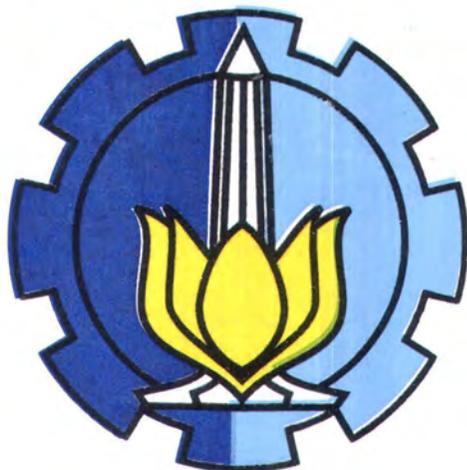


15.714/H/02



**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
LAYANAN INFORMASI BERBASIS WAP DI JURUSAN
TEKNIK INFORMATIKA ITS**

TUGAS AKHIR

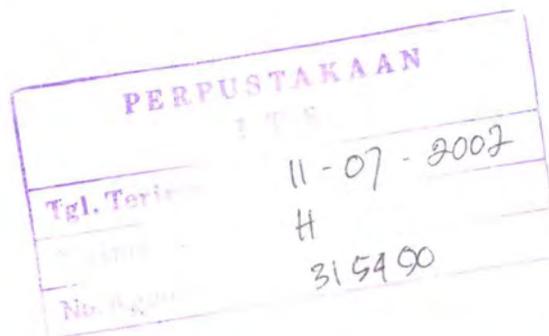


RSIF
005.1
Daw
β-1
2002

OLEH :

I KOMANG YUN DARMAWAN
5197.100.039

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2002**



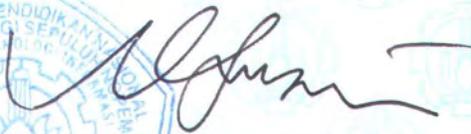
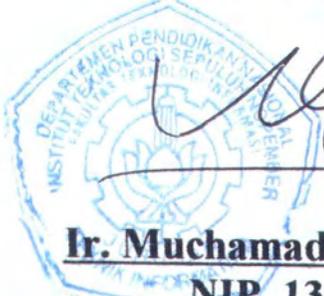
**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
LAYANAN INFORMASI BERBASIS WAP DI JURUSAN
TEKNIK INFORMATIKA ITS**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember surabaya**

Mengetahui /Menyetujui,

Dosen Pembimbing

Ir. Muchamad Husni, M.Kom.
NIP. 131.141.100

**SURABAYA
April 2002**

ABSTRAK

Telepon seluler saat ini semakin banyak yang memiliki fasilitas WAP (*Wireless Application Protocol*), tetapi para pemiliknya jarang sekali memanfaatkan fasilitas tersebut. Hal ini diakibatkan oleh sedikitnya ketersediaan layanan yang benar-benar bermanfaat. Hal ini berlaku juga pada kalangan mahasiswa dan dosen di Jurusan.

Dalam Tugas Akhir ini dibuat sebuah aplikasi berbasis WAP yang menyediakan informasi-informasi penting bagi pengguna khususnya mahasiswa dan dosen. Aplikasi ini memungkinkan pengguna untuk mengakses informasi yang diperlukan sewaktu-waktu dari mana saja. Juga dilengkapi dengan aplikasi yang berbasis *web* untuk memudahkan pengelolaan informasi yang ditampilkan.

Aplikasi telah di ujicoba dengan menggunakan *web server* Apache, LDAP sebagai penampung data dan diletakkan pada lingkungan dengan menggunakan sistem operasi Linux. Untuk klien digunakan ponsel *Siemens M35i* serta beberapa *emulator* untuk perbandingan. Dari ujicoba ini didapatkan hasil yang memuaskan, yang ditunjukkan oleh berjalannya semua fungsi dari aplikasi serta waktu akses yang cepat.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadapan Tuhan Yang Maha Kuasa, karena atas perkenan-Nya lah maka penulis dapat menyelesaikan semua rangkaian pengerjaan Tugas Akhir ini. Laporan Tugas Akhir dengan judul :

Perancangan dan Pembuatan Perangkat Lunak Layanan

Informasi Berbasis Wap di Jurusan Teknik Informatika ITS

ini disusun sebagai penutup rangkaian pelaksanaan Tugas Akhir yang merupakan kewajiban bagi setiap mahasiswa dalam menyelesaikan program strata satu (S1) di Jurusan Teknik Informatika, ITS.

Dalam kesempatan ini penulis ingin mengucapkan terima kasih dan penghargaan yang sebesar-besarnya bagi semua pihak yang telah membantu dalam pelaksanaan Tugas Akhir ini baik secara langsung maupun tidak langsung. Secara khusus penulis mengucapkan terima kasih kepada :

1. Bapak Muchamad Husni selaku Dosen Pembimbing sekaligus Dosen Wali yang telah membimbing penulis mulai dari awal perkuliahan sampai pelaksanaan tugas akhir ini selesai.
2. Bapak Agus Sulistya selaku pembimbing kerja praktek yang memberikan ide awal dari tugas akhir ini.
3. Bapak dan Ibu, orang tua penulis yang selalu memberikan dorongan, nasehat dan kasih sayangnya.
4. Kakak penulis atas kasih sayang dan kerelaannya *melimpahkan* "barang-barang berharga" miliknya kepada penulis.
5. Tita Karlita atas perhatian, saran-saran, dorongan dan keceriaan serta semua "warna" yang telah diberikannya

6. Sepupu-sepupuku yang lucu, yang telah memberikan keceriaan
7. Nyomen (komputer dan kamarnya yang sering penulis pinjam), Devi (komputer dan masukan-masukannya secara teknis), Dees, Gupta, Edi, Andri dan Meonx atas kebersamaannya selama ini dan menyediakan suasana rumah yang kondusif, Dwi yang menjadi teman curhat dan bercanda, Soe yang menjadi teman seperjuangan di awal-awal kuliah (moga cepet lulus) dan Gung Gus untuk modemnya.
8. Yanti, Rina, Trisna, dan Sapphire atas dorongan, kebersamaan dan segala bantuan yang diberikan selama ini
9. Mastra atas semua saran dan semangatnya
10. Nonik yang telah banyak memberikan pandangan dan mau berbagi untuk berbagai hal
11. Guruh (yg telah membantu pada saat persiapan sidang), Wiro, Zola, Gershom, Damen, Andi, Nova, Wiwik, Mike, Novan dan semua teman-teman COD atas kebersamaannya
12. Dodi & Gusgung atas persahabatannya sejak SMA
13. Mas Suyudi dan Mas Rudi sebagai admin di lab komputing, Mas Anank, Mas Cahyono dan Mas Puji yang menjadi teman begadang di lab Komputing
14. Mas Yudi dan semua staf pengajaran jurusan
15. Indosat – M3 untuk kartunya yang “smart”
16. Blizzard atas Diablo I & II serta Starcraft, Microsoft untuk Age of Empires dan AOE II:Age of Kings, Eidos untuk Revenant, Fate of The Dragon, Commandos I & II (Pyro Studios), Electronic Arts atas Superbike, FIFA dan Need for Speednya, Liquid Entertainment untuk Battle Realms,

Magnetic Fields untuk Rally Championship 2000, yang semuanya telah menemani penulis dalam mengisi waktu luang dan menghilangkan kejenuhan.

17. Bob Marley, President of The United States of America, RHCP, Pearl Jam, Metallica, Nirvana, The Police, Sting, Phil Collins, Gigi, Jamrud, Boomerang, Dewa, Padi atas musik/lagunya yang selalu menemani penulis dari awal perkuliahan.

Akhir kata, besar harapan penulis Buku Tugas Akhir ini dapat memberikan manfaat bagi kita semua.

Surabaya, April 2002

Penulis

DAFTAR ISI

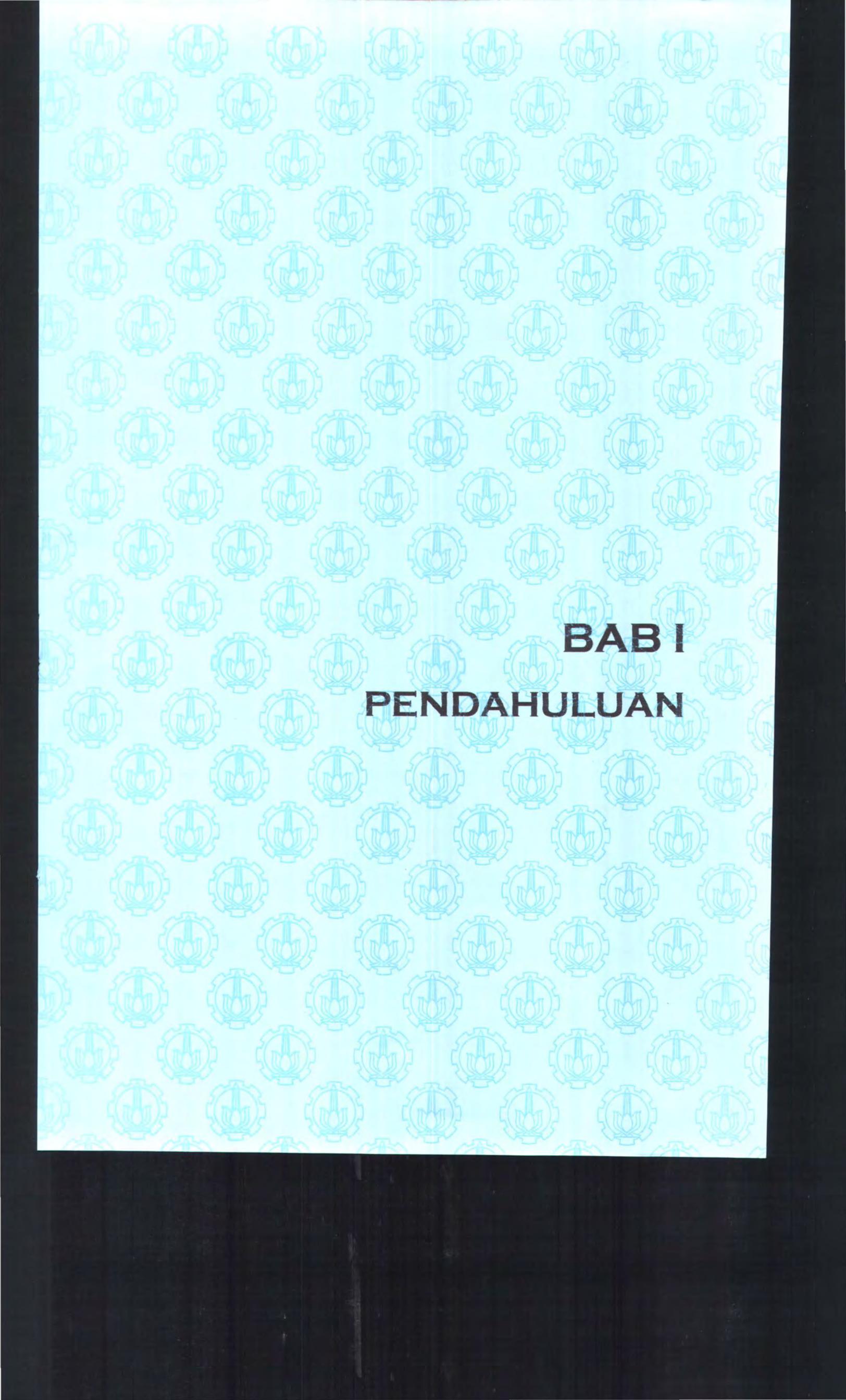
ABSTRAK	<i>i</i>
KATA PENGANTAR	<i>ii</i>
DAFTAR ISI	<i>v</i>
DAFTAR GAMBAR	<i>vii</i>
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERMASALAHAN	2
1.3 TUJUAN	3
1.4 RUANG LINGKUP PERMASALAHAN	3
1.5 METODOLOGI DAN SISTEMATIKA PEMBAHASAN	3
BAB II WIRELESS APPLICATION PROTOCOL (WAP)	6
2.1 PERBANDINGAN MODEL WAP DAN MODEL WWW	6
2.2 KOMPONEN MODEL WAP	8
2.2.1 WAP Client	8
2.2.2 WAP Gateway	10
2.2.3 WAP Server	10
2.3 HUBUNGAN CLIENT, GATEWAY DAN SERVER	11
2.4 ARSITEKTUR WAP	12
2.4.1 Wireless Application Environment (WAE)	13
2.4.2 Wireless Session Protocol (WSP)	13
2.4.3 Wireless Transaction Protocol (WTP)	14
2.4.4 Wireless Transport Layer Security (WTLS)	14
2.4.5 Wireless Datagram Protocol (WDP)	15
2.5 WIRELESS MARKUP LANGUAGE (WML)	16
BAB III LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP).	18
3.1 CARA KERJA LDAP	19
3.2 LDAP BACKEND, OBJECT DAN ATTRIBUTE	19
3.3 FILE-FILE KONFIGURASI	22
3.3.1 File 'slapd.conf'	22
3.3.2 File 'ldap.conf'	24
3.3.3 File 'slapd.oc.conf'	25
3.3.4 File 'slapd.at.conf'	26
3.4 INTERAKSI DENGAN DATA	26
3.4.1 Penambahan	27

3.4.2 Pencarian	27
3.4.3 Modifikasi	28
3.4.4 Penghapusan	28
BAB IV PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK..	29
4.1 PERANCANGAN PERANGKAT LUNAK	29
4.1.1 Arsitektur	29
4.1.2 Basisdata - LDAP	31
4.1.3 Desain Antarmuka	34
4.1.3.1 WAP Based	34
4.1.3.2 Web Based	36
4.1.4 Proses	37
4.1.4.1 WAP Based	37
4.1.4.2 Web Based	39
4.2 PEMBUATAN PERANGKAT LUNAK	40
4.2.1 LDAP	40
4.2.1.1 <i>Objectclass</i> Dan Pembentukan Direktori	41
4.2.1.3 Koneksi	43
4.2.2 WAP Based	44
4.2.2.1 Link	44
4.2.2.2 Masukan	45
4.2.2.3 Pencarian	46
4.2.2.4 Tampilan	50
4.2.2.5 Kondisi Server dan Jaringan	51
4.2.3 Web Based	52
4.2.3.1 Pencarian dan Penampilan Data pada Form Edit dan Hapus	53
4.2.3.2 Penambahan Data	54
4.2.3.3 Modifikasi Data	55
4.2.3.4 Penghapusan Data	56
BAB V UJICOBA DAN ANALISA	57
5.1 LINGKUNGAN UJI COBA	57
5.2 KONFIGURASI SERVER	57
5.3 KONFIGURASI SIEMENS M35I	58
5.4 UJI COBA APLIKASI	59
5.4.1 Uji Coba Dengan Menggunakan Emulator Dan Analisisanya	60
5.4.2 Uji Coba Menggunakan Ponsel Siemens M35i Dan Analisisanya	68
BAB VI KESIMPULAN DAN SARAN	69
6.1 KESIMPULAN	69
6.2 SARAN	69
DAFTAR PUSTAKA	71

DAFTAR GAMBAR

Gambar 2.1 Contoh Jaringan WAP	7
Gambar 2.2 Model WWW	8
Gambar 2.3 Model WAP	8
Gambar 2.4 WAP Client	9
Gambar 2.5 WAP Client, Gateway dan Server	11
Gambar 2.6 Arsitektur WAP	12
Gambar 3.1 Interaksi LDAP	19
Gambar 3.2 Interaksi LDAP, dengan referral	19
Gambar 4.1 Tes pebandingan LDAP dan basisdata relasional	30
Gambar 4.2 Model arsitektur perangkat lunak	31
Gambar 4.3 Struktur basisdata	32
Gambar 4.4 Urutan model direktori	32
Gambar 4.5 Atribut dari tiap tipe informasi	33
Gambar 4.6 <i>Value</i> untuk suatu atribut	34
Gambar 4.7 Struktur Halaman WAP	35
Gambar 4.8 <i>Objectclass</i>	41
Gambar 4.9 File LDIF, Pembentukan Direktori	41
Gambar 4.10 <i>Script</i> konversi CSV ke LDIF	42
Gambar 4.11 Potongan script penulisan file LDIF hasil konversi	43
Gambar 4.12 Koneksi dengan Autentikasi	44
Gambar 4.13 Link pada WAP Page	44
Gambar 4.14 Masukan pada halaman Wap	45
Gambar 4.15 Masukan untuk <i>filter</i> dikosongkan	46
Gambar 4.16 Pengisian <i>filter</i> dari <i>string</i> masukan	47
Gambar 4.17 Hasil lebih dari tiga <i>record</i>	47
Gambar 4.18 Pencarian semua record, tampil tiga sekaligus	48
Gambar 4.19 Pencarian 1-3 <i>record</i>	49
Gambar 4.20 Pencarian data tunggal	50
Gambar 4.21 Menampilkan Informasi	50
Gambar 4.22 Menampilkan <i>link</i> untuk <i>softkey</i>	51
Gambar 4.23 Pengecekan Port	52
Gambar 4.24 Mengecek Kondisi Jaringan	52
Gambar 4.25 Form Login	53
Gambar 4.26 Pencarian, mengambil dn	54
Gambar 4.27 Penambahan data	54
Gambar 4.28 Modifikasi Data	55
Gambar 4.29 Penghapusan data	56
Gambar 5.1 Penambahan <i>Type</i> untuk WML	58
Gambar 5.2 Menu Awal, <i>Emulator</i>	61
Gambar 5.3 Masukan	62
Gambar 5.4 Hasil Pencarian	63

Gambar 5.5 Konfirmasi dan Akhir Pencarian	64
Gambar 5.6 Status Server dan jaringan	65
Gambar 5.7 Informasi, CCWap Browser	66
Gambar 5.8 Informasi, UP.SDK	66
Gambar 5.9 <i>Debug Info</i> , CCWap Browser	67



BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Telepon seluler saat ini semakin banyak yang memiliki fasilitas WAP, tetapi para pemiliknya jarang sekali memanfaatkan fasilitas tersebut. Hal ini diakibatkan oleh sedikitnya ketersediaan layanan yang benar-benar bermanfaat. Hal ini berlaku juga pada kalangan mahasiswa/i di Jurusan.

Walaupun telah ada sebuah contoh aplikasi yang dibuat yang berhubungan dengan kegiatan kemahasiswaan, dalam hal ini perwalian, namun hal ini masih sangat kurang karena tidak dapat memberikan layanan yang maksimal bagi mahasiswa. Aplikasi ini hanya berguna pada saat proses perwalian yang mungkin sebenarnya tidak perlu di-*mobile*-kan karena jadwalnya yang sudah pasti selama satu minggu dan tetap harus berkonsultasi langsung ke dosen wali. Lain halnya dengan aplikasi yang akan dibuat ini, yang bersifat menyediakan informasi yang berhubungan dengan keseharian mahasiswa dan dosen, yang dapat diperlukan sewaktu-waktu.

Untuk keperluan akademis, baik mahasiswa maupun dosen diperlukan suatu layanan baik jadwal kuliah, ujian maupun kegiatan lainnya yang dapat diakses kapan saja dan dimana saja.

Selain itu, untuk dapat menjaga komunikasi antar mahasiswa maupun dengan dosen, layanan sejenis buku telepon elektronik sangat dibutuhkan. Dengan adanya layanan ini, setiap mahasiswa dapat dengan mudah mengetahui



data no telepon, alamat serta email dari dosen atau mahasiswa lainnya yang hendak dihubungi.

Bagi mahasiswa yang telah atau hampir lulus akan jarang sekali berada di sekitar kampus. Padahal lowongan-lowongan yang ditawarkan oleh suatu perusahaan biasanya diserahkan ke Jurusan untuk diumumkan. Dengan adanya layanan berbasis WAP yang menampilkan lowongan-lowongan tadi, mahasiswa dapat dengan mudah mengetahuinya dan selanjutnya dapat melihat data selengkapnya di Jurusan.

Mahasiswa ingin mengetahui informasi tentang beasiswa yang ditawarkan sesegera mungkin, sehingga dapat dipersiapkan segala persyaratan yang diperlukan dengan cepat.

Seorang administrator jaringan, yang biasanya seorang mahasiswa ataupun dosen, yang memiliki kesibukan sendiri, sewaktu-waktu perlu memeriksa kondisi jaringannya, baik *daemon-daemon* yang berjalan di server maupun hubungan dengan laboratorium-laboratorium lainnya ataupun kondisi hubungan dengan dunia luar/internet, sehingga jika terjadi suatu kesalahan dapat dilakukan tindakan yang tepat.

1.2 PERMASALAHAN

Permasalahan yang dihadapi adalah bagaimana memberikan layanan yang dapat diakses oleh setiap pengguna handphone dengan fasilitas WAP dalam kaitannya dengan penggunaan LDAP sebagai penyimpan data.

1.3 TUJUAN

Tujuan pembuatan Tugas Akhir ini adalah merancang dan membuat perangkat lunak yang bertujuan untuk memudahkan mahasiswa dan dosen dalam kegiatan akademis sehari-hari. Dan juga bagi seorang administrator jaringan untuk dapat terus mengawasi kinerja jaringannya.

1.4 RUANG LINGKUP PERMASALAHAN

Dalam penulisan tugas akhir ini difokuskan pada permasalahan dengan ruang lingkup sebagai berikut :

- 'database' hanya untuk tingkat S1 reguler dan dosen
- tidak memfokuskan pada keamanan aplikasi web maupun wap
- tidak memfokuskan pada aplikasi web
- tidak memfokuskan pada keamanan data pada server
- penambahan / edit data hanya bisa dilakukan oleh administrator melalui web ataupun *shell* Linux
- aplikasi bisa diakses / view oleh siapa saja

1.5 METODOLOGI DAN SISTEMATIKA PEMBAHASAN

Metodologi yang digunakan dalam penyusunan Tugas Akhir ini adalah :

1. Studi Literatur

Studi literatur melalui buku-buku penunjang baik cetak maupun elektronik mengenai WAP, WML, LDAP dan PHP, serta contoh-contoh *code* yang tersedia di internet

2. Perancangan Perangkat Lunak

Perancangan perangkat lunak ini meliputi pembuatan struktur dari direktori yang digunakan untuk menyimpan data serta desain antar muka pengguna.

3. Pembuatan Perangkat Lunak

Meliputi proses pemrograman perangkat lunak serta implementasinya pada server yang dapat diakses dari luar (memiliki IP *Visible*).

4. Uji Coba dan Evaluasi

Uji coba jalannya perangkat lunak secara keseluruhan serta melakukan evaluasi.

Sistematika pembahasan dalam Tugas Akhir ini adalah sebagai berikut :

Bab I Pendahuluan

Pada Bab I ini dibahas mengenai latar belakang pemilihan judul Tugas Akhir, permasalahan yang dihadapi dalam penyelesaian, tujuan, serta metodologi dan sistematika pembahasan.

Bab II *Wireless Application Protocol*

Dalam Bab II ini membahas mengenai teori-teori yang berhubungan dengan WAP serta lingkungannya..

Bab III *Lightweight Directory Access Protocol*

Dalam Bab III ini membahas mengenai teori-teori yang berhubungan dengan LDAP yang digunakan sebagai penyimpan data.

Bab IV Perancangan dan Pembuatan Perangkat Lunak

Pada Bab IV ini dibahas mengenai tahap-tahap proses perancangan dan implementasinya.

Bab V Uji Coba dan Evaluasi

BAB II

WIRELESS APPLICATION PROTOCOL (WAP)

Wireless Application Protocol (WAP) merupakan standar industri yang mendefinisikan protokol komunikasi dan lingkungan aplikasi sehingga memungkinkan sebuah ponsel ataupun alat lain yang memiliki micro browser untuk mengakses internet. WAP didesain untuk piranti-piranti kecil, seperti telepon genggam dan *Personal Digital Assistant(PDA)* yang memiliki banyak keterbatasan, diantaranya [WAPARCH] :

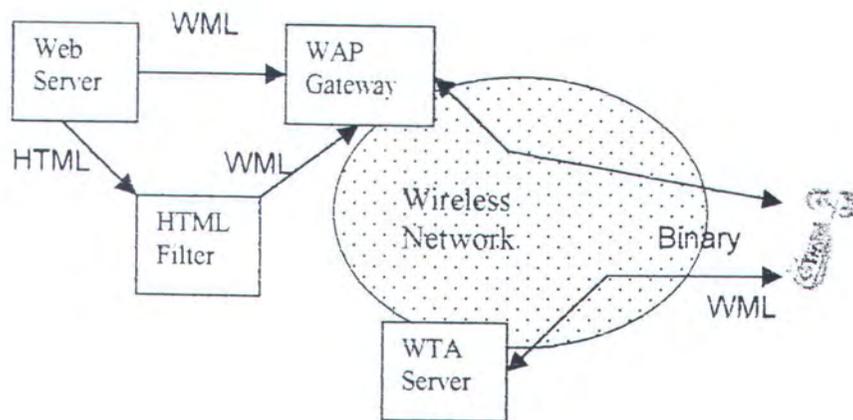
- Layar kecil
- Fasilitas untuk masukan *pemakai* terbatas
- Koneksi jaringan dengan *bandwith sempit*
- Memori dan sumber komputasi yang terbatas

WAP diturunkan dari standar Internet (WWW), sehingga sebagian besar memiliki kesamaan.

2.1 PERBANDINGAN MODEL WAP DAN MODEL WWW

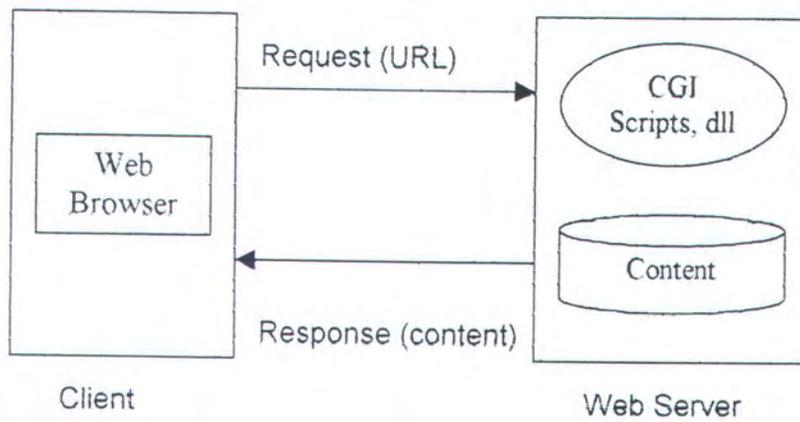
Model WAP didesain dengan mengambil ide dari Model World Wide Web (WWW). Karena mengambil ide dari model WWW, maka otomatis model WAP juga menganut aliran client-server seperti yang ada pada WWW. Perbedaan utama antara WWW dan WAP adalah adanya WAP Gateway yang berfungsi sebagai penterjemah. Gambar 2.1 memberikan gambaran mengenai sebuah

contoh jaringan WAP. Gambar 2.2 dan 2.3 memberikan gambaran mengenai perbedaan antara model WAP dan WWW.

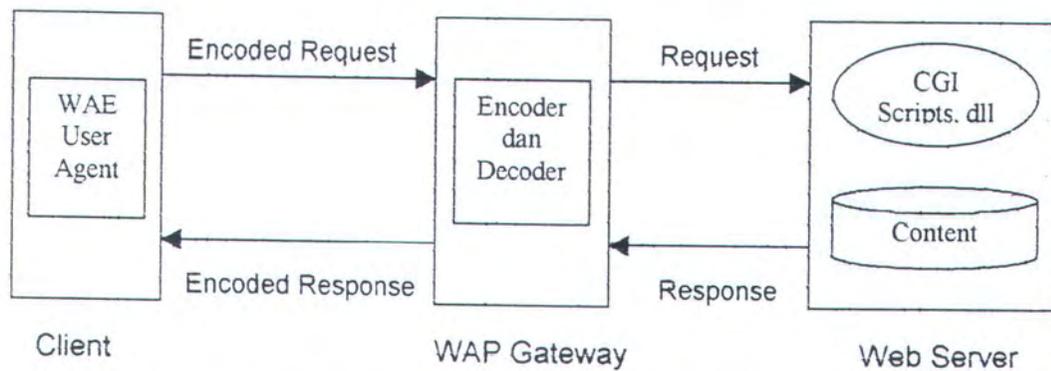


Gambar 2.1 Contoh Jaringan WAP [WAPARCH]

Pada gambar 2.2 terlihat bahwa pada model WWW, client meminta data dengan format standar dan kemudian server akan menanggapi permintaan tersebut dengan mengirim data dalam format standar pula. Sedangkan pada model WAP (gambar 2.3) untuk mengakses aplikasi yang ada di server, client harus terkoneksi dulu dengan suatu WAP Gateway dan mengirim permintaan yang terkodekan akan suatu data. Di WAP gateway, data tersebut didekodekan menjadi data dalam format standar sehingga dapat dimengerti oleh server. Selanjutnya prosesnya sama dengan yang terjadi pada model WWW. WAP gateway akan mengkodekan lagi data dari server baru kemudian dikirimkan ke klien. Hal ini akan dibahas secara lebih detail pada bagian lain.



Gambar 2.2 Model WWW [WAPARCH]



Gambar 2.3 Model WAP [WAPARCH]

2.2 KOMPONEN MODEL WAP

Pada gambar 2.3 terlihat bahwa model Wap terdiri atas tiga komponen pokok yaitu *WAP Client*, *WAP Gateway*, dan *WAP Server*. Komponen-komponen dalam model WAP tersebut akan dibahas berikut ini.

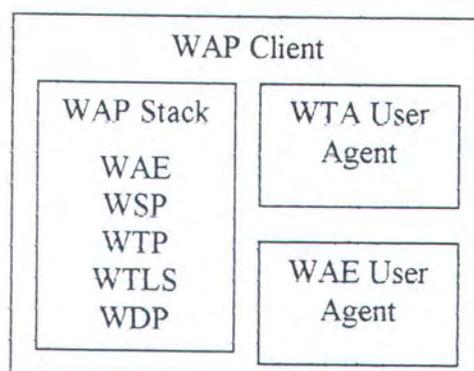
2.2.1 WAP Client

Spesifikasi WAP memberi kebebasan kepada perusahaan pembuat alat (telepon genggam) untuk mengembangkan perangkat lunak yang sesuai dengan produk masing-masing. Tidak ada keharusan untuk menggunakan bentuk

tertentu atau harus menggunakan tampilan tertentu bahkan cara menyajikan isi dan cara navigasi pun bebas ditentukan oleh perusahaan yang bersangkutan.

Hanya satu syarat yang harus dipenuhi agar suatu alat bisa dikatakan sesuai dengan standar WAP, yaitu alat tersebut harus mengimplementasikan *Wireless Application Environment (WAE) User Agent*, *Wireless Telephony Applications (WTA) User Agent* dan *WAP Stack* (gambar 2.4).

WAE User Agent adalah *micro browser* dimana isi dari internet ditampilkan. *Micro browser* ini bertugas untuk menerima *Wireless Markup Language (WML)*, *WMLScript* dan *Wireless Bitmap (WBMP)* [WAPWAE].



Gambar 2.4 WAP Client

WTA User Agent bertugas untuk menerima file-file WTA dari WTA server kemudian menjalankannya. Dalam *WTA User Agent* sudah termasuk fungsi untuk pengorganisasian buku telepon, identifikasi pemanggil dan pemanggilan kepada nomor tertentu.

WAP Stack berguna bagi telepon genggam agar dapat terhubung ke internet dengan menggunakan protokol WAP. *WAP stack* banyak berhubungan dengan Arsitektur WAP yang akan dibahas pada bagian lain.

2.2.2 WAP Gateway

Wireless Application Protocol Gateway menjadi sebuah jembatan antara dunia internet dan dunia nirkabel. Sesungguhnya WAP Gateway adalah sebuah proxy yang mempunyai dua fungsi utama, yaitu sebagai protokol gateway dan sebagai CODEC (Coder/Decoder).

Sebagai gateway, WAP Gateway harus mendukung protokol WAP dan protokol IP. Selain itu WAP gateway harus dapat mengkonversi permintaan yang dikirim oleh klien yang menggunakan *Wireless Session Protokol (WSP)* menjadi protokol yang digunakan oleh server (misal HTTP) dan begitu juga sebaliknya.

Kemampuan CODEC berguna untuk menkonversi WML dan WMLScript menjadi bentuk yang lebih optimal sehingga cocok untuk dikirim melalui bandwidth yang kecil. Selain itu fungsi CODEC juga untuk mengkonversi HTML menjadi WML. Disamping kemampuan-kemampuan diatas beberapa gateway juga memiliki kemampuan *security*.

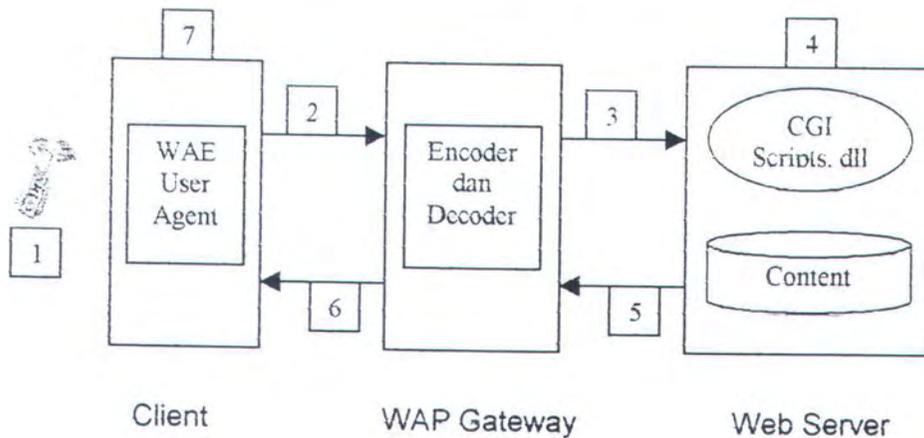
2.2.3 WAP Server

Sama halnya dengan WWW pada internet yang memerlukan server untuk menyimpan isi web, WAP juga memerlukan server. Secara fisik server WWW dan server WAP adalah sama, juga software yang digunakan adalah sama. Yang membedakan antara keduanya adalah isinya. Jadi dapat disimpulkan bahwa server WWW dapat digunakan sebagai server WAP dan pemrograman server side pada WWW juga dapat digunakan pada WAP.



2.3 HUBUNGAN CLIENT, GATEWAY DAN SERVER

Secara skematis hal-hal yang terjadi pada saat koneksi antara client dengan server dapat dilihat seperti pada gambar berikut.



Gambar 2.5 WAP Client, Gateway dan Server

Mengacu pada gambar diatas, urutan peristiwa yang terjadi pada saat koneksi antara client dengan server adalah sebagai berikut :

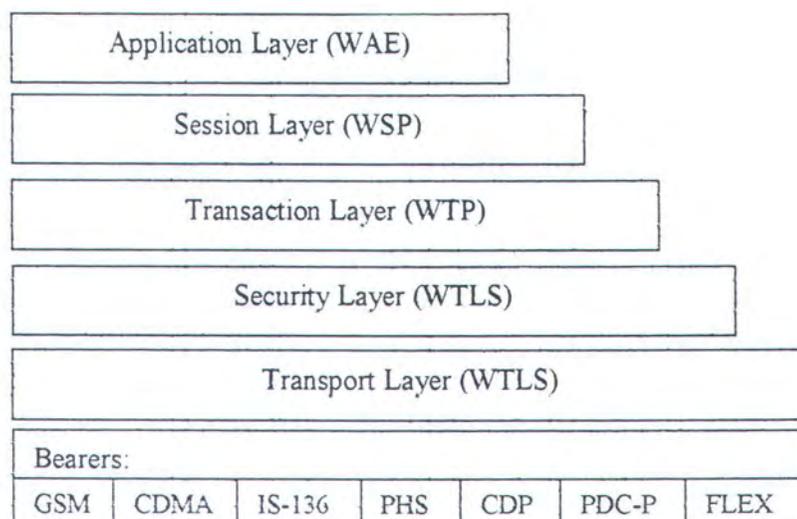
- (1) User memasukkan alamat URL
- (2) Sebuah permintaan WSP (WSP Request) dikirimkan ke WAP gateway dalam bentuk terkodekan.
- (3) Di WAP Gateway, pesan tersebut didekodekan dan permintaan WSP itu diubah menjadi permintaan HTTP (HTTP/HTTPS Request). Nama domain juga diubah menjadi alamat IP.
- (4) Tergantung dari URL, server mengirimkan isinya, bisa berupa WML, HTML, file gambar atau menjalankan file-file CGI/ASP/PHP.
- (5) Server mengirimkan HTTP/HTTPS response kembali ke WAP Gateway bersama dengan isi yang diminta.
- (6) Setelah sampai di WAP gateway, HTTP header diubah menjadi WML header dan juga mungkin dilakukan kompilasi WMLScript jika diperlukan.

Di WAP Gateway juga dapat dilakukan konversi gambar dari JPG/GIF menjadi WBMP. Kemudian sebuah WSP response dikirim bersama dengan isi yang sudah dikodekan kepada client.

- (7) Client menerima WSP response dan isi yang sudah dikodekan. Akhirnya microbrowser yang terdapat pada client menampilkan isinya kepada user.

2.4 ARSITEKTUR WAP

[WAPARCH] Arsitektur WAP menyediakan lingkungan yang luas bagi pengembangan aplikasi untuk peralatan komunikasi yang bersifat mobile. Hal ini dicapai dengan pembuatan desain yang berlapis-lapis.



Gambar 2.6 Arsitektur WAP

Setiap layer bisa diakses oleh layer di atasnya, begitu juga oleh layanan dan aplikasi yang lain.

Arsitektur WAP yang berlapis-lapis memungkinkan layanan dan aplikasi lain untuk memanfaatkan fitur-fitur yang tersedia dengan baik. Aplikasi eksternal

dapat mengakses *session*, *transaction*, *security* dan *transport layer* secara langsung. Bagian berikut memberikan penjelasan dari berbagai layer diatas.

2.4.1 Wireless Application Environment (WAE)

WAE merupakan lingkungan aplikasi yang bersifat umum berdasarkan pada kombinasi antara teknologi *WWW* dan *Mobile Telephony*. Tujuan utama dari WAE adalah untuk membentuk lingkungan yang memungkinkan operator dan penyedia layanan untuk membangun aplikasi dan layanan yang bisa mencakup berbagai *wireless platform* secara efisien. Dalam WAE ditentukan sebuah lingkungan *micro-browser* yang memiliki kompatibilitas dengan *WML*, *WMLScript*, *WTA*, dan *Content Format*.

2.4.2 Wireless Session Protocol (WSP)

Wireless Session Protocol (WSP) menyediakan layer aplikasi *WAP* dengan antarmuka yang konsisten untuk dua *session service*. Yang pertama adalah layanan *connection-oriented* yang bekerja diatas protokol *transaction layer WTP*. Yang kedua adalah sebuah layanan *connectionless* yang bekerja pada sebuah layanan datagram (*Wireless Datagram Protocol*) aman ataupun tidak (*WDP*).

WSP terdiri dari layanan yang sesuai untuk menjelajahi aplikasi dengan fungsi-fungsi berikut : kemampuan dan *semantic HTTP/1.1* yang mampu dikodekan secara real time, *session* dengan waktu hidup panjang (*Long-lived session state*), memberhentikan sementara (*suspend*) dan mengembalikan (*resume*) dengan kemampuan perpindahan antar *session (migration)*, negosiasi fitur protocol.

WSP didesain untuk memungkinkan sebuah WAP Gateway untuk menghubungkan sebuah klien WSP untuk berkomunikasi dengan sebuah server standar HTTP.

2.4.3 Wireless Transaction Protocol (WTP)

Wireless Transaction Protocol (WTP) berjalan diatas sebuah layanan datagram dan bertindak sebagai sebuah protokol yang *light-weight transaction-oriented* yang sesuai untuk di implementasikan pada "*thin*" client. WTP beroperasi secara efisien melalui jaringan datagram nirkabel, aman maupun tidak dan menyediakan fitur-fitur berikut :

- Tiga macam kelas layanan :
 - Unreliable one-way request,
 - Reliable one-way request,
 - Reliable two-way request-reply transaction;
- Optional user-to-user reliability - WTP user memicu konfirmasi dari setiap *message* yang diterima
- Optional out-of-band data on acknowledgements
- PDU concatenation dan delayed acknowledgement untuk mengurangi jumlah *message* yang dikirimkan
- Transaksi asynchronous.

2.4.4 Wireless Transport Layer Security (WTLS)

WTLS adalah protokol keamanan yang didasarkan pada protokol Transport Layer Security (TLS), yang sebelumnya dikenal dengan *Secure Socket Layer* (SSL). WTLS dimaksudkan untuk bekerja dengan WAP Transport protocol

dan telah dioptimalkan untuk digunakan pada kanal komunikasi *narrow-band*.

WTLS menyediakan fitur-fitur berikut :

- *Data Integrity* – memastikan bahwa data yang dikirim antara terminal dan server aplikasi tidak berubah atau terkorupsi.
- *Privacy* – memastikan data yang ditransmisikan bersifat pribadi/rahasia, tidak dapat dimengerti oleh siapapun yang mencegat pengiriman data tersebut.
- *Authentication* – memperlihatkan keotentikan dari terminal dan server aplikasi.
- *Denial-of-service protection* – mendeteksi dan menolak data yang tidak berhasil diverifikasi sehingga dapat lebih melindungi layer di atasnya.

Aplikasi dapat secara selektif menentukan apakah fasilitas WTLS akan dihidupkan/digunakan atau tidak sesuai dengan tingkat keamanan yang diperlukan dan karakteristik dari jaringan (misalnya, *privacy* dapat dimatikan pada jaringan yang sudah menyediakan layanan sejenis pada layer dibawahnya).

2.4.5 Wireless Datagram Protocol (WDP)

Transport layer protocol pada arsitektur WAP mengarah pada Wireless Datagram Protocol (WDP). Layer WDP beroperasi diatas layanan pengantar (*bearer*) yang disediakan oleh berbagai jenis tipe jaringan. Yang dimaksud dengan *bearer* adalah semua komponen yang berhubungan dengan komunikasi antara ponsel dan *base station*. Ada berbagai jenis bearer yang bisa digunakan, antara lain GSM, CDMA, SMS, CSD, USSD, dsb. Sebagai layanan transport umum, WDP menawarkan layanan yang konsisten kepada layer-layer di atasnya dan berkomunikasi secara transparan melalui salah satu layanan pengantar yang tersedia.

WDP menyediakan sebuah antarmuka yang telah umum kepada layer di atasnya, *Security*, *Session* dan *Application* yang dapat berfungsi secara bebas tanpa tergantung pada jaringan wireless yang mendasarinya. Dengan menjaga interface dan fitur dasar dari transport layer tetap konsisten, *global interoperability* dapat dicapai dengan memanfaatkan gateway perantara.

2.5 WIRELESS MARKUP LANGUAGE (WML)

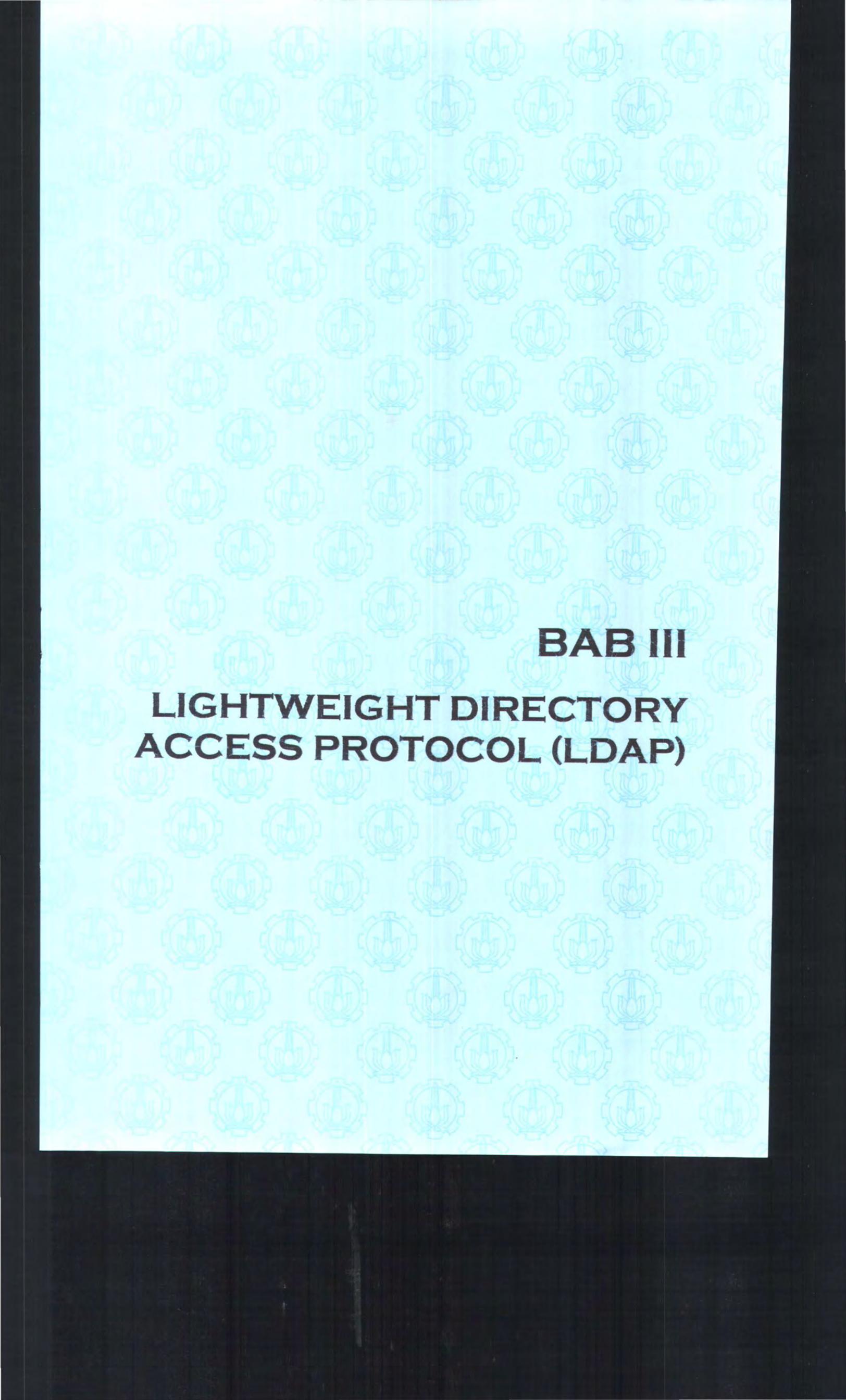
[WAPWAE] WML adalah sebuah *tag-based document language*. WML merupakan hasil perpaduan antara HTML dengan Handheld Device Markup Language (HDML). WML termasuk dalam kategori tipe dokumen XML. WML telah di optimalkan untuk memberikan tampilan antar muka dan interaksi dengan user pada kemampuan peralatan yang terbatas, seperti telepon dan *wireless mobile terminal* lainnya.

WML dan lingkungan yang mendukungnya telah di desain dengan mempertimbangkan *narrow-band device* tertentu termasuk diantaranya layar yang kecil, fasilitas masukan bagi user yang terbatas, koneksi jaringan berpita sempit, keterbatasan memori dan sumber daya komputasi yang juga terbatas.

WML menggunakan istilah *card* dan *deck*. WML mengandung sebuah konstruksi yang mengijinkan dokumen, dalam suatu aplikasi, untuk terbentuk oleh lebih dari satu *card*. Sebuah interaksi dengan user digambarkan dalam sekumpulan *card*, yang bisa dikelompokkan dalam satu dokumen (biasanya disebut sebuah *deck*). Secara logika, seorang user akan bernavigasi melewati sekumpulan WML *card*. Intruksi yang terdapat dalam sebuah *card* bisa memanggil sebuah layanan dari *origin server* sesuai dengan yang dibutuhkan oleh interaksi tersebut. *Deck* diambil dari *origin server* sesuai kebutuhan. WML

deck bisa disimpan sebagai file statis pada sebuah *origin server*, atau bisa juga secara dinamis dibangkitkan oleh sebuah *content generator* yang berjalan pada sebuah *origin server*. Setiap *card*, dalam sebuah *deck*, mengandung sebuah spesifikasi untuk interaksi tertentu dengan user. WML memiliki berbagai *feature*, termasuk diantaranya :

- Dukungan terhadap Teks dan Gambar.
- Dukungan terhadap Masukan User.
- Navigasi dan tumpukan *History*.
- Dukungan karakter internasional.
- *MMI Independence*.
- *Narrow-band Optimisation*.



BAB III
LIGHTWEIGHT DIRECTORY
ACCESS PROTOCOL (LDAP)

BAB III

LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)

[LDAPHOW] LDAP merupakan sebuah protokol *client-server* untuk mengakses sebuah layanan direktori (*directory service*). Sebuah direktori terlihat seperti sebuah database, akan tetapi informasi yang ada pada sebuah direktori secara umum akan lebih banyak dibaca daripada ditulis. Sebagai konsekuensinya, direktori biasanya tidak menerapkan penanganan transaksi yang kompleks ataupun skema *roll-back* yang biasa digunakan pada database standar untuk melaksanakan update yang kompleks dan dalam jumlah yang banyak.

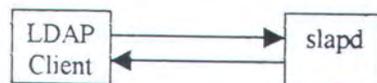
Direktori dirancang untuk dapat memberikan respon yang cepat untuk menerima operasi *lookup* atau *search* dalam jumlah yang banyak. Direktori memiliki kemampuan untuk mereplikasi informasi dalam rangka meningkatkan ketersediaan dan kehandalan, dilain pihak, juga sekaligus mengurangi waktu respon. Ketika informasi pada sebuah direktori direplikasi, inkonsistensi antara replikasi masih bisa diterima, asalkan akhirnya akan *sincron* lagi.

Ada berbagai cara untuk menyediakan layanan direktori. Metode yang berbeda mengijinkan informasi yang dapat disimpan pada direktori tersebut dalam bentuk yang berbeda, bagaimana informasi tersebut bisa direferensi, diquerykan dan di *update*, bagaimana informasi itu dilindungi dari akses yang tidak terotorisasi dan lain-lain. Beberapa layanan direktori adalah lokal, menyediakan layanan terhadap suatu konteks terbatas (seperti, layanan finger

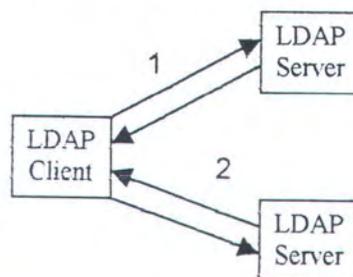
pada sebuah host). Sedangkan layanan yang lain bersifat global, menyediakan layanan terhadap lebih banyak hal.

3.1 CARA KERJA LDAP

Layanan direktori LDAP didasarkan pada model client-server. Satu atau lebih server LDAP berisi data yang membentuk LDAP *directory tree* atau LDAP *backend database*.



Gambar 3.1 Interaksi LDAP [SLAPDUMICH]



Gambar 3.2 Interaksi LDAP, dengan referral [SCAL]

Sebuah klien LDAP menghubungkan diri ke sebuah server LDAP dan memberinya sebuah "pertanyaan". Server memberikan respon berupa jawaban yang diminta (gambar 3.1), atau dengan sebuah penunjuk kemana (server lain / *referral*) klien bisa mendapatkan informasi mengenai pertanyaannya (gambar 3.2).

3.2 LDAP BACKEND, OBJECT DAN ATTRIBUTE

[LDAPHOW] Slapd (*Stand-alone LDAP Daemon*) yang merupakan server dari LDAP, dapat mengandung tiga jenis database backend yang bisa

dipilih. Diantaranya LDBM, *disk-based* database dengan kinerja yang tinggi; SHELL, sebuah database *interface* terhadap perintah UNIX atau *shell script*; dan PASSWD, sebuah database *password* file sederhana.

Dalam tugas akhir ini digunakan jenis backend LDBM. Pemilihan tipe ini dikarenakan oleh kemampuannya yang sesuai dengan yang dibutuhkan.

Untuk dapat melakukan *import* dan *export* informasi direktori antar server direktori LDAP-based, atau untuk sekumpulan perubahan yang akan diterapkan pada sebuah direktori, format filenya dikenal dengan LDIF (*LDAP Data Interchange Format*). Sebuah file LDIF menyimpan informasi dalam bentuk hirarki entry *object-oriented*.

Sebuah file LDIF biasanya berbentuk seperti berikut ini :

```
dn: o=TUdelft, c=NL
o: TUdelft
objectclass: organization

dn: cn=Luiz Malere, o=TUdelft, c=NL
cn: Luiz Malere
sn: Malere
mail: malere@yahoo.com
objectclass: person
```

Sebagaimana terlihat, setiap entry dapat diidentifikasi secara unik dengan *distinguished Name*, atau dn. DN terdiri atas nama dari entri ditambah dengan nama path yang berguna untuk melacak entri tersebut kembali ke tingkat atas dari hirarki direktori tersebut.

Pada LDAP, sebuah *object class* mendefinisikan koleksi atribut yang bisa digunakan untuk menjelaskan sebuah entri. Standar LDAP menyediakan beberapa tipe dasar dari *object class* yang diantaranya sebagai berikut :

- lokasi, seperti nama negara dan deskripsi,
- organisasi dalam direktori, dan keterangan tentang organisasi itu sendiri,

- unit organisasi,
- orang (perorangan) pada direktori.

Dari tipe dasar yang telah disediakan diatas, dapat dikembangkan dengan menambahkan object class sesuai dengan yang diperlukan.

Data pada sebuah direktori direpresentasikan sebagai pasangan *attribute-value*. Setiap informasi diasosiasikan dengan sebuah atribut yang dapat membedakannya dari yang lain. Sebagai contoh, *commonName(cn)*, atribut ini digunakan untuk menyimpan nama seseorang. Seseorang bernama Jonas Salk akan direpresentasikan pada direktori sebagai :

```
cn: Jonas Salk
```

Setiap orang yang dimasukkan dalam direktori didefinisikan dengan sekumpulan atribut pada object class *person*. Atribut lain yang mungkin digunakan untuk mendefinisikan object ini termasuk diantaranya :

```
givenname: Jonas
surname: Salk
mail: jonass@airius.com
```

Required attribute merupakan atribut yang harus ada untuk entri tersebut menggunakan suatu object class. Setiap entri memerlukan atribut *objectClass*, yang menghubungkan entri tersebut dan object class mana ia tergabung.

Allowed attributes adalah atribut yang boleh ada (opsional) pada sebuah entry. Sebagai contoh, pada class objek *person*, atribut *cn* dan *sn* adalah *required* attribute. Atribut *description*, *telephoneNumber*, *seeAlso*, dan *userpassword* diperbolehkan untuk lebih melengkapi, tapi tidak diharuskan.

Setiap atribut memiliki cara penulisan sendiri. Sintaks tersebut menggambarkan tipe informasi yang diterima oleh atribut tersebut:

- bin → *binary*



- ces → *case exact string* (case harus sesuai pada proses pencocokan)
- cis → *case ignore string* (case di abaikan pada saat perbandingan)
- tel → *telephone number string* (seperti cis tapi *blank* dan *dash* '-' di abaikan pada saat perbandingan)
- dn → *distinguished name*

3.3 FILE-FILE KONFIGURASI

OpenLdap memiliki beberapa file konfigurasi yang memiliki fungsi yang sangat penting.

3.3.1 File 'slapd.conf'

File yang default nya terdapat di direktori `/etc/openldap/` ini berisikan informasi konfigurasi untuk *daemon* slapd, termasuk di dalamnya sekumpulan konfigurasi global yang akan digunakan oleh slapd sebagai satu kesatuan (termasuk semua backend), diikuti oleh nol atau lebih definisi database backend yang berisi informasi khusus dari backend tersebut.

Secara umum format dari slapd.conf adalah sebagai berikut :

```
# option ini akan diterapkan pada setiap database
<global configuration options>

# option definisi dan konfigurasi untuk database 1
database <backend 1 type>
<configuration options specific to backend 1>

# option untuk database berikutnya
...
```

Option global bisa digantikan pada sebuah backend (untuk option yang muncul lebih dari satu, yang terakhir akan dipakai). Baris kosong dan baris

komentar yang dimulai dengan sebuah karakter '#' akan diabaikan. Jika sebuah baris diawali dengan spasi, maka baris itu dianggap merupakan sambungan dari baris sebelumnya.

Parameter/argumen pada baris konfigurasi dipisahkan dengan spasi/tab (*white space*). Jika sebuah argumen mengandung white space, argumen tersebut harus terletak diantara tanda petik dua. Jika sebuah argumen mengandung petik dua (") atau karakter *backslash* (\), karakter tersebut harus didahului dengan karakter backslash.

Untuk keterangan mengenai option khusus backend bisa dilihat langsung pada contoh dibawah. Contoh isi dari file *slapd.conf*. (File ini sebaiknya hanya boleh dibaca oleh root/ldap manager)

```
include      /etc/openldap/slapd.at.conf
include      /etc/openldap/slapd.oc.conf
```

Membaca informasi konfigurasi tambahan dari file yang disebutkan.

```
schemacheck  off
```

Menjalankan pengecekan *schema* atau tidak.

```
referral     ldap://10.0.0.1/
```

Penentuan server lain yang akan dirujuk, bila data yang dicari tidak ada pada database lokal.

```
pidfile      /var/run/slapd.pid
```

Nama file yang menyimpan proses id dari server ldap.

```
argsfile     /var/run/slapd.args
```

File yang menyimpan *command line option* dari server pada saat dijalankan.

```
database     ldbm
```

3.4.3 Modifikasi

Untuk *shell based*, diasumsikan file `/tmp/entrymods` berisikan :

```
dn: cn=Modify Me, o=University of Michigan,
c=US
mail=modme@terminator.rs.itd.umich.edu
+title=Grand Poobah
+jpegPhoto=/tmp/modme.jpeg
-description
```

Kemudian dijalankan perintah berikut.

```
ldapmodify -b -r -f /tmp/entrymods
```

Dari bagian diatas terlihat bahwa data dengan *dn* seperti pada baris teratas akan dirubah dengan mengganti atribut *mail*, penambahan *title* dan *jpegPhoto* serta penghilangan bagian *description*.

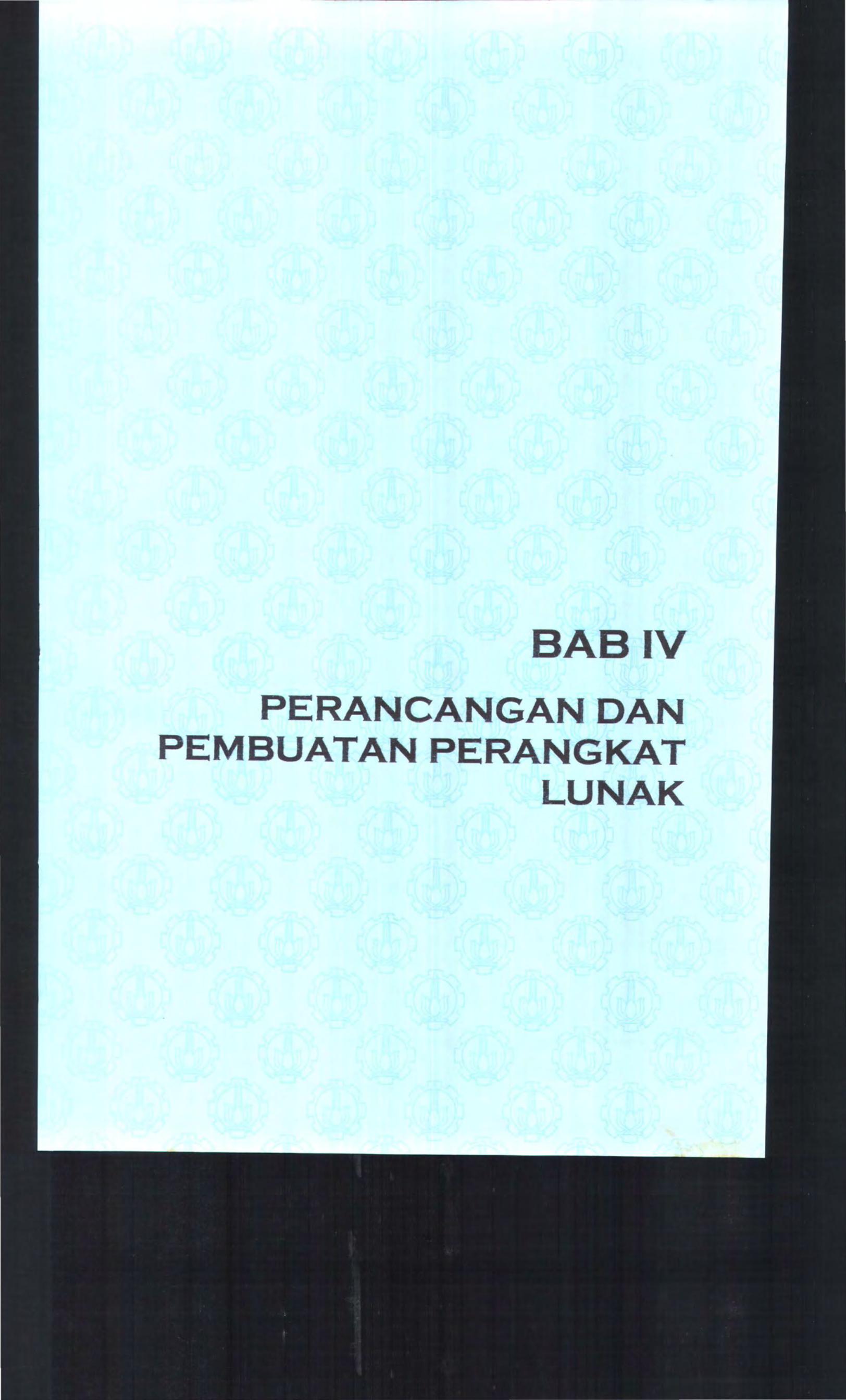
```
$r = ldap_modify($ds, "cn=John Jones, o=My
Company, c=US", $info);
```

Untuk fungsi PHP diatas, data dengan *dn* seperti dalam tanda petik akan diganti dengan data yang terdapat pada variable *\$info*.

3.4.4 Penghapusan

```
ldapdelete 'cn=Luiz Malere,o=TUdelft,c=NL'
$r = ldap_delete($ds, " cn=Luiz Malere,
o=TUdelft, c=NL ");
```

Kedua perintah/fungsi diatas menghasilkan hal yang sama, yaitu terhapusnya data dengan *dn* dalam tanda '' dan " ".



BAB IV
PERANCANGAN DAN
PEMBUATAN PERANGKAT
LUNAK

BAB IV

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan perancangan dan pembuatan perangkat lunak layanan informasi berbasis WAP dengan memanfaatkan LDAP. Perancangan dan pembuatan perangkat lunak ini berdasarkan atas cakupan ruang lingkup masalah.

4.1 PERANCANGAN PERANGKAT LUNAK

Tahap perancangan berfungsi untuk memudahkan implementasi perangkat lunak. Tahap ini meliputi arsitektur, basisdata, antarmuka, dan proses dari perangkat lunak.

4.1.1 Arsitektur

Perangkat lunak ini melibatkan beberapa komponen, yaitu

- Apache 1.3.12
- OpenLDAP 1.2.9
- PHP 4.0.0

Penggunaan LDAP sendiri dilakukan karena aplikasi memanfaatkan jaringan nirkabel telepon seluler yang memiliki bandwidth sangat terbatas sehingga mengharuskan perancangan aplikasi yang mengeluarkan hasil dalam jumlah sedikit untuk setiap kali *query*-nya.

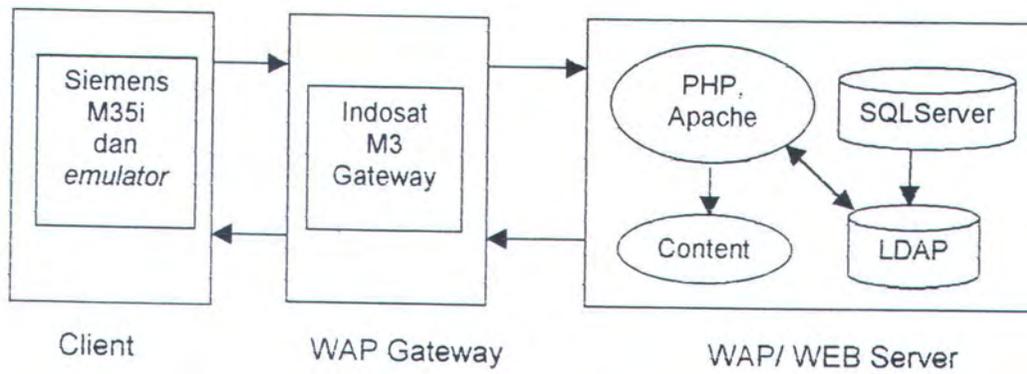
Dibawah ini merupakan hasil pengujian perbandingan kecepatan antara LDAP dan database relasional, yang menjadi dasar pemilihan LDAP, dengan melakukan pencarian berdasarkan ekspresi regular terhadap 10.000 *entry* yang terdapat di server baik LDAP maupun basisdata relasional. Kolom kedua berisikan ekspresi regular yang digunakan. Kolom ketiga merupakan jumlah data dalam persen yang dihasilkan dari proses *query*. Kolom keempat dan kelima merupakan waktu yang diperlukan untuk menyelesaikan *query*.

<u>Query type</u>	<u>Reg. Ex.</u>	<u>% data</u>	<u>LDAP (ms)</u>	<u>SQL (ms)</u>
X	abel*	0.07%	28.71	227.43
X	A*	20%	8.6	1.54
X	[abcdefghij]*	100%	5.5	0.35
*X	*zaveni	0.01%	230.00	2694.00
*X	*a	5%	24.40	4.92
*X	*[abcdefghij]	30%	9.54	1.13
*X	*[a-z]	100%	6.36	0.40
X	*y*	20%	6.97	1.16
X	*w*	50%	5.25	0.74
X	*e*	80%	4.98	0.49

Gambar 4.1 Tes pebandingan LDAP dan basisdata relasional

Dari gambar diatas dapat dilihat bahwa pemakaian LDAP lebih baik jika dari *query* yang dilaksanakan menghasilkan keluaran yang sedikit. Dengan kata lain jika menginginkan hasil pencarian dalam jumlah besar sebaiknya digunakan basisdata relasional [LDAPDSERV].

Misalnya untuk *query* keempat yang menghasilkan keluaran data sebanyak 0.01% dari total data atau 1 *entry* penggunaan LDAP terlihat lebih cepat hampir 122 kali lipat dibandingkan basisdata relasional.

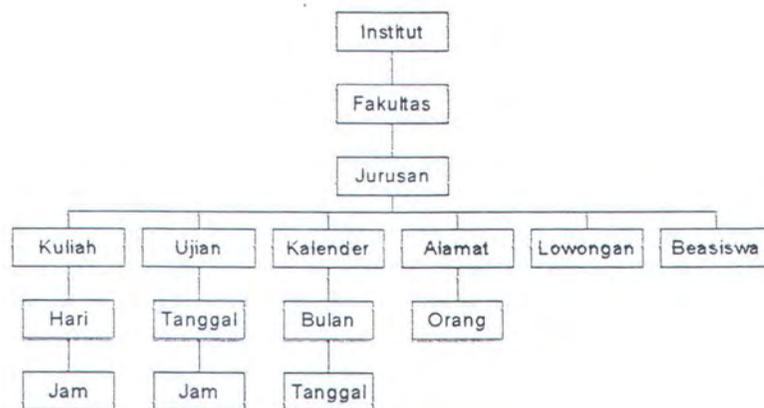


Gambar 4.2 Model arsitektur perangkat lunak

Mengambil bentuk model WAP pada bab II, gambar diatas merupakan permodelan dari perangkat lunak yang dibuat. Dari gambar terlihat bahwa yang berfungsi sebagai client adalah handphone dan emulator, sedangkan sebagai gateway digunakan WAP gateway milik Indosat. Untuk server dapat dilihat adanya PHP sebagai *content generator*. Selain itu juga terlihat adanya interaksi antara LDAP dengan SQLServer, dimana hal ini berupa pemasukan data awal dengan melakukan konversi data hasil ekspor dari SQLServer menjadi file dalam format LDIF sehingga tidak diperlukannya lagi pemasukan data yang sudah ada secara manual.

4.1.2 Basisdata - LDAP

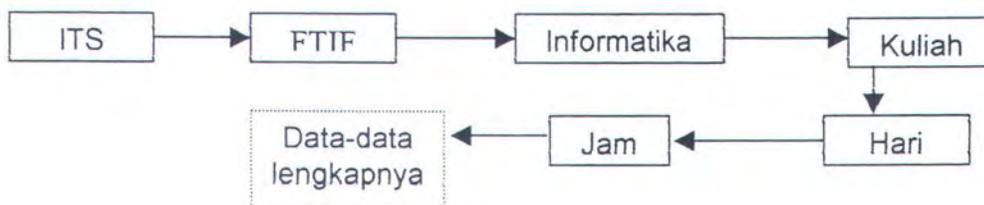
Seperti yang telah disebutkan sebelumnya, untuk penyimpanan data digunakan LDAP versi dari OpenLDAP. Berikut ini merupakan struktur dari "basisdata" yang dipakai.



Gambar 4.3 Struktur basisdata

Gambar diatas menunjukkan level-level dari data yang tersimpan dalam LDAP yang berbentuk seperti *directory tree*. Pada level teratas terdapat Institut yang diikuti fakultas dan jurusan yang termasuk di dalamnya. Hal ini ditujukan untuk kemungkinan pengembangan selanjutnya yang diharapkan bisa mencakup semua jurusan di ITS.

Level berikutnya berisikan tipe dari data yang ditampungnya, atau dengan kata lain informasi-informasi yang disimpan untuk jurusan tersebut, begitu seterusnya benar-benar seperti sebuah direktori.



Gambar 4.4 Urutan model direktori

Gambar diatas menunjukkan susunan direktori untuk data Jadwal kuliah. Untuk data lainnya memiliki bentuk yang serupa. Kotak dengan garis utuh memiliki fungsi yang sama dengan direktori sedangkan kotak dengan garis

Hari : Senin
Jam : 07.30
Ruang : TC-03
Matakuliah : Sistem Digital
Dosen : Supeno Djanali

Gambar 4.6 *Value* untuk suatu atribut

Semua informasi yang disimpan pada LDAP berbentuk pasangan *attribute-value* seperti pada gambar diatas.

4.1.3 Desain Antarmuka

Untuk pendesainan antarmuka banyak dibatasi oleh model koneksi WAP yang penuh keterbatasan. Selain itu juga dipertimbangkan efisiensi waktu yang diperlukan oleh pengguna untuk mendapatkan hasil yang diinginkan, sehingga dihindari adanya menu yang terlalu kompleks atau yang levelnya dalam. Untuk itu antarmuka dibuat sesederhana mungkin tanpa mengurangi fungsinya. Setelah diteliti halaman Web based dan WAP based yang diperlukan seperti di bawah ini.

4.1.3.1 WAP Based

- Index. Halaman ini merupakan pintu gerbang bagi pengguna yang bermaksud untuk mengakses layanan yang tersedia. Pada halaman ini terdapat semua link yang menuju ke setiap layanan yang diantaranya : buku alamat, lowongan, beasiswa, jadwal kuliah, kalender akademik, jadwal ujian, status server dan jaringan. Urutan ini dipakai berdasarkan kemungkinan layanan yang paling sering diakses oleh pengguna, dengan urutan dari atas ke bawah.

- Buku Alamat. Link ini menyediakan fasilitas pencarian orang lengkap dengan semua data-datanya.
- Lowongan. Pada link ini akan ditampilkan lowongan-lowongan pekerjaan yang ditawarkan oleh suatu perusahaan.
- Beasiswa. Menuju kepada penawaran-penawaran beasiswa.

Struktur Halaman WAP



Gambar 4.7 Struktur Halaman WAP

- Cari. Halaman ini berupa sebuah form dengan satu buah masukan sejenis *textbox*. Penggunaan satu masukan dirasa cukup dan juga atas dasar pertimbangan kesederhanaan dan kecepatan pengisian dibandingkan dengan menggunakan beberapa masukan.
- Konfirmasi. Halaman ini tampil jika dari pencarian ditemukan lebih dari 3 *record* sehingga perlu ditanyakan kepada pengguna apakah akan menampilkan semua hasil pencarian atau mengulang memasukkan *filter* yang lebih baik.

- Hasil. Pada halaman ini menampilkan hasil pencarian sesuai *string* yang dimasukkan pada form masukan sebelumnya.
- Status Server. Ditampilkan status dari *daemon* yang biasanya terdapat pada sebuah server. Apakah daemon tersebut sedang berjalan atau tidak aktif.
- Network Status. Berisikan status dari jaringan, yang termasuk diantaranya kondisi hubungan antara server dengan host lain yang bisa berupa server dari laboratorium lain, proxy, server email dan dengan dunia luar (internet).

4.1.3.2 Web Based

- Halaman login/otentikasi. Pada halaman ini pengguna diharapkan memasukkan nama user dan password sebagai *manager* LDAP untuk dapat melakukan penambahan, modifikasi dan penghapusan data.
- Halaman pemilihan layanan dan jenis *record*. Berfungsi bagi administrator / manager LDAP untuk memilih jenis record yang akan diedit.
- Halaman penambahan data. Halaman ini berupa sebuah form isian yang digunakan untuk menambah data sesuai dengan pilihan yang dilakukan pada halaman sebelumnya.
- Halaman pengeditan data. Halaman ini berisi masukan untuk mencari data mana yang akan di modifikasi. Pada bagian bawah terdapat sebuah form isian yang digunakan untuk pengeditan data hasil pencarian tadi.
- Halaman penghapusan data. Halaman ini berisi masukan untuk mencari data mana yang akan di hapus. Terdapat form yang berguna untuk memperlihatkan hasil pencarian tadi dan memastikan apakah benar data tersebut akan dihapus.

4.1.4 Proses

Berikut ini dijelaskan proses-proses yang terdapat pada aplikasi, baik untuk bagian yang berbasis WAP maupun WEB, secara langkah demi langkah untuk mengetahui secara garis besar apa saja yang terjadi.

4.1.4.1 WAP Based

Seperti telah disebutkan sebelumnya aplikasi ini bertujuan hanya untuk menampilkan berbagai informasi yang diperlukan, tidak ada proses edit data melalui ponsel. Untuk dapat menampilkan data sesuai yang diinginkan diperlukan proses pencarian. Sebelum pencarian dapat dilakukan, diperlukan adanya koneksi terlebih dahulu secara *anonymous* kepada LDAP server, lain halnya jika akan melakukan proses modifikasi diperlukan model koneksi yang terautentikasi. Setelah melakukan hubungan itu, baru dapat dilaksanakan proses pencarian dengan memberikan *filter* yang sesuai dengan keinginan, yang diberikan lewat masukan-masukan yang telah disediakan.

Jika pada form yang menyediakan masukan untuk digunakan sebagai *filter* tidak diisi oleh pengguna atau dari *filter* tersebut menghasilkan lebih dari tiga *record*, maka aplikasi akan menanyakan kepada pengguna apakah akan ditampilkan semua hasil pencarian tersebut ataukah mengganti *filter* yang digunakan. Kecuali untuk data orang, akan langsung diminta untuk memperbaiki *filter*. Hal ini ditanyakan karena jika pencarian menghasilkan banyak *record* maka akan diperlukan koneksi ke server LDAP secara berulang-ulang. Hal ini terpaksa dilakukan karena data yang dapat diterima oleh ponsel sangat terbatas sehingga harus dikirimkan sebagian-sebagian. Oleh karena itu diharapkan pengguna memasukkan *filter* yang kemungkinan menghasilkan keluaran dalam jumlah kecil.

Setelah proses pencarian diatas, dilaksanakan pengambilan data. Pada saat pengambilan data ini LDAP memberikan keleluasaan untuk memilah-milah atribut mana saja yang akan diambil. Hal ini bisa dilakukan karena hasil pencarian disimpan dalam bentuk *array*.

Setelah data yang diinginkan diambil, data tersebut kemudian ditampilkan ke user, karena pertimbangan kemampuan ponsel yang terbatas, data yang ditampilkan sangatlah sederhana, yang diutamakan adalah bahwa informasi tersebut tersampaikan dengan baik dan cepat.

Untuk layanan lainnya, tanpa menggunakan LDAP, memanfaatkan kemampuan PHP. Layanan yang menampilkan status server berisikan kondisi dari *daemon*, aplikasi-aplikasi yang bertindak sebagai server dari suatu layanan pada sistem operasi Linux. Untuk mengetahui server dari layanan apa saja yang sedang berjalan dilakukan dengan memeriksa kondisi dari *port* yang digunakan oleh tiap-tiap layanan tersebut. Jika port tersebut terbuka, dapat disimpulkan bahwa server layanan yang menggunakan *port* tersebut sedang berjalan. Dan sebaliknya.

Untuk mengetahui kondisi suatu koneksi dengan *host* lain (dalam jaringan) digunakan perintah standar *ping* yang tujuannya adalah untuk mengetahui apakah hubungan dengan *host* tersebut masih terbuka dan untuk mengetahui seberapa cepat/lambat hubungan itu sendiri dengan melihat waktu yang dibutuhkan oleh *host* yang di *ping* tersebut untuk membalas paket-paket yang dikirim.

4.1.4.2 Web Based

Login harus menggunakan nama *user* yang sesuai dengan yang terdapat pada file konfigurasi *slapd.conf*, pada bagian *rootdn*. Untuk perangkat lunak ini digunakan *default* nya, yaitu 'Manager'. Untuk *password* juga disesuaikan dengan file konfigurasi tadi, pada bagian *rootpw*. *Username* dan *password* yang telah dimasukkan tadi akan dikirimkan secara tersembunyi (*hidden*) untuk diperiksa kemudian pada saat penambahan/modifikasi data.

Penambahan data dilakukan dengan langkah sebagai berikut :

- koneksi ke server sebagai *Manager*
- memasukkan semua masukan dari form pada halaman sebelumnya ke sebuah *array* tunggal yang nantinya akan ditambahkan bersamaan dengan *distinguish name (dn)* yang diambil dari sebagian data tersebut. Misal untuk data orang yang diambil sebagai *dn* adalah atribut nama. Hal ini dilakukan karena nama merupakan sesuatu yang unik.

Sedangkan untuk melakukan modifikasi terhadap suatu data, dilakukan pencarian terlebih dahulu terhadap data tersebut. Hal ini diperlukan karena proses modifikasi memerlukan *dn* dari data tersebut. Algoritma :

- *string* yang dimasukkan diproses seperti pada bagian *wap based* diatas
- *dn* dari data tersebut diambil untuk digunakan selanjutnya. Hal ini terutama berguna jika ada perubahan terhadap *value* dari sebuah atribut yang digunakan sebagai bagian dari *dn*, sedangkan *dn* itu sendiri tidak bisa dirubah. Misalnya data orang, yang dimasukkan ke *dn* adalah atribut nama. Jika ada perubahan pada nama, tidak akan mungkin menyimpan perubahan tersebut tanpa mengetahui *dn*-nya. Jika yang mengalami perubahan adalah atribut yang tidak dipakai untuk *dn*, hal itu masih bisa dilakukan.

- hasil yang didapat ditampilkan, pada sebuah form isian untuk diedit
- hasil perubahan diproses menyerupai pada saat penambahan data, hanya saja yang digunakan adalah fungsi *modify* dan *dn* yang dipakai adalah *dn* yang lama, bukan mengambil lag dari data hasil modifikasi tadi.

Untuk menghapus data prosesnya hampir sama dengan penambahan dan modifikasi, tapi yang dibutuhkan hanya bagian *dn*-nya saja, karena seperti disebut sebelumnya yang membedakan antara satu *record* dengan lainnya adalah *dn* tersebut. Tidak ada dua *record* atau lebih yang memiliki *dn* yang sama. Namun sebelum data benar-benar dihapus, dilakukan pencarian terlebih dahulu, sama seperti pada awal proses modifikasi, hanya saja disini tujuannya selain untuk mendapatkan *dn* juga untuk memastikan apakah benar data tersebut yang akan dihapus.

4.2 PEMBUATAN PERANGKAT LUNAK

Berikut ini akan dibahas mengenai pembuatan perangkat lunak sesuai dengan hasil perancangan sebelumnya.

4.2.1 LDAP

Pembuatan perangkat lunak diawali dengan penanganan LDAP sebagai penampung data. Dalam bagian ini dijelaskan hal-hal yang berhubungan dengan persiapan penggunaan LDAP itu sendiri, seperti pembentukan struktur direktori sampai dengan cara melakukan koneksi ke server LDAP.

4.2.1.1 Objectclass Dan Pembentukan Direktori

Sesuai dengan perancangan maka diperlukan susunan *objectclass* yang dapat membentuk struktur tersebut. *Objectclass-objectclass* ini diletakkan pada file konfigurasi LDAP yaitu *slapd.oc.conf*. Setelah menemukan susunan *objectclass* yang sesuai dengan perancangan diatas, barulah dapat dimulai pembentukan direktorinya. Hal ini sebenarnya prosesnya sama saja dengan penambahan data, pada posisi file, hanya saja urutan penambahannya harus diperhatikan. Yang pertama kali ditambahkan harus dari tingkat paling atas, dalam hal ini top-organisasi yang berisikan "ITS" sebagai level teratas. Penambahan ini juga sekaligus membentuk database yang akan dibuat. Selanjutnya ditambahkan sesuai dengan level-level pada gambar 4.3.

```
objectclass top requires objectClass
objectclass organisasi requires objectClass, o
objectclass fakultas requires objectClass, fak
objectclass jurusan requires objectClass, jur
objectclass tipe requires objectClass, tp
objectclass bulan requires objectClass, bln
objectclass hari requires objectClass, hr
objectclass paket_kuliah
  requires objectClass, jam, hr, ruang, matakuliah, dosen
objectclass orang
  requires objectclass, nama, alamat, telpl, email, status
objectclass lowongan requires objectclass, persh, posisi,
  batas
...
```

Gambar 4.8 Objectclass

```
dn: fak=FTIF, o=ITS
objectClass:fakultas
fak:FTIF

dn:fak=FTI, o=ITS
objectClass:fakultas
fak:FTI
```

Gambar 4.9 File LDIF, Pembentukan Direktori

Gambar diatas merupakan potongan *script* yang digunakan untuk mengkonversi hasil ekspor data dari SQLServer yang berformat *Comma Separated Values* (CSV) ke dalam format LDIF yang dikenal oleh LDAP, khusus untuk data orang.

```

sub print_entry
{
    local ($nr,$namax,$alamatx,$telplx,$tel2x,$emailx) = @_;

    print OUT "dn: nama=$namax, tp=b-alamat, jur=Informatika,
              fak=FTIF, o=ITS\n";
    print OUT "objectClass : orang\n";
    print OUT "nama : $namax\n";
    print OUT "alamat : $alamatx\n";
    if ( $telplx ne "" )
    { print OUT "telpl : $telplx\n";      }
    if ( $tel2x ne "" )
    { print OUT "tel2 : $tel2x\n";      }
    if ( $emailx ne "" )
    { print OUT "email : $emailx\n";    }
    print OUT "\n";
}

```

Gambar 4.11 Potongan script penulisan file LDIF hasil konversi

Gambar 4.11 berisikan fungsi yang digunakan untuk penulisan ke sebuah file, hasil konversi pada *script* sebelumnya (gambar 4.10).

4.2.1.3 Koneksi

Untuk dapat mengakses data pada LDAP server diperlukan koneksi terlebih dahulu. Setelah koneksi berhasil dilakukan selanjutnya adalah *binding* baik secara *anonymous* maupun ter-autentikasi. Secara *anonymous* dapat dilakukan jika hanya akan melakukan operasi pencarian atau dengan kata lain hanya melihat saja tanpa melakukan perubahan terhadap data-data yang ada.

```

$ds=ldap_connect("10.0.0.1");
if ($ds)
{
    $r=ldap_bind($ds,"cn=$User, o=ITS", "$Pass");
    ...
}

```

Gambar 4.12 Koneksi dengan Autentikasi

Untuk koneksi *anonymous* menggunakan format yang hampir sama hanya saja tanpa username dan password. Koneksi ter-autentikasi ini diperlukan apabila hendak melakukan perubahan baik penambahan, modifikasi maupun penghapusan data.

4.2.2 WAP Based

Pada bagian WAP *based* ini dibicarakan semua hal yang menyangkut bagian aplikasi yang berbasiskan WAP, mulai dari cara penggunaan *link*, permintaan masukan sampai cara menampilkan informasi kepada pengguna.

4.2.2.1 Link

Untuk dapat berpindah antar card dan *deck* diperlukan sebuah *link* serupa dengan penggunaan *link* pada html.

```

<card id="main" title="T C">
  <do type="prev" label="Back"><prev/></do>
  <p>Layanan :<br/>
    <a href="form-cari.php?Tipe=4" title="Go">Buku
      Alamat</a><br/>
  .
  .

```

Gambar 4.13 Link pada WAP Page

record untuk tipe tersebut. Jika pengguna menjawab tidak, maka akan diminta untuk memasukkan *filter* dengan benar.

Selain masukan *Stringx* tersebut, juga dikirimkan variabel "tipe" yang berguna untuk membedakan jenis *record* satu dengan lainnya. Untuk seterusnya variabel *tipe* inilah yang digunakan untuk membedakan tiap jenis *record* yang akan diakses.

4.2.2.3 Pencarian

Untuk pencarian, sesuai dengan perancangan, ada tiga jenis, yaitu untuk semua record, kecuali data alamat, terjadi jika pada halaman permintaan *filter* dibiarkan kosong. Yang kedua adalah jika dari *filter* yang dimasukkan menghasilkan *record* lebih dari satu, dan yang ketiga adalah jika menghasilkan hanya satu *record* saja.

Semuanya sebenarnya menggunakan cara yang sama, tetapi dengan sedikit modifikasi pada cara pengambilan dan menampilkannya.

```
if($Stringx == "")
{
    //string kosong
    if($tipe == 1)
    {
        $filename = 'kuliah.php';
        $filter = "(objectclass=*kuliah)";
    }
    else
    if($tipe == 2)
    { //tipe yang lain }
    ...
}
```

Gambar 4.15 Masukan untuk *filter* dikosongkan

Jika pengguna tidak memasukkan *string* pada halaman sebelumnya, maka dianggap pengguna ingin menampilkan semua *record* yang ada untuk tipe informasi tersebut. `$filename` berisikan nama file yang akan dituju jika pengguna menjawab "ya" pada saat dikonfirmasi. *Script* tersebutlah yang nantinya akan menampilkan semua *record* yang ada.

Gambar 4.16 memperlihatkan `$filter` yang nantinya akan digunakan sebagai *filter* pencarian yang sebenarnya, diisi dengan *string* yang telah dimasukkan oleh pengguna.

```

if($Stringx == "")
{ ... }
else
{
  if($tipe == 1)
    $filter="(&(objectclass=*kuliah) (|(hr=*$Stringx*)
              (jam=*$Stringx*) (ruang=*$Stringx*)
              (matakuliah=*$Stringx*) (dosen=*$Stringx*)))";
  else ...
}

```

Gambar 4.16 Pengisian *filter* dari *string* masukan

```

if($x > 3)
{
  if($tipe == 4)
  {
    echo "Lebih dari tiga record. Perbaiki Filter !";
    echo "<do type=\"option\" label=\"back\">";
    echo "<prev/>";
    echo "</do>";
  }
  else
  {
    echo "Lebih dari tiga.<br/>Tampilkan tiap record ?";
    echo "<do type=\"accept\" label=\"Ya\">";
    echo "<go href=\"\$filename\">";
    echo "<postfield name=\"tipe\" value=\"\$tipe\"/>";
    echo "</go></do>";
    echo "<do type=\"option\" label=\"Tidak\">";
    echo "<prev/>";
    echo "</do>";
  }
}
}

```

Gambar 4.17 Hasil lebih dari tiga *record*

Dari gambar diatas terlihat bahwa jika hasil yang didapat dengan *filter* dikosongkan ataupun diisi tapi menghasilkan lebih dari tiga *record* maka akan ditanyakan kepada pengguna apakah akan menampilkan semua *record* atau kembali untuk mengisi *filter*. Jika pengguna menjawab ya maka akan dialihkan ke *script* lain untuk melakukan pencarian. Hal ini tidak berlaku untuk data orang. Untuk data orang maksimal hasil yang akan ditampilkan adalah untuk tiga orang, jika dari pencarian menghasilkan lebih dari tiga maka pengguna diminta memperbaiki *filter*.

```

$i = $counter;
//connect + bind
$sr=ldap_search($ds,"o=ITS", "objectclass=*beasiswa");
$x=ldap_count_entries($ds, $sr);
$entry = ldap_first_entry($ds, $sr);
if($i<1)
{
    $i = 0;
}
if($x > 0)
{
    if(($i > 0) and ($i <= $x))
    {
        for($a=0;$a<$i;$a++)
        {
            $entry=ldap_next_entry($ds, $entry);
        }
    }
}
if($i <= $x)
{
    if($x > 3)
        if($x - $i >=3)
            $y = 3;
        else
            $y = $x - $i;
    else $y = $x;
    for($a=0;$a<$y;$a++)
    {
        if($entry)
            $attrs = ldap_get_attributes($ds, $entry);

        //ambil dan tampilkan data
        $entry = ldap_next_entry($ds, $entry);
        $i = $i + 1;
    }
}
}

```

Gambar 4.18 Pencarian semua record, tampil tiga sekaligus

Untuk dapat mengambil setiap record dan menampilkannya tiga sekaligus pada satu *card*, pencarian harus dilakukan berulang sesuai dengan jumlah record (gambar 4.18). Hal ini dapat dilakukan dengan memanfaatkan variabel tambahan yang berguna sebagai *counter* sehingga pada pencarian berikutnya diketahui data mana yang seharusnya diambil.

Yang kedua, untuk *filter* yang menghasilkan keluaran lebih dari satu tapi kurang atau sama dengan tiga, dapat dilihat pada gambar 4.19. Pada gambar tersebut terlihat adanya perulangan pengambilan data sebanyak jumlah *record*.

```

$dn = "o=ITS";
$sr = ldap_search($ds, $dn, $filter);
$x = ldap_count_entries($ds,$sr);

if($Stringx == "")
{ ... }
else if(($x <= 3) & ($x > 1))
{
    $entry = ldap_first_entry($ds, $sr);

    for($a=0;$a<$x;$a++)
    {
        if($entry)
            $attrs = ldap_get_attributes($ds, $entry);

        if($tipe == 1)
        {
            $hari = $attrs["hr"][0];
            $jam = $attrs["jam"][0];
            $ruang = $attrs["ruang"][0];
            $matkul = $attrs["matakuliah"][0];
            $dosen = $attrs["dosen"][0];
            ... ..
        }
        else if($tipe == 2)
        { ... .. }

        $entry = ldap_next_entry($ds, $entry);
    }
}

```

Gambar 4.19 Pencarian 1-3 *record*

Yang terakhir untuk filter yang menghasilkan hanya satu keluaran dapat dilihat pada gambar 4.20.

```

$dn = "o=ITS";
$sr = ldap_search($ds, $dn, $filter);
$х = ldap_count_entries($ds, $sr);

if($Stringx == "")
{ ... }
else if(($x <= 3) & ($x > 1))
{ ... }
else if($x == 1)
{
    $entry = ldap_first_entry($ds, $sr);
    if($entry)
        $attrs = ldap_get_attributes($ds, $entry);

    if($stipe == 1)
    {
        $hari = $attrs["hr"][0];
        $jam = $attrs["jam"][0];
        $ruang = $attrs["ruang"][0];
        $matkul = $attrs["matakuliah"][0];
        $dosen = $attrs["dosen"][0];
        ...
    }
}

```

Gambar 4.20 Pencarian data tunggal

4.2.2.4 Tampilan

Karena adanya keterbatasan layar tampilan, tampilan dibuat sesederhana mungkin dengan tanpa mengurangi informasi yang harus ditampilkan.

```

echo "<big>$nama</big><br/>";
echo "$salamat<br/>";
echo "$stelp1<br/>";
echo "$stelp2<br/>";
echo "$semail<br/>";
echo "$status<br/>"

```

Gambar 4.21 Menampilkan Informasi

Pada baris teratas menggunakan <big> dengan tujuan agar pengguna lebih terpusat perhatiannya dan untuk yang ditampilkan lebih dari satu *record*

dalam satu *card* akan berkesan sebagai pembatas antara *record* yang satu dengan yang lain, sehingga akan mempermudah dalam pembacaan.

```

if($i<$x)
{
    echo "<do type=\"accept\" label=\"next\">";
    echo "<go href=\"beasiswa.php\">";
    echo "<postfield name=\"counter\" value=\"$i\"/>";
    echo "</go></do>";
    echo "<do type=\"option\" label=\"back\">";
    echo "<prev/>";
    echo "</do>";
}
else
{
    echo "<do type=\"accept\" label=\"menu\">";
    echo "<go href=\"index.php\">";
    ...
}

```

Gambar 4.22 Menampilkan *link* untuk *softkey*

Gambar 4.22 diatas berguna untuk menampilkan *link* "next" dan "back" yang di asosiasikan dengan *softkey* *accept* dan *option*. "next" berguna untuk menampilkan kelompok data berikutnya, setelah tiga pertama. Jika habis akan ditampilkan "menu" untuk *softkey* tersebut dan pengguna akan dibawa kembali ke halaman awal.

4.2.2.5 Kondisi Server dan Jaringan

Untuk pengecekan kondisi server dan jaringan digunakan perintah-perintah yang dimiliki oleh php. Untuk mengetahui *daemon* apa saja yang pada saat itu berjalan di server digunakan fungsi pengecekan *port*. Jika *port* tersebut terbuka maka disimpulkan *daemon* yang menggunakan *port* tersebut sedang berjalan begitu juga sebaliknya.

```

Sports_array = array(
    "1" => array ("ftpd", 21),
    "2" => array ("DNS", 53),
    "3" => array ("apache", 80),
    "4" => array ("LDAP", 389),
    ...
);
for($i=1;$i<10;$i++)
{
    if(!fsockopen("10.0.0.1", Sports_array[$i][1],
        $errno, $errstr, 30))
    {
        //Tampilkan daemon itu tidak berjalan
        $err_found=1;
    }
    else
    { //Tampilkan daemon itu berjalan }
}

```

Gambar 4.23 Pengecekan Port

```

echo "<br/>Komputing :";
system("ping -c 1 10.126.10.21 |grep 64 | cut -d '=' -f 4");
echo "<br/>TCMail :";
system("ping -c 1 10.126.11.11 |grep 64 | cut -d '=' -f 4");
echo "<br/>Proxy :";
system("ping -c 1 203.155.180.84 |grep 64 | cut -d '=' -f 4");
...

```

Gambar 4.24 Mengecek Kondisi Jaringan

Sedangkan untuk mengetahui kondisi koneksi terhadap host lain, seperti lab lain, proxy, email server atau internet digunakan perintah internal Linux yang dapat diakses oleh php, dalam hal ini *ping*.

4.2.3 Web Based

Untuk bagian web based, halaman pertama yang ditemui adalah halaman login. Halaman ini meminta masukan *username* dan *password* untuk *manager* LDAP yang harus diberikan untuk proses *editing*.

```

if($radiol == 1)
    $filter = "(&(objectclass=*kuliah)(|(hr=*$string*)
                (jam=*$string*)(ruang=*$string*)
                (matakuliah=*$string*)(dosen=*$string*)))";
else
if($radiol == ??)
// tipe data yang lain

$dn = "o=ITS";
$sr=@ldap_search($ds, $dn, $filter);
$entry = @ldap_first_entry($ds, $sr);

$dn_x = @ldap_get_dn($ds, $entry);
$value = @ldap_explode_dn($dn_x, 1);
$dn_y = $value[0];
$dn_z = $value[1];

if($entry)
    $attrs = @ldap_get_attributes($ds, $entry);
    if($radiol == 1)
    {
        $hari = $attrs["hr"][0];
        $jam = $attrs["jam"][0];
        $ruang = $attrs["ruang"][0];
        $matkul = $attrs["matakuliah"][0];
        $dosen = $attrs["dosen"][0];
    }
    else { ... }

```

Gambar 4.26 Pencarian, mengambil dn

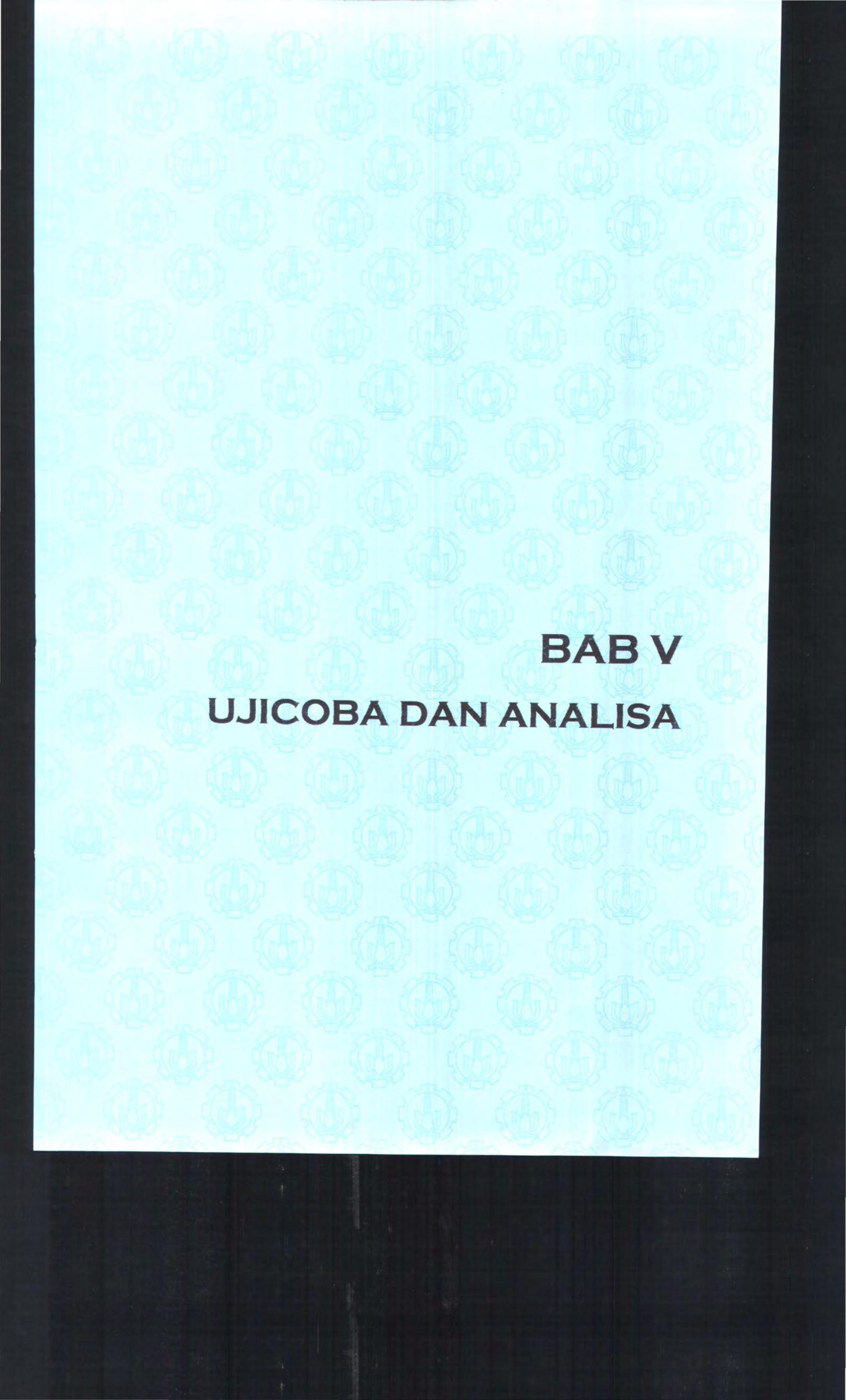
4.2.3.2 Penambahan Data

```

$str = "jur=informatika, fak=FTIF, o=ITS";
if($Tipe == "1")
{
    $info["hr"]=$Harix;
    $info["jam"]=$Jamx;
    $info["ruang"]=$Ruangx;
    $info["matakuliah"]=$Matkulx;
    $info["dosen"]=$Dosenx;
    $info["objectclass"]="paket_kuliah";
    $r = @ldap_add($ds, "jam=$Jamx, hr=$Harix, tp=kuliah,
                    $str", $info);
}
else { //data yang lainnya }

```

Gambar 4.27 Penambahan data



BAB V
UJICoba DAN ANALISA

BAB V

UJICOBA DAN ANALISA

Pada bab ini akan dibahas mengenai pelaksanaan ujicoba, langkah-langkah dan hasilnya serta evaluasi terhadap hasil ujicoba tersebut.

5.1 LINGKUNGAN UJI COBA

Server (untuk emulator) :

- AMD Duron 750 Mhz, SDRam 160 Mb
- Linux RedHat 6.0, Apache 1.3.12, OpenLdap 1.2.9, PHP 4.00

Client :

- Intel Celeron 566, SDRam128 Mb
- Windows 98, CCWap 3.2, UP.SDK 4.1

Server (untuk ponsel) :

- Intel Pentium 166 MMX, SDRam 64 Mb
- Linux RedHat 6.2, Apache 1.3.12, OpenLdap 1.2.9, PHP 4.00

Client :

- Siemens M35i

5.2 KONFIGURASI SERVER

Apache Web server memiliki file konfigurasi utama yaitu *httpd.conf*, hampir semua *setting* yang dipakai terdapat di file tersebut. Untuk aplikasi ini yang terkait langsung hanyalah bagian yang berhubungan dengan kemampuan

menjalankan *script* PHP dan WML. Untuk PHP otomatis telah ditambahkan pada saat kompilasi Apache dan PHP. Sedangkan agar Apache dapat menerima file dalam format WML, *script* pada gambar 5.1 dibawah perlu ditambahkan sendiri.

```
addtype text/vnd.wap.wml wml
addtype image/vnd.wap.wbmp wbmp
AddType text/vnd.wap.wmlscript wmls
AddType application/vnd.wap.wmlc wmlc
AddType application/vnd.wap.wmlscriptc wmlsc
AddType application/x-httpd-php .wml
```

Gambar 5.1 Penambahan *Type* untuk WML

Bagian ini bisa ditambahkan pada file konfigurasi *httpd.conf* jika memiliki hak untuk mengeditnya atau *access.conf*.

5.3 KONFIGURASI SIEMENS M35I

Untuk dapat mengakses internet dengan menggunakan ponsel Siemens M35i ada beberapa hal yang perlu di *setting* terlebih dahulu. Langkah-langkahnya sebagai berikut :

1. Tekan *Softkey* untuk *Menu*
2. Pilih *Setup*
3. Pilih *FAX/Data Mode*
4. Pilih *Rec. FAX/Data*
5. Kembali ke menu utama. Pilih *Internet*
6. Pilih *Profiles*
7. Pilih salah satu *profile*
8. Pilih *Settings*,
9. Mulai dari sini diperlukan masukan. Untuk *profile name* bisa diisi apa saja, misal *TA*

10. Untuk *Dial-up number*, no akses WAP Gateway, dalam hal ini IM3, 688
11. *Call type* = ISDN
12. *Login name* = csd, IM3 memiliki kemampuan untuk mengakses internet (*WAP content*) memanfaatkan GPRS, tetapi telepon seluler Siemens M35i tidak memiliki kemampuan tersebut, hanya bisa menggunakan *Circuit Swtch Data*(CSD)
13. *Password* = im3
14. *IP address* = 010.019.019.019
15. Tekan *Softkey* untuk perubahan menu, untuk masuk ke bagian pemilihan URL yang akan dituju.

Langkah diatas, serta isi dari masukan-masukannya terbatas pada penggunaan ponsel Siemens M35i dan SIM card Indosat M3. Untuk ponsel dan SIM card lainnya akan memiliki setting yang berbeda.

Selain menggunakan WAP gateway yang dimiliki oleh provider SIM card dapat juga menggunakan ISP biasa, misal Centrin, dengan memanfaatkan WAP gateway yang tersedia gratis di internet, seperti Anytimenow.com.

Pemilihan penggunaan provider dan WAP gateway penyedia SIM card didasarkan atas pertimbangan biaya yang lebih murah. Untuk pengaksesan WAP melalui WAP gateway milik IM3 biaya yang dikenakan adalah Rp 550,- /menit, tanpa biaya lain lagi. Sedangkan jika menggunakan ISP Centrin misalnya, akan dikenakan biaya pulsa lokal Rp 880,- /menit, dan selain itu juga biaya berlangganan di Centrin.

5.4 UJI COBA APLIKASI

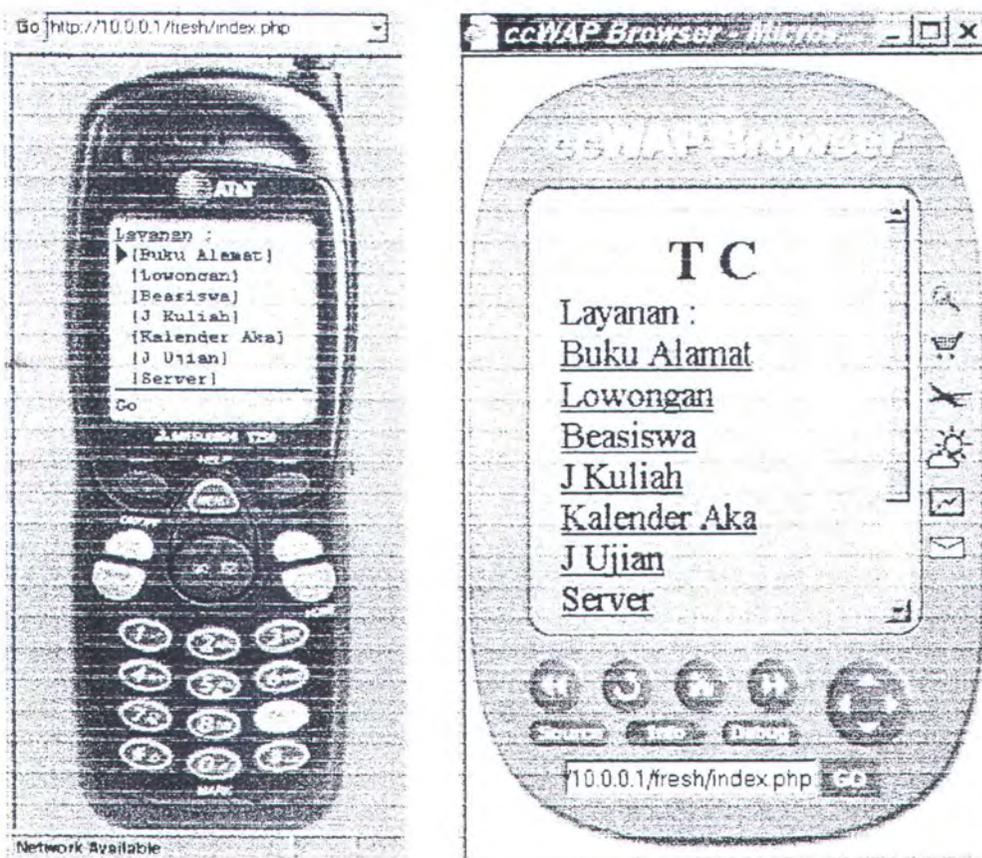
Untuk menguji aplikasi ini dilakukan dua jenis uji coba. Yang pertama menggunakan *emulator* dan selanjutnya dengan ponsel yang sebenarnya. Hal ini

bertujuan untuk, yang pertama mengurangi biaya percobaan, dan yang kedua adalah membandingkan hasilnya.

5.4.1 Uji Coba Dengan Menggunakan Emulator Dan Analisisnya

Pada ujicoba ini digunakan *emulator* dari CCWap Browser 3.2 dan UP.SDK 4.1 yang dapat didownload secara gratis. UP.SDK 4.1 memiliki tambahan *window* untuk menampilkan output sejenis *debug* yang berguna untuk melihat jalannya ujicoba misalnya parameter yang dikirimkan dan pesan-pesan kesalahan. CCWap Browser juga memiliki fasilitas untuk menampilkan informasi-informasi yang cukup berguna seperti tentang WML yang di-load tersebut, *history*, dan variabel-variabel yang digunakan serta parameter-parameter yang dilewatkan antar WML.

Untuk ujicoba dengan *emulator* ini server dan client berada dalam suatu jaringan lokal. Hasil yang diperoleh telah sesuai dengan yang diinginkan. Pada halaman berikut akan ditampilkan hasil-hasil ujicoba pada kedua *emulator* tadi.

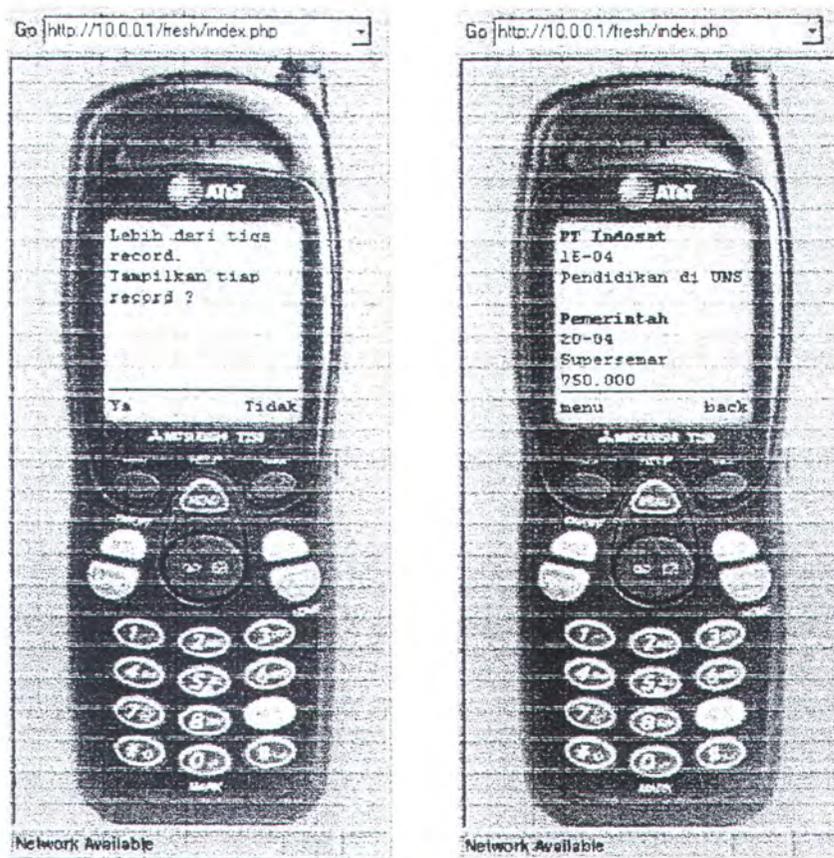


(a)

(b)

Gambar 5.2 Menu Awal, Emulator

Gambar diatas menunjukkan tampilan awal aplikasi pada emulator UP.SDK (a) dan CCWap Browser (b). Dari gambar tersebut terlihat layanan-layanan informasi yang ditawarkan. Pada bagian bawah (a), dibawah garis, merupakan tempat ditampilkannya *function keys*, dalam hal ini adalah *accept* dengan label *Go*. *Function key accept* selalu diasosiasikan dengan *softkey* pada sebelah kiri, sedangkan *option* pada sebelah kanan. Hal ini tidak terdapat pada CCWap Browser (b). Perbedaan yang lain terletak pada munculnya *title* dari card pada CCWap Browser, sedangkan pada UP.SDK tidak diperlihatkan.

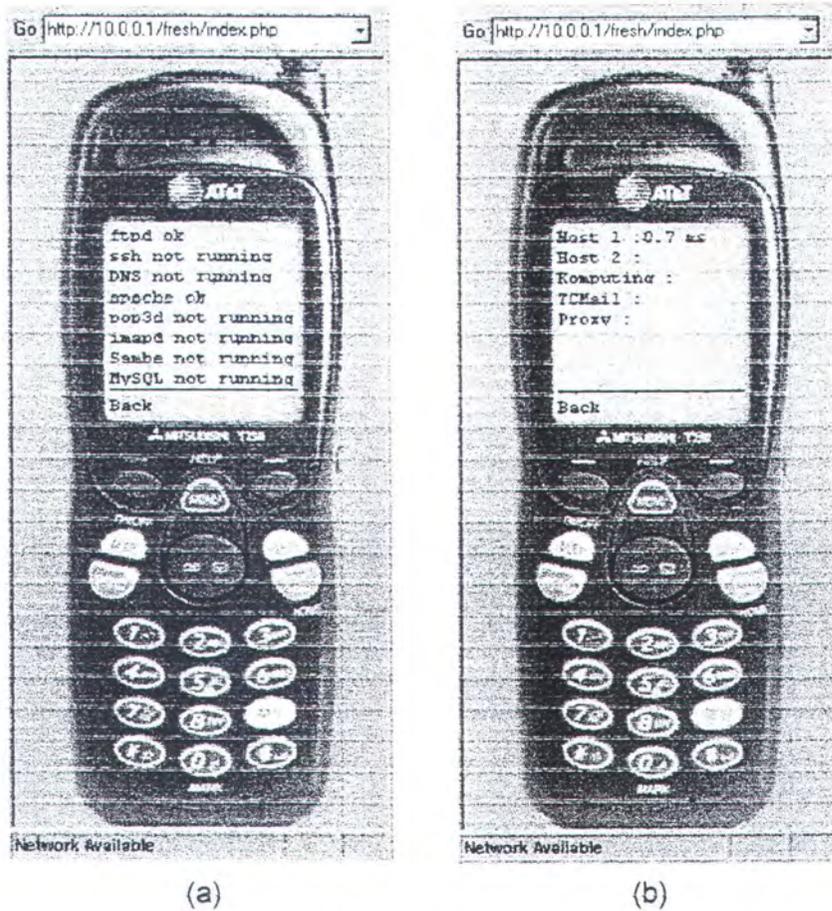


(a)

(b)

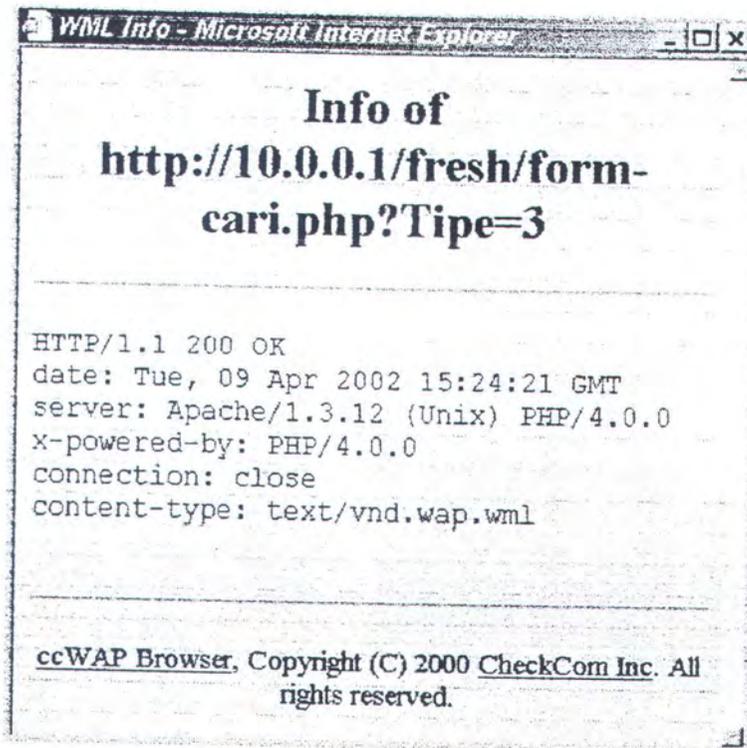
Gambar 5.5 Konfirmasi dan Akhir Pencarian

Gambar 5.5 (a) diatas, seperti telah disebutkan sebelumnya, akan tampil jika terdapat lebih dari tiga *record* yang sesuai dengan kriteria pengguna. Jika pengguna menjawab "Ya" maka akan ditampilkan semua *record* dalam format satu halaman terdiri dari tiga *record* (gambar 5.4). Sedangkan gambar 5.5 (b) menunjukkan kondisi akhir dari suatu pencarian, kelompok *record* terakhir. Berbeda dengan sebelumnya, dimana pada *function key accept* berisikan "next" maka pada halaman terakhir ini diisi link untuk kembali ke menu utama.



Gambar 5.6 Status Server dan jaringan

Gambar 5.6 (a) diatas merupakan halaman yang menunjukkan kondisi server. Dari hasil tersebut terlihat bahwa *daemon* ftp dan apache sedang berjalan, sedangkan samba, imap, dan lainnya tidak berjalan. Gambar 5.6 (b) menunjukkan kondisi hubungan antara server dengan host lain. Dari hasil tersebut terlihat bahwa hubungan dengan host 1 masih terbuka, sedang dengan yang lainnya tidak. Kedua informasi ini sangat berguna bagi seorang administrator. Ia dapat mengetahui dengan segera jika terjadi suatu kegagalan/kesalahan pada server maupun jaringannya.



Gambar 5.7 Informasi, CCWap Browser

Gambar diatas memperlihatkan informasi tentang file yang diakses oleh CCWap Browser. Disebutkan pula bahwa server menggunakan Apache versi 1.3.12 dan PHP 4.00.

```

cache hit: <http://10.0.0.1/fresh/index.php>
cache miss: {
                <http://10.0.0.1/fresh/form-cari.php?Tipe=6>
            }
net request: {
                <http://10.0.0.1/fresh/form-cari.php?Tipe=6>
            }

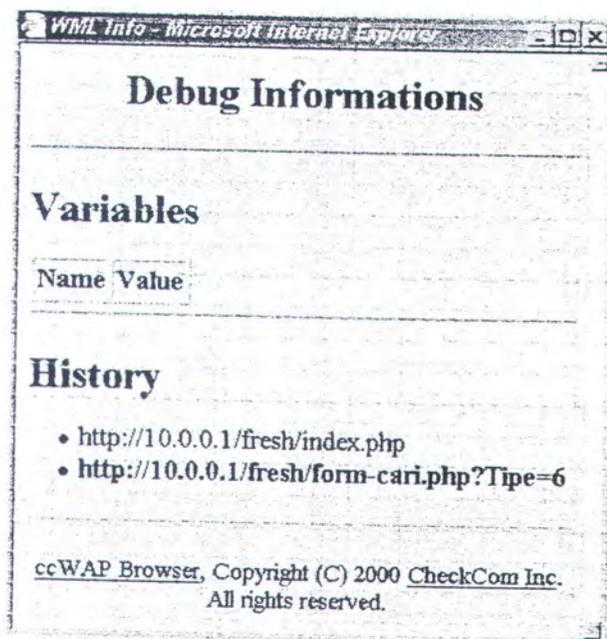
HTTP GET Request: http://10.0.0.1/fresh/form-cari.php?Tipe=6
Checking for URL '10.0.0.1/fresh/form-cari.php?Tipe=6' in
Whitelist
    not found

----- DATA SIZE -----
Uncompiled data from HTTP is 393 bytes.
...found Content-Type: text/vnd.wap.wml.
Compiled WAP binary is 183 bytes.
-----

```

Gambar 5.8 Informasi, UP.SDK

Gambar diatas merupakan sebagian informasi yang dikeluarkan oleh UP.SDK. dari keluaran itu dapat dilihat bahwa *emulator* UP.SDK memiliki *cache* untuk menyimpan halaman yang sudah pernah diakses. Pada baris pertama terlihat bahwa *index.php* tidak diambil dari *server*, melainkan dari *cache*. Sedangkan *form-cari.php* dengan parameter *tipe=6* tidak terdapat di *cache* sehingga perlu diminta dari *server*. Pada bagian akhir menunjukkan tipe dari data yang diakses, yaitu *text/vnd.wap.wml* dan dalam kondisi belum terkompilasi.



Gambar 5.9 Debug Info, CCWap Browser

Sedangkan gambar diatas menunjukkan informasi *debug* oleh CCWap Browser. Dalam informasi tersebut dapat diketahui *history* dari halaman-halaman yang telah diakses sampai saat tertentu. Selain itu juga memperlihatkan variabel-variabel yang digunakan dan dilewatkan antar halaman.



BAB VI
KESIMPULAN DAN SARAN

BAB VI

KESIMPULAN DAN SARAN

6.1 KESIMPULAN

Dari hasil pengamatan selama proses perancangan, pembuatan, dan proses uji coba perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Dalam tugas akhir ini telah berhasil dibuat sebuah aplikasi yang dapat diakses sewaktu-waktu dan dari mana saja dengan memanfaatkan ponsel, sehingga membantu mahasiswa maupun dosen dalam kegiatan sehari-hari.
2. Dengan pertimbangan akan keterbatasan yang dimiliki ponsel dan perlunya untuk mengetahui informasi yang diinginkan dengan segera, penggunaan LDAP dirasa sangat berguna dalam pengembangan aplikasi berbasis WAP ini, karena kemampuannya dalam hal proses *look-up* yang cepat serta transaksi yang tidak rumit dan data yang dipakai cenderung bersifat sederhana.

6.2 SARAN

Saran-saran untuk pengembangan selanjutnya adalah :

1. Untuk *record* orang/alamat akan sangat berguna bila mencakup semua jurusan yang ada, sehingga dapat dimanfaatkan oleh setiap orang di jurusan apapun. Hal ini bisa diterapkan dengan, misalnya, server untuk tiap jurusan diletakkan di jurusan masing-masing dengan memanfaatkan fasilitas *referral* yang dimiliki LDAP.

2. Penambahan fasilitas keamanan, misalnya dengan menggunakan *username* dan *password* sebelum dapat mengakses informasi yang tersedia.
3. Pemanfaatan SMS untuk informasi yang bersifat mendadak/segera, misalnya perubahan jadwal ujian atau panggilan terhadap seseorang mahasiswa.

DAFTAR PUSTAKA

- [LDAPDSERV] Shukla Shridhar, Deshpande Anand, ACM SIGMOD 2000 Dallas, USA, "LDAP Directory Services: Just Another Database Application?"
- [WAPARCH] WAP Forum, Ltd, (1998), *"Wireless Application Protocol : Architecture Specification"*
- [WAPWAE] WAP Forum, Ltd, (2000), *"Wireless Application Protocol : Wireless Application Environment Overview"*
- [WAPWML13] WAP Forum, Ltd, (2000), *"Wireless Application Protocol : Wireless Markup Language Specification Version 1.3"*
- [LDAPHOW] Malere, Luiz Ernesto Pinheiro, (2000-2001), *"LDAP Linux HOWTO v1.04"*
- [SCAL] Howes T A, Smith M C, April 28, 1995, "A Scalable, Deployable, Directory Service Framework for the Internet" URL : info.isoc.org/HMP/PAPER/173/html/paper.html
- [SLAPDUMICH] URL : <http://www.umich.edu/~dirsvcs/ldap/doc/guides/slapd/3.html>
- [LDAPMAN] University of Michigan, (1998-1999), *"LDAP Manual, Linux Documentation"*
- [RFC1777] W. Yeong, T. Howes, S. Kille, (1995), RFC1777: *"Lightweight Directory Access Protocol"*
- [RFC2254] T. Howes, (1997), RFC 2254 : *"The String Representation of LDAP Search Filters"*
- [PHPMAN] PHP Documentation Group, (1997, 1998, 1999, 2000), *"PHP Manual"*