi pengujian, teknik pengujian, kasus uji, serta hasil pengujian.

Bab VII Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian aplikasi, serta saran-saran untuk pengembangan lebih lanjut.



BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi ini.. Kajian pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan skripsi ini, meliputi penerapan teknologi J2ME untuk pembuatan sistem katalog perpustakaan berbasis *mobile*. Dasar teori yang diperlukan berdasar kajian pustaka untuk penyusunan skripsi ini adalah teknologi Java, J2ME, koneksi HTTP pada J2ME, PHP dan MySQL.

2.1 Katalog Perpustakaan

Perpustakaan ialah tempat atau kumpulan buku-buku yang disusun teratur, rapi, dan sistematis.

Katalog merupakan istilah umum yang sering diartikan sebagai suatu daftar barang atau benda yang terdapat pada tempat tertentu[SUHE-07:01].

Dalam kaitannya dengan perpustakaan, katalog adalah daftar bahan pustaka baik berupa buku maupun non-buku seperti majalah, surat kabar, microfilm, slide, dan lain-lain yang dimiliki dan tersimpan pada suatu atau sekelompok perpustakaan. Dalam katalog perpustakaan tercantum informasi-informasi penting dari suatu bahan pustaka yang biasanya dipakai oleh pengunjung perpustakaan sebagai bahan informasi yang menyangkut fisik bahan pustaka, isi, ataupun informasi-informasi lainnya seperti judul bahan pustaka, nama pengarang, edisi, cetakan, kota terbit, penerbit, tahun terbit, subjek bahasan, ISBN dan lain-lain[SUHE-07:01].

Fungsi dari katalog perpustakaan adalah alat atau media untuk mencari dan menemukan bahan pustaka yang dibutuhkan oleh pengunjung perpustakaan secara cepat, tepat, dan akurat[SUHE-07:02].

2.2 Teknologi Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari Java *platform*

(generasi awalnya adalah Java Development Kit). Java berdiri diatas sebuah mesin *interpreter* yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca *bytecode* dalam file `.class` dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM[SHA-06:1].

Platform Java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah kompiler, *debugger* dan kakas lain yang dipaket dalam Java development Kit (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari *platform* Java. Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi `.java` harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi[SHA-06:2].

2.2.1 Java2 Standard Edition (J2SE)

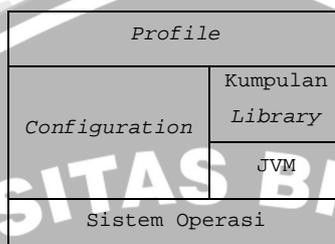
J2SE adalah inti dari bahasa pemrograman Java. JDK (*Java Development Kit*) adalah salah satu *tool* dari J2SE untuk mengkompilasi dan menjalankan program Java. Di dalamnya terdapat *tool* untuk mengkompilasi program Java dan JRE (*Java Runtime Environment*). Tool J2SE yang salah satunya adalah `jdk 1.5` merupakan *tool open source* dari Sun [SHA-06:4].

2.2.2 Java 2 Micro Edition (J2ME)

Java2 Micro Edition atau yang biasa disebut dengan J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer dekstop yang biasanya lebih kecil dibandingkan perangkat komputer dekstop.

J2ME biasa digunakan pada telepon selular, pager, personal digital assistants (PDA's) dan sejenisnya[SHA-06:5].

J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE[SHA-06:5]. Arsitektur J2ME dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur J2ME
Sumber: [SHA-06:5]

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya adalah pada ukuran aplikasi, karena ukuran memori pada ponsel sangat terbatas[SHA-06:6].

2.2.2.1 J2ME Configuration

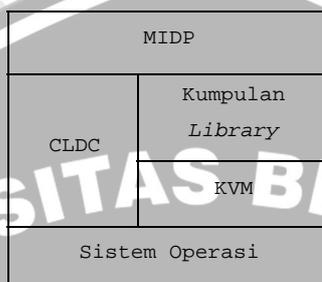
J2ME Configuration mendefinisikan lingkungan kerja J2ME *runtime*. Oleh karena setiap *handheld devices* memiliki fitur-fitur yang berbeda-beda, *J2ME Configuration* ini dirancang, yakni menyediakan *library* standar yang mengimplementasikan fitur standar dari sebuah *handheld devices*.

Dalam J2ME telah didefinisikan dua buah konfigurasi yaitu CLDC (Connected Limited Device Configuration) untuk perangkat kecil dan CDC (Connected Device Configuration) untuk perangkat yang lebih besar[SHA-06:06].

a. *Connected Limited Device Configuration* (CLDC)

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon selular, *pager*, dan PDA.

Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari package, kelas, dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*) [SHA-06:08]. Posisi CLDC pada arsitektur J2ME dapat dilihat pada Gambar 2.2.



Gambar 2.2 Posisi CLDC pada J2ME
Sumber: [SHA-06:07]

b. Connected Device Configuration (CDC)

CDC atau *Connected Device Configuration* adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada Java yang memiliki standardisasi. CDC terdiri dari *virtual machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform*. Berikut adalah perbandingan antara CLDC dan CDC [SHA-06:08]:

Tabel 2.1 Perbandingan antara CLDC dan CDC

CLDC	CDC
Mengimplementasikan sebagian dari J2SE	Mengimplementasikan seluruh fitur J2SE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM
Digunakan pada perangkat genggam (handphone, PDA, twoway pager) dengan memori terbatas (160-512 KB)	Digunakan pada perangkat genggam (internet TV, Nokia Communicator, car TV) dengan memori minimal 2MB
Prosesor : 16/32 bit	Prosesor : 32 bit

Sumber: [SHA-06:08]

2.2.2.2 J2ME Profile

Jika J2ME Configuration menyediakan librari-librari Java untuk implementasi fitur-fitur standar dari sebuah handhled devices, J2ME profile menyediakan implementasi-implementasi tambahan yang sangat spesifik dari sebuah *handhled devices*. Sebagai analogi, sebuah handphone



Nokia/Siemens/Motorola memiliki kemampuan untuk menelepon ke suatu nomor *handphone* lain karena ini merupakan kemampuan standar dari sebuah *handphone*. Namun, masing-masing *handphone* tentulah ada fitur-fitur tersendiri, misalnya Siemens bisa menyimpan file MP3, Nokia memiliki game tersendiri dan lain sebagainya. Kemampuan standar itulah yang diimplementasikan oleh J2ME Configuration, sedangkan kemampuan/fitur lain yang sangat bergantung pada jenis perangkat *handhled* yang digunakan akan diimplementasikan oleh J2ME Profile.

Ada lima kategori J2ME Profile saat ini, yakni:

- Mobile Information Device Profile (MIDP)
- Foundation Profile (FP)
- Personal Profile
- RMI Profile
- Personal Digital Assistance Profile

MIDP atau *Mobile Information Device Profile* adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan, dan penyimpanan persisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan MIDP 2.0 dibanding MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan tone, tone sequence, dan file WAV walaupun tanpa adanya Mobile Media API (MMAPI). Posisi MIDP pada arsitektur J2ME dapat dilihat pada Gambar 2.2. Berikut adalah perbandingan MIDP 1.0 dan MIDP 2.0[SHA-06:09]:

Tabel 2.2 Perbandingan MIDP 1.0 dan MIDP 2.0

Spesifikasi	MIDP 1.0	MIDP 2.0
Display	96 x 54	96 x 54
Kedalaman Display	1-bit	1-bit
Bentuk piksel (rasio aspek)	Mendekati 1 : 1	Mendekati 1 : 1
Input	Keyboard dan touch screen	Keyboard dan touch screen
Memori	128 KB memori non-volatile untuk komponen MIDP 8 KB memori non-volatile untuk data persisten yang dibuat oleh aplikasi 32 KB memori volatile untuk JRE	256 KB memori non-volatile untuk komponen MIDP 8 KB memori non-volatile untuk data persisten yang dibuat oleh aplikasi 128 KB memori volatile untuk JRE
Jaringan	Dua arah, tanpa kabel (wireless)	Dua arah, tanpa kabel (wireless)
Library J2ME yang bukan merupakan library J2SE	javax.microedition.lcdui, javax.microedition.midlet,	javax.microedition.lcdui, javax.microedition.midlet,

	javax.microedition.rms	javax.microedition.rms, javax.microedition.media, javax.microedition.pki
Multimedia	-	Memiliki kemampuan untuk memainkan file multimedia (suara dan video)

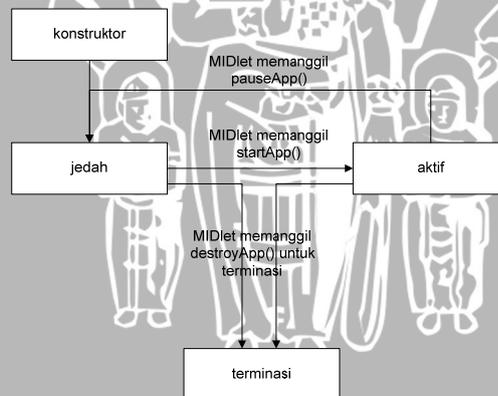
Sumber: [SHA-06:09]

2.2.2.3 MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari klas `javax.microedition.midlet` yang didefinisikan pada MIDP. MIDlet berupa sebuah klas abstrak yang merupakan subklas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.

a. Daur hidup MIDlet

MIDlet terdiri dari beberapa metode yang harus ada yaitu `constructor()`, `protected void startApp() throws MIDletStateChangeException`, `protected void pauseApp()`, `protected void destroyApp(Boolean unconditional) throws MIDletStateChangeException`. Alur hidup MIDlet dapat dilihat pada Gambar 2.3.



Gambar 2.3 Daur hidup MIDlet

Sumber : [SHA-06:12]

Ketika MIDlet dijalankan maka akan diinisialisasi dengan kondisi *pause* dan dijalankan `pauseApp()`, kondisi berikutnya adalah fungsi MIDlet dijalankan, yaitu pada `startApp()`. Metode yang ada tersebut diimplementasikan sebagai `protected`, hal ini dimaksudkan agar MIDlet lain tidak dapat memanggil metode tersebut. Pada saat pemakai keluar dari MIDlet, maka metode `destroyApp()` akan dijalankan sebelum MIDlet benar-benar tidak berjalan lagi. Metode

`notifyDestroyed()` akan dipanggil sebelum MIDlet benar-benar tidak berjalan lagi, `destroyApp()` akan memanggil `notifyDestroyed()`, dan `notifyDestroyed()` akan memberitahu *platform* untuk menterminasi MIDlet dan membersihkan semua sumber daya yang mengacu pada MIDlet.

Dalam implementasinya, MIDlet memiliki struktur direktori sebagai berikut:

- `src`
Menyimpan *source code* untuk MIDlet dan klas lain yang diperlukan.
- `res`
menyimpan sumber daya yang dibutuhkan oleh MIDlet, seperti misalnya gambar icon
- `lib`
menyimpan file JAR atau ZIP yang berisi *library* tambahan yang dibutuhkan MIDlet
- `bin`
menyimpan file JAR, JAD, dan file manifest yang berisi muatan komponen MIDlet

b. Atribut MIDlet

MIDlet juga memiliki atribut *package* sebagai berikut:

Tabel 2.3 Atribut MIDlet

Nama Atribut	Nilai dan Fungsi
MIDlet-Name	Nama MIDlet yang juga sebagai nama untuk file JAR-nya. Misalnya <code>WirelessTrader</code>
MIDlet-Version	Nomor versi dari MIDlet yang juga sebagai nomor versi pada file JAR-nya. Misalnya <code>1.0.1</code>
MIDlet-Vendor	Nama <i>provider</i> MIDlet. Misalnya <code>Wireless Java Inc.</code>
MIDlet- <i>n</i>	Atribut yang mendeskripsikan MIDlet. Nilai <i>n</i> diganti dengan nilai numerik dimulai dari angka 1. format nilai dari ini terkait dengan atribut yang
MicroEdition-Profile	Versi dari spesifikasi MIDlet yang dapat berjalan. Dapat lebih dari satu versi yang dipisah dengan spasi, contoh dari atribut ini adalah <code>MIDP-2.0</code>
MicroEdition-Configuration	Konfigurasi J2ME yang dibutuhkan untuk MIDlet.
MIDlet-Description	Deskripsi MIDlet. Misalnya MIDlet untuk koneksi internet.
MIDlet-Icon	Icon yang digunakan MIDlet. Misalnya <code>wireless.png</code>
MIDlet-Info-URL	URL dari file yang berisi informasi mengenai MIDlet. Misalnya <code>http://www.wireless.com/info.html</code>
MIDlet-Data-Size	Nilai minimum dari besarnya tempat penyimpanan persisten yang dibutuhkan (dalam <i>byte</i>) tidak termasuk yang dibutuhkan untuk menginstal aplikasi. Misalnya
MIDlet-JAR-URL	URL dari file JAR. Misalnya <code>http://www.wireless.com/MIDlet.jar</code>
MIDlet-JAR-Size	Ukuran file JAR dalam <i>byte</i>

MIDlet-Install-Notify	Sebuah URL yang digunakan untuk melaporkan sukses atau gagalnya penginstalan MIDlet dari <i>Remote Server</i>
MIDlet-Delete-Confirm	Pesan yang ditampilkan ke pemakai sebelum MIDlet dihapus dari perangkat dimana MIDlet diinstal
MIDlet-specification	Pengembang MIDlet dapat menyediakan konfigurasi minimum untuk MIDlet dengan memasukkan atribut ini.

Sumber : [SHA-06:13]

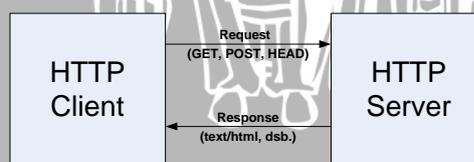
2.2.2.4 Koneksi HTTP pada J2ME

J2ME menyediakan beberapa cara komunikasi antara peranti *mobile* dengan *server*, antara lain[MAR-06:29]:

- Dengan menggunakan email melalui koneksi socket.
- Dengan menggunakan sistem pesan, seperti: SMS dan lain sebagainya.
- Dengan menggunakan koneksi HTTP
- Dengan menggunakan media koneksi lainnya, contohnya: infra-red, kabel data, dan bluetooth.

Dari keempat koneksi diatas, koneksi HTTP adalah koneksi yang banyak digunakan dan menjadi standar koneksi MIDP. Untuk koneksi email pun, sebaiknya menggunakan *proxy* HTTP karena tidak semua peranti mobile MIDP menyediakan fasilitas koneksi socket (Spesifikasi MIDP 2.0 menjelaskan bahwa koneksi socket tidak wajib diimplementasikan) [MAR-06:29].

Mekanisme koneksi HTTP secara umum dan sederhana bisa digambarkan dengan dua kata: PERMINTAAN dan RESPON (Request-Response). Jika client meminta sesuatu, maka server HTTP akan memberi respon sesuai dengan apa yang diminta[MAR-06:30].



Gambar 2.4 Mekanisme HTTP

Sumber: [MAR-06:30]

Mekanisme koneksi HTTP pada MIDP melalui tiga tahap status, yaitu: SETUP, CONNECTED, CLOSED. Parameter "Request" diatur pada status SETUP. Pada status CONNECTED, diharapkan kembalinya suatu respon dari HTTP server. Setelah proses-proses dilakukan, maka koneksi tersebut harus ditutup pada status CLOSED. MIDP menyediakan *interface* `HttpConnection` untuk menangani koneksi HTTP. Objek `HttpConnection` akan berisi hasil

pemanggilan metode `open()` dari kelas statis `Connector`, yang telah di-casting menjadi `HttpConnection`[MAR-06:30].

Interface `HttpConnection` menyediakan metode-metode untuk mengatur jenis "Request" dan properti-properti "Request" pada status SETUP. Metode-metode tersebut adalah[MAR-06:30]:

- Metode `setRequestMethod()` untuk mengatur jenis "Request"(GET, POST, HEAD).
- Metode `setRequestProperty()` untuk mengatur property-properti "Request", seperti: User-Agent, Content-Language, dan sebagainya.

Untuk memproses "Request" dan menerima "Response", interface `HttpConnection` menyediakan dua metode, yaitu[MAR-06:31]:

- Metode `openOutputStream()` untuk mengirimkan "Request" yang dilengkapi dengan perintah-perintah HTTP, misalnya; LIST. Bila tidak ada perintah yang dikirimkan, metode ini tidak perlu dieksekusi.
- Metode `openInputStream()` untuk menerima isi "Respon".

2.2.2.5 J2ME dan WAP

Untuk membuat aplikasi berbasis wireless terdapat beberapa pilihan teknologi, antara lain WAP dan J2ME. WAP dengan bahasa WML bekerja baik pada aplikasi berbasis *server*, *text centric application* seperti *weather report*, *stock quotes*, dll. Namun terdapat kekurangan lain seperti [AID-07]:

- a. *Small display dan Limited User Input Facility*. Keterbatasan dalam tampilan (seperti animasi)
- b. Tidak efektif untuk komputasi lokal dan penyimpanan dengan koneksi ke jaringan
- c. *Security Issue* yang menimpa WAP Gateway. Dalam WAP, device *retrieve* dan *men-display* data dengan *microbrowser* dan dibutuhkan suatu gateway untuk jembatan antara jaringan internet dan wireless. Hal ini merupakan hal potensial dalam gangguan keamanan data (konversi protokol dari protokol internet menjadi protocol wireless)
- d. *Must Always Connected*. Aplikasi berbasis browser tidak mampu bekerja pada kondisi yang tidak terjangkau baik oleh sinyal provider

Teknologi Java (J2ME) memiliki kelebihan antara lain[AID-07]:

- a. *Enhances User Experiences*. Mendukung grafis yang rumit dan handal dalam komputasi lokal
- b. *Running Every Where, Any Where, Over Any Device*. J2ME ditujukan untuk cakupan user dan device yang luas, dibangun diatas JVM yang menghasilkan kode *portable (Cross Platform Compatibility)*
- c. *Save Network Delivery* dan *End to End Security HTTP*. J2ME menyediakan SSL untuk melakukan enkripsi terhadap data yang dikirimkan ke dalam jaringan melalui *HTTP Connection*
- d. *Disconnected Acces dan Synchronization*. *Device* tetap dapat menjalankan aplikasi walaupun sudah berada *out of coverage area* atau *terdisconnection*
- e. *Dinamic Delivery of Application and Services*. Dapat di download secara OTA (*Over The Air*) dan running dalam berbagai device (karena memiliki JVM)
- f. *Scalability and Performance*. Berkurangnya koneksi *device* ke server karena aplikasi berbasis J2ME dapat dijalankan secara *offline (stand alone)* maupun *on line* sehingga meningkatkan performansi server

Namun, J2ME dan WAP merupakan teknologi yang berbeda walaupun sama-sama berjalan diatas teknologi wireless. WAP menyediakan pilihan teknologi browser yang dapat digunakan untuk *surfing* dan download aplikasi J2ME[AID-07].

2.3 PHP

PHP merupakan bahasa *scripting server-side*, di mana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*-lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirimkan kepada *client* yang melakukan permintaan[PRAa-04:76].

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web server* Apache. Namun, belakangan PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami[KADa-03:02].

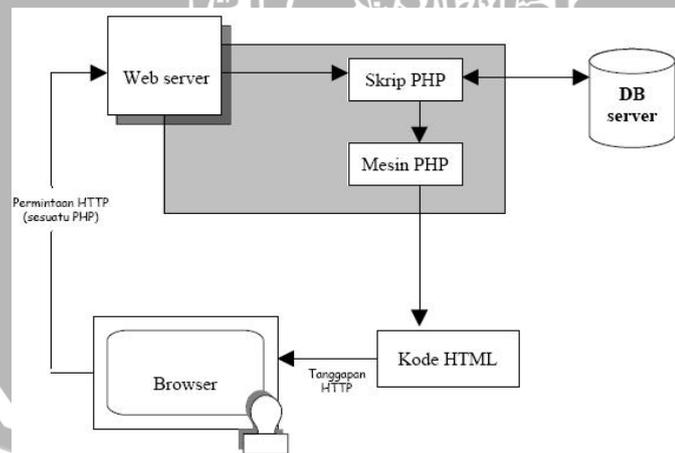
Sebagian perintah PHP berasal dari bahasa C, Java dan Perl dengan beberapa tambahan fungsi khusus. Penulisan skrip PHP dapat dilakukan dengan

menggunakan dua cara, yaitu dengan *embedded script* dan *non-embedded script*. *Embedded script* adalah meletakkan skrip PHP di antara tag-tag HTML murni, sedangkan *non-embedded script* adalah penulisan skrip murni PHP, di mana tag-tag HTML diletakkan didalamnya[PRAa-04:76].

Untuk dapat lebih memahami tentang PHP, maka akan dibahas lebih lanjut mengenai konsep kerja PHP, variabel pada PHP, *Script* PHP, *Style* kode PHP dan kelebihan dari PHP itu sendiri.

2.3.1 Konsep Kerja PHP

Model kerja PHP diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari berkas yang diminta dan apabila sudah didapatkan maka *web server* akan mengirimkannya ke mesin PHP [KADa-03:5]. Mesin inilah yang kemudian akan memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya *web server* akan menyampaikan hasil tersebut ke *client*. Skema konsep kerja PHP ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Skema konsep kerja PHP

Sumber: [KADa-03:5]

2.3.2 Script PHP

Dalam bukunya, menurut Prasetyo [03:4-5] terdapat dua cara yang sering digunakan untuk menuliskan *script* atau program PHP, yaitu:

1. Embedded Script

Cara ini dilakukan dengan meletakkan *script* PHP di antara *tag-tag* HTML.

Berikut contoh penggunaannya:

```
<html>
<head>
<title>embedded script</title>
</head>
<body>
<?
echo "Ini adalah contoh embedded script";
?>
</body>
</html>
```

2. Non Embedded Script

Non-embedded script merupakan pembuatan program murni PHP, dimana *tag-tag* HTML yang diletakkan didalamnya. Berikut contoh penggunaannya:

```
<?
echo "<html>";
echo "<head>";
echo "<title>non embedded script</title>";
echo "</head>";
echo "<body>";
echo "Ini adalah contoh non embedded script";
echo "</body>";
echo "</html>";
?>
```

2.3.3 Style Kode PHP

PHP memberikan keleluasaan ketika menulis kode program, dengan catatan tetap mengacu pada ketentuan yang telah ditetapkan. Tag memiliki arti penting untuk menyatakan bahwa kode yang telah dibuat merupakan kode PHP. Berkaitan dengan hal ini, tersedia empat corak mode penulisan tag yang mengidentifikasi program PHP[PRAc-05:3-6].

1. *Style* XML

Corak pertama atau sering disebut model standar memiliki kemiripan dengan penulisan dokumen XML (*Extensible Markup Language*), yakni diawali dengan `<?php` dan diakhiri dengan `?>`. Adapun dikatakan sebagai *style* standar, karena tidak memerlukan konfigurasi tambahan, dan

penggunaannya sangat disarankan. Contoh implementasinya dalam program adalah sebagai berikut:

```
<?php
echo "Style XML";
?>
```

2. *Style SCRIPT*

Cara penulisannya tidak jauh berbeda ketika membuat kode program JavaScript atau VBScript. Penulisan style ini diawali dengan `<SCRIPT LANGUAGE = 'PHP'>` dan diakhiri dengan `</SCRIPT>`. Berikut contoh penggunaannya:

```
<SCRIPT LANGUAGE = 'PHP'>
    echo "Style Javascript";
</SCRIPT>
```

3. *Short Style*

Style ini cukup praktis untuk digunakan karena hanya diawali dengan `<?` dan diakhiri dengan `?>`.

```
<?
echo "Short style";
?>
```

Untuk menggunakannya diperlukan konfigurasi tambahan. Pada file `php.ini`, di baris directive bernama `short_open_tag` berikan nilai `On` atau `True`.

```
Short_open_tag = On
```

PHP menyarankan agar menghindari penggunaan short style ketika mengembangkan aplikasi atau library yang dimaksudkan untuk didistribusikan ulang, karena adanya kemungkinan server yang tidak mendukung style ini terlebih ketika kita tidak mampu melakukan pengontrolan secara langsung.

4. *Style ASP*

Bentuk terakhir tag terakhir ini adalah seperti ASP (*Active Server Pages*), yakni diawali dengan `<%` dan diakhiri dengan `%>`. Untuk menggunakannya diperlukan sedikit konfigurasi dalam `php.ini`.

```
;Allow ASP-Style <% %> tags.
asp_tags = On
```

Contoh penulisannya sebagai berikut:

```
<%
echo "Style ASP";
%>
```

2.3.4 Variabel pada PHP

Variabel digunakan dalam program untuk menyimpan suatu nilai dan nilai yang ada di dalamnya dapat diubah sewaktu-waktu. Berikut contoh penggunaannya[KADa-03:12-14]:

```
<?php
    $nama = "Abdul Kadir";
    printf("Halo, %s", $nama);
?>
```

Pada kode diatas, **nama** adalah nama variabel. Pada program PHP, variabel selalu ditulis dengan diawali tanda dolar (\$). Semua nama variabel bersifat *case-sensitive*. Nama variabel yang valid harus diawali oleh huruf atau garis bawah yang diikuti oleh sejumlah huruf, angka, atau garis bawah.

PHP dapat digunakan untuk mengirim atau melewati suatu variabel (*pass*) diantara beberapa halaman yang berbeda dalam aplikasi kita melalui beberapa cara, yakni melalui URL dan *session*. Selain itu, PHP juga dapat digunakan untuk mengirim suatu variabel atau nilai yang ditangani melalui form (dientri oleh pemakai melalui keyboard) dan kemudian nilai ini dikirim ke skrip yang lain.

2.3.5 Kelebihan PHP

Kelebihan PHP dari bahasa pemrograman lain[WIK-08]:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan Apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan dalam berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah *system*.

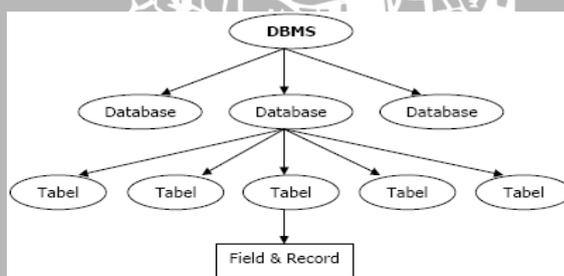
2.4 Basis Data

Basis data (atau *database*) adalah kumpulan informasi yang disimpan secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer[SOL-03:1].

Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan user (perngguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada[SOL-03:1].

Sedangkan RDBMS atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Disamping RDBMS, terdapat jenis DBMS lain, misalnya *Hierarchy DBMS*, *Object Oriented DBMS*, dsb[SOL-03:1].

Dalam konsep database, urutan atau hierarki database sangatlah penting. Urutan atau hierarki database digambarkan dalam gambar sebagai berikut[SOL-03:1]:



Gambar 2.1 Hierarki database

Sumber: [SOL-03:1]

2.4.1 MySQL

MySQL adalah suatu sistem manajemen database relasional atau *Relationship Database Management System* (RDBMS). Suatu database relasional menyimpan data dalam tabel yang terpisah. Hal ini menambah kecepatan dan fleksibilitas. Tabel-tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu

permintaan. SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses database[SUW-08:16].

MySQL adalah perangkat lunak *open source*. *Open source* berarti dapat digunakan dan dimodifikasi oleh siapa saja. Semua orang dapat *men-download* MySQL dari internet dan menggunakannya secara gratis[SUW-08:16].

2.4.1.1 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan antara lain [PRAd-03:3]:

- *Portability*
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Solaris* dan lain sebagainya.
- *Open Source*
MySQL didistribusikan secara *open source* sehingga dapat digunakan secara gratis.
- *Multi-User*
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance Tuning*
MySQL memiliki kecepatan yang tinggi dalam menangani sebuah *Query*.
- *Column Types*
MySQL mendukung tipe kolom yang banyak di antaranya *signed/unsigned integer*, *float*, *double*, *char*, *varchar*, *text*, *blob*, *date*, *time*, *datetime*, *timestamp*, *year*, *set*, *enum*.
- *Command dan Functions*
MySQL mendukung penuh perintah *select* dan *where* dalam *query*.
- *Security*
MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan system perizinan yang mendetail serta password terenkripsi.

- *Scalability dan Limits*
MySQL mampu menangani basis data dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.
- *Interface*
MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- *Connectivity*
MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix soket (Unix), atau Named Pipes (Windows NT)
- *Clients dan Tools*
MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database.

2.4.2 SQL (Structured Query Language)

Pada dasarnya SQL dikelompokkan menjadi 5 kelompok, antara lain[KADb-03:106-108]:

1. DDL (*Data Definition Language*)

DDL merupakan kelompok perintah yang digunakan untuk membuat, memanipulasi, atau menghapus *schema* yang ada di dalam database. Bahasa yang terdapat di dalam DDL hanya berhubungan dengan *schema* database, tidak berhubungan dengan data yang ada di dalamnya. *Schema* adalah struktur yang dimiliki oleh sebuah database. Yang termasuk kelompok DDL ini:

- CREATE* : untuk menciptakan table atau indeks
- ALTER* : untuk mengubah struktur table
- DROP* : untuk menghapus table atau indeks

2. DML (*Data Manipulation Language*)

DML adalah kelompok perintah yang digunakan untuk memanipulasi data yang terdapat di dalam database seperti menambah data, menghapus data, dan lain-lain. Yang termasuk DML adalah :

- SELECT* : untuk memilih data

- DELETE* : untuk menghapus data
- INSERT* : untuk menambah data
- UPDATE* : untuk mengubah data

3. DCL (*Data Control Language*)

DCL berisi perintah-perintah untuk mengendalikan pengaksesan data dan mengatur user MySQL. Yang termasuk DCL adalah :

- GRANT* : untuk memberikan kendali pada pengaksesan data
- REVOKE* : untuk mencabut kemampuan pengaksesan data
- LOCK TABLE* : untuk mengunci table

4. Pengendali transaksi

Adalah perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi. Yang termasuk kelompok ini adalah:

- COMMIT* : untuk menyetujui rangkaian perintah yang berhubungan erat (disebut transaksi) yang telah berhasil dilakukan
- ROLLBACK* : untuk membatalkan transaksi yang dilakukan karena adanya kesalahan atau kegagalan pada salah satu rangkaian perintah.

5. Pengendali programatik

Mencakup pernyataan-pernyataan yang berhubungan dengan pemanfaatan SQL dalam bahasa lain (SQL yang dilekatkan). Yang termasuk dalam kelompok ini adalah :

- CLOSE* : untuk menutup kursor (pointer yang menunjuk ke tabel)
- DECLARE* : untuk mendeklarasikan kursor
- FETCH* : untuk mengambil nilai baris berikutnya
- OPEN* : untuk membuka kursor

BAB III

METODE PENELITIAN

Pada bab ini dijelaskan mengenai langkah-langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan bagaimana aplikasi dapat dikembangkan lebih lanjut.

3.1 Studi Literatur

Berupa kajian pustaka terhadap sumber-sumber bacaan yang relevan yang dapat menunjang dalam proses pengembangan sistem. Kajian terhadap berbagai literatur ini ditujukan untuk mendapatkan landasan teori yang diperlukan, landasan teori yang dimaksud antara lain yaitu :

- a. Katalog Perpustakaan
- b. Teknologi Java
 - *Java2 Standard Editon (J2SE)*
 - *Java2 Micro Edition (J2ME)*
- c. PHP
 - Konsep Kerja PHP
 - Script PHP
 - Style Kode PHP
 - Variabel pada PHP
 - Kelebihan PHP
- d. Basis Data
 - MySQL
 - SQL (*Structured Query Language*)

Sumber bacaan tersebut dapat berupa *text book*, buku panduan pemrograman, tugas akhir, *paper*, tutorial pemrograman, dan sumber bacaan *softcopy* lain yang didapatkan dari *internet*. Sumber-sumber bacaan tersebut diletakkan pada daftar pustaka.

3.2 Perancangan

Tahap ini terdiri dari 2 bagian, yaitu analisis kebutuhan (*Requirement Analysis*) dan perancangan (*Design*).

3.2.1 Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini ditujukan untuk mendapatkan kebutuhan apa saja yang diperlukan dan yang ingin didapatkan dari sistem ini, dan kemudian menganalisisnya. Metode analisis yang digunakan adalah *Object Oriented Analysis* dengan menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Analisis kebutuhan ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memecahkan permasalahan yang dihadapi oleh pengguna. Pada analisis kebutuhan ini diawali dengan identifikasi aktor-aktor, penjabaran daftar kebutuhan dan kemudian memodelkannya ke dalam suatu diagram *use case*. Setiap *use case* dapat dijelaskan lebih lengkap dalam skenario *use case*. Skenario *use case* menjelaskan secara detail interaksi dari masing-masing *use case* yang terdapat pada diagram *use case*.

3.2.2 Perancangan (*Design*)

Perancangan aplikasi dilakukan untuk mempermudah implementasi, analisis algoritma dan pengujian. Perancangan aplikasi berdasarkan *Object Oriented Design* yaitu menggunakan pemodelan UML (*Unified Modeling Language*). Pada tahap ini dibuat suatu diagram pemodelan sistem aplikasi secara keseluruhan yang sesuai dengan analisis kebutuhan sistem yang digambarkan dalam diagram *use-case*. Perancangan dilakukan dengan membuat *Sequence Diagram* yang memperlihatkan interaksi pengguna dengan objek-objek dalam aplikasi dan *Class Diagram* untuk melihat hubungan antar kelas, objek dan *interface* yang ada di dalam aplikasi.

3.3 Implementasi

Implementasi aplikasi dilakukan dengan mengacu kepada hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan aplikasi yang

telah dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi tiap klas pada file program, implementasi algoritma, implementasi antarmuka aplikasi dan beberapa kendala dalam implementasi. Implementasi perangkat lunak dilakukan dengan menggunakan dua bahasa pemrograman berorientasi objek. Bahasa pemrograman yang dipakai adalah J2ME pada sisi *client* dan PHP pada sisi *server*. J2ME untuk menangani tampilan atau antarmuka *user* sedangkan PHP untuk komunikasi dengan basis data. Implementasi basis data menggunakan MySQL.

3.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak pada skripsi ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya.

Strategi pengujian yang digunakan yaitu pengujian unit, pengujian integrasi, dan pengujian validasi. Dan metode pengujian yang digunakan adalah metode pengujian *white box* dan *black box*. Pengujian dimulai dari pengujian unit, kemudian dilanjutkan dengan pengujian integrasi, dan berakhir pada pengujian validasi. Pada tahap pengujian unit digunakan metode pengujian *white box* dengan teknik *basis path*, sedangkan pada pengujian integrasi digunakan teknik *white box*, dilakukan dengan mengambil input klas yang lolos pada pengujian unit, kemudian klas yang lolos pada pengujian unit diintegrasikan melalui klas kontrol untuk diujikan. Pada tahap pengujian validasi digunakan teknik *black-box*.

3.5 Pengambilan Kesimpulan dan Saran

Pada tahap ini diambil kesimpulan dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap yang paling akhir adalah pengambilan saran terhadap penelitian yang selanjutnya sehingga bisa menyempurnakan kekurangan yang ada dan mengembangkan pada tingkat pokok kajian yang lebih lanjut.

BAB IV

PERANCANGAN PERANGKAT LUNAK

Bab ini membahas mengenai perancangan perangkat lunak, yang meliputi dua tahap. Pada tahap pertama dilakukan proses analisis kebutuhan (*requirement analysis*), sedangkan tahap yang kedua adalah proses perancangan. Pada tahap analisis kebutuhan digunakan pemodelan dengan diagram *use case*. Pada proses perancangan perangkat lunak digunakan pemodelan dalam bentuk diagram klas (*class diagram*), dan diagram urutan (*sequence diagram*).

4.1 Analisis Kebutuhan (*Requirements Analysis*)

Proses analisis ini mengambil acuan dari hasil pengumpulan, pemahaman dan penetapan kebutuhan-kebutuhan (*requirements*) yang ingin didapatkan oleh pengguna, dan harus mampu disediakan oleh sistem perangkat lunak. Pada analisis kebutuhan ini diawali dengan identifikasi aktor-aktor, penjabaran daftar kebutuhan dan kemudian memodelkannya ke dalam suatu diagram *use case*. Analisis kebutuhan ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memecahkan permasalahan yang dihadapi oleh pengguna.

4.1.1 Identifikasi Aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor-aktor yang akan berinteraksi dengan sistem. Tabel 4.1 memperlihatkan dua buah aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Deskripsi Aktor

Aktor	Deskripsi Aktor
<i>Guest</i>	<i>Guest</i> merupakan semua aktor pengguna yang belum terautentifikasi melalui <i>login</i> , hanya registrasi dan <i>login</i> saja yang dapat dilakukan oleh aktor ini.
<i>Member</i>	<i>Member</i> merupakan aktor pengguna yang sudah terautentikasi melalui <i>login</i> dan dapat menggunakan fasilitas pencarian informasi katalog perpustakaan.

Sumber : Analisis Kebutuhan

4.1.2 Daftar Kebutuhan

Daftar kebutuhan merupakan daftar yang menguraikan kebutuhan-kebutuhan *user* yang harus disediakan oleh perangkat lunak baik kebutuhan fungsional maupun non fungsional. Daftar kebutuhan fungsional sistem disertai dengan nama *use case* yang merepresentasikan fungsionalitas dari kebutuhan tersebut. Daftar kebutuhan fungsional dan kebutuhan non-fungsional tersebut ditunjukkan pada Tabel 4.2.

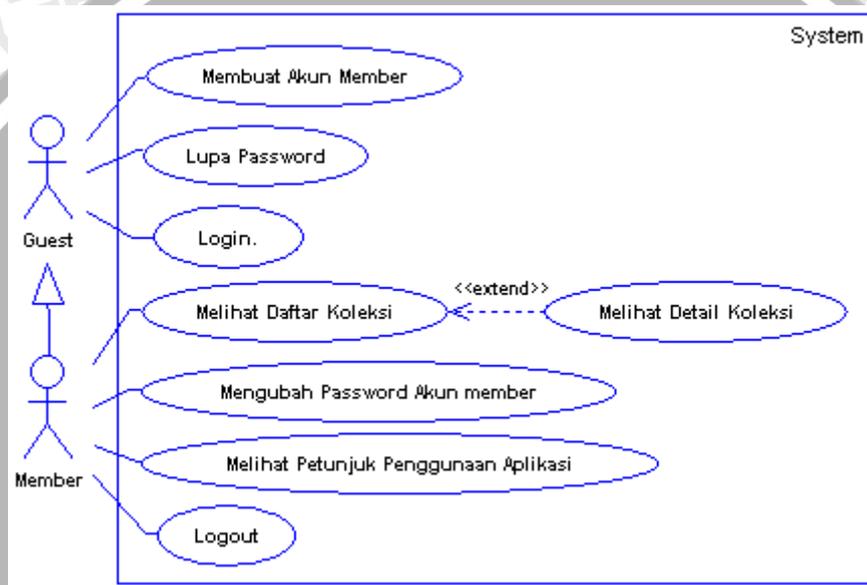
Tabel 4.2 Daftar kebutuhan fungsional dan non fungsional

ID	Requirements	Aktor	Nama Use Case
F01	Sistem harus dapat menyediakan fasilitas untuk membuat akun baru. Akun baru tersebut dapat digunakan untuk mengakses fungsionalitas sistem.	Guest	Membuat Akun Member
F02	Sistem harus menyediakan fasilitas <i>login</i> yang digunakan untuk melakukan proses autentikasi terhadap pengguna yang ingin menggunakan fasilitas sistem sebagai Member.	Guest	Login
F03	Sistem harus menyediakan fasilitas untuk membantu pengguna mendapatkan kembali password yang valid saat mereka lupa dengan password yang dimilikinya.	Guest	Lupa Password
F04	Sistem harus dapat digunakan untuk melihat daftar semua koleksi skripsi, tesis, disertasi, referensi, diktat mata kuliah, jurnal, kamus, laporan PKL dan klipng yang ada di Ruang Baca Teknik Elektro.	Member	Melihat Daftar Koleksi
F05	Sistem harus dapat digunakan untuk melihat koleksi secara lebih detail. Data detail koleksi yang ditampilkan yaitu judul asli, judul seri, ISBN, edisi, cetakan, jilid, pengarang, penerbit, tahun terbit, bahasa, deskripsi fisik, jumlah koleksi, subject, kode rak buku, dewey, bibliografi, jenis koleksi, paket, tipe media dan abstrak.	Member	Melihat Detail Koleksi
F06	Sistem harus dapat digunakan untuk mengubah password yang dimiliki oleh akun Member.	Member	Mengubah Password Akun Member
F07	Sistem harus menyediakan petunjuk atau bantuan kepada pengguna jika mengalami kesulitan dalam menjalankan aplikasi.	Member	Melihat Petunjuk Penggunaan Aplikasi
F08	Sistem harus menyediakan fasilitas <i>logout</i> yang digunakan untuk menghapus <i>session</i> Member dan keluar dari menu utama Member.	Member	Logout
N01	Sistem harus mampu direalisasikan pada perangkat <i>java mobile phone</i> yang sebenarnya (bukan emulator).	--	--

Sumber : Analisis Kebutuhan

4.1.3 Diagram Use Case

Kebutuhan-kebutuhan fungsional yang diperlukan oleh pengguna dan harus disediakan oleh sistem akan dimodelkan pada diagram *use case*. Pada sistem ini terdapat delapan buah *use case*, yaitu *use case* Membuat Akun Member, Login, Lupa Password, Melihat Daftar Koleksi, Melihat Detail Koleksi, Melihat Petunjuk Penggunaan Aplikasi, Mengubah Password Akun Member dan Logout dengan dua buah aktor yaitu *Guest* dan *Member*. Gambar 4.1 merupakan *use case diagram* dari sistem ini.



Gambar 4.1 Use Case Diagram Sistem Katalog Perpustakaan
Sumber: Analisis Kebutuhan

4.1.4 Skenario Use Case

Skenario *use case* menjelaskan secara detail interaksi dari masing-masing *use case* yang terdapat pada diagram *use case*. Skenario *use case* berisi uraian nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi dan pasca-kondisi yang diharapkan setelah berjalannya fungsional *use case*. Skenario *use case* juga memberikan penjelasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor (aliran utama), serta kejadian alternatif yang akan terjadi jika suatu kondisi tidak bisa terpenuhi (aliran alternatif).

Tabel 4.3 Skenario *use case* Membuat Akun Member.

Use case	Membuat akun Member
Aktor	Guest
Tujuan	Membuat akun baru sebagai Member
Deskripsi	Untuk membuat akun baru, aktor harus mengisi <i>field</i> username, password, nama, alamat, jenis kelamin, tempat dan tanggal lahir, telepon, email, jenis tanda pengenal, nomer tanda pengenal dan jenis anggota, setelah itu proses pembuatan akun baru dilakukan.
Pra-kondisi	Menu utama Guest dalam keadaan aktif
Pos-kondisi	Diperoleh akun baru sebagai Member.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Register".	2. Menampilkan halaman "Register" yang terdiri atas <i>field</i> "username", "password", "konfirmasi password", "nama", "alamat", "jenis kelamin", "tempat dan tanggal lahir", "telepon", "email", "jenis tanda pengenal", "no tanda pengenal" dan "jenis angota", selain itu tersedia tombol "Save", "Reset" dan "Back".
3. Aktor mengisi <i>field-field</i> pada form kemudian menekan tombol "Save".	4. Melakukan proses validasi isian aktor untuk memeriksa apakah user telah mengisi data dengan benar dan lengkap, kemudian jika semua data telah terisi dengan benar maka ditampilkan halaman baru yang menyatakan bahwa pembuatan akun baru berhasil dan terdapat tombol "OK".
5. Aktor menekan tombol "OK"	6. Sistem akan menampilkan menu utama Guest.
Aliran Alternatif 1 : Aktor mengosongkan semua isian <i>field</i>	
1. Aktor menekan tombol "Reset" (langkah ke-3 pada aliran utama).	2. Sistem mengosongkan semua isian <i>field</i> pada halaman "Register" sehingga aktor dapat memulai untuk mengisi <i>field-field</i> tersebut kembali.
Aliran Alternatif 2 : Aktor membatalkan proses registrasi	
1. Aktor menekan tombol "Back" (langkah ke-3 pada aliran utama).	2. Sistem menampilkan halaman sebelumnya yaitu menu utama Guest.
Aliran Alternatif 3 : Eksepsi jika proses validasi gagal	
	1. Menampilkan pernyataan peringatan kesalahan (<i>alert</i>), jika terdapat kesalahan berupa isian <i>field</i> yang belum lengkap atau terdapat perbedaan antara isian <i>field</i> "password" dan <i>field</i> "konfirmasi password" (tanggapan sistem ke-4 pada aliran utama). <i>Alert</i> terdiri dari pernyataan peringatan kesalahan dan tombol

	"OK".
2. Aktor menekan tombol "OK".	3. Sistem akan menampilkan halaman "Register" dengan <i>field</i> yang masih terisi oleh data sebelumnya.

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk login. Kebutuhan tersebut direpresentasikan oleh *use case* Login. Tabel 4.4. merupakan skenario *use case* Login.

Tabel 4.4 Skenario *use case* Login.

Use case	Login.	
Aktor	Guest	
Tujuan	Melakukan autentikasi pengguna yang akan menggunakan fasilitas sistem sebagai Member.	
Deskripsi	Untuk memulai proses <i>login</i> , terlebih dahulu aktor harus memasukkan <i>field</i> isian <i>username</i> dan <i>password</i> , setelah itu proses <i>login</i> dilakukan.	
Pra-kondisi	Menu utama Guest dalam keadaan aktif.	
Pos-kondisi	Guest berubah status menjadi Member dan ditampilkan menu utama Member yang berisi menu fasilitas-fasilitas dari sistem.	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Pada menu utama Guest, aktor memilih untuk melakukan proses <i>login</i> dengan cara menekan pilihan "Login".	2. Menampilkan halaman "Login" yang terdiri atas <i>field</i> "username" dan "password", selain itu tersedia tombol "Login", "Forgot Password", dan "Back".	
3. Aktor mengisi <i>field</i> "username" dan "password" kemudian menekan tombol "Login".	4. Melakukan proses validasi <i>field</i> isian "username" dan "password" untuk memeriksa aktor berhak menggunakan fasilitas sistem atau tidak. Jika aktor teridentifikasi sebagai Member maka ditampilkan menu utama Member antara lain: <ul style="list-style-type: none"> • <i>Catalog</i>, untuk melakukan pencarian katalog perpustakaan dan menampilkan hasilnya • <i>Account Password</i>, untuk mengubah password sebelumnya • <i>Help</i>, untuk melihat bantuan atau petunjuk dalam menjalankan aplikasi. • <i>Logout</i>, untuk menghapus <i>session</i> Member dan keluar dari menu utama Member Pada menu utama Member terdapat tombol "Select" dan "Exit".	
Aliran Alternatif 1 : Aktor membatalkan proses login		
1. Aktor memilih menu "Back" (langkah ke-3 pada aliran utama).	2. Menampilkan halaman sebelumnya yaitu menu utama Guest.	
Aliran Alternatif 2 : Eksepsi jika proses validasi gagal		
	1. Menampilkan pernyataan peringatan kesalahan (<i>alert</i>), jika terdapat kesalahan berupa isian <i>field</i> yang belum lengkap	

	(diantaranya field <i>username</i> dan <i>password</i>). Jika kesalahan terletak pada isian <i>username</i> dan <i>password</i> yang tidak cocok dengan data yang ada pada database, maka sistem akan menampilkan pernyataan peringatan kesalahan (<i>alert</i>) bahwa aktor harus memasukkan nilai <i>username</i> dan <i>password</i> yang valid. <i>Alert</i> terdiri dari pernyataan peringatan kesalahan dan tombol "OK"
2. Aktor menekan tombol "OK"	3. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Login". Data yang ada di isian <i>field</i> akan dikosongkan sehingga <i>user</i> dapat mengisinya kembali.
Aliran Alternatif 3 : Aktor memilih menu yang ada di menu utama Member	
1. Aktor memilih menu "Select" (tanggapan sistem ke-4 pada aliran utama).	2. Sistem akan menampilkan halaman baru. Proses interaksi selanjutnya berdasarkan menu yang dipilih seperti yang digambarkan pada skenario <i>use case</i> melihat daftar koleksi pada tabel 4.6, mengubah password akun Member pada tabel 4.8, melihat petunjuk penggunaan aplikasi pada tabel 4.9, logout pada tabel 5.0
Aliran Alternatif 4 : Aktor keluar dari aplikasi	
1. Aktor memilih menu "Exit" (tanggapan sistem ke-4 pada aliran utama).	2. Sistem akan menutup antarmuka yang sedang aktif dan mengakhiri aplikasi.

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk membantu pengguna yang lupa dengan password yang dimilikinya. Kebutuhan tersebut direpresentasikan oleh *use case* Lupa Password. Tabel 4.5 merupakan skenario *use case* Lupa Password.

Tabel 4.5 Skenario *use case* Lupa Password

<i>Use case</i>	Lupa Password
Aktor	Member
Tujuan	Membantu pengguna saat kehilangan atau lupa password
Deskripsi	Untuk mengetahui password akun Member, aktor harus memasukkan isian <i>field</i> "username", "nama", dan "no. tanda pengenal".
Pra-kondisi	Menu utama Guest dalam keadaan aktif.
Pos-kondisi	Diperoleh kembali password akun Member yang sebelumnya hilang atau lupa.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Pada halaman "Menu Utama Guest", aktor memilih menu "Forgot Password".	2. Menampilkan halaman "Forgot Password" yang berisi isian <i>field</i> "username", "nama", dan "no. tanda pengenal" serta tersedia tombol "Send", dan "Back"
3. Aktor menekan menu "Send".	4. Melakukan proses validasi isian aktor untuk memeriksa apakah user telah mengisi data



	dengan benar dan lengkap, kemudian jika semua data telah terisi dengan lengkap dan data tersebut ada dalam database maka ditampilkan halaman baru berisi informasi password akun Member yang valid dan terdapat tombol "OK".
5. Aktor menekan menu "OK".	6. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Login".
Aliran Alternatif 1 : Kembali ke tampilan sebelumnya	
1. Aktor menekan tombol "Back". (langkah ke-3 pada aliran utama).	2. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Menu Utama Guest".
Aliran Alternatif 2 : Eksepsi jika proses permintaan password yang valid gagal	
	1. Menampilkan pernyataan peringatan kesalahan (<i>alert</i>), jika terdapat kesalahan berupa isian <i>field</i> yang belum lengkap atau terdapat perbedaan antara data yang dimasukkan pada isian field dan data yang tersimpan dalam database (tanggapan sistem ke-4 pada aliran utama). <i>Alert</i> terdiri dari pernyataan peringatan kesalahan dan tombol "OK".
2. Aktor menekan tombol "OK".	3. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Forgot Password".

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk melakukan proses pencarian dan melihat daftar koleksi. Kebutuhan tersebut direpresentasikan oleh *use case* Melihat Daftar Koleksi. Tabel 4.6 merupakan skenario *use case* Melihat Daftar Koleksi.

Table 4.6 Skenario *use case* Melihat Daftar Koleksi.

<i>Use case</i>	Melihat Daftar Koleksi.
Aktor	Member
Tujuan	Untuk melakukan proses pencarian katalog perpustakaan dan menampilkan hasilnya.
Deskripsi	Untuk melihat daftar koleksi, aktor terlebih dahulu harus memilih menu "Catalog" kemudian memilih koleksi yang tersedia seperti skripsi, tesis, disertasi, referensi, dll, memilih kriteria berdasarkan judul, pengarang penerbit atau tahun terbit, memilih drop-down list konsentrasi yang diinginkan serta memasukkan kata kunci kemudian setelah menekan tombol "Search" maka akan ditampilkan informasi tertentu seperti ISBN, judul, pengarang, penerbit dan kode rak buku.
Pra-kondisi	Menu utama Member dalam keadaan aktif.
Pos-kondisi	Informasi tertentu yang berhubungan dengan koleksi yang diinginkan dapat ditampilkan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Pada menu utama Member, aktor memilih pilihan	2. Menampilkan halaman "Catalog" yang berisi

untuk melihat daftar koleksi dengan cara memilih menu "Catalog".	isian <i>field</i> pilihan koleksi yang tersedia, pilihan kriteria berdasarkan judul, pengarang, penerbit atau tahun terbit, pilihan drop-down list konsentrasi yang diinginkan dan isian <i>field</i> kata kunci serta tersedia tombol "Search", dan "Back".
3. Aktor memilih berdasarkan koleksi tertentu, kriteria tertentu, memilih paket tertentu, mengisi <i>field</i> isian kata kunci dan menekan tombol "Search".	4. Melakukan proses pencarian dan menampilkan daftar koleksi yang berisi informasi tertentu seperti ISBN, judul, pengarang, penerbit dan kode rak buku, tersedia menu "Detail" dan "Back".
Aliran Alternatif 1 : Menampilkan detail koleksi	
1. Aktor memilih salah satu judul koleksi diantara daftar koleksi yang ada kemudian menekan tombol "Detail" (tanggapan sistem ke-4 pada aliran utama).	2. Sistem akan menampilkan halaman baru. Proses interaksi selanjutnya seperti yang digambarkan pada skenario <i>use case</i> melihat detail koleksi pada tabel 4.6
Aliran Alternatif 2 : Kembali ke tampilan sebelumnya	
1. Aktor menekan tombol "Back" (tanggapan sistem ke-4 pada aliran utama).	2. Sistem akan menampilkan antarmuka atau halaman sebelumnya yaitu menu utama Member.
Aliran Alternatif 3 : Eksepsi jika isian kata kunci kosong	
1. Aktor menekan tombol "Search" tanpa mengisi isian jenis koleksi, kriteria, atau konsentrasi. (langkah ke-3 pada aliran utama).	2. Memberikan pernyataan peringatan (<i>alert</i>) bahwa aktor harus mengisikan isian <i>field</i> jenis koleksi, kriteria, atau konsentrasi, dan tersedia tombol "OK".
3. Aktor menekan tombol "OK".	4. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Catalog".
Aliran Alternatif 4 : Eksepsi jika data yang dicari tidak ditemukan	
1. Aktor menekan tombol "Search" setelah memilih koleksi, kriteria dan konsentrasi yang diinginkan, serta mengisi isian kata kunci. (langkah ke-3 pada aliran utama).	2. Memberikan pernyataan peringatan (<i>alert</i>) bahwa data yang dicari tidak ditemukan, dan tersedia tombol "OK".
3. Aktor menekan tombol "OK".	4. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Catalog".

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk melihat informasi tertentu yang berhubungan dengan koleksi secara lebih detail. Kebutuhan tersebut direpresentasikan oleh *use case* Melihat Detail Koleksi. Tabel 4.7 merupakan skenario *use case* Melihat Detail Koleksi.

Tabel 4.7 Skenario *use case* Melihat Detail Koleksi

<i>Use case</i>	Melihat Detail Koleksi.
Aktor	Member
Tujuan	Untuk melihat detail koleksi yang dipilih aktor.
Deskripsi	Untuk menampilkan detail koleksi yang telah dipilih. Data koleksi yang ditampilkan adalah judul asli, judul seri, ISBN, edisi, cetakan, jilid, pengarang, penerbit, tahun terbit, bahasa, deskripsi fisik, jumlah koleksi, subject, kode rak buku, dewey, bibliografi, jenis koleksi, paket, tipe media dan abstrak.
Pra-kondisi	Daftar koleksi telah ditampilkan sistem.
Pos-kondisi	Informasi detail koleksi ditampilkan sistem.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor menekan menu "Detail" setelah memilih salah satu judul koleksi pada daftar koleksi yang ditampilkan.	2. Menampilkan halaman "Catalog Detail" yang berisi informasi secara lebih detail seperti judul asli, judul seri, ISBN, edisi, cetakan, jilid, pengarang, penerbit, tahun terbit, bahasa, deskripsi fisik, jumlah koleksi, subject, kode rak buku, dewey, bibliografi, jenis koleksi, paket, tipe media dan abstrak, tersedia menu "Back".
3. Aktor menekan menu "Back".	4. Menampilkan halaman sebelumnya berupa tampilan daftar koleksi.

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk mengubah password. Kebutuhan tersebut direpresentasikan oleh *use case* Mengubah Password Akun Member. Tabel 4.8 merupakan skenario *use case* Mengubah Password Akun Member.

Tabel 4.8 Skenario *use case* Mengubah Password Akun Member

<i>Use case</i>	Mengubah Password Akun Member
Aktor	Member
Tujuan	Untuk mengubah dan mendapatkan password yang baru.
Deskripsi	Untuk mengubah password, aktor harus mengisi beberapa <i>field</i> . Setelah aktor mengisi <i>field-field</i> tersebut maka proses pengubahan password dilakukan.
Pra-kondisi	Menu utama Member dalam keadaan aktif.
Pos-kondisi	Diperoleh password yang baru.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Pada menu utama, aktor memilih menu "Account Password".	2. Menampilkan halaman "Account Password" yang terdiri atas field "password lama", "password baru", dan "konfirmasi password", selain itu tersedia menu "Change" dan "Back".

3. Aktor mengisi field yang ada dan menekan menu "Change".	4. Melakukan proses perubahan password. Jika proses tersebut berhasil dilakukan, maka akan ditampilkan halaman baru yang berisi pernyataan bahwa proses perubahan password berhasil. Tersedia menu "OK".
5. Aktor menekan menu "OK".	6. Sistem akan menampilkan halaman sebelumnya yaitu menu utama Member.
Aliran Alternatif 1 : Kembali ke tampilan sebelumnya	
1. Aktor menekan tombol "Back". (langkah ke-3 pada aliran utama).	2. Sistem akan menampilkan halaman sebelumnya yaitu menu utama Member.
Aliran Alternatif 2 : Eksepsi jika proses perubahan gagal	
	1. Menampilkan pernyataan peringatan kesalahan (<i>alert</i>), jika terdapat kesalahan berupa isian <i>field</i> yang belum lengkap. Jika kesalahan terdapat pada isian <i>field</i> "password lama" yang tidak sesuai dengan data yang tersimpan pada database, maka sistem akan menampilkan peringatan kesalahan bahwa password lama tidak valid. Jika kesalahan terletak pada isian <i>field</i> "password baru" dan "konfirmasi password" yang tidak sama, maka sistem akan menampilkan pernyataan peringatan kesalahan (<i>alert</i>) bahwa aktor harus memasukkan nilai "password baru" dan "konfirmasi password" yang sama. <i>Alert</i> terdiri dari pernyataan peringatan kesalahan dan tombol "OK".
2. Aktor menekan tombol "OK".	3. Sistem akan menampilkan halaman sebelumnya yaitu halaman "Account Password".

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk melihat petunjuk penggunaan aplikasi. Kebutuhan tersebut direpresentasikan oleh *use case* Melihat Petunjuk Penggunaan Aplikasi. Tabel 4.9 merupakan skenario *use case* Melihat Petunjuk Penggunaan Aplikasi.

Tabel 4.9 Skenario *use case* Melihat Petunjuk Penggunaan Aplikasi

Use case	Melihat Petunjuk Penggunaan Aplikasi
Aktor	Member
Tujuan	Menyediakan petunjuk atau bantuan kepada aktor jika mengalami kesulitan dalam menjalankan aplikasi
Deskripsi	Untuk menampilkan informasi yang berisi petunjuk atau bantuan dalam menjalankan atau menggunakan aplikasi. Informasi petunjuk meliputi semua fitur atau fasilitas yang terdapat dalam aplikasi ini beserta beberapa hal yang harus diperhatikan dan dipenuhi oleh aktor.
Pra-kondisi	Menu utama Member dalam keadaan aktif.
Pos-kondisi	Diperoleh keterangan atau penjelasan mengenai semua fasilitas yang tersedia dalam sistem beserta hal-hal yang harus diperhatikan dan dipenuhi oleh aktor.

Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Pada menu utama Member, aktor memilih menu "Help".	2. Menampilkan halaman "Help" yang berisi informasi petunjuk atau bantuan dalam menggunakan aplikasi serta tersedia tombol "Back".
3. Aktor menekan menu "Back".	4. Sistem akan menampilkan halaman sebelumnya yaitu menu utama Member.

Sumber : Analisis Kebutuhan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk menghapus *session* Member. Kebutuhan tersebut direpresentasikan oleh *use case* Logout. Tabel 4.10 merupakan skenario *use case* Logout.

Tabel 4.10 Skenario *use case* Logout

Use case	Logout
Aktor	Member
Tujuan	Menghapus <i>session</i> Member dan keluar dari menu utama Member.
Deskripsi	Untuk melakukan proses menghapus <i>session</i> Member, terlebih dahulu aktor memilih menu "Logout".
Pra-kondisi	Menu utama Member dalam keadaan aktif.
Pos-kondisi	<i>Session</i> Member terhapus dan keluar dari menu utama Member
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Pada menu utama Member, aktor memilih menu "Logout".	2. Menghapus <i>session</i> Member dan menampilkan halaman baru yang berisi pesan bahwa aktor telah <i>logout</i> . Tersedia menu "OK".
3. Aktor menekan menu "OK".	4. Sistem akan menutup halaman yang sedang aktif dan keluar dari aplikasi.

Sumber : Analisis Kebutuhan

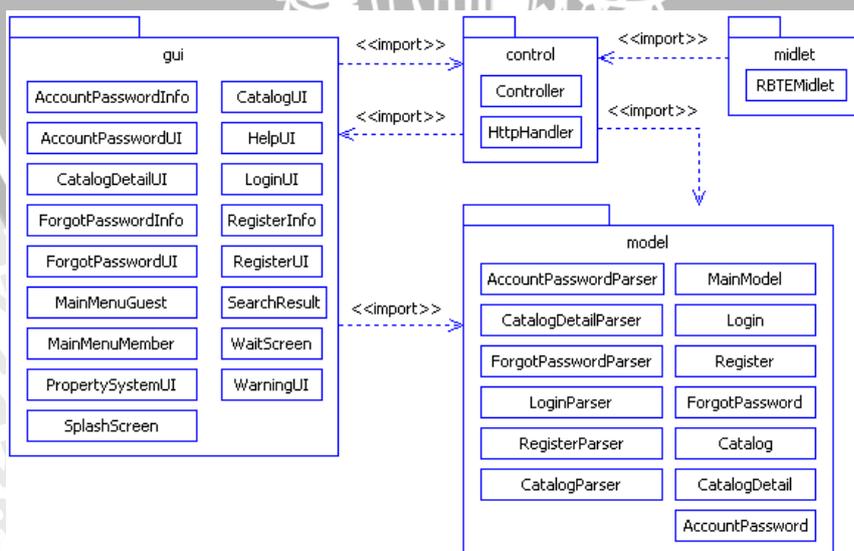
4.2 Perancangan Perangkat Lunak

Object Oriented Design (OOD) berhubungan dengan pengembangan model berorientasi *object* dari sistem perangkat lunak untuk mengimplementasikan kebutuhan-kebutuhan yang telah teridentifikasi pada analisis kebutuhan (*Object Oriented Analysis/OOA*). *Use case diagram* yang telah dimodelkan pada tahap analisis kebutuhan memberikan sebuah pandangan terhadap kebutuhan fungsional dari *user* yang harus disediakan oleh sistem perangkat lunak. Proses perancangan perangkat lunak mempunyai dua tahap, yaitu perancangan umum dan perancangan detail. Perancangan umum menggambarkan

relasi antar *package* dan *class* sebagai pemodelan sistem secara keseluruhan. Perancangan detail menggunakan *class diagram* dan *sequence diagram* sebagai pemodelan perangkat lunak. Perancangan detail melakukan perancangan terhadap pola hubungan antar komponen-komponen detail (*class* dan *object*) sehingga mampu membentuk sebuah fungsi kerja yang mampu memberikan pelayanan terhadap kebutuhan *user*. Sistem Katalog Perpustakaan berbasis *mobile* menggunakan basis data yang sudah ada pada aplikasi sebelumnya yaitu aplikasi Ruang Baca Teknik Elektro berbasis *web*. Beberapa tabel yang digunakan adalah tabel penerbit, paketkoleksi, paket, kondisi, subjectkoleksi, subject, koleksi, dewey, jeniskoleksi, pengarangkoleksi, pengarang, salinan, jenispengadaan, tipemedia, hakakses, jenistandapengenal, anggota_rbte, dan jenisanggota. *Entity-Relationship Diagram* Sistem Katalog Perpustakaan ditunjukkan pada bagian lampiran skripsi ini.

4.2.1 Perancangan Umum

Perancangan umum Sistem Katalog Perpustakaan menggambarkan relasi antar *package* dan *class* sebagai pemodelan sistem secara keseluruhan. Sistem Katalog Perpustakaan mempunyai 33 buah *class* yang telah dikelompokkan menjadi 5 buah *package* yaitu *package* gui, control, model, dan midlet. Relasi antar *package* yang digunakan untuk Sistem Katalog Perpustakaan dapat dilihat pada Gambar 4.2.



Gambar 4.2 Package Diagram Sistem Katalog Perpustakaan
Sumber: Perancangan



menampilkan *list* menu yang dapat diakses oleh *Guest* atau *Member*. *Class* *CatalogParser*, *CatalogDetailParser*, *LoginParser*, *RegisterParser*, *AccountPasswordParser* dan *ForgotPasswordParser* bertugas menerima data respon server dari *class* *HttpHandler*. *Class* *RegisterInfo*, *AccountPasswordInfo*, *ForgotPasswordInfo*, dan *CatalogDetailUI* mengambil data dari *class* yang ada di paket model dan menampilkan informasi yang dibawahnya ke *user*.

4.2.2 Perancangan Detail

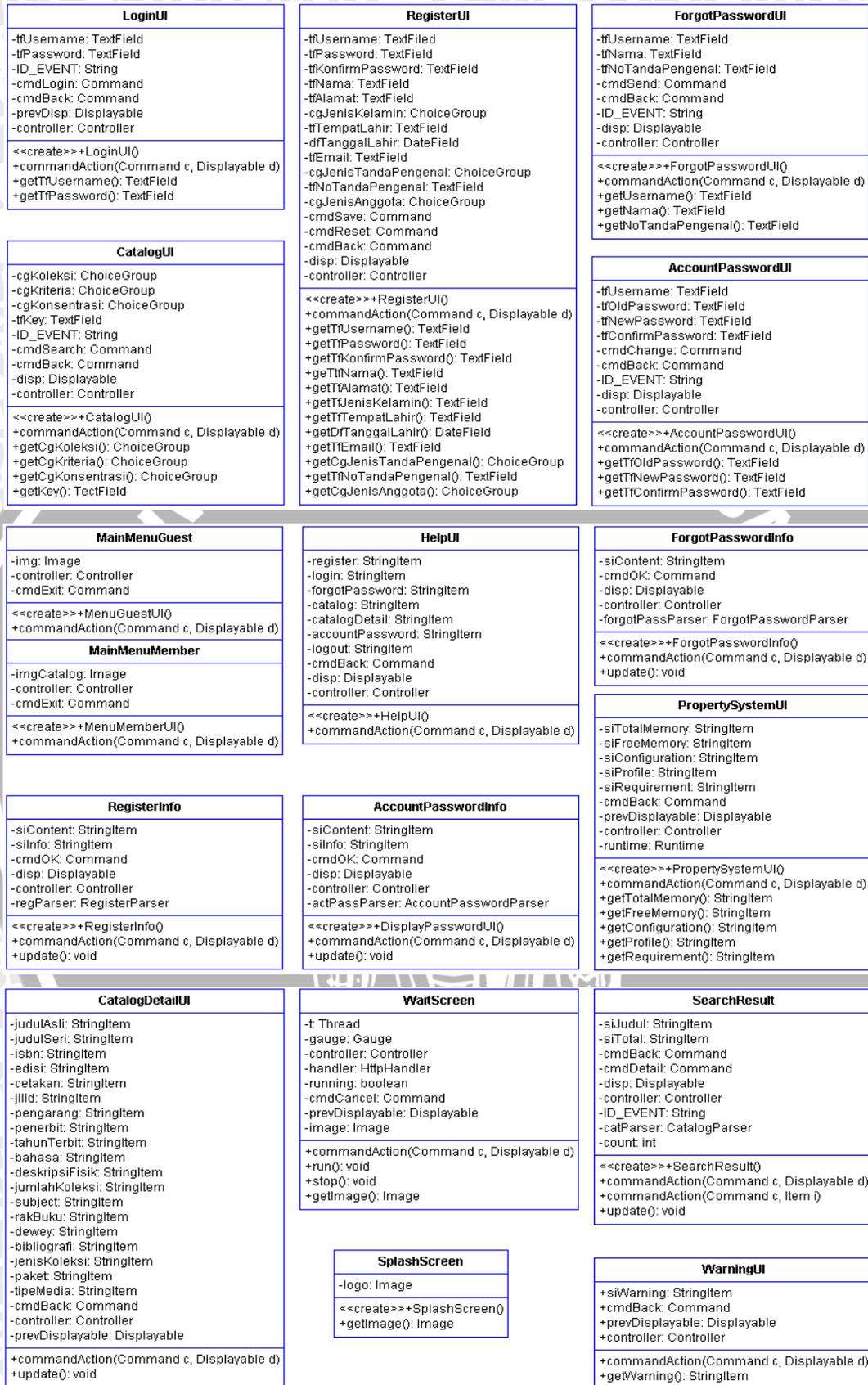
Perancangan detail menggunakan *class diagram* dan *sequence diagram* sebagai pemodelan perangkat lunak. Perancangan detail menjelaskan mengenai pola hubungan antar komponen-komponen detail (*class* dan *object*) sehingga mampu membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan *user*.

4.2.2.1 Diagram Kelas (*Class Diagram*)

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah *object* dan merupakan inti dari pengembangan dan desain berorientasi *object*. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Sistem Katalog Perpustakaan mempunyai *class-class* yang telah dikelompokkan ke dalam berbagai *package* diantara lain *package* *gui*, *control*, *midlet*, dan *model*.

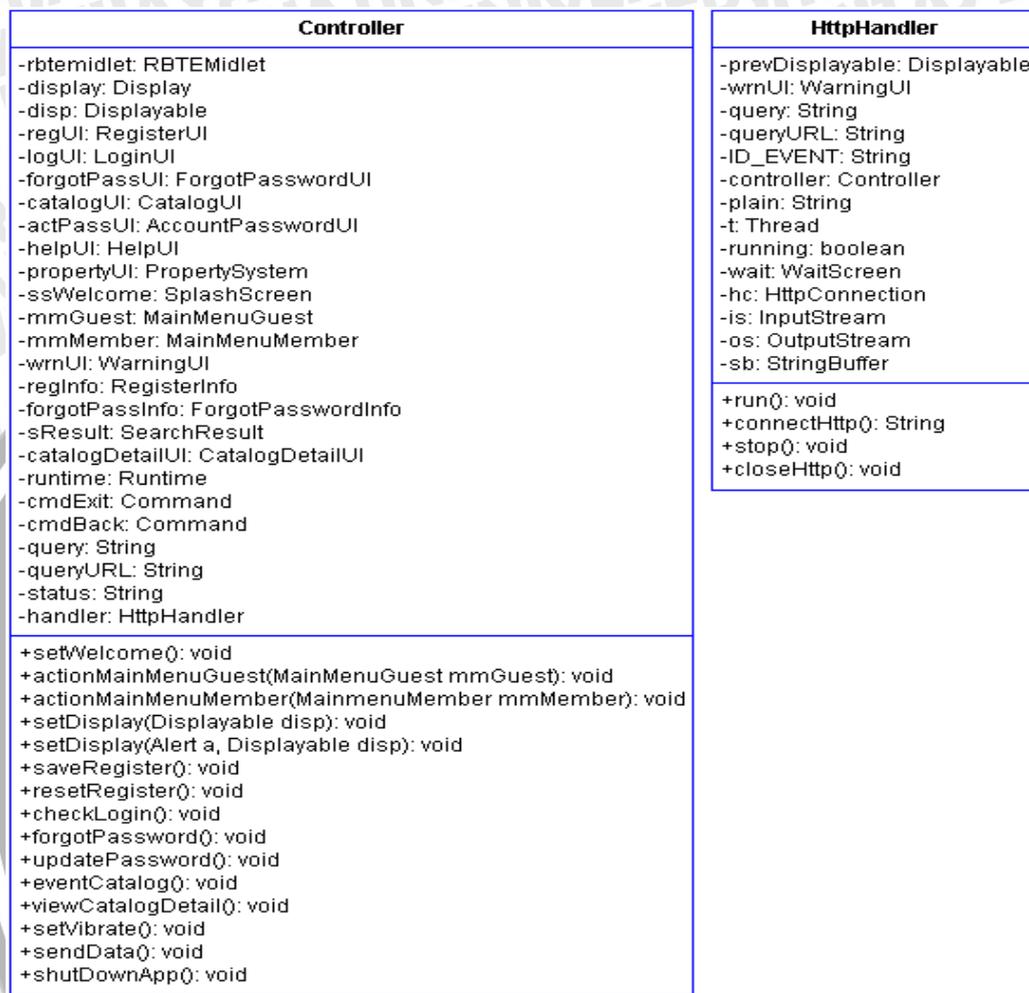
Package *gui* terdiri dari 17 buah *class* antara lain *AccountPasswordUI*, *AccountPasswordInfo*, *CatalogDetailUI*, *CatalogUI*, *ForgotPasswordInfo*, *ForgotPasswordUI*, *HelpUI*, *LoginUI*, *MainMenuGuest*, *MainMenuMember*, *PropertySystemUI*, *RegisterInfo*, *RegisterUI*, *SearchResult*, *SplashScreen*, *WaitScreen*, *WarningUI*. Atribut dan operasi dari *class-class* yang ada di *package* *gui* digambarkan dalam *class diagram* pada Gambar 4.4.



Gambar 4.4 Class diagram anggota package gui
Sumber: Perancangan

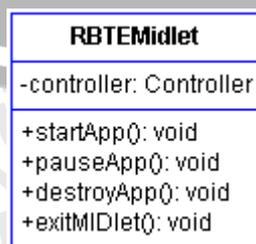


Package control terdiri dari 2 buah class antara lain Controller dan HttpHandler. Atribut dan operasi dari class-class yang ada di package control digambarkan dalam class diagram pada Gambar 4.5.



Gambar 4.5 Class diagram anggota package control
 Sumber: Perancangan

Package midlet terdiri dari satu buah class yaitu RBTEMidlet. Atribut dan operasi dari class RBTEMidlet digambarkan dalam class diagram pada Gambar 4.6.



Gambar 4.6 Class diagram anggota package midlet
 Sumber: Perancangan



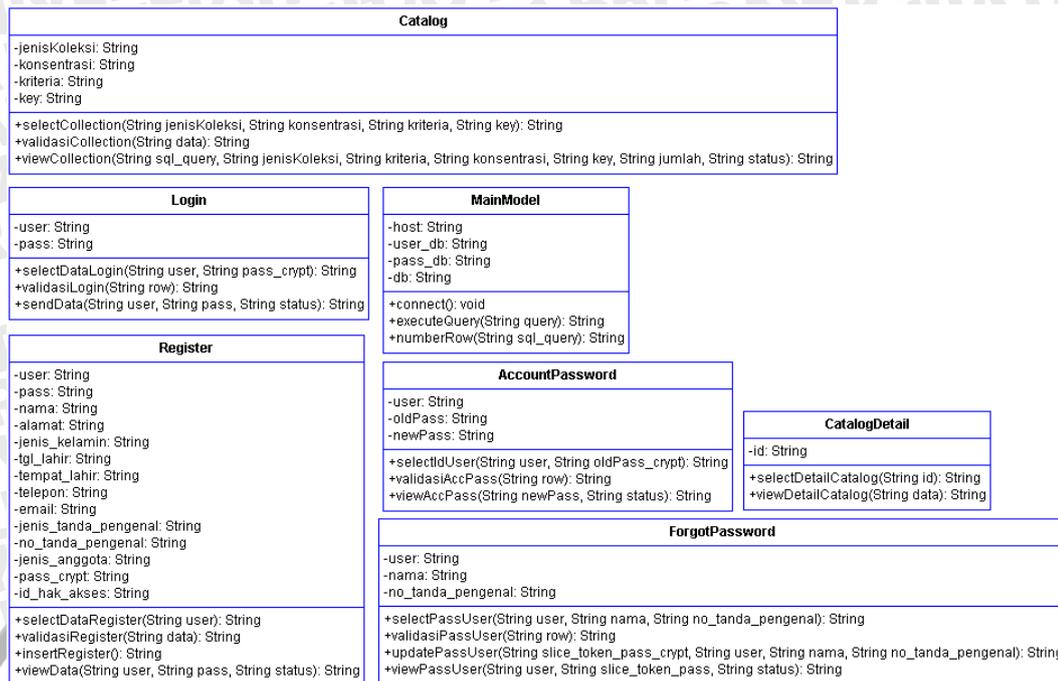
Package model yang terdapat di *client* terdiri dari 6 buah *class* antara lain LoginParser, CatalogParser, AccountPasswordParser, RegisterParser, CatalogDetailParser, ForgotPasswordParser. Atribut dan operasi dari *class-class* yang ada di *package* model *client* digambarkan dalam *class diagram* pada Gambar 4.7.



Gambar 4.7 Class diagram anggota package model di *client*
Sumber: Perancangan

Package model yang terdapat di *server* terdiri dari 7 buah *class* antara lain Login, Catalog, AccountPassword, Register, CatalogDetail, ForgotPassword, MainModel. Atribut dan operasi dari *class-class* yang ada di *package* model *server* digambarkan dalam *class diagram* pada Gambar 4.8.





Gambar 4.8 Class diagram anggota package model di server
Sumber: Perancangan

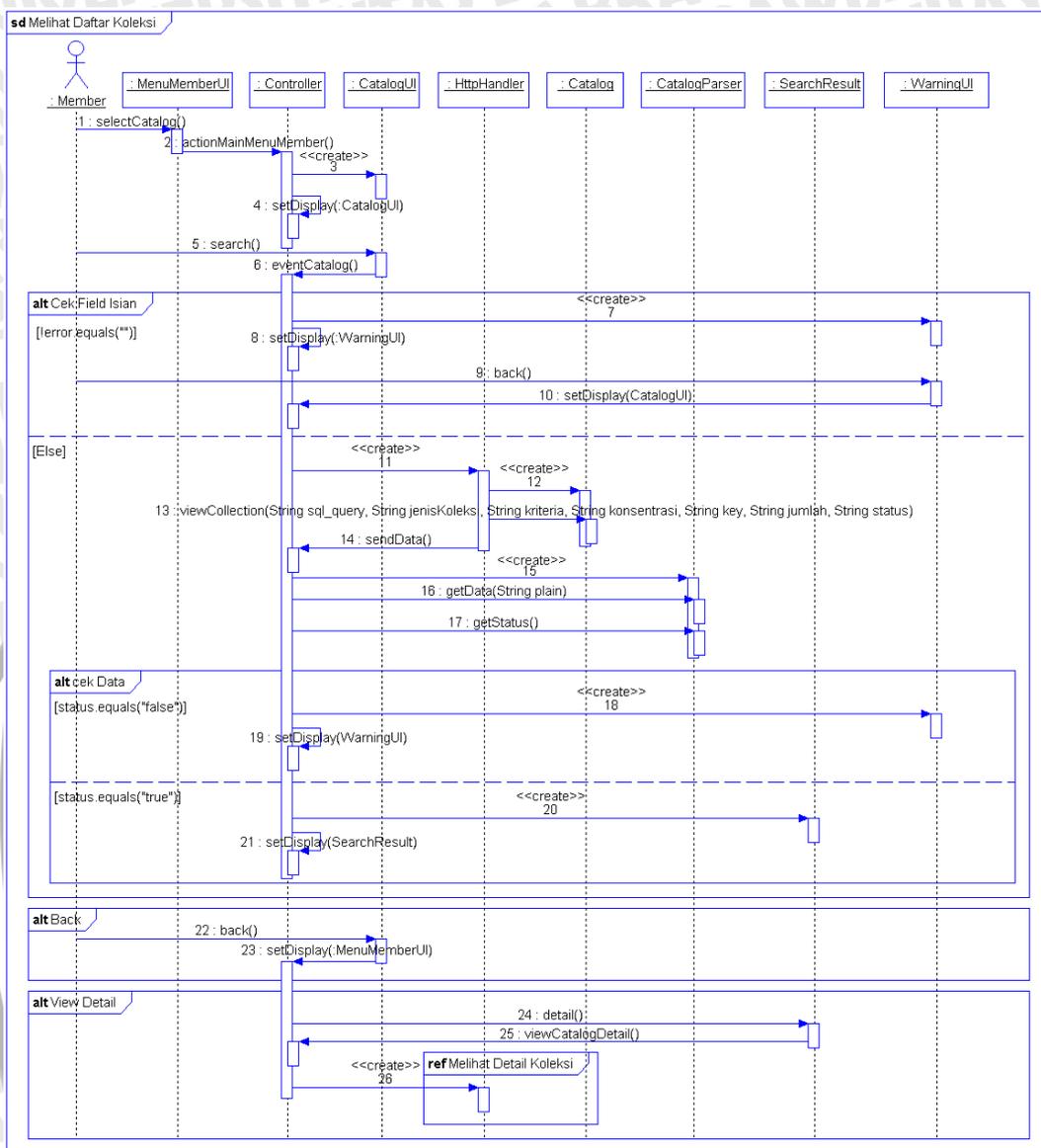
4.2.2.2 Diagram Sekuensial (Sequence Diagram)

Pemodelan aliran jalannya proses interaksi antar *object* atau *class* yang disusun berdasarkan urutan waktu ditunjukkan oleh *sequence diagram*. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram sekuensial (*sequence diagram*) disusun dengan mengambil acuan pada *use case* dan *class-class* yang membentuk fungsionalitas yang digambarkan pada *use case* tersebut.

- **Diagram sekuensial Melihat Daftar Koleksi**

Diagram sekuensial ini menggambarkan interaksi antar *object* atau *class* ketika seorang pengguna melakukan proses pencarian koleksi. Pengguna harus memilih koleksi yang tersedia, kriteria, konsentrasi, dan memasukkan kata kunci. Diagram sekuensial Melihat Daftar Koleksi mempunyai referensi ke diagram sekuensial Melihat Detail Koleksi yang dijelaskan pada Gambar 4.9.

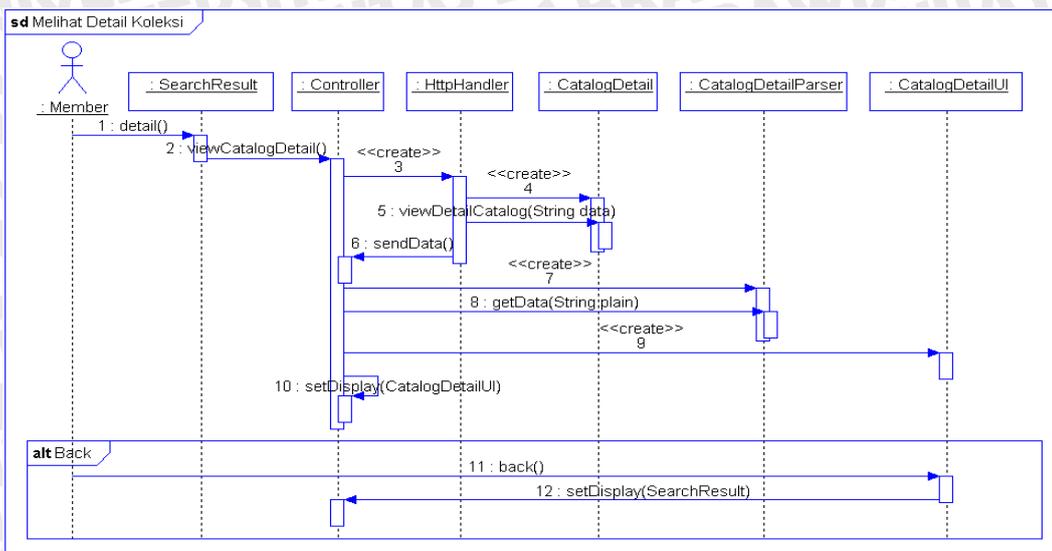




Gambar 4.9 Diagram sekuensial Melihat Daftar Koleksi
 Sumber: Perancangan

- **Diagram sekuensial Melihat Detail Koleksi**

Diagram sekuensial ini menggambarkan interaksi antar *object* atau *class* ketika seorang pengguna melihat data koleksi secara lebih detail. Pengguna harus memilih judul koleksi yang diinginkan. Data koleksi yang ditampilkan adalah judul asli, judul seri, isbn, edisi, cetakan, jilid, pengarang, penerbit, tahun terbit, bahasa, deskripsi fisik, jumlah koleksi, subject, kode rak buku, dewey, bibliografi, jenis koleksi, paket, tipe media, dan abstrak.



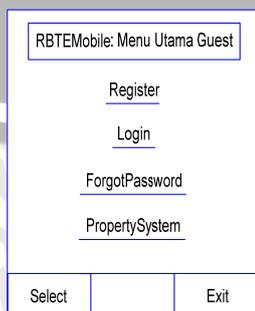
Gambar 4.10 Diagram sekuensial Melihat Detail Koleksi
Sumber: Perancangan

4.2.3 Perancangan User Interface

User interface merupakan tampilan dimana user berinteraksi dengan sistem. Desain User Interface dari Sistem Katalog Perpustakaan terdiri dari Menu Utama Guest, Menu Utama Member, Form Register, Form Login, Form Forgot Password, Form Catalog, dan Form Account Password.

4.2.3.1 Perancangan Menu Utama Guest

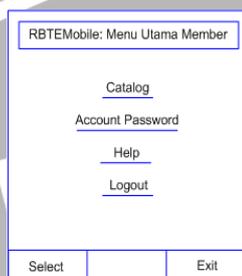
Menu Utama Guest adalah User Interface yang ditampilkan pertama kali saat user menjalankan aplikasi. Menu Utama Guest terdiri dari dua buah list menu yaitu Register dan Login serta dua buah Command yaitu Select, dan Exit. Menu Register untuk membuat akun baru sebagai Member. Menu Login untuk masuk ke Menu Utama Member dan dapat menggunakan fasilitas sistem sebagai Member. Desain Menu Utama Guest ditunjukkan oleh gambar berikut



Gambar 4.11 Desain Menu Utama Guest
Sumber: Perancangan

4.2.3.2 Perancangan Menu Utama Member

Menu Utama Member ditampilkan setelah *user* berhasil melakukan login dan berhak menggunakan fasilitas sistem sebagai Member. Menu Utama Member terdiri dari empat buah *list* menu yaitu Catalog, Account Password, Help, dan Logout serta dua buah *Command* yaitu Select, dan Exit. Desain Menu Utama Member ditunjukkan oleh gambar berikut



Gambar 4.12 Desain Menu Utama Member

Sumber: Perancangan

4.2.3.3 Perancangan Form Login

Untuk dapat mengakses sistem sebagai Member, pengguna harus melakukan proses *login* terlebih dahulu. *Form Login* terdiri dari dua buah *TextField* yaitu Username, dan Password serta tiga buah *Command* yaitu Login, Forgot Password, dan Back. Desain *Form Login* ditunjukkan oleh gambar berikut



Gambar 4.13 Desain Form Login

Sumber: Perancangan

4.2.3.4 Perancangan Form Catalog

Form Catalog merupakan *User Interface* untuk melakukan proses pencarian dan melihat daftar koleksi skripsi, tesis, disertasi, referensi, diktat mata kuliah, jurnal, kamus, laporan PKL, atau klipng yang ada di Ruang Baca Teknik Elektro. *Form Catalog* terdiri dari tiga buah *ChoiceGroup* yaitu Koleksi, Kriteria, dan Konsentrasi serta satu buah *TextField* yaitu Kata Kunci. *Form Catalog*

mempunyai dua buah *Command* yaitu Search dan Back. Desain *Form Catalog* ditunjukkan oleh gambar berikut

Gambar 4.14 Desain *Form Catalog*
Sumber: Perancangan

4.2.3.5 Perancangan *Form Register*

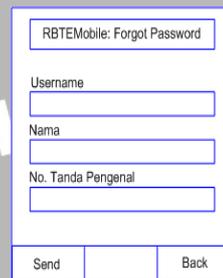
Form Register merupakan *User Interface* untuk membuat akun baru sebagai Member. Member dapat menggunakan semua fasilitas sistem. *Form Register* terdiri dari sepuluh buah *TextField* yaitu Username, Password, Konfirmasi Password, Nama, Alamat, Tanggal Lahir, Tempat Lahir, Nomer Telepon, Email, dan Nomer Tanda Pengenal serta tiga buah *ChoiceGroup* yaitu Jenis Kelamin, Jenis Tanda Pengenal, dan Jenis Anggota. *Form Register* mempunyai tiga buah *Command* yaitu Save, Reset, dan Back. Desain *Form Register* ditunjukkan oleh gambar berikut

Gambar 4.15 Desain *Form Register*
Sumber: Perancangan



4.2.3.6 Perancangan *Form Forgot Password*

Form Forgot Password merupakan *User Interface* untuk membantu pengguna mendapatkan kembali password yang valid saat mereka lupa dengan password yang dimilikinya. *Form Forgot Password* terdiri dari tiga buah *TextField* yaitu Username, Nama, dan Nomer Tanda Pengenal serta dua buah *Command* yaitu Send, dan Back. Desain *Form Forgot Password* ditunjukkan oleh gambar berikut

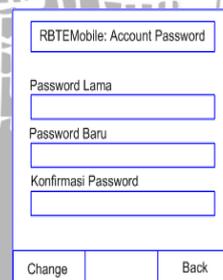


RBTEMobile: Forgot Password	
Username	<input type="text"/>
Nama	<input type="text"/>
No. Tanda Pengenal	<input type="text"/>
Send	Back

Gambar 4.16 Desain *Form Forgot Password*
Sumber: Perancangan

4.2.3.7 Perancangan *Form Account Password*

Form Account Password merupakan *User Interface* untuk mengubah password yang dimiliki oleh akun Member dan mendapatkan password yang baru. *Form Account Password* terdiri dari tiga buah *TextField* yaitu Password Lama, Password Baru, dan Konfirmasi Password serta dua buah *Command* yaitu Change, dan Back.



RBTEMobile: Account Password	
Password Lama	<input type="text"/>
Password Baru	<input type="text"/>
Konfirmasi Password	<input type="text"/>
Change	Back

Gambar 4.17 Desain *Form Account Password*
Sumber: Perancangan

BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak sebelumnya. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi tiap *class* pada file program, implementasi algoritma, dan implementasi antarmuka.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sebuah sistem katalog perpustakaan berbasis J2ME yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak

5.1.1 Spesifikasi Perangkat Keras

Pengembangan sistem katalog perpustakaan berbasis J2ME menggunakan perangkat keras dengan spesifikasi yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core(TM) 2 Duo 1.5 GHz
Memori (RAM)	2.99 GHz
Hardisk	Seagate ST9120822AS, kapasitas 120 GB, 5400 rpm
Mother Board	WISTRON 30CD
VGA Card	Intel 965 Graphics Media Accelerator
Mobile Phone	Sony Ericsson G502

Sumber: Implementasi

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan sistem katalog perpustakaan berbasis J2ME menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak komputer

Sistem operasi	Microsoft Windows XP SP2 version 2002
Bahasa pemrograman	Java2 Micro Edition, PHP
Lingkungan pemrograman	JDK 1.6.0_06, Java(TM) Standart Edition Runtime Environment (build 1.6.0_06-b02), J2ME Wireless Tool Kit 2.5.2 Apache Version 2.2.11, PHP 5.2.8, MySQL 5.0.51
IDE (<i>Integrated Development Environment</i>)	Netbeans 6.5, Netbeans Mobility pack 8.0, Notepad++ 3.8

Sumber: Implementasi

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Aplikasi ini dibangun hanya menyediakan layanan pencarian informasi katalog yang ada di RBTE.
2. Informasi yang ditampilkan bukan keseluruhan isi bahan pustaka melainkan hanya *list* atau daftar bahan pustaka (judul, nama pengarang, edisi, cetakan, penerbit, tahun terbit, ISBN).
3. Teknologi yang digunakan pada sisi *client* adalah *Java2 Micro Edition*(J2ME), pada sisi *server* menggunakan Apache sebagai web *server*, MySQL sebagai penyimpan data dan PHP sebagai bahasa *server side* untuk komunikasi data dari dan ke *server*.
4. Koneksi antara *client* dan *server* menggunakan protokol HTTP.
5. Menggunakan basis data yang sudah ada dalam basis data Ruang Baca Teknik Elektro Universitas Brawijaya.

5.3 Implementasi Class pada File Program

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program *.java dan *.php. Tabel 5.3 menjelaskan mengenai pasangan antara *class* dengan *file* program yang digunakan untuk mengimplmentasikannya.

Tabel 5.3 Implementasi *class* pada kode program *.java dan *.php

No.	Nama Paket	Nama Klas	Nama File Program
1	midlet	RbteMidlet	RbteMidlet.java
2		LoginUI	Retrieval.java
3		RegisterUI	RetrievalResult.java

4	gui	ForgotPasswordUI	ForgotPasswordUI.java	
5		PropertySystemUI	PropertySystemUI.java	
6		CatalogUI	CatalogUI.java	
7		AccountPasswordUI	AccountPasswordUI.java	
8		HelpUI	HelpUI.java	
9		SearchResult	SearchResult.java	
10		WaitScreen	WaitScreen.java	
11		WarningUI	WarningUI.java	
12		MainMenuGuest	MainMenuGuest.java	
13		MainMenuMember	MainMenuMember.java	
14		SplashScreen	SplashScreen.java	
15		CatalogDetailUI	CatalogDetailUI.java	
16		ForgotPasswordInfo	ForgotPasswordInfo.java	
17		AccountPasswordInfo	AccountPasswordInfo.java	
18		RegisterInfo	RegisterInfo.java	
19		control	Controllor	Controllor.java
20			HttpHandler	HttpHandler.java
21		model	LoginParser	Parser.java
22	RegisterParser		RegisterParser.java	
23	CatalogParser		CatalogParser.java	
24	CatalogDetailParser		CatalogDetailParser.java	
25	ForgotPasswordParser		ForgotPasswordParser.java	
26	AccountPasswordParser		AccountPasswordParser.java	
27	Login		Login.php	
28	Register		Register.php	
29	Catalog		Catalog.php	
30	CatalogDetail		CatalogDetail.php	
31	ForgotPassword		ForgotPassword.php	
32	AccountPassword		AccountPassword.php	
33	MainModel		MainModel.php	

Sumber: Implementasi

5.4 Implementasi Algoritma

Sistem katalog perpustakaan berbasis *mobile* mempunyai proses utama yaitu proses pencarian koleksi.

5.4.1 Implementasi Algoritma Pencarian Koleksi

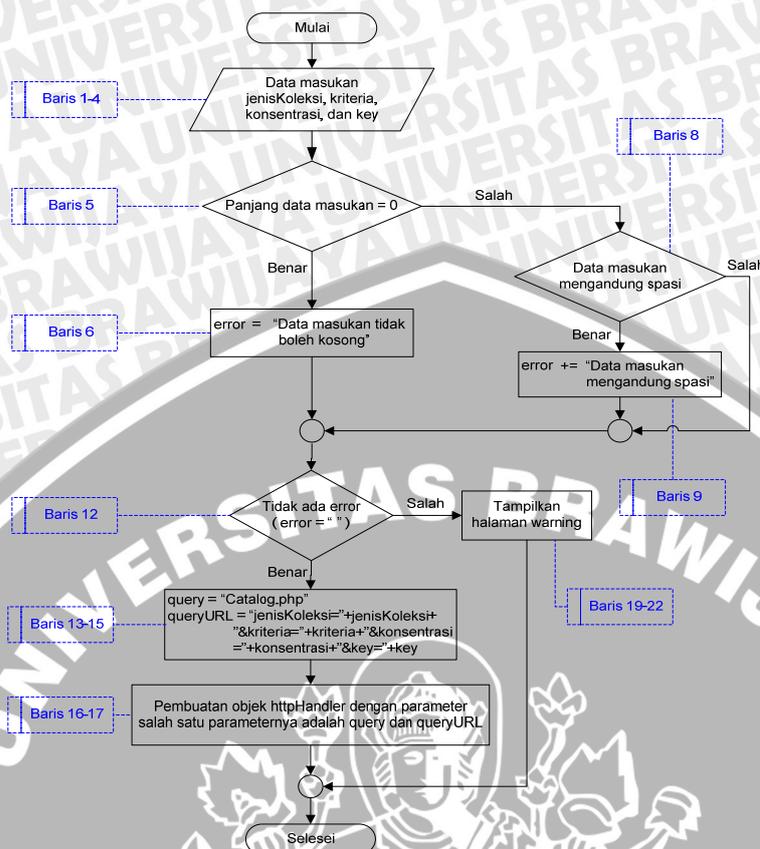
Proses pencarian koleksi melibatkan proses yang ada di *client* dan proses yang ada di *server*. Proses yang ada di *client* diimplementasikan dengan *script* J2ME, sedangkan yang ada di *server* diimplementasikan dengan *script* PHP.

Beberapa *class* yang diimplementasikan dengan J2ME adalah *class* CatalogUI dalam paket *gui*, *class* Controller dalam paket *control*, *class* HttpHandler dalam paket *control*, *class* CatalogParser dalam paket *model*, dan *class* SearchResult dalam paket *gui*. *Class* CatalogUI mempunyai fungsi menampilkan *user interface* dan menangkap *input* data dari *user*. *Class* Controller mempunyai tugas mengontrol aksi dari *user*. *Class* HttpHandler mempunyai fungsi mengirimkan *request user* ke *server* dengan *method request* adalah POST dan menangkap *respon* dari *server* melalui *HttpConnection*. *Class* Parser mempunyai fungsi untuk mengolah data dalam bentuk *plain text* (data sederhana) yang telah didapat dari *respon server* menjadi format data yang bisa ditampilkan ke *user*. *Class* SearchResult mempunyai fungsi mengambil data hasil dari *class* Parser dan menampilkannya ke *user*.

Beberapa *class* yang diimplementasikan dengan PHP adalah *class* MainModel dan *class* Catalog. *Class* MainModel mempunyai fungsi untuk melakukan koneksi ke basis data MySQL. *Class* Catalog mempunyai fungsi mencari dan mengambil data yang ada di basis data, melakukan validasi jika data yang dicari tidak ada dalam basis data, dan mengirimkan data dengan format data *plain text* (data sederhana).

Algoritma pencarian koleksi pada sisi *client* diterapkan pada *class* Controller dalam paket *control*, *class* HttpHandler dalam paket *control*, dan *class* CatalogParser dalam paket *model*.

Representasi algoritma pencarian koleksi untuk *eventCatalog()* ditunjukkan dalam diagram alir pada Gambar 5.1 dan *pseudocode* pada Gambar 5.2.



Gambar 5.1 Diagram alir pencarian koleksi untuk eventCatalog()

Sumber: Implementasi

METHOD eventCatalog PARAMETER displayable, ID_EVENT, jenisKoleksi, kriteria, konsentrasi, key

DECLARATION:

TYPE displayable IS Displayable
 ID_EVENT, jenisKoleksi, kriteria, konsentrasi, key IS String
 queryURL, query, error IS String
 httpHandler IS HttpHandler
 wrnUI IS WarningUI
 controller IS Controller

DESCRIPTION:

```

1  jk <- CALL jenisKoleksi.length()
2  kr <- CALL kriteria.length()
3  ks <- CALL konsentrasi.length()
4  ky <- CALL key.length()
5  IF panjang data masukan = 0 THEN
6      error = "Data tidak boleh kosong"
7  ELSE
8      IF data masukan mengandung spasi THEN
9          error = "Data mengandung spasi"
10     ENDIF
11  ENDIF
12  IF CALL error.equals("") THEN
13      queryURL <- "Catalog.php"
14      query <- "jenisKoleksi="+jenisKoleksi+"&kriteria="+kriteria+
15              "&konsentrasi="+konsentrasi+"&key="+key
16      handler <- CREATE OBJECT HttpHandler(controller,
17              displayable, ID_EVENT, queryURL, query)
    
```



```

18 ELSE
19   wrnUI <- CREATE OBJECT WarningUI(controller, displayable)
20   CALL wrnUI.setTitle("RBTEMobile : Search Catalog Warning")
21   CALL wrnUI.getWarning().setText(error)
22   CALL controller.setDisplay(wrnUI)
23 ENDIF

```

Gambar 5.2 Algoritma pencarian koleksi untuk eventCatalog()

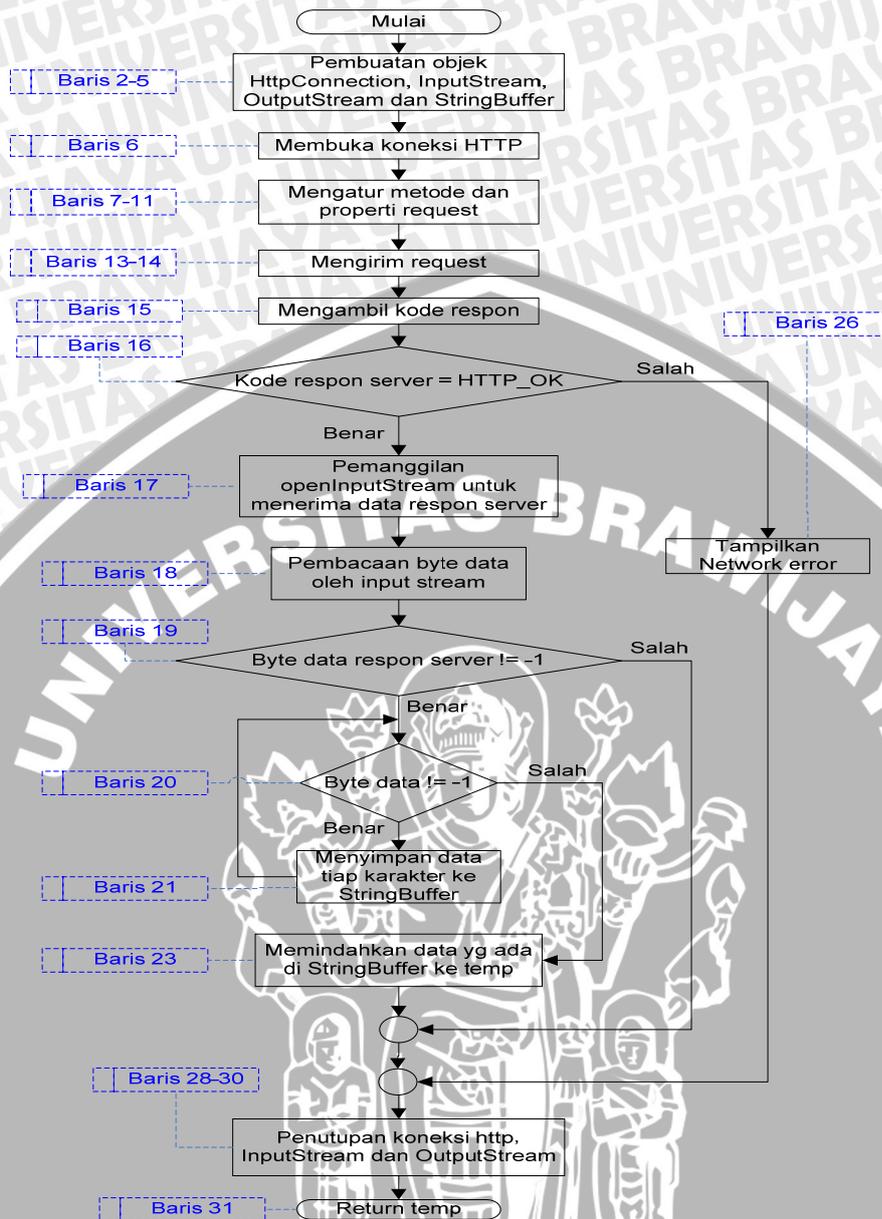
Sumber: Implementasi

Penjelasan algoritma pencarian koleksi untuk eventCatalog() pada Gambar 5.2 yaitu:

1. Baris 1-4 menjelaskan pengambilan panjang data masukan jenisKoleksi, kriteria, konsentrasi, dan key
2. Baris 5-6 menjelaskan pemeriksaan panjang data masukan jika sama dengan 0 maka nilai error sama dengan "Data tidak boleh kosong"
3. Baris 8-9 menjelaskan pemeriksaan jika data masukan mengandung spasi maka nilai error sama dengan "Data mengandung spasi"
4. Baris 12-17 menjelaskan pemeriksaan jika nilai error sama dengan kosong (`error.equals("")`) maka dibuat objek handler untuk memproses data masukan dan melakukan koneksi HTTP. *Class* `HttpHandler` memproses isi string query menjadi sebuah request untuk melakukan koneksi ke *server* melalui HTTP dengan alamat yang sesuai dengan *path* URL dalam string queryURL.
5. Baris 18-22 menjelaskan aliran alternatif jika nilai error tidak sama dengan kosong maka dilakukan pembuatan objek `WarningUI` dan menampilkan informasi error ke *user*.

Representasi algoritma pencarian koleksi untuk `connectHttp()` ditunjukkan dalam diagram alir pada Gambar 5.3 dan *pseudocode* pada Gambar

5.4.



Gambar 5.3 Diagram alir pencarian koleksi untuk connectHttp()
 Sumber: Implementasi

METHOD connectHttp PARAMETER displayable, ID_EVENT, jenisKoleksi, kriteria, konsentrasi, key

DECLARATION:
 TYPE prevDisplayable, displayable IS Displayable
 temp, url, queryURL, query IS String
 len IS long
 ch IS int
 httpHandler IS HttpHandler

DESCRIPTION:



```

1 url <- "http://www.server.com/"+queryURL
2 HttpURLConnection hc <- null
3 InputStream is <- null
4 OutputStream os <- null
5 StringBuffer sb <- CREATE OBJECT StringBuffer()
6 hc <- CALL (HttpURLConnection)Connector.open(url)
7 CALL hc.setRequestMethod(CALL HttpURLConnection.POST)
8 CALL hc.setRequestProperty("User-Agent", "Profile/MIDP-2.0
9 Configuration/CLDC-1.0")
10 CALL hc.setRequestProperty("Content-Language", "en-US")
11 CALL hc.setRequestProperty("Content-Type", "application/x-www-form-
12 urlencoded")
13 os <- CALL hc.openOutputStream()
14 CALL os.write(CALL query.getBytes())
15 rc <- CALL hc.getResponseCode()
16 IF rc == CALL HttpURLConnection.HTTP_OK THEN
17 is <- CALL hc.openInputStream()
18 WHILE (ch <- CALL is.read()) != -1
19 CALL sb.append((char)ch)
20 ENDWHILE
21 temp <- CALL sb.toString()
22 ELSE
23 throw IOException("Koneksi jaringan error, coba lagi")
24 ENDIF
25 CALL is.close()
26 CALL os.close()
27 CALL hc.close()
28 RETURN temp

```

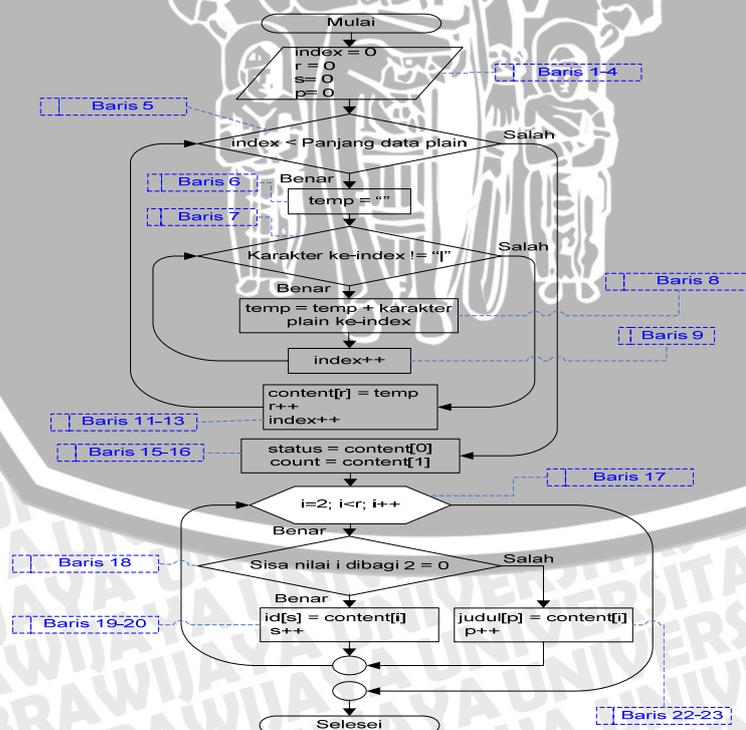
Gambar 5.4 Algoritma pencarian koleksi untuk connectHttp()

Sumber: Implementasi

Penjelasan algoritma pencarian koleksi untuk connectHttp() pada Gambar 5.4 yaitu:

1. Baris 1-5 menjelaskan inisialisasi alamat url dan pembuatan objek HttpURLConnection, InputStream, OutputStream dan StringBuffer. Proses yang melakukan koneksi jaringan dengan menggunakan koneksi HTTP diciptakan dalam *thread* sendiri. Hal ini berguna untuk mencegah *deadlock*. *Deadlock* terjadi karena pada saat koneksi jaringan diciptakan, MIDP akan menanyakan konfirmasi penggunaan *airtime* (koneksi internet via GPRS, dan sebagainya).
2. Baris 6 menjelaskan pemanggilan method open() dari *class* statis Connector, yang telah di-*casting* menjadi HttpURLConnection. Pemanggilan metode open() untuk membuka koneksi HTTP dan mengirim *request* ke *server* dengan alamat yang dispesifikasikan pada string url
3. Baris 7-11 menjelaskan pengaturan metode jenis *request* (POST) dan pengaturan properti-properti *request* seperti User-Agent, Content-Language, dan Content-Type

4. Baris 13-14 menjelaskan pemanggilan `openOutputStream` untuk mengirimkan *request*
 5. Baris 15 menjelaskan pengambilan kode *respon* dari server
 6. Baris 16 menjelaskan pemeriksaan terhadap kode *respon*. Bila kode *respon* tidak sama dengan 200 (HTTP_OK) maka ditampilkan *network error*
 7. Baris 17 menjelaskan pemanggilan `openInputStream` untuk menerima isi *respon* dari server
 8. Baris 18-20 menjelaskan pengumpulan tiap karakter dari *respon server* dan menyimpannya ke objek `StringBuffer`
 9. Baris 21 menjelaskan pengambilan isi dari objek `StringBuffer` untuk disimpan di `temp`
 10. Baris 23 menjelaskan penampilan informasi error ke user dengan isi pesan "Koneksi jaringan error, coba lagi"
 11. Baris 25-27 menjelaskan penutupan koneksi HTTP, `InputStream` dan `OutputStream`
 12. Baris 28 menjelaskan nilai kembalian dari method ini adalah `temp`
- Representasi algoritma pencarian koleksi untuk `getData()` ditunjukkan dalam diagram alir pada Gambar 5.5 dan *pseudocode* pada Gambar 5.6.



Gambar 5.5 Diagram alir pencarian koleksi untuk `getData()`

Sumber: Implementasi

METHOD getData PARAMETER plain

DECLARATION:

TYPE

plain, temp IS String
index, r, s, p IS int

DESCRIPTION:

```

1  index <- 0
2  r <- 0
3  s <- 0
4  p <- 0
5  WHILE index < panjang data plain
6    temp <- ""
7    WHILE karakter ke-index != "|"
8      temp = temp + karakter plain ke-index
9      INCREMENT index
10   ENDWHILE
11   content[r] <- temp
12   INCREMENT r
13   INCREMENT index
14 ENDWHILE
15 status <- content[0]
16 count <- content[1]
17 FOR i=2 to r DO
18   IF i mod 2 == 0 THEN
19     id[s] <- content[i]
20     INCREMENT s
21   ELSE
22     judul[p] <- content[i]
23     INCREMENT p
24   ENDIF
25 ENDFOR

```

Gambar 5.6 Algoritma pencarian koleksi untuk getData()

Sumber: Implementasi

Penjelasan algoritma pencarian koleksi untuk getData() pada Gambar 5.6

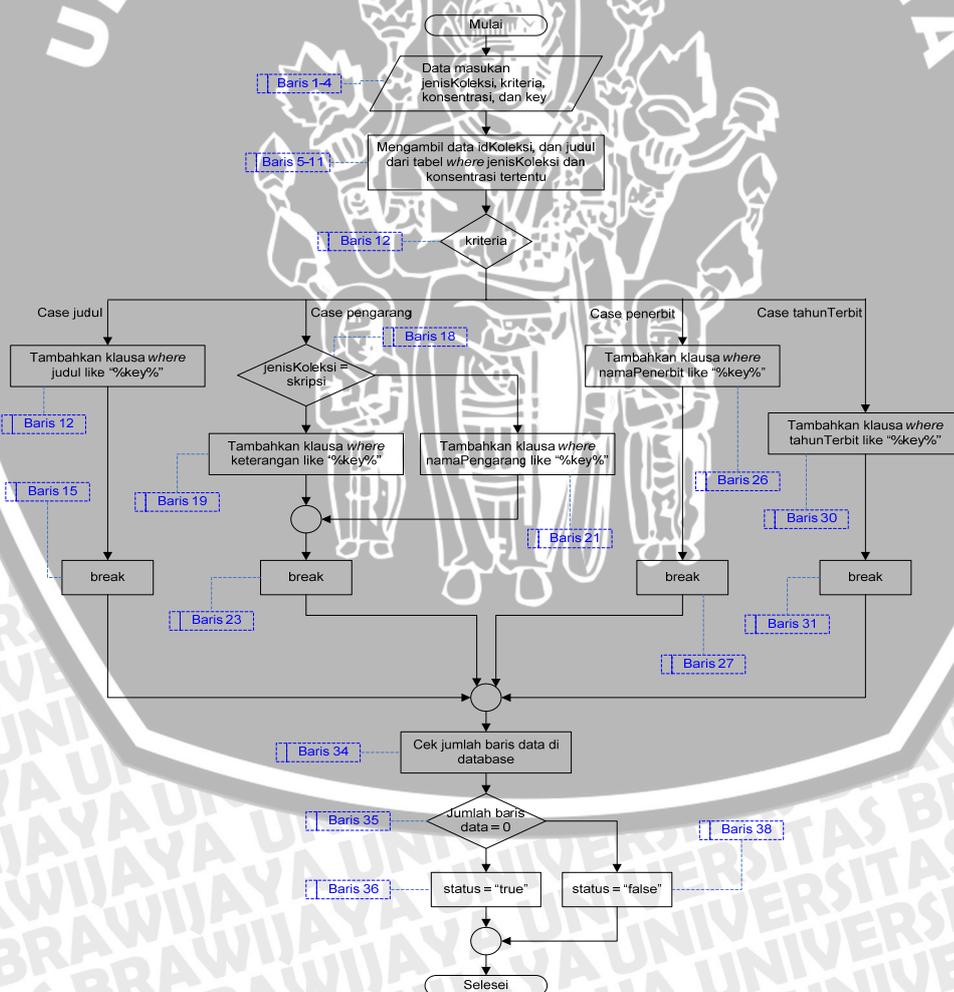
yaitu:

1. Baris 1-4 menjelaskan pemberian nilai ke index, r, s dan p.
2. Baris 5 menjelaskan pemeriksaan terhadap panjang data plain
3. Baris 6 menjelaskan pemberian nilai ke String temp
4. Baris 7-10 menjelaskan penyimpanan karakter penyusun plain ke dalam String temp sampai ditemukan karakter pemisah “|”
5. Baris 11 menjelaskan penyimpanan String temp ke String array content
6. Baris 15 menjelaskan penyimpanan String array content *index* ke 0 ke dalam String status
7. Baris 16 menjelaskan penyimpanan String array content *index* ke 1 ke dalam String count

8. Baris 17-25 menjelaskan penyimpanan data content (dimulai index ke-2) ke dalam String id apabila sisa nilai index dibagi dengan 2 hasilnya adalah 0 dan disimpan ke String judul apabila sisa nilai index dibagi dengan 2 hasilnya adalah tidak sama dengan 0

Algoritma pencarian koleksi pada sisi *server* diimplementasikan dengan *script* PHP. *Server* yang dijalankan dengan Apache menerima *request* dan memproses data yang dikirimkan dengan bantuan *script* PHP. *Script* PHP juga melakukan komunikasi dengan basis data MySQL dan memberikan hasilnya dalam format data *plain text* (data sederhana). Algoritma ini diterapkan pada *class* Catalog.

Representasi algoritma pencarian koleksi untuk `selectCollection()` ditunjukkan dalam diagram alir pada Gambar 5.7 dan *pseudocode* pada Gambar 5.8.



Gambar 5.7 Diagram alir pencarian koleksi untuk `selectCollection()`

Sumber: Implementasi

```

METHOD selectCollection PARAMETER jenisKoleksi, kriteria, konsentrasi, key
DECLARATION:
TYPE
jenisKoleksi, kriteria, konsentrasi, key IS String
Data IS int
DESCRIPTION:
1  jenisKoleksi <- $_POST["jenisKoleksi"]
2  kriteria <- $_POST["kriteria"]
3  konsentrasi <- $_POST["konsentrasi"]
4  key <- $_POST["key"]
5  Query database table koleksi k,paket p,penerbit pn,pengarang pg,
6      pengarangkoleksi pgk,paketkoleksi pk,jeniskoleksi jk
7  get k.idKoleksi,k.judul
8  where pk.idKoleksi=k.idKoleksi and pk.idPaket=p.idPaket and
9      pgk.idKoleksi=k.idKoleksi and pgk.idPengarangl=pg.idPengarang and
10     k.idPenerbit=pn.idPenerbit and k.idJenisKoleksi=jk.idJenisKoleksi
11     and jk.idJenisKoleksi="jenisKoleksi" and p.idPaket="konsentrasi"
12 SWITCH kriteria
13     CASE judul OF
14         Append where k.judul like "%key%"
15     BREAK
16     ENDCASE
17     CASE pengarang OF
18         IF jenisKoleksi = skripsi THEN
19             Append where k.keterangan like "%key%"
20         ELSE
21             Append where pg.namaPengarang like "%key%"
22         ENDIF
23     BREAK
24     ENDCASE
25     CASE penerbit OF
26         Append where pn.namaPenerbit like "%key%"
27     BREAK
28     ENDCASE
29     CASE tahun terbit OF
30         Append where k.tahunTerbit like "%key%"
31     BREAK
32     ENDCASE
33 ENDSWITCH
34 Data <- Cek jumlah baris data di database
35 IF Data != 0 THEN
36     status <- "true"
37 ELSE
38     status <- "false"
39 ENDIF

```

Gambar 5.8 Algoritma pencarian koleksi untuk selectCollection()

Sumber: Implementasi

Penjelasan algoritma pencarian koleksi selectCollection() pada Gambar 5.8 yaitu:

1. Baris 1-4 menjelaskan masukan data dari *client* berupa data jenisKoleksi, kriteria, konsentrasi, dan key.
2. Baris 5-11 menjelaskan pembuatan *query* ke basis data untuk mendapatkan data idKoleksi dan judul berdasarkan jenis koleksi dan konsentrasi yang dipilih oleh *user*. Pencarian ini melibatkan tabel koleksi, paket, penerbit, pengarang, pengarangkoleksi, paketkoleksi, dan jeniskoleksi.

3. Baris 12-33 menjelaskan penambahan *query* basis data klausa *where* berdasarkan kriteria tertentu seperti judul, pengarang, penerbit, dan tahun terbit
4. Baris 34 menjelaskan pemeriksaan jumlah baris data yang ada di basis data
5. Baris 35-36 menjelaskan pemeriksaan terhadap jumlah baris data. Apabila jumlah baris data yang ada tidak sama dengan 0 ($Data \neq 0$) maka nilai status adalah true
6. Baris 37-39 menjelaskan aliran alternatif apabila jumlah baris data adalah 0 maka nilai status adalah false

5.5 Implementasi Antarmuka Aplikasi

Aplikasi Sistem Katalog Perpustakaan berbasis J2ME mempunyai 8 buah antarmuka utama yaitu Menu Utama Guest, Menu Utama Member, Login untuk masuk ke dalam Menu Utama Member dan dapat menggunakan fasilitas pencarian informasi katalog perpustakaan, Register untuk mendaftar sebagai anggota sehingga dapat menggunakan keseluruhan fasilitas dari sistem, Forgot Password untuk membantu pengguna mendapatkan kembali password yang valid saat mereka lupa dengan *password* yang dimilikinya, Property System untuk mengetahui informasi spesifikasi handphone meliputi CLDC dan MIDP *version*, Catalog untuk melakukan proses pencarian informasi katalog perpustakaan, Catalog Detail untuk melihat keseluruhan informasi mengenai koleksi pustaka yang dipilih, Account Password untuk mengubah password yang dimiliki oleh pengguna.

5.5.1 Implementasi Antarmuka Menu Utama Guest

Antarmuka Menu Utama Guest ini ditampilkan ketika pengguna pertama kali menjalankan aplikasi. Menu Utama Guest menyediakan menu atau proses yang harus dilakukan oleh pengguna sebelum bisa masuk ke Menu Utama Member. Tampilan antarmuka Menu Utama Guest ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Antarmuka aplikasi Menu Utama Guest

Sumber: Implementasi

Gambar 5.9 merupakan gambar hasil implementasi antarmuka Menu Utama Guest. Penjelasan masing-masing menu dari antarmuka Menu Utama Guest diuraikan pada Tabel 5.4

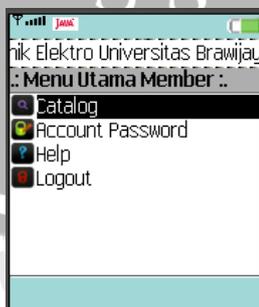
Tabel 5.4 Penjelasan menu pada antarmuka Menu Utama Guest

No	Label Menu	Penjelasan
1	Login	Menu ini digunakan untuk melakukan poses autentifikasi terhadap pengguna yang ingin mengakses semua fungsionalitas sistem
2	Register	Menu ini digunakan untuk membuat akun baru sebagai member sebelum melakukan proses login
3	Forgot Password	Menu ini digunakan untuk mendapatkan password yang valid saat pengguna lupa dengan <i>password</i> yang dimilikinya
4	Property System	Menu ini digunakan untuk melihat informasi spesifikasi sistem <i>mobile phone</i> dan requirement aplikasi
5	Exit	Menu ini digunakan untuk keluar dari aplikasi

Sumber : Implementasi

5.5.2 Implementasi Antarmuka Menu Utama Member

Antarmuka Menu Utama Member ditampilkan setelah pengguna berhasil melakukan Login untuk proses autentifikasi terhadap pengguna. Tampilan antarmuka Menu Utama Member ditunjukkan dalam Gambar 5.10.



Gambar 5.10 Antarmuka aplikasi Menu Utama Member

Sumber: Implementasi

Gambar 5.10 merupakan gambar hasil implementasi antarmuka Menu Utama Member. Penjelasan masing-masing menu dari antarmuka Menu Utama Member diuraikan pada Tabel 5.5

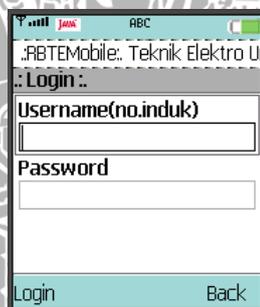
Tabel 5.5 Penjelasan menu pada antarmuka Menu Utama Member

No	Label Menu	Penjelasan
1	Catalog	Menu ini digunakan untuk melakukan proses pencarian informasi katalog perpustakaan
2	Account Password	Menu ini digunakan untuk mengubah <i>password</i> yang dimiliki oleh pengguna
3	Help	Menu ini menyediakan petunjuk penggunaan aplikasi
4	Logout	Menu ini digunakan untuk keluar dari aplikasi

Sumber : Implementasi

5.5.3 Implementasi Antarmuka Login

Antarmuka aplikasi Login menyediakan fasilitas untuk proses autentikasi terhadap pengguna agar dapat masuk dan menggunakan fasilitas sistem sebagai *member* dengan memenuhi kolom isian *username*(no.induk) dan *password*. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Antarmuka aplikasi Login

Sumber: Implementasi

Gambar 5.11 merupakan gambar hasil implementasi antarmuka Login. Penjelasan masing-masing tombol dari antarmuka Login diuraikan pada Tabel 5.6

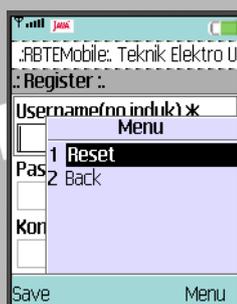
Tabel 5.6 Penjelasan menu pada antarmuka Login

No	Label Tombol	Penjelasan
1	Login	Menu ini digunakan untuk melakukan proses autentikasi terhadap pengguna yang ingin masuk ke dalam Menu Utama Member. Pengguna harus mengisi <i>field username</i> (no.induk) dan <i>password</i> yang valid atau sesuai dengan data hasil proses Registrasi
2	Back	Menu ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Guest

Sumber : Implementasi

5.5.4 Implementasi Antarmuka Register

Antarmuka Register digunakan untuk membuat akun baru sebagai *member*. Akun baru tersebut dapat mengakses semua fungsionalitas sistem. *Field* isian yang harus diisi oleh pengguna adalah *username(no.induk)*, *password*, konfirmasi *password*, nama, alamat, jenis kelamin, tanggal lahir, tempat lahir, no. telepon, email, jenis tanda pengenal, no. tanda pengenal, jenis anggota. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.12.



Gambar 5.12 Antarmuka aplikasi Register
Sumber: Implementasi

Gambar 5.12 merupakan gambar hasil implementasi antarmuka Register. Penjelasan masing-masing tombol dari antarmuka Register diuraikan pada Tabel 5.7

Tabel 5.7 Penjelasan menu pada antarmuka Register

No	Label Tombol	Penjelasan
1	Save	Menu ini digunakan untuk menyimpan data pengguna ke dalam basis data MySQL. Jika proses penyimpanan data berhasil, maka pengguna dapat melakukan proses login dan masuk ke Menu Utama Member. Pengguna harus mengisi <i>field username(no.induk)</i> , <i>password</i> , konfirmasi <i>password</i> , nama, alamat, jenis kelamin, tanggal lahir, tempat lahir, no.telpon, enail, jenis tanda pengenal, dan jenis anggota
2	Reset	Menu ini digunakan untuk mengkosongkan semua field pada halaman Register.
3	Back	Menu ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Guest

Sumber : Implementasi

5.5.5 Implementasi Antarmuka Forgot Password

Antarmuka Forgot Password menyediakan fasilitas untuk membantu pengguna mendapatkan kembali password yang valid saat mereka lupa dengan password yang dimilikinya. Antarmuka Forgot Password terdapat *field* isian

username, nama, dan no. tanda pengenal yang harus diisi oleh pengguna. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.13.



Gambar 5.13 Antarmuka aplikasi Forgots Password
Sumber: Implementasi

Gambar 5.13 merupakan gambar hasil implementasi antarmuka Forgots Password. Penjelasan masing-masing tombol dari antarmuka Forgots Password diuraikan pada Tabel 5.8

Tabel 5.8 Penjelasan menu pada antarmuka Forgots Password

No	Label Tombol	Penjelasan
1	Send	Tombol ini digunakan untuk melihat <i>password</i> yang valid jika pengguna lupa dengan <i>password</i> yang dimilikinya. Pengguna harus mengisi <i>field username(no.induk)</i> , nama, dan no. tanda pengenal
2	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Guest

Sumber : Implementasi

5.5.6 Implementasi Antarmuka Catalog

Antarmuka Catalog menyediakan fasilitas untuk melakukan proses pencarian informasi katalog perpustakaan dengan mengisi *field* jenis koleksi, kriteria, konsentrasi, dan key. *Field* jenis koleksi, kriteria dan konsentrasi harus diisi oleh pengguna sedangkan *field* key bersifat optional. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.14.



Gambar 5.14 Antarmuka aplikasi Catalog
Sumber: Implementasi



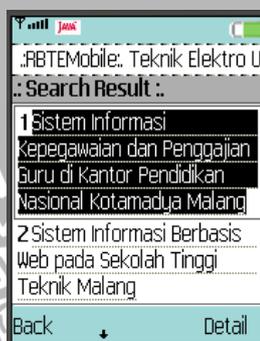
Gambar 5.14 merupakan gambar hasil implementasi antarmuka Catalog. Penjelasan masing-masing tombol dari antarmuka Catalog diuraikan pada Tabel 5.9

Tabel 5.9 Penjelasan menu pada antarmuka Catalog

No	Label Tombol	Penjelasan
1	Search	Tombol ini digunakan untuk melakukan proses pencarian informasi katalog perpustakaan. Pengguna harus mengisi <i>field</i> jenis koleksi, kriteria, konsentrasi, dan kata kunci
2	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Member

Sumber : Implementasi

Tampilan antarmuka untuk menampilkan hasil pencarian ditunjukkan pada Gambar 5.15.



Gambar 5.15 Antarmuka aplikasi Search Result

Sumber: Implementasi

Gambar 5.15 merupakan gambar hasil implementasi antarmuka Search Result. Penjelasan masing-masing tombol dari antarmuka Search Result diuraikan pada Tabel 5.10

Tabel 5.10 Penjelasan menu pada antarmuka Search Result

No	Label Tombol	Penjelasan
1	Detail	Tombol ini digunakan untuk melihat informasi koleksi secara lebih detail. Pengguna harus memilih salah satu judul yang terdapat pada halaman Search Result
2	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Member

Sumber : Implementasi

Apabila pengguna menginginkan untuk melihat informasi koleksi secara lebih detail maka pengguna dapat memilih menu Detail. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.16.





Gambar 5.16 Antarmuka aplikasi Catalog Detail
Sumber: Implementasi

Gambar 5.16 merupakan gambar hasil implementasi antarmuka Catalog Detail. Penjelasan masing-masing tombol dari antarmuka Catalog Detail diuraikan pada Tabel 5.11.

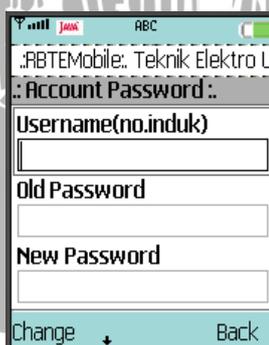
Tabel 5.11 Penjelasan menu pada antarmuka Catalog Detail

No	Label Tombol	Penjelasan
1	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu halaman Catalog

Sumber : Implementasi

5.5.7 Implementasi Antarmuka Account Password

Antarmuka Account Password menyediakan fasilitas untuk dapat mengubah *password* akun *Member*. Antarmuka Account Password menyediakan *field* isian *username*(no.induk), *password* lama, *password* baru, dan konfirmasi *password* baru. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.17.



Gambar 5.17 Antarmuka aplikasi Account Password
Sumber: Implementasi

Gambar 5.17 merupakan gambar hasil implementasi antarmuka Account Password. Penjelasan masing-masing tombol dari antarmuka Account Password diuraikan pada Tabel 5.12



Tabel 5.12 Penjelasan menu pada antarmuka Account Password

No	Label Tombol	Penjelasan
1	Change	Tombol ini digunakan untuk mengubah <i>password</i> akun <i>member</i> . Pengguna harus mengisi <i>field username</i> (no.induk), <i>password</i> lama, <i>password</i> baru, dan konfirmasi <i>password</i> baru
2	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Member

Sumber : Implementasi

5.5.8 Implementasi Antarmuka Property System

Antarmuka Property System menyediakan fasilitas untuk dapat melihat informasi spesifikasi sistem pada handphone dan konfigurasi (CLDC dan MIDP) standar yang harus dipenuhi untuk menjalankan aplikasi sistem katalog perpustakaan ini. Tampilan antarmuka untuk aplikasi ini ditunjukkan dalam Gambar 5.18.



Gambar 5.18 Antarmuka aplikasi Property System

Sumber: Implementasi

Gambar 5.18 merupakan gambar hasil implementasi antarmuka Property System. Penjelasan masing-masing tombol dari antarmuka Property System diuraikan pada Tabel 5.13

Tabel 5.13 Penjelasan menu pada antarmuka Property System

No	Label Tombol	Penjelasan
1	Back	Tombol ini digunakan untuk kembali ke tampilan sebelumnya yaitu Menu Utama Member

Sumber : Implementasi

BAB VI

PENGUJIAN PERANGKAT LUNAK

Pada bab ini dilakukan proses pengujian dan analisis terhadap sistem katalog perpustakaan berbasis mobile yang telah dibangun. Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian integrasi dan pengujian validasi. Pada pengujian unit dan pengujian integrasi, akan digunakan teknik pengujian *White Box (White Box Testing)*. Pada pengujian validasi akan digunakan teknik pengujian *Black Box (Black Box Testing)*.

6.1 Pengujian

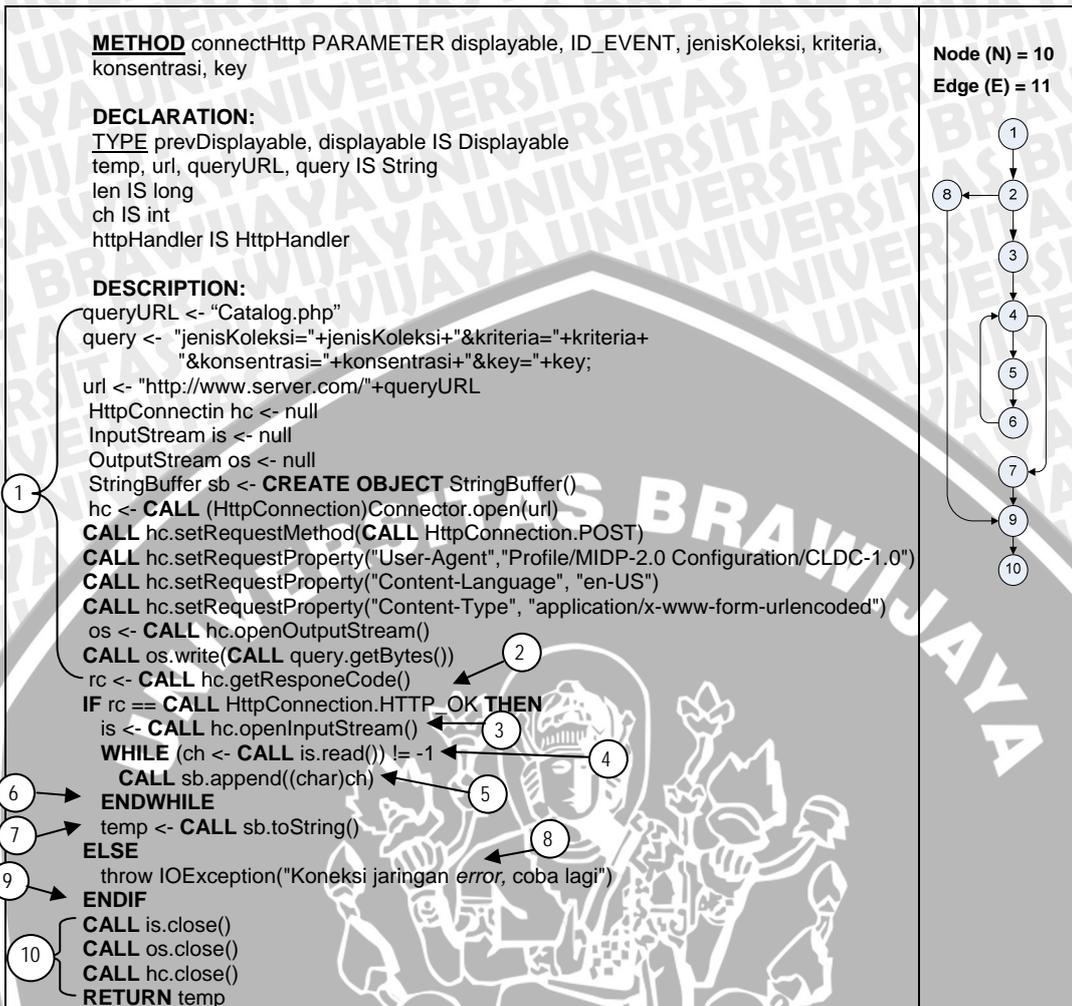
Proses pengujian dilakukan melalui empat tahapan (strategi) yaitu pengujian unit, pengujian integrasi, pengujian validasi dan pengujian perfonmansi koneksi.

6.1.1 Pengujian Unit

Sistem yang dibangun dengan paradigma pemrograman berorientasi objek menerapkan pengujian unit untuk suatu metode (operasi) dari suatu kelas. Pada pengujian unit sistem ini, digunakan teknik pengujian *White Box (White Box Testing)* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Penulisan laporan skripsi ini hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

a. Pengujian Unit untuk Operasi `connectHttp()`

Operasi `connectHttp()` merupakan implementasi metode pembantu dari algoritma pencarian koleksi. Operasi (metode) `connectHttp()` terdapat di dalam klas `HttpHandler`. Gambar 6.1 memperlihatkan proses pemodelan dalam *flow graph* pada operasi `connectHttp()`.



Gambar 6.1 Pemodelan algoritma connectHttp() ke dalam flow graph

Sumber: Pengujian

Pemodelan ke dalam flow graph yang telah dilakukan terhadap operasi connectHttp() menghasilkan jumlah kompleksitas siklomatis (cyclomatic complexity) melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 11 - 10 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai cyclomatic complexity yang telah dihasilkan dari perhitungan yaitu 2 ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-3-4-5-6-4-7-9-10

Jalur 2 : 1-2-3-4-7-9-10

Jalur 3 : 1-2-8-9-10

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.1.



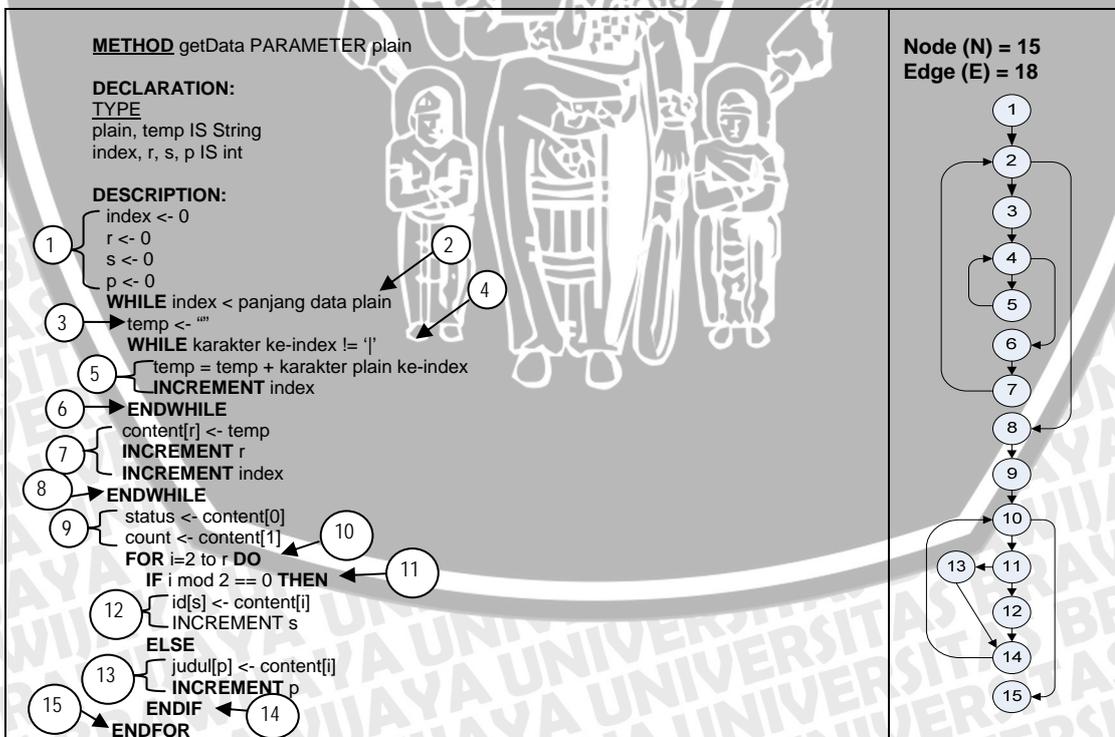
Tabel 6.1 Test case untuk pengujian unit operasi connectHttp()

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "informatika" key = ""	Semua data karakter hasil dari respon server ditambahkan dalam StringBuffer dan disimpan di temp	Semua data karakter hasil dari respon server ditambahkan dalam StringBuffer dan disimpan di temp
2	jenisKoleksi = "" kriteria = "" konsentrasi = "" key = ""	Isi dari temp adalah kosong apabila pemeriksaan nilai pengembalian dari pembacaan byte data respon server oleh input stream adalah -1	Isi dari temp adalah kosong apabila pemeriksaan nilai pengembalian dari pembacaan byte data respon server oleh input stream adalah -1
3	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "informatika" key = ""	Sistem menampilkan pesan koneksi jaringan <i>error</i> apabila kode respon server tidak sama dengan 200 (HTTP_OK)	Sistem menampilkan pesan koneksi jaringan <i>error</i> apabila kode respon server tidak sama dengan 200 (HTTP_OK)

Sumber: Pengujian

b. Pengujian Unit untuk Operasi getData()

Operasi `getData()` merupakan implementasi metode pembantu dari algoritma pencarian koleksi. Operasi (metode) `getData()` terdapat di dalam kelas `CatalogParser`. Gambar 6.2 memperlihatkan proses pemodelan dalam *flow graph* pada operasi `getData()`.



Gambar 6.2 Pemodelan algoritma `getData()` ke dalam *flow graph*

Sumber: Pengujian



Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `getData()` menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 18 - 15 + 2 \\ &= 5 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G)=5$ ditentukan lima buah ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-3-4-5-4-6-7-2-8-9-10-11-12-14-10-15

Jalur 2 : 1-2-3-4-5-4-6-7-2-8-9-10-11-13-14-10-15

Jalur 3 : 1-2-8-9-10-11-12-14-10-15

Jalur 4 : 1-2-3-4-6-7-2-8-9-10-11-12-14-10-15

Jalur 5 : 1-2-3-4-5-4-6-7-2-8-9-10-15

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.2.

Tabel 6.2 Test case untuk pengujian unit operasi `getData()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	plain = "true 1 13 Aplikasi Sistem Informasi pada Kejuaraan Renang di Malang"	Nilai id = 13	Nilai id = 13
2	plain = "true 1 13 Aplikasi Sistem Informasi pada Kejuaraan Renang di Malang"	Nilai judul = Aplikasi Sistem Informasi pada Kejuaraan Renang di Malang	Nilai judul = Aplikasi Sistem Informasi pada Kejuaraan Renang di Malang
3	plain = ""	id dan judul tidak mempunyai nilai	id dan judul tidak mempunyai nilai
4	plain = "true113Aplikasi Sistem Informasi pada Kejuaraan Renang di Malang"	id dan judul tidak mempunyai nilai	id dan judul tidak mempunyai nilai
5	plain = "true 1 13 "	id dan judul tidak mempunyai nilai	id dan judul tidak mempunyai nilai

Sumber: Pengujian

c. Pengujian Unit untuk Operasi `selectCollection()`

Operasi `selectCollection()` merupakan implementasi metode pembantu dari algoritma pencarian koleksi. Operasi (metode)

selectCollection() terdapat di dalam klas Catalog. Gambar 6.3 memperlihatkan proses pemodelan dalam *flow graph* pada operasi selectCollection().



Gambar 6.3 Pemodelan algoritma selectCollection() ke dalam *flow graph*
Sumber: Pengujian

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi selectCollection() menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 24 - 20 + 2 \\
 &= 6
 \end{aligned}$$



Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G)=6$ ditentukan enam buah ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-3-4-15-16-17-18-20

Jalur 2 : 1-2-3-4-15-16-17-19-20

Jalur 3 : 1-2-3-6-7-8-10-15-16-17-18-20

Jalur 4 : 1-2-3-6-7-9-10-15-16-17-18-20

Jalur 5 : 1-2-3-11-12-15-16-17-18-20

Jalur 6 : 1-2-3-13-14-15-16-17-18-20

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.3.

Tabel 6.3 Test case untuk pengujian unit operasi `selectCollection()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "teknik informatika" key = "sistem informasi"	Data skripsi konsentrasi teknik informatika dengan judul sistem informasi tersedia di basis data dan status bernilai true	Data skripsi konsentrasi teknik informatika dengan judul sistem informasi tersedia di basis data dan status bernilai true
2	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "teknik informatika" key = "xxxxxx"	Data skripsi konsentrasi teknik informatika dengan judul xxxxxx tidak tersedia di basis data dan status bernilai false	Data skripsi konsentrasi teknik informatika dengan xxxxxx tidak tersedia di basis data dan status bernilai false
3	jenisKoleksi = "skripsi" kriteria = "pengarang" konsentrasi = "informatika" key = "satrio"	Data skripsi konsentrasi teknik informatika oleh satrio tersedia di basis data dan status bernilai true	Data skripsi konsentrasi teknik informatika oleh satrio tersedia di basis data dan status bernilai true
4	jenisKoleksi = "referensi" kriteria = "pengarang" konsentrasi = "teknik tenaga listrik" key = ""	Data referensi konsentrasi teknik tenaga listrik berdasarkan semua pengarang tersedia di basis data dan status bernilai true	Data referensi konsentrasi teknik tenaga listrik berdasarkan semua pengarang tersedia di basis data dan status bernilai true
5	jenisKoleksi = "referensi" kriteria = "penerbit" konsentrasi = "teknik elektronika" key = "McGraw"	Data referensi konsentrasi teknik elektronika dengan penerbit McGraw tersedia di basis data dan status bernilai true	Data referensi konsentrasi teknik elektronika dengan penerbit McGraw tersedia di basis data dan status bernilai true
6	jenisKoleksi = "skripsi" kriteria = "tahun terbit" konsentrasi = "teknik informatika" key = "2007"	Data skripsi konsentrasi teknik informatika dengan tahun terbit 2007 tersedia di basis data dan status bernilai true	Data skripsi konsentrasi teknik informatika dengan tahun terbit 2007 tersedia di basis data dan status bernilai true

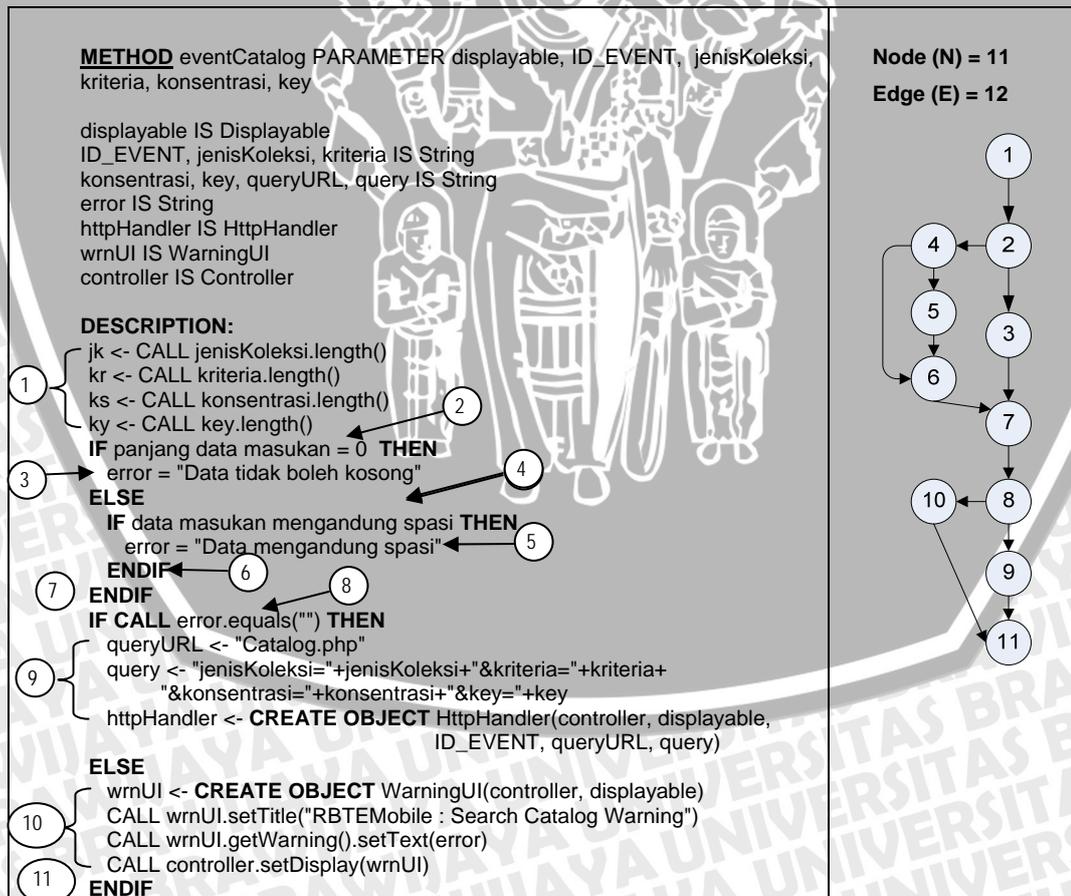
Sumber: Pengujian

6.1.2 Pengujian Integrasi

Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa kelas untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi yang dijadikan sebagai obyek uji adalah kelas-kelas yang menggabungkan kinerja dari kelas-kelas yang lain. Pengujian integrasi dilakukan terhadap sistem ini menggunakan strategi *bottom-up*, dimana modul-modul yang diintegrasikan masing-masing diuji terlebih dahulu dalam pengujian unit dan kemudian bergerak menuju ke pengujian modul-modul kontrol yang mengintegrasikannya.

a. Pengujian Integrasi untuk Operasi eventCatalog()

Operasi eventCatalog() merupakan implementasi metode pembantu dari algoritma proses pencarian koleksi. Operasi (metode) eventCatalog() terdapat di dalam kelas Controller. Gambar 6.4 memperlihatkan proses pemodelan dalam *flow graph* pada operasi eventCatalog().



Gambar 6.4 Pemodelan algoritma eventCatalog() ke dalam flow graph

Sumber: Pengujian

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `eventCatalog()` menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 13 - 11 + 2 \\ &= 4 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G)=4$ ditentukan empat buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-3-7-8-10-11

Jalur 2 : 1-2-4-5-6-7-8-10-11

Jalur 3 : 1-2-4-6-7-8-9-11

Jalur 4 : 1-2-4-6-7-8-10-11

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.4.

Tabel 6.4 Test case untuk pengujian integrasi operasi `eventCatalog()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	jenisKoleksi = "" kriteria = "judul" konsentrasi = "" key = ""	Sistem menampilkan halaman wrnUI karena data masukan jenisKoleksi dan konsentrasi adalah kosong	Sistem menampilkan halaman wrnUI dengan isi pesan adalah error
2	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "Informatika" key = " "	Sistem menampilkan halaman wrnUI karena data masukan key mengandung spasi	Sistem menampilkan halaman wrnUI dengan isi pesan adalah error
3	jenisKoleksi = "skripsi" kriteria = "judul" konsentrasi = "Informatika" key = ""	Pembuatan objek httpHandler	Pembuatan objek httpHandler
4	Tidak dapat diuji secara mandiri dan merupakan satu rangkaian dengan jalur 3	-	-

Sumber: Pengujian

6.1.3 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi

menggunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan.

6.1.3.1 Kasus Uji Validasi

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan kasus uji masing-masing.

1. Kasus Uji Membuat Akun Member

Tabel berikut menunjukkan kasus uji membuat akun member.

Tabel 6.5 Kasus Uji Membuat Akun Member

Nama kasus uji	Kasus uji membuat akun member
Obyek uji	Kebutuhan nomor 1 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk membuat akun member
Prosedur uji	<ol style="list-style-type: none"> 1. Membuka antarmuka "Register" pada halaman "Menu Utama Guest" 2. Mengisi <i>field</i> "Username(no.induk)", "Password", "Konfirmasi Password", "Nama", "Alamat", "Jenis Kelamin", "Tanggal Lahir", "Tempat Lahir", "No. Telepon", "Email", "Jenis Tanda Pengenal", "No. Tanda Pengenal", dan "Jenis Anggota". 3. Menekan tombol 'Save'.
Hasil yang diharapkan	Aplikasi dapat melakukan proses pembuatan akun Member dan menampilkan informasi bahwa pembuatan akun berhasil dilakukan.

Sumber: Pengujian

2. Kasus Uji Login

Tabel berikut menunjukkan kasus uji login.

Tabel 6.6 Kasus Uji Login

Nama kasus uji	Kasus uji login
Obyek uji	Kebutuhan nomor 2 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk login sehingga pengguna dapat menggunakan fasilitas untuk Member
Prosedur uji	<ol style="list-style-type: none"> 1. Membuka antarmuka "Login" pada halaman "Menu Utama Guest" 2. Mengisi <i>field</i> "Username(no.induk)", dan "Password". 3. Menekan tombol 'Login'.
Hasil yang diharapkan	Aplikasi dapat melakukan validasi <i>username</i> (no.induk) dan <i>password</i> yang diisi oleh pengguna dan menampilkan halaman "Menu Utama Member".

Sumber: Pengujian

3. Kasus Uji Lupa Password

Tabel berikut menunjukkan kasus uji Lupa Password.

Tabel 6.7 Kasus Uji Lupa Password

Nama kasus uji	Kasus uji lupa password
Obyek uji	Kebutuhan nomor 3 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk membantu pengguna mendapatkan kembali <i>password</i> yang valid saat mereka lupa dengan <i>password</i> yang dimilikinya
Prosedur uji	<ol style="list-style-type: none"> 1. Membuka antarmuka “Forgot Password” pada halaman “Menu Utama Guest” 2. Mengisi <i>field</i> ”Username(no.induk)”, “Nama”, dan “No. Tanda Pengenal”. 3. Menekan tombol “Send”.
Hasil yang diharapkan	Aplikasi dapat melakukan validasi <i>username</i> (no.induk), nama, dan no.tanda pengenal yang diisi oleh pengguna dan menampilkan <i>password</i> pengguna yang valid.

Sumber: Pengujian

4. Kasus Uji Melihat Daftar Koleksi

Tabel berikut menunjukkan kasus uji Melihat Daftar Koleksi.

Tabel 6.8 Kasus Uji Melihat Daftar Koleksi

Nama kasus uji	Kasus uji melihat daftar koleksi
Obyek uji	Kebutuhan nomor 4 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk melihat daftar semua koleksi skripsi, tesis, disertasi, referensi, diktat mata kuliah, jurnal, kamus, laporan PKL dan klipng yang ada di Ruang Baca Teknik Elektro.
Prosedur uji	<ol style="list-style-type: none"> 1. Membuka antarmuka “Catalog” pada halaman “Menu Utama Member” 2. Mengisi <i>field</i> ”Jenis Koleksi”, “Kriteria”, “Konsentrasi”, dan “Key”. 3. Menekan tombol “Search”.
Hasil yang diharapkan	Aplikasi dapat melakukan validasi Jenis Koleksi, Kriteria, Konsentrasi, dan Key yang diisi oleh pengguna dan menampilkan daftar koleksi hasil pencarian.

Sumber: Pengujian

5. Kasus Uji Melihat Detail Koleksi

Tabel berikut menunjukkan kasus uji Melihat Detail Koleksi.

Tabel 6.9 Kasus Uji Melihat Detail Koleksi

Nama kasus uji	Kasus uji melihat detail koleksi
Obyek uji	Kebutuhan nomor 5 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk melihat data koleksi secara lebih detail
Prosedur uji	<p>Berada pada halaman “Search Result” yang berisi daftar koleksi hasil pencarian</p> <p>Memilih salah satu judul koleksi yang tersedia</p> <p>Menekan tombol “Detail”.</p>
Hasil yang diharapkan	Aplikasi dapat menangkap id dari judul yang dipilih oleh pengguna dan menampilkan data koleksi secara lebih detail.

Sumber: Pengujian

6. Kasus Uji Mengubah Password Akun Member

Tabel berikut menunjukkan kasus uji Mengubah Password Akun Member.

Tabel 6.10 Kasus Uji Mengubah Password Akun Member

Nama kasus uji	Kasus uji mengubah password akun member
Obyek uji	Kebutuhan nomor 6 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk mengubah <i>password</i> yang dimiliki oleh Member
Prosedur uji	Membuka antarmuka “Account Password” pada halaman “Menu Utama Member” Mengisi field “Username(no.induk)”, “Old Password”, “New Password”, dan “Confirm Password” Menekan tombol “Change”.
Hasil yang diharapkan	Aplikasi dapat melakukan validasi Username(no.induk), Old Password, New Password, dan Confirm Password yang diisi oleh pengguna dan menyimpan perubahan password akun Member serta menampilkan informasi bahwa perubahan password berhasil dilakukan.

Sumber: Pengujian

7. Kasus Uji Logout

Tabel berikut menunjukkan kasus uji Logout.

Tabel 6.11 Kasus Uji Logout

Nama kasus uji	Kasus uji logout
Obyek uji	Kebutuhan nomor 8 modul pendukung sistem (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat memenuhi kebutuhan fungsional untuk keluar dari “Menu Utama Member”
Prosedur uji	Menekan menu “Logout” pada antarmuka “Menu Utama Member”
Hasil yang diharapkan	Aplikasi dapat keluar dari antarmuka “Menu Utama Member” dan kembali ke halaman “Menu Utama Guest”.

Sumber: Pengujian

6.1.3.2 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan 6.1.3.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.12.

Tabel 6.12 Hasil pengujian validasi

No	Kasus Uji	Hasil yang didapatkan	Status
1	Kasus Uji Membuat Akun Member	Aplikasi dapat melakukan proses pembuatan akun Member dan menampilkan informasi bahwa pembuatan akun berhasil dilakukan.	Valid
2	Kasus Uji Login	Aplikasi dapat melakukan validasi <i>username</i> (no.induk) dan <i>password</i> yang diisi oleh pengguna dan menampilkan halaman “Menu Utama Member”.	Valid
3	Kasus Uji Lupa Password	Aplikasi dapat melakukan validasi <i>username</i> (no.induk), nama, dan no.tanda pengenal yang diisi oleh pengguna dan menampilkan password pengguna yang valid.	Valid
4	Kasus Uji Melihat Daftar Koleksi	Aplikasi dapat melakukan validasi Jenis Koleksi, Kriteria, Konsentrasi, dan Key yang diisi oleh pengguna dan menampilkan daftar koleksi hasil pencarian.	Valid
5	Kasus Uji Melihat Detail Koleksi	Aplikasi dapat menangkap id dari judul yang dipilih oleh pengguna dan menampilkan data koleksi secara lebih detail.	Valid

6	Kasus Uji Mengubah Password Akun Member	Aplikasi dapat melakukan validasi Username(no.induk), Old Password, New Password, dan Confirm Password yang diisi oleh pengguna dan menyimpan perubahan password akun Member serta menampilkan informasi bahwa perubahan password berhasil dilakukan.	Valid
7	Kasus Uji Logout	Aplikasi dapat keluar dari antarmuka "Menu Utama Member" dan kembali ke halaman "Menu Utama Guest".	Valid

Sumber: Pengujian

6.1.4 Pengujian Performansi Koneksi

Pengujian performansi koneksi digunakan untuk mengetahui performa koneksi ke *local server*. Pengujian performansi koneksi terdiri dari pengujian waktu akses *query* dan pengujian performansi *Web Server*.

6.1.4.1 Pengujian Waktu Akses Query

Pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan dalam melakukan *query* pada basis data *db_rbte*. Prosedur yang dijalankan dengan mengakses *QueryTime.php*. File *QueryTime.php* dibuat khusus untuk melakukan pengujian waktu akses *query* dengan menggunakan fungsi *microtime()* pada PHP.

A. Proses Pengujian

1. Pengujian waktu akses *query* terhadap tabel *anggota_rbte*

Hasil pengujian waktu akses *query* dalam *milisecond(ms)* terhadap tabel *anggota_rbte* dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query select count(idAnggota) from anggota_rbte* selama 10 kali perulangan ditunjukkan pada Tabel 6.13.

Tabel 6.13 : Hasil pengujian waktu akses *query* terhadap tabel *anggota_rbte*

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.164	Query 1: waktu = 0.170999999999992	Query 1: waktu = 0.175000000000004	Query 1: waktu = 0.203999999999998
	Query 2: waktu = 0.155000000000002	Query 2: waktu = 0.180999999999999	Query 2: waktu = 0.175999999999995	Query 2: waktu = 0.217000000000002
	Query 3: waktu = 0.150999999999998	Query 3: waktu = 0.17	Query 3: waktu = 0.287000000000004	Query 3: waktu = 0.197
	Query 4: waktu = 0.152000000000001	Query 4: waktu = 0.195999999999997	Query 4: waktu = 0.180000000000007	Query 4: waktu = 0.195000000000006
	Query 5: waktu = 0.152000000000001	Query 5: waktu = 0.176000000000006	Query 5: waktu = 0.174000000000001	Query 5: waktu = 0.193999999999992
	Query 6: waktu = 0.157999999999999	Query 6: waktu = 0.168999999999997	Query 6: waktu = 0.171000000000003	Query 6: waktu = 0.195999999999997
	Query 7: waktu = 0.148000000000001	Query 7: waktu = 0.166999999999992	Query 7: waktu = 0.348000000000001	Query 7: waktu = 0.348000000000001



	Query 8: waktu = 0.15400000000002 Query 9: waktu = 0.139 Query 10: waktu = 0.14500000000001	Query 8: waktu = 0.166 Query 9: waktu = 0.1810000000001 Query 10: waktu = 0.17100000000003	0.17199999999995 Query 8: waktu = 0.19499999999995 Query 9: waktu = 0.19999999999998 Query 10: waktu = 0.193	Query 8: waktu = 0.20200000000004 Query 9: waktu = 0.193 Query 10: waktu = 0.18899999999999
Rata-Rata Waktu Query	0.1518	0.1748	0.1923	0.2135

Sumber : Pengujian

2. Pengujian waktu akses *query* terhadap tabel *pengarangkoleksi*

Hasil pengujian waktu akses *query* dalam *milisecond(ms)* terhadap tabel *pengarangkoleksi* dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query* `select count(idKoleksi) from pengarangkoleksi` selama 10 kali perulangan ditunjukkan pada Tabel 6.14.

Tabel 6.14 : Hasil pengujian waktu akses *query* terhadap tabel *pengarangkoleksi*

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.14700000000001 Query 2: waktu = 0.14500000000006 Query 3: waktu = 0.14200000000009 Query 4: waktu = 0.143 Query 5: waktu = 0.15799999999999 Query 6: waktu = 0.14000000000003 Query 7: waktu = 0.139 Query 8: waktu = 0.14200000000009 Query 9: waktu = 0.13800000000008 Query 10: waktu = 0.14199999999998	Query 1: waktu = 0.167 Query 2: waktu = 0.17400000000001 Query 3: waktu = 0.166 Query 4: waktu = 0.17 Query 5: waktu = 0.17800000000001 Query 6: waktu = 0.17 Query 7: waktu = 0.163 Query 8: waktu = 0.16400000000003 Query 9: waktu = 0.166 Query 10: waktu = 0.169	Query 1: waktu = 0.17499999999993 Query 2: waktu = 0.17400000000001 Query 3: waktu = 0.18600000000002 Query 4: waktu = 0.17400000000001 Query 5: waktu = 0.17199999999995 Query 6: waktu = 0.28099999999998 Query 7: waktu = 0.18600000000002 Query 8: waktu = 0.17500000000004 Query 9: waktu = 0.17100000000003 Query 10: waktu = 0.17600000000006	Query 1: waktu = 0.20799999999999 Query 2: waktu = 0.20599999999993 Query 3: waktu = 0.20199999999992 Query 4: waktu = 0.20500000000001 Query 5: waktu = 0.19800000000003 Query 6: waktu = 0.20200000000004 Query 7: waktu = 0.20299999999995 Query 8: waktu = 0.20200000000004 Query 9: waktu = 0.197 Query 10: waktu = 0.23200000000001
Rata-Rata Waktu Query	0.14360000000003	0.1687	0.187	0.20549999999999

Sumber : Pengujian

3. Pengujian waktu akses *query* terhadap tabel *paketkoleksi*

Hasil pengujian waktu akses *query* dalam *milisecond(ms)* terhadap tabel *paketkoleksi* dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query* `select count(idKoleksi) from paketkoleksi` selama 10 kali perulangan ditunjukkan pada Tabel 6.15.

Tabel 6.15 : Hasil pengujian waktu akses *query* terhadap tabel paket koleksi

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.14400000000003 Query 2: waktu = 0.14199999999998 Query 3: waktu = 0.14099999999995 Query 4: waktu = 0.14100000000006 Query 5: waktu = 0.16399999999994 Query 6: waktu = 0.14400000000003 Query 7: waktu = 0.14199999999998 Query 8: waktu = 0.143 Query 9: waktu = 0.143 Query 10: waktu = 0.14199999999998	Query 1: waktu = 0.18499999999999 Query 2: waktu = 0.20399999999998 Query 3: waktu = 0.15100000000001 Query 4: waktu = 0.226 Query 5: waktu = 0.14799999999998 Query 6: waktu = 0.15199999999999 Query 7: waktu = 0.143 Query 8: waktu = 0.15799999999999 Query 9: waktu = 0.14399999999998 Query 10: waktu = 0.14700000000001	Query 1: waktu = 0.17699999999998 Query 2: waktu = 0.17299999999998 Query 3: waktu = 0.17500000000004 Query 4: waktu = 0.17300000000009 Query 5: waktu = 0.25600000000003 Query 6: waktu = 0.17500000000004 Query 7: waktu = 0.17200000000006 Query 8: waktu = 0.17299999999998 Query 9: waktu = 0.18599999999991 Query 10: waktu = 0.224	Query 1: waktu = 0.17299999999998 Query 2: waktu = 0.27300000000008 Query 3: waktu = 0.17500000000004 Query 4: waktu = 0.278 Query 5: waktu = 0.17399999999999 Query 6: waktu = 0.17699999999998 Query 7: waktu = 0.17300000000009 Query 8: waktu = 0.18399999999996 Query 9: waktu = 0.19400000000003 Query 10: waktu = 0.20799999999999
Rata-Rata Waktu Query	0.14459999999999	0.16579999999999	0.18840000000001	0.2009

Sumber : Pengujian

4. Pengujian waktu akses *query* terhadap tabel pengarang

Hasil pengujian waktu akses *query* dalam *milisecond*(ms) terhadap tabel pengarang dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query* `select count(idPengarang) from pengarang` selama 10 kali perulangan ditunjukkan pada Tabel 6.16.

Tabel 6.16: Hasil pengujian waktu akses *query* terhadap tabel pengarang

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.17299999999998 Query 2: waktu = 0.17299999999998 Query 3: waktu = 0.23200000000001 Query 4: waktu = 0.17400000000001 Query 5: waktu = 0.166 Query 6: waktu = 0.15999999999999 Query 7: waktu = 0.18000000000001 Query 8: waktu =	Query 1: waktu = 0.17700000000001 Query 2: waktu = 0.19 Query 3: waktu = 0.17300000000001 Query 4: waktu = 0.364 Query 5: waktu = 0.17800000000001 Query 6: waktu = 0.18099999999999 Query 7: waktu = 0.17299999999998 Query 8: waktu = 0.17099999999998	Query 1: waktu = 0.20200000000004 Query 2: waktu = 0.19799999999992 Query 3: waktu = 0.34999999999996 Query 4: waktu = 0.20599999999993 Query 5: waktu = 0.19499999999995 Query 6: waktu = 0.22100000000003 Query 7: waktu = 0.19999999999998 Query 8: waktu =	Query 1: waktu = 0.21400000000005 Query 2: waktu = 0.30799999999997 Query 3: waktu = 0.21000000000004 Query 4: waktu = 0.20399999999998 Query 5: waktu = 0.19800000000003 Query 6: waktu = 0.20700000000007 Query 7: waktu = 0.20500000000001 Query 8: waktu =

	0.176999999999998 Query 9: waktu = 0.172999999999998 Query 10: waktu = 0.166999999999997	Query 9: waktu = 0.17 Query 10: waktu = 0.172000000000001	waktu = 0.193999999999992 Query 9: waktu = 0.186999999999994 Query 10: waktu = 0.194000000000003	waktu = 0.390999999999992 Query 9: waktu = 0.257999999999998 Query 10: waktu = 0.191999999999997
Rata-Rata Waktu Query	0.177499999999999	0.1949	0.214699999999997	0.2387

Sumber : Pengujian

5. Pengujian waktu akses query terhadap tabel penerbit

Hasil pengujian waktu akses query dalam *milisecond(ms)* terhadap tabel penerbit dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi query `select count(idPenerbit) from penerbit` selama 10 kali perulangan ditunjukkan pada Tabel 6.17.

Tabel 6.17: Hasil pengujian waktu akses query terhadap tabel penerbit

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.146000000000003 Query 2: waktu = 0.164999999999997 Query 3: waktu = 0.159999999999999 Query 4: waktu = 0.141999999999998 Query 5: waktu = 0.151000000000001 Query 6: waktu = 0.137 Query 7: waktu = 0.137 Query 8: waktu = 0.155000000000002 Query 9: waktu = 0.140000000000003 Query 10: waktu = 0.149999999999998	Query 1: waktu = 0.149999999999998 Query 2: waktu = 0.139 Query 3: waktu = 0.139 Query 4: waktu = 0.153000000000001 Query 5: waktu = 0.141999999999998 Query 6: waktu = 0.139999999999997 Query 7: waktu = 0.336 Query 8: waktu = 0.139999999999997 Query 9: waktu = 0.139 Query 10: waktu = 0.154999999999996	Query 1: waktu = 0.174999999999993 Query 2: waktu = 0.177000000000009 Query 3: waktu = 0.174000000000001 Query 4: waktu = 0.174000000000001 Query 5: waktu = 0.183999999999996 Query 6: waktu = 0.172999999999998 Query 7: waktu = 0.171999999999995 Query 8: waktu = 0.176000000000006 Query 9: waktu = 0.174000000000001 Query 10: waktu = 0.308000000000009	Query 1: waktu = 0.228999999999998 Query 2: waktu = 0.229000000000001 Query 3: waktu = 0.218800000000004 Query 4: waktu = 0.215000000000004 Query 5: waktu = 0.219999999999998 Query 6: waktu = 0.217779999999998 Query 7: waktu = 0.209000000000005 Query 8: waktu = 0.203999999999998 Query 9: waktu = 0.201999999999998 Query 10: waktu = 0.201999999999999
Rata-Rata Waktu Query	0.1483	0.163299999999999	0.188700000000001	0.214658

Sumber : Pengujian

6. Pengujian waktu akses query terhadap tabel paket

Hasil pengujian waktu akses query dalam *milisecond(ms)* terhadap tabel paket dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi query `select count(idPaket) from paket` selama 10 kali perulangan ditunjukkan pada Tabel 6.18.



Tabel 6.18: Hasil pengujian waktu akses *query* terhadap tabel paket

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.14899999999995 Query 2: waktu = 0.152000000000004 Query 3: waktu = 0.174000000000001 Query 4: waktu = 0.166999999999992 Query 5: waktu = 0.166 Query 6: waktu = 0.17 Query 7: waktu = 0.16499999999997 Query 8: waktu = 0.167000000000003 Query 9: waktu = 0.166 Query 10: waktu = 0.161000000000008	Query 1: waktu = 0.176000000000006 Query 2: waktu = 0.184999999999999 Query 3: waktu = 0.175000000000004 Query 4: waktu = 0.173000000000009 Query 5: waktu = 0.172999999999998 Query 6: waktu = 0.171999999999995 Query 7: waktu = 0.269999999999999 Query 8: waktu = 0.176000000000006 Query 9: waktu = 0.175999999999995 Query 10: waktu = 0.192000000000008	Query 1: waktu = 0.212 Query 2: waktu = 0.205 Query 3: waktu = 0.228 Query 4: waktu = 0.210000000000001 Query 5: waktu = 0.221 Query 6: waktu = 0.21999 Query 7: waktu = 0.212 Query 8: waktu = 0.219898 Query 9: waktu = 0.209999999999999 Query 10: waktu = 0.218	Query 1: waktu = 0.205 Query 2: waktu = 0.397 Query 3: waktu = 0.202999999999999 Query 4: waktu = 0.207 Query 5: waktu = 0.199 Query 6: waktu = 0.386 Query 7: waktu = 0.205 Query 8: waktu = 0.204 Query 9: waktu = 0.194 Query 10: waktu = 0.189
Rata-Rata Waktu Query	0.1637	0.186800000000002	0.2155888	0.2389

Sumber : Pengujian

7. Pengujian waktu akses *query* terhadap tabel jeniskoleksi

Hasil pengujian waktu akses *query* dalam *milisecond*(ms) terhadap tabel jeniskoleksi dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query* `select count(idJenisKoleksi) from jeniskoleksi` selama 10 kali perulangan ditunjukkan pada Tabel 6.19.

Tabel 6.19: Hasil pengujian waktu akses *query* terhadap tabel jeniskoleksi

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.145999999999998 Query 2: waktu = 0.143 Query 3: waktu = 0.139 Query 4: waktu = 0.137999999999997 Query 5: waktu = 0.139 Query 6: waktu = 0.155000000000002 Query 7: waktu = 0.141000000000006 Query 8: waktu = 0.140000000000003 Query 9:	Query 1: waktu = 0.165000000000001 Query 2: waktu = 0.162000000000000 Query 3: waktu = 0.165000000000003 Query 4: waktu = 0.162000000000003 Query 5: waktu = 0.161 Query 6: waktu = 0.160999999999997 Query 7: waktu = 0.163999999999998 Query 8: waktu =	Query 1: waktu = 0.186999999999999 Query 2: waktu = 0.178000000000001 Query 3: waktu = 0.174999999999998 Query 4: waktu = 0.176999999999998 Query 5: waktu = 0.176999999999998 Query 6: waktu = 0.197 Query 7: waktu = 0.176000000000001 Query 8: waktu =	Query 1: waktu = 0.205000000000001 Query 2: waktu = 0.209999999999993 Query 3: waktu = 0.296999999999999 Query 4: waktu = 0.218999999999997 Query 5: waktu = 0.198999999999995 Query 6: waktu = 0.208000000000001 Query 7: waktu = 0.210000000000004 Query 8:

	waktu = 0.139 Query 10: waktu = 0.137000000000005	0.160000000000003 Query 9: waktu = 0.162999999999998 Query 10: waktu = 0.160000000000000	0.176000000000001 Query 9: waktu = 0.174999999999998 Query 10: waktu = 0.195	waktu = 0.197 Query 9: waktu = 0.195999999999997 Query 10: waktu = 0.194999999999995
Rata-Rata Waktu Query	0.141700000000001	0.1623	0.1813	0.213599999999999

Sumber : Pengujian

8. Pengujian waktu akses *query* terhadap tabel koleksi

Hasil pengujian waktu akses *query* dalam *milisecond*(ms) terhadap tabel koleksi dengan jumlah data sebanyak 200, 400, 800 dan 1600 data entry dengan konfigurasi *query* `select count(idKoleksi) from koleksi` selama 10 kali perulangan ditunjukkan pada Tabel 6.20.

Tabel 6.20: Hasil pengujian waktu akses *query* terhadap tabel koleksi

Jumlah Data	200	400	800	1600
Hasil Waktu Query	Query 1: waktu = 0.168999999999997 Query 2: waktu = 0.166 Query 3: waktu = 0.151000000000001 Query 4: waktu = 0.152000000000004 Query 5: waktu = 0.151999999999999 Query 6: waktu = 0.146999999999995 Query 7: waktu = 0.149000000000001 Query 8: waktu = 0.149000000000001 Query 9: waktu = 0.148000000000004 Query 10: waktu = 0.148000000000004	Query 1: waktu = 0.174000000000001 Query 2: waktu = 0.175000000000004 Query 3: waktu = 0.175000000000004 Query 4: waktu = 0.172999999999998 Query 5: waktu = 0.187999999999997 Query 6: waktu = 0.175999999999995 Query 7: waktu = 0.174000000000001 Query 8: waktu = 0.174000000000001 Query 9: waktu = 0.172999999999998 Query 10: waktu = 0.179999999999996	Query 1: waktu = 0.174000000000001 Query 2: waktu = 0.174999999999998 Query 3: waktu = 0.234999999999999 Query 4: waktu = 0.280999999999998 Query 5: waktu = 0.180999999999999 Query 6: waktu = 0.180999999999999 Query 7: waktu = 0.184000000000002 Query 8: waktu = 0.175000000000004 Query 9: waktu = 0.174000000000001 Query 10: waktu = 0.174000000000001	Query 1: waktu = 0.232999999999993 Query 2: waktu = 0.205000000000001 Query 3: waktu = 0.203999999999998 Query 4: waktu = 0.201000000000001 Query 5: waktu = 0.197 Query 6: waktu = 0.205999999999993 Query 7: waktu = 0.193 Query 8: waktu = 0.195000000000006 Query 9: waktu = 0.201000000000001 Query 10: waktu = 0.207999999999999
Rata-Rata Waktu Query	0.153100000000001	0.176199999999999	0.1934	0.204299999999999

Sumber : Pengujian

B. Hasil Pengujian dan Analisis

Database server dinstall di *local server* sekaligus digunakan sebagai *client*. Basis data `db_rbt` pada *local server* dapat menangani permintaan *query* dari *client* dengan jumlah 1600 data. Waktu akses *query* memerlukan waktu rata-rata terlama 0.2389 ms. Tabel 6.21 menunjukkan waktu akses *query* dalam *milisecond*(ms) dengan 200, 400, 800 dan 1600 data entry untuk setiap tabel.

Tabel 6.21 : Tabel rata-rata pengujian waktu akses *query*

Nama Tabel	waktu akses dengan 200 data entry (ms)	waktu akses dengan 400 data entry (ms)	waktu akses dengan 800 data entry (ms)	waktu akses dengan 1600 data entry (ms)
anggota_rbte	0.1518	0.1748	0.1923	0.2135
pengarangkoleksi	0.143600000000003	0.1687	0.187	0.205499999999999
paketkoleksi	0.144599999999999	0.165799999999999	0.188400000000001	0.2009
pengarang	0.177499999999999	0.1949	0.214699999999997	0.2387
penerbit	0.1483	0.163299999999999	0.188700000000001	0.214658
paket	0.1637	0.186800000000002	0.2155888	0.2389
jeniskoleksi	0.141700000000001	0.1623	0.1813	0.213599999999999
koleksi	0.153100000000001	0.176199999999999	0.1934	0.204299999999999

Sumber : Pengujian

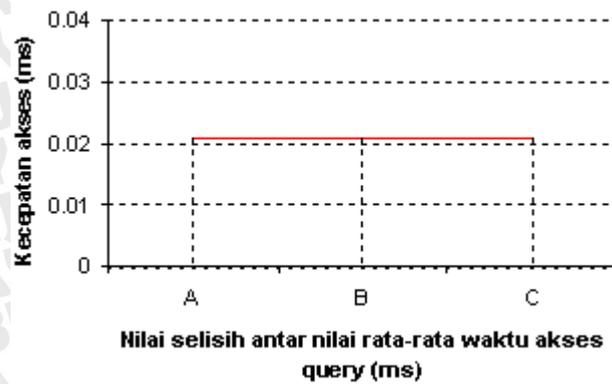
Berdasarkan Tabel 6.21 diperoleh nilai rata-rata waktu akses *query* dengan 200 data entry adalah $A_{200} = 0.1530375$ ms. Nilai rata-rata waktu akses *query* dengan 400 data entry adalah $A_{400} = 0.1741$ ms. Nilai rata-rata waktu akses *query* dengan 800 data entry adalah $A_{800} = 0.1951736$ ms. Nilai rata-rata waktu akses *query* dengan 1600 data entry adalah $A_{1600} = 0.21625725$ ms. Gambar 6.5 menunjukkan grafik hubungan antara data entry dan nilai rata-rata kecepatan waktu akses *query*.



Gambar 6.5 Grafik hubungan antara data entry dan rata-rata kecepatan waktu akses *query*
 Sumber : Pengujian

Nilai selisih antara nilai rata-rata waktu akses *query* dengan 400 data *entry* dan 200 data *entry* adalah $A = A_{400} - A_{200} = 0.1741 - 0.1530375 = 0.0210625$ ms. Nilai selisih antara nilai rata-rata waktu akses *query* dengan 800 data dan 400 data adalah $B = A_{800} - A_{400} = 0.1951736 - 0.1741 = 0.0210736$ ms. Nilai selisih antara nilai rata-rata waktu akses *query* dengan 1600 data dan 800 data adalah $C = A_{1600} - A_{800} = 0.21625725 - 0.1951736 = 0.02108365$ ms. Gambar 6.6 menunjukkan grafik linieritas kecepatan waktu akses *query*.





Gambar 6.6 Grafik linieritas kecepatan waktu akses query
 Sumber : Pengujian

6.1.4.2 Pengujian Performansi Web Sever

Pengujian performansi *web server* dilakukan untuk mengetahui kinerja dari kemampuan *web server* untuk menangani *request* dari *client*. Tabel 6.22 menerangkan *hardware* yang digunakan untuk diuji kemampuannya sebagai *web server* sekaligus sebagai *computer client*.

Tabel 6.22 : Spesifikasi perangkat keras untuk *server* Aplikasi Katalog Perpustakaan berbasis J2ME

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core(TM) 2 Duo 1.5 GHz
Memori (RAM)	2.99 GHz
Hardisk	Seagate ST9120822AS, kapasitas 120 GB, 5400 rpm
Mother Board	WISTRON 30CD
VGA Card	Intel 965 Graphics Media Accelerator

Sumber : Pengujian

Untuk mengetahui kinerja dari kemampuan *web server* digunakan dua alat bantu yaitu *httperf* dan *autobench*, sekilas penjelasan mengenai masing alat bantu tersebut :

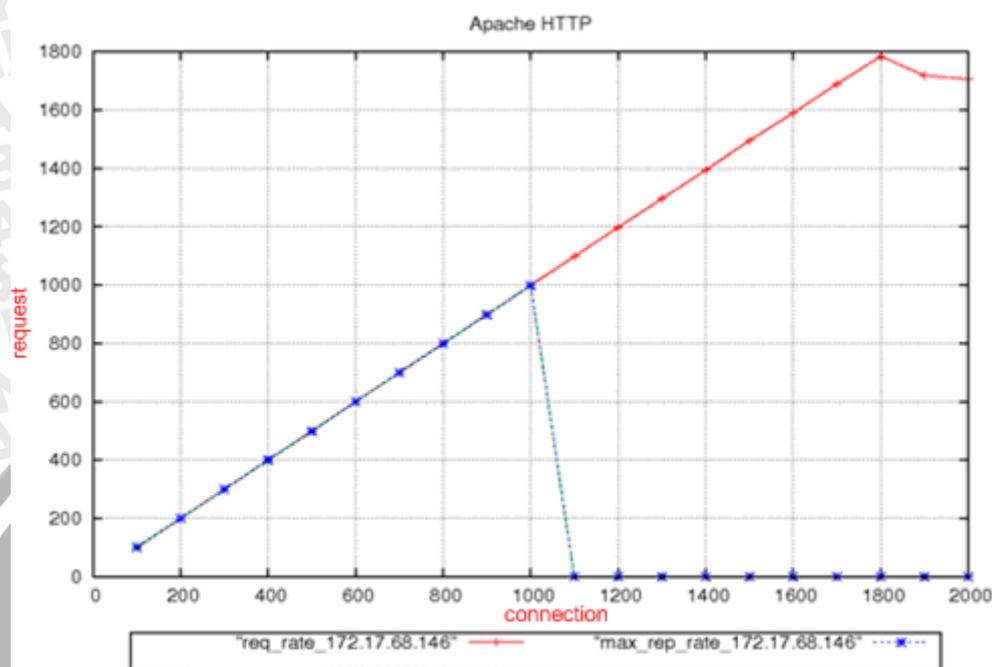
1. Httperf

Httperf adalah alat bantu untuk menguji performansi dari *web server*. Menyediakan berbagai fasilitas untuk menghasilkan berbagai macam beban kerja pada pengukuran terhadap kinerja *web server*.

2. Autobench

Adalah alat bantu untuk mengotomasi proses *benchmark* dari *web server*, dan digunakan bersama *httperf* untuk menghasilkan grafik hasil pengukuran.

Gambar 6.7 menunjukkan pengujian performansi *web server*, dengan 100 sampai 2000 koneksi yang *me-request* secara bersamaan.



Gambar 6.7 : Pengujian Performansi *Web Server* Apache HTTP

Sumber : Pengujian

Httpperf dan Autobench tidak dijalankan pada komputer yang menjadi *server* melainkan dijalankan pada sebuah komputer *client* supaya tidak mempengaruhi hasil pengujian, dan diterapkan pada jaringan lokal.

Parameter yang diberikan untuk pengujian ini adalah dengan memberi kasus uji dimana httpperf membuka antara 100 koneksi sampai dengan 2000 koneksi dengan ketentuan apabila *server* tidak merespon dalam waktu 10 detik maka akan dihitung sebagai *error*.

Pada Gambar 6.7 menunjukkan *request client* mengalami kenaikan, yakni pada titik (1800,1780). Dari data tersebut didapatkan hasil bahwa *web server* Apache HTTP memiliki performansi terbaik pada saat terdapat 1000 *request* dengan 1000 koneksi yang dilakukan oleh *client* secara bersamaan.

BAB VII

PENUTUP

7.1 Kesimpulan

Bedasarkan hasil pengujian dan analisis yang dilakukan terhadap kinerja sistem, diambil kesimpulan sebagai berikut:

1. Kebutuhan-kebutuhan (fungsional dan non-fungsional) untuk merancang sistem sistem katalog perpustakaan dengan menggunakan teknologi *Java2 Micro Edition* (J2ME) telah dirumuskan di dalam daftar kebutuhan. Daftar kebutuhan ini yang menjadi acuan untuk perancangan yang lebih detail.
2. Sistem ini dapat melakukan koneksi ke *web server* Apache untuk mengirim dan mengambil data yang ada di *database sever* MySQL
3. Sistem ini dapat melakukan proses pencarian dan menampilkan daftar koleksi yang ada di Ruang Baca Teknik Elektro.
4. Sistem ini dapat menampilkan informasi data koleksi secara lebih detail. Data yang ditampilkan adalah judul asli, judul seri, ISBN, edisi, cetakan, jilid, pengarang, penerbit, tahun terbit, bahasa, deskripsi fisik, jumlah koleksi, subject, kode rak buku, dewey, bibliografi, jenis koleksi, paket, dan tipe media
5. Pengujian waktu akses *query* pada basis data *db_rbt_e* dengan 200, 400, 800 dan 1600 data *entry* memberikan hasil bahwa waktu akses membutuhkan waktu rata-rata terlama 0.2389 ms dengan 1600 data *entry*. Linieritas kecepatan waktu akses *query* antar data *entry* mendekati 0.02 ms.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak sistem katalog perpustakaan dengan menggunakan teknologi *Java2 Micro Edition* (J2ME) antara lain:

1. Desain atau tampilan sistem dapat dikembangkan menjadi lebih menarik dengan menggunakan komponen Canvas pada J2ME
2. Perangkat lunak dapat dikembangkan dengan menambahkan fasilitas untuk mendownload file koleksi yang ada di Ruang Baca Teknik Elektro

3. Koneksi antara *client* dengan *server* dapat menggunakan komponen `HttpsConnection` yang disediakan oleh `J2ME MIDP 2.1`



DAFTAR PUSTAKA

- [SHA-06] Shalahudin, Muhammad dan Rosa A.S. 2006. *Pemrograman J2ME: Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung: Penerbit Informatika.
- [PRA-04] Prasetyo, Didik Dwi. 2004. *Belajar Sendiri Membangun Aplikasi Web pada Sistem Database Terdistribusi*. Jakarta: Penerbit PT Elex Media Komputindo Kelompok Gramedia.
- [KAD-03] Kadir, Abdul. 2003. *Dasar Pemrograman WEB dinamis Menggunakan PHP*. Yogyakarta: Penerbit ANDI.
- [SOL-06] Solichin, Achmad. 2006. *Pengenalan DBMS dan MySQL*. Akses dari:
http://webdosen.bl.ac.id/dosen/050023/materi/pw2_pertemuan11.pdf
. Tanggal Akses: 3 September 2008.
- [SUW-08] Suwarno, Willy Bayuardi. 2008. *Pemrograman PHP dan MySQL*. Akses dari: <http://willy.situshijau.co.id/downloads/pemrograman-php-mysql>. Tanggal Akses: 3 September 2008.
- [MAR-06] Mardiono, Tri. 2006. *Membangun Solusi Mobile Business dengan Java*. Jakarta: Penerbit PT Elex Media Komputindo.
- [AID-07] Aidi, Adek. 2007. *J2ME: Bright Generation Of Wireless Tech*. Akses dari: <http://lailiaidi.blogspot.com/2007/06/j2me-bright-generation-of-wireless-tech.html>. Tanggal akses: 22 September 2008.
- [SUH-08] Suharyanti. 2008. *Pengantar Dasar Ilmu Perpustakaan*. Lembaga Pengembangan Pendidikan (LPP) dan UPT Penerbitan dan Pencetakan UNS (UNS Press) Universitas Sebelas Maret Surakarta.
- [SUHE-07] Suhendar, Yahya. 2007. *Pedoman Katalogisasi: Cara Mudah Membuat Katalog Perpustakaan*. Jakarta: Kencana.

