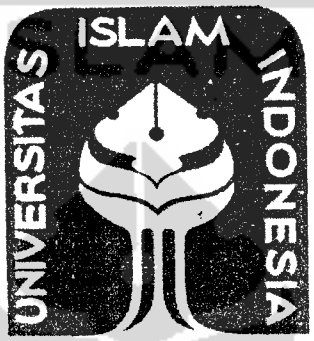
MOBILE OPAC DENGAN JAVA SEBAGAI IMPLEMENTASI

DARI SISTEM TERDISTRIBUSI

BERBASIS XML-RPC

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Program Studi Teknik Informatika



Oleh :

Nama : **Arlinto Wibisono.R**

No. Mahasiswa : **02 523 018**

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

MOBILE OPAC

DENGAN JAVA SEBAGAI IMPLEMENTASI

DARI SISTEM TERDISTRIBUSI BERBASIS XML-RPC

TUGAS AKHIR

Disusun Oleh:

Nama : **ARLINTO WIBISONO.R**
No.Mahasiswa : **02 523 018**

Yogyakarta, 7 Juni 2007

Telah Diterima dan Disetujui dengan baik oleh:

Dosen Pembimbing,

(Taufiq Hidayat, ST.,MCS.)

LEMBAR PENGESAHAN PENGUJI

MOBILE OPAC
DENGAN JAVA SEBAGAI IMPLEMENTASI
DARI SISTEM TERDISTRIBUSI BERBASIS XML-RPC
TUGAS AKHIR

oleh :

Nama : Arlinto Wibisono.R

No.Mahasiswa : 02523018

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 29 Juni 2007

Tim Penguji

Taufiq Hidayat ST., MCS.
Ketua

Sri Kusumadewi, S.Si. MT
Anggota I

Syarif Hidayat, S.Kom.
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi. SSi., M.Kom

LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR

Saya yang bertanda tangan di bawah ini,

Nama : **ARLINTO WIBISONO.R**
No. Mahasiswa : **02 523 018**

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri.
Apabila dikemudian dari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.
Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 7 Juni 2007

(Arlinto Wibisono.R)

PERSEMBAHAN...

[Dari Arlint] • Sebuah Karya Awal [illegible]

- **Allah SWT atas Ridho, Maaf dan Kasih Sayang yang tulus kepada hamba-hamba-Nya, bersama kekasih-Nya, Nabi Muhammad SAW, panutan kebahagian dunia akhirat, Amin...und,**

- Orang Tua Arlint, Bapanya Drs.H.Dulrachman und Bunya Hj.Sri Sularti, Terimakasih semua ya, perhatian tulus ikhlas ke Halin, sungguh kebanggaan di dunia bisa menjadi putra Bapanya and Bunya, und lebih [illegible] di akhirat amin ya Rob, semoga [illegible] [illegible].

- Fräuleine Hj.Ratih Suliantari, SS. Mbaknya....terimakasih semua perhatian, saran, dukungan dan doanya, I.Allah Halin dampingi mbak sekuat tenaga.Halin, Amin...,und,

-

- Kenangan selama setahun terakhir terhitung TA-ku disetujui, terimakasih, dewasa dan logis Arlint digembleng di masa ini,

-

- Masjid Ulil Albab UII rumah Allah SWT,

-

- Dusun, warga tempat KKN di Tanjung sari beserta anggota mandirinya ☺ Icha SH, Rofik ST, Hendy,

HALAMAN MOTTO

"Sebaik-baiknya manusia adalah yang paling bermanfaat bagi manusia"

(HR. Tarmidzi)

"..dan penuhilah janji; sesungguhnya janji itu pasti diminta pertanggungjawabanya".

(Al-Isra' [17]: 34)

"Sesungguhnya disamping kesukaran itu ada kemudahan"

(Al-Insyirah :6)

"Sesungguhnya Allah suka jika salah satu dari kalian bekerja dan menyempurnakan pekerjaanya." (HR Baihaqi)

" Dan siapa yang berpaling dari peringatan-Ku, maka sesungguhnya baginya penghidupan yang sempit...."

(Thaha [20]: 124)

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum, Wr.Wb.

Alhamdulillahirabbil'alamin, dengan memanjatkan puji syukur kehadirat Allah SWT atas limpahan hidayah, kekuatan, petunjuk, taufik serta inayah-Nya sehingga Tugas Akhir yang berjudul "**Mobile OPAC dengan Java Sebagai Implementasi dari Sistem Terdistribusi Berbasis Xml-Rpc**" ini dapat diselesaikan dengan baik.

Tugas Akhir ini ditulis sebagai syarat untuk mendapatkan gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia setelah menempuh proses akademis selama kurang lebih delapan semester.

Penyusun berharap Tugas Akhir ini dapat memberikan manfaat dan memberikan konstribusi yang baik khususnya bagi Jurusan Teknik Informatika Universitas Islam Indonesia.

Dalam menyusun tugas akhir ini, penyusun telah banyak mendapat bantuan, bimbingan dari berbagai pihak, maka dari itu dalam kesempatan ini perkenankanlah penyusun menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang telah memberikan perlindungan, rahmat, hidayah dan semua kenikmatan-Nya hingga penyusun dapat menyelesaikan Tugas Akhir ini dengan sebaik-baiknya.
2. Kedua orang tua ayahanda Drs.H.Dulrachman dan ibunda Hj.Sri Sularti serta kakanda Hj.Ratih Suliantari Rachman, SS. tercinta yang tak henti-hentinya memberikan doa dan semua dukungan.
3. Bapak Prof. Dr. Edy Suandi Hamid, selaku rektor Universitas Islam Indonesia.
4. Bapak Fathul Wahid, ST, M.Sc, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.

SARI

Perpustakaan merupakan sumber informasi bagi seseorang dimana dalam perpustakaan terdapat buku, jurnal, makalah atau sumber-sumber lainya. Termasuk dalam bagian sistem informasi perpustakaan (SIMPUS) terdapat sistem OPAC (Online Public Access Catalog), yaitu sistem yang menyediakan informasi katalog dari perpustakaan. Kegunaan dari sistem ini adalah membantu pengunjung perpustakaan untuk melakukan penelusuran sumber daya yang ada dalam perpustakaan .

Aplikasi mobile OPAC adalah aplikasi OPAC pada perangkat bergerak dengan menggunakan konsep sistem terdistribusi, dengan magsud, memudahkan bagi user untuk melakukan penelusuran sederhana maupun advance (ahli). Dimudahkan dalam arti sistem dapat melakukan penelusuran kepada lain perpustakaan dengan atribut penelusuran yang sama, seperti atribut pada penelusuran perpustakaan yang kali pertama dituju.

Hasil yang didapatkan dari penelusuran katalog oleh user adalah daftar koleksi perpustakaan yang dapat dijadikan referensi sebelum user melakukan peminjaman kepada perpustakaan yang mempunyai koleksi yang diinginkan oleh user.

Kata kunci : SIMPUS, OPAC, catalog, perpustakaan, web, sistem terdistribusi, penelusuran sederhana, penelusuran advance (ahli), user.

TAKARIR

Katalog	Koleksi (sumberdaya) perpustakaan.
Mobile device	Perangkat Bergerak.
Client	Peminta layanan.
OPAC	Penelusuran katalog perpustakaan.
Server	Penyedia layanan.
Application Server	Aplikasi yang berada pada server.
Database Server	Database yang berada pada server.
MIDlet	Aplikasi java mobile dengan konfigurasi menggunakan CLDC dan profile MIDP.
XML-RPC	Teknologi pemanggilan metode (procedure) pada lain tempat menggunakan protocol RPC dan data yang dilewatkan (ditukarkan) adalah XML.
Web Service	Layanan yang disediakan oleh web
SistemTerdistribusi	Sistem yang terintegrasi antar beeberapa sistem biasanya pada lokasi yang berberda-beda.
container web/ web server	Tempat ditempatkanya suatu aplikasi web.
Deploy	Proses menempatkan sebuah komponen pada sebuah lingkungan operasional.
MIDP	Profile pada spesifikasi java ME.
DDC	Dewey Decimal Classification, yaitu klasifikasi yang dibuat oleh Melvil Dewey sebagai klasifikasi pengetahuan

	untuk keperluan menyusun buku di perpustakaan.
Request-Response	Meminta-Menanggapi.
Library	Kumpulan program atau fungsi.
Stream	*A*liran byte data yang dikirim dari pengirim ke penerima.
POST	Metode pengiriman data ke server dengan tidak mengirimkan beberapa parameter.
GET	Metode pengiriman data ke server dengan mengirimkan beberapa parameter.
XML	Format data.
RPC	Protokol yang dapat melakukan eksekusi prosedur atau metode pada tempat yang berbeda.
IDL	Interface objek pada CORBA.
SOAP	*S*alah satu protokol web service yang telah terstandarisasi oleh W3C (konsorsium web).
CLDC	Konfigurasi pada java ME.
Platform	Perangkat.
Enterprise Beans	Salah satu modul dalam java EE.
Servlet	Program java pada server.
Context	Sekumpulan binding.
Enterprise Archive	Kompresisasi file pada aplikasi enterprise.
Web Archive	Kompresisasi file pada modul web.

BAB V SIMPULAN dan SARAN 124
5.1 Simpulan 124
5.2 Saran 125
DAFTAR PUSTAKA 126
LAMPIRAN 128

DAFTAR TABEL

Tabel 2.1	Kode Respon *HTTP*	11
Tabel 2.2	Metode *HTTP Request MIDP 1.0*	11
Tabel 2.3	Metode-Metode Package *javax.microedition.io.HttpConnection*	12
Tabel 2.3a	Lanjutan Metod Package *javax.microedition.io.HttpConnection*	12
Tabel 2.4	Daftar *Package Apache XML-RPC 3.0*	15
Tabel 2.5	Metode-Metode pada *Package Servlet*	24
Tabel 2.5a	Lanjutan Metode-Metode pada *Package Servlet*	24
Tabel 3.1	Struktur Tabel *jenisuii*	77
Tabel 3.2	Struktur Tabel *koleksiuii*	77
Tabel 3.3	Struktur Tabel *lokasiuii*	78
Tabel 3.4	Struktur Tabel *LOKASI*	79
Tabel 3.5	Struktur Tabel *KOLEKSI*	79
Tabel 3.6	Struktur Tabel *JENIS*	80
Tabel 4.1a	Intisari Proses Pengujian Penelusuran	122
Tabel 4.1b	Lanjutan Intisari Proses Pengujian Penelusuran	123

DAFTAR GAMBAR

Gambar 2.1	Referensi OSI Tujuh Lapis.....	9
Gambar 2.2	*Marshalling* data *XML-RPC*.....	13
Gambar 2.3	Arsitektur *Web* dengan *XML-RPC*.....	14
Gambar 2.4	Ruang Lingkup *Java Development Kit (JDK) 1.6*.....	17
Gambar 2.5	Arsitektur *Java Micro Edition*.....	18
Gambar 2.6	Daur Hidup *MIDlet*.....	19
Gambar 2.7	Arsitektur Aplikasi *Multi- Tier*.....	21
Gambar 3.1	*Use Case Diagram Mobile OPAC*.....	39
Gambar 3.2	*Use Case Diagram Server Mobile OPAC*.....	40
Gambar 3.3	Diagram *Class Midlet (WebToMobileClientMIDlet)*.....	41
Gambar 3.4	Diagram *Class Stub (WebToMobileClient)*.....	41
Gambar 3.5	Diagram *class* Proses Bisnis *Server Gateway*.....	42
Gambar 3.6	Diagram *Class Stub Server (WebToMobileServlet)*.....	42
Gambar 3.7	Diagram *Class* Proses Bisnis Perpsus UII.....	43
Gambar 3.8	Diagram *Class AksesDatabaseUii*.....	43
Gambar 3.9	Diagram *Class LogikaBisnisServerPerpusUgm* UGM.....	44
Gambar 3.10	Diagram *Class KoneksiDatabasePerpusUgm*.....	44
Gambar 3.11	Diagram *Class* keterikatan *Class-Class* Penelusuran UII...	46
Gambar 3.12	Diagram *Class* keterikatan *Class* Penelusuran UGM.....	47
Gambar 3.13	Diagram *Class* Keterikatan *Class* Penelusuran Lain *Server*.....	48
Gambar 3.14	Diagram *Class* Keterikatan *Class* Ubah *URL Gateway*.....	49
Gambar 3.15	Diagram *Class* Generalisasi Logika Bisnis Server.....	49
Gambar 3.16	*Sequence Diagram* Penelusuran Sederhana Judul UII.....	50
Gambar 3.17	*Sequence Diagram* Penelusuran Sederhana Pengarang UII.	51
Gambar 3.18	*Sequence Diagram* Penelusuran Sederhana Judul UGM.....	52

Gambar 3.19	*Sequence Diagram* Penelusuran Sederhana Pengarang UGM	53
Gambar 3.20	*Sequence Diagram* Penelusuran *Advance* UII	54
Gambar 3.21	*Sequence Diagram* Penelusuran *Advance* UGM	55
Gambar 3.22	*Sequence Diagram* Penelusuran Judul UII Lain *Server*	56
Gambar 3.23	*Sequence Diagram* Penelusuran Pengarang UII Lain *Server*	57
Gambar 3.24	*Sequence Diagram* Penelusuran Judul UGM Lain *Server*	58
Gambar 3.25	*Sequence Diagram* Penelusuran Pengarang UGM Lain *Server*	59
Gambar 3.26	*Sequence Diagram* Penelusuran *Advance* UII ke Lain *Server*	60
Gambar 3.27	*Sequence Diagram* Penelusuran *Advance* UGM Lain *Server*	61
Gambar 3.28	*Sequence Diagram* Pengubahan *URL Server Gateway*	62
Gambar 3.29	*Sequence Diagram* Lihat Bantuan Program	63
Gambar 3.30	Diagram Jaringan Sistem *Mobile OPAC*	64
Gambar 3.31	Rancangan Antarmuka *Splash Screen*	68
Gambar 3.32	Rancangan Antarmuka Menu Utama	69
Gambar 3.33	Rancangan Antarmuka Menu *OPAC*	69
Gambar 3.34	Rancangan Antarmuka Penelusuran Sederhana	70
Gambar 3.35	Racangan Antarmuka Hasil Penelusuran Sederhana	71
Gambar 3.36	Rancangan Antarmuka Penelsuran *Advance* (Ahli)	71
Gambar 3.37	Racangan Antarmuka Hasil Penelusuran *Advance*	72
Gambar 3.38	Rancangan Antarmuka Ubah *URL Server Gateway*	72
Gambar 3.39	Rancangan Antarmuka Perkenalan Program	73
Gambar 3.40	Rancangan Antarmuka Menu Bantuan	73
Gambar 3.41	Rancangan Antarmuka Bantuan Penelusuran Sederhana	74
Gambar 3.42	Rancangan Antarmuka Penelusuran *Advance*	74
Gambar 3.43	Rancangan Antarmuka Bantuan *Setting*	75
Gambar 3.44	Rancangan Antarmuka Tentang Program	75

Gambar 3.45	Rancangan Antarmuka Peringatan Kesalahan.....	76
Gambar 3.46	Rancangan Antarmuka Pemberitahuan *Server* Telah Aktiv.	77
Gambar 3.47	Rancangan Antarmuka Pemberitahuan Stub Server Aktiv..	77
Gambar 3.48	Relasi Tabel *Database* Perpus UII.....	79
Gambar 3.49	Relasi Tabel Database Perpus UGM.....	81
Gambar 4.1	Tampilan *Splash Screen*.....	84
Gambar 4.2	Tampilan Menu Utama.....	85
Gambar 4.3	Menu Penelusuran *OPAC*.....	86
Gambar 4.4	Tampilan Penelusuran Sederhana.....	87
Gambar 4.5	Tampilan Hasil Penelusuran Sederhana.....	87
Gambar 4.6	Tampilan Penelusuran Advance (Ahli).....	88
Gambar 4.7	Hasil Tampilan Penelusuran *Advance* (Ahli).....	89
Gambar 4.8	Tampilan Halaman *Setting URL Gateway*.....	89
Gambar 4.9	Tampilan Halaman Perkenalan Program.....	90
Gambar 4.10	Tampilan Menu Bantuan Program.....	91
Gambar 4.11	Tampilan Halaman Bantuan Penelusuran Sederhana.....	91
Gambar 4.12	Tampilan Bantuan Penelusuran *Advance* (Ahli).....	92
Gambar 4.13	Tampilan Halaman Bantuan *Setting*.....	92
Gambar 4.14	Tampilan Halaman Tentang Program.....	93
Gambar 4.15	Tampilan Antarmuka Status *Server* Telah Aktiv.....	94
Gambar 4.16	Tampilan Status *Stub Server* Telah Aktiv.....	95
Gambar 4.17	Tampilan Pengujian Penelusuran Sederhana UII.....	107
Gambar 4.18	Tampilan Hasil Pengujian Penelusuran Sederhana Judul UII.....	108
Gambar 4.19	Tampilan Pengujian Penelusuran *Advance* UGM.....	109
Gambar 4.20	Tampilan Hasil Pengujian Penelusuran *Advance* UGM...	110
Gambar 4.21	Tampilan Pengujian Penelusuran Sederhana perpus UGM.	111
Gambar 4.22	Tampilan Hasil Pengujian Penelusuran UGM Berdasar Judul....	112
Gambar 4.23	Tampilan Pengujian Penelusuran *Advance* UII.....	113
Gambar 4.24	Tampilan Hasil Pengujian Penelusuran *Advance* UII.....	113
Gambar 4.25	Tampilan Pengujian Penelusuran Sederhana Judul UII.....	115
Gambar 4.26	Tampilan Hasil Pengujian Penelusuran Sederhana UII.....	115

Gambar 4.27 *Tampilan Pengujian Setting URL gateway*..... 117
Gambar 4.28 Tampilan Pengujian Penelusuran Sederhana UGM..... 118
Gambar 4.29 Tampilan Pesan Kesalahan pada Peguujian URL..... 118
Gambar 4.30 Tampilan Halaman Penelusuran *Advance* UII..... 119
Gambar 4.31 Tampilan Halaman Pesan Kesalahan..... 120
Gambar 4.32 Tampilan Pengujian Penelusuran *Advance* UGM..... 121
Gambar 4.33 Tampilan Kesalahan Pengujian Jika *Server Gateway Down*..... 121

DAFTAR LAMPIRAN

Lampiran A 128

Konfigurasi yang dibutuhkan oleh *server*..... 129

Konfigurasi pada *Server* Utama atau *Gateway*..... 129

Konfigurasi pada Server Perpustakaan UII..... 129

Konfigurasi pada Perpustakaan UGM..... 130

Lampiran B 131

Data koleksi perpustakaan..... 132

Data Koleksi Perpustakaan UII..... 132

Data Koleksi Perpustakaan UGM..... 133

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi (TI) sangat pesat saat ini, khususnya dalam bidang *web*, dimana pada teknologi *web*, sumua informasi dapat diperoleh walaupun berada pada lokasi yang berainan tempat, karena mempunyai konsep sistem terdistribusi. Fakta ini banyak diperoleh dari banyaknya masalah yang dapat diselesaikan, sebagai contoh pada bidang perpustakaan.

Perpustakaan merupakan sumber informasi bagi seseorang, dimana dalam perpustakaan terdapat berbagai sumber daya perpustakaan (koleksi perpustakaan) diantaranya buku, jurnal, makalah atau lainya. Perpustakaan yang baik adalah perpustakaan yang disediakannya berbagai macam kebutuhan dari penggunanya. Berdasarkan jenisnya, terdapat jenis perpustakaan konvensional yaitu yang saat ini banyak tersedia. Perpustakaan konvensional yang ada saat ini telah memberikan manfaat penting bagi para penggunanya, yaitu disediakannya berbagai macam kebutuhan para pengguna untuk mencari atau menelusuri berbagai sumber daya yang ada di perpustakaan tersebut.

Masalah yang sering ada pada perpustakaan konvensional yaitu tidak disediakannya pencarian katalog sumber daya perpustakaan yang ada dengan mudah, cepat, dan efesien bagi para pengguna. Pengguna diharuskan untuk mencari secara manual (satu persatu) menurut indeksnya masing-masing dan mengakibatkan penelusur katalog diharuskan mempunyai usaha yang lebih atau penelusur direpotkan untuk mencari kebutuhan mereka pada perpustakaan secara manual, ditambah semakin besar koleksi perpustakaan tersebut, tentunya pencari juga diharuskan mempunyai usaha yang lebih untuk menemukan sumber daya perpustakaan (koleksi) yang dibutuhkan. Masalah pencarian katalog secara konvensional atau tradisional juga mempunyai keterbatasan lagi, yaitu jika penelusur katalog menginginkan untuk

d. Menggunakan konsep atau spesifikasi *XML-RPC* untuk melakukan pemanggilan *remote procedure* atau metode pada lain *server*.

e. Tidak membahas masalah keamanan atau enkripsi data.

f. Konsep penelitian adalah simulasi sistem *OPAC* pada *MIDlet* dengan *server* yang terhubung, dan disimulasikan dengan menggunakan simulator *MIDlet* yang disediakan oleh *tool java*.

1.4 Tujuan Penelitian

Adapun tujuan penelitian adalah terciptanya sebuah sistem berbasis *mobile* sebagai aplikasi pada sisi *client* yang mempunyai kemampuan :

a. Mengakses aplikasi pada sisi *server* dan *database server*.

b. Melakukan penelusuran pada sistem *OPAC* dengan konsep terdistribusi antar server yang telah didaftarkan.

c. Menerapkan konsep sistem terdistribusi dengan *XML-RPC* yaitu dapat dilakukanya pemanggilan metode-metode antar *server* yang telah terdaftar.

d. Menanampilkan hasil penelusuran *OPAC* .

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah :

a. Bagi pengembang, mempermudah pengembang aplikasi terdistribusi *OPAC* dengan teknologi *XML-RPC* yang notabene lebih mudah, sederhana dan ringan dibandingkan dengan metode sistem teridistribusi (*web service*) lainya.

b. Bagi pengguna sistem, dipermudah dalam melakukan penelusuran *katalog* karena tidak dilakukan secara manual dan berkat sistem terdistribusi, pencarian meliputi seluruh jaringan perpustakaan yang sudah terdaftar atau tergabung.

1.6 Metodologi Penelitian

Metode yang digunakan untuk penyelesaian tugas akhir adalah

a. Menganalisis suatu sistem *OPAC* yang ada, apa saja yang dimuat dalam sistem tersebut atau yang dibutuhkan oleh sistem, dan menyiapkan peralatan-peralatan yang dibutuhkan.
b. Melakukan wawancara kepada pihak perpustakaan untuk mengetahui sistem informasi perpustakaan atau katalogisasi pada perpustakaan.
c. Melakukan studi pustaka tentang *XML-RPC* dan pemrograman berorentasi objek dengan *java* dan yang studi pustaka lainya yang menunjang dari analisis sistem yang ada.
d. Merancang desain sistem yang dibangun menggunakan model perancangan *UML*.
e. Implementasi sistem.
f. Pengujian dan analisis kinerja.

   Pengujian dilakukan dengan melakukan *input* data penelusuran untuk memastikan sistem telah bekerja dan analisis kerja sistem dengan melakukan keadaan-keadaan diluar dari kamampuan atau kinerja sistem
g. Penulisan laporan dan kesimpulan penelitian.

1.7 Sistematika Penelitian

Sebagai uraian keseluruhan tentang masalah yang akan dibahas dalam laporan ini, maka sistematika laporan dibagi dalam lima bab, sebagai berikut :

Bab I: Pendahuluan

Berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, hipotesis, metodologi penelitian, dan sistematika penelitian.

Bab II: Landasan Teori

Penjelasan mengenai sistem terdistribusi, *XML-RPC*, basis data, teknologi *java* dan materi pendukung lainya.

Bab III: Metodologi

Memuat uraian tentang langkah-langkah dalam penelitian sesuai dengan judulnya. Yaitu analisis perangkat lunak, perancangan kebutuhan perangkat lunak yang akan dikembangkan dan implementasi dari perangkat lunak yang sudah dirancang dan dianalisis sebelumnya.

Bab IV: Hasil dan Pembahasan

Berisi tentang uaraian bagaimana hasil dari aktivitas atau metodologi penelitan yang sudah disampaikan pada bab sebelumnya dan dilakukanya pembahasan mengenai hasil yang sudah dicapai, diperoleh selama melakukan penelitian tugas akhir seperti mengapa hasil diperoleh, kelebihan, kekurangan dari hasil yang diperoleh.

BabV: Simpulan dan Saran

Berisi tentang uraian simpulan dan saran hasil analisis kerja yang telah dibuat pada bab sebelumnya yaitu hasil dan pembahasan.

BAB II

LANDASAN TEORI

2.1 Perpustakaan

Perpustakaan merupakan salah satu sarana tempat pembelajaran yang dapat menjadi sebuah kekuatan untuk mencerdaskan bangsa. Perpustakaan sangat membantu dalam tersedianya informasi atau ilmu pengetahuan kepada banyak orang. Perpustakaan menurut definisinya dibagi menjadi dua, yaitu perpustakaan tradisional dan perpustakaan *modern*. Perpustakaan dalam arti tradisional adalah sekumpulan koleksi buku, majalah dan, lain sebagainya.

Seiring dengan perkembangan teknologi muncul istilah perpustakaan modern, sehingga dapat didefinisikan kembali bahwa perpustakaan modern adalah tempat untuk mengakses informasi dalam berbagai format, sebagai contoh *CD (Compact Disc)*, majalah, buku, jurnal, penelitian dan lain sebagainya, apakah informasi itu disimpan dalam gedung perpustakaan itu atau tidak.

2.2 Katalog dan Sistem *OPAC (Online Public Access Catalog)*

Pada era globalisasi saat ini, dibutuhkan suatu perpustakaan yang mempunyai image atau citra yang bagus dan untuk mewujudkannya, maka diperlukan tiga pilar citra utama yaitu :

1. Membangun citra perpustakaan.
2. Meningkatkan citra pustakawan.
3. Mengembangkan perpustakaan yang berbasis teknologi informasi dan komunikasi atau *Information and Communication Technology (ICT Based)*.

Pilar pertama dan kedua merupakan pilar yang berada dalam internal perpustakaan (perpustakaan itu sendiri) yang keduanya harus diperhatikan agar saling berjalan sesuai dengan fungsinya, yaitu pihak perpustakaan dan pustakawan mampu untuk membangun image atau citra yang baik agar pengguna perpustakaan dapat menggunakan perpustakaan sesuai dengan kebutuhan,

semaksimal mungkin dan perpustakaan dapat berjalan sesuai dengan fungsi dan perannya.

ICT diperlukan sebagai sarana dalam meningkatan kualitas layanan dan operasional yang telah membawa perubahan yang besar dalam perpustakaan. Perubahan tersebut dapat diukur dengan diterapkanya sistem informasi perpustakaan dan perpustakaan *digital*.

Seseorang dalam mencari informasi yang dibutuhkan di perpustakaan pasti menginginkan untuk segera mendapatkan informasi yang ingin diperoleh, contoh, pengunjung perpustakaan menginginkan untuk mencari jurnal tentang informatika, sudah pasti pengunjung tersebut menginginkan untuk didapatkanya jurnal tersebut secara mudah dan cepat. Untuk itu, sebuah perpustakaan biasanya sudah menyediakan sebuah tata organisasi perpustakaan atau jika perpustakaan tersebut menggunakan komputer sebagai alat bantu perpustakaan, sering disebut dengan Sistem Informasi Perpustakaan (SIMPUS). Banyak perpustakaan yang sudah menerapkan manajemen informasi mereka dengan SIMPUS, sebagai contoh beberapa pepustakaan universitas berikut telah menerapkannya yaitu universitas Brawijaya (Brawijaya *University Library*), universitas Airlangga (Airlangga *University Library*), dan lainya.

Sistem Informasi Perpustakaan menyediakan beragam informasi yang sangat kompleks, dimulai dengan kegiatan sirkulasi, pengadaan inventarisasi, administrasi, katalogisasi, pengolahan, statistic, pengolahan data karyawan dan lain sebagainya.

Dalam meningkatkan image perpustakaan sesuai dengan pilar yang ketiga, yaitu mengembangkan perpustakaan berbasis teknologi informasi maka, perpustakaan menyediakan suatu sistem yang berkaitan dengan katalogisasi yaitu sistem yang berguna dalam penelusuran katalog oleh pengguna perpustakaan, sistem tersebut dinamakan OPAC (Online Public Access Catalog), sehingga pengguna dimudahkan dalam melakukan penelusuran katalog.

OPAC meyediakan pilihan atribut penelusuran katalog baik itu penelusuran berdasar atribut judul, pengarang, dan lainya tergatung pilihan yang disediakan oleh sistem, sehingga pengguna sistem mendapatkan koleksi perpustakaan yang

ditelusurinya dan sekaligus memberikan kepuasan bagi pengguna sistem yang tentunya akan meningkatkan image atau citra perpustakaan.

2.3 Sistem Terdistribusi

Sistem terdistribusi dapat diartikan sebagai sistem yang tersusun dari subsistem-sub-sistem yang relatif independen dan otonom yang biasanya tersebar secara fisik. [DIR05]

Dalam sistem terdistribusi terdapat aspek yang harus dipenuhi yaitu komunikasi, koordinasi, sinkronisasi, dan konkuerensi. Dari keempat aspek tersebut, cara pengimilementasiaanya adalah dengan sistem terdistribusi terbuka yaitu saling beroprasi (*interoperability*) dan bebas implementasi (tiap pabrikan komponen bebas mengimplementasikan produknya dalam caranya sendiri-sendiri).

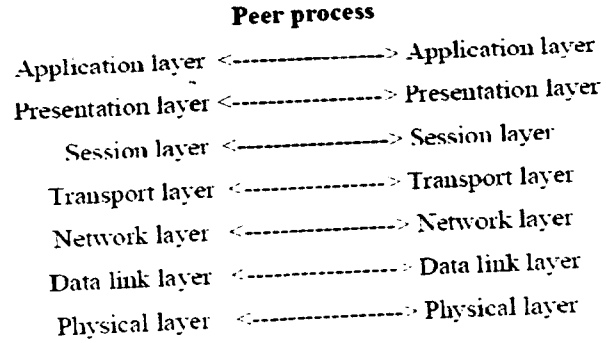
2.4 *TCP/IP*

TCP/IP adalah salah satu jenis *protokol* yg memungkinkan kumpulan komputer untuk berkomunikasi dan bertukar data didalam suatu *network* (jaringan).[PUR00]

Dikarenakan *TCP/IP* adalah serangkaian *protokol* di mana setiap *protokol* melakukan sebagian dari keseluruhan tugas komunikasi jaringan, maka tentulah implementasinya tak lepas dari arsitektur jaringan itu sendiri.

Arsitektur rangkaian protokol *TCP/IP* mendifinisikan berbagai cara agar *TCP/IP* dapat saling menyesuaikan.

Karena *TCP/IP* merupakan salah satu lapisan protokol *OSI (Open System Interconnections)*, berarti bahwa hierarki TCP/IP merujuk kepada tujuh lapisan OSI tersebut. Gambar 2.1 adalah model referensi *OSI* tujuh lapisan, yang mana setiap lapisan menyediakan tipe khusus pelayanan jaringan.



Gambar 2.1. Referensi *OSI* Tujuh Lapis

Tiga lapisan teratas biasa dikenal sebagai "*upper layer protocol*" sedangkan empat lapisan terbawah dikenal sebagai "*lower level protocol*".

Tiap lapisan berdiri sendiri tetapi fungsi dari masing-masing lapisan bergantung dari keberhasilan operasi *layer* atau lapisan sebelumnya. Sebuah lapisan pengirim hanya perlu berhubungan dengan lapisan yang sama di penerima (jadi misalnya lapisan *data link* penerima hanya berhubungan dengan data link pengirim) selain dengan satu layer di atas atau dibawahnya.

Model dengan menggunakan lapisan ini merupakan sebuah konsep yg penting karena suatu fungsi yg rumit yg berkaitan dengan komunikasi dapat dipecahkan menjadi sejumlah unit yg lebih kecil. Tiap lapisan bertugas memberikan layanan tertentu pada lapisan diatasnya dan juga melindungi lapisan diatasnya dari rincian cara pemberian layanan tersebut. Tiap lapisan harus transparan sehingga modifikasi yg dilakukan atasnya tidak akan menyebabkan perubahan pada lapisan yang lain. Lapisan menjalankan perannya dalam pengalihan data dengan mengikuti peraturan yang berlaku untuknya dan hanya berkomunikasi dengan lapisan yang setingkat.

Akibatnya sebuah layer pada satu sistem tertentu hanya akan berhubungan dengan lapisan yang sama dari sistem yang lain. Proses ini dikenal sebagai "*Peer process*".

Dalam keadaan sebenarnya tidak ada data yang langsung dialihkan antar lapisan yang sama dari dua sistem yang berbeda ini. Lapisan atas akan

memberikan data dan kendali ke lapisan dibawahnya sampai lapisan yang terendah dicapai. Antara dua lapisan yang berdekatan terdapat *"interface"* (antarmuka). *Interface* ini mendifinisikan operasi dan layanan yang diberikan olehnya ke lapisan lebih atas. Tiap lapisan harus melaksanakan sekumpulan fungsi khusus yang dipahami dengan sempurna. Himpunan lapisan dan protokol dikenal sebagai "arsitektur jaringan".

2.5 *GPRS* dan Koneksi *HTTP*

Untuk *client* agar dapat menghubungi *server* (aplikasi *web* yang dibuat dan ditempatkan pada suatu *container web* atau sering disebut juga *web server*) diperlukan suatu koneksi yaitu koneksi *HTTP*, karena koneksi *HTTP* adalah koneksi yang banyak digunakan dan menjadi standar koneksi *MIDP*. Koneksi *HTTP* menggunakan *GPRS (General Packet Radio Service)* yaitu layanan komunikasi tanpa kawat berbasis paket. *GPRS* menjanjikan kecepatan mulai dari 56 *kbps* sampai 114 *kbps*, sehingga memungkinkan akses internet biasa sampai multimedia bagi komputer, *notebook* dan *handheld*.

GPRS dibuat berdasarkan komunikasi *GSM (Global System for Mobile communication)*. *GPRS* secara teori lebih murah daripada sambungan telepon seluler, karena kanal yang dipakai dibagi beramai-ramai oleh sejumlah pengguna.

HTTP atau Hypertext Transfer Protocol adalah suatu aturan yang perlu diikuti oleh *web browser* untuk meminta dan mengambil (*request response*) suatu dokumen yang disediakan oleh *web server*. Pengambilan dari *web server* biasanya berupa kode. *response* atau response code beserta dokumen yang diminta oleh klien. [SHL06]

Sebuah *client (MIDlet)* melakukan suatu *request* data pencarian kepada *application web* dengan menggunakan metode *request* dan hasil pencarian dikembalikan oleh *application web* dengan menggunakan metode *reaponse*.

Mekanisme koneksi *HTTP* pada *MIDP* melalui tiga tahap status, yaitu : *SETUP, CONNECTED, CLOSED*. Parameter "*Request*" diatur pada *SETUP*. Pada status *CONNECTED*, diharapkan kembalinya suatu respon dari *HTTP server*.

Setelah proses-proses dilakukan, maka koneksi tersebut harus ditutup pada status *CLOSED* .

Pada ponsel koneksi *HTTP* ini menggunakan *GPRS*. *HTTP response code* pada *Java ME* adalah seperti pada tabel 2.1.

Tabel 2.1. Kode Respon *HTTP*

Nama Koneksi HTTP	Nilai	Keterangan
HTTP_OK	200	*Request* sukses.
HTTP_MOVED_PERM	301	*Request* sumber daya telah dipindah secara permanen yang tertulis pada header.
HTTP_MOVED_TEMP	302	Request sumber data secara permanen ke *URL* yang tertulis pada header.
HTTP_SEE_OTHER	303	*Request* sumber daya telah dipindah sementara secara permanen ke *URL* yang tertulis pada *header*.
HTTP_BAD_REQUEST	400	*Request* gagal karena salah satu format atau tempat.
HTTP_FORBIDDEN	403	*Request* valid, tetapi *server* tidak mengijinkan aksi yang diminta.
HTTP_NOT_FOUND	404	*Request* menuju *URL* yang tidak ada.

Metode *HTTP* request yang didukung oleh *MIDP* 1.0 adalah seperi pada tabel 2.2.

Tabel 2.2. Metode *HTTP Request MIDP 1.0*

Metode	Simbol Konstan	Keterangan
GET	HttpConnection.GET	Mengirimkan atau meminta informasi bagian dari *URL*.
HEAD	HttpConnection.HEAD	Mengirimkan atau meminta informasi meta dari sebuah sumber daya (resource).
POST	HttpConnection.POST	Mengirimkan atau meminta informasi melalui aliran (stream) yang berbeda.

Metode-metode pada javax.microedition.io.HttpConnection, *library* yang menyediakan fungsi-fungsi untuk koneksi *HTTP* adalah seperti pada tabel 2.3.

Tabel 2.3. Metode-Metode Package *javax.microedition.io.HttpConnection*

Metode	Keterangan
void setRequestMethod(String method)	Mengeset metode permintaan (*request*) (*GET,POST atau HEAD*).
void setRequestProperty(String key,String value)	Mengeset properti permintaan(*request*).
String getRequestMethod()	Mengembalikan nilai metode permintaan (*request*).
String getRequestProperty(String key)	Mengembalikan nilai properti dengan nama masukan.
String getResponseCode()	Mengembalikan nilai kode respon.
String getResponseMessage()	Mengembalikan nilai pesan respon.
String getHeaderField(int n)	Mengembalikan nilai *header* yang memiliki nama masukan.
long getHeaderFieldDate(String name,long def)	Mengembalikan field nama sebagai tipe long yang mereprentasikan tanggal (date).

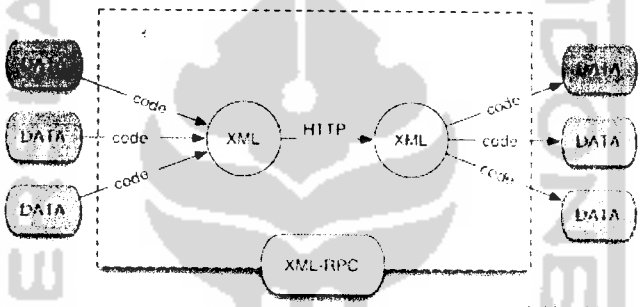
Tabel 2.3a. Lanjutan Metode-Metode *Package* javax.microedition.io.HttpConnection

Metode	Keterangan
int getHeaderFieldInt(String name,int def)	Mengembalikan field nama sebagai tipe integer.
String getHeaderFieldKey(int n)	Mengembalikan nilai nama field header dengan menggunakan indeks.
long getDate()	Mengembalikan nilai *header* tanggal (date).
long getExpiration()	Mengembalikan nilai *header* kadaluarsa (expires).
long getLastModified()	Mengembalikan *header* tanggal terakhir modifikasi (last-modified).
String getFile()	Mengembalikan nama file dari *URL*.
String getHost()	Mengembalikan nama host dari *URL*.
int getPort()	Mengembalikan nilai port dari *URL*.
String getProtocol()	Mengembalikan nilai *protocol* dari *URL*.
String getQuery()	Mengembalikan nilai *query* untuk metode permintaan *GET*.
String getRef()	Mengembalikan nilai porsi keterkaitan dari *URL*.
String getURL()	Mengembalikan nilai *URL*.

Terdapat perbedaan pada penulisan address metode *POST* dan *GET*. Jika pada post metode dikirimkan tidak dengan properti karena dikirim melalui *stream* dan jika *GET metod* dikirimkan dengan menggunakan properti.

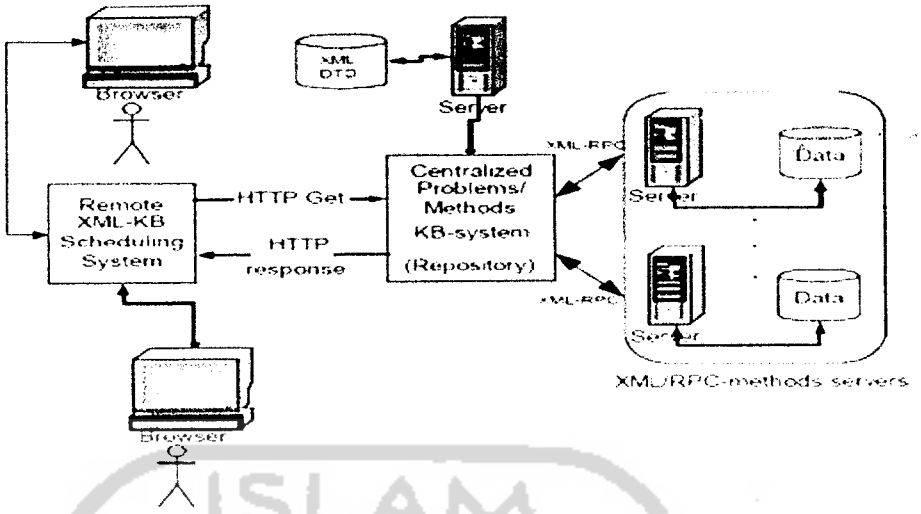
2.6 *XML-RPC*

Adalah suatu spesifikasi dan sekumpulan implementasi yang memungkinkan suatu perangkat lunak mampu berjalan diatas sistem operasi yang berbeda-beda, berjalan pada lingkungan yang berbeda-beda, untuk melakukan pemanggilan procedure melalui internet. Pemanggilan prosedure mneggunakan *HTTP* sebagai transport dan *XML* sebagai *encoding* –datanya *XML-RPC* dibuat untuk memudahkan dan sederhana ketika dilakukanya proses pemanggilan, pemrosesan dan respon.



Gambar 2.2. *Marshalling* data *XML-RPC*

Prinsip dari *XML-RPC* ini adalah, dilakukanya pemanggilan suatu prosedur melalui protokol *HTTP* dengan menggunakan *encoding XML*. Sehingga cara ini sangat cocok untuk sistem komputasi terdistribusi, di mana lingkungan sistem operasinya berbeda-beda.



Gambar 2.3. Arsitektur *Web* dengan *XML-RPC*

Library XML-RPC yang disediakan oleh *Apache* yaitu *XML-Apache* sudah mencapai versi 3.0, berikut ini adalah daftar library yang dimiliki oleh versi 3.0 :

a. xmlrpc-client-3.0.
b. xmlrpc-server-3.0.
c. commons-logging-1.1.
d. xmlrpc-common-3.0.
e. ws-commons-util-1.0.1.
f. xmlrpc-client-3.0-sources.
g. xmlrpc-common-3.0-sources.
h. xmlrpc-server-3.0-sources.

Dan daftar packages dari Apache *XML-RPC* 3.0 adalah seperti dalam table 2.4.

Tabel 2.4. Daftar *Package Apache XML-RPC 3.0*

Packages
org.apache.xmlrpc
org.apache.xmlrpc.client
org.apache.xmlrpc.client.util
org.apache.xmlrpc.common
org.apache.xmlrpc.jaxb
org.apache.xmlrpc.metadata
org.apache.xmlrpc.parser
org.apache.xmlrpc.serializer
org.apache.xmlrpc.parser
org.apache.xmlrpc.util
org.apache.xmlrpc.webserver

2.6.1 *XML*

Extensible Markup Language (XML) adalah bahasa markup serbaguna yang direkomendasikan W3C untuk mendeskripsikan berbagai macam data. *XML* menggunakan *markup tags* seperti halnya *HTML* namun penggunaannya tidak terbatas pada tampilan halaman *web* saja. Tag *XML* dapat dibuat sendiri sesuai dengan kebutuhan. *XML* berbentuk struktur data pohon.

2.6.2 *RPC*

RPC adalah sebuah protokol yang memungkinkan program komputer berjalan pada satu *host* dan mengakibatkan kode dapat dieksekusi pada *host* yang lain tanpa kebutuhan programmer secara eksplisit menkodekan ini. Ketika kode yang dipertanyakan tersebut ditulis dengan prinsip-prinsip berorientasi obyek, *RPC* adalah sering disandarkan sebagai perintah jarak jauh atau metode perintah jarak jauh.

RPC adalah paradigma yang mudah dan poluler dalam mengimplementasikan model *client-server* dari komputerisasi terdistribusi. Sebuah *RPC* ada berkat inisiatif client (*caller*) yang mengirimkan pemesanan kepada sistem jarak jauh (*server*) untuk mengeksekusi prosedur tertentu yang disertai argumen. Hasil

pemesanan tersebut dikembalikan oleh klien (*caller*). Dalam hal ini banyak sekali variasi dan penyimpangan dalam berbagai macam implementasinya, sebagai akibat bervariasinya protokol *RPC* yang berbeda.

Untuk memungkinkan *server* dapat diakses oleh *client* yang berbeda, sejumlah standarisasi sistem *RPC* telah dibuat. Kebanyakan ini menggunakan *IDL* (*Interface Deskription Language*) yang memungkinkan berbagai macam *platform* menggunakan *RPC*. Contoh dari sistem tersebut diantaranya Sun *RPC* (dikenal juga *ONC RPC*), *DCE* (Distributed Computing Enviroment), Microsoft *DCOM* dan *activeX* yang berbasis *DCE*, serta *CORBA*.

Saat ini banyak *vendor* telah mulai menggunakan *XML* sebagai *IDL*, dan *HTTP* sebagai *network protocols*. Kelebihan dari sistem ini, dikenal sebagai *web services*, adalah standarisasi dan sederhana dimana *IDL* adalah text file yang dipahami secara luas serta *HTTP* yang dibangun diatas sistem oprasi yang modern. Satu contoh dari system *RPC* adalah *SOAP*, dikembangkan dari sistem sebelumnya *XML-RPC*.[TCP07]

2.7 Bahasa Pemrograman *Java*

Java menurut definisi daru *Sun Microsystem* yaitu nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. [SHL06]

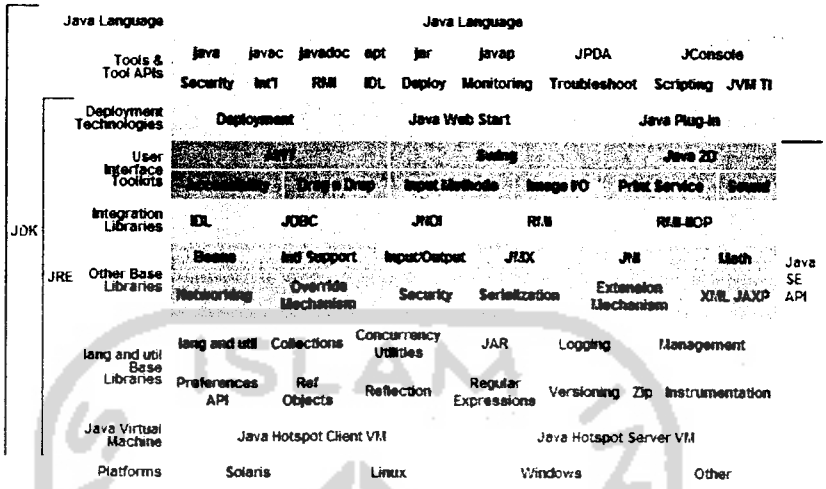
Bahasa *java* dapat dijalankan pada berbagai sistem operasi asalkan terdapat *JVM (Java Virtual Machine)*. Program *java* dapat dijalankan dengan melakukan *compile* .java menjadi .class (*bytecode*), dan untuk menjalankan *bytecode* dibutuhkan *JRE (java runtime environment)*.

2.7.1 *Java™ Platform, Standard Edition*

Java™ Platform, Standard Edition adalah inti dari bahasa pemrograman *java*. *JDK (Java Development Kit)* adalah salah satu *tool* dari *Java Standard Edition* untuk mengkompilasi dan menjalankan program *java*, di dalamnya terdapat *tool* untuk mengkompilasi program *java* dan *JRE*. [SHL06]

Tool Java SE yang digunaan dalam penelitian ini adalah sun *JDK 1.6.*

Gambar 2.4 adalah gambar ruang lingkup *Java™ Platform, Standard Edition6*



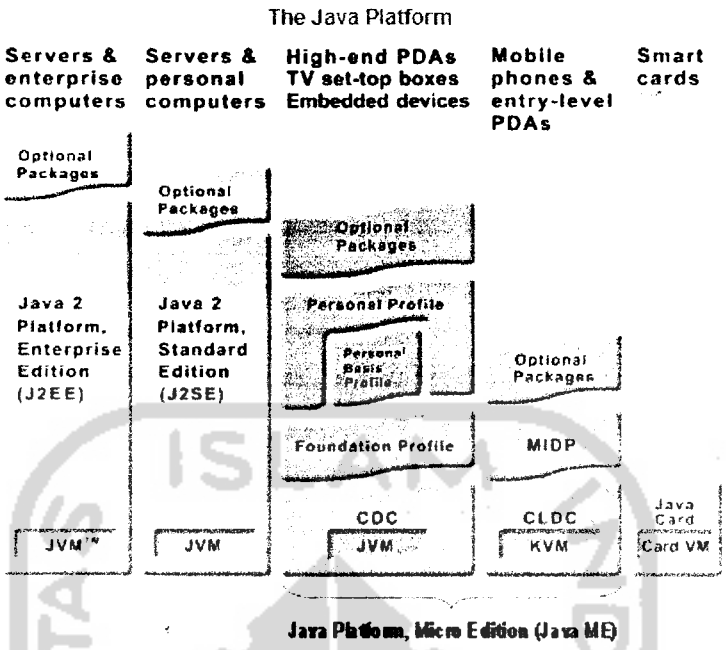
Gambar 2.4. Ruang Lingkup *Java Development Kit (JDK) 1.6*

2.7.2 *Java™ Platform, Micro Edition*

Adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak *java* pada barang elektronik beserta perangkat pendukungnya. Biasa digunakan pada telepon selular, *pager*, *Personal Digital Assistants (PDA)* dan sejenisnya. [SHL06]

Java Micro Edition adalah bagian dari *Java Standard Edition* sehingga juga memiliki keterbatasan terutama jika diaplikasilan pada ponsel. *Java Micro Edition* sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi merek ponsel maupun kemampuan ponsel dan dukungan terhadap teknologi *Java Micro Edition*. Keterbatasan lainya adalah pada ukuran aplikasi, karena memori pada ponsel terbatas. Sebagaian besar ponsel tidak mengijinkan aplikasi *Java Micro Edition* menulis *file* karena alasan keamanan.

Arsitektur *Java Micro Edition* dapat dilihat pada gambar 2.5.



Gambar 2.5. Arsitektur *Java Micro Edition*

a. *Configuration* merupakan java library minimum. Hanya mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar *device*. *Configuration* dibagi mejadi dua yaitu *CDC* (*Connected Device Configuration)* dan *CLDC* (*Connected Limited Device Configuration*), pada penelitin ini digunakan *CLDC* karena sesuai dengan ukuran memori yang digukana oleh ponsel.

b. *Profile* yaitu membahas suatu yang spesifik untuk sebuah perangkat. Terdapat dua buah *profile* yaitu *MIDP* dan *Foundation Profile (FP). MIDP* digunakan oleh *configuration CLDC* dan *Foundation Profile* digunakan oleh *CDC.*

c. *JVM* adalah mesin *runtime* yang digunakan oleh *java* untuk menjalankan *bytecode* atau file *class* yaitu file hasil compilasi dari file *.java.* Pada *CLDC* menggunakan *JVM* bernama *KVM (Kilo byte Virtual Machine)* dan pada *CDC* menggunakan jvm bernama *CVM (C-Virtual Machine).*

b. Atribut *MIDlet*

Setiap *MIDlet* yang dibuat pasti memiliki atribut yaitu semua informasi yang ada pada *MIDlet* tersebut atau meta data *MIDlet*, beberapa atribut tersebut adalah MIDlet-name, MIDlet-version, MIDlet-vendor dan lain sebagainya.

Java Archive

Dalam mendistribusikan aplikasi *MIDlet*, seharusnya aplikasi dilakukan pembungkusan (archive) data, yaitu semua *class* atau *library* aplikasi yang dibutuhkan dipaket atau dibungkus menjadi satu sehingga dapat dengan mudah didistribusikan.

Java memiliki dua buah *archive* yaitu *JAD (Java Application Descriptor)* digunakan untuk keperluan pemetaan file dan berisi kumpulan deskripsi file *JAR* dan *JAR* adalah kumpulan *resource* yang dibutuhkan oleh aplikasi dibungkus menjadi satu.

2.7.2.2 *Java™ Platform, Micro Edition Wireless Toolkit (JME WTK)*

Adalah kakas yang mneyediakan lingkungan emulator. *Java™ Platform, Micro Edition WTK* berbasiskan *CLDC* dan *MIDP*. [SHL06]

Java™ Platform, Micro Edition WTK adalah program yang meniru kerja ponsel yang didukung oleh *MIDP* oleh karena itu, belum tentu *MIDlet* yang berjalan di *emulator* juga berjalan pada ponsel yang sebenarnya.

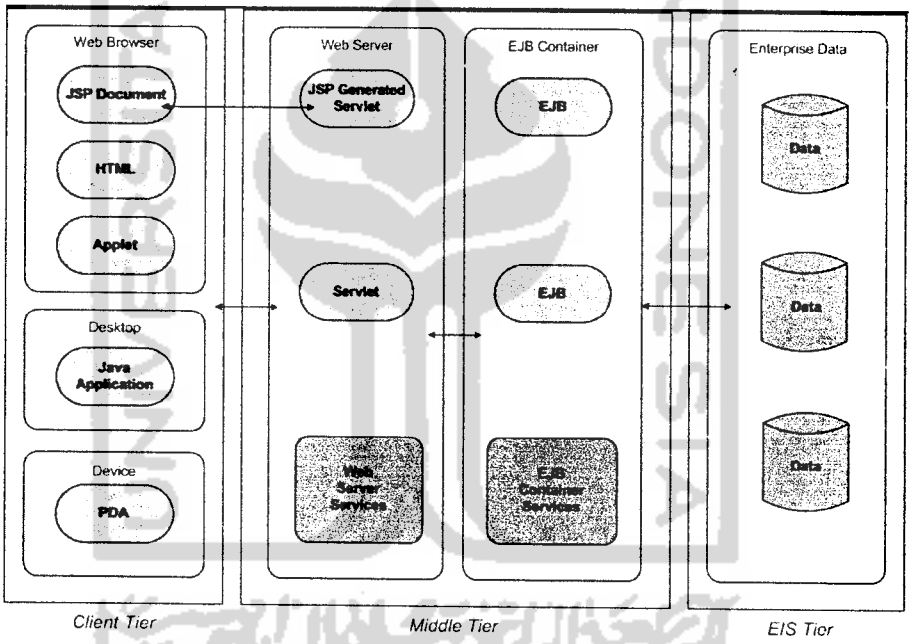
2.7.3 Java™ Platform, Enterprice Edition

Java™ Platform, Enterprice Edition adalah kumpulan teknologi yang cukup kuat dan berada di atas lingkungan *Java™ Platform, Standard Edition. Java™ Platform, Enterprice Edition* berbasis pada *java2* yang berusaha untuk menyediakan sebuah lingkungan aplikasi yang bersifat reliable dan stabil serta dapat dijalankan pada beberapa lingkungan sistem operasi. [SHL06]

Java Enterprice Edition menggunakan modul aplikasi multitier terdistribusi untuk aplikasi perusahaan. *Application logic* dibagi menjadi komponen-komponen sesuai dengan fungsinya dan komponen-komponen aplikasi lainya membuat *Java*

Enterprice Edition terinstal di komputer yang berbeda bergantung pada *tier* di lingkungan *Java Enterprice Edition applications* yang terbagi menjadi :

a. Komponen klien-*tier* yang berjalan di komputer *client*.
b. Komponen *web tier* yang berjalan di *Java Enterprice Edition server*, berupa *servlet* dan *JSP*.
c. Komponen business yang berjalan di *Java Enterprice Edition server*, merupakan kode bisnis, dimana *logic* yang mengatasi atau menangani kebutuhan domain bisnis tertentu seperti perbangkan atau keuangan, ditangani oleh *enterprise bean* yang berjalan di sisi tier bisnis.
d. *Enterprise Information System (EIS)-tier software* yang berjalan di *EIS server*.



Gambar 2.7 Arsitektur Aplikasi *Multi- Tier*

Java™ Platform, Enterprice Edition menyediakan dua buah *container* besar yaitu *container EJB* dan *container web* dimana *container EJB* digunakan untuk mengelola dan mengeksekusi *Enterprise Bean* yang juga disebut dengan *bean* dan

container web digunakan untuk mengelola dan mengeksekusi *servlet* serta *Java Server Pages* atau yang disebut juga dengan *JSP*.

Package pada Java™ Platform, Enterprice Edition dapat berupa *Enterprise Archive (EAR)*, *Java Archive(JAR)* yang merupakan kumpulan file dalam sebuah packet dan *Web Archive(WAR)* dimana *EAR* biasanya merupakan gabungan dari file-file *JAR* yang biasa digunakan oleh bean, sedangkan *WAR* biasa digunakan oleh *servlet* dan *JSP*.

Java™ Platform, Enterprice Edition mempunyai modul yaitu :

a. *EJB, Enterprise Java Bean*, terdiri dari *enterprise bean* dan *descriptor* dari *EJB deployment*. *EJB* biasanya menggunakan package berupa *JAR* dengan ekstensi file .jar.
b. *Web*, seperti *servlet* dan *JSP*. Terdiri dari file-file *class servlet* atau file-file class yang dibutuhkan *JSP*, file gambar dan *HTML*, dan *descriptor* dari *WEB deployment*. *Package* yang digunakan biasanya adalah *WAR* dengan ekstensi .war
c. *Modul Adapter Resource*, berisi seluruh *interface* java, kelas, *native libraries*, *resource descriptor* dan dokumen lainya dengan ekstensi *.rar (Resource adapter archive)*.
d. Aplikasi *client*, terdiri dari file class client dan *descriptor client*. *Package* yang digunakan adalah *JAR* dengan ekstensi file .jar.

Java™ Platform, Enterprice Edition server juga menyediakan layanan dalam bentuk kontainer untuk tiap tipe. Kontainer adalah antarmuka di antara sebuah komponen dan *platform low spesifik* yang mendukung komponen. Sebelum sebuah web, enterprise bean dan komponen client dapat dieksekusi, ia harus dirakit ke aplikasi *Java™ Platform, Enterprice Edition* dan disebarkan ke kontainernya. Proses perakitan melibatkan setting container untuk tiap komponen di aplikasi *Java™ Platform, Enterprice Edition* itu sendiri. Kontainer juga mengatur layanan yang tidak dapat dikonfigurasi seperti *enterprise bean* dan daur hidup *servlet*, koneksi ke *database*, *data persistence* dan akses ke *platform Java Enterprice Edition API*.

Java™ Platform, Enterprice Edition 5.0 mempunyai layanan standard yaitu :

a. HyperTeks Transfer Protocol / Hyperteks Transfer Protocol Secure (HTTP/HTTPS).
b. *Java Transaction API (JTA)* yang menyediakan antarmuka transaksi untuk menempelkan sistem pemrosesan transaksi.
c. *Remote Method Invocation to Internet Inter-ORB Protocol (RMI-IIOP),* komponen EJB menggunakan layanan ini untuk melakukan komunikasi.
d. *Java Database Connectivity (JDBC),* JDBC menyediakan antarmuka untuk mengeksekusi pernyataan SQL (Structured Query Language).
e. *Java Message Service (JMS),* sebuah layanan pesan asinkron yang memungkinkan user mengirim atau menerima pesan.
f. *Java Mail,* memungkinkan pengiriman dan penerimaan email melalui transportasi pesan dan penyimpanan pesan.
g. *Java Naming and Directory Interface (JNDI)* digunakan untuk megakses direktori seperti Light Weight Directory Access Protocol (LDAP).
h. *JavaBeans Activation Framework (JAF),* java mail menggunakan JAF untuk menangani berbagai macam Multipurpose Intenet Mail (MIME) yang mungkin berada di dalam pesan email.
i. *Web Service.*

2.7.3.1 *Servlet*

Servlet adalah kelas-kelas di *java* yang digunakan untuk memperluas kemampuan server yang menyimpan aplikasi yang akan diakses melalui model pemrograman berbasis *request-response*. *Servlet API* terdiri dari 2 *package,* yaitu *javax.servlet* dan *javax.servlet.http.* [UTD04]

Package berisi kelas dan *interface* yang spesifik ke *HTTP.* Meskipun servlet dapat merespon berbagai macam tipe *request,* biasanya digunakan untuk memperluas kemampuan yang disimpan di dalam *web server.*

Pada sisi *server, servlet* memproses permintaan dan menghasilkan *output* dinamis yang dikirim kembali ke *client. Servlet* umumnya tidak memiliki *user interface,* tetapi kita bisa menyediakannya pada sisi *client. Servlet* mempunyai

keuntungan yaitu pada *load* pertama kali disimpan dalam *RAM* dan seterusnya memanggil yang sudah ada pada *RAM*.

Tabel 2.5 penjelasan singkat dari sejumlah class dan *interface* yang sering digunakan pada package *servlet*:

Tabel 2.5. Metode-Metode pada *Package Servlet*

Nama	Jenis	Penjelasan
GenericServlet	Class	Mendefinisikan suatu *servlet* generic, yang *protocol-independent*.
RequestDispatcher	Interface	Mendefinisikan suatu object yang menerima *request* dari client dan mengirimkanya ke sumberdaya manapun (semacam *servlet*, *HTML*, *JSP*) di *server*.
Servlet	Interface	Menentukan method yang harus diimplementasikan semua *servlet*.
ServletConfig	Interface	Suatu object konfigurasi *servlet* yang dipakai oleh servlet container untuk menyampaikan informasi ke sebuah *servlet* selama inisialisasi.
ServletContext	interface	Menyatakan sejumlah method yang dipakai oleh suatu *servlet* untuk berkomuunikasi dengan *servlet container*.
ServletException	Class	Mendefinisikan suatu *Exception* umum yang bisa di-throw oleh *servlet* saat menemui kesulitan.
ServletInputStream	Class	Menyediakan input stream untuk pembacaan data binary dari *request client*, termasuk method readLine yang efesien untuk membaca data suatu baris pada satu saat.
ServletOutputStream	Class	Menyediakan output stream untuk pengiriman data binary ke client.
ServletRequest	Interface	Mendefinisikan suatu object untuk memberikan informasi *request client* ke *servlet*.
Servlet Response	Interface	Mendefinisikan suatu object untuk membantu *servlet* mengirim *response* ke *client*.
SingleThreadModel	Interface	*Servlet* menangani satu *request*.

2.7.3.4 *JNDI*

Java Naming And Directory Interface adalah *API* yang menyediakan fungsionalitas penamaan dan direktori ke suatu aplikasi yang ditulis menggunakan java. Aplikasi *Java* menggunakan *JNDI API* untuk mengakses berbagai layanan *naming dan directory*. [UTD04]

Untuk menggunakan *JNDI*, kita harus mempunyai kelas-kelas *JNDI* dan minimal satu atau lebih *service provider*, service profider untuk *Layanan Naming Directory* adalah:

a. *Leightweight Directory Access Protocol (LDAP).*
b. *Common Object Request Broker Architecture (CORBA).*
c. *Common Object Service(COS) name service.*
d. *Java remote Methode Invocation (RMI) Registry.*

Umumnya *JNDI* dibagi menjadi lima *package* yaitu :

a. *javax.naming.*
b. *javax.naming.directory.*
c. *javax.namingevent.*
d. *javax.naming.ldap.*
e. *javax.naming.spi* .

2.8 *Database*

Database adalah sekumpulan data yang saling berhubungan. Data adalah fakta yang dapat direkam dan memiliki arti implicit. Sebagai contoh, nama, nomor telepon, dan alamat seseorang. Kumpulan data yang berhubungan dan memiliki arti secara implisit disebut database. [CAH06]

Database Management System (DBMS) adalah sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara suatu database, contoh *DBMS* yang ada dan banyak sekarang ini adalah *MySQL* dan *Oracle*. Dengan kata lain, *DBMS* merupakan *general-purpose software* sistem yang memfasilitaskan proses-proses seperti pendefinisian, pembuatan, manipulasi dan sharing *database* antara berbagai pengguna dan aplikasi. Beberapa fungsi lainya yang disediakan oleh *DBMS* adalah keamana dan pemeliharaan. Informasi yang disimpan pada

katalog *DBMS* disebut juga dengan meta data yang menjelaskan struktur utama *database*.

2.8.1 *SQL (Structure Query Language)*

SQL digunakan untuk berkomunikasi dengan suatu *database*. Pernyataan-pernyataan *SQL* digunakan untuk melakukan fungsi-fungsi seperti *update* data pada *database* atau pengambilan data pada *database*. [CAH06]

Pernyataan *SQL* dapat dibagi menjadi dua kategori yaitu pernyataan-pernyataan *Data Manipulation Language (DML)* dan *Data Definition Laguage (DDL)*. Pernyataan pernyataan *DML* berkaitan dengan data, baik pemngambilan atau pemodifikasian data supaya tetap *up-to-date*, sedangkan pernyataan-pernyataan *DDL* berkaitan dengan pembuatan atau pengubahan *table* dan objek-objek database lainya seperti *view* dan *index*.

Berikut ini merupakan pernyataan-pernyataan *DML*, antar lain :

a. *SELECT* digunakan untuk *query* dan menampilkan data dari sebuah *database*.
b. *INSERT* digunakan untuk mengisi *table* yang baru dibuat.
c. *DELETE* digunakan untuk menghapus sebuah baris tertentu.
d. *UPDATE* digunakan untuk mengubah nilai dari suatu kolom.

Berikut ini merupakan pernyataan-pernyataan *DDL*, antara lain:

a. *CRATE TABLE* digunakan untuk membuat table dengan nama kolom yang ditentukan oleh pengguna.
b. *DROP TABLE* digunakan untuk menghapus semua baris atau *record* dan menghapus definisi tabel dari *database*.
c. *ALTER TABLE* digunakan untuk menambah atau menghapus sebuah kolom, dari suatu tabel.

2.8.2 *JDBC*

JDBC (Java Database Connectivity) merupakan *API (Application Programming Interface)* untuk mengakses secara virtual berbagai jenis tabular data. *JDBC API* terdiri dari sekumpulan *class* dan *interface* yang ditulis dalam bahasa pemrogrman *java* yang menyediakan standar *API* sebagai *tools* dalam

menggembangkan *database* dan membarikan kemudahan bagi pengembang untuk membuat aplikasi-aplikasi *database* yang sepenuhnya dalam bahasa pemrograman *java*.

JDBC API memungkinkan untuk melakukan tiga hal, yaitu:

a. Membangun koneksi dengan *database* atau berbagai jenis sumber data tabular data.
b. Mengirim pernyatan-pernyataan *SQL*.
c. Memproses hasil.

JDBC 3.0 API terdiri dari dua *package* yaitu :

a. *java.sql package*, merupakan *core package* dari *JDBC API*.
b. *javax.sql package*, merupakan *package* tambahan untuk menambahkan kemampuan *server-side* dari *JDBC API*.

2.9 *UML*

Penggunaan *UML*, merupakan bahasa yang *process-independent*, yaitu bahasa yang saling bebas terhadap proses yang ada di dalamnya. Tujuan penggunaan *UML* yaitu menyederhanakan proses perancangan (deskripsi dan desain) sebelum dilakukan pembangunan sistem. Pembuatan *UML* merupakan sebuah proses yang iteratif dan inkremental. *UML* disini dipandang dari sudut sketsa (bersifat ekploratif) secara *forward engineering* yaitu membuat sebuah diagram *UML* terlebih dahulu sebelum melakukan *coding*.

Seperti yang dikeluarkan oleh *OMG (Object Management Group)* yaitu badan yang melakukan standarisasi *UML*, bahwa *UML* mempunyai 13 diagram resmi dan *UML* memiliki beberapa konsep dasar yang diabstraksikan dalam bentuk *structural classification*, *dynamic behavior*, dan *model management*

2.9.1 *Use Case Diagram*

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri. *Use case* dimulai dengan mejabarkan skenario yang berhubungan dengan sistem. Inti dari skenario adalah pengguna mempunyai tujuan yang sama kepada

sistem. Sebuah *use case* adalah serangkaian sekenario yang dikemas menjadi satu oleh tujuan pengguna umum. Para pengguna dapat disebut sebagai aktor.

Seorang aktor dapat menggunakan banyak *use case* dan begitu pula sebaliknya satu *use case* dapat digunakan oleh beberapa aktor. Aktor tidak hanya manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem tersebut adalah sebuah aktor. Setiap *use case* memiliki aktor utama untuk meminta sistem memberikan sebuah layanan.

Setiap langkah dalam *use case* merupakan sebuah elemen dalam interaksi antara aktor dan sistem. Setiap langkah merupakan pernyataan sederhana dan dengan jelas menunjukan siapa yang menjalankan langkah tersebut, dan langkah tersebut harus merupakan tujuan aktor bukan mekanisme aktor.

2.9.2 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti pewarisan, *asosiasi*, dan lain-lain.

Class memiliki tiga area pokok :

a. Nama (dan *stereotype*).
b. Atribut.
c. Metoda.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
c. *Public*, dapat dipanggil oleh siapa saja.

Class juga mempunyai hubungan antar *class*, yaitu:

a. *Asosiasi*, yaitu hubungan *statis* antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
b. *Agregasi*, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
c. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.[PUR03]

2.9.3 *Sequence Diagram*

Sequence diagram, secara khusus, merupakan :

a. Penjabaran sebuah skenario sederhana.
b. Menunjukan penerapan skenario.
c. Menunjukan interaksi dengan menampilkan setiap partisipan dengan garis alir secara vertikal dan pengurutan pesan dari atas ke bawah. [FOW05]

BAB III

METODOLOGI

3.1 Analisis Kebutuhan

3.1.1 Metode Analisis

Metode analisis dalam penelitian ini, digunakan metode dengan pendekatan berarah-obyek *(object oriented)*, dengan pendekatan tersebut, program bersifat *moduler*. Dalam konsep pemrograman berorentasi obyek, modul-modul tersebut dapat digunakan kembali (*reusable*), sehingga dalam pembuatan aplikasi yang besar, dimudahkan bagi pengembang untuk memelihara aplikasi yang dikembangkan. Metode analisis penelitian ini juga dilakukan dengan pengumpulan data melalui observasi, studi literatur berupa buku dan sumber lainya yang mendukung dan berkaitan dengan obyek yang diteliti yaitu *OPAC (Online Public Access Catalog)*.

Terdapat standar dalam perancangan perangkat lunak berarah obyek yaitu dengan standar *UML*. *UML* digunakan untuk memodelkan perancangan piranti lunak (sistem) sebelum rancangan tersebut diimplementasikan ke dalam bahasa pemrogrman. *UML* memodelkan perancangan dengan beberapa macam diagram, diantaranya *use case diagram*, *class diagram*, dan *sequence diagram*.

Dengan digunakanya rancangan model sebelum dilakukan implementasi perangkat lunak, diharapkan pengembangan sistem dapat berjalan sesuai dengan rancangan yang telah dibuat sebelumnya dan mencapai tujuan yang diharapkan dari proses perancangan.

3.1.2 Hasil Analisis

Dari analisis yang telah dilakukan sebelumnya, maka akan didapatkan hasil analisis. Hasil analisis adalah langkah untuk menentukan kebutuhan perangkat lunak. Kebutuhan yang dimagsud adalah kebutuhan masukan *(input)*, proses

keluaran (*output*), perangkat lunak, antarmuka dari sistem yang telah dirancang dan yang akan diimplementasikan kedalam bahasa pemrograman.

3.1.2.1 Kebutuhan Masukan

Untuk mempermudah dalam menganalisis kebutuhan masukan sistem maka, sistem dapat dipecah menjadi *sub-sub* sistem dari sistem yang dipandang secara global. Kebutuhan masukan dapat lebih diperinci sebagai berikut :

a. Pada sisi *client* (*MIDlet*) terhadap *server* yang berhubungan secara langsung yaitu server yang kali pertamanya dituju (*default*) sebagai *server* utama atau *server gateway*.

   Pada kondisi ini, kebutuhan masukan berupa atribut penelusuran sederhana (judul koleksi dan pengarangnya) dan atribut penelusuran *advance* (judul, jenis, pengarang, nomor *DDC*), sebagai contoh, *client* meminta penelusuran sederhana, dengan kata kunci *"Sistem Informasi"* sebagai parameter yang akan dilempar kepada server utama (sebagai masukan kepada server utama).

b. Pada sisi *server* (*server XML-RPC* atau *gateway*) kepada *server* lainya (*server-server* perpustakaan yang telah terdaftar).

   Kondisi ini adalah implementasi dari sistem terdistribusi menggunakan *XML-RPC* yaitu *server* utama (*server gateway*) dapat melakukan permintaan penelusuran kepada *server-server* lainya dalam hal ini adalah *server* lain perpustakaan universitas yang terdaftar dengan cara melempar parameter penelusuran yang dikirim dari *client* kepada *server* perustakaan untuk dilakukan penelusuran. Pada kondisi ini, *server* yang meminta penelusuran (apakah itu *server gateway* kepada *server* perpustakaan atau antar *server* perpustakaan) bisa disebut juga *server* sebagai client bagi *server-server* lainya. Kebutuhan masukan yang diperlukan bagi *server-server* lainya adalah *input* parameter-parameter *method* dari server yang sudah diketahui oleh server yang melakukan permintaan penelusuran. Ini berarti, setiap *server* (*gateway*, UII, UGM) harus mempunyai metode atau dari *server-server* lainnya yang dapat dihubungi atau dipanggil.

te
y
in
di

Pada saat *server gateway* meminta penelusuran kepada server perpustakaan tidak diketahui oleh *user* (penelusur), *user* hanya mengetahui pengaksesan terhadap *server* perpustakaan yang dituju.

3.1.2.2 Kebutuhan Proses

3.1.2.4
K
penger
a. Si
cl
K
di
dil
be
be
b. Pe
sis
ber
kar
dar
dap
Jav
dari
c. *Jav*
1.6,
(*MI*
d. *Jav*
dan
den
e. *IDE*
dan

Setelah dilakukan analisis pada sistem, maka dibutuhkan proses-proses sebagai berikut:

a. Pembuatan aplikasi pada *server* (*server gateway* dan perpustakaan) yang dapat memproses dan memberikan permintaan penelusuran dari *client* atau dengan kata lain, aplikasi pada *server* dapat diartikan sebagai program yang berada pada *server* yang akan memberikan jawaban dari permintaan penelusuran katalog kepada *client* dengan cara mengakses data pada *database servernya*.
b. Manajemen *database* koleksi perpustakaan (katalog) pada masing-masing *server* perpustakaan .
c. Penelusuran data koleksi perpustakaan dengan meminta kepada *server* yang pertama kali dituju (*gateway*) untuk melewatkan parameter penelusuran kepada *server* perpustakaan.
d. Menampilkan hasil penelusuran pada *MIDlet* dengan melihat detail dari penelusuran tersebut.

3.1.2.3 Kebutuhan Keluaran Sistem

Sedangkan kebutuhan keluaran sistem dapat juga dipandang dari berbagai sisi, sama halnya dengan saat dilakukanya analisis terhadap kebutuhan masukan. Kebutuhan keluaran sistem dapat dipandang sebagai berikut:

a. Dari sisi *server* perpustakaan yang melakukan permintaan penelusuran terhadap *server* perpustakaan lainya, kebutuhan keluaran adalah hasil *query* dari *database server* pada *server* perpustakaan dan mengembalikanya kepada *server* perpustakaan yang melakukan permintaan.
b. Dari sisi *server* yang dituju oleh *client* atau *user*, selain kebutuhan dari sisi *server* yang melakukan permintaan penelusuran terhadap *server* lainya (jika tidak ditemukan dalam *server* perpustakaan yang pertama kali dituju),

termasuk hasil penelusuran *database server* pada *server* itu sendiri (*server* yang melakukan permintaan kepada *server* yang lainya) kepada *client*, yaitu informasi mengenai katalog yang ditelusuri atau diinginkan oleh *user* jika ditemukan pada *server* perpustakaan yang pertama kali dituju.

3.1.2.4 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan untuk dilakukan-nya penelitian pengembangan sistem *mobile OPAC* berbasis *XML-RPC* adalah sebagai berikut:

a. Sistem Operasi dalam pengembangan perangkat lunak pada sisi *server* dan *client* adalah *Microsoft Windows XP Professional SP2 Edition* dan *Linux Kubuntu*, meskipun tidak menutup kemungkinan untuk membangun *server* diatas pada sistem operasi lainya (*Apple MAC OS*, *Solaris* dll), ini dapat dilakukan karena konsep dari sistem terdistribusi adalah ketidak-bergantungan dari sistem operasi yang digunakan (sistem operasi dapat beragam) dan bahasa pemrograman yang beragam (*resource* yang beragam).
b. Perangkat lunak pada sisi server (*servlet*), tidak hanya dapat berjalan pada sistem operasi *Microsoft Windows* saja, melainkan dapat dijalankan pada berbagai mesin tanpa memperhatikan sistem operasi dan perangkat keras karena dikembangkan dengan bahasa *java*. Konsep ini merupakan konsep dari *java* yaitu *WORA (Write Once Run Anywhere)*, artinya kode program dapat dijalankan dimanapun mesin, dengan syarat mesin tersebut mempunyai *Java Virtual Machine (JVM)* sebagai *interpreter* dari *java code* sebagai hasil dari proses *compile* dari *source code java*.
c. *Java™ Platform, Standard Edition* dengan *JDK* menggunakan *Sun Java JDK 1.6*, sebagai platform dasar pengembangan perangkat lunak, baik di sisi *client* (*MIDlet*) dan *server*.
d. *Java™ Platform, Enterprice Edition Java, Java EE 5.0* sebagai spesifikasi dan *API* yang digunakan dalam pengembangan aplikasi *servlet* pada *server* dengan *container web*.
e. *IDE NetBeans 5.5* digunakan dalam pengembangan sistem pada sisi *server* dan *mobile client*.

f. *NetBeans Mobility Pack 5.5* sebagai *Add-on* (tambahan) NetBeans yang menyediakan pengembangan *MIDlet (visual MIDlet)*.

g. *DBMS (Database Management System)* yang akan digunakan adalah *MySQL server 5.0* dan *Oracle 10g Express Edition (XE) Server* yang digunakan pada server perpustakaan yang berbeda-beda guna memberikan perbadaan spesifikasi mesin yang menandakan adanya komunikasi antar mesin yang berbeda.

h. *NetBeans Profiler 5.5* sebagai *add-on* konfigurasi *IDE NetBeans 5.5.*

i. *NetBeans UML add on* sebagai *Add-on NetBeans* yang menyediakan komponen untuk melakukan *generate code* program menjadi suatu diagram *UML.*

j. *Java™ Platform, Micro Edition Wireless Toolkit (Java ME WTK)* sebagai spesifikasi dan *API* dasar pengembangan *MIDlet. Java ME WTK* juga menyediakan *emulator* dalam menjalankan MIDlet.

k. *Library Apache XML-RPC v3.0* . Merupakan *library* dari *XML-RPC* untuk application server yang dibuat.

l. *Mysql-connector-java-3.1.8-bin.jar* sebagai penghubung *java* dengan *DBMS MySQL server* atau disebut juga *JDBC API (Java Database Connectivity Application Programming Interface).*

m. *Mysql-gui-tools-5.0-r5-win32* sebagai perangkat manajemen *database MySQL* secara visual (*front end*).

n. *Oracle JDBC connector (ojdbc14.jar)* sebagai penghubung *java* dengan *DBMS ORACLE server 10g ExpressEdition (XE).*

o. *Server-server Java EE*, sebagai *container* untuk *source* aplikasi *server. Server* menggunakan *Apache Tomcat 5.5, Sun Java System Application Server 9.0 PE.*

p. Dan *tools* lainya seperti *Apache Ant 1.7* (untuk *proses deploy servlet*).

3.1.2.5 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan adalah sebagai berikut :

Untuk pengembangan aplikasi, penulis menggunakan sebuah *PC* sebagai *server* perpustakaan UII, UGM dan sebuah *laptop* sebagai *server gateway* dan sebuah kabel jaringan RJ-45.

Spesifikasi *PC* yang digunakan yaitu :

a. *Processor Intel Core 2 Duo* 1.86 *GHz.*
b. *LCD Monitor WXGA.*
c. *RAM DDR2* 1 *GB.*
d. *Hardisk* 160 *GB.*

Spesifikasi *laptop* yang digunakan yaitu :

a. *Processor Intel Celeron* 1.6 *Ghz.*
b. LCD Monitor WXGA.
c. RAM DDR2 512 *MB.*
d. *Hardisk* 80 *GB.*

3.1.2.6 Kebutuhan Antarmuka

Kebutuhan antarmuka yang dibutuhkan yaitu *interface* yang berjalan pada telepon seluler (*emulator*) berupa *MIDlet* saja, karena aplikasi pada *server,* hampir tidak menggunakan antarmuka (hanya sebagian saja). Antarmuka yang digunakan pada MIDlet diantaranya adalah:

a. Antarmuka pembukaan (*Splash Screen*).
b. Antarmuka menu utama meliputi antarmuka *OPAC*, *setting URL server gateway* dan bantuan program.
c. Antarmuka *OPAC* dibagi menjadi antarmuka penelusuran sederhana dan *advance*.
d. Antarmuka bantuan dibagi menjadi antarmuka perkenalan , bantuan dan tentang program.
e. Antarmuka untuk melihat hasil penelusuran sederhana.
f. Antarmuka untuk melihat hasil penelusuran *advance* (ahli).
g. Antarmuka untuk peringatan kesalahan menghubungi *server.*

3.2 Perancangan Sistem

3.2.1 Metode Perancangan

Perancangan sistem mobile *OPAC* ini, digunakan metodologi berarah obyek, orieantasi tersebut memandang suatu sistem menjadi obyek-obyek, artinya sistem dipecah menjadi obyek-obyek yang lebih kecil. Sistem *mobile OPAC* yang berarsitektur *multi-tier* juga sangat didukung oleh pendekatan berarah obyek, karena sistem terpecah menjadi lapisan *interface* (antarmuka) untuk *client*, proses bisnis pada *server* dan data pada lapisan informasi. Untuk memodelkannya, digunakan pemodelan *UML* (*Unified Modelling Language*) yang menjadi standar dalam pemodelan berarah objek dan mempunyai karakteristik yaitu bahasa yang *process-independent* (bahasa yang saling bebas terhadap proses didalamnya).

3.2.2 Hasil Perancangan

Aplikasi sistem *mobile OPAC* ini dibagi menjadi tiga, meliputi:

a. Aplikasi *client mobile*, yaitu *MIDlet* (*interface* aplikasi pada *client*) yang berfungsi sebagai penelusur katalog, sebagai antarmuka yang menyediakan sarana input data (parameter) untuk merequest data pencarian katalog kepada *server*.
b. Aplikasi pada *server*, yaitu aplikasi berada pada *server* (aplikasi *web*), yang melewatkan parameter penelusuran dari *MIDlet client (server gateway)* dan yang melakukan proses dari permintaan penelusuran katalog oleh *client* selanjutnya dikembalikanya hasil penelusuran kepada *client*.
c. *Database*, yaitu database pada *server* yang menyimpan data-data koleksi perpustakaan.

Perancangan sistem *mobile OPAC* dengan arsitektur *multi-tier* ini meliputi beberapa tahap perancangan yaitu:

a. Perancangan aplikasi pada pengguna atau *client*, yaitu perancangan antarmuka *MIDlet* untuk pengguna memasukan parameter penelusuran katalog.

b. Perancangan *database* pada *server*, yaitu skema *database* relasional yang nantinya akan diterapkan pada *DBMS (Database Management System)* sebagai *database* koleksi perpustakaan.
c. Perancangan aplikasi pada *server*, yaitu perancangan aplikasi (aplikasi *web*) yang berjalan pada sisi *server* untuk melewatkan dan merespon *request* penelusuran oleh *client*.

3.2.3 Perancangan *Unified Modelling Language (UML) dan Lainya*

Dalam penelitian ini, digunakan beberapa diagram saja yang dapat mewakili perancangan sistem. Dalam perancangan sistem *mobile OPAC* ini digunakan (dibatasi) oleh tiga buah diagram beserta kegunaannya, sebagai berikut:

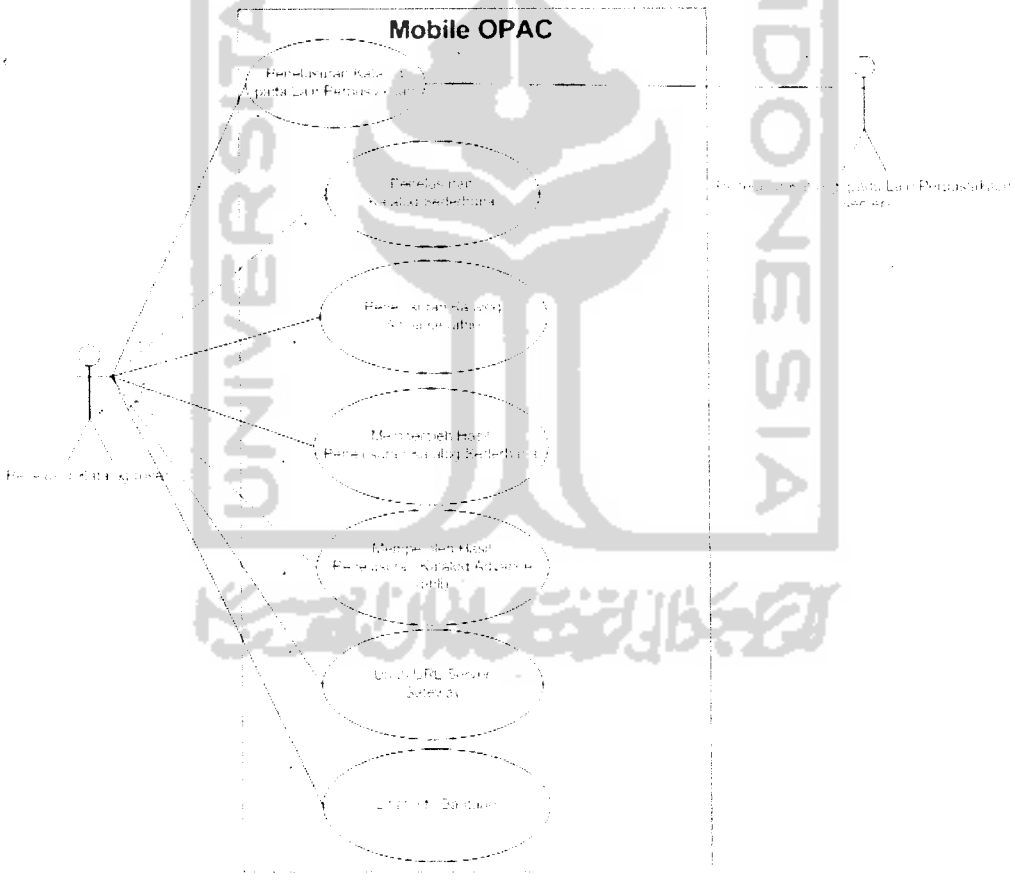
a. *Use Case Diagram*, mempunyai kegunaan yaitu bagaimana pengguna berinteraksi dengan sebuah sistem.
b. *Class Diagram*, mempunyai kegunaan yaitu hubungan-hubungan antara class, fitur, dan
c. *Sequence Diagram*, mempunyai kegunaan yaitu interaksi antar objek.

3.2.3.1 *Use Case Diagram*

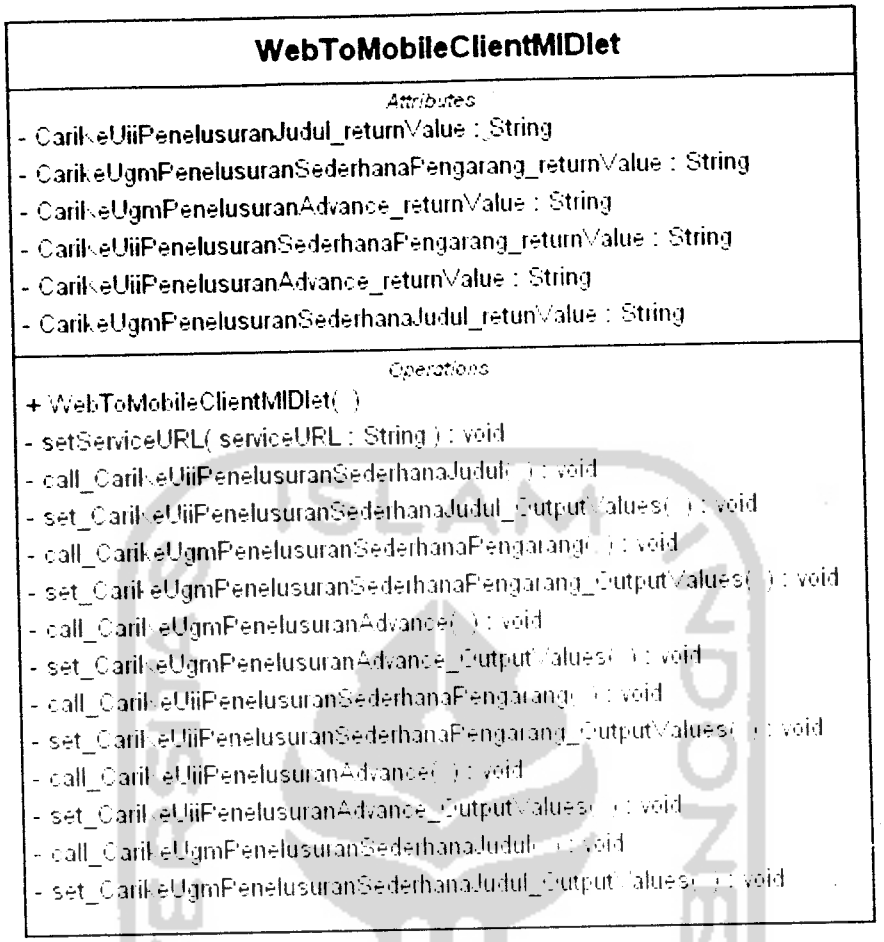
Berikut penjelasan dari diagram *use case* yang menggambarkan hubungan aktor dengan *use case* pada sistem *mobile OPAC* dan untuk lebih jelasnya dapat dilihat pada gambar 3.1 dan gambar 3.2 :

a. Sistem dalam hal ini adalah sistem *mobile OPAC* adalah lingkungan dimana semua komponen *usecase* itu berada dan sistem antar *server* yang terhubung (server *gateway* dan *server-server* perpustakaan).
b. Aktor yang terlibat dalam sistem adalah pengguna *MIDlet* yang berfungsi dalam melakukan penelusuran katalog pada perpustakaan universitas dan *server gateway* yang akan melakukan penelusuran kepada *server* perpustakaan lain yang telah terdaftar dengan cara melemparkan atau memberikan parameter penelusuran kepada server perpustakaan lainya.
c. *Use case* penelusuran sederhana, akan melakukan proses penelusuran secara sederhana yaitu dengan menggunakan parameter penelusuran yang sederhana .

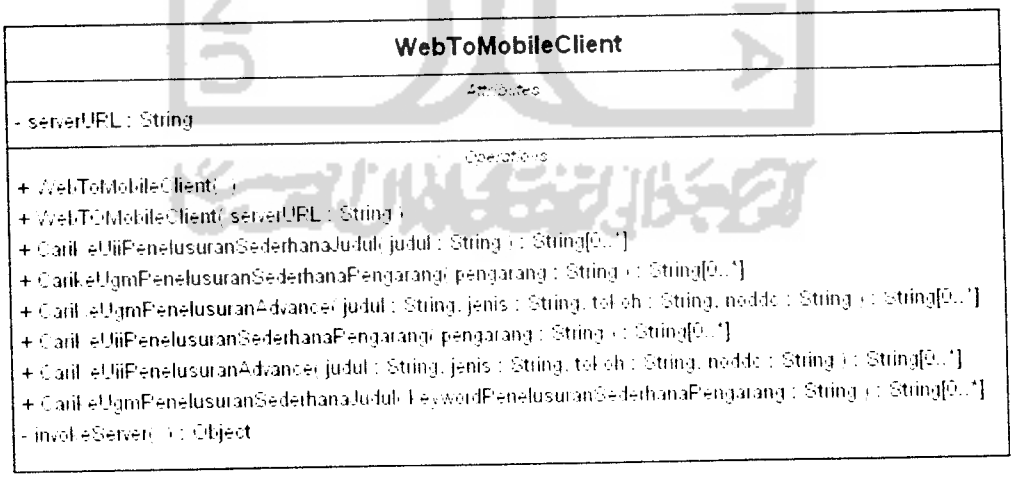
d. *Use case* penelusuran *advance*, akan melakukan proses penelusuran secara detail yaitu dengan menggunakan parameter penelusuran yang mendetail.
e. *Use case* memperoleh hasil penelusuran, akan menampilkan hasil dari penelusuran katalog perpustakaan.
f. *Use case* pencarian pada lain sistem, akan melakukan penelusuran pada sistem lain dengan melempar parameter yang dikirim dari client, baik parameter penelusuran sederhana atau parameter penelusuran detail .
g. *Use case* ubah *URL server gateway*, akan melakukan proses pengubahan *URL server gateway*.
h. *Use case* lihat info bantuan, akan melakukan proses informasi bantuan program (*MIDlet Mobile OPAC*)



Gambar 3.1. *Use Case Diagram Mobile OPAC*



Gambar 3.3. Diagram *Class Midlet (WebToMobileClientMIDlet)*



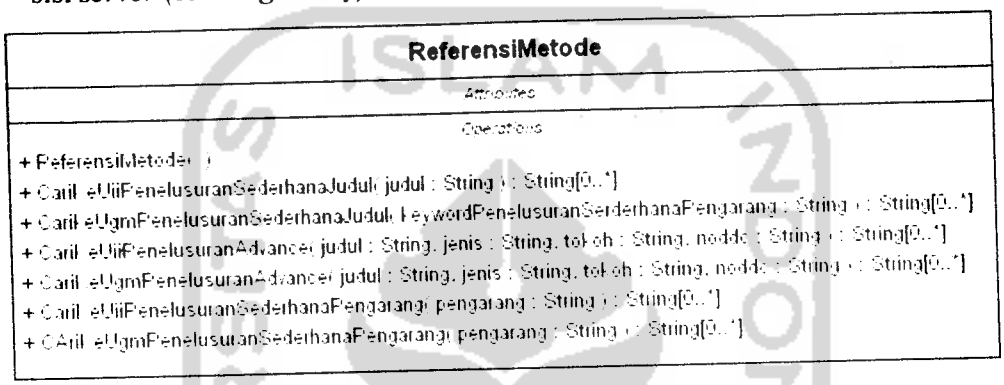
Gambar 3.4. Diagram *Class Stub (WebToMobileClient)*

b. Rincian *Class* pada *server Gateway*

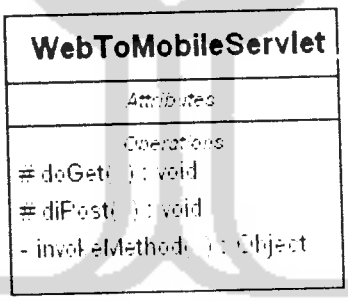
Berikut, rincian *class* yang digunakan dalam *server* utama atau *gateway*

a. *Class ReferensiMetode*, merupakan *class* yang digunakan untuk melempar parameter penelusuran (sederhana dan *advance*) yang dikirim dari *client* kepada *server-server* perpustakaan yang dituju. Untuk lebih jelasnya dapat dilihat pada gambar 3.5.

b. *Class WebToMobileServlet*, merupakan *class* yang berfungsi sebagai *stub* pada sisi *server (server gateway)*. Untuk jelasnya dapat dilihat pada gambar 3.6.



Gambar 3.5. Diagram *class* Proses Bisnis *Server Gateway (ReferensiMetode)*



Gambar 3.6. Diagram C*lass Stub Server (WebToMobileServlet)*

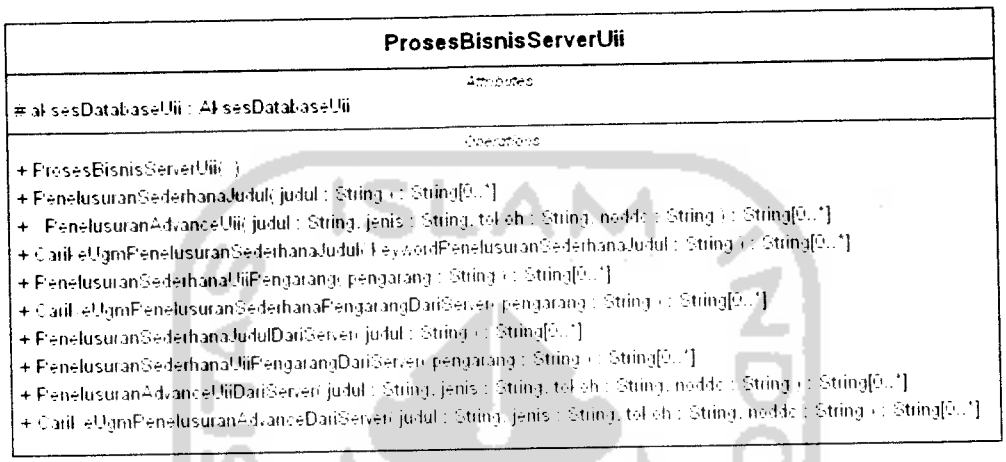
c. Rincian *Class* pada *Server* Perpustakaan UII

Berikut, rincian *class-class* yang digunakan dalam *server* perpustakaan UII,

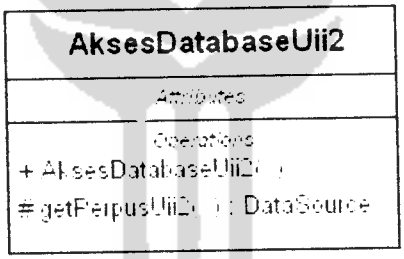
a. *Class ProsesBisnisServerUii*, merupakan *class* yang digunakan untuk menangani proses bisnis *server* perpustakaan UII yaitu me*request* penelusuran sederhana, *advance* dari *client* dan penelusuran kepada lain *server* jika tidak

diketemukan dalam *database* perpustakaan UII. Untuk lebih jelasnya dapat dilihat pada gambar .3.7.

b. *Class AksesDatabaseUii2*, merupakan *class* yang berfungsi sebagai pembangun koneksi kepada *database server* perpustakaan UII. Untuk jelasnya dapat dilihat pada gambar 3.8.



Gambar 3.7. Diagram *Class* Proses Bisnis Perpsus UII



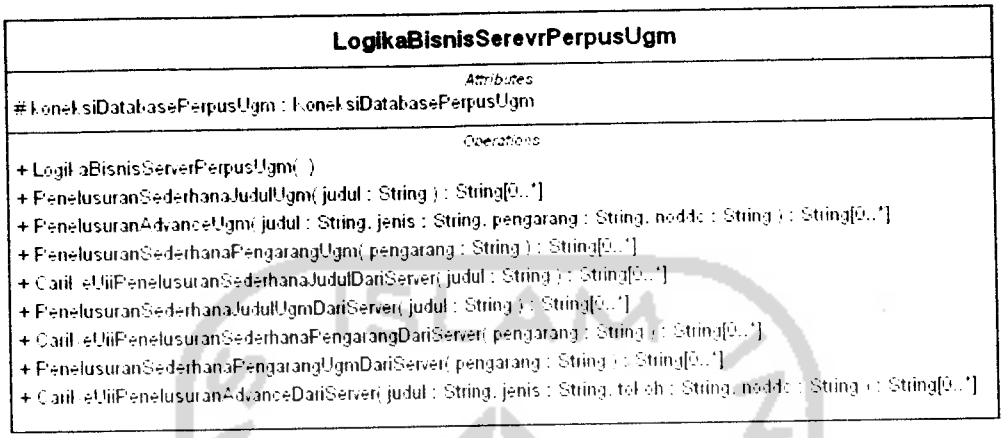
Gambar 3.8. Diagram *Class AksesDatabaseUii2*

d. Rincian *Class* pada *Server* Perpustakaan UGM

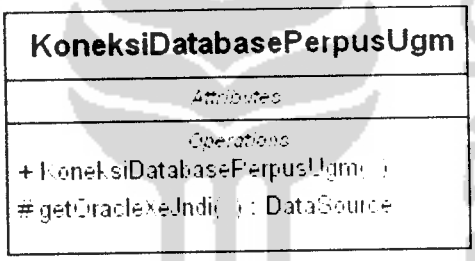
Berikut, rincian *class-class* yang digunakan dalam *server* perpustakaan UGM.

a. *Class LogikaBisnisServerPerpusUgm*, merupakan *class* yang digunakan untuk menangani proses bisnis *server* perpustakaan UGM yaitu me*request* penelusuran sederhana, *advance* dari *client* dan penelusuran kepada lain *server* jika tidak diketemukan dalam *database* perpustakaan UGM. Untuk lebih jelasnya dapat dilihat pada gambar 3.9.

b. *Class KoneksiDatabasePerpusUgm*, merupakan *class* yang berfungsi sebagai pembangun koneksi kepada *database server* perpustakaan UGM. Untuk jelasnya dapat dilihat pada gambar 3.10.



Gambar 3.9. Diagram C*lass LogikaBisnisServerPerpusUgm* UGM



Gambar 3.10. Diagram C*lass KoneksiDatabasePerpusUgm*

2. *Class Diagram* Sesuai dengan Proses

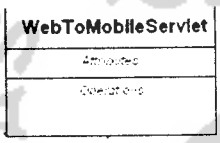
Class-class diagram dalam sistem *mobile OPAC* dapat dihubungkan sesuai dengan proses atau skenario yang ada.

Karena dalam perancangan menggunakan alat pendesain UML dari NetBeans, maka terdapat bebrapa simbol yang dapat dijelaskan lebih lanjut.Berikut adalah keterangan gambar yang dibutuhkan dalam keterikatan class-class di setiap proses yang berjalan pada sistem mobile OPAC :

1. Asosiasi, dilambangkan dengan garis panah dengan anak panah tidak penuh,

   contoh :

2. Garis pembatas antar client dan server, contoh :

3. Class, contoh : 

4. Generalisasi, dilambangkan dengan garis panah dengan anak panah penuh ,

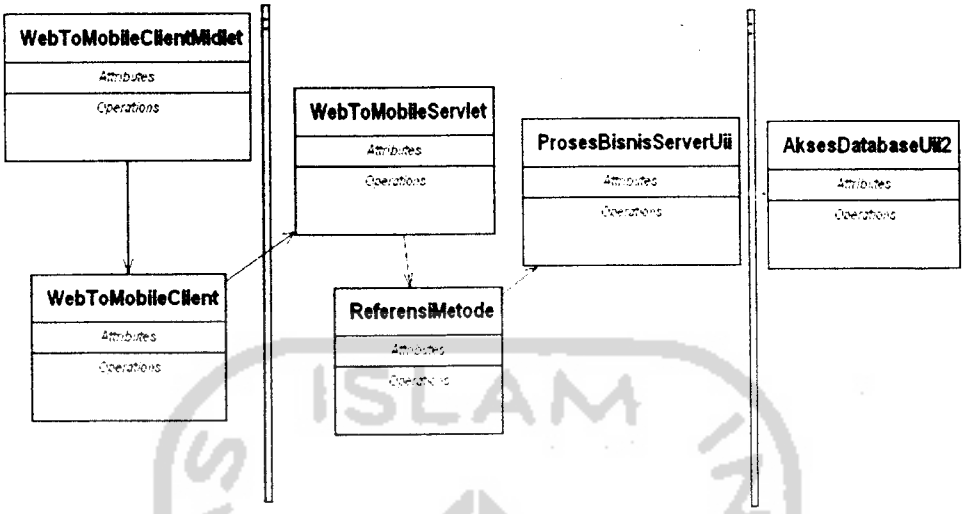
   contoh :

a. *Class Diagram* Proses 1

Class diagram pada proses 1 menunjukan skenario penelusuran sederhana, a*dvance* dan melihat hasil penelusuran sederhana, a*dvance pada* perpustakaan UII).

Terdapat enam buah class yang saling berhubungan atau berasosiasi, yaitu class *WebToMobileClientMidlet* berhubungan dengan *WebToMobileClient*, *WebToMobileClient* berhubungan dengan *WebToMobileServlet*, *WebToMobileServlet* berhubungan dengan *ReferensiMetode*, *ReferensiMetode* berhubungan dengan *ProsesBisnisServerUii* dan *ProsesBisnisServerUii* berhubungan dengan *AksesDatabaseUii2*.

Diagram hubungan *class-class* yang terlibat seperti pada gambar 3.11.



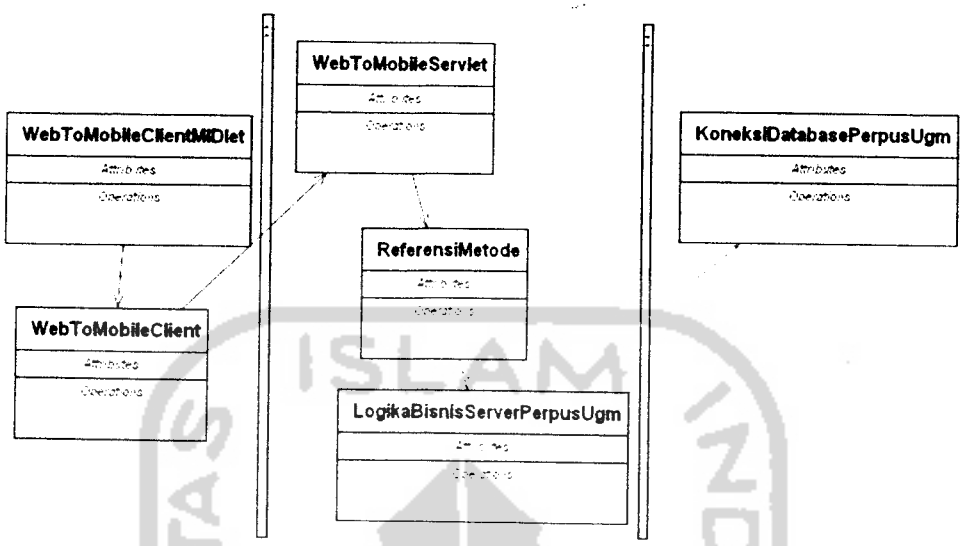
Gambar 3.11. Diagram C*lass* keterikatan C*lass-Class* Penelusuran UII

b. *Class Diagram* Proses 2

Class diagram pada proses 2 menunjukan skenario penelusuran sederhana, a*dvance* dan melihat hasil penelusuran sederhana, a*dvance* pada perpustakaan UGM.

Terdapat enam buah *class* yang saling berhubungan atau berasosiasi, yaitu *class WebToMobileClientMidlet* berhubungan dengan *WebToMobileClient, WebToMobileClient* berhubungan dengan *WebToMobileServlet, WebToMobileServlet* berhubungan dengan *ReferensiMetode*, *ReferensiMetode* berhubungan dengan *LogikaBisnisServerPerpusUgm* dan *LogikaBisnisServerPerpusUgm* berhubungan dengan *KoneksiDatabasePerpusUgm.*

Diagram hubungan *class-class* yang terlibat seperti pada gambar 3.12.



Gambar 3.12. Diagram *Class* keterikatan *Class-Class* Penelusuran UGM

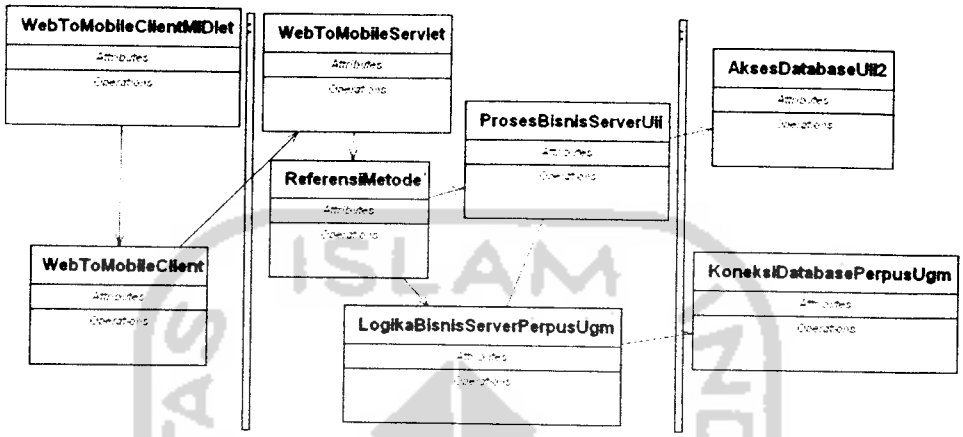
c. *Class Diagram* Proses 3

Class diagram pada proses 3 menunjukan skenario penelusuran sederhana, *advance* perpustakaan UII dan UGM, kepada lain *server* dan melihat hasil penelusurannya.

Terdapat delapan buah *class* yang saling berhubungan atau berasosiasi, yaitu *class WebToMobileClientMidlet* berhubungan dengan *WebToMobileClient*, *WebToMobileClient* berhubungan dengan *WebToMobileServlet*, *WebToMobileServlet* berhubungan dengan *ReferensiMetode*.
Jika dilakukan proses penelusuran pada UGM maka *ReferensiMetode* berhubungan dengan *LogikaBisnisServerPerpusUgm* dan *LogikaBisnisServerPerpusUgm* berhubungan dengan *KoneksiDatabasePerpusUgm*. Jika dilakukan penelusuran pada UII maka *ProsesBisnisServerUii* dan *ProsesBisnisServerUii* berhubungan dengan *AksesDatabaseUii2*.

Class ProsesBisnisServerUii mempunyai hubungan dua arah dengan *class Logika BisnisServerPerpusUgm* jika pada perpustakaan UII tidak diketemukan koleksi pada UII dan begitu pula sebaliknya.

Diagram hubungan *class-class* yang terlibat seperti pada gambar 3.13.



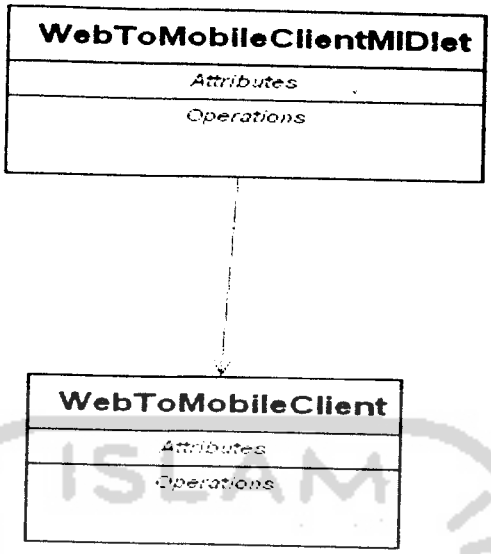
Gambar 3.13. Diagram *Class* Keterikatan *Class-Class* Penelusuran Lain *Server*

d. *Class Diagram* Proses 4

Class diagram pada proses 4 menunjukan hubungan *class-class* pada proses pengubahan *URL* dan melihat bantuan program.

Terdapat dua buah *class* yang berhubungan atau berasosiasi yaitu *class WebToMobileClientMidlet* dan *class WebToMobileClient.*

Diagram hubungan *class-class* yang terlibat seperti pada gambar 3.14.

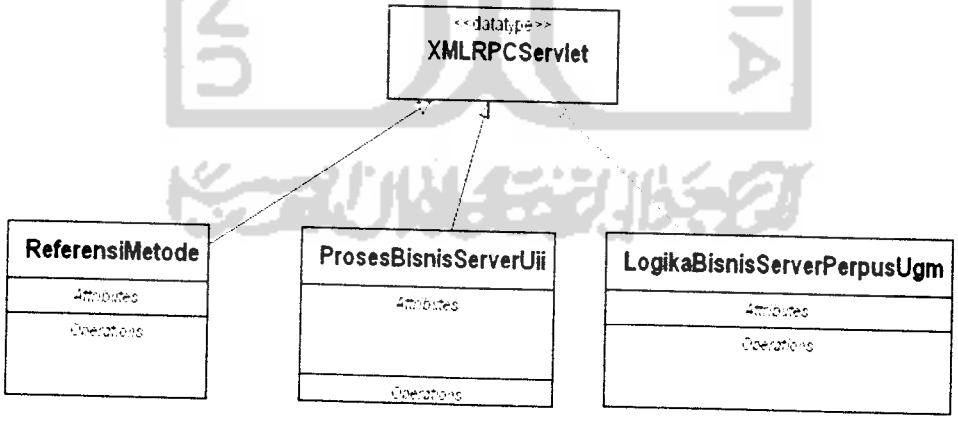


Gambar 3.14. Diagram *Class* Keterikatan *Class-Class* Ubah *URL Gateway*

e. *Class Diagram* Proses 5

Class diagram pada proses 4 menunjukan hubungan *class-class* berkaitan dengan generalisasi *class-class* logika bisnis *server* UII, UGM dan utama (*gateway*). Generalisasi adalah kebalikan dari *extends*.

Diagram hubungan *class-class* yang terlibat seperti pada gambar 3.15.



Gambar 3.15. Diagram *Class* Generalisasi Logika Bisnis UII, UGM dan Utama

3.2.3.3 *Sequence Diagram*

Sequence diagram yang menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari objek-objek yang ada pada aplikasi *sistem mobile OPAC*, yaitu :

1. *Sequence Diagram* Proses 1

Sequence diagram pada proses 1 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UII berdasarkan judul dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada *sequence diagram* proses 1 yaitu aktor *(user)*, *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *ProsesBisnisServerUii* dan *AksesDatabaseUii*.

Pada *sequence diagram* proses 1, parameter yang dikirimkan adalah judul.

Gambar 3.16 menunjukan *sequence diagram* penelusuran sederhana perpustakaan UII berdasarkan judul dan melihat hasil penelusurannya.

Gambar 3.16. *Sequence Diagram* Penelusuran Sederhana Judul UII

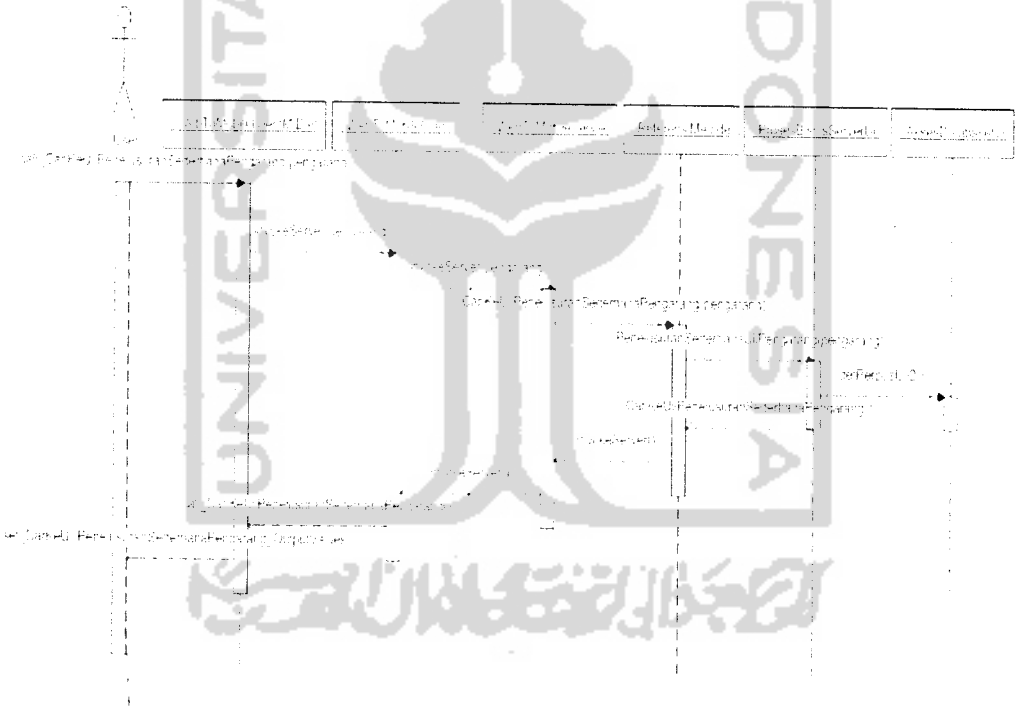
2. *Sequence Diagram* Proses *2*

Sequence diagram pada proses 1 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UII berdasarkan pengarang dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada sequence diagram proses 2 yaitu aktor *(user)*, *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *ProsesBisnisServerUii* dan *AksesDatabaseUii*.

Pada *sequence diagram* proses 2, parameter yang dikirimkan adalah pengarang.

Gambar 3.17 menunjukan *sequence diagram* penelusuran sederhana perpustakaan UII berdasarkan pengarang dan melihat hasil penelusurannya.



Gambar 3.17. *Sequence Diagram* Penelusuran Sederhana Pengarang UII

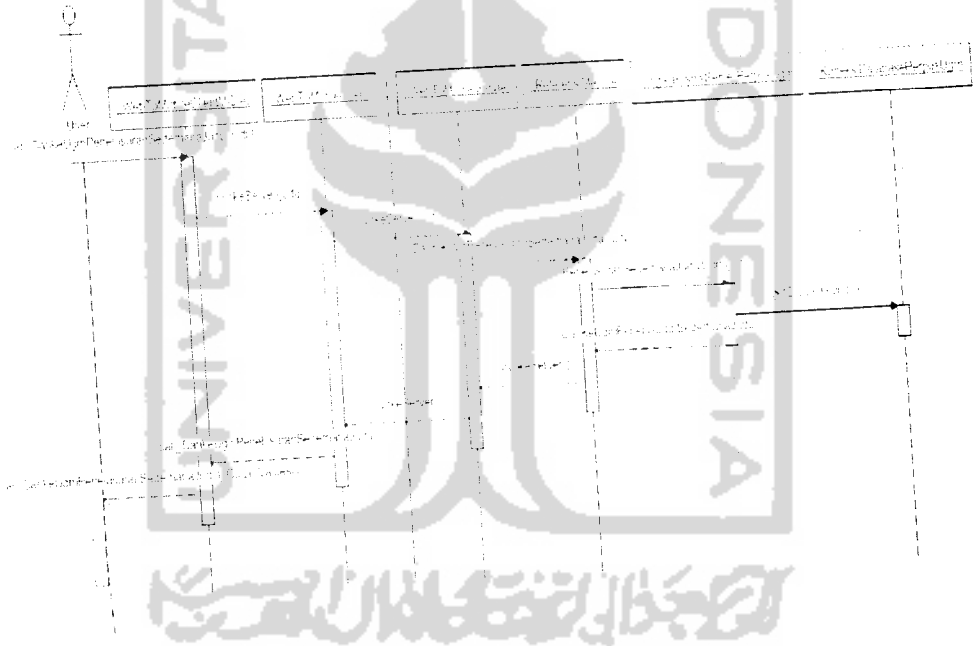
3. *Sequence Diagram* Proses 3

Sequence diagram pada proses 3 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UGM berdasarkan judul dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada *sequence diagram* proses 3 yaitu aktor *(user)*, *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm*.

Pada *sequence diagram* proses 3, parameter yang dikirimkan adalah judul.

Gambar 3.18 menunjukan *sequence diagram* penelusuran sederhana perpustakaan UGM berdasarkan judul dan melihat hasil penelusurannya.



Gambar 3.18. *Sequence Diagram* Penelusuran Sederhana Judul UGM

4. *Sequence Diagram* Proses 4

Sequence diagram pada proses 4 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UGM berdasarkan pengarang dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada sequence diagram proses 4 yaitu aktor *(user)*, *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm*.

Pada *sequence diagram* proses 4, parameter yang dikirimkan adalah pengarang.

Gambar 3.19 menunjukan *sequence diagram* penelusuran sederhana perpustakaan UGM berdasarkan pengarang dan melihat hasil penelusurannya.

Gambar 3.19. *Sequence Diagram* Penelusuran Sederhana Pengarang UGM

5. *Sequence Diagram* Proses 5

Sequence diagram pada proses 5 adalah *sequence* pada proses penelusuran *advance* pada perpustakaan UII dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada *sequence diagram* proses 5 yaitu aktor (*user*), *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *ProsesBisnisServerUii* dan *AksesDatabaseUii*.

Pada *sequence diagram* proses 5, parameter yang dikirimkan adalah judul, jenis, tokoh, dan nomor *DDC* .

Gambar 3.20 menunjukan *sequence diagram* penelusuran *advance* perpustakaan UII dan melihat hasil penelusurannya.

Gambar 3.20. *Sequence Diagram* Penelusuran *Advance* UII

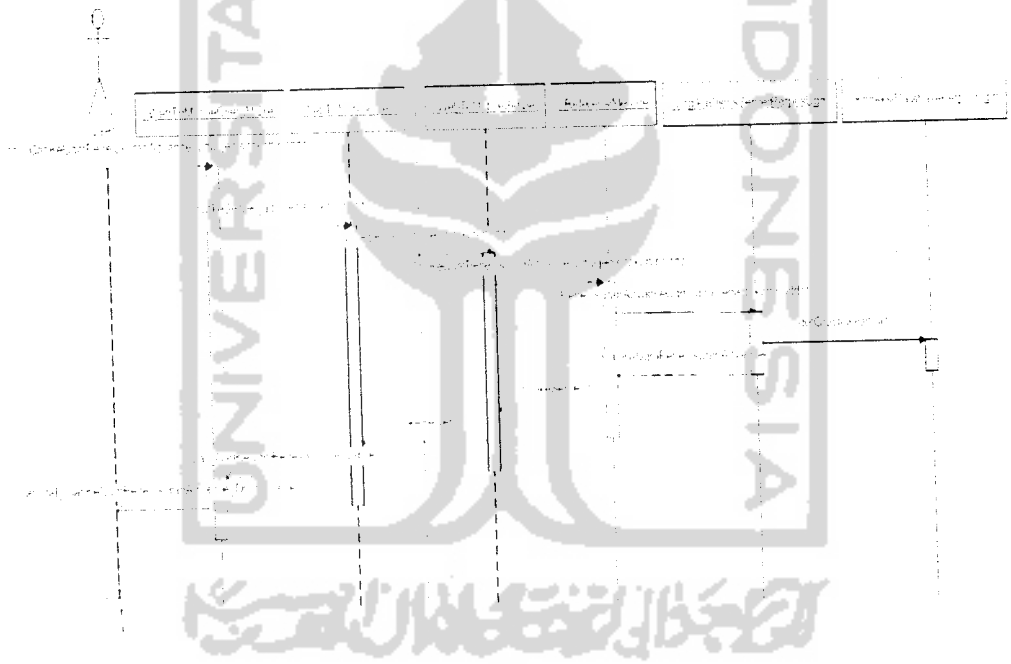
6. *Sequnce Diagram* Proses 6

Sequence diagram pada proses 6 adalah *sequence* pada proses penelusuran *advance* pada perpustakaan UGM dan melihat hasil penelusuranya.

Terdapat tujuh buah objek yang terlibat pada *sequence diagram* proses 6 yaitu aktor (*user*), *WebToMobileClientMidlet*, *WebToMobileClient*, *WebToMobileServlet*, *ReferensiMetode*, *LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm*.

Pada *sequence diagram* proses 6, parameter yang dikirimkan adalah judul, jenis, tokoh, dan nomor *DDC* .

Gambar 3.21 menunjukan *sequence diagram* penelusuran *advance* perpustakaan UGM dan melihat hasil penelusurannya.



Gambar 3.21. *Sequence Diagram* Penelusuran A*dvance* UGM

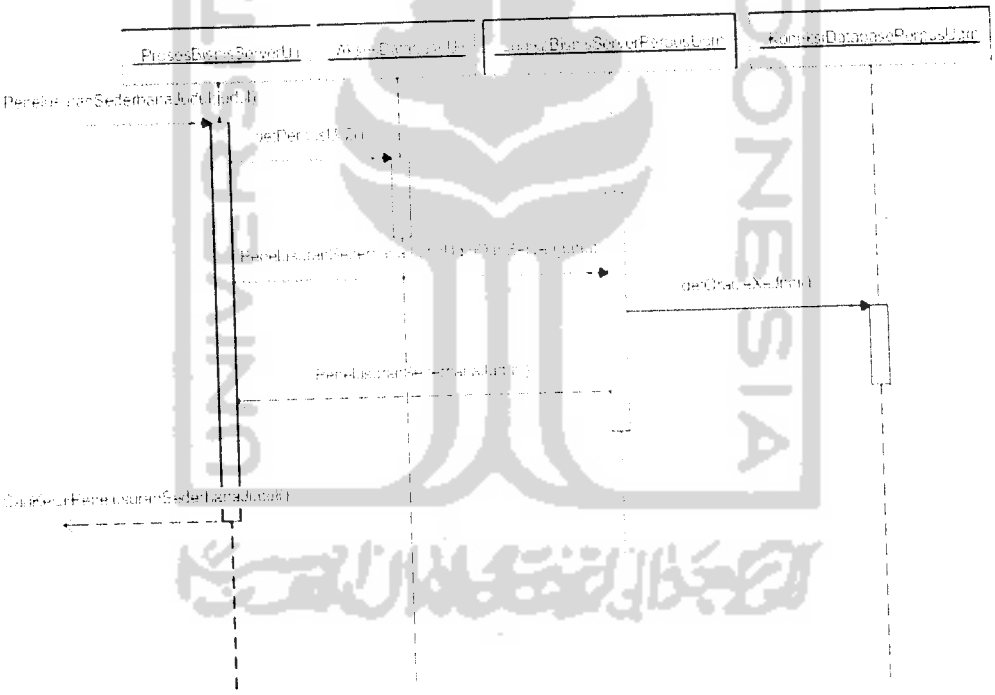
7. *Sequence Diagram* Proses 7

Sequence diagram pada proses 7 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UII berdasarkan judul pada lain server, yaitu server perpustakaan UGM, jika pada perpustakaan UII tidak diketemukan koleksi perpustakaan yang ditelusuri oleh *user*, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada *sequence diagram* proses 7 yaitu *ProsesBisnisServerUii, AksesDatabaseUii, LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm.*

Pada *sequence diagram* proses 7, parameter yang dikirimkan adalah judul.

Gambar 3.22 menunjukan *sequence diagram* penelusuran sederhana judul perpustakaan UII ke lain *server.*



Gambar 3.22. *Sequence Diagram* Penelusuran Judul UII Lain *Server*

8. *Sequence Diagram* Proses 8

Sequence diagram pada proses 8 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UII berdasarkan pengarang pada lain *server*, yaitu *server* perpustakaan UGM, jika pada perpustakaan UII tidak diketemukan koleksi perpustakaan yang ditelusuri oleh *user*, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada *sequence diagram* proses 8 yaitu *ProsesBisnisServerUii, AksesDatabaseUii, LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm.*

Pada *sequence* diagram proses 8, parameter yang dikirimkan adalah pengarang.

Gambar 3.23 menunjukan *sequence diagram* penelusuran sederhana pengarang perpustakaan UII ke lain *server.*

Gambar 3.23. *Sequence Diagram* Penelusuran Pengarang UII Lain *Server*

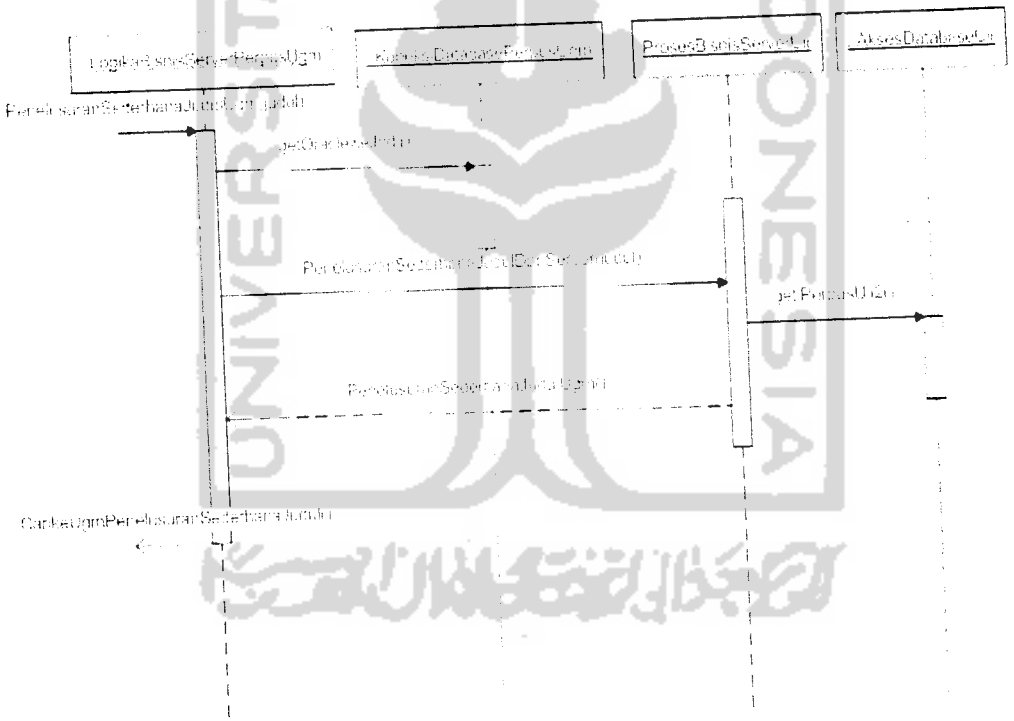
9. *Sequence Diagram* Proses 9

Sequence diagram pada proses 9 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UGM berdasarkan judul pada lain *server*, yaitu *server* perpustakaan UII, jika pada perpustakaan UGM tidak diketemukan koleksi perpustakaan yang ditelusuri oleh *user*, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada *sequence* diagram proses 9 yaitu *LogikaBisnisServerPerpusUgm*, *KoneksiDatabasePerpusUgm*, *ProsesBisnisServerUii*, dan *AksesDatabaseUii*.

Pada sequence diagram proses 9, parameter yang dikirimkan adalah judul.

Gambar 3.24 menunjukan *sequence diagram* penelusuran sederhana judul perpustakaan UGM ke lain *server*.



Gambar 3.24. *Sequence Diagram* Penelusuran Judul UGM Lain *Server*

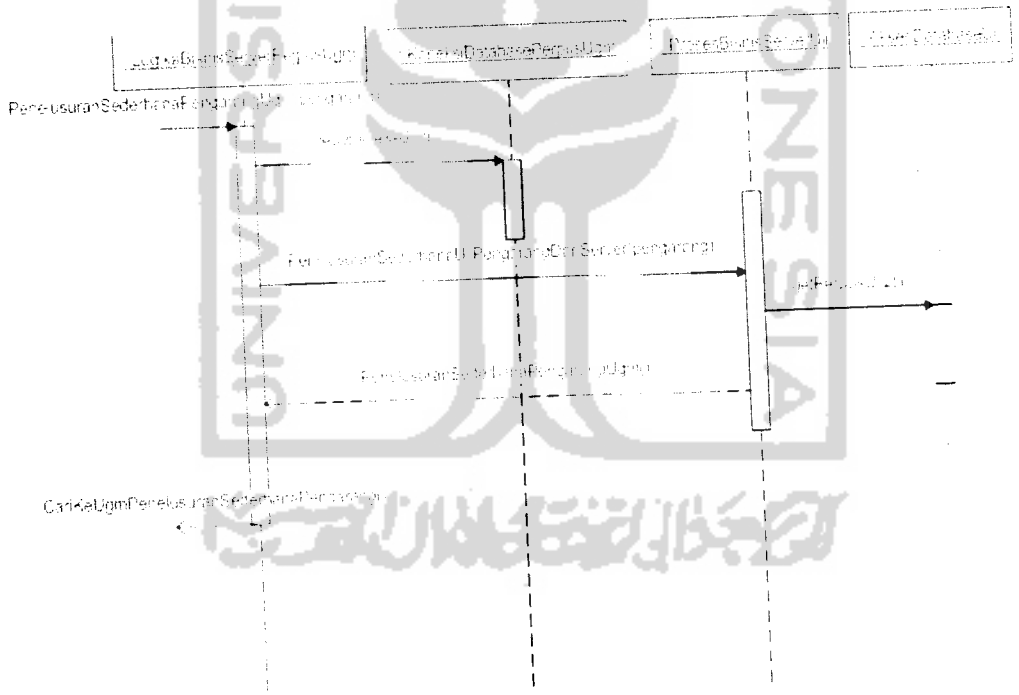
10 *Sequence Diagram* Proses 10

Sequence diagram pada proses 10 adalah *sequence* pada proses penelusuran sederhana pada perpustakaan UGM berdasarkan pengarang pada lain *server*, yaitu *server* perpustakaan UII, jika pada perpustakaan UGM tidak diketemukan koleksi perpustakaan yang ditelusuri oleh user, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada *sequence diagram* proses 10 yaitu *LogikaBisnisServerPerpusUgm, KoneksiDatabasePerpusUgm, ProsesBisnisServerUii,* dan *AksesDatabaseUii.*

Pada *sequence diagram* proses 10, parameter yang dikirimkan adalah pengarang.

Gambar 3.25 menunjukan *sequence diagram* penelusuran sederhana pengarang perpustakaan UGM ke lain *server.*



Gambar 3.25. *Sequence Diagram* Penelusuran Pengarang UGM Lain *Server*

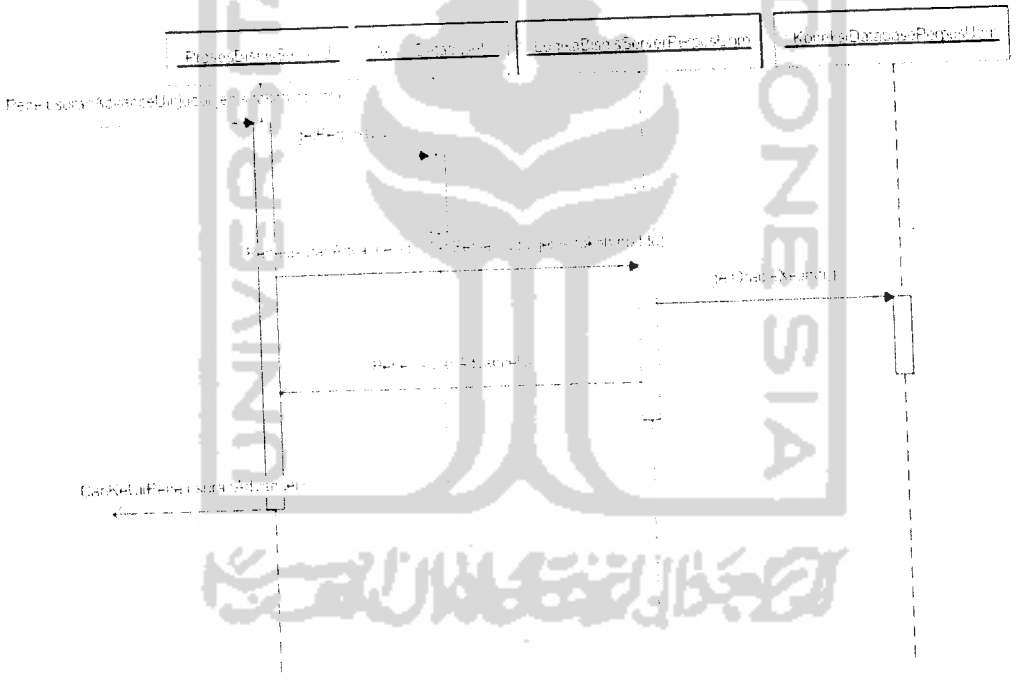
11. *Sequence Diagram* Proses 11

Sequence diagram pada proses 11 adalah *sequence* pada proses penelusuran *advance* pada perpustakaan UII pada lain *server*, yaitu *server* perpustakaan UGM, jika pada perpustakaan UII tidak diketemukan koleksi perpustakaan yang ditelusuri oleh *user*, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada sequence diagram proses 11 yaitu *ProsesBisnisServerUii, AksesDatabaseUii, LogikaBisnisServerPerpusUgm* dan *KoneksiDatabasePerpusUgm*.

Pada *sequence diagram* proses 11, parameter yang dikirimkan adalah judul, jenis, tokoh, dan nomor DDC.

Gambar 3.26 menunjukan *sequence diagram* penelusuran *advance* perpustakaan UII ke lain *server*.



Gambar 3.26. *Sequence Diagram* Penelusuran *Advance* UII ke Lain *Server*.

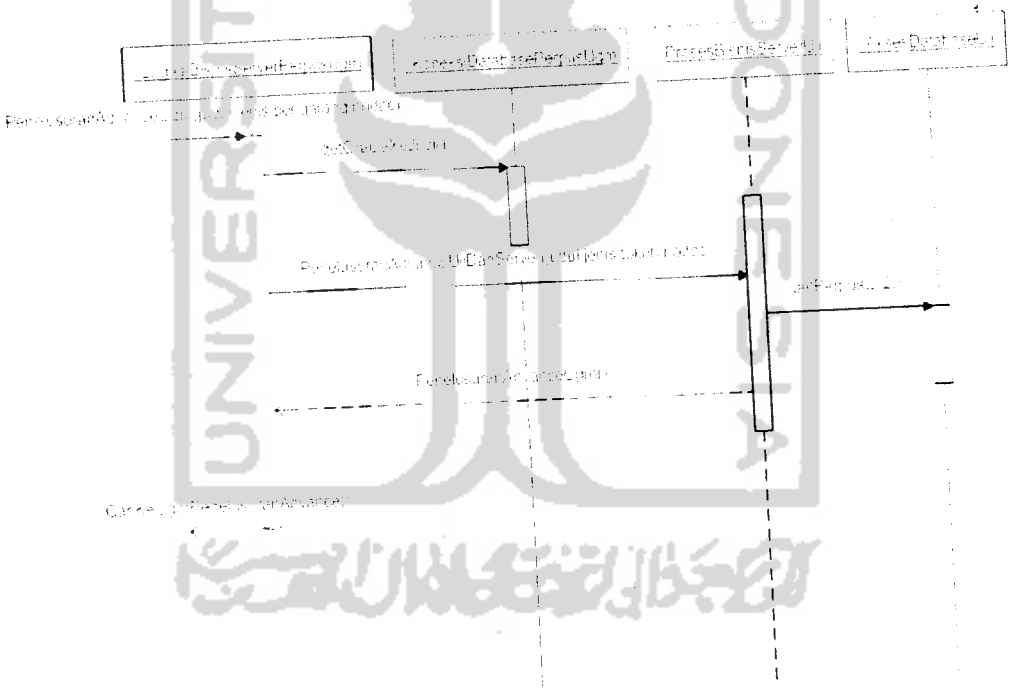
12. *Sequence Diagram* Proses

Sequence diagram pada proses 12 adalah *sequence* pada proses penelusuran *advance* pada perpustakaan UGM pada lain *server*, yaitu *server* perpustakaan UII, jika pada perpustakaan UGM tidak diketemukan koleksi perpustakaan yang ditelusuri oleh *user*, dan untuk proses lihat hasil penelusuranya.

Terdapat empat buah objek yang terlibat pada sequence diagram proses 12 yaitu *LogikaBisnisServerPerpusUgm, KoneksiDatabasePerpusUgm, ProsesBisnisServerUii*, dan *AksesDatabaseUii*.

Pada *sequence diagram* proses 12, parameter yang dikirimkan adalah judul, jenis, tokoh, dan nomor *DDC*.

Gambar 3.27 menunjukan *sequence diagram* penelusuran *advance* perpustakaan UGM ke lain *server*.



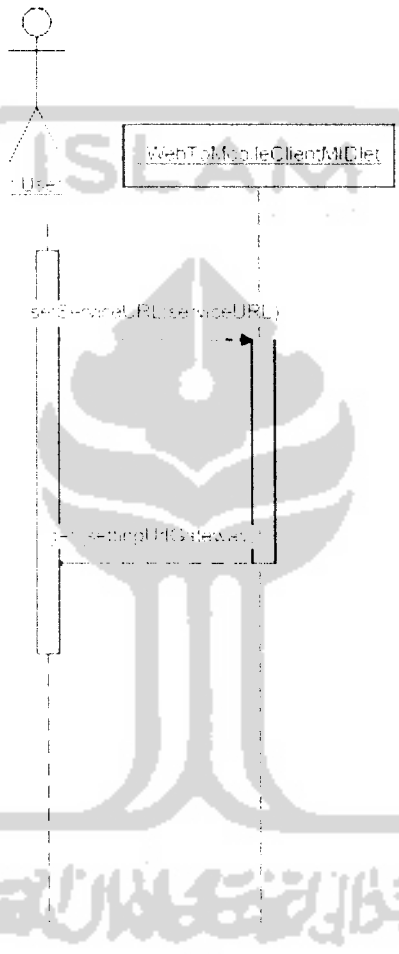
Gambar 3.27. *Sequence Diagram* Penelusuran *Advance* UGM ke Lain *Server*.

13. *Sequence Diagram* Proses 13

Sequence diagram pada proses 13 adalah *sequence* pada proses pengubahan *URL server gateway.*

Terdapat dua buah objek yang telibat, yaitu *user* dan *WebToMobileClientMidlet.*

Gambar 3.28 menunjukan *sequence diagram* pengubahan *URL server gateway.*



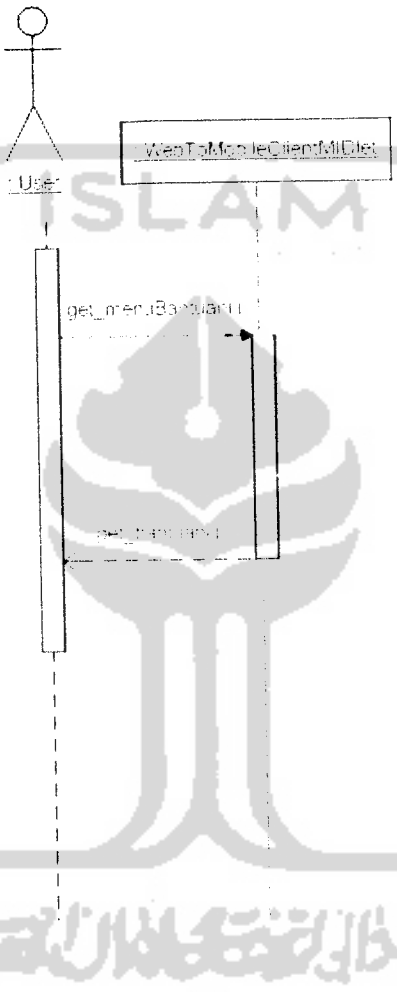
Gambar 3.28. *Sequence Diagram* Pengubahan *URL Server Gateway*

14. *Sequence Diagram* Proses 14

Sequence diagram pada proses 14 adalah *sequence* pada proses pengubahan *URL server gateway.*

Terdapat dua buah objek yang telibat, yaitu *user* dan *WebToMobileClientMidlet.*

Gambar 3.29 menunjukan *sequence diagram* lihat bantuan program.

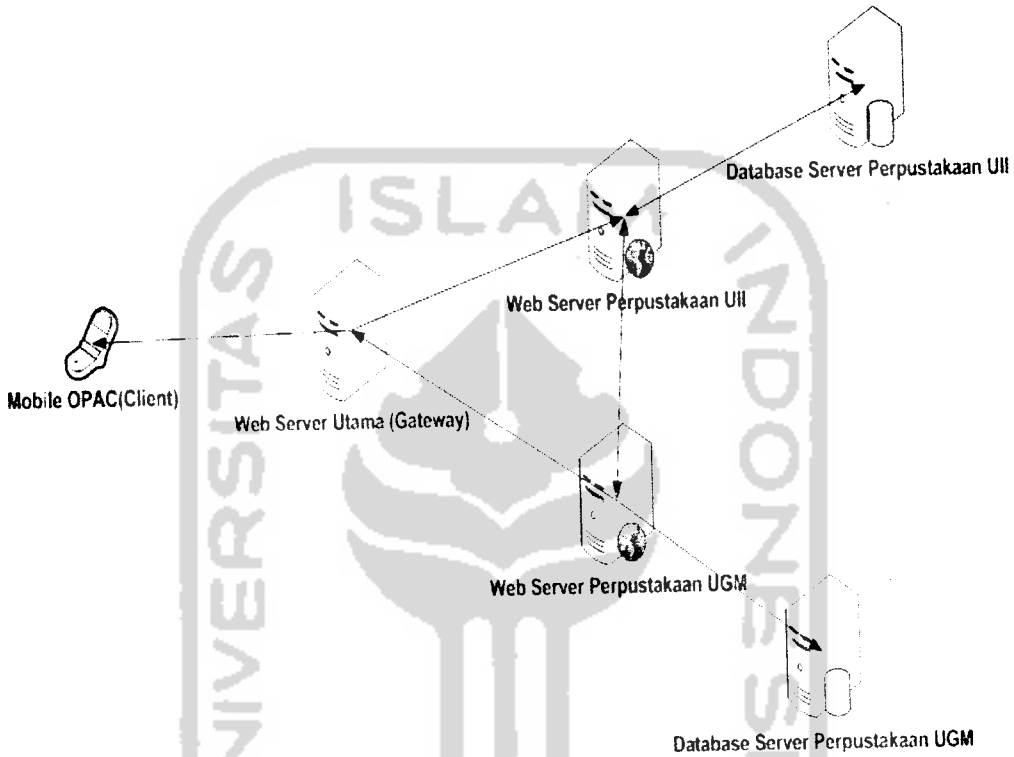


Gambar 3.29. *Sequence Diagram* Lihat Bantuan Program

3.2.3.4 Perancangan Diagram Jaringan Sistem *Mobile OPAC*

Perancangan diagram jaringan sistem *mobile OPAC* merupakan skenario sistem m*obile OPAC* yang ditinjau dari hubungan antar objek jaringan.

Gambar 3.30 Menunjukan Diagram Jaringan Sistem *Mobile OPAC* atau Skenario Sistem *Mobile OPAC*.



Gambar 3.30. Diagram Jaringan Sistem *Mobile OPAC*

3. Perancangan Struktur Direktori *File* Perpustakaan UII

Dirancang terdapat *package* yang berisi *class-class* sebagai berikut :

a. *Package org.apache.xmlrpc.webserver*

Berfungsi untuk meletakan file *properties* dari *XML-RPC*. Berguna untuk mendefinisikan layanan yang disediakan oleh *XML-RPC* untuk dilakukan *invoke* dari lain *server*.

b. *Package serveruii*

Berisi dua buah *class* yaitu :

a. *class prosesbisnisserveruii*, yaitu *class* untuk memproses permintaan *client* atau melakukan *query* permintaan penelusuran sederhana atau *advance* dari *client*.
b. *class AksesDatabaseUii2*, yaitu *class* yang berfungsi sebagai penghubung kepada *database server* perpustakaan UII.

c. *Package Libraries*

Berfungsi sebagai *package libraies* yang digunakan oleh *server* perpustakaan UII.

4. Perancangan Struktur Direktori *File* Perpustakaan UGM

Dirancang terdapat *package* yang berisi *class-class* sebagai berikut :

a. *Package org.apache.xmlrpc.webserver*

Untuk meletakan file *properties* dari *XML-RPC*. Berguna untuk mendefinisikan layanan yang disediakan oleh *XML-RPC* untuk dilakukan *invoke* dari lain *server*.

b. *Package serverperpusugm*

Berisi dua buah *class* yaitu :

a. *Class LogikaBisnisServerPerpusUgm*, yaitu *class* untuk memproses permintaan *client* atau melakukan *query* permintaan penelusuran sederhana atau *advance* dari *client*.
b. *Class koneksiDatabasePerpusUgm*, yaitu *class* yang berfungsi sebagai penghubung kepada *database server* perpustakaan UGM.

c. *Package Libraries*

Berfungsi sebagai *package libraies* yang digunakan oleh *server* perpustakaan UII.

3.2.3.6 Perancangan Antarmuka (*Interface*)

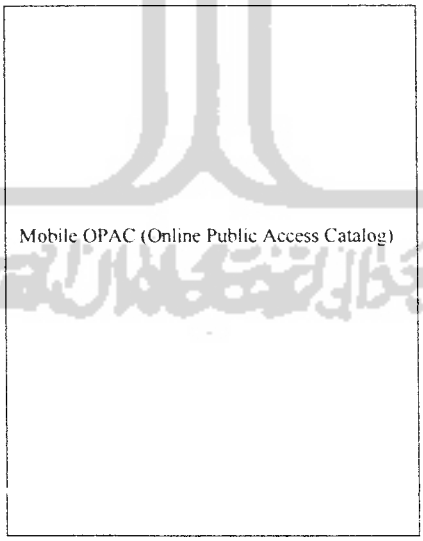
Perancangan antarmuka pada sistem *mobile OPAC* dibagi menjadi dua yaitu, perancangan pada sisi *client* dan *server*. Perancangan antarmuka sangatlah diperlukan karena antarmuka merupakan level presentasi (pada lapisan presentasi) dalam konsep yang digunakan oleh sistem *mobile OPAC* yaitu sistem terdistribusi. Pada lapisan proses bisnis juga mempunyai bebarapa antarmuka, tetapi hanya untuk menunjukan status dari *server-server* perpustakaan, apakah aktiv atau tidak. Berikut perancangan pada sistem *mobile OPAC*.

1. Perancangan Antarmuka pada Sisi *Client (MIDlet)*

Beberapa rancangan antarmuka (*form*) pada *MIDlet* dapat dijelaskan lebih rinci sebagai berikut :

a. Perancangan Antarmuka *Splash Screen*

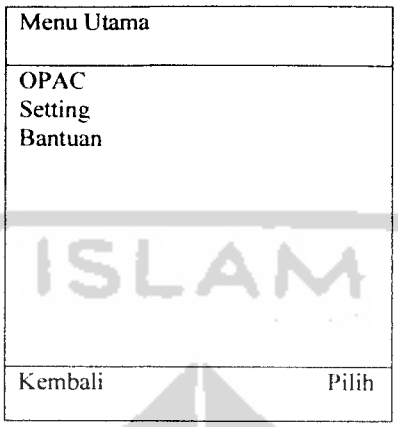
Splash screen dirancang untuk tampilan pertama kali pada saat *MIDlet mobile OPAC* dijalankan oleh *user*, berfungsi untuk memberitahukan tentang *MIDlet* yang sedang dijalankan kepada user. Untuk lebih jelasnya dapat dilihat pada gambar 3.31.



Gambar 3.31. Rancangan Antarmuka *Splash Screen*

b. Perancangan Antarmuka Menu Utama

Menu utama dirancang dengan tujuan, dapat dilakukanya pemilihan fasilitas yang disediakan *MIDlet Mobile OPAC* oleh *user*. *User* hanya memilih pilihan yang telah disediakan. Untuk lebih jelasnya dapat dilihat pada gambar 3.32.

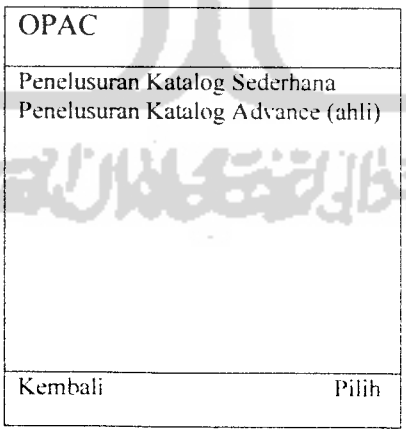


Gambar 3.32. Rancangan Antarmuka Menu Utama

c. Perancangan Menu *OPAC* (Pemilihan Jenis Penelusuran)

OPAC merupakan fasilitas utama yang disediakan oleh *MIDlet Mobile OPAC*. Dalam antarmuka menu *OPAC*, disediakan pemilihan jenis penelusuran kepada *user*.

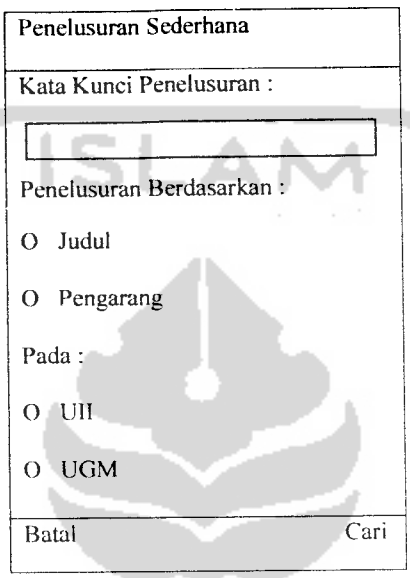
Untuk lebih jelasnya dapat dilihat pada gambar 3.33.



Gambar 3.33. Rancangan Antarmuka Menu *OPAC*

d. Perancangan Antarmuka Penelusuran Sederhana

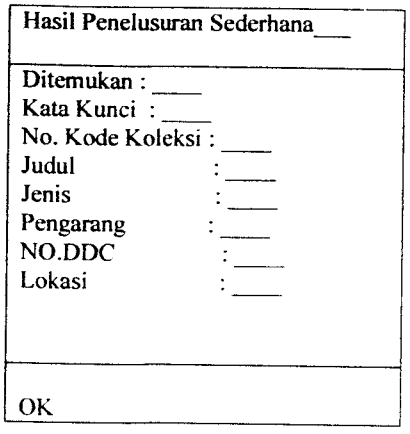
Penelusuran sederhana merupakan penelusuran yang disediakan kepada *user* dengan disediakanya kriteria penelusuran yang terbatas yaitu judul dan pengarang, oleh karena itu, dirancang antarmuka yang menyediakan pilihan kriteria penelusuran tersebut dan pemilihan universitas yang kali pertama akan dilakukan penelusuran. Untuk lebih jelasnya dapat dilihat pada gambar 3.34.



Gambar 3.34. Rancangan Antarmuka Penelusuran Sederhana

e. Perancangan Antarmuka Hasil Penelusuran Sederhana

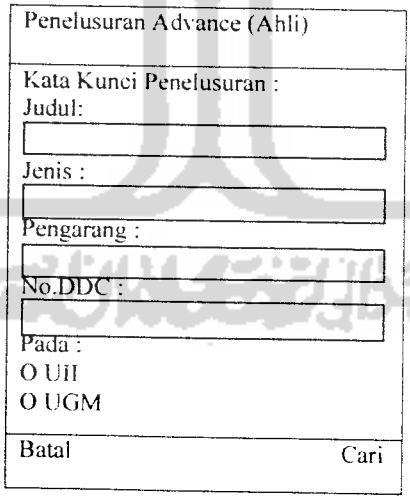
Rancangan antarmuka hasil yang ditampilkan setelah dilakukanya penelusuran sederhana dapat lebih dijelaskan lagi pada gambar 3.35.



Gambar 3.35. Racangan Antarmuka Hasil Penelusuran Sederhana

f. Perancangan Antarmuka Penelusuran *Advance* (Ahli)

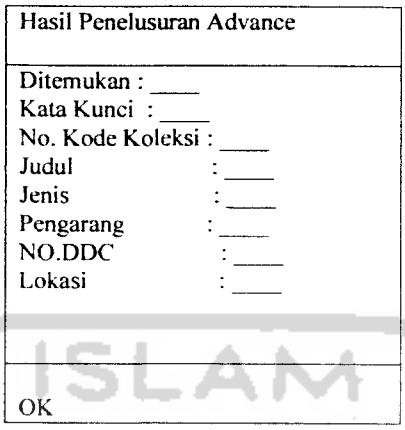
Rancangan antarrruka penelusuran *advance* (ahli), dirancang lebih banyak memuat informasi yang akan disajikan kepada *user*, karena, penelusuran *advance* memerlukan kriteria penelusuran yang lebih banyak daripada penelusuran sederhana seperti judul, jenis, pengarang dan nomor *DDC*. Berikut dengan pilihan perpustakaan universitas mana akan dilakukan penelusuran untuk kali pertamanya.Untuk lehih jelasnya dapat dilihat pada gambar 3.36.



Gambar 3.36. Rancangan Antarmuka Penelsuran *Advance* (Ahli)

g. Perancangan Antarmuka Hasil Penelusuran *Advance* (Ahli)

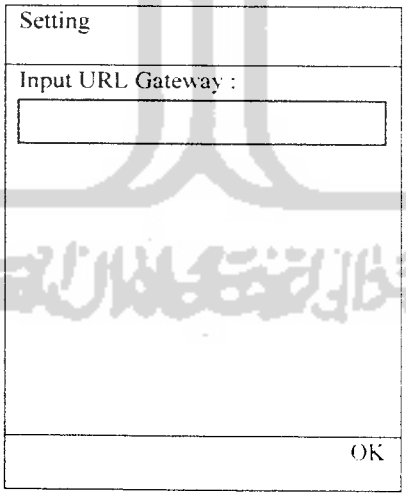
Rancangan antarmuka penelusuran *advance* dapat dilihat lebih jelas pada gambar 3.37.



Gambar 3.37. Racangan Antarmuka Hasil Penelusuran *Advance*

h. Perancangan Antarmuka Pengubahan *URL Sever Gateway*

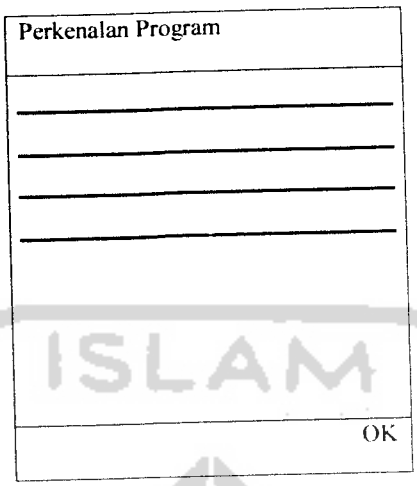
Racangan antarmuka pengubahan *URL sever gateway* diperlukan karena, jika suatu saat *URL server gateway* berubah atau alamat *server* berganti, maka *user* dapat melakukan pengubahan *URL server gateway* dengan URL yang *valid*. Untuk lebih jelasnya dapat dilihat pada gambar 3.38.



Gambar 3.38. Rancangan Antarmuka Ubah *URL Server Gateway*

i. Perancangan Antarmuka Perkenalan Program

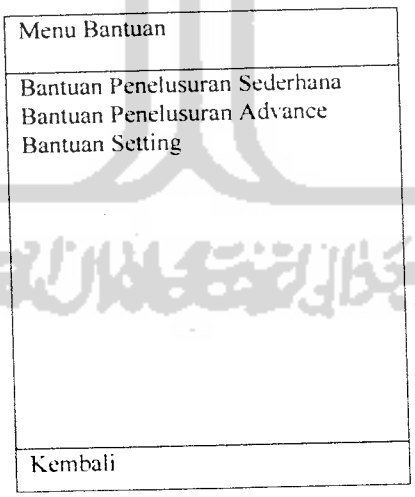
Rancangan antarmuka perkenalan *MIDlet mobile OPAC* dapat dilihat lebih jelas pada gambar 3.39.



Gambar 3.39. Rancangan Antarmuka Perkenalan Program

j. Perancangan Antarmuka Menu Bantuan

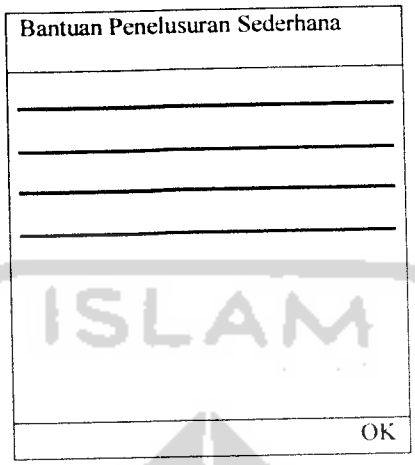
Rancangan antarmuka menu bantuan dapat dilihat lebih jelas pada gambar 3.40. Menu bantuan disini, *user* dapat memilih bantuan-bantuan *MIDlet* yang telah disediakan.



Gambar 3.40. Rancangan Antarmuka Menu Bantuan

k. Perancangan Antarmuka Bantuan Penelusuran Sederhana

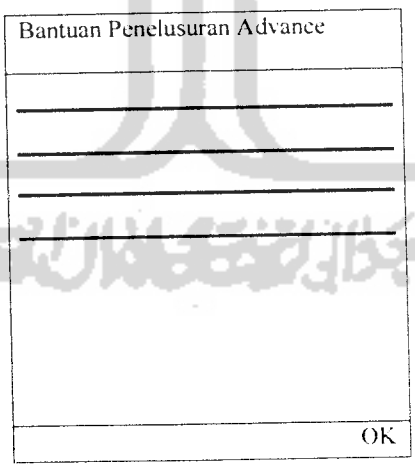
Rancangan antarmuka bantuan penelusuran sederhana *MIDlet mobile OPAC* dapat dilihat lebih jelas pada gambar 3.41.



Gamabar 3.41. Rancangan Antarmuka Bantuan Penelusuran Sederhana

l. Perancangan Antarmuka Bantuan Penelusuran *Advance* (Ahli)

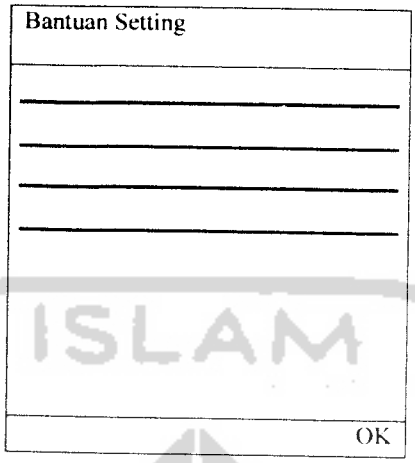
Rancangan antarmuka bantuan penelusuran *advance MIDlet mobile OPAC* dapat dilihat lebih jelas pada gambar 3.42.



Gambar 3.42. Rancangan Antarmuka Penelusuran *Advance*

m. Perancangan Antarmuka Bantuan *Setting*

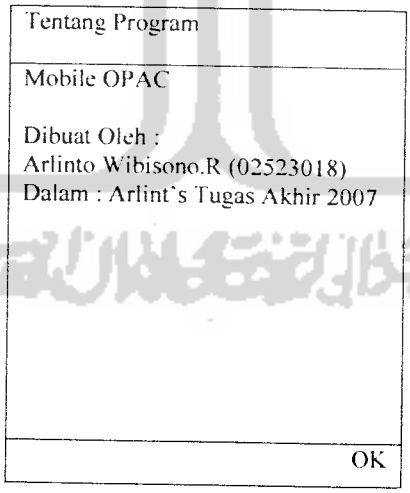
Rancangan antarmuka bantuan *setting MIDlet mobile OPAC* dapat dilihat lebih jelas pada gambar 3.43.



Gambar 3.43. Rancangan Antarmuka Bantuan *Setting*

n. Perancangan Antarmuka Tentang Program

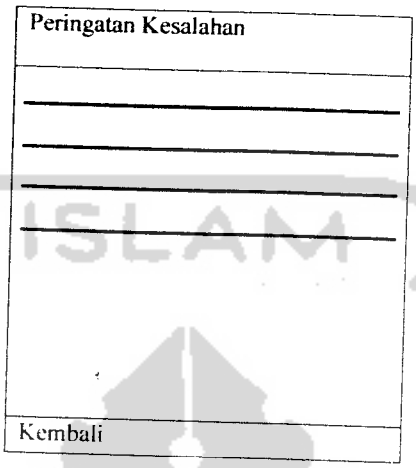
Perancangan antarmuka tentang dirancang untuk ditampilkanya informasi tentang dan oleh siapa *MIDlet mobile OPAC* dibangun.Untuk lebih jelasnya dapat dilihat pada gambar 3.44.



Gambar 3.44. Rancangan Antarmuka Tentang Program

o. Perancangan Antarmuka Peringatan Kesalahan *Server*

Perancangan antarmuka peringatan kesalahan *server* hanya dilakukan dengan merancanng suatu informasi yang dapat ditampilkan kepada *user*, dikarenakan *MIDlet mobile OPAC* gagal untuk melakukan hubungan dengan *server*. Untuk lebih jelasnya dapat dilihat pada gambar 3.45.

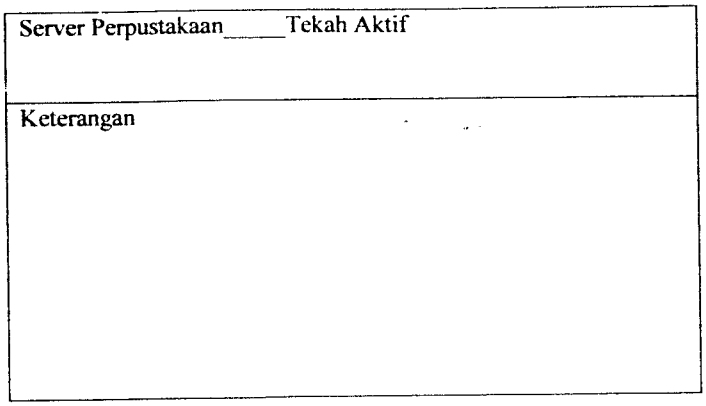


Gambar 3.45. Rancangan Antarmuka Peringatan Kesalahan

2. Perancangan Antarmuka di Sisi *Server*

a. Perancangan Antarmuka Pemberitahuan Bahwa *Server* Perpustakaan UII, UGM dan Gateway Telah Aktiv

Gambar 3.46 menunjukan pemberitahuan status *server* perpustakaan bahwa *server* perpustakaan (UII, UGM dan *gateway*) telah aktiv atau telah di-*deploy* dan dijalankan. Untuk lebih jelasnya dapat dilihat pada gambar 3.46.



Gambar 3.46. Rancangan Antarmuka Pemberitahuan *Server* Telah Aktiv

b. Perancangan Antarmuka Stub Pada Server Telah Aktiv

Gambar 3.47 menunjukan pemberitahuan status *stub server* perpustakaan, bahwa *server* gateway telah aktiv atau telah di-*deploy* dan dijalankan. Untuk lebih jelasnya dapat dilihat pada gambar 3.47.

Gambar 3.47. Rancangan Antarmuka Pemberitahuan *Stub Server* Aktiv

3.2.3.7 Perancangan *Database*

1. Perancangan *Database* Perpustakaan UII

a. Struktur Tabel *Database* Perpustakaan UII

Dalam *Database* Perpustakaan UII, dirancang dan digunakan tiga buah tabel dari tabel utama koleksi perpustakaan di UII yang terlampir dalam lampiran B, diantaranya :

a. Tabel *jenisuii*, merupakan tabel jenis koleksi perpustakaan di UII. Untuk lebih jelasnya dapat dilihat pada tabel 3.1.
b. Tabel *koleksiuii*, merupakan tabel koleksi perpustakaan di UII. Untuk lebih jelasnya dapat dilihat pada tabel 3.2.
c. Tabel *lokasiuii*, merupakan tabel lokasi perpustakaan di UII. Untuk lebih jelasnya dapat dilihat pada tabel 3.3.

Tabel 3.1. Struktur Tabel *jenisuii*

NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	kdJenisUII	*varchar*	20	Kode jenis (*PK*)
2	jenisUii	*varchar*	50	Jenis koleksi UII

Tabel 3.2. Struktur Tabel *koleksiuii*

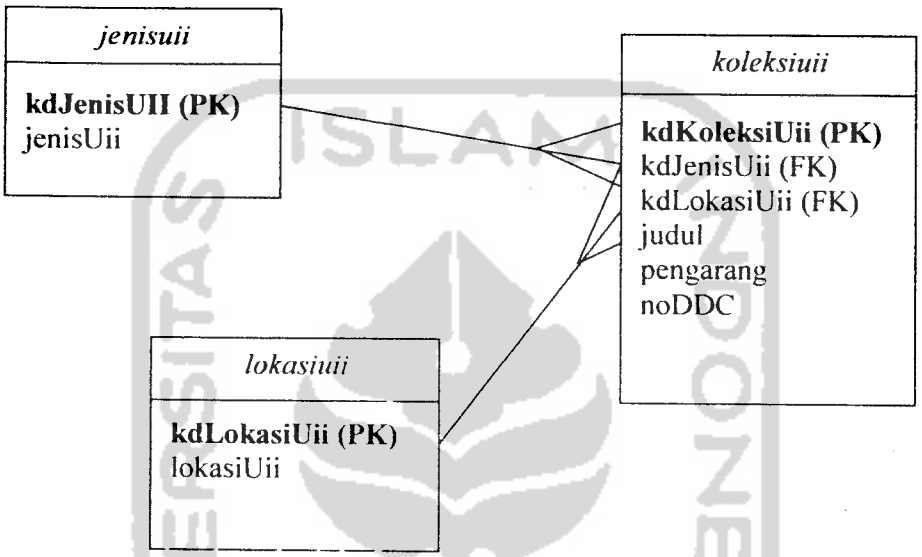
NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	kdKoleksiUii	*varchar*	15	Kode koleksi UII (PK)
2	kdJenisUii	*varchar*	15	Kode jenis (FK)
3	kdLokasiUii	*varchar*	15	Kode lokasi (FK)
4	Judul	*varchar*	100	Judul koleksi UII
5	Pengarang	*varchar*	50	Nama pengarang
6	noDDCUii	*varchar*	10	Nomor *DDC*

Tabel 3.3. Struktur Tabel *lokasiuii*

NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	kdLokasiUii	*varchar*	20	Kode lokasi (*PK*)
2	lokasiUii	*varchar*	100	Lokasi perpus UII

b. Relasi Antar Tabel pada Database Perpustakaan UII

Diagram relasi antar tabel dapat dilihat lebih jelas pada gambar 3.48.



Gambar 3.48. Relasi Tabel *Database* Perpustakaan UII

2. Perancangan *Database* Perpustakaan UGM

a. Struktur Tabel *Database* Perpustakaan UGM

Dalam *Database* Perpustakaan UGM, dirancang dan digunakan tiga buah tabel dari tabel utama koleksi perpustakaan di UGM yang terlampir dalam lampiran B, diantaranya :

a. Tabel *LOKASI*, merupakan tabel lokasi perpustakaan UGM. Untuk lebih jelasnya dapat dilihat pada tabel 3.4.
b. Tabel *KOLEKSI*, merupakan tabel koleksi perpustakaan UGM. Untuk lebih jelasnya dapat dilihat pada tabel 3.5.
c. Tabel *JENIS*, merupakan tabel jenis koleksi perpustakaan UGM. Untuk lebih jelasnya dapat dilihat pada tabel 3.6.

Tabel 3.4. Struktur Tabel *LOKASI*

NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	ID_LOKASI	*varchar*	15	Kode lokasi (*PK*)
2	LOKASI	*varchar*	100	Lokasi perpus UGM

Tabel 3.5. Struktur Tabel *KOLEKSI*

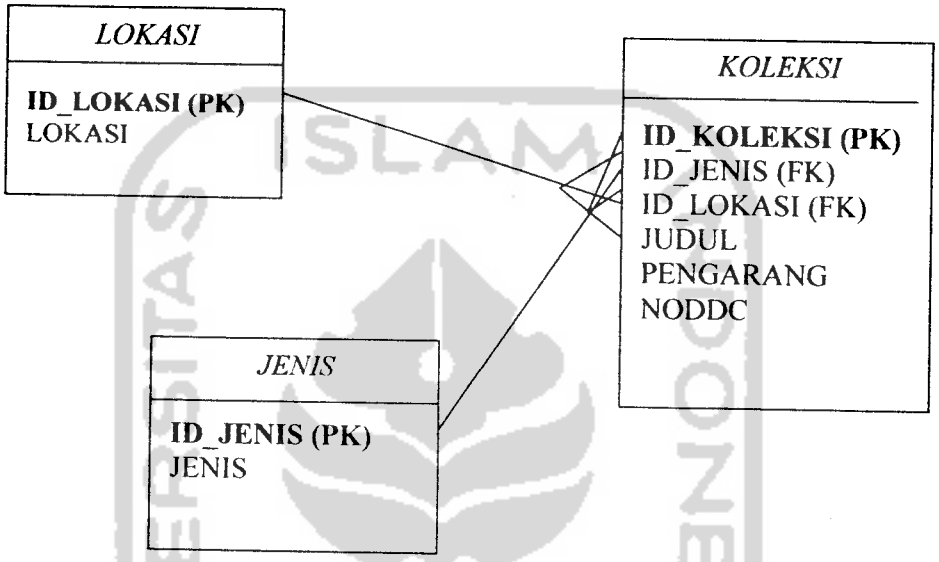
NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	ID_KOLEKSI	*varchar*	15	Kode koleksi UGM (PK)
2	ID_JENIS	*varchar*	35	Kode jenis (FK)
3	ID_LOKASI	*varchar*	30	Kode lokasi (FK)
4	JUDUL	*varchar*	100	Judul koleksi UGM
5	PENGARANG	*varchar*	100	Nama pengarang
6	NODDC	*varchar*	30	Nomor *DDC*

Tabel 3.6. Struktur Tabel *JENIS*

NO	Nama *Field*	*Type*	Ukuran	Keterangan
1	ID_JENIS	*varchar*	15	Kode jenis (*PK*)
2	JENIS	*varchar*	40	Jenis koleksi perpus UGM

b. Relasi Antar Tabel pada *Database* Perpustakaan UGM

Diagram relasi antar tabel dapat dilihat lebih jelas pada gambar 3.49.



Gambar 3.49 Relasi Tabel *Database* Perpustakaan UGM

BAB IV

HASIL dan PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi Sistem *Mobile OPAC* Secara Umum

Implementasi sistem adalah tahap dimana sistem mampu diterapkan kedalam keadaan yang sesungguhnya, setelah perancangan yang sudah dilakukan sebelumnya. Dari implementasi, akan diketahui keadaan sistem yang telah dibuat, apakah dapat berjalan dengan baik atau tidak, dan menghasilkan keluaran sistem yang sesuai dengan perancangan yang telah dibuat.

Implementasi sistem *mobile OPAC* hanya dilakukan pada *emulator MIDlet* saja karena aplikasi pada server tidak di*host*ingkan pada *webserver*, hanya berjalan pada *localhost* saja, karena sangat jarang ditemukan perusahaan *web hosting* yang memberikan layanan *hosting* sesuai dengan kebutuhan penelitian.

4.1.2 Implementasi Struktur Sistem *Mobile OPAC*

Sistem *mobile OPAC* adalah suatu kesatuan sistem terintegrasi antara *client* dan *server*. Konsep kerjanya adalah sistem terdistribusi. Pada perancangan, telah disebutkan bahwa sistem terdiri atas *client* yang meminta penelusuran katalog perpustakaan kepada *server*. Secara umum, implementasi struktur sistem *mobile OPAC* dibagi menjadi dua, yaitu implementasi struktur *mobile OPAC MIDlet* dan aplikasi *server* pada *web server*.

4..1.2.1 Implementasi Struktur *Mobile OPAC MIDlet (Client)*

Implementasi dari perancangan *MIDlet Mobile OPAC* adalah diimplementasikanya rancangan *class diagram* ke dalam bahasa pemrograman *Java*. Terdapat dua buah *class* dan satu buah *file* konfigurasi *XML* yang

menerangkan properti-peroperti yang dimiliki *client* dan *server stub*. Semuanya terdapat dalam *package clientmidlet*.

Implementasinya sebagai berikut :

a. *Class WebToMobileClientMIDlet* sebagai *class* utama yaitu *class MIDlet Mobile OPAC* dan,

b. *Class WebToMobileClient* sebagai *stub* di sisi *client* yang menghubungi *stub* dari sisi *server* (*server gateway* atau *server* utama).

4.1.2.2 Implementasi Struktur Aplikasi *Web (Server)*

Implementasi struktur di sisi *server* (aplikasi *server*) yaitu, diimplmentasikanya tiga buah *web server* yaitu *server* utama (*server gateway*), server perpustakaan UII, UGM dan satu buah *stub*. Ketiganya menggunakan struktur direktori *Jakarta*. Struktur direktori *Jakarta* disini hanyalah standarisasi struktur *file* aplikasi *web*. Struktur tersebut adalah sebagai berikut :

a. *Web Pages* yaitu direktori dimana *folder WEB-INF* berada. Didalamnya terdapat file-file konfigurasi *XML* dari *web*.

b. *Configuration Files*, yaitu *folder* yang didalamnya terdapat file-file konfigurasi *server*, jika ada.

c. *Server Resources*, yaitu *folder* ditempatkanya resources *server*, jika ada.

d. *Souce Packages*, yaitu *folder* ditempatkanya kode-kode *servlet*. Kode *servlet* pada sisi *server* secara umum dibagi lagi menjadi beberapa *package* yaitu :

   a. *Package org.apache.xmlrpc.webserver*, terdapat dalam *server* UII dan *server* UGM, yaitu *package* dimana diletakannya konfigurasi layanan yang disediakan oleh *server*, untuk dipanggil (*invoke*) dari *server* lainya, dalam hal ini adalah *server gateway* (server utama), *server* UII dan *server* UGM.

   b. *Package* utama *server*, pada *server* perpustakaan UII dinamakan *package serveruii*, berisi proses bisnis *servlet* perpustakaan UII. Pada *server* perpustakaan UGM dinamakan *package serverperpusugm*, berisi proses bisnis perpustakaan UGM dan pada *server gateway* dinamakan *package*

c. *.referensimetodeclient*, berisi proses bisnis *servlet gateway* dan *file* pendukung lainya (*class stub* yang berada di sisi *server*, dinamakan *WebToMobileServlet*).

e. *Test Libraries*, yaitu *folder* ditempatkanya kode-kode *servlet test*, jika ada.

f. Libraires, yaitu *folder* dimana ditempatkanya *library-library* yang dibutuhkan oleh *server*. Pada *server* perpustakaan UII, *library* yang ditempatkan yaitu *Apache XML-RPC 3.0* dan *JDBC Connector MySQL*. Pada *server* perpustakaan UGM, *library* yang ditempatkan yaitu *Apache XML-RPC 3.0* dan *JDBC Connector Oracle* dan pada *server gateway* ditempatkan *libraries Apache XML-RPC 3.0*.

g. *Test Libraries*, yaitu *folder* dimana ditempatkanya *test library* dari *server*, jika ada.

4.1.3 Implemetasi Antarmuka Sistem *Mobile OPAC*

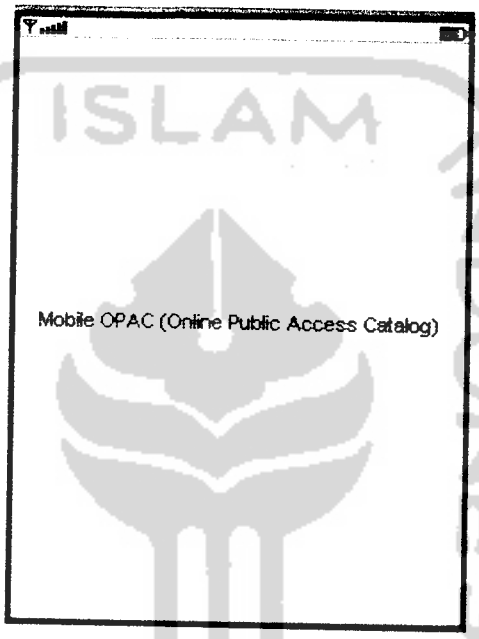
Secara umum implementasi antar muka sistem *mobile OPAC* juga dibedakan menjadi dua, yaitu implementasi antarmuka *MIDlet mobile OPAC (*sisi *client)* dan antarmuka aplikasi *web* (sisi *server*).

4.1.3.1 Impelemtasi Antarmuka *Mobile OPAC (Client)*

Pada *client*, antarmuka diimplementasikan atau dipresentasikan pada *class WebToMobileClientMIDlet*, sedangkan *class stub* yang berada di sisi *client* tidak mempunyai antarmuka atau tidak dipresentasikan. Tampilan antarmuka menggunakan *emulator platform J2ME Wireless Toolkit 2.2* yang sudah diikut sertakan dalam modul *Netbeans Mobility Pack* dengan *device DafaultColorPhone, CLDC 1.1 dan MIDP 2.0*. Tampilan pada *emulator* dapat berbeda-beda, tergantung dari jenis dan versi *emulator* yang digunakan, setiap aplikasi yang dijalankan pada emulator tidak pasti dapat berjalan pada *handheld* secara nyata, karena semua masih menyesuaikan dengan kemampuan dan kapasitas *handheld* itu sendiri.

1. Tampilan Halaman Muka

Pada halaman muka, yaitu pada saat *mobile OPAC* dijalankan pada *emulator*, akan ditampilkan halaman pertama, yaitu *splash screen* , yang menerangkan awal dari *MIDlet mobile OPAC* dan mempunyai durasi selama tiga detik untuk menampilkanya (dapat dijeda), setelah tampil, lalu menuju layar aktiv menu utama. Gambar 4.1 menunjukan tampilan halaman *splash screen mobile OPAC*.



Gambar 4.1. Tampilan *Splash Screen*

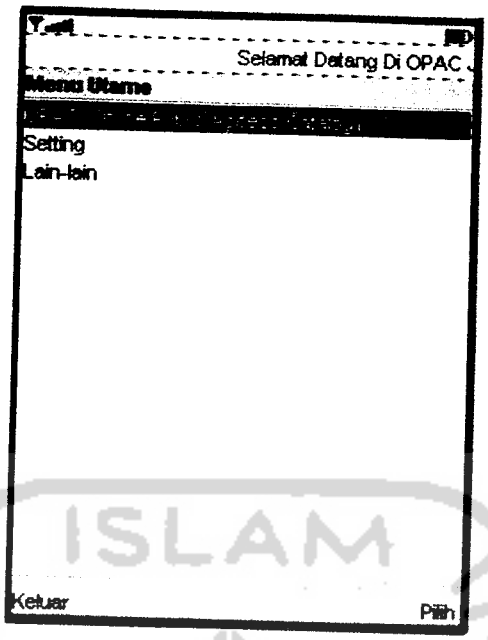
2. Tampilan Halaman Menu Utama

Menu utama adalah menu awal dan dasar yang dapat dilipih oleh *user*. Pada tampilan menu utama terdapat tiga pilihan menu, yaitu :

a. Menu *OPAC (Online Public Access Catalog)* yaitu menu untuk melakukan penelusuran katalog perpustakaan.

b. Menu *setting* yaitu menu untuk melakukan perubahan *URL server* utama atau *gateway*.

c. Menu bantuan yaitu menu untuk mengetahui petunjuk bantuan program.



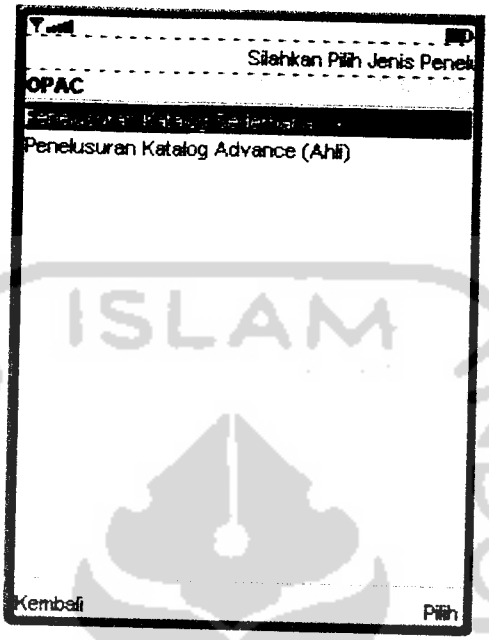
Gambar 4.2. Tampilan Menu Utama

3. Tampilan Halaman *OPAC (Online Public Acces Catalog)*

Halaman *OPAC* terdapat dalam sub menu utama *OPAC*. Halaman ini menampilkan jenis dari penelusuran katalog. Untuk lebih jelasnya dapat dilihat pada gambar 4.3. Halaman untuk *OPAC* terdiri atas dua buah halaman, yaitu :

a. Halaman penelusuran sederhana, halaman ini berfungsi untuk melakukan penelusuran sederhana.

b. Halaman penelusuran *Advance* (ahli), halaman ini berfungsi untuk melakukan penelusuran *advance* (ahli).



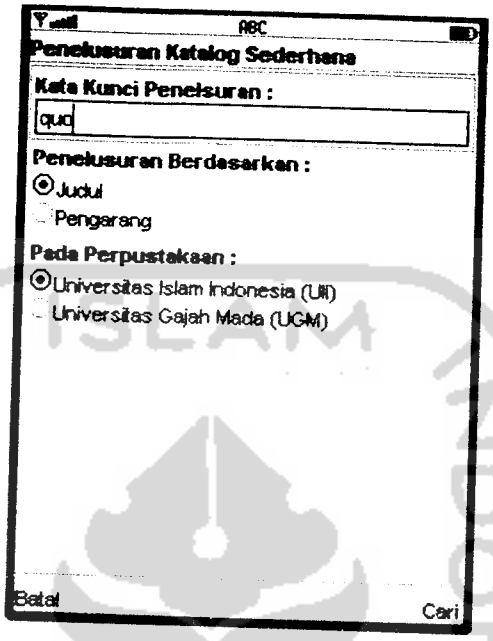
Gambar 4.3. Menu Penelusuran *OPAC*

4. Tampilan Halaman Penelusuran Sederhana

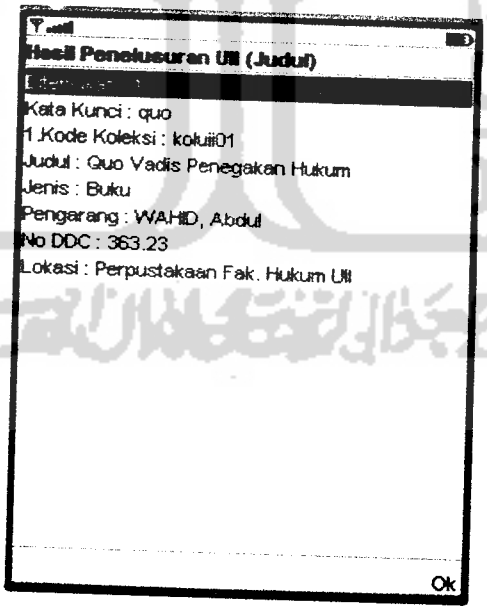
Pada halaman penelusuran sederhana, *user* dapat memasukan kata kunci penelusuran berdasarkan judul atau pengarang, dan perpustakaan universitas mana yang akan kali pertama ditelusuri. Pemilihan kriteria penelusuran dan pemilihan perpustakaan universitas dilakukan dengan memilih salah satu dari pilihan yang ada, tidak bisa kedua-duanya. Lebih jelasnya dapat dilihat pada gambar 4.4.

Setelah memasukan kata kunci pemilihan dan atribut lainya, user dapat melakukan penelusuran dengan menekan tombol perintah cari.

Lebih jelasnya dapat dilihat pada gambar 4.5.



Gambar 4.4. Tampilan Penelusuran Sederhana



Gambar 4.5. Tampilan Hasil Penelusuran Sederhana

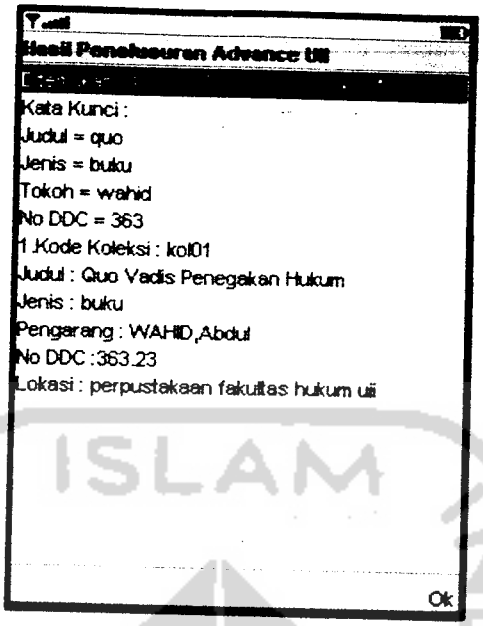
5. Tampilan Halaman Penelusuran *Advance* (Ahli)

Pada halaman penelusruan *advance, user* dapat memasukan kata kunci penelusuran berdasarkan judul (dan/atau) jenis (dan/atau) pengarang (dan/atau) nomor *DDC* dan perpustakaan universitas mana yang akan kali pertama ditelusuri.

Setelah memasukan kata kunci pemilihan dan atribut lainya, *user* dapat melakukan penelusuran dengan menekan tombol cari. Lebih jelasnya dapat dilihat pada gambar 4.6.

Gambar 4.6. Tampilan Penelusuran Advance (Ahli)

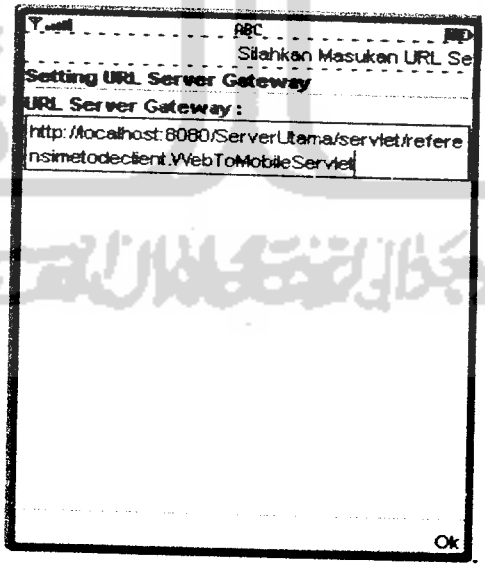
Dan hasil dari penelusuran *advance* dapat dilihat pada gambar 4.7.



Gambar 4.7. Hasil Tampilan Penelusuran *Advance* (Ahli)

6. Tampilan Halaman Setting URL Gateway

Halaman untuk setting *URL gateway* berguna untuk melakukan perubahan *URL server gateway* jika diperlukan, terdiri dari satu buah halaman , yaitu sebuah *text field* untuk memasukan *URL server gateway.*

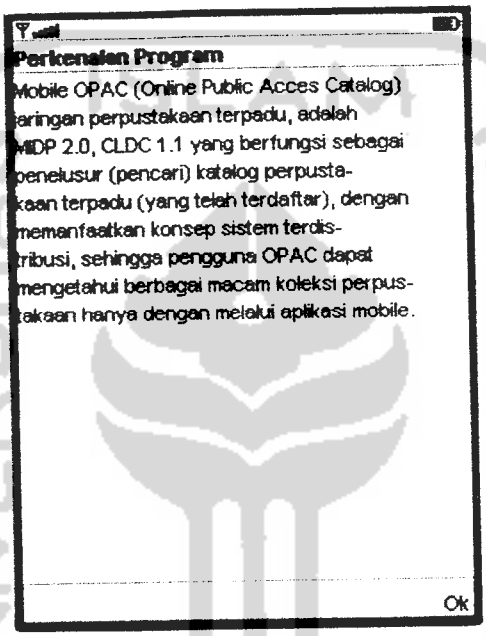


Gambar 4.8. Tampilan Halaman *Setting URL Gateway*

7. **Tampilan Halaman Bantuan**

Adalah halaman yang berkaitan dengan aplikasi *MIDlet mobile OPAC*, halaman ini menujukan apa, bagaimana dan tentang aplikasi *MIDlet mobile OPAC*. Halaman untuk bantuan terdiri dari tiga buah halaman yaitu :

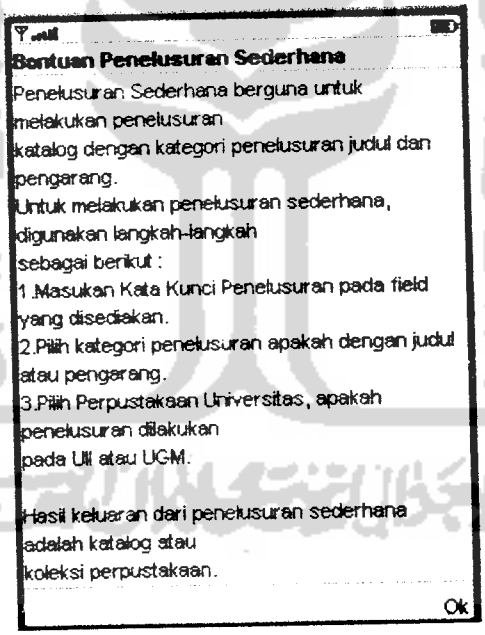
a. Halaman Perkenalan Program, halaman ini menampilkan perkenalan atau pengertian dari *MIDlet mobile OPAC*. Untuk lebih jelasnya dapat dilihat gambar 4.9.



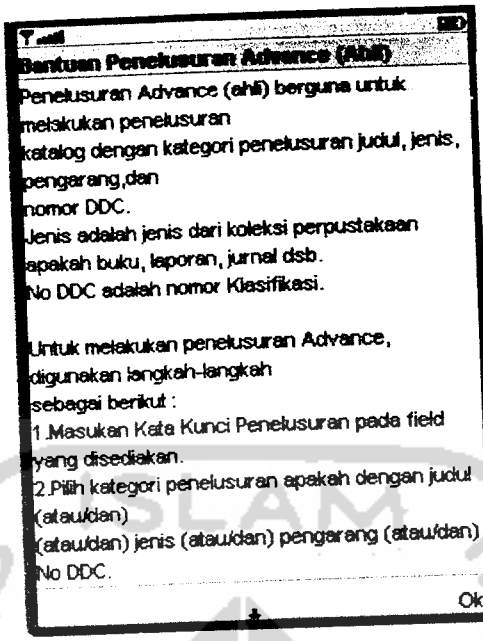
Gambar 4.9 Tampilan Halaman Perkenalan Program

b. Halaman Bantuan Program, halaman ini terdiri dari tiga buah halaman yaitu:
   a. Halaman bantuan penelusuran sederhana, berisi bantuan mengenai penelusuran sederhana. Lebih jelasnya dapat dilihat pada gambar 4.11.
   b. Halaman bantuan penelusuran *advance*, bantuan mengenai penelusuran *advance*. Lebih jelasnya dapat dilihat pada gambar 4.12.
   c. Halaman bantuan setting,bantuan mengenai *setting URL server gateway*. Lebih jelasnya dapat dilihat pada gambar 4.13.

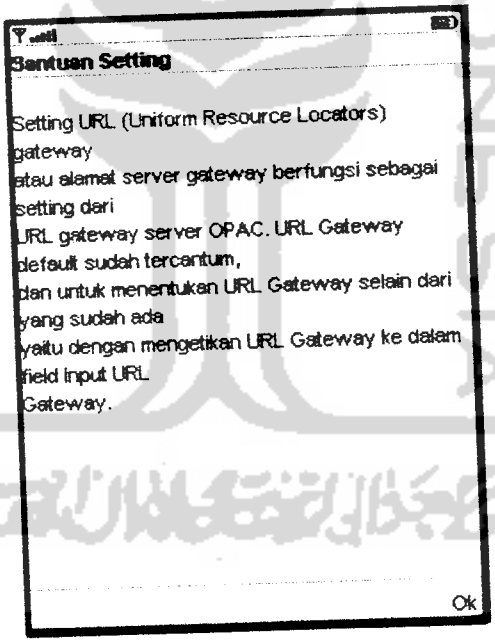
Gambar 4.10. Tampilan Menu Bantuan Program



Gambar 4.11. Tampilan Halaman Bantuan Penelusuran Sederhana

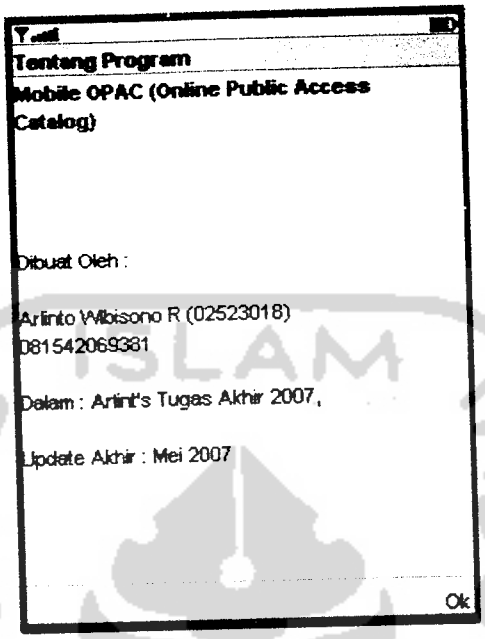


Gambar 4.12. Tampilan Bantuan Penelusuran *Advance* (Ahli)



Gambar 4.13. Tampilan Halaman Bantuan *Setting*

c. Halaman Tentang Program, berisi tentang penjelasan pembuat aplikasi *MIDlet mobile OPAC*. Untuk lebih jelasnya dapat dilihat pada gambar 4.14.



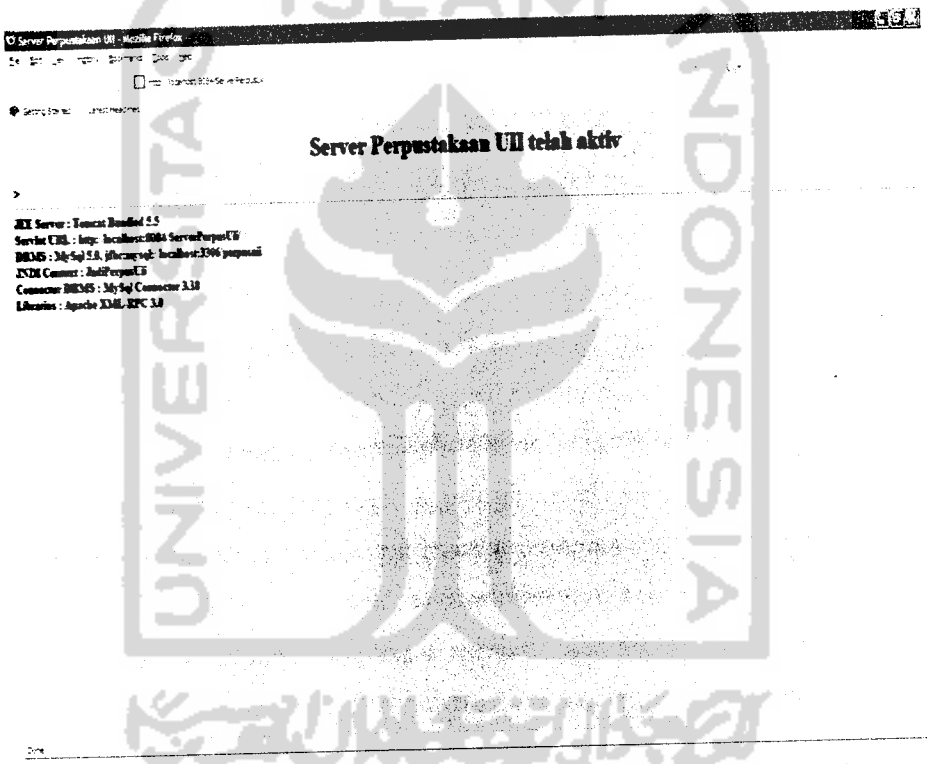
Gambar 4.14. Tampilan Halaman Tentang Program

4.1.4 Impelemtasi Antarmuka *Mobile OPAC (Server)*

Secara umum untuk implementasi antarmuka *server* tidak ada karena aplikasi *server* hanya bertugas memproses permintaan penelusuran *client* dan memberikan hasil penelusuran kepada *server*.

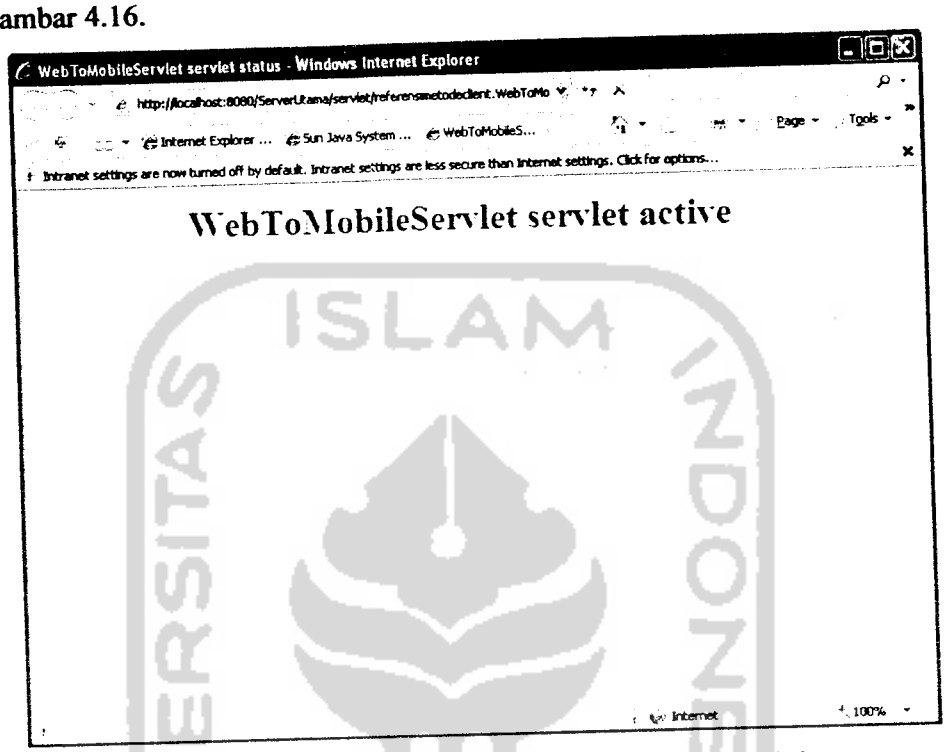
Hanya beberapa saja implementasi antarmuka di sisi *server*, antarmuka tersebut juga hanya untuk memberitahukan status dari *server*, seperti :

a. Antarmuka pemberitahuan jika *server* perpustakaan UII, UGM dan server *gateway* telah aktiv. Untuk lebih jelasnya dapat dilihat pada gambar 4.15.



Gambar 4.15. Tampilan Antarmuka Status *Server* Telah Aktiv

b. Antarmuka pemberitahuan jika *stub* pada sisi *server* yang terdapat pada *server gateway* telah aktiv, atau pemberitahuan bahwa *servlet* telah di-*deploy* dan untuk menangani *request GET*. Untuk lebih jelasnya dapat dilihat pada gambar 4.16.



Gambar 4.16 Tampilan Status *Stub Server* Telah Aktiv

4.1.5 Implementasi Prosedural

Implementasi procedural pada aplikasi ini dikelompokan kedalam kelas-kelas seperti yang telah dijelaskan pada bagian perancangan yaitu:

4.1.5.1 Implementasi Prodesural pada Sisi *Client*

1. ***Class WebToMobileClientMIDlet***

   Metode-metode yang terdapat pada *class* ini adalah sebagai berikut:

a. Metode untuk melakukan *setting URL server gateway*, istansiasi dari *class stub* atau *WebToMobileClient* pada *client*.

   Nama metode : *setServiceURL(String serviceURL)*.

Jenis : *private.*

b. Metode untuk melakukan pemanggilan metode penelusuran sederhana perpustakaan UII berdasarkan judul pada *stub client.*

Nama metode : *call_CariKeUiiPenelusuranSederhanaJudul().*

Jenis : *private.*

c. Metode untuk melakukan *setting* nilai penelusuran sederhana perpustakaan UII berdasarkan judul dari kembalian *server* perpustakaan UII.

Nama metode : *set_CariKeUiiPenelusuranSederhanaJudul_OutputValues().*

Jenis : *private.*

d. Metode untuk melakukan pemanggilan metode penelusuran sederhana perpustakaan UGM berdasarkan pengarang pada *stub client.*

Nama metode : *call_CarikeUgmPenelusuranSederhanaPengarang().*

Jenis : *private.*

e. Metode untuk melakukan *setting* nilai penelusuran sederhana perpustakaan UGM berdasarkan pengarang dari kembalian *server* perpustakaan UGM.

Nama metode :

set_CarikeUgmPenelusuranSederhanaPengarang_OutputValues().

Jenis : *private.*

f. Metode untuk melakukan pemanggilan metode penelusuran *advance* perpustakaan UGM pada *stub client.*

Nama metode : *call_CariKeUgmPenelusuranAdvance().*

Jenis : *private.*

g. Metode untuk melakukan *setting* nilai penelusuran *advance* perpustakaan UGM dari kembalian *server* perpustakaan UGM.

Nama metode : set_*CariKeUgmPenelusuranAdvance_OutputValues().*

Jenis : *private.*

h. Metode untuk melakukan pemanggilan metode penelusuran sederhana perpustakaan UII berdasarkan pengarang pada *stub client.*

Nama metode : *call_CariKeUiiPenelusuranSederhanaPengarang().*

Jenis : *private.*

i. Metode untuk melakukan *setting* nilai penelusuran sederhana perpustakaan UII berdasarkan pengarang dari kembalian *server* perpustakaan UII.

Nama metode : *set_CariKeUiiPenelusuranSederhanaPengarang_OutputValues()*.

Jenis : *private*.

j. Metode untuk melakukan pemanggilan metode penelusuran *advance* perpustakaan UII pada *stub client*.

Nama metode : *call_CariKeUiiPenelusuranAdvance()*.

Jenis : *private*.

k. Metode untuk melakukan *setting* nilai penelusuran *advance* perpustakaan UII dari kembalian *server* perpustakaan UII.

Nama metode : *set_CariKeUiiPenelusuranAdvance_OutputValues()*.

Jenis : *private*.

l. Metode untuk melakukan pemanggilan metode penelusuran sederhana perpustakaan UGM berdasarkan judul pada *stub client*.

Nama metode : *call_CarikeUgmPenelusuranSederhanaJudul()*.

Jenis *: private*.

m. Metode untuk melakukan *setting* nilai penelusuran sederhana perpustakaan UGM berdasarkan judul dari kembalian *server* perpustakaan UGM.

Nama metode : *set_CarikeUgmPenelusuranSederhanaJudul_OutputValues()*.

Jenis : *private*.

2. Class *WebToMobileClient* (*Stub* di Sisi *Client*)

Metode-metode yang terdapat pada *class* ini adalah sebagai berikut:

a. Metode untuk melakukan *invoke* penelusuran sederhana pada *server* perpustakaan UII berdasarkan judul.

Nama metode : *CariKeUiiPenelusuranSederhanaJudul(String Judul)*.

Jenis : *public*.

b. Metode untuk melakukan *invoke* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan pengarang.

Nama metode : *CarikeUgmPenelusuranSederhanaPengarang(String pengarang).*

Jenis : *public.*

c. Metode untuk melakukan *invoke* penelusuran *advance* pada *server* perpustakaan UGM.

Nama metode : *CarikeUgmPenelusuranAdvance(String judul,String jenis,String tokoh,String noddc).*

Jenis : *public*

d. Metode untuk melakukan *invoke* penelusuran sederhana pada *server* perpustakaan UII berdasarkan pengarang.

Nama metode : *CariKeUiiPenelusuranSederhanaPengarang(String pengarang).*

Jenis : *public.*

e. Metode untuk melakukan *invoke* penelusuran *advance* pada *server* perpustakaan UII.

Nama metode : *CariKeUiiPenelusuranAdvance(String judul,String jenis,String tokoh,String noddc).*

Jenis : *public.*

f. Metode untuk melakukan *invoke* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan pengarang.

Nama metode : *CarikeUgmPenelusuranSederhanaJudul(String keywordPenelusuranSederhanaPengarang).*

Jenis : *public.*

g. Metode untuk melakukan *invoke* kepada *server* secara dinamis.

Nama metode : *invokeServer(int requested, Object parameters[0..*]).*

Jenis : *private.*

h. Metode untuk menulis *Object.*

Nama metode : *writeObject(DataOutputStream out, Object o).*

Jenis : *private.*

i. Metode untuk membaca *Object*.

Nama metode : *readObject(DataInOut in)*.

Jenis : *private*.

4.1.5.2 Implementasi Prodesural pada Sisi *Server*

1. Implementasi Prodesural pada Sisi *Server* Utama atau *Gateway*

a. *Class ReferensiMetode*

Metode-metode yang terdapat pada *class* ini adalah sebagai berikut:

a. Metode untuk *invoke server* perpustakaan UII dan melewatkan parameter dari *client* untuk penelusuran sederhana pada *server* perpustakaan UII berdasarkan judul menggunakan *XML-RPC*.

Nama metode : *CariKeUiiOenelusuranSederhanaJudul(String judul)*.

Jenis : *public*.

b. Metode untuk *invoke server* perpustakaan UGM dan melewatkan parameter dari *client* untuk penelusuran sederhana pada *server* perpustakaan UGM berdasarkan judul menggunakan *XML-RPC*.

Nama metode : *CarikeUgmPenelusuranSederhanaJudul(String keywordPenelusuranSederhanaPengarang)*.

Jenis : *public*.

c. Metode untuk *invoke server* perpustakaan UII dan melewatkan parameter dari *client* untuk penelusuran *advance* pada *server* perpustakaan UII menggunakan *XML-RPC*.

Nama metode : *CariKeUiiPenelusuranAdvance(String judul,String jenis,String tokoh,String noddc)*.

Jenis : *public*.

d. Metode untuk *invoke server* perpustakaan UGM dan melewatkan parameter dari *client* untuk penelusuran *advance* pada *server* perpustakaan UGM menggunakan *XML-RPC.*

Nama metode : *CariKeUgmPenelusuranAdvance(String judul,String jenis,String tokoh,String noddc).*

Jenis : *public.*

e. Metode untuk *invoke server* perpustakaan UII dan melewatkan parameter dari *client* untuk penelusuran sederhana pada *server* perpustakaan UII berdasarkan pengarang menggunakan *XML-RPC.*

Nama metode : *CariKeUiiPenelusuranSederhanaPengarang(String pengarang).*

Jenis : *public.*

f. Metode untuk *invoke server* perpustakaan UGM dan melewatkan parameter dari *client* untuk penelusuran sederhana pada *server* perpustakaan UGM berdasarkan pengarang menggunakan *XML-RPC.*

Nama metode : *CariKeUgmPenelusuranSederhanaPengarang(String pengarang).*

Jenis : *public.*

b. *Class WebToMobileServlet (Stub* pada Sisi *Server)*

Metode-metode yang terdapat pada class ini adalah sebagai berikut:

a. Metode untuk menangani permintaan *request GET* dari *client*, mengembalikan pesan bahwa *servlet* telah di-*deploy.*

Nama metode: *doGet(HttpServletRequest req,HttpServletResponse resp).*

Jenis : *protected.*

b. Metode untuk menangani permintaan *request POST* dari *client.*

Nama metode : *doPost (HttpServletRequest req,HttpServletResponse resp).*

Jenis : *protected.*

c. Bagian dari metode untuk menangani permintaan *request POST* dari *client.*

Nama metode : *getDataOutput(DataOutput out).*

Jenis : *protected.*

d. Metode untuk *invoke server.*

Nama metode : *invokeMethod(HttpSession seddion, int requested, DataInput input).*

Jenis *: private.*

2. Implementasi Prodesural pada Sisi *Server* Perpustakaan UII

a. *Class ProsesBisnisServerUii*

Metode-metode yang terdapat pada *class* ini adalah sebagai berikut :

a. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UII berdasarkan judul.

Nama metode : *PenelusuranSederhanaJudul(String judul).*

Jenis : *public.*

b. Metode untuk melakukan proses *query* penelusuran *advance* pada *server* perpustakaan UII.

Nama metode : *PenelusuranAdvanceUii(String judul,String jenis,String tokoh,String noddc).*

Jenis : *public.*

c. Metode untuk melemparkan parameter (pengarang) panelusuran sederhana kepada *server* perpustakaan UGM berdasarkan pengarang jika hasil *query* tidak diketemukan dalam *server* perpustakaan UII.

Nama metode : *CarikeUgmPenelusuranSederhanaJudul(String keywordPenelusuranSederhanaJudul).*

Jenis : *public.*

d. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UII berdasarkan pengarang.

Nama metode : *PenelusuranSederhanaUiiPengarang(String pengarang).*

Jenis : *public.*

e. Metode untuk melemparkan parameter (judul) panelusuran sederhana kepada *server* perpustakaan UGM berdasarkan judul jika hasil *query* tidak diketemukan dalam *server* perpustakaan UII.

Nama metode : *CarikeUgmPenelusuranSederhanaPengarangDariServer(String pengarang)*.

Jenis : *public*.

f. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UII berdasarkan judul setelah *server* perpustakaan UGM melempar parameter judul jika hasil *query* tidak diketemukan pada *server* perpustakaan UGM.

Nama metode : *PenelusuranSederhanaJudulDariServer(String judul)*.

Jenis : *public*.

g. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UII berdasarkan pengarang setelah *server* perpustakaan UGM melempar parameter pengarang jika hasil *query* tidak diketemukan pada *server* perpustakaan UGM.

Nama metode : *PenelusuranSederhanaUiiPengarangDariServer(String pengarang)*.

Jenis : *public*.

h. Metode untuk melakukan proses *query* penelusuran *advance* pada *server* perpustakaan UII setelah *server* perpustakaan UGM melempar parameter penelusuran *advance* jika hasil *query* tidak diketemukan pada server perpustakaan UGM.

Nama metode: PenelusuranAdvanceUiiDariServer(String judul, String jenis, String tokoh, String noddc).

Jenis : *public*.

i. Metode untuk melemparkan parameter panelusuran *advance* kepada *server* perpustakaan UGM jika hasil *query* tidak diketemukan dalam *server* perpustakaan UII.

Nama metode: *CariKeUgmPenelusuranAdvanceDariServer(String judul, String jenis, String tokoh, String noddc).*

Jenis : *public.*

b. *Class AksesDatabaseUii2*

Metode-metode yang terdapat pada *class* ini adalah sebagai berikut:

a. Metode untuk melakukan koneksi kepada *database server* perpus UII.

Nama metode : *getPerpusUii2().*

Jenis : *protected.*

3. Implementasi Prodesural pada Sisi *Server* Perpustakaan UGM

a. *Class LogikaBisnisServerPerpusUgm*

Metode-metode yang terdapat pada class ini adalah sebagai berikut :

a. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan judul.

Nama metode : *PenelusuranSederhanaJudulUgm(String judul).*

Jenis : *public.*

b. Metode untuk melakukan proses *query* penelusuran *advance* pada *server* perpustakaan UGM.

Nama metode : *PenelusuranAdvanceUgm(String judul, String jenis, String pengarang, String noddc).*

Jenis : *public.*

c. Metode untuk melemparkan parameter (judui) panelusuran sederhana kepada *server* perpustakaan UII berdasarkan judul jika hasil *query* tidak diketemukan dalam *server* perpustakaan UGM.

   Nama metode : CariKeUiiPenelusuranSederhanaJudulDariServer(String judul).

   Jenis : *public.*

d. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan pengarang.

   Nama metode : *PenelusuranSederhanaPengarangUgm(String pengarang).*

   Jenis : *public.*

e. Metode untuk melemparkan parameter (pengarang) panelusuran sederhana kepada *server* perpustakaan UII berdasarkan pengarang jika hasil *query* tidak diketemukan dalam *server* perpustakaan UGM.

   Nama metode : *CarikeUiiPenelusuranSederhanaPengarangDariServer(String pengarang).*

   Jenis : *public.*

f. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan judul setelah *server* perpustakaan UII melempar parameter pengarang jika hasil *query* tidak diketemukan pada *server* perpustakaan UII.

   Nama metode : *PenelusuranSederhanaJudulUgmDariServer(String judul).*

   Jenis : *public.*

g. Metode untuk melakukan proses *query* penelusuran sederhana pada *server* perpustakaan UGM berdasarkan pengarang setelah *server* perpustakaan UII melempar parameter pengarang jika hasil *query* tidak diketemukan pada *server* perpustakaan UII.

   Nama metode : *PenelusuranSederhanaPengarangUgmDariServer(String pengarang).*

   Jenis : *public.*

h. Metode untuk melemparkan parameter panelusuran *advance* kepada *server* perpustakaan UII jika hasil *query* tidak diketemukan dalam *server* perpustakaan UGM.

Nama metode : *CariKeUiiPenelusuranAdvanceDariServer(String judul, String jenis, String tokoh, String noddc).*

Jenis *: public.*

i. Metode untuk melakukan proses *query* penelusuran *advance* pada *server* perpustakaan UGM setelah *server* perpustakaan UII melempar parameter penelusuran *advance* jika hasil *query* tidak diketemukan pada *server* perpustakaan UII.

Nama metode : *PenelusuranAdvanceUgmDariServer(String judul, String jenis, String pengarang, String noddc).*

Jenis : *public.*

b. *Class KoneksiDatabasePerpusUgm*

Metode-metode yang terdapat pada class ini adalah sebagai berikut :

a. Metode untuk melakukan koneksi kepada *database server* perpus UGM.

Nama metode : *getOracleXeJndi( ).*

Jenis : *protected.*

4.2 Pembahasan

4.2.1 Pengujian Proses Penelusuran

Pengujian disini dilakukan dengan cara melakukan input data (kata kunci penelusuran), apakah itu pada penelusuran sederhana (pada form penelusuran sederhana) atau pada penelusuran *advance* (pada form penelusuran *advance*).

Untuk penelusuran sederhana, kata kunci penelusuran ialah kata kunci sebagai parameter penelusuran berdasarkan pada kriteria judul atau pengarang, sekaligus pemilihan pada perpustakaan universitas mana yang akan kali pertama ditelusuri.

Untuk penelusuran *advance* kata kunci (parameter) penelusuran yang digunakan yaitu judul, jenis, pengarang, dan nomor *DDC*.

Data-data mengenai koleksi perpustakaan sudah ada pada *database server* masing-masing perpustakaan universitas, *user* hanya dapat melakukan penelusuran data-data tersebut.

Semua data baik data konfigurasi yang dibutuhkan dalam pengujian oleh *server* dan data-data koleksi perpustakaan UII dan UGM terlampir pada lampiran.

Pengujian aplikasi *mobile OPAC* dilakukan dengan beberapa cara sebagai berikut :

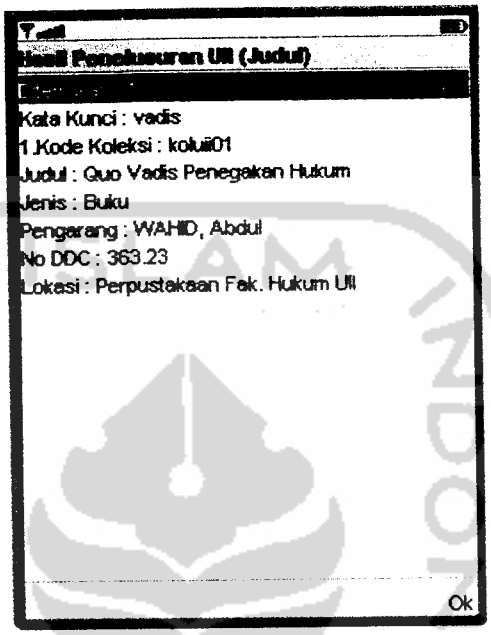
4.2.1.1 Pengujian Proses 1

Pengujian pada Proses 1 adalah penelusuran sederhana pada perpustakaan UII berdasarkan pada kriteria judul. Pengujian ini dilakukan dengan memasukan kata kunci penelusuran sederhana pada UII, memilih pilihan judul dan memilih pilihan perpustakaan UII, ini berarti bahwa *user* menginginkan dilakukanya penelusuran sederhana berdasarkan judul pada perpustakaan Universitass Islam Indonesia.

Sebagai contoh dilakukan pengujian dengan memasukan kata kunci "*vadis*", pilihan kriteria pada judul dan pilihan perpustakaan pada UII, dan untuk mengririm kepada *server user* harus menekan tombol perintah cari. Untuk lebih jelasnya dapat dilihat contoh pada gambar 4.17.

Gambar 4.17. Tampilan Pengujian Penelusuran Sederhana UII

Jika pada *database server* perpustakan UII diketemukan koleksi dengan judul koleksi yang mengandung kata *"vadis"*, maka, *server* akan mengembalikan data koleksi kepada *user (client)*. Untuk lebih jelasnya dapat dilihat pada gambar 4.18.



Gambar 4.18. Tampilan Hasil Pengujian Penelusuran Sederhana Judul UII

4.2.1.2 Pengujian Proses 2

Pengujian pada Proses 2 adalah Dengan Cara Melakukan Penelusuran *Advance* pada Perpustakaan UGM.

Seperti halnya penelusuran sederhana, penelusuran *advance* menggunakan kata kunci untuk melakukan penelusuran, tetapi macamnya lebih banyak dibandingkan dengan penelusuran sederhana. Untuk mengujinya, *user* diharuskan memasukan kata kunci pada parameter-parameter yang telah disediakan, seperti judul, jenis, pengarang dan nomor *DDC*. User diperbolehkan untuk mengisi semua ataupun salah satu saja, misal *user* hanya menginginkan penelusuran koleksi pada perpustakaan UGM dengan kata kunci *"Islam"* dan jenis koleksi *"buku"*.

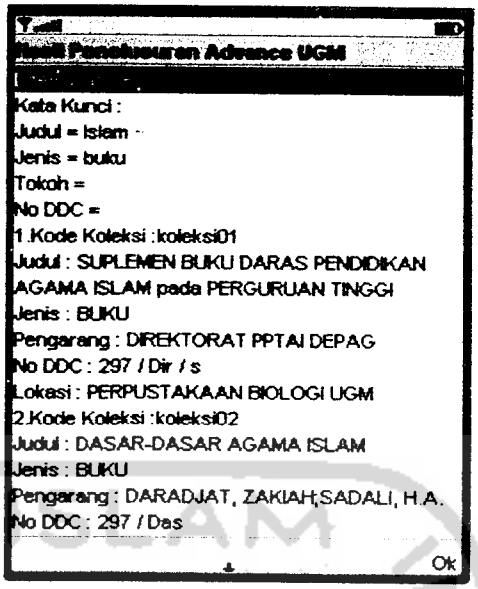
Setelah memasukan kata kunci penelusuran, *user* diharuskan menekan tombol cari untuk mengirim *request* kepada *server.*

Untuk lebih jelasnya dapat dilihat pada gambar 4.19.

Gambar 4.19. Tampilan Pengujian Penelusuran *Advance* UGM

Jika pada *server* perpustakaan UGM diketemukan data koleksi yang di*request* oleh *user*, maka *server* akan mengembalikan data koleksi tersebut kepada *user*. Untuk lebih jelasnya, dapat dilihat pada gambar 4.20

.

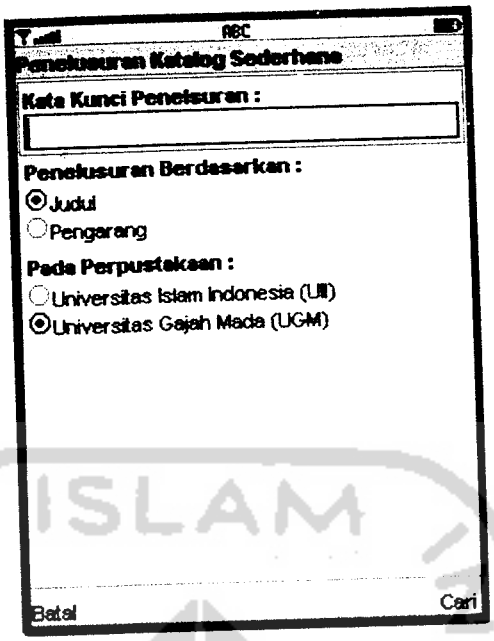


Gambar 4.20. Tampilan Hasil Pengujian Penelusuran *Advance* UGM

4.2.1.2 Pengujian Proses 3

Pengujian pada Proses 3 adalah penelusuran sederhana pada perpustakaan UGM berdasarkan judul, dengan mengosongkon kata kunci penelusuran.

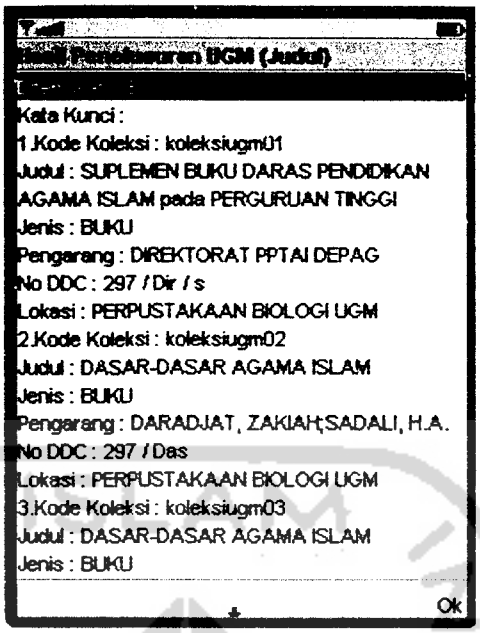
User hanya memilih pilihan kriteria penelusuran judul dan pilihan penelusuran pada perpustakaan UGM, setelah itu untuk mengirimkan kepada *server*, *user* harus menekan tombol cari. Untuk lebih jelasnya dapat dilihat pada gambar 4.21.



Gambar 4.21. Tampilan Pengujian Penelusuran Sederhana perpus UGM

Hasil yang diperoleh jika dilakukan pengujian kata kunci dikosongkan (tidak memasukan kata kunci penelusuran) adalah *server* mnegembalikan semua koleksi yang dimiliki oleh database *server* perpustakaan UGM.

Sehingga, didapatkan semua data koleksi. Untuk lebih jelasnya dapat dilihat dalam gambar 4.22.



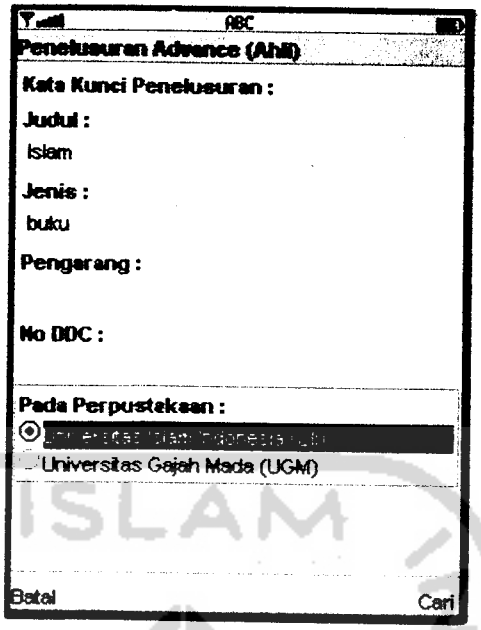
Gambar 4.22. Tampilan Hasil Pengujian Penelusuran UGM Berdasar Judul

4.2.1.3 Pengujian Proses 4

Pengujian proses 4 adalah pengujian *advance* yang dilakukan pada perpustakaan UII. Pengujian dilakukan dengan memasukan kata kunci yang tidak terdapat dalam koleksi perpustakaan UII atau tidak ada dalam *database* perpustakaan UII.

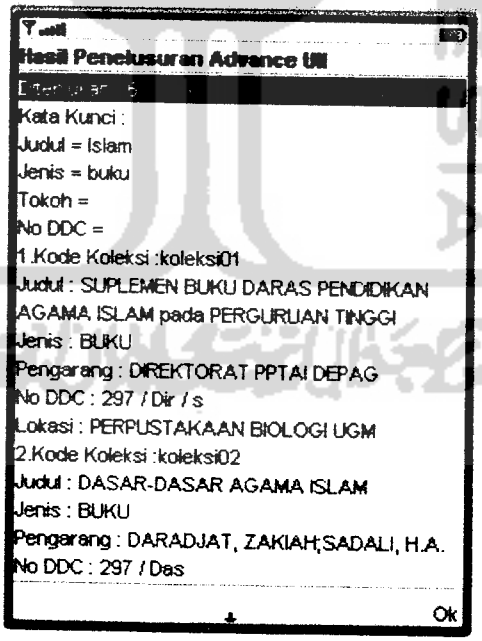
Sebagai contoh, *user* memasukan kata kunci penelusuran *advance "Islam"* dan jenis *"buku"* dan untuk mengirimkan kepada *server*, *user* harus menekan tombol perintah cari.

Untuk lebih jelasnya dapat dilihat pada gambar 4.23.



Gambar 4.23. Tampilan Pengujian Penelusuran *Advance* UII

Setelah dilakukan pencarian pada *database server* UII, maka *server* UII akan mengembalikan data seperti pada gambar 4.24.



Gambar 4.24. Tampilan Hasil Pengujian Penelusuran *Advance* UII

Hasil pengujian penelusuran *advance* pada perpustakaan *UII* menunjukan bahwa hasil yang dikembalikan oleh *server* perpustakaan UII adalah hasil katalog dari perpustakaan UGM, ini dimungkinkan, karena pada aplikasi *mobile OPAC* yang mempunyai konsep sistem terdistribusi, akan melempar parameter pencarian kepada lain *server* yang terdaftar, dalam hal ini adalah *sever* perpustakaan UGM, jika pada *server* UII tidak diketemukan data yang diminta oleh *user*, parameter yang dikirim oleh perpustakaan UII akan ditangkap oleh perpustakaan UGM yang selanjutnya akan melakukan penelusuran pada *server*nya sendiri, jika diketemukan, *server* perpustakaan UGM akan mengembalikan pada *server* perpustakaan UII dan *server* perpustakaan UII akan mengembalikan hasil penelusuran kepada *user*, sehingga dapat dilihat bahwa koleksi yang dikembalikan berada pada lokasi perpustakaan UGM.

G

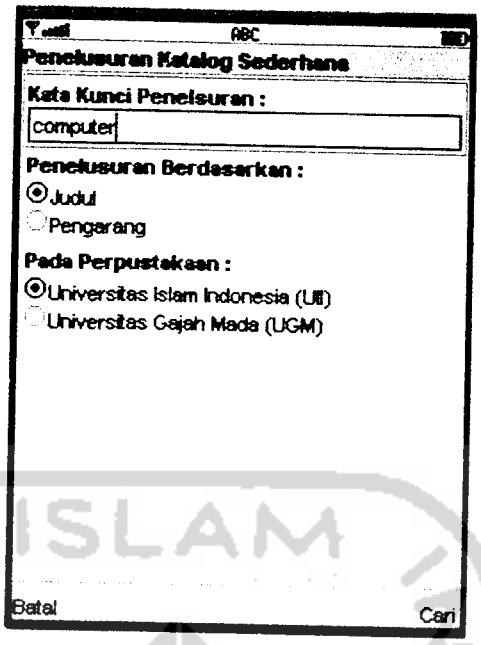
Setelah

nya dap

4.2.1.4 Pengujian Proses 5

Pengujian proses 5 adalah penelusuran sederhana berdasarkan judul pada perpustakaan UII, dilakukan dengan cara *user* memasukan kata kunci penelusuran *"computer"* dan memilih pilihan kriteria judul dan pemilihan perpustakaan UII. Untuk lebih jelasnya dapat dilihat pada gambar 4.25.

Gan



Gambar 4.25. Tampilan Pengujian Penelusuran Sederhana Judul UII

Setelah itu, didapatkan hasil penelusuran dari *server* UII. Untuk lebih jelasnya dapat dilihat pada gambar 4.26.

Gambar 4.26. Tampilan Hasil Pengujian Penelusuran Sederhana UII

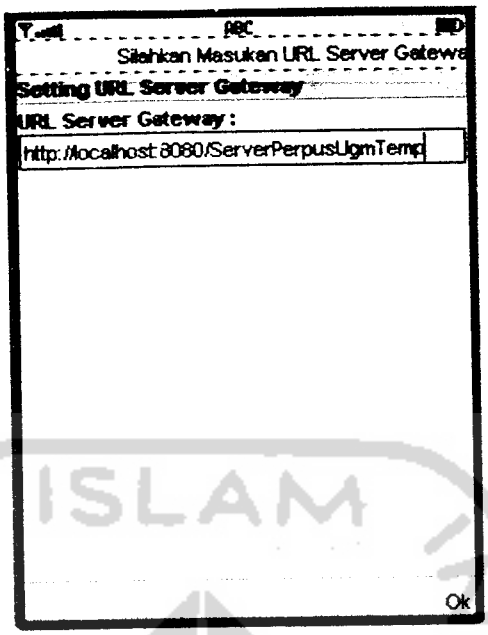
Hasil menunjukan bahwa penelusuran dengan kata kunci *"computer "* tidak diketemukan. Pada saat *server* UII melakukan penelusuran, data tidak diketemukan oleh *server* dan *server* perpustakaan UII melempar parameter kepada lain perpustakaan (UGM), tetapi pada *server* UGM juga tidak diketemukan, maka *server* perpustakaan UGM akan mengembalikan hasil penelusuran yang kosong kepada *server* UII dan *server* UII memberitahukan bahwa koleksi dengan kata kunci *"computer"* tidak diketemukan kepada *user*.

4.2.1.6 Pengujian Proses 6

Pengujian proses 6 yaitu pengujian tentang *service URL gateway* adalah pengujian yang dilakukan kepada masukan *URL server gateway*, dimana dilakukan pengujian dengan memasukan *URL server gateway* yang tidak benar atau salah.

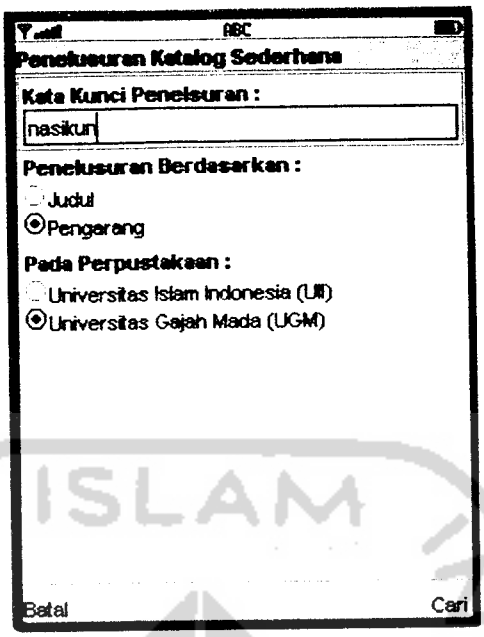
URL server gateway menandakan tempat dimana *server gateway* berada. Seperti contoh dilakukan pengujian, pada *URL* diubah dengan URL *http://localhost:8080/ServerPerpusUgmtemp/*.

User harus mengetikan *URL* tersebut pada *field* yang disediakan pada *form setting url gateway*. Setelah itu, *user* diharuska menekan tombol perintah OK. Untuk lebih jelasnya dapat dilihat dalam gambar 4.27.



Gambar 4.27. *Tampilan Pengujian Setting URL gateway*

Setelah dilakukan perubahaan pada *URL server gateway* yang bukan *default URL* (*default URL* adalah *http://localhost:8080/ServerUtama/servlet/referensimetodeclient.WebToMobileServlet* diubah menjadi *URL http://localhost:8080/ServerPerpusUgmtemp*), maka, akan dilakukan uji penelusuran sederhana berdasarkan pengarang pada perpustakaan UGM. Untuk lebih jelasnya dapat dilihat gambar 4.28.



Gambar 4.28 Tampilan Pengujian Penelusuran Sederhana UGM

Setelah dilakukan perubahan *URL*, maka akan menghasilkan hasil penelusuran seperti pada gambar 4.29.

Gambar 4.29 Tampilan Pesan Kesalahan pada Peguujian *URL*

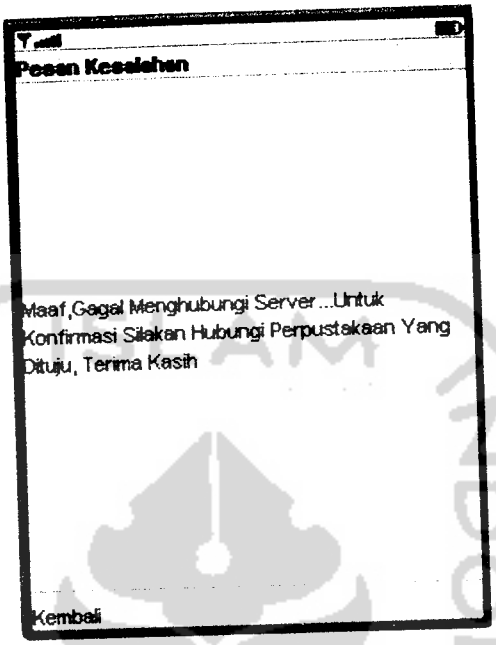
Karena *server URL gateway* tidak *default* atau *URL* salah, maka akan dikembalikan berupa peringatan bahwa terjadi kesalahan pada saat menghubungi *server.*

4.2.1.7 Pegujian Proses 7

Pengujian proses 7 dilakukan dengan melakukan penelusuran *advance* pada perpustakaan UII dengan mematikan server UII. Semua *field* pada *form* penelusuran *advance* UII diisi dan dikirim kepada *sever* dengan menekan tombol perintah cari. Untuk lebih jelasnya dapat dilihat dalam gambar 4.30.

Gambar 4.30. Tampilan Halaman Penelusuran *Advance* UII

Hasil penelusuran yang di kembalikan oleh server adalah seperti dalam gambar 4.31.



Gambar 4.31. Tampilan Halaman Pesan Kesalahan

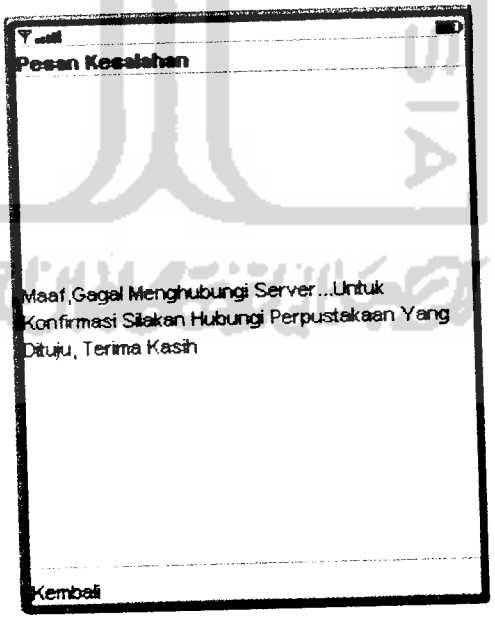
4.2.1.8 Pengujian Proses 8

Pengujian proses 8 dilakukan jika *server gateway* mati atau tidak aktiv adalah dilakukan dengan cara melakukan penelusuran *advance* pada *server* perpustakaan UII,

dengan mengisi semua kriteria pencarian dan mengirimkanya kepada *server*. Untul lebih Jelasnya dapat dilihat pada gambar 4.32.

Gambar 4.32. Tampilan Pengujian Penelusuran *Advance* UGM

Hasil yang didapatkan adalah seperti gambar 4.33.



Gambar 4.33. Tampilan Kesalahan Pengujian Jika *Server Gateway Down*

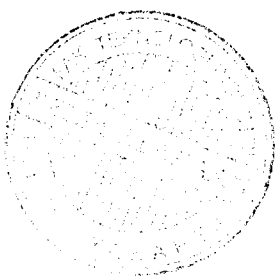
Hasil tersebut didapatkan, karena *client* tidak dapat menghubungi *server*, sehingga muncul peringatan bahwa *server* tidak dapat dihubungi atau terdapat kesalahan pada *server*.

4.2.2. Intisari Proses Pengujian Penelusuran

Setelah dilakukan beberapa proses pengujian penelusuran maka,dapat dibuat sebuah intisari dari proses pengujian penelusuran yaitu dapat dilihat lebih jelas pada tabel 4.1.a dan table 4.1.b.

Tabel 4.1.a. Intisari Proses Pengujian Penelusuran

No	Proses	Pengujian	Variabel Pengujian	Kesimpulan
1	Proses 1	Penelusuran Sederhana UII	Judul = "vadis"	Diperoleh data hasil pencarian berupa kode koleksi, judul, jenis, pengarang, nomor *DDC*, dan lokasi perpustakaan sesuai dengan variabel pengujian.
2	Proses 2	Penelusuran *Advance* UGM	Judul = "Islam" Jenis = "Buku"	Diperoleh data hasil pencarian berupa kode koleksi, judul, jenis, pengarang, nomor *DDC*, dan lokasi perpustakaan sesuai dengan variabel pengujian.
3	Proses 3	Penelusuran Sederhana UGM	Dikosongkan	Diperoleh semua data koleksi yang dimiliki oleh perpustakaan UGM.
4	Proses 4	Penelusuran *Advance* UGM	Dikosongkan	Diperoleh semua data koleksi yang dimiliki oleh perpustakaan UII.
5	Proses 5	Penelusuran Sederhana UII	Judul = "Computer"	Tidak diketemukan karena koleksi tidak tersedia pada perpustakaan UII maupun UGM.



Tabel 4.1.b. Lanjutan Intisari Proses Pengujian Penelusuran

No	Proses	Pengujian	Variabel Pengujian	Kesimpulan
6	Proses 6	Penelusuran Sederhana UGM	Pengarang = "nasikun" dan *input URL gateway* = "*http://localhost:8080/ServerPerpusUgmtemp/*"	Data hasil koleksi penelusuran tidak didapatkan dan muncul pesan kesalahan menghubungi *server* karena input *URL gateway* salah.
7	Proses 7	Penelusuran *Advance* UII	Judul = "vadis", Jenis = "buku", Pengarang = "wahid" dan nomor *DDC* = "363" dengan mematikan server UII	Data hasil koleksi penelusuran tidak didapatkan dan muncul pesan kesalahan menghubungi server karena server UII dimatikan.
8	Proses 8	Penelusuran *Advance* UGM	Judul = "suplemen", Jenis = "buku", Pengarang = "depag" dan nomor *DDC* = "297" dengan mematikan *server gateway*	Data hasil koleksi penelusuran tidak didapatkan dan muncul pesan kesalahan menghubungi *server* karena *server Gateway* dimatikan..

BAB V

SIMPULAN dan SARAN

5.1 Simpulan

Kesimpulan yang dapat diambil dari pengembangan sistem *mobile OPAC* yaitu :

a. Sistem yang dikembangkan menggunakan konsep terdistribusi sangat memberikan keuntungan baik kepada pengembang dan kepada pengguna sistem, karena sistem dibagi menjadi tiga buah level yaitu presentasi, logika bisnis dan data.
b. Aplikasi dengan menggunakan konsep sistem terdistribusi dapat berjalan dengan sumber daya sistem yang berbeda-beda, apakah antar sistem operasi atau *DBMS*, karena salah satu keuntungan dari sistem terdistribusi adalah interoperasi sumber daya.
c. Pengembangan *web* berbasis *java* sangat mudah dilakukan dengan menggunakan *IDE NetBeans*.
d. Dalam penelitian, dirancang dan diimplementasikanya sistem dengan sistem operasi, *DBMS*, *server Java EE*, *device* yang berbeda dimagsudkan untuk memberikan simulasi bahwa system dapat bekerja pada kondisi sumber daya system yang berbeda-beda.
e. Setelah dilakukan pengembangan system *mobile OPAC* dalam penelitian ini, maka dapat dibuktikan dari suatu definisi tentang sistem terdistribusi yaitu suatu kesatuan sistem yang saling berbagi *resource* (sumber daya) dan berada pada tempat yang berbeda-beda.
f. Meskipun *XML-RPC* adalah teknologi *web service* yang *open* dan tidak *standar* dari *W3C*, *XML-RPC* masih mempunyai fungsi yang tidak kalah dengan tekniologi *web service* lainya (*SOAP*) yang terstandarisasi oleh *W3C*.

5.2. Saran

Beberapa saran untuk pengembangan sistem pada penelitian yang berikutnya adalah sebagai berikut :

a. Atribut peneluruan lebih diperbanyak dari penelitian yang sudah ada, karena semakin banyaknya koleksi perpustakaan, maka diperlukan atribut penelusuran yang lebih lengkap untuk menghasilkan penelusuran katalog perpustakaan yang lebih akurat.
b. Tata letak komponen antarmuka *client* lebih dipermudah bagi *user*.
c. Dimungkinkan semakin bertambahnya koleksi perpustakaan, dan pengguna yang mengakses *server* perpustakaan dengan platform yang berbeda-beda maka, akan lebih efektif jika sistem dikembangkan menggunakan teknologi *web service* lainya seprti *SOAP*.
d. Perubahan sudut pandang *mobile OPAC* yaitu dari orientasi penelitian menjadi fungisonalitas secara nyata.

DAFTAR PUSTAKA

[BUD05] Budiharto, Widodo. 2005. *Panduan Lengkap Pemrograman J2EE.* Yogyakarta: Andi.

[CAH06] Cahyono, Setiyo. 2006. *Pemrograman Database Menggunakan MySQL dan JAVA.* Bandung: Informatika.

[DIR05] Dirgahayu, Teduh. 2005. *Diktat Kulian Sistem Tersebar.* Yogyakarta: Universitas Islam Indonesia.

[FOW05] Fowler. 2005. *UML Distiled Edisi3.* Yogyakarta: Andi.

[HAR05] Hariyanto, Bambang. 2005. *Esensi-Esens Bahasa Pemrograman Java.* Bandung: Informatika.

[MAR06] Mardiono, Tri. 2006. *Membanagun Solusi Mobile Business dengan Java.* Jakarta: Elex Media Komputindo.

[PUR00] Purwandari, Betty. 2000. *Sistem Terdistribusi iki 30220* (online) Aviable at

[PUR03] Dharwiyanti, Sri & Wahono,Romi Satria, 2003. *Pengantar Unified Modeling Language (UML)* (online) available at

[SHL06] Shalahudin,M & A.S. Rosa. 2006. *J2EE Dalam Aplikasi Enterprise.* Bandung: Informatika.

[SHL06] Shalahudin,M & A.S, Rosa. 2006. *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile.* Bandung: Informatika.

[TCP07] 2007. *Pengenalan TCP* (online) available at

[THA05] Thamura, Frans. 2005. *Teknik Mengembangkan Aplikasi Enterprise dengan Teknologi Open Source Berbasis Java.* Yogyakarta: Andi.

[UTD04] Utdirartatmo, Firrar. 2004. *Pengembangan Aplikasi Java Berbasis Web dengan JBuilder*. Yogyakarta: Gava Media.

LAMPIRAN A

Konfigurasi-konfigurasi yang dibutuhkan oleh *server* :

1. Konfigurasi yang dibutuhkan pada *server* Utama atau *Gateway* :
   a. *JEE Server* : *Sun Java System Application Server PE 9.0.*
   b. *Servlet URL* : http://localhost:8080/ServerUtama
   c. *Stub* :
      http://localhost:8080/ServerUtama/Servlet/referensimetodeclient.WebToMobileServlet

2. Konfigurasi yang dibutuhkan pada *server* Perpustakaan UII
   a. Layanan yang disediakan oleh Perpustakaan UII
      Terletak dalam *package org.apache.xmlrpc.webserver*
      `layananperpusuii=serveruii.ProsesBisniServerUii`
   b. *JEE Server* : *Tomcat Bundled 5.5*
   c. *Servlet URL* : http://localhost:8084/ServerPerpusUii/
   d. *DBMS* : *MySql 5.0*, jdbc:mysql://localhost:3306/perpusuii
   e. *JNDI Context* : PerpusUii2
   f. *Connector DBMS : MySql Connector 3.18*
   g. *Libraries : Apache XML-RPC 3.0*

3. Konfigurasi pada *server* Perpustakaan UGM

   a. Layanan yang disediakan Perpustakaan UGM

      Terletak dalam *package org.apache.xmlrpc.webserver*

      ```
      layananperpusugm=serverperpusugm.LogikaBisnisServerPer
      pusUgm
      ```

   b. *JEE Server : Sun Java System Application Server PE 9.0*

   c. *Servlet URL* : http://localhost:8080/ServerPerpusUgmtemp/

   d. *DBMS : Oracle 10g XE*, jdbc:oracle:thin:@localhost:1521

   e. *JNDI Context : OracleXeJndi*

   f. *Connector DBMS : Ojdbc1.4*

   g. *Libraries : Apache XML-RPC 3.0*

LAMPIRAN B

Data koleksi perpustakaan :

1. Data Koleksi Perpustakaan UII

NO	Kode Koleksi	Judul	Jenis	Pengarang	No.DDC	Lokasi Perpus
1	koluii01	Quo Vadis Penegakan Hukum	Buku	WAHID, Abdul	363.23	Fak. Hukum UII
2	koluii02	Tindak Pidana Korupsi di Indonesia	Buku	WIYONO	343	Fak. Hukum UII
3	koluii03	Laporan Tahunan 2004: PT. Petrosea Tbk.	Laporan Tahunan	Petrosea	336.068 1	Fak. Ekonomi UII
4	koluii04	Laporan Tahunan : Annual Report	Laporan Tahunan	Petrosea	336.068 1	Fak. Ekonomi UII
5	koluii05	Biosorpsi Ion Pb2+ dengan Menggunakan Cangkang Telur Ayam	Laporan Penelitian	INDRIYANTI, Danik	660.207	FTI UII
6	koluii06	Analisa Kadar Pati pada Tepung Bonggol Pisang	Laporan Penelitian	SUSANTO, Tony	660.207	FTI UII
7	koluii07	edit. Da'irot Ma'arif	Referensi	WAJDI, Muhammad. Farid	2X0.3	Fak. Ilmu Agama Islam UII
8	koluii08	Ensiklopedi Islam Buku	Referensi	RIDWAN, Kafrawi	2X0.3	Fak. Ilmu Agama Islam UII
9	koluii09	Applied psychology	Jurnal	MURPHY, K.R	150.5	Fak.Psikologi UII
10	koluii10	Applied psychology	Jurnal	BOBKO, Philip	150.5	Fak.Psikologi UII

2. Data Koleksi Perpustakaan UGM

NO	Kode Koleksi	Judul	Jenis	Pengarang	No.DDC	Lokasi Perpus
1	koleksiugm01	SUPLEMEN BUKU DARAS PENDIDIKAN AGAMA ISLAM pada PERGURUAN TINGGI	BUKU	DIREKTORAT PPTAI DEPAG	297 / Dir / s	BIOLOGI UGM
2	koleksiugm02	DASAR-DASAR AGAMA ISLAM	BUKU	DARADJAT, ZAKIAH;SADALI, H.A.	297 / DAS	BIOLOGI UGM
3	koleksiugm03	DASAR-DASAR AGAMA ISLAM	BUKU	DARADJAT, ZAKIAH	97 / Das / d	PERTANIAN UGM
4	koleksiugm04	TUNTUTAN AGAMA ISLAM	BUKU	PROYEK PENERANGAN, BIMBINGAN dan DAKWAH/KHUTBAH AGAMA ISLAM	297.1 / Pro /t	PERTANIAN UGM
5	koleksiugm05	EMPAT KULIAH ISLAM pada PERGURUAN TINGGI	BUKU	RASJIDI, H.M	297.1 /ras /E	TEKNIK UGM
6	koleksiugm06	POKOK-POKOK AGAMA ISLAM (TINJAUAN SELINTAS)	BUKU	NASIKUN	297 / NAS /p	TEKNIK UGM
7	koleksiugm07	PROSIDING SEMINAR MANAJEMEN dan PEMASARAN INFORMASI PERPUSTAKAAN,	PROSEDING		R / 025.5.1[082.2] / P	BIOLOGI UGM

LAMPIRAN A

Konfigurasi-konfigurasi yang dibutuhkan oleh *server* :

1. Konfigurasi yang dibutuhkan pada *server* Utama atau *Gateway* :
   a. *JEE Server* : *Sun Java System Application Server PE 9.0.*
   b. *Servlet URL* : http://localhost:8080/ServerUtama
   c. *Stub* :
      http://localhost:8080/ServerUtama/Servlet/referensimetodeclient.WebToMobileServlet

2. Konfigurasi yang dibutuhkan pada *server* Perpustakaan UII
   a. Layanan yang disediakan oleh Perpustakaan UII
      Terletak dalam *package org.apache.xmlrpc.webserver*
      `layananperpusuii=serveruii.ProsesBisniServerUii`
   b. *JEE Server* : *Tomcat Bundled 5.5*
   c. *Servlet URL* : http://localhost:8084/ServerPerpusUii
   d. *DBMS* : *MySql 5.0*, jdbc:mysql://localhost:3306/perpusuii
   e. *JNDI Context* : PerpusUii2
   f. *Connector DBMS : MySql Connector 3.18*
   g. *Libraries : Apache XML-RPC 3.0*

3. Konfigurasi pada *server* Perpustakaan UGM

   a. Layanan yang disediakan Perpustakaan UGM

      Terletak dalam *package org.apache.xmlrpc.webserver*

      ```
      layananperpusugm=serverperpusugm.LogikaBisnisServerPerpusUgm
      ```

   b. JEE Server : Sun Java System Application Server PE 9.0

   c. Servlet URL : http://localhost:8080/ServerPerpusUgmtemp/

   d. DBMS : Oracle 10g XE, jdbc:oracle:thin:@localhost:1521

   e. JNDI Context : OracleXeJndi

   f. Connector DBMS : Ojdbc1.4

   g. Libraries : Apache XML-RPC 3.0

LAMPIRAN B

Data koleksi perpustakaan :

1. Data Koleksi Perpustakaan UII

NO	Kode Koleksi	Judul	Jenis	Pengarang	No.DDC	Lokasi Perpus
1	koluii01	Quo Vadis Penegakan Hukum	Buku	WAHID, Abdul	363.23	Fak. Hukum UII
2	koluii02	Tindak Pidana Korupsi di Indonesia	Buku	WIYONO	343	Fak. Hukum UII
3	koluii03	Laporan Tahunan 2004: PT. Petrosea Tbk.	Laporan Tahunan	Petrosea	336.068 1	Fak. Ekonomi UII
4	koluii04	Laporan Tahunan : Annual Report	Laporan Tahunan	Petrosea	336.068 1	Fak. Ekonomi UII
5	koluii05	Biosorpsi Ion Pb2+ dengan Menggunakan Cangkang Telur Ayam	Laporan Penelitian	INDRIYANTI, Danik	660.207	FTI UII
6	koluii06	Analisa Kadar Pati pada Tepung Bonggol Pisang	Laporan Penelitian	SUSANTO, Tony	660.207	FTI UII
7	koluii07	edit. Da'irot Ma'arif	Referensi	WAJDI, Muhammad. Farid	2X0.3	Fak. Ilmu Agama Islam UII
8	koluii08	Ensiklopedi Islam Buku	Referensi	RIDWAN, Kafrawi	2X0.3	Fak. Ilmu Agama Islam UII
9	koluii09	Applied psychology	Jurnal	MURPHY, K.R	150.5	Fak.Psikologi UII
10	koluii10	Applied psychology	Jurnal	BOBKO, Philip	150.5	Fak.Psikologi UII

2. Data Koleksi Perpustakaan UGM

NO	Kode Koleksi	Judul	Jenis	Pengarang	No.DDC	Lokasi Perpus
1	koleksiugm01	SUPLEMEN BUKU DARAS PENDIDIKAN AGAMA ISLAM pada PERGURUAN TINGGI	BUKU	DIREKTORAT PPTAI DEPAG	297 / Dir / s	BIOLOGI UGM
2	koleksiugm02	DASAR-DASAR AGAMA ISLAM	BUKU	DARADJAT, ZAKIAH;SADALI, H.A.	297 / DAS	BIOLOGI UGM
3	koleksiugm03	DASAR-DASAR AGAMA ISLAM	BUKU	DARADJAT, ZAKIAH	97 / Das / d	PERTANIAN UGM
4	koleksiugm04	TUNTUTAN AGAMA ISLAM	BUKU	PROYEK PENERANGAN, BIMBINGAN dan DAKWAH/KHUTBAH AGAMA ISLAM	297.1 / Pro /t	PERTANIAN UGM
5	koleksiugm05	EMPAT KULIAH ISLAM pada PERGURUAN TINGGI	BUKU	RASJIDI, H.M	297.1 /ras /E	TEKNIK UGM
6	koleksiugm06	POKOK-POKOK AGAMA ISLAM (TINJAUAN SELINTAS)	BUKU	NASIKUN	297 / NAS /p	TEKNIK UGM
7	koleksiugm07	PROSIDING SEMINAR MANAJEMEN dan PEMASARAN INFORMASI PERPUSTAKAAN,	PROSEDING		R / 025.5.1[082.2] / P	BIOLOGI UGM

NO	Kode Koleksi	Judul	Jenis	Pengarang	No.DDC	Lokasi Perpus
8	koleksiugm08	PROSIDING SEMINAR MANAJEMEN dan PEMASARAN INFORMASI PERPUSTAKAAN 21 DESEMBER 1992,	PROSEDING		025.5[063] / Pro	PERTANIAN UGM