

Amie, terimakasih kerjasamanya, pertemanaya, Arlint humas dapat nilai apa yah ☺

• Semua Voip 2002, semua Voip 2002, Semua Voip .2002 und maaf ya buanyak baged kalau tidak disertakan satu-satu.

• **Mbak Chirst makasih Arlint buat Biolanya, Mas Tanto yang suabar ngajari Enterprise Servernya, Sunmor mania Lindy, Dea, Wawan keep healthy and ceria...,Gadged2ku..., Orang-orang pendukung, teman, kenal Arlint atau semua, terimakasih ya**

Comming Up, in Theater Dunia Akhirat.....

"Ijab sah,

Und

Der große Geld,

dan Tuntunan



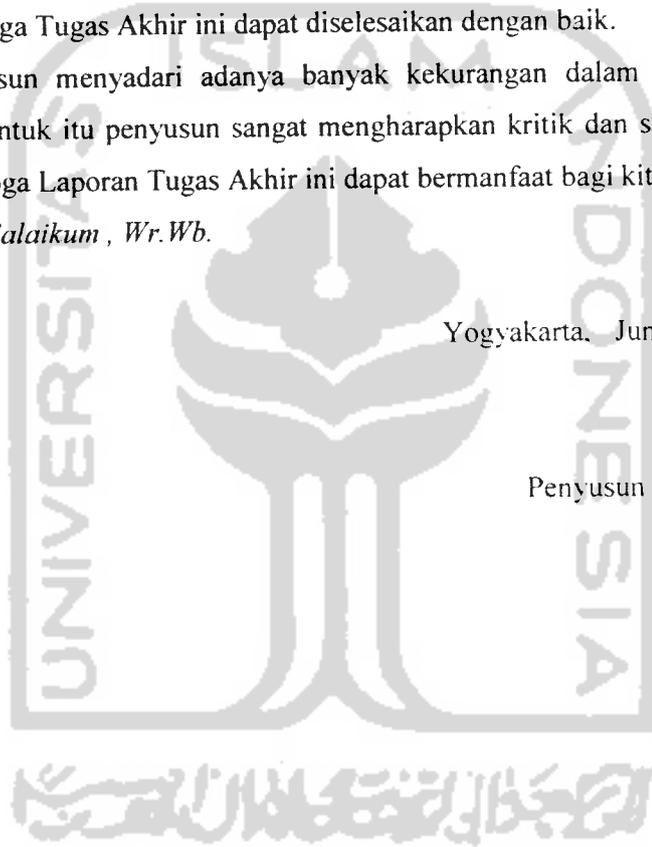
5. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Program Studi Teknik Informatika, Universitas Islam Indonesia
6. Bapak Taufiq Hidayat ST., MCS., selaku dosen pembimbing yang telah membimbing sehingga Tugas Akhir ini dapat diselesaikan dengan sebaik-baiknya.
7. Seluruh karyawan dan staf bagian pengajaran khususnya bagian KP/TA yang telah membantu penyusun dalam perijinan dan surat menyurat.
8. Terima kasih juga kepada semua teman-teman yang tidak dapat disebutkan satu persatu, yang telah membantu memberikan dukungan dan ilmu sehingga Tugas Akhir ini dapat diselesaikan dengan baik.

Penyusun menyadari adanya banyak kekurangan dalam Laporan Tugas Akhir ini, untuk itu penyusun sangat mengharapkan kritik dan saran dari semua pihak. Semoga Laporan Tugas Akhir ini dapat bermanfaat bagi kita semua.

Wassalamu'alaikum , Wr.Wb.

Yogyakarta, Juni 2007

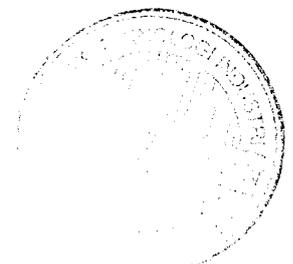
Penyusun



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN DOSEN PENGUJI.....	iii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO	vii
KATA PENGANTAR.....	viii
SARI.....	x
TAKARIR.....	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvi
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	2
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	4
1.7 Sistematika Penelitian.....	4
BAB II LANDASAN TEORI.....	6
2.1 Perpustakaan	6
2.2 Katalog dan Sistem <i>OPAC</i>	6
2.3 Sistem Terdistribusi.....	8
2.4 <i>TCP/IP</i>	8
2.5 <i>GPRS</i> dan Koneksi <i>HTPP</i>	10

2.6	<i>XML-RPC</i>	13
	2.6.1 <i>XML</i>	15
	2.6.2 <i>RPC</i>	15
2.7	Bahasa Pemrograman Java	16
	2.7.1 <i>Java™ Platform, Standard Edition</i>	16
	2.7.2 <i>Java™ Platform, Micro Edition</i>	17
	2.7.3 <i>Java™ Platform, Enterprice Edition</i>	20
2.8	<i>Database</i>	26
	2.8.1 <i>SQL</i>	27
	2.8.2 <i>JDBC</i>	27
2.9	<i>UML</i>	28
	2.9.1 <i>Use Case Diagram</i>	28
	2.9.2 <i>Class Diagram</i>	29
	2.9.3 <i>Sequence Diagram</i>	30
BAB III	METODOOGI	31
3.1	Analisis Kebutuhan	31
	3.1.1 Metode Analisis	31
	3.1.2 Hasil Analisis	31
3.2	Perancangan Sistem	37
	3.2.1 Metode Perancangan	37
	3.2.2 Hasil Perancangan	37
	3.2.3 Perancangan UML dan Lainnya	38
BAB IV	HASIL dan PEMBAHASAN	82
4.1	Hasil	82
	4.1.1 Implementasi Sistem <i>Mobile OPAC</i> Secara Umum	82
	4.1.2 Implementasi Struktur Sistem <i>Mobile OPAC</i>	82
	4.1.3 Implemetasi Antarmuka Sistem <i>Mobile OPAC</i>	84
	4.1.4 Impelemtasi Antarmuka <i>Mobile OPAC (Server)</i>	95
	4.1.5 Implementasi Prosedural	96
4.2	Pembahasan	106
	4.2.1 Pengujian Proses Penelusuran	106



menelusur buku yang ada dan ternyata tidak dapat ditemukan, tentunya akan membuat kecewa pencarinya. Dan dimungkinkan juga setelah gagal ditemukan, penelusur menginginkan untuk menelusuri pada perpustakaan lain yang ada. Ini sangat merepotkan dimana penelusur diharuskan untuk berpindah tempat dari satu perpustakaan kepada perpustakaan lainnya.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas dapat dirumuskan bagaimana merancang suatu aplikasi *mobile* sebagai aplikasi pada sisi *client* yang dapat membantu pengguna sistem *OPAC* (*Online Public Access Catalog*) dalam melakukan penelusuran secara *online* melalui koneksi *GPRS*, tanpa sepengetahuan pengguna sistem, *server* akan melakukan penelusuran kepada lain *server*, dan untuk mengetahui dimana letak sumber daya perpustakaan (buku, laporan, dan lain sebagainya) yang akan dicari atau ditelusuri oleh pengguna sistem, maka perlu dibangun pada setiap *server* perpustakaan, suatu sistem (aplikasi) pada *server* (*Application Server*) berikut dengan *database server* sebagai tempat untuk penyimpanan data koleksi perpustakaan.

1.3 Batasan Masalah

Untuk mengurangi masalah yang melebar atau meluasnya pembahasan dan tidak terfokus pada pokok permasalahan, maka diadakan pembatasan masalah yaitu :

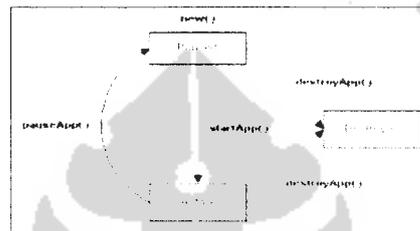
- a. Digunakan media telepon selular yang mendukung *MIDlet* untuk melakukan penelusuran sistem *OPAC* dan menggunakan koneksi *GPRS* pada aplikasi *java client*, berikut pengembalian permintaan penelusuran koleksi juga dikembalikan dan dapat disajikan pada telepon seluler.
- b. Penelusuran koleksi dilakukan pada jaringan perpustakaan yang telah didefinisikan atau didaftarkan sebelumnya.
- c. Sumber daya yang dicari berupa koleksi buku, penelitian, skripsi atau karya ilmiah, tergantung dari koleksi yang tersedia dari universitas yang telah didefinisikan sebelumnya.

2.7.2.1 MIDlet

Aplikasi Java™ Platform, Micro Edition yang menggunakan Configuration CLDC dan *profile MIDP* dapat dinamakan *MIDlet* atau *MIDlet* adalah aplikasi yang ditulis untuk *MIDP*. *MIDP* adalah bagian dari class *javax.microedition.midlet.MIDlet* yang didefinisikan pada *MIDP*. *MIDlet* harus mempunyai beberapa metode yang harus ada yaitu *constructor()*, *protected void startApp() throws MIDletStateChangeException*, *protected void pauseApp()*, *protected void destroyApp(boolean unconditional) throws MIDletStateChangeException*.

MIDlet mempunyai :

- a. Daur Hidup *MIDlet*.



Gambar 2.6. Daur Hidup *MIDlet*

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

Tabel 2.5a Lanjutan Metode-Metode pada *Package Servlet*

Nama	Jenis	Penjelasan
UnaviableException	Class	Mendefinisikan suatu <i>Exception</i> yang di-throw oleh <i>servlet</i> untuk menandakan dirinya secara temporer atau permanen tidak aktif.

2.7.3.2 Apache Tomcat

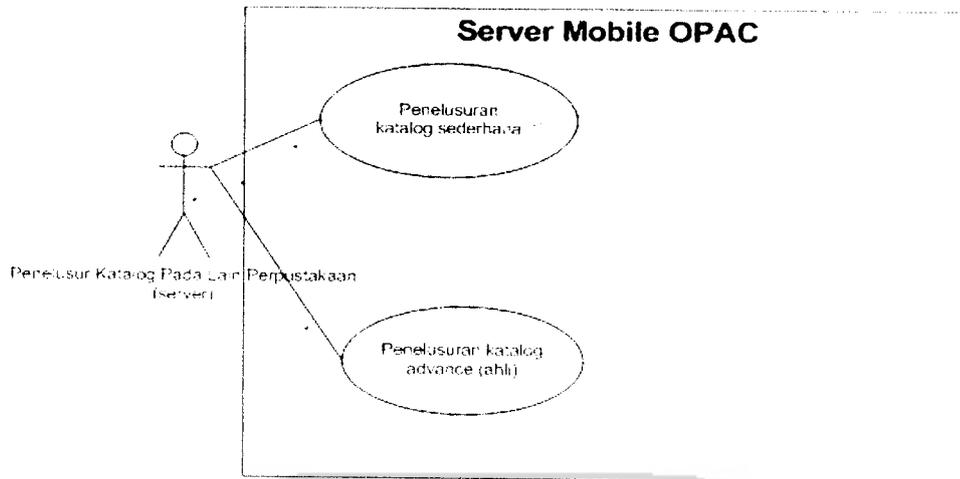
Seperti yang sudah diterangkan sebelumnya bahwa *Java™ Platform, Enterprise Edition* mempunyai layanan berupa *container* dan *container* ini dibagi menjadi dua, yaitu yang disebut *Java™ Platform, Enterprise Edition container lite* (*container* yang ringan) dan ada pula yang disebut *Java™ Platform, Enterprise Edition full feature*. Contoh *container* yang *full* adalah *Websphere*, *Weblogic*, *jBoss* dan *Jonas* adapun yang *lite* merupakan implementasi dari *servlet* saja yang merupakan salah satu spesifikasi *Java™ Platform, Enterprise Edition*. *Java™ Platform, Enterprise Edition* tersebut diciptakan berdasarkan kebutuhan akan sebuah solusi yang lebih ringan, salah satu contohnya adalah *jetty*, dan *tomcat*. [THA05]

Tomcat merupakan implementasi *servlet* yang berbentuk *application server* yang dapat berjalan seperti halnya *web server*. Keistimewaan *tomcat* ini adalah semua *JavaBean* yang dikembangkan dalam aplikasi *java* lainnya seperti *java desktop*, *java web service* ataupun *java mobile* dapat digunakan di dalam *tomcat*.

2.7.3.3 ANT

Ant adalah sebuah alat bantu dalam *deployment java* yang berbasiskan *open source* yang bekerja seperti halnya *make* pada *C*. *Ant* saat ini sudah mencapai versi 1.6. [MAR06]

Ant bekerja berdasarkan sebuah file project yang umumnya berbentuk file *XML* dengan nama *built.xml*.



Gambar 3.2. Use Case Diagram Server Mobile OPAC

3.2.3.2 Class Diagram

1. Rincian Class Diagram

Dalam sistem *mobile OPAC*, class diagram yang digunakan adalah model *three-tier*. *Three-tier class diagram* membagi sudut pandang menjadi *user service (interface)*, *business service (proses bisnis)* dan *data service*.

Rincian *class-class* yang digunakan dalam *mobile OPAC* sebagai berikut :

a. Rincian Class pada User Service

Berikut, rincian *class* yang digunakan

- a. Class *MIDlet*, digunakan *WebToMobileClientMIDlet*, sebagai *class* utama, disini (*MIDlet*), hanya dirancang untuk satu *class* saja sehingga semua atribut dan operasi tergabung menjadi satu. Untuk lebih jelasnya dapat dilihat pada gambar 3.3.
- b. Class *stub*, digunakan *WebToMobileClient*, sebagai *class client stub*, yaitu *class* yang berfungsi sebagai *interface* (antarmuka) bagi *client (MIDlet)* kepada *server (gateway)*. Untuk lebih jelasnya dapat dilihat pada gambar 3.4.

3.2.3.5 Perancangan Struktur Direktori *File*

1. Perancangan Struktur Direktori *File MIDlet*

Dirancang, terdapat *package referensimetodeclient* yang berisi dua buah *class* sebagai berikut:

- a. *WebToMobileClientMIDlet*, yaitu *class MIDlet mobile OPAC*, sebagai presentasi kepada *user* agar dapat menggunakan *MIDlet mobile OPAC*.
- b. *WebToMobileClient*, yaitu *class* yang berfungsi sebagai *stub* yang berada pada sisi *client*.

2. Perancangan Struktur Direktori *File Server Utama atau Gateway*

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

a. *Package referensimetodeclient*

Berisi dua buah *class* yaitu :

- a. *Class ReferensiMetode*, yaitu *class* untuk melempar parameter penelusuran sederhana atau *advance* dari *client* ke *server-server* perpustakaan UII atau UGM.
- b. *Class WebToMobileServlet*, yaitu *class* yang berfungsi sebagai *stub* yang berada pada sisi *server*.

b. *Package Libraries*

Berfungsi sebagai *package libraies* yang digunakan oleh *server* utama atau *gateway*.