

**RANCANG BANGUN SISTEM INFORMASI *YELLOW PAGES*
KOTA PEKANBARU DENGAN TEKNOLOGI *J2ME*
*BERBASIS MOBILE DEVICE***

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik pada
Jurusan Teknik Informatika

Oleh :

RIO SUNDARKO
NIM : 1045102559



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

RANCANG BANGUN SISTEM INFORMASI *YELLOW PAGES* KOTA PEKANBARU DENGAN TEKNOLOGI *J2ME* BERBASIS *MOBILE DEVICE*

RIO SUNDARKO
NIM : 10451025559

Tanggal Sidang : 24 Januari 2011

Periode Wisuda : Februari 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No.155 Pekanbaru

ABSTRAK

Pencarian nomor telepon seseorang ataupun perusahaan melalui buku *yellow pages* membutuhkan waktu yang lumayan lama. Tulisan yang kecil dan banyaknya data menjadi penyebab lamanya pencarian. Selain layanan buku telepon, PT. Telkom juga menyediakan layanan 108 dan *website yellow pages*. Tapi semua memiliki kelemahan masing-masing. Maka penelitian ini membahas perancangan dan implementasi sistem informasi *yellow pages* menggunakan *platform J2ME* dan jaringan *General Packet Radio Service (GPRS)* sebagai media pengirim dan penerima data. Sistem informasi *yellow pages* yang dibangun menggunakan *J2ME* tersebut bertindak sebagai *client*, sedangkan pada *server* menggunakan *Java Server Pages (JSP)* sebagai halaman untuk melakukan pemrosesan sistem, dan penghubung antara *database* dengan *client*. *Handphone* digunakan sebagai media pencarian data, sehingga pencarian dapat dilakukan kapan dan dimana saja. *Handphone* yang digunakan pada sistem informasi ini harus mendukung teknologi *Java* dan *GPRS*. Berdasarkan hasil pengujian, disimpulkan bahwa sistem ini dapat bekerja pada *handphone* apapun yang memiliki dukungan teknologi *Java* dan *GPRS*. Sehingga dapat diakses kapan dan dimana saja oleh masyarakat umum. Kekurangan pada sistem yang dibangun yaitu tidak dapat melakukan *data entry* melalui *handphone*.

Kata Kunci : Database, *GPRS*, *Handphone*, *Java*, *JSP*, *J2ME*.

***BUILD AND DESIGN YELLOW PAGES OF PEKANBARU
INFORMATION SYSTEM USING JAVA TECHNOLOGY AND
MOBILE DEVICE***

**RIO SUNDARKO
NIM : 10451025559**

Date of Final Exam : January 24th, 2011

Graduation Ceremony Period : February, 2011

*Technique Informatics Departement
Sciences and Technology Faculty
State Islamic University of Sultan Syarif Kasim Riau
Jl. Soebrantas No.155 Pekanbaru*

ABSTRACT

Search person's phone number or company through yellow pages book takes a fairly long time. Small writing and the wealth of data into the cause of the length of the search. In addition to the phone book service, PT. Telkom also provides services 108 and yellow pages websites. But all have weaknesses of each. So this study discusses the design and implementation of information systems yellow pages using J2ME platform and network General Packet Radio Service (GPRS) as the media sender and recipient data. Yellow pages information system built using J2ME are acting as a client, while on the server using Java Server Pages (JSP) as a page to do the processing system, and a liaison between the client database. Mobile search is used as a medium for data, so that searches can be done anytime and anywhere. Mobile is used on these information systems must support Java technology and GPRS. Based on test results, concluded that this system can work on any mobile phone that has support for Java technology and GPRS. So that can be accessed anytime and anywhere by the public. Lack of a system built that is unable to perform data entry via mobile phones.

Keywords : Database, GPRS, Handphone, Java, JSP, J2ME.

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	Error!
Bookmark not defined.	
LEMBAR PENGESAHAN	Error!
Bookmark not defined.	
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	Error!
Bookmark not defined.	
LEMBAR PERNYATAAN	Error!
Bookmark not defined.	
LEMBAR PERSEMBAHAN	Error!
Bookmark not defined.	
ABSTRAK	1
<i>ABSTRACT</i>	2
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	Error!
Bookmark not defined.	
DAFTAR SINGKATAN	Error!
Bookmark not defined.i	
DAFTAR LAMPIRAN.....	Error!
Bookmark not defined.	
BAB I Pendahuluan	I-1
1.1 Latar Belakang Masalah	I-1
1.2 Rumusan Masalah.....	I-8
1.3 Batasan Masalah	I-8
1.4 Tujuan	I-9
1.5 Manfaat Penelitian	I-9
1.6 Sistematika Pembahasan.....	I-10

BAB II Landasan Teori	II-1
2.1 Bahasa Pemrograman Java	II-1
2.1.1 Java versi lama	II-3
2.1.2 Java 1.2	II-5
2.1.2.1 <i>Java 2 Standard Edition (J2SE)</i>	II-5
2.1.2.2 <i>Java 2 Enterprise Edition (J2EE)</i>	II-6
2.1.2.2.1 Arsitektur dan komponen J2EE	II-7
2.1.2.2.2 <i>Java 2 Micro Edition (J2ME)</i>	II-10
2.2 Jaringan	II-15
2.2.1 Jaringan Kabel	II-15
2.2.1.1 Kabel <i>copper</i> (telepon)	II-15
2.2.1.2 Kabel <i>Coaxial</i>	II-15
2.2.1.3 Kabel Fiber Optic	II-16
2.2.2 Jaringan Nirkabel (<i>Wireless</i>)	II-16
2.2.2.1 Teknologi 802.11 (2,4 GHz)	II-16
2.2.2.1.1 <i>Bluetooth</i>	II-16
2.2.2.1.2 Wi-Fi (<i>Wireless Fidelity</i>)	II-17
2.2.2.2 Teknologi GSM	II-17
2.2.2.2.1 HSDPA	II-17
2.2.2.2.2 UMTS/WCDMA (3G)	II-18
2.2.2.2.3 GPRS	II-18
2.3 <i>Unified Modelling Language (UML)</i>	II-20
2.3.1 <i>Use case Diagram</i>	II-20
2.3.2 <i>Activity Diagram</i>	II-21
2.3.3 <i>Statechart Diagram</i>	II-22
2.3.4 <i>Class Diagram</i>	II-22
2.3.5 <i>Sequence Diagram</i>	II-23
2.3.6 <i>Deployment Diagram</i>	II-24
2.4 Object Oriented Analysis and Design (OOAD)	II-25
2.4.1 Sistem Berorientasi Objek	II-25
2.4.2 Analisis Berorientasi Objek	II-26

2.4.3 Desain Berorientasi Objek	II-27
BAB III Metodologi penelitian	III-1
3.1 Penelitian Pendahuluan.....	III-2
3.2 Identifikasi Masalah.....	III-2
3.3 Penetapan Tujuan.....	III-2
3.4 Pengumpulan Data.....	III-3
3.5 Analisa dan Perancangan	III-3
3.6 Implementasi dan Pengujian.....	III-4
BAB IV ANALISA DAN PERANCANGAN.....	IV-1
4.1 Deskripsi Sistem Yang Lama	IV-1
4.2 Deskripsi Sistem Yang Baru.....	IV-2
4.2.1 Sistem Yang Akan Dibangun	IV-4
4.2.1.1 Sistem Yang Dibangun Pada Sisi <i>Handphone</i>	IV-5
4.2.1.2 Sistem Yang Akan Dibangun Pada Sisi Komputer	IV-7
4.3 Fungsi Sistem.....	IV-8
4.3.1 Fungsi Sistem pada Sisi <i>Handphone</i>	IV-8
4.3.2 Fungsi Sistem pada Sisi <i>Server</i>	IV-8
4.4 Karakteristik Pengguna.....	IV-9
4.5 Pemodelan UML (<i>Unified Modelling Language</i>).....	IV-9
4.5.1 UML (<i>Unified Modelling Language</i>) Pada Sisi <i>Handphone</i>	IV-10
4.5.1.1 <i>Use Case Diagram</i>	IV-10
4.5.1.2 <i>Activity Diagram</i>	IV-10
4.5.1.3 <i>Statechart Diagram</i>	IV-12
4.5.1.4 <i>Class Diagram</i>	IV-13
4.5.1.5 <i>Sequence Diagram</i>	IV-14
4.5.1.6 <i>Deployment Diagram</i>	IV-17
4.5.2 UML (<i>Unified Modelling Language</i>) Pada Sisi <i>Server</i>	IV-18
4.5.2.1 <i>Use Case Diagram</i>	IV-18
4.5.2.2 <i>Activity Diagram</i>	IV-19
4.5.2.2.1 <i>Activity Diagram Login</i>	IV-19

4.5.2.2.2	<i>Activity Diagram</i> Ubah Password	IV-19
4.5.2.2.3	<i>Activity Diagram</i> Tambah Data	IV-20
4.5.2.2.4	<i>Activity Diagram</i> Ubah Data	IV-20
4.5.2.2.5	<i>Activity Diagram</i> Hapus Data	IV-21
4.5.2.3	<i>Statechart Diagram</i>	IV-21
4.5.2.3.1	<i>Statechart Diagram</i> Login	IV-21
4.5.2.3.2	<i>Statechart Diagram</i> Menampilkan Data Individu	IV-22
4.5.2.3.3	<i>Statechart Diagram</i> Menampilkan Data Perusahaan	IV-22
4.5.2.3.4	<i>Statechart Diagram</i> Mengubah Password	IV-22
4.5.2.3.5	<i>Statechart Diagram</i> Menambah Data	IV-23
4.5.2.3.6	<i>Statechart Diagram</i> Mengubah Data	IV-23
4.5.2.3.7	<i>Statechart Diagram</i> Menghapus Data	IV-23
4.5.2.4	<i>Class Diagram</i> Pada Sisi Server	IV-24
4.5.2.5	<i>Sequence Diagram</i>	IV-25
4.5.2.5.1	<i>Sequence Diagram</i> Login Administrator	IV-25
4.5.2.5.2	<i>Sequence Diagram</i> Ubah Password	IV-26
4.5.2.5.3	<i>Sequence Diagram</i> Pengelolaan Data Individu	IV-27
4.5.2.5.4	<i>Sequence Diagram</i> Pengelolaan Data Perusahaan	IV-28
4.5.2.6	<i>Deployment Diagram</i>	IV-29
4.6	Perancangan Antarmuka Pengguna Sistem	IV-29
4.6.1	Perancangan Menu Utama Pada <i>Handphone</i>	IV-30
4.6.2	Perancangan Antarmuka Keluaran Data pada <i>Handphone</i>	IV-31
BAB V	IMPLEMENTASI DAN PENGUJIAN	V-1
5.1	Implementasi	V-1
5.1.1	Lingkungan Implementasi	V-1
5.1.1.1	Perangkat Keras	V-1

5.1.1.2 Perangkat lunak	V-2
5.1.2 Batasan Implementasi.....	V-2
5.1.3 Hasil Implementasi.....	V-3
5.1.3.1 Hasil Implementasi Modul	V-3
5.1.3.1.1 Modul pada <i>Handphone</i>	V-3
5.1.3.1.2 Modul pada <i>Server</i>	V-3
5.1.3.2 Hasil Implementasi Layar.....	V-4
5.1.3.2.1 Implementasi Pada J2ME	
Wireless Toolkit	V-5
5.1.3.2.1.1 Menampilkan Splash	V-5
5.1.3.2.1.2 Menampilkan Menu Pencarian	
Data Individu	V-5
5.1.3.2.1.3 Menampilkan Menu Pencarian	
Data Perusahaan	V-6
5.1.3.2.1.4 Menampilkan <i>Form Input Keyword</i>	
Berdasarkan Nama Pada	
Data Individu	V-6
5.1.3.2.1.5 Menampilkan <i>Form Input Keyword</i>	
Berdasarkan Alamat Pada	
Data Perusahaan	V-7
5.1.3.2.1.6 Menampilkan <i>Form Input Keyword</i>	
Berdasarkan Nama Pada	
Data Individu	V-7
5.1.3.2.1.7 Menampilkan <i>Form Input Keyword</i>	
Berdasarkan Alamat Pada	
Data Perusahaan	V-8
5.1.3.2.1.8 Menampilkan Hasil Pencarian	
Berdasarkan Nama Pada	
Data Individu	V-8
5.1.3.2.1.9 Menampilkan Hasil Pencarian	

	Berdasarkan Alamat Pada	
	Data Individu	V-9
5.1.3.2.1.10	Menampilkan Hasil Pencarian	
	Berdasarkan Nama Pada	
	Data Perusahaan	V-9
5.1.3.2.1.11	Menampilkan Hasil Pencarian	
	Berdasarkan Alamat Pada	
	Data Perusahaan	V-10
5.1.3.2.2	Implementasi Pada <i>Handphone</i>	V-10
5.1.3.2.3	Implementasi Antarmuka <i>Administrator</i>	V-11
5.1.3.2.3.1	Antarmuka <i>Login Administrator</i>	V-11
5.1.3.2.3.2	Antarmuka Pengelolaan Data	
	Individu	V-11
5.1.3.2.3.3	Antarmuka Pengelolaan Data	
	Perusahaan	V-12
5.1.3.2.3.4	Antarmuka Ubah <i>Password</i>	
	<i>Administrator</i>	V-12
5.2	Pengujian Sistem.....	V-13
5.2.1	Perangkat Keras Pengujian.....	V-13
5.2.2	Identifikasi dan Rencana Pengujian	V-13
5.2.2.1	Pengujian Dengan <i>Handphone</i>	V-13
5.2.2.1.1	Pengujian Menampilkan Hasil Pencarian Data	
	Individu	V-19
5.2.2.1.2	Pengujian Menampilkan Hasil Pencarian Data	
	Perusahaan	V-20
5.2.3	Analisa Pengujian Sistem.....	V-21
BAB VI	PENUTUP	VI-1
6.1	Kesimpulan	VI-1
6.2	Saran	VI-2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Yellow Pages merupakan salah satu layanan yang diberikan kepada pelanggan TELKOM untuk mencari nama, alamat, nomor telepon baik individu maupun perusahaan yang dapat dicari melalui buku yang dikeluarkan oleh TELKOM untuk setiap tahunnya. Jika kita ingin mengetahui alamat seseorang maka kita dapat membuka buku *yellow pages* dan mencari nama seseorang berdasarkan urutan abjad yang telah disediakan. Data-data yang masuk ke dalam buku *yellow pages* merupakan data individu/perusahaan yang terdaftar sebagai pemilik nomor telepon milik TELKOM.

Data yang termuat di dalam buku *yellow pages* tersebut tidaklah sedikit, maka diperlukan kesabaran, ketelitian dan keuletan kita dalam mencari alamat seseorang. Selain data yang termuat dalam buku *yellow pages* banyak, huruf yang tertera juga kecil sehingga memerlukan waktu yang cukup lama untuk dapat menemukan alamat ataupun nomor telepon seseorang atau perusahaan yang kita inginkan.

Selain layanan *yellow pages* yang dikeluarkan oleh PT. Infomedia Nusantara, perusahaan ini juga memberikan layanan informasi 108 yang selama ini dikenal masyarakat sebagai *call center* untuk bertanya tentang nomor telepon pelanggan Telkom. Layanan 108 bisa diakses oleh para pelanggan dari seluruh operator, baik

yang menggunakan telepon rumah PSTN (*Public Switch Telephone Network*), *fixed wireless* seperti Flexi, GSM dan operator lain.

Dua produk yang dikeluarkan oleh PT. Infomedia Nusantara ini memiliki kelebihan dan kekurangan masing-masing. Untuk layanan buku *yellow pages*, *customer* dapat mengetahui informasi data individu dan data perusahaan. Data yang dikeluarkan oleh buku *yellow pages* tersebut berisi nama, alamat dan nomor telepon individu ataupun perusahaan. Untuk data individu yang terdapat di buku *yellow pages* dimulai dari abjad A dan diakhiri dengan abjad Z. Jika terdapat nama individu yang sama, maka kita dapat melihat alamat orang yang kita cari, jika sesuai maka kita dapat langsung mendapatkan nomor telepon orang tersebut. Untuk data perusahaan, dapat langsung dicari melalui urutan abjad, atau dapat dicari di lembaran *yellow pages* yang berada di akhir buku, di sini kita dapat melihat data perusahaan yang mencantumkan keunggulan perusahaannya. Bukan hanya nama perusahaan, alamat dan nomor telepon saja yang disediakan, tetapi juga didukung dengan gambar dan motto perusahaan. Jadi semakin mempermudah *customer* untuk mencari perusahaan yang sesuai dengan kebutuhan. Sedangkan untuk layanan 108 hanya menginformasikan tentang data perusahaan dan data layanan publik. Pihak *customer service* (CS) 108 akan memberikan nomor telepon perusahaan ataupun layanan public yang kita inginkan, dan jika diminta, maka pihak CS dapat memberikan alamat perusahaan yang kita cari.

Dari sisi kekurangan, layanan buku *yellow pages* membutuhkan waktu yang cukup banyak bagi *customer* untuk mencari informasi yang diinginkan karena harus

mencari satu persatu data penduduk ataupun data perusahaan yang diinginkan. Tetapi dengan layanan informasi 108, *customer* hanya menunggu sejenak karena operator akan mencari data yang diinginkan melalui database yang mereka miliki. Karena ramainya *customer* yang menelepon ke 108, maka kita harus cepat meminta nomor telepon perusahaan yang kita inginkan, dan jika kita tidak mendapatkan nomor telepon tersebut maka kita akan malas untuk kembali bertanya. Sedangkan untuk buku *yellow pages* kita dapat membolak-balik kertas apabila kita masih ragu untuk data perusahaan yang kita cari, kita tidak harus terburu-buru dengan waktu yang disediakan.

Sebagai penyedia nama, alamat dan nomor telepon individu dan perusahaan terbesar di Indonesia, sudah seharusnya *yellow pages* mengembangkan sistem yang ada. Dimana memberikan kemudahan bagi *user* (pengguna) untuk dapat mengakses secara langsung, cepat dan tepat nama, alamat dan nomor telepon individu ataupun perusahaan yang mereka inginkan. Dengan memanfaatkan teknologi informasi yang berkembang saat ini, maka *yellow pages* dapat memasukkan data-data nama, alamat, nomor telepon individu dan perusahaan ke dalam database berbasis *web* yang terkoneksi *internet* yang dapat diakses melalui *handphone*.

Handphone adalah salah satu alat komunikasi yang dapat dibawa kemanapun oleh penggunanya. Ditambah lagi sekarang ini, hampir semua jenis *handphone* telah membuat aplikasi layanan GPRS sehingga memudahkan bagi user untuk mengakses *internet* dari *handphone* mereka. Peluang ini harus dimanfaatkan oleh *yellow pages*

dimana pelayanan pencarian nama, alamat dan nomor telepon individu ataupun perusahaan dapat diakses melalui *handphone*.

Yellow pages telah memiliki *website* sendiri yaitu www.yellowpages.co.id, tetapi *website* ini tidak memiliki *database* nama, alamat, nomor telepon individu yang berada di daerah Kota Pekanbaru. Sehingga, user tidak dapat menemukan data individu yang berada di Pekanbaru. Maka diperlukan *database* data-data individu yang terdaftar di *yellow pages*. Salah satu bahasa pemrograman yang dapat digunakan untuk membangun sistem informasi tersebut adalah *Java Server Pages* (JSP).

Selain itu, penggunaan komputer sebagai media pengaksesan dirasakan kurang optimal, karena sekarang ini lebih banyak orang-orang memilih menggunakan layanan *internet* yang tersedia di dalam *handphone* mereka dibandingkan harus pergi ke warung *internet* (*warnet*). Oleh karena itu, dibutuhkan suatu media yang banyak digunakan oleh orang dan dapat digunakan untuk mengakses *internet*, salah satu dari media tersebut adalah *handphone*.

Untuk membangun sistem informasi *yellow pages* Kota Pekanbaru yang dapat diakses melalui *handphone* dibutuhkan suatu bahasa pemrograman yang khusus, karena *handphone* berbeda dengan komputer, salah satu perbedaannya adalah *handphone* memiliki memori yang terbatas. Bahasa pemrograman yang dapat digunakan untuk membangun sistem informasi *yellow pages* Kota Pekanbaru pada *handphone* adalah *Java* menggunakan teknologi *Java 2 Micro Edition* (J2ME). J2ME merupakan lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak *Java* pada alat elektronik beserta perangkat pendukungnya. J2ME biasa

digunakan pada *handphone*, *pager*, *Personal Digital Assistants* (PDA) dan sejenisnya.

Beberapa penelitian tentang pengembangan aplikasi dan sistem pada *handphone* telah banyak menggunakan bahasa pemrograman *J2ME*. Beberapa contoh judul penelitian yang menggunakan teknologi *J2ME* sebagai bahasa pemrogramannya yaitu: “Emulasi Aplikasi Pemantauan Ruang Melalui *Handphone* Menggunakan *Webcam*” (Putro dkk, 2008) dan “Aplikasi Pelaporan Berita Emergeni Secara Visual dan Tekstual Lewat *Handphone*” (Santoso dkk, 2008)

Dalam penelitian tentang implementasi teknologi GPRS dan *Java 2 Micro Edition* untuk aplikasi pemantauan ruangan melalui *handphone* menggunakan *webcam* dijelaskan bahwa *Webcam* yang terhubung dengan *server* melakukan proses *capture* ruangan pada periode waktu tertentu (Putro dkk, 2008). *Handphone* yang telah mendukung teknologi *Java* dan GPRS dipakai untuk mengakses hasil *capture* gambar yang telah disimpan. Pengujian dilakukan dengan menggunakan *emulator* Nokia 3300 dalam sebuah *local area network*. Waktu *transfer* rata-rata untuk gambar berukuran 36 KB pada adalah 0,9 detik. Dari hasil pengujian, disimpulkan bahwa sistem ini cocok jika dipakai untuk melakukan *monitoring* dengan *frame rate* yang rendah.

Sedangkan penelitian Santoso adalah tentang pelaporan berita emergenasi secara *visual*. Sebenarnya pelaporan dapat dilakukan dengan menggunakan banyak media seperti *email*, SMS dan telepon akan tetapi *email* dan telepon relatif sulit dijangkau pada saat situasi emergenasi tersebut terjadi. Sedangkan SMS hanya dapat melaporkan

situasi emergensi tersebut hanya berupa teks saja (Putro dkk, 2008). Aplikasi ini memberikan alternatif lain untuk mempermudah dan mempercepat proses pelaporan situasi emergensi dengan menggunakan *handphone* ke sebuah *website* secara *visual* dan tekstual. Implementasinya menggunakan *database* pada *server*, menggunakan *protocol* HTTP pada *platform* J2EE, dan MIDlet pada *handphone* untuk mengirim berita emergensi melalui *General Packet Radio Service* (GPRS) dengan menggunakan *platform Java 2 Micro Edition* (J2ME).

Pada penelitian sistem informasi *yellow pages* ini, akan dilakukan perancangan sistem informasi dengan mengambil data yang berada pada buku *yellow pages* dan menambah media informasi (komputer) dengan media informasi yang bersifat *mobile* yang banyak digunakan oleh orang dan mudah dibawa kemanapun, dalam hal ini adalah *handphone*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka permasalahan dalam tugas akhir ini adalah, "Bagaimana merancang dan membangun sistem informasi *Yellow Pages* Kota Pekanbaru agar dapat diakses melalui *mobile device* menggunakan bahasa pemrograman *Java* dan JSP ? "

1.3 Batasan Masalah

Agar lebih terarah dan tidak menyimpang dari tujuan semula, maka penulis membuat batasan-batasan. Penulis menyadari keterbatasan dan kemampuan penulis

dalam menalar serta mencurahkan buah pikiran. Oleh karena itu penulisan Tugas Akhir ini dibatasi:

1. Data yang akan digunakan berasal dari data yang termuat dalam buku *Yellow Pages* yang dikeluarkan oleh PT. Infomedia Nusantara tahun 2010.
2. Jumlah data yang digunakan adalah 1000 data.
3. Tidak menggunakan teknologi WAP.
4. Jenis *handphone* yang digunakan adalah *handphone* yang didukung oleh fasilitas GPRS dan *Java*.

1.4 Tujuan

Penelitian yang dilakukan bertujuan:

1. Merancang dan membangun sistem informasi *Yellow Pages* dengan menggunakan teknologi J2ME pada *handphone* serta membangun sistem untuk pengelolaan *database* menggunakan JSP.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Memberikan kemudahan bagi setiap orang untuk mencari informasi mengenai data individu yang tinggal di Kota Pekanbaru.
2. Tidak memerlukan jasa penyewaan warnet (*warung internet*) untuk mengakses sistem informasi *Yellow Pages*, karena hanya dengan

menggunakan *handphone*, kita mendapatkan informasi data yang kita cari tersebut.

3. Biaya yang dikeluarkan untuk mendapatkan informasi data relatif lebih murah dibandingkan dengan jasa penyewaan *internet*.
4. Waktu yang diperlukan untuk mendapatkan informasi data yang dicari menjadi lebih cepat.

1.6 Sistematika Pembahasan

Sistematika pembahasan tugas akhir ini dibagi menjadi 6 (enam) bab. Setiap bab terdiri dari sub bab dan penjelasan yang tersusun sehingga mudah untuk dipahami. Berikut penjelasan tentang masing-masing bab:

BAB I Pendahuluan

Merupakan deskripsi umum dari tugas akhir ini, yang meliputi: latar belakang masalah, rumusan masalah, batasan masalah, tujuan penyusunan tugas akhir serta sistematika pembahasan tugas akhir.

BAB II Landasan Teori

Berisi penjelasan tentang teori dasar yang akan diterapkan dalam tahap analisis sistem. Teori-teori tersebut, antara lain: bahasa pemrograman *Java* (J2ME, JSP), dan GPRS.

BAB III Metodologi Penelitian

Dalam bab ini menjelaskan mengenai cara yang dilakukan dalam menyelesaikan persoalan yang menjadi objek penelitian.

BAB IV Analisis dan Perancangan

Berisi pembahasan mengenai deskripsi kebutuhan sistem dan perancangan komponen.

BAB V Implementasi dan Pengujian

Berisi pembahasan mengenai lingkungan pengembangan aplikasi dan hasil pengujian aplikasi.

BAB VI Penutup

Dalam bab ini akan dijelaskan beberapa kesimpulan yang didapatkan dari pembahasan tentang sistem informasi akademik yang dibangun, serta saran untuk pengembangan selanjutnya.

BAB II

LANDASAN TEORI

Teori-teori yang digunakan sebagai referensi pada Tugas Akhir ini adalah bahasa pemrograman *Java*, Jaringan, dan *Unified Modelling Language* (UML).

2.1 Bahasa Pemrograman Java

Java dibuat dan diperkenalkan pertama kali oleh sebuah tim *Sun Microsystems* yang dipimpin oleh Patrick Naughton dan James Goslin pada tahun 1991 dengan *code name* Oak. Tahun 1995 *Sun* mengubah nama *Oak* tersebut menjadi *Java*. Ide pertama kali pembuatan Java adalah motivasi untuk membuat sebuah bahasa pemrograman yang bersifat *portable* dan *platform independent* (tidak tergantung mesin atau sistem operasi) yang dapat digunakan untuk membuat piranti lunak yang dapat ditanamkan (*embedded*) pada berbagai macam peralatan elektronik biasa, seperti: *microwave*, *remote control*, telepon, *card reader* dan sebagainya (Indrajani dan Martin, 2004).

Pada saat pertama kali rancangan *Java* dibuat, jaringan internet juga mulai berkembang pesat, yang lebih dikenal dengan *World Wide Web* (*WWW*) atau *Web*. Ide portabilitas dan *platform independent* yang dimiliki oleh Java juga merupakan hal yang diperlukan oleh *Web*. Java semakin terangkat popularitasnya sebagai bahasa pemrograman *Web* yang *portable* akibat dari semakin terkenalnya *Web*, terutama dengan diperkenalkannya *applet*, yang merupakan suatu aplikasi yang didesain

sedemikian rupa untuk ditransmisikan melalui *internet* dan dieksekusi oleh *Web Browser* yang *compatible* dengan *Java*.

Ketenaran *Java* sebagai bahasa pemrograman terkadang mengakibatkan kerancuan, yaitu menganggap *Java* sama dengan *JavaScript*. Walaupun keduanya menggunakan bahasa pemrograman yang mirip, namun secara esensi keduanya berbeda. *JavaScript* merupakan *scripting language* yang digabung dengan *HTML*, sehingga memungkinkan suatu *web page* mampu berinteraksi lebih baik dengan penggunanya. *JavaScript* tidak memerlukan kompilasi karena hanya diinterpretasikan oleh *web browser*. *Java* merupakan *full programming language*, secara esensi sama dengan bahasa pemrograman lainnya seperti *Visual Basic* dan *C++*, yaitu kode *Java* yang dibuat harus dikompilasi menjadi bentuk yang dapat dieksekusi oleh *run-time* sistem *Java* (*Java Virtual Machine*). Program *Java* yang dibuat sebagai *applet* tidak dieksekusi oleh *web browser*, *web browser* hanya bertindak sebagai *container* yang memungkinkan *applet* tersebut dieksekusi.

Hasil kompilasi *Java* adalah berupa *bytecode*, yang merupakan sekumpulan instruksi yang teroptimasi. Diperlukan suatu program khusus untuk menerjemahkan dan mengeksekusi setiap baris instruksi agar instruksi tersebut dapat dieksekusi layaknya program biasa, program inilah yang disebut dengan *Java Virtual Machine* (*JVM*) yang merupakan interpreter dari *bytecode*. Proses konversi dari *source code* menjadi *bytecode*, memungkinkan suatu program dapat dieksekusi di lingkungan yang berbeda (*multiplatform*), yang diperlukan hanyalah *JVM* yang harus diimplementasikan untuk setiap *platform* dimana program tersebut akan dieksekusi,

sedangkan *bytecode*-nya tetap sama. Hal ini akan memudahkan *programmer* karena tidak perlu lagi memikirkan dimana program yang dibuat akan dieksekusi.

Java menurut definisi dari *Sun* adalah nama dari sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java berdiri diatas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine (JVM)*. *JVM* inilah yang akan membaca *bytecode* dalam *file .class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, dengan syarat pada sistem operasi tersebut terdapat *JVM* (Solahuddin dan Rosa A.S, 2006)

Java telah mengalami beberapa pengembangan, yaitu (Raharjo, 2007):

1. Java versi lama
2. Java 1.2 (Java 2)

2.1.1 Java versi lama

Java merupakan bagian dari riset Sun untuk menciptakan perangkat lunak untuk perangkat keras elektronik. Tujuan Java pada saat itu adalah menjadi perangkat lunak yang kecil, efisien dan *portable* untuk berbagai perangkat keras. Java yang mempergunakan nama Oak pada mulanya kurang mendapat perhatian komersial sampai akhirnya diperkenalkan bersama HotJava dan Netscape Incorporated.

Sun menyediakan perangkat lunak untuk membuat program java dengan nama *Java Development Kit (JDK)* yang mempunyai beberapa versi dengan *platform* untuk

SPARC dan X86 Solaris, Windows NT dan Windows 95. JDK dimulai dengan versi 1.0x kemudian 1.1x dan yang terakhir versi 1.2 (disebut dengan Java 2).

Versi awal Java ditahun 1996 sudah merupakan versi *release* sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya, yaitu :

1. `java.lang` digunakan untuk kelas elemen-elemen dasar.
2. `java.io` digunakan untuk kelas *input* dan *output*, termasuk penggunaan berkas.
3. `java.util` digunakan untuk kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. `java.net` digunakan untuk kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. `java.awt` digunakan untuk kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. `java.applet` digunakan untuk kelas dasar aplikasi antar muka untuk diterapkan pada *web browser*.

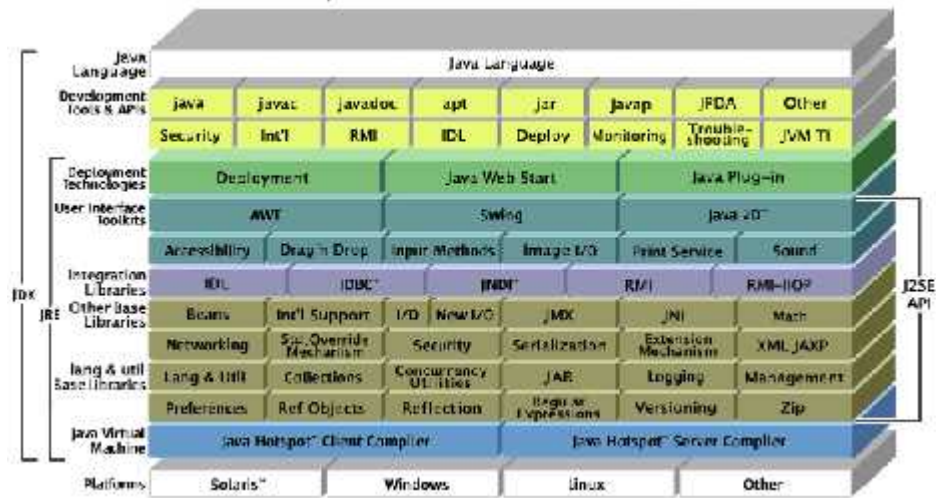
2.1.2 Java 1.2

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2 yaitu (Raharjo dkk, 2007) :

1. *Java 2 Standard Edition (J2SE)*, yang digunakan untuk mengembangkan aplikasi-aplikasi *desktop* dan *applet* (aplikasi Java yang dapat dijalankan di dalam *web browser*).
2. *Java 2 Enterprise Edition (J2EE)*, merupakan *superset* dari J2SE yang digunakan untuk mengembangkan aplikasi-aplikasi berskala besar (*enterprise*) seperti pembuatan aplikasi-aplikasi di sisi *server* dengan menggunakan *Enterprise JavaBeans*, aplikasi *Servlet* dan *JSP*, dan teknologi lainnya.
3. *Java 2 Micro Edition (J2ME)*, merupakan *subset* dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat-perangkat kecil yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh.

2.1.2.1 Java 2 Standard Edition (J2SE)

J2SE adalah inti dari bahasa pemrograman *Java*. *Java Development Kit (JDK)* merupakan salah satu *tool* dari J2SE untuk mengkompilasi dan menjalankan program *Java*. *Tool* J2SE salah satunya adalah *JDK1.5*. Gambar 2.1 di bawah ini menunjukkan arsitektur J2SE.



Gambar 2.1 Arsitektur J2SE (Sumber : www.write-technical.com, 2008)

2.1.2.2 Java 2 Enterprise Edition (J2EE)

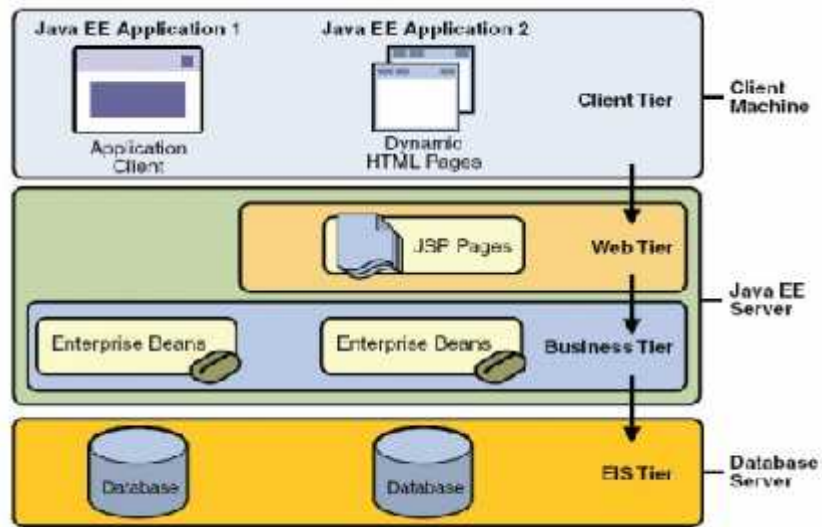
Java 2 Enterprise Edition (J2EE) adalah *platform* teknologi Java yang dikembangkan untuk mendukung aplikasi *web* berbasis Java. J2EE merupakan paket yang berisi beberapa *class* dan *interface* untuk mengembangkan aplikasi berbasis *web*, misalnya *class Servlet*, *Java Server Pages*, dan *JavaBeans*.

Secara khusus, J2EE merupakan model aplikasi terdistribusi *multitier* dengan kemampuan untuk menggunakan kembali komponen yang telah dibuat, integrasi dengan XML (*eXtensible Markup Language*), model *security*, serta kontrol transaksi yang fleksibel.

2.1.2.2.1 Arsitektur dan komponen J2EE

J2EE menggunakan model aplikasi terdistribusi *multitier*. Pada umumnya, aplikasi *multitier* J2EE terdiri atas *3-tier* sehingga aplikasi akan didistribusikan pada tiga tempat yang berbeda, yaitu *computer client*, *computer server*, serta *database* (totoltotol.wordpress.com, 2008).

Aplikasi yang dibangun menggunakan J2EE terdiri atas komponen-komponen yang sering juga disebut dengan *J2EE Component*. Komponen J2EE merupakan Unit-unit program yang memiliki fungsi dan dapat berkomunikasi dengan komponen lain sehingga dapat digabungkan untuk membangun aplikasi J2EE yang lebih besar. Gambar 2.2 di bawah ini menunjukkan arsitektur J2EE.



Gambar 2.2 Arsitektur J2EE (Sumber : totoltotol.wordpress.com, 2008)

Berikut ini adalah penjelasan dari gambar 2.2 :

1. *Client Machine*

Client Machine adalah bagian yang berinteraksi langsung dengan *user*. Java menyediakan dua solusi untuk bagian ini yaitu dengan menggunakan aplikasi Java yang diletakkan pada *client* dan dapat terhubung ke *server* atau menggunakan halaman *web* HTML dinamik.

2. *Java EE server*

Java EE Server dibagi menjadi dua bagian, yaitu *web tier* dan *business tier*. Pada *web tier* terdapat aplikasi JSP (*Java server pages*). *Web tier* ini lebih ke arah *web* dinamik (dapat menggantikan *web* HTML dinamik di komputer *client*). *Business tier* adalah sebuah abstraksi dari kegiatan-kegiatan yang ada dalam proses bisnis di perusahaan. Pada *Java EE server* terdapat program Java seperti beans, dan servlet, dan masing-masing program tersebut adalah sebuah fungsi dalam proses bisnis. Hal ini sesuai dengan fungsi J2EE sebagai sebuah *IT system* yang mendukung bisnis perusahaan. *Java EE server* merupakan sebuah *server* yang mendukung teknologi Java, contohnya adalah *GlassFish* (*sun application server* pada Netbeans), dan Apache Tomcat.

3. *Database server*

Database server adalah tempat data-data perusahaan disimpan, seperti *oracle* atau merupakan EIS (*Enterprise Information Sistem*) internal perusahaan. *Database server* menyediakan konektivitas dari aplikasi Java kepada *database* atau sistem internal perusahaan, contohnya *Java database connector*.

J2EE merupakan standar referensi Sun's untuk pengembangan perusahaan, pertama disampaikan pada desember 1999, untuk versi 1.3 elemen-elemen intinya sebagai berikut (Mardiyah, 2008) :

1. Bahasa Java

Java merupakan bahasa berorientasi objek berasal dari C++ dengan fitur pengkodean yang sederhana seperti manajemen memory melalui *gerbage colection, no pointer* dan lain-lain. Java dirancang untuk "*write once, run anywhere*", tujuan akan tercapai menggunakan *bytecode portable*.

2. JVM/JRE

JRE (*Java Runtime Environment*) berisi *Java Virtual Machine*, seperti saat *compiler* dan *class-class* dasar. *Class* Java dikompilasi dalam *platform* tidak bergantung *bytecode* yang dieksekusi dalam JVM.

3. JSP dan Servlet

Java Server Pages (JSP) analogi dari teknologi ASP, menyediakan kemampuan untuk membangun komposisi *web pages* dinamis pada HTML dengan menempelkan komponen dinamis, misalnya referensi beans. Servlet dijelaskan sebagai applet yang dijalankan sebagai *web server* dan secara tradisional *Comman Gateway Interface* (CGI) digunakan untuk membangun aplikasi *web* yang dinamis.

4. EJB

Enterprise Java Beans (EJB) digunakan untuk membangun aplikasi terdistribusi yang menyediakan *framework* komunikasi dan eksekusi untuk komponen

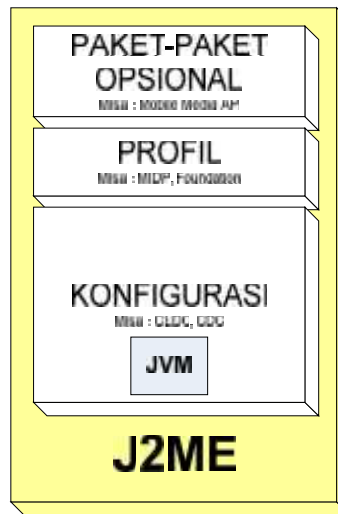
terdistribusi. Layanan kritis yang diberikan oleh *container* EJB antara lain: manajemen transaksi, keamanan, manajemen *resource* dan *persistence*.

5. JDBC

Java Database Connectivity (JDBC) merupakan sekumpulan API yang dihubungkan dengan *database* relasional, memanipulasi isinya, dan memproses *output* (keluaran) dengan pernyataan SQL. Banyak *vendor database* yang telah dikembangkan berbasis JDBC API.

2.1.2.2.2 *Java 2 Micro Edition (J2ME)*

(Raharjo, 2007) menyebutkan bahwa *J2ME* merupakan sebuah kombinasi yang terbentuk antara sekumpulan *interface* Java yang sering disebut dengan Java API (*Application Programming Interface*) dengan JVM yang didesain khusus untuk alat, yaitu JVM dengan ruang yang terbatas KVM (*Kilobyte Virtual Machine*). Kombinasi tersebut kemudian digunakan untuk melakukan pembuatan aplikasi-aplikasi yang dapat berjalan diatas alat elektronik beserta perangkat pendukungnya dalam hal ini perangkat *mobile device*. Gambar 2.3 di bawah ini menunjukkan arsitektur *J2ME*.



Gambar 2.3 Arsitektur J2ME (Sumber : Raharjo, 2007)

Komponen teknologi J2ME adalah sebagai berikut :

1. Konfigurasi (*configuration*) terdiri dari *library* dasar dan *virtual Machine* untuk banyak jenis *device*, JVM yang dimaksud tidak seperti JVM yang terdapat pada J2SE, melainkan JVM yang telah didesain secara khusus untuk alat.

Konfigurasi J2ME terdiri atas 2 jenis, yaitu:

- a. CDC (*Connected Device Configuration*), meliputi PDA *high-end*, navigasi *mobile*, dan sebagainya.
 - b. CLDC (*Connected Limited Device Configuration*), meliputi *mobile phone*, PDA pada umumnya, *pager*, dan sebagainya.
2. Profil (*profile*) merupakan tambahan API (*Application Programming Interface*) dan spesifikasi lainnya untuk membuat aplikasi untuk tipe *device* yang lebih spesifik. Terdapat beberapa profil seperti pada gambar, namun yang lebih sering

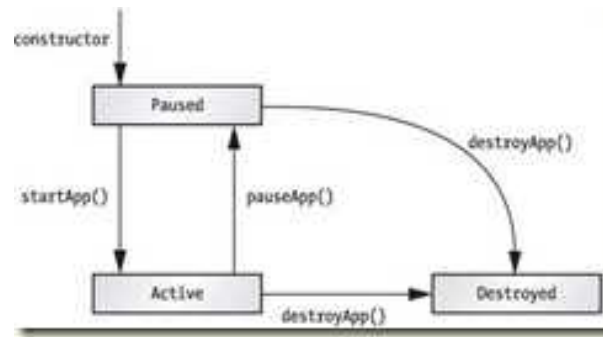
dipakai adalah MIDP (*Mobile Information Device Profile*). Berikut ini beberapa profil yang tersedia di dalam J2ME :

- a. MIDP, yaitu profil yang menyediakan beberapa *library* Java untuk implementasi dasar antarmuka (GUI), implementasi jaringan (*networking*), *database* dan *timer*.
 - b. PADP (*Personal Digital Assistant Profile*), yaitu profil untuk PDA yang memperluas fungsi-fungsi pada konfigurasi CLDC, dan digunakan khusus untuk menambahkan kemampuan-kemampuan lebih apabila dibandingkan dengan penggunaan profil MIDP.
 - c. *Foundation Profile*, yaitu profil yang digunakan untuk konfigurasi CDC, profil ini menambahkan beberapa kelas dari J2SE ke dalam konfigurasi CDC, dan berperan juga sebagai pondasi untuk membentuk profil baru lainnya.
 - d. *Personal Profile*, yaitu profil yang mendefinisikan ulang persoalan Java sebagai profil yang dapat digunakan sebagai profil dalam J2ME, profil ini merupakan hasil perluasan dari *Foundation Profile*.
 - e. *RMI Profile*, yaitu profil yang menambahkan dukungan RMI (*Remote Metode Invocation*) ke dalam konfigurasi CDC.
3. Paket-paket Opsional merupakan paket-paket tambahan yang dibutuhkan oleh aplikasi sehingga pada saat *deployment* paket-paket tersebut dapat didistribusikan juga sebagai bagian dari aplikasi yang dibuat.

Pada penelitian ini, profil yang digunakan adalah MIDP dan menggunakan konfigurasi CLDC. API yang disediakan untuk CLDC 1.1 dan MIDP 2.0 adalah sebagai berikut:

1. CDLC 1.1
 - a. java.lang
 - b. java.lang.ref
 - c. java.io
 - d. java.util
 - e. javax.microedition.io
2. MIDP 2.0
 - a. javax.microedition
 - b. javax.microedition.media
 - c. javax.microedition.midlet
 - d. javax.microedition.rms

Setiap aplikasi J2ME (*Midlet*) akan berada dalam salah satu dari tiga *state* yaitu *Paused*, *Active*, atau *Destroyed*. Saat *constructor* dipanggil, maka *Midlet* akan berada dalam kondisi *Paused*. Jika metode `startApp()` dipanggil, maka *Midlet* akan aktif, sebaliknya, jika *Midlet* menemui metode `destroyApp()`, maka *Midlet* akan di buang dari *memory* (*Destroyed*). Gambar 2.4 di bawah ini menunjukkan siklus hidup J2ME (tukangjava.wordpress.com, 2008).



Gambar 2.4 Siklus hidup J2ME (Sumber : tukangjava.wordpress.com, 2008)

Terdapat 3 metode yang harus ada dalam setiap *Midlet*.

1. `startApp()`

Agar *MIDlet* memperoleh fokus dan menjadikannya *active*. Metode ini dipanggil jika *MIDlet* baru saja dibuat atau kembali dari kondisi *Paused*.

2. `pauseApp()`

Metode ini dieksekusi jika *MIDlet* tidak memiliki fokus, misalnya jika ada telepon masuk saat menjalankan *MIDlet* tersebut maka aplikasi tidak memiliki tampilan.

3. `destroyApp()`

Metode ini dipakai untuk membebaskan seluruh *resource* yang digunakan oleh *MIDlet*.

2.2 Jaringan

Tipe jaringan yang digunakan sangat mempengaruhi faktor-faktor lain seperti kecepatan, fleksibilitas dan biaya. Ada 2 tipe jaringan yang digunakan untuk membuat suatu jaringan yaitu jaringan kabel dan jaringan nirkabel (*wireless*) (Wibisono, 2008).

2.2.1 Jaringan Kabel

Jaringan kabel akan menurunkan nilai fleksibilitas yang dimiliki jaringan tersebut, karena keberadaan jaringan akan bergantung kepada keberadaan dari kabel fisik.

2.2.1.1 Kabel *copper* (telepon)

Kabel ini merupakan kabel telepon biasa yang pada umumnya dipakai untuk *dial-up* menggunakan *modem* atau *modem* DSL. Instalasi kabel ini biayanya cukup murah namun memiliki masalah “*the last mile*”. Semakin jauh dari protokolnya, maka kecepatan yang ada akan semakin menurun.

2.2.1.2 Kabel *Coaxial*

Kabel ini adalah kabel yang biasanya dipakai untuk koneksi *cable*, baik untuk TV kabel maupun untuk *cable modem*. Instalasi kabel ini cukup mahal, dapat melalui bawah tanah atau menggunakan tiang untuk jaringan fisiknya. Kabel ini tidak memiliki masalah “*the last mile*”.

2.2.1.1. Kabel *Fiber Optic*

Kabel ini merupakan kabel yang memiliki kemampuan untuk mentransfer data dengan kecepatan cahaya, sehingga banyak digunakan sebagai jaringan *backbone* suatu negara. Investasinya paling mahal diantara jaringan kabel lainnya.

2.2.2 Jaringan Nirkabel (*Wireless*)

Jaringan nirkabel akan meningkatkan fleksibilitas namun menurunkan tingkat keamanan, reliabilitas serta kecepatan yang dapat dicapai jika dibandingkan dengan jaringan kabel. Meskipun demikian, jaringan nirkabel berkembang sedemikian rupa untuk meningkatkan kecepatan (4G, HSDPA) dan keamanannya (*Wireless Equivalent Privacy* dan *Wi-Fi Protected Access*). Jaringan nirkabel, berdasarkan teknologinya dibagi menjadi Teknologi 802.11 dan Teknologi GSM.

2.2.2.1 Teknologi 802.11 (2,4 GHz)

Teknologi 802.11 (2,4 GHz) terdiri dari *Bluetooth* dan Wi-Fi (*Wireless Fidelity*).

2.2.2.1.1 Bluetooth

Bluetooth merupakan nama yang diberikan untuk teknologi baru dengan menggunakan *short-range radio links* untuk menggantikan koneksi kabel *portable* atau alat elektronik yang sudah pasti. Tujuannya adalah mengurangi kompleksitas, *power* serta biaya. Bluetooth diimplementasikan pada tempat-tempat yang tidak mendukung sistem *wireless* seperti di rumah atau di jalan untuk membentuk *Personal*

Area Networking (PAN), yaitu peralatan yang digunakan secara bersama-sama. Bluetooth merupakan solusi untuk jaringan terbatas, baik dari sisi kecepatan dan area cakupannya. Biasa dipakai untuk menghubungkan perangkat-perangkat dalam satu area kecil.

2.2.2.1.2 Wi-Fi (*Wireless Fidelity*)

Wi-Fi mempunyai kecepatan yang lebih tinggi dibandingkan Bluetooth dan area cakupan yang luas. Area cakupan (*hotspots*) bisa diperluas menggunakan *access points*. Kelemahan dari Wi-Fi selain kecepatan dan keamanan, termasuk banyaknya interferensi sinyal 2.4 GHz yang ada di publik.

2.2.2.2 Teknologi GSM

Teknologi 802.11 (2,4 GHz) terdiri dari HSDPA (*High Speed Download Packet Access*), UMTS/WCDMA (3G) dan GPRS.

2.2.2.2.1 HSDPA

Penggunaan jaringan GSM dengan kecepatan transmisi data *mobile* maksimum 14.4 Mbps *downlink* dan 768 Kbps *uplink*. Perkembangan HSDPA (*High Speed Download Packet Access*) generasi ketiga, yaitu HSOPA (Super 3G) mempunyai kecepatan *downlink* 200 Mbps dan *uplink* 100 Mbps. HSDPA dikenal dan digolongkan ke dalam teknologi 3.5G, karena kecepatan transfer datanya lebih bagus dibanding 3G. 3G pada permulaannya maksimal sampai 384 Kbit/s, sedangkan dengan HSDPA dapat dinaikkan menjadi 14 Mbit/s.

2.2.2.2.2 UMTS/WCDMA (3G)

Penggunaan jaringan *Global Sistem for Mobile Communications* (GSM) dengan kecepatan transmisi data *mobile* maksimum 384 kbps *downlink*. *Universal Mobile Telecommunications Sistem* (UMTS) / *Wideband Code Division Multiple Access* (WCDMA) mempunyai layanan data secara teori sampai 11 Mbit/s, meskipun dalam perkembangannya yang disebarkan ke pengguna dalam jaringan performanya hanya mencapai 384 Kbit/s untuk *handset* R99 dan 1-2 Mbit/s untuk *handset* HSDPA untuk koneksi *downlink*.

2.2.2.2.3 GPRS

General Packet Radio Service (GPRS) merupakan sistem transmisi berbasis paket untuk GSM. Dalam teorinya GPRS menjanjikan kecepatan mulai dari 56 Kbps sampai 115 Kbps, sehingga memungkinkan akses *internet*, pengiriman data multimedia ke komputer, *notebook* dan *handheld computer*. Namun, dalam implementasinya, hal tersebut sangat tergantung faktor-faktor sebagai berikut:

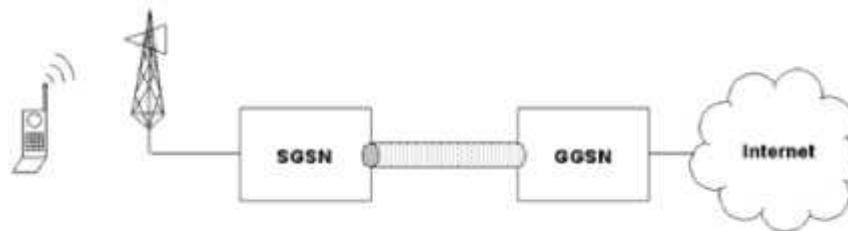
1. Konfigurasi dan alokasi *time slot* pada level BTS
2. *Software* yang dipergunakan
3. Dukungan fitur dan aplikasi *handphone* yang digunakan

Hal ini menjelaskan mengapa pada saat-saat tertentu dan di lokasi tertentu akses GPRS terasa lambat.

GPRS menggunakan sistem komunikasi *packet switch* sebagai cara untuk mentransmisikan datanya. *Packet switch* adalah sebuah sistem dimana data yang akan

ditransmisikan dibagi menjadi bagian-bagian kecil (paket) lalu ditransmisikan dan diubah kembali menjadi data semula. Sistem ini dapat mentransmisikan ribuan bahkan jutaan paket per detik. Transmisi dilakukan melalui PLMN (*Public Land Mobile Network*) dengan menggunakan IP *backbone*. Karena memungkinkan untuk pemakaian kanal transmisi secara bersamaan oleh pengguna lain maka biaya akses GPRS, secara teori, lebih murah daripada biaya akses CSD.

GPRS didesain untuk menyediakan layanan *transfer* paket data pada jaringan GSM dengan kecepatan yang lebih baik dari GSM. Kecepatan yang lebih baik ini didapat dengan menggunakan *coding scheme* (CS) yang berbeda dari GSM. Gambar 2.5 di bawah ini menunjukkan arsitektur GPRS.



Gambar 2.5 Arsitektur GPRS (Sumber : www.linuxjournal.com, 2008)

Komponen-komponen utama jaringan GPRS adalah :

- a. *GGSN (Gateway GPRS Support Node)* merupakan gerbang penghubung jaringan GPRS ke jaringan *internet*.
- b. *SGSN (Serving GPRS Support Node)* merupakan gerbang penghubung jaringan *BSS/BTS* ke jaringan GPRS. Komponen ini berfungsi untuk mengantarkan paket data ke MS (*Mobile Station*).

2.3 Unified Modelling Language (UML)

UML adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifacts* (model, deskripsi, atau *software*) dari sistem *software*, untuk memodelkan bisnis, dan sistem *non-software* lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks (Suhendar dan Hariman Gunadi, 2002).

UML memiliki beberapa diagram antara lain: *use case diagram*, *activity diagram*, *statechart diagram*, *class diagram*, *sequence diagram*, dan *deployment diagram*.

2.3.1 Use case Diagram

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menganalisa kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja (Suhendar dan Hariman Gunadi, 2002). Selama tahap desain, *use case diagram* menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case diagram*. Contoh *use case diagram* dapat dilihat pada gambar 2.6.

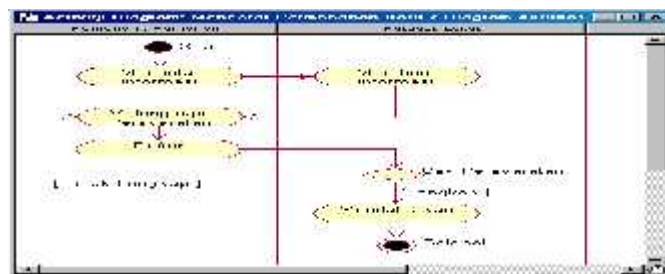


Gambar 2.6

Contoh *use case diagram* (Sumber : Suhendar dan Hariman Gunadi, 2002)

2.3.2 *Activity Diagram*

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses (Suhendar dan Hariman Gunadi, 2002). Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas kedalam keadaan sesaat (*state*). *Activity diagram* bermanfaat untuk membantu memahami proses secara keseluruhan. *Activity diagram* juga berguna ketika ingin menggambarkan perilaku paralel atau bagaimana menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Contoh *activity diagram* dapat dilihat pada gambar 2.7.

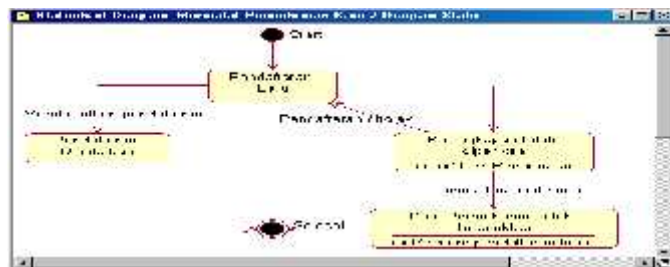


Gambar 2.7 Contoh *activity diagram*

(Sumber: Suhendar dan Hariman Gunadi, 2002)

2.3.3 Statechart Diagram

Statechart diagram digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas (Suhendar dan Hariman Gunadi, 2002). *Statechart diagram* khususnya digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek, sedangkan *activity diagram* lebih cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses. Contoh *statechart diagram* dapat dilihat pada gambar 2.8.



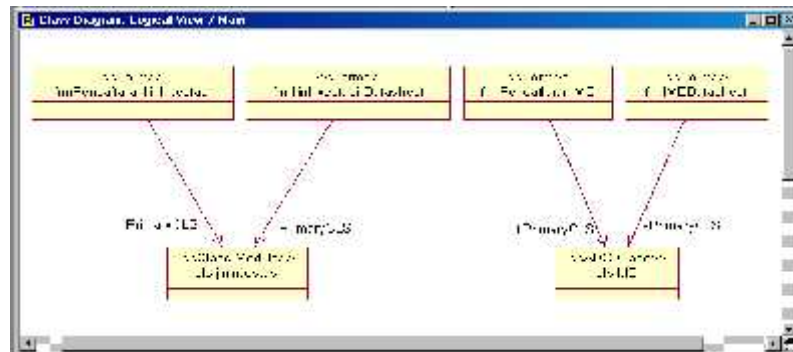
Gambar 2.8 Contoh *statechart diagram*

(Sumber : Suhendar dan Hariman Gunadi, 2002)

2.3.4 Class Diagram

Class diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe *diagram* yang banyak dipakai. *Class diagram* memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Selama proses analisis, *class diagram*

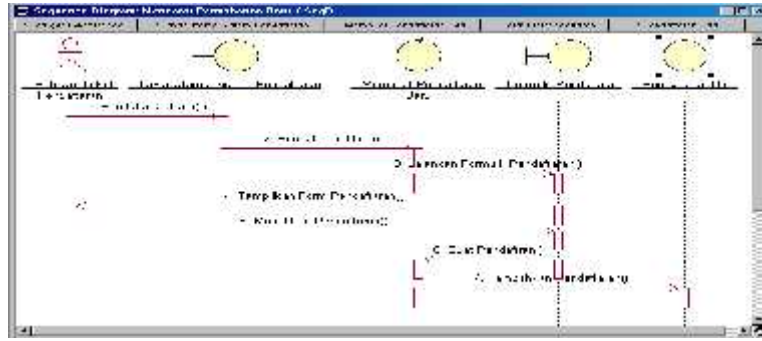
memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku system (Suhendar dan Hariman Gunadi, 2002). Selama tahap desain, *class diagram* berperan dalam menganalisa struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Contoh *class diagram* dapat dilihat pada gambar 2.9.



Gambar 2.9 Contoh *class diagram* (Sumber : Suhendar dan Hariman Gunadi, 2002)

2.3.5 *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case* (Suhendar dan Hariman Gunadi, 2002). Tipe diagram ini sebaiknya digunakan pada awal tahap desain atau analisis karena kesederhanaannya dan mudah dimengerti. Contoh *sequence diagram* dapat dilihat pada gambar 2.10.

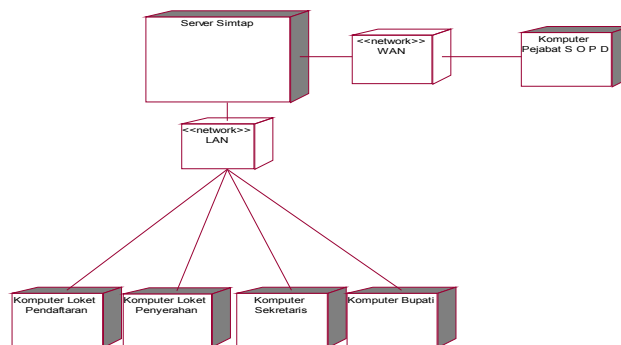


Gambar 2.10 Contoh *sequence diagram*

(Sumber : Suhendar dan Hariman Gunadi, 2002)

2.3.6 *Deployment Diagram*

Setiap model hanya memiliki satu *deployment diagram*. Diagram ini memperlihatkan pemetaan *software* kepada *hardware*. Diagram ini bermanfaat ketika membuat model suatu sistem yang diterapkan dalam lingkungan arsitektur yang terdistribusi dimana penerapan aplikasi dan *server* pada lokasi yang berbeda (Suhendar dan Hariman Gunadi, 2002). Contoh *deployment diagram* dapat dilihat pada gambar 2.11.



Gambar 2.11 Contoh *deployment diagram*

(Sumber : Suhendar dan Hariman Gunadi, 2002)

2.4. Object Oriented Analysis and Design (OOAD)

Di dalam Wikipedia dijelaskan bahwa *Object oriented analysis and design* (OOAD) adalah pendekatan rekayasa perangkat lunak yang model sistem sebagai sekelompok objek yang berinteraksi. Setiap objek yang mewakili beberapa entitas yang menarik dalam sistem yang dimodelkan, dan dicirikan oleh kelas, para negara (elemen data), dan perilaku. Berbagai model dapat dibuat untuk menunjukkan struktur statis, dinamis perilaku, dan run-time penyebaran benda berkolaborasi ini. Ada beberapa notasi yang berbeda untuk mewakili model ini, seperti Unified Modeling Language (UML) (http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design).

Analisis berorientasi obyek (OOAD) berlaku model objek-teknik untuk menganalisis persyaratan fungsional sistem. Object-oriented design (OOD) menguraikan model-model analisis untuk menghasilkan spesifikasi implementasi. OOA fokus terhadap sistem apa yang bekerja dan OOD focus terhadap bagaimana sistem tersebut bekerja (http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design).

2.4.1. Sistem Berorientasi Objek

Sebuah sistem berorientasi objek terdiri dari objek. Perilaku sistem hasil dari kerjasama dari objek tersebut. Kolaborasi antara obyek melibatkan mereka mengirimkan pesan satu sama lain. Mengirim pesan berbeda dari memanggil salah satu fungsi dalam bahwa ketika sebuah benda sasaran akan menerima pesan, ia

sendiri memutuskan apa fungsi untuk melaksanakan layanan pesan itu. Pesan yang sama dapat dilaksanakan oleh berbagai fungsi yang berbeda, yang dipilih tergantung pada keadaan objek target.

Pelaksanaan "mengirim pesan" bervariasi tergantung pada arsitektur sistem yang dimodelkan, dan dengan lokasi objek yang dikomunikasikan.

2.4.2. Analisis Berorientasi Objek

Analisis berorientasi obyek (OOA) melihat pada masalah domain, dengan tujuan untuk memproduksi sebuah model konseptual informasi yang ada di daerah yang sedang dianalisis. Model analisis tidak mempertimbangkan kendala-kendala pelaksanaan yang mungkin ada, seperti concurrency, distribusi, ketekunan, atau bagaimana sistem harus dibangun. Kendala pelaksanaan ditangani selama desain berorientasi objek (OOD).

Sumber-sumber untuk analisis dapat persyaratan tertulis pernyataan, dokumen visi yang formal, wawancara dengan stakeholder atau pihak yang berkepentingan lainnya. Sebuah sistem dapat dibagi menjadi beberapa domain, yang mewakili bisnis yang berbeda, teknologi, atau bidang yang diminati, masing-masing dianalisis secara terpisah.

Hasil analisis berorientasi objek adalah deskripsi dari apa sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih kelas UML diagram, dan sejumlah interaksi diagram. Mungkin juga mencakup

beberapa jenis antarmuka pengguna mock-up. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan.

2.4.3 Desain Berorientasi Objek

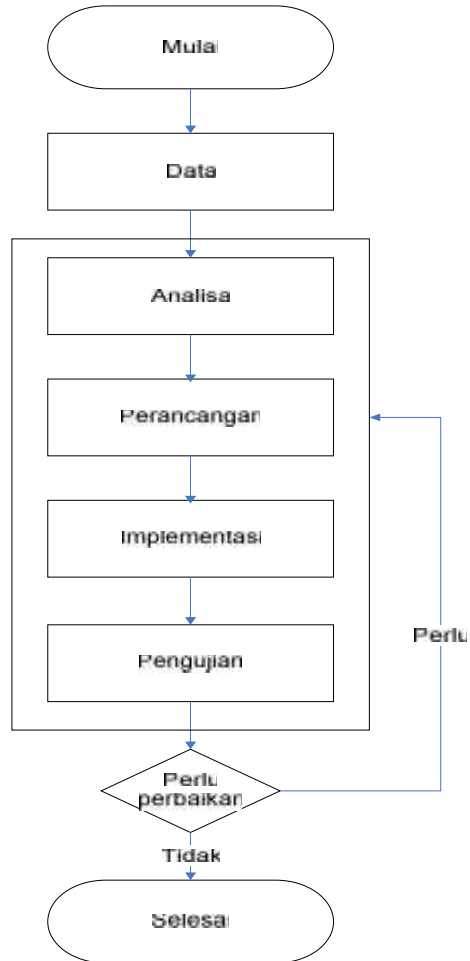
Object-oriented design (OOD) mengubah model konseptual yang dihasilkan dalam analisis berorientasi objek memperhitungkan kendala yang dipaksakan oleh arsitektur yang dipilih dan setiap non-fungsional - teknologi atau lingkungan - kendala, seperti transaksi throughput, response time, run - waktu platform, lingkungan pengembangan, atau bahasa pemrograman.

Konsep dalam model analisis pelaksanaan dipetakan ke kelas dan interface. Hasilnya adalah model solusi domain, sebuah penjelasan rinci tentang bagaimana sistem harus dibangun.

BAB III

METODOLOGI PENELITIAN

Secara garis besar langkah-langkah penelitian dapat dilihat sebagai berikut.



Gambar 3.1 Langkah-langkah penelitian

Gambar 3.1 pada langkah analisa dan perancangan sampai dengan implementasi dan pengujian selanjutnya disebut dengan metode pengembangan sistem. Pengembangan sistem yang akan dilakukan menggunakan metode *Object*

Oriented Analysis Design (OOAD). Tahap-tahap yang dilakukan pada pengembangan sistem adalah menganalisa data-data yang telah dikumpulkan, perancangan, penulisan kode program dan terakhir pengujian. Apabila pada setiap tahap yang dilakukan memiliki kekurangan maka harus kembali ke tahap sebelumnya untuk menyelesaikannya, apabila telah selesai maka dapat melanjutkan ke tahap selanjutnya.

3.1 Penelitian Pendahuluan

Di dalam penelitian, dilakukan penelitian tahap awal untuk mencari informasi-informasi awal yang terdapat pada buku yellow pages, website yellow pages. Informasi-informasi ini akan digunakan untuk mengidentifikasi masalah.

3.2 Identifikasi Masalah

Dengan memanfaatkan informasi-informasi yang didapat dari penelitian pendahuluan dan studi pustaka yang telah dilakukan, maka dilakukan tahap berikutnya yaitu mengidentifikasi masalah. Pada tugas akhir ini masalah yang akan diidentifikasi adalah bagaimana membangun sebuah Sistem Informasi Yellow Pages dengan teknologi J2ME.

3.3 Penetapan Tujuan

Penetapan tujuan sangat diperlukan untuk menjawab permasalahan yang ada. Penetapan tujuan dilakukan setelah mengidentifikasi masalah. Tujuan akan ditetapkan dengan cara mengetahui dan menentukan apa saja yang perlu dipertahankan,

ditingkatkan, dihilangkan, dievaluasi dan diperbarui masalah yang ada dapat teratasi. Dipertahankan misalnya sistem yang akan dibangun mempunyai cara kerja yang sama dengan sistem yang lama. Ditingkatkan misalnya apakah harus menambah jenis media pengkasesan. Dievaluasi maksudnya mengevaluasi hal-hal yang harus dipertahankan, ditingkatkan dan dihilangkan, apakah hal-hal tersebut menghasilkan tujuan sesuai dengan yang diinginkan. Hasil evaluasi beberapa hal tersebut harus mendapatkan suatu solusi apakah sistem harus diperbarui atau tidak.

3.4 Pengumpulan Data

Pengumpulan data diperoleh dengan cara :

- a. Mencari informasi yang terdapat pada buku yellow pages Kota Pekanbaru.
- b. Wawancara adalah metode pengumpulan data yang dilakukan dengan melakukan tanya jawab pada narasumber, khususnya tanya jawab cara kerja sistem informasi yang sudah ada. Dalam hal ini, yang menjadi narasumber adalah karyawan PT. Telkom Kota Pekanbaru.
- c. Studi pustaka dimaksudkan untuk memperoleh teori-teori dan konsep-konsep yang mendasar mengenai materi yang berhubungan dalam pengembangan Sistem Informasi Kependudukan yaitu dengan cara mempelajari teknologi Java dari buku-buku, artikel-artikel, jurnal-jurnal dan media lainnya.

3.5 Analisa dan Perancangan

Analisa dilakukan setelah data yang dikumpulkan telah lengkap agar selanjutnya mulai merancang sebuah Sistem Informasi Yellow Pages dengan teknologi J2ME serta pengelolaan datanya pada *server*. Pada saat menganalisa data, ada beberapa tahap yang harus dilakukan, yaitu mengidentifikasi kebutuhan sistem, fungsi sistem, memodelkan sistem yang akan dibangun, karakteristik pengguna, memodelkan UML, merancang lingkungan implementasi, serta merancang antarmuka pengguna sistem yang akan dibangun. Kebutuhan sistem yang diidentifikasi adalah kebutuhan sistem yang sedang berjalan dan kebutuhan sistem yang akan dibangun. Secara umum fungsi sistem dibagi menjadi dua bagian yaitu fungsi sistem yang dibangun pada sisi *handphone* dan fungsi sistem yang dibangun pada sisi *server*. Pengguna sistem terbagi menjadi dua kategori yaitu *administrator* dan penduduk. Analisa pada sistem menggunakan UML, yaitu terdiri dari *use case diagram*, *activity diagram*, *statechart diagram*, *class diagram* dan *sequence diagram*, dan *deployment diagram*. Perancangan lingkungan implementasi digunakan untuk mengetahui perangkat lunak apa saja yang diperlukan pada tahap implementasi. Perancangan antarmuka pengguna sistem digunakan untuk memudahkan tahap implementasi pada saat membangun antarmuka pengguna.

3.6 Implementasi dan Pengujian

Tahap terakhir yaitu pembuatan sistem berdasarkan analisa yang telah dilakukan. Tahap implementasi merupakan tahap penulisan kode program

berdasarkan analisa dan perancangan yang telah dilakukan. Bahasa pemrograman yang digunakan untuk membangun sistem adalah J2ME pada *handphone* dan JSP pada *server*. Selanjutnya dilakukan pengujian terhadap sistem yang telah dibangun agar dapat diketahui hasilnya. Pengujian dilakukan dengan beberapa *handphone* sehingga dapat diketahui apakah berhasil koneksi ke dalam Sistem Informasi Yellow Pages dan dapat mengakses data penduduk yang dicari.

BAB IV

ANALISA DAN PERANCANGAN

Analisa dan perancangan merupakan tahap yang dilakukan setelah semua data yang diperlukan terkumpul. Setelah data terkumpul, dilakukan proses analisa kebutuhan-kebutuhan sistem lama sebagai acuan untuk membangun sistem baru.

4.1 Deskripsi Sistem Yang Lama

Sistem informasi *yellow pages* telah ada di situs www.yellowpages.co.id. Sistem informasi *yellow pages* yang telah ada hanya dapat diakses secara maksimal melalui komputer atau laptop menggunakan jaringan *internet*. Hal ini karena website tersebut memang ditujukan untuk pengguna komputer. Jika kita membuka melalui *handphone* hasil tampilan dari proses tidak maksimal. Tampilan yang keluar jika membuka situs ini melalui *handphone* adalah tulisan membesar melebihi layar *handphone* dan membutuhkan *download* data yang besar sehingga memakan biaya yang besar juga. Selain itu sistem informasi *yellow pages* tersebut hanya berisi data perusahaan dikota-kota besar Indonesia. Jika *user* ingin mencari data individu melalui sistem yang sudah ada maka hasil pencarian tidak ditemukan

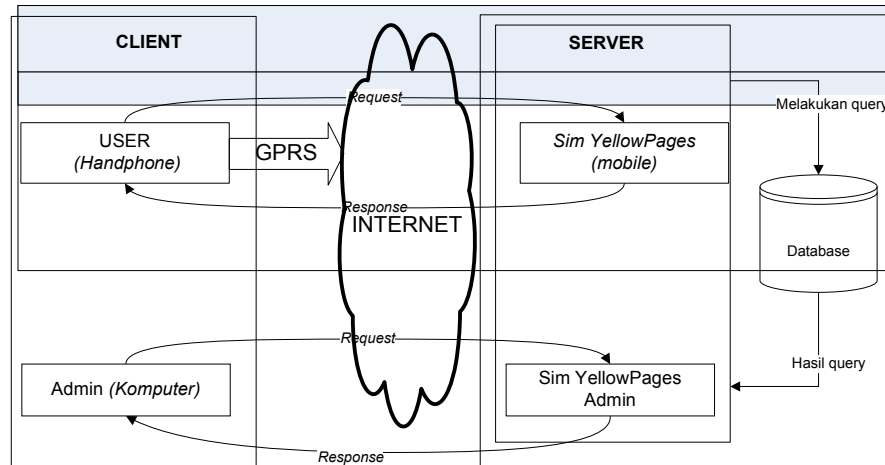
Keadaan pada sistem yang sedang berjalan adalah:

1. Bahasa pemrograman yang digunakan adalah PHP dan *database* MySQL.
2. Pada hak akses *user*, sistem dapat digunakan untuk menampilkan data perusahaan yang telah terdata oleh sistem informasi *yellow pages* yaitu mengenai nama, alamat dan nomor telepon.

3. Untuk masuk ke dalam sistem harus membuka website *yellow pages* terlebih dahulu.
4. Hanya menampilkan data perusahaan yang berada di kota-kota besar di Indonesia.
5. Tidak semua data perusahaan dipekanbaru ada didalam *database*.
6. Pada hak akses *administrator*, sistem dapat digunakan untuk pengelolaan *database* seperti mengelola data perusahaan yang berada di kota-kota besar dan belum memasukkan data-data individu seperti yang tertera pada buku *yellow pages*.

4.2 Deskripsi Sistem Yang Baru

Deskripsi sistem dimaksudkan untuk memberikan gambaran mengenai modul-modul yang akan dirancang pada sistem ini dan hubungan antara modul-modul tersebut akan mempermudah proses perancangan sistem yang akan dibangun. Sistem pada sisi *handphone* yang akan dirancang adalah modul-modul untuk mengirimkan *parameter* ke *server* dan modul untuk melakukan proses *parsing* hasil *query* yang dikirimkan *server*. Sistem pada sisi *server* yang akan dirancang adalah modul-modul untuk pemrosesan *query* dengan *parameter* yang dikirimkan melalui *handphone* dan beberapa *query* untuk pengelolaan *database*. Blok diagram sistem informasi *yellow pages* yang akan dibangun dapat dilihat pada gambar 4.1.



Gambar 4.1 Blok diagram sistem informasi *yellow pages*.

Blok diagram diatas menjelaskan bahwa *client* melakukan *request* ke *server* dengan dua media yaitu *handphone* dan komputer. Pada dasarnya setiap media yang digunakan memiliki cara kerja yang sama ketika mengakses sistem informasi *yellow pages*, yang membedakan hanya jaringan yang digunakan, media *handphone* menggunakan jaringan GPRS terlebih dahulu sedangkan media komputer menggunakan jaringan *internet* langsung. Cara kerja sistem informasi *yellow pages* ini adalah:

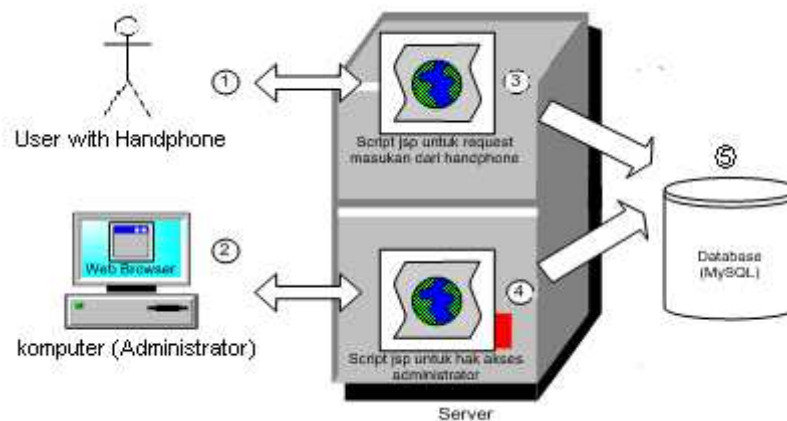
1. *Client* melakukan *request* ke *server*.
2. *Server* memeriksa permintaan *query* dari *client*.
3. *Server* melakukan *query* ke *database*.
4. *Server* memberikan respon hasil *query* kepada *client*.

Sebelum pengaksesan terhadap sistem, *user* harus membuka Sistem Informasi *yellow pages* terlebih dahulu kemudian memasukkan nama atau alamat individu atau perusahaan yang akan dicari. Jika nama atau alamat yang dicari

tidak terdaftar, *server* akan menolak pengaksesan terhadap data individu atau perusahaan yang dicari.

4.2.1 Sistem yang Akan Dibangun

Model sistem yang akan dibangun ini dirumuskan sebagai fungsi yang menggambarkan hubungan antar objek-objek yang berperan dalam sistem.



Gambar 4.2 Model Sistem

Objek-objek yang berperan pada model sistem diatas, yaitu :

1. *Handphone* sebagai media untuk mengakses sistem informasi individu dengan hak akses *user*, didalamnya di-*install*-kan file `.jar` yang dibuat. Tidak ada dilakukan *setting konfigurasi* pada *handphone*, karena setting pada program *java* telah bernilai *default*. Fungsi yang dapat dilakukan oleh *handphone* adalah menampilkan data nama, alamat, nomor telepon individu atau perusahaan.
2. Komputer (*administrator*) menggunakan *Web browser* sebagai media untuk mengakses sistem informasi *yellow pages* dengan hak akses

administrator. Fungsi yang dapat dilakukan adalah mengelola semua data yang digunakan untuk pemrosesan pada sistem.

3. *Script JSP* untuk *request* masukan dari *handphone* sebagai antarmuka antara *handphone* dengan *database*.
4. *Script JSP* untuk hak akses *administrator* sebagai antarmuka untuk mengelola *database* pada hak akses *administrator* untuk pengaksesan menggunakan *web browser*.
5. *Database* (MySQL) sebagai tempat penyimpanan data yang akan digunakan untuk pemrosesan pada sistem.

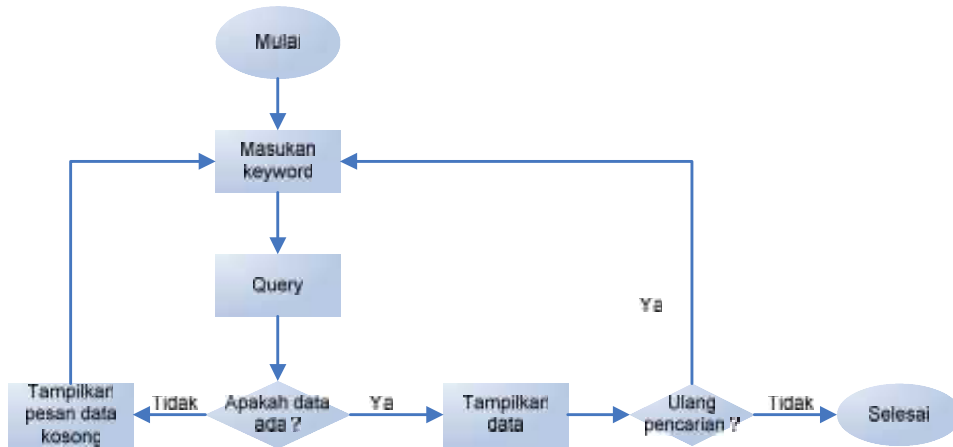
Kebutuhan sistem pada sistem yang akan dikembangkan terbagi menjadi dua bagian yaitu kebutuhan sistem pada *handphone* dan kebutuhan sistem pada komputer dengan hak akses *administrator*.

4.2.1.1 Sistem Yang Dibangun Pada Sisi *Handphone*

Sistem pada sisi *handphone* dibangun khusus untuk pengguna yang akan melakukan pencarian data individu ataupun perusahaan pada sistem informasi *yellow pages* melalui *handphone*.

Kebutuhan sistem pada *handphone* yaitu :

1. Bahasa pemrograman yang digunakan adalah *J2ME*.
2. Untuk dapat mengakses sistem informasi *yellow pages* harus membuka website *yellow pages* terlebih dahulu dengan menggunakan GPRS.
3. Untuk dapat menampilkan data individu atau perusahaan, *user* harus mengetikkan kata kunci yang telah disediakan sesuai data yang ingin dicari.



Gambar 4.3 Alur Proses Pada *Handphone*

Penjelasan dari gambar diatas adalah :

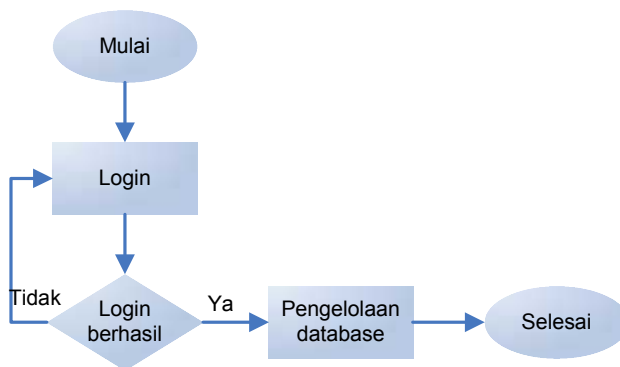
1. Proses **Mulai** : *User* membuka aplikasi *Yellow Pages* yang ada di *Handphone*
2. Proses **Masukan Keyword** : *User* memasukkan kata kunci untuk pencarian data. Kata kunci yang akan disediakan antara lain nama dan alamat. Pada pencarian data perusahaan ada pilihan jenis perusahaan barang atau jasa.
3. Proses **Query** : Data yang dimasukkan oleh *user* akan dikirim ke server dan server melakukan proses *query* ke *database*.
4. Proses **Tampilkan pesan data kosong** : Apabila hasil dari *query* tidak menemukan data yang dicari, maka sistem akan mengirimkan pesan kepada *user* bahwa data tidak ditemukan.
5. Proses **Tampilkan data** : Jika proses *query* mendapatkan data yang dicari maka data tersebut akan ditampilkan. Data yang ditampilkan antara lain nama, alamat dan nomor telepon. Sedangkan pada data perusahaan akan ditampilkan jenis usaha.

4.2.1.2 Sistem Yang Akan Dibangun Pada Sisi Komputer

Sistem pada sisi server akan dijalankan menggunakan komputer dengan hak akses terbatas. Sistem ini hanya dapat digunakan oleh *administrator* dari sistem informasi *yellow pages*. Sistem ini berbasis *web* dan menggunakan jaringan *internet* sebagai media penghubungnya.

Kebutuhan sistem pada komputer dengan hak akses *administrator*, yaitu :

1. Bahasa pemrograman yang digunakan adalah JSP dan *database* MySQL.
2. Untuk *Login* dengan cara mengisi *User ID* dan *password*.
3. Untuk memasukkan data individu atau perusahaan, *administrator* harus login terlebih dahulu.
4. Sistem pada hak akses digunakan untuk pengelolaan *database* seperti mengelola data individu dan data perusahaan.



Gambar 4.4 Alur Proses Pada Komputer *Administrator*

Penjelasan dari gambar diatas adalah :

1. Proses ***Login*** : Proses ini hanya dapat dilakukan oleh *administrator* sistem. Administrator harus memasukkan *user name* dan *password* dengan benar.

2. Proses **Login berhasil** : Proses ini memeriksa apakah data *login* yang dimasukkan oleh *administrator* benar atau tidak. Jika benar, maka administrator dapat memasuki sistem *database*. Jika tidak maka sistem akan mengulang proses *login*.
3. Proses **Pengelolaan database** : Proses ini hanya dilakukan oleh *administrator*. Proses yang terjadi antara lain menambah data, mengedit data dan menghapus data.

4.3 Fungsi Sistem

Secara umum fungsi sistem dibagi menjadi dua bagian yaitu fungsi sistem yang dibangun pada sisi *handphone* dan fungsi sistem yang dibangun pada sisi *server*.

4.3.1 Fungsi Sistem pada Sisi *Handphone*

Sistem yang dibangun pada sisi *handphone* memiliki fungsi-fungsi sebagai berikut:

1. Menampilkan data individu berupa nama, alamat dan nomor telepon.
2. Menampilkan data perusahaan berupa nama, alamat, nomor telepon dan jenis usaha.

4.3.2 Fungsi Sistem pada Sisi *Server*

Sistem yang dibangun pada sisi *server* memiliki fungsi-fungsi sebagai berikut:

1. Mengelola *database* yaitu menambah, mengubah dan menghapus data yang terdapat didalam *database*.

2. Sebagai antarmuka penghubung antara *handphone* dengan *database*.

4.4 Karakteristik Pengguna

Pengguna sistem terbagi menjadi dua kategori yaitu *administrator* dan *user*. *Administrator* hanya dapat menggunakan komputer sedangkan *user* hanya dapat menggunakan *handphone*. Tabel 4.1 menjelaskan hak akses pengguna ke sistem.

Tabel 4.1 Karakteristik Pengguna

No.	Kategori Pengguna	Hak akses ke sistem	Keterangan
1.	<i>User (Handphone)</i>	a. Menampilkan nama individu dan perusahaan b. Menampilkan alamat individu dan perusahaan c. Menampilkan no telepon individu dan perusahaan	Mempunyai hak akses terbatas yaitu hanya untuk proses mencari dan menampilkan.
2.	<i>Administrator (computer)</i>	a. <i>Login</i> dengan <i>User ID</i> dan <i>password</i> yang dimiliki <i>administrator</i> b. Mengelola seluruh data yang akan digunakan pada sistem informasi <i>yellow pages</i>	Memiliki hak akses penuh terhadap sistem antara lain menambah, mengedit dan menghapus data.

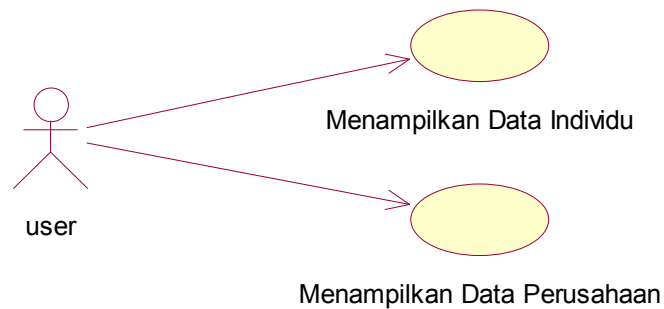
4.5 Pemodelan UML (*Unified Modelling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *use case diagram*, *activity diagram*, *statechart diagram*, *class diagram* dan *sequence diagram*, *deployment diagram*. UML ini terbagi menjadi dua bagian yaitu UML pada sisi *Handphone* dan UML pada sisi *Server*.

4.5.1 UML (*Unified Modelling Language*) Pada Sisi *Handphone*

4.5.1.1 *Use Case Diagram*

Use case diagram menggambarkan hubungan dari setiap objek yang ada pada sebuah sistem. Gambar 4.5 dibawah ini menggambarkan aliran *use case diagram* untuk pengaksesan melalui *handphone*.



Gambar 4.5 Aliran *use case diagram* (pengaksesan melalui *Handphone*)

Deskripsi gambar 4.5 tentang *use case diagram* (pengaksesan melalui *handphone*) dapat dilihat pada tabel 4.2.

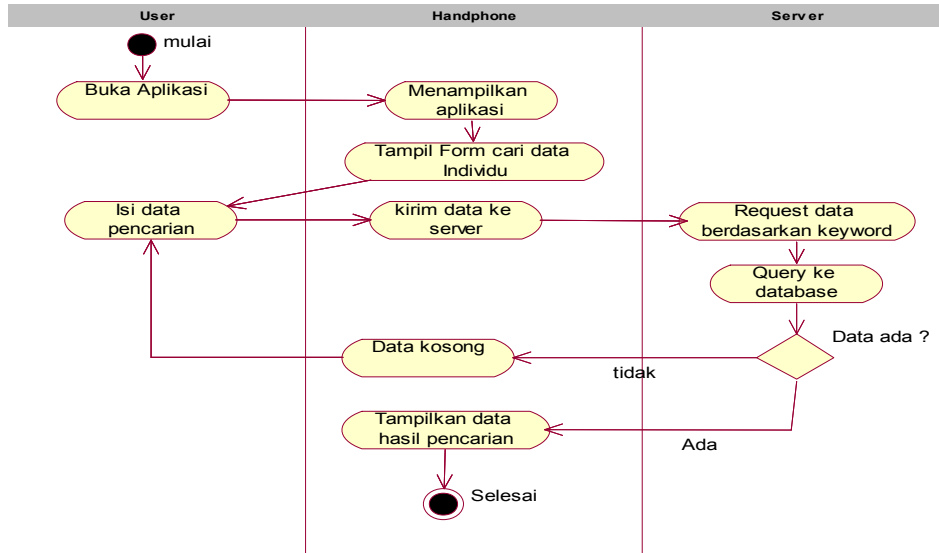
Tabel 4.2 *Use case diagram* (pengaksesan melalui *handphone*)

No	Nama <i>Use Case</i>	Deskripsi
1.	Menampilkan Data Individu	Proses untuk Menampilkan hasil dari pencarian data individu.
2.	Menampilkan Data Perusahaan	Proses untuk Menampilkan hasil dari pencarian data perusahaan

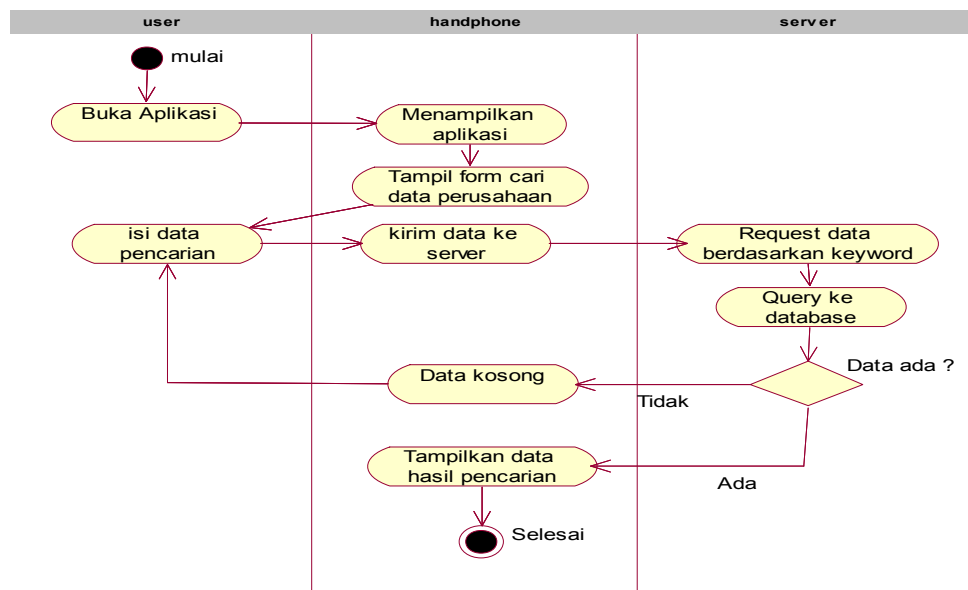
4.5.1.2 *Activity Diagram*

Activity Diagram merupakan alur kerja yang terjadi pada setiap *use case*. *Activity diagram* yang ada pada analisa ini mencakup *activity diagram* setiap *use*

case. Gambar 4.6 dibawah ini menjelaskan *activity diagram* menampilkan data individu menggunakan *handphone*.



Gambar 4.6 *Activity diagram* menampilkan data individu



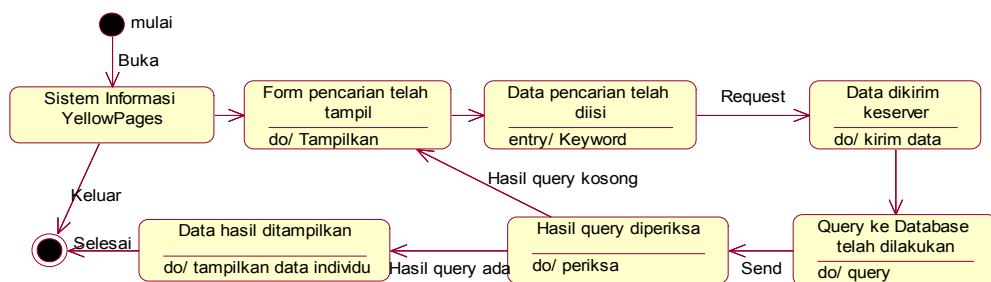
Gambar 4.7 *Activity diagram* menampilkan data perusahaan

Untuk menampilkan data individu atau data perusahaan yang telah dimasukkan ke dalam sistem, *user* harus membuka aplikasi terlebih dahulu,

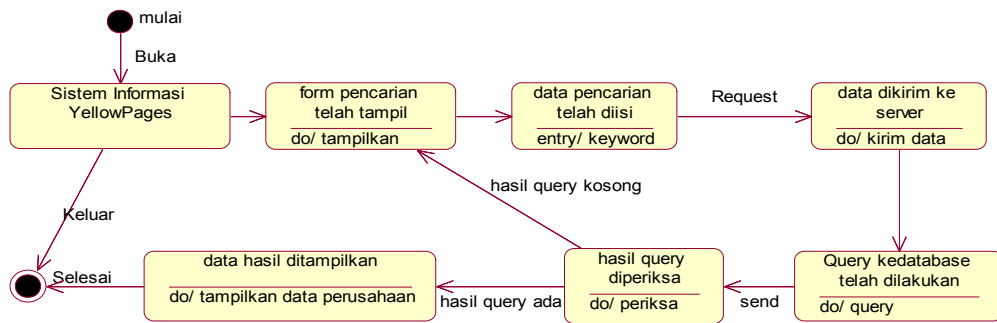
setelah menu utama tampil, *user* memilih menu *searching*, sehingga pada *handphone* menampilkan *form* *searching* individu atau perusahaan, kemudian *user* memilih *keyword* yang diinginkan, setelah itu *user* memilih perintah “OK” pada *handphone* untuk mengirimkan data tersebut ke *server*. *Server* mengambil masukan data yang dikirim, kemudian dilakukan proses *query* ke *database*, dan mengirimkan hasil *query* tersebut ke *handphone*. Pada *handphone* dilakukan proses *parsing*, apabila *query* yang dihasilkan kosong maka *handphone* menampilkan pesan kegagalan, apabila hasil *query* ada maka data ditampilkan.

4.5.1.3 Statechart Diagram

Diagram ini menunjukkan urutan *state* ke *state* yang lain dan aksi yang dilakukan pada sebuah *state*, yang digunakan untuk memodelkan perilaku dinamik dari sebuah objek. *Statechart diagram* yang dijelaskan pada analisa ini juga menjelaskan masing-masing *use case* yang ada pada *use case diagram*. Gambar 4.8 dibawah ini menjelaskan *statechart diagram* untuk menampilkan data individu dan Gambar 4.9 merupakan *statechart diagram* untuk menampilkan data perusahaan.



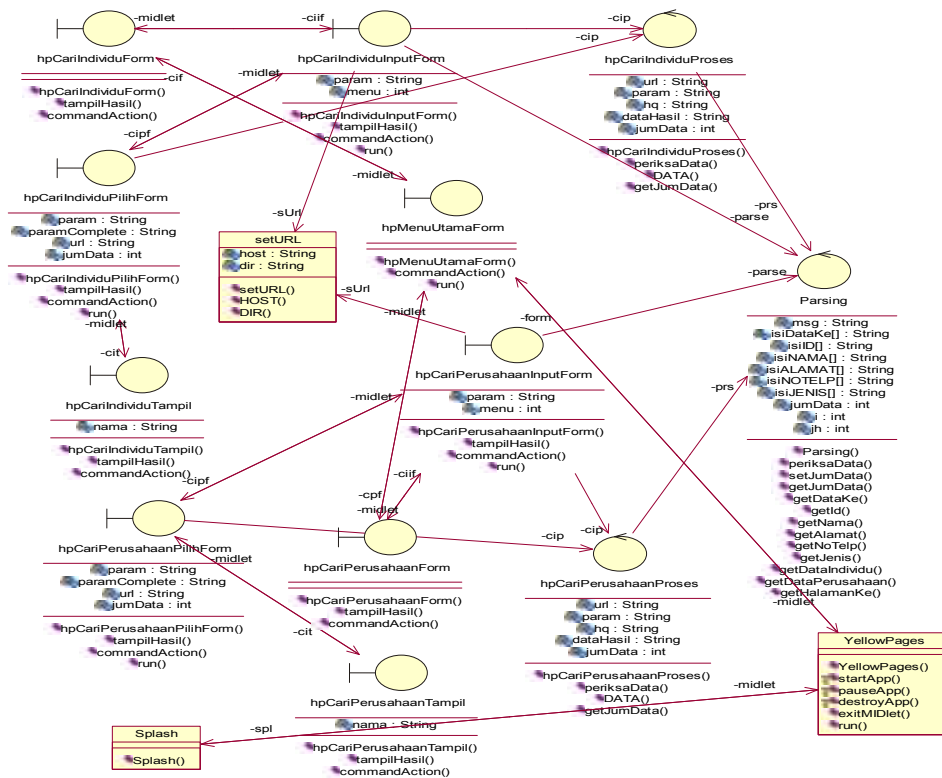
Gambar 4.8 *Statechart diagram* menampilkan data individu



Gambar 4.9 Statechart diagram menampilkan data perusahaan

4.5.1.4 Class Diagram

Class diagram yang dijelaskan pada analisa ini adalah class diagram pada J2ME dan class diagram pada server. Gambar 4.10 menjelaskan class diagram pada J2ME.



Gambar 4.10 Class diagram pada J2ME

Deskripsi gambar 4.10 tentang *class diagram* pada *J2ME* dapat dilihat pada tabel

4.3.

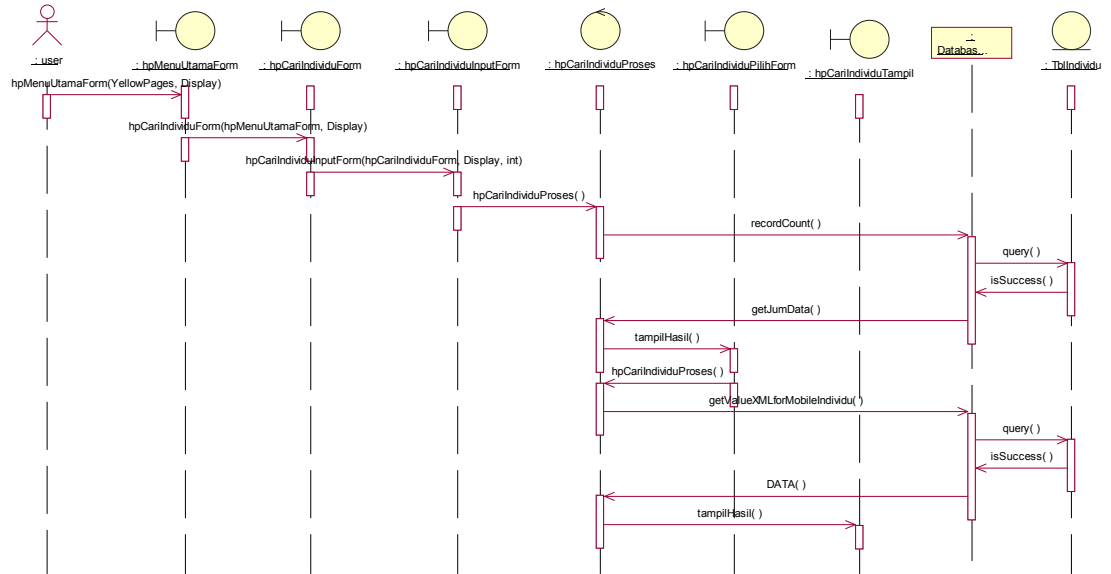
Tabel 4.3 Deskripsi *Class diagram* pada *J2ME*

No	Nama <i>Class</i>	Deskripsi
1.	YellowPages	<i>Class</i> untuk mengeksekusi tampilan awal sistem.
2.	Splash	<i>Class</i> untuk mengeksekusi gambar tampilan
3.	Parsing	<i>Class</i> untuk <i>parsing</i> data yang dihasilkan.
4.	setURL	<i>Class</i> untuk mendefinisikan alamat <i>server</i> .
5.	hpCariIndividuForm	<i>Class</i> sebagai antarmuka untuk mencari data individu.
6.	hpCariIndividuInputForm	<i>Class</i> sebagai antarmuka untuk menginput pencarian data individu.
7.	hpCariIndividuProses	<i>Class</i> untuk mengeksekusi data individu yang diinginkan.
8.	hpCariIndividuPilihForm	<i>Class</i> sebagai antarmuka untuk memilih keyword pencarian individu.
9.	hpMenuUtamaForm	<i>Class</i> sebagai antarmuka untuk memilih menu.
10.	hpCariIndividuTampil	<i>Class</i> untuk menampilkan data individu.
11.	hpCariPerusahaanForm	<i>Class</i> sebagai antarmuka untuk mencari data perusahaan.
12.	hpCariPerusahaanInputForm	<i>Class</i> sebagai antarmuka untuk menginput pencarian data perusahaan.
13.	hpCariPerusahaanProses	<i>Class</i> untuk mengeksekusi data perusahaan yang diinginkan.
14.	hpCariPerusahaanPilihForm	<i>Class</i> sebagai antarmuka untuk memilih keyword pencarian perusahaan.
15.	hpCariPerusahaanTampil	<i>Class</i> untuk menampilkan data perusahaan.

4.5.1.5 *Sequence Diagram*

Sequence Diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. *Diagram* ini secara khusus berhubungan dengan *use case*. *Sequence diagram* memperlihatkan tahap demi tahap untuk menghasilkan sesuatu didalam

use case. Gambar 4.11 dibawah ini menggambarkan *Sequence diagram* untuk menampilkan data individu.



Gambar 4.11 *Sequence diagram* menampilkan data individu

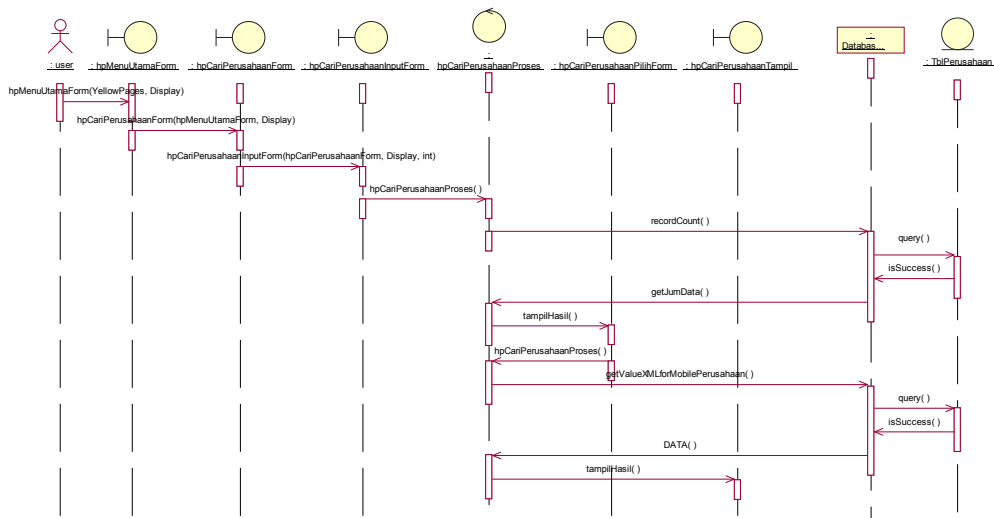
Deskripsi gambar 4.11 tentang *sequence diagram* menampilkan data individu pada *handphone* dapat dilihat pada tabel 4.4.

Tabel 4.4 Deskripsi *sequence diagram* menampilkan data individu

No.	Nama <i>Method</i>	Deskripsi
1.	Menu Utama()	<i>Constructor</i> untuk menampilkan antarmuka permintaan menu utama
2.	hpCariIndividuForm()	<i>Constructor</i> untuk menampilkan antarmuka untuk mencari data individu.
3.	hpCariIndividuInputForm ()	<i>Constructor</i> untuk menampilkan antarmuka menginput pencarian data individu.
4.	hpCariIndividuProses ()	<i>Constructor</i> untuk mengeksekusi data individu yang diinginkan.
5.	recordCount()	<i>Method</i> untuk permintaan data ke <i>database</i> .
6.	<i>query</i> ()	Melakukan perintah <i>query</i> ke <i>database</i>

7.	isSuccess()	Method untuk pemberitahuan jika data yang diinginkan ada.
8.	getJumData()	Method untuk pemberitahuan jumlah data yang didapat
9.	tampilHasil()	Method untuk menampilkan jumlah data
10.	getValueXMLforMobileIndividu()	Method untuk meminta semua data yang didapat
11.	DATA()	Method untuk mengirim semua data yang didapat

Gambar 4.12 dibawah ini menggambarkan *sequence diagram* untuk menampilkan data perusahaan.



Gambar 4.12 *Sequence Diagram* menampilkan data perusahaan

Deskripsi gambar 4.12 tentang *sequence diagram* menampilkan data individu pada *handphone* dapat dilihat pada tabel 4.5.

Tabel 4.5 Deskripsi *sequence diagram* menampilkan data Perusahaan

No.	Nama Method	Deskripsi
1.	Menu Utama()	Constructor untuk menampilkan antarmuka permintaan menu utama
2.	hpCariPerusahaanForm()	Constructor untuk menampilkan antarmuka untuk

		mencari data perusahaan.
3.	hpCariPerusahaanInputForm ()	<i>Constructor</i> untuk menampilkan antarmuka menginput pencarian data perusahaan.
4.	hpCariPerusahaanProses ()	<i>Constructor</i> untuk mengeksekusi data perusahaan yang diinginkan.
5.	recordCount()	<i>Method</i> untuk permintaan data ke <i>database</i> .
6.	<i>query</i> ()	Melakukan perintah <i>query</i> ke <i>database</i>
7.	isSuccess()	<i>Method</i> untuk pemberitahuan jika data yang diinginkan ada.
8.	getJumData()	<i>Method</i> untuk pemberitahuan jumlah data yang didapat
9.	tampilHasil()	<i>Method</i> untuk menampilkan jumlah data
10.	getValueXMLforMobilePerusahaan ()	<i>Method</i> untuk meminta semua data yang didapat
11.	DATA()	<i>Method</i> untuk mengirim semua data yang didapat

4.5.1.6 Deployment Diagram

Deployment Diagram ini menjelaskan hubungan antara *software* dan *hardware* yang ada pada sisi *handphone*. Gambar 4.13 dibawah ini menggambarkan hubungan antara *software* dan *hardware* yang digunakan pada sistem informasi yang dibangun.



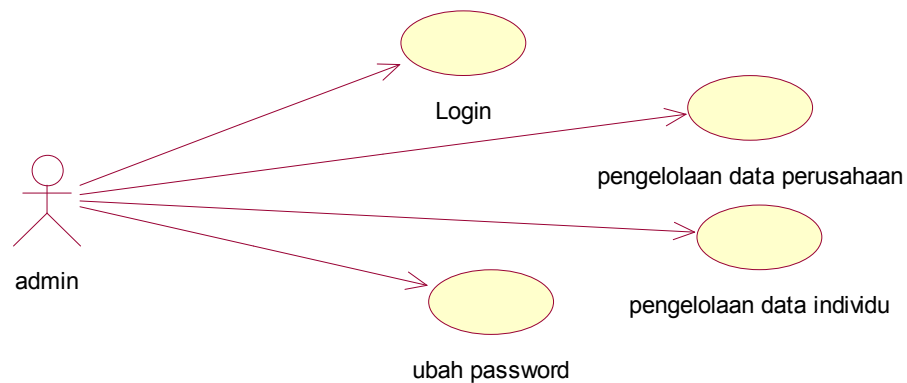
Gambar 4.13 *Deployment Diagram* sisi *handphone*

Gambar 4.13 Menjelaskan cara kerja bagaimana *handphone* dapat mengakses sistem. *Handphone* menggunakan jaringan GPRS terlebih dahulu untuk terhubung ke jaringan *internet* dan dapat mengakses sistem informasi.

4.5.2 UML (*Unified Modelling Language*) Pada Sisi Server

4.5.2.1 Use Case Diagram

Use case diagram menggambarkan hubungan dari setiap objek yang ada pada sebuah sistem.. Gambar 4.14 dibawah ini menggambarkan aliran *use case diagram* untuk pengaksesan melalui *komputer*



Gambar 4.14 Aliran *use case diagram* (pengaksesan melalui *Komputer*)

Deskripsi gambar 4.5 tentang *use case diagram* (pengaksesan melalui *Komputer*) dapat dilihat pada tabel 4.6.

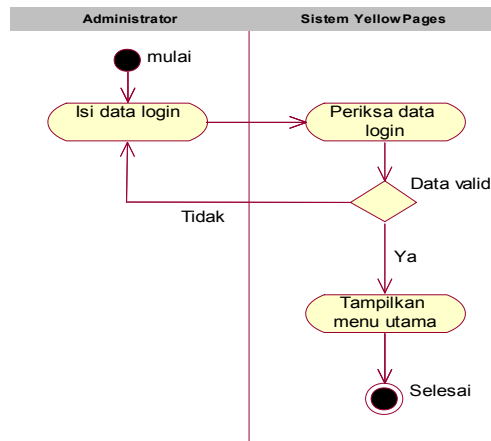
Tabel 4.6 *Use case diagram* (pengaksesan melalui *Komputer*)

No	Nama <i>Use Case</i>	Deskripsi
1.	<i>Login</i>	Proses untuk <i>login</i> ke Sistem..
2.	Pengelolaan Data Perusahaan	Proses pengelolaan data perusahaan.
3.	Pengelolaan Data Individu	Proses pengelolaan data individu.
4.	Ubah <i>Password</i>	Proses untuk mengubah <i>password</i>

4.5.2.2 Activity Diagram

4.5.2.2.1 Activity Diagram Login

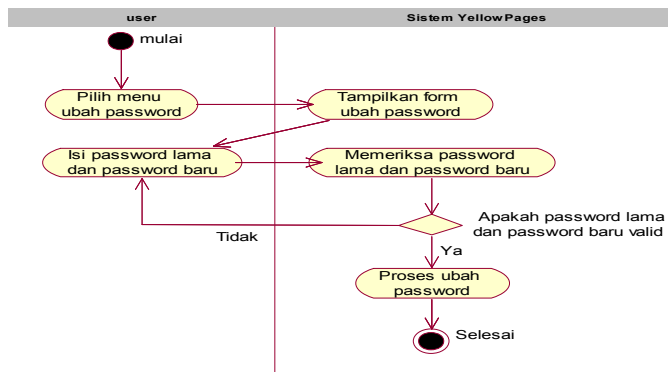
Activity diagram login menjelaskan alur kerja yang harus dilalui untuk melakukan *login* melalui *komputer*.



Gambar 4.15 Activity Diagram Proses Login

4.5.2.2.2 Activity Diagram Ubah Password

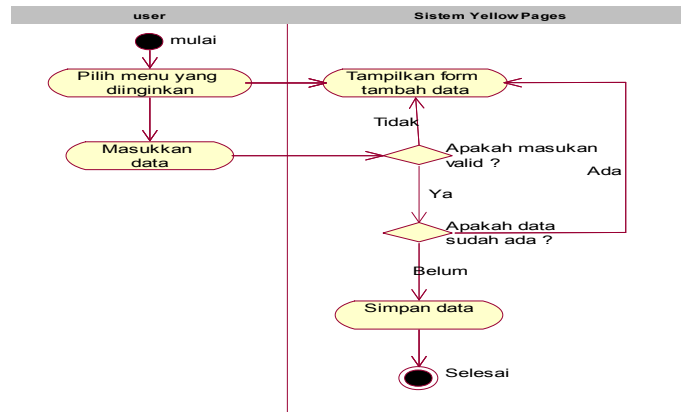
Activity diagram mengubah *password* menggambarkan tahap-tahap yang harus dilalui untuk mengubah *password*.



Gambar 4.16 Activity Diagram Proses Ubah Password

4.5.2.2.3 Activity Diagram Tambah Data

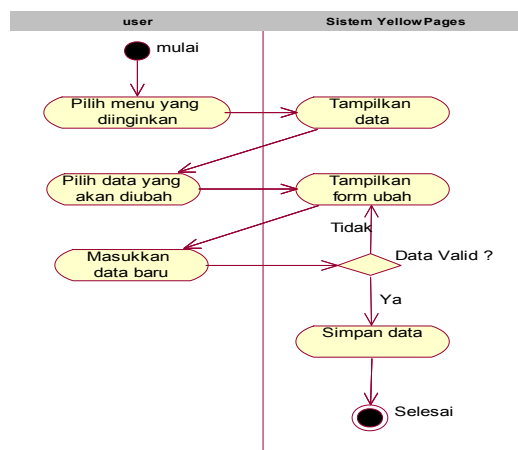
Activity diagram tambah data menggambarkan tahap-tahap yang harus dilalui untuk menambah data. *Activity diagram* tambah data mencakup penambahan data pada data individu dan data perusahaan.



Gambar 4.17 *Activity Diagram* Penambahan Data

4.5.2.2.4 Activity Diagram Ubah Data

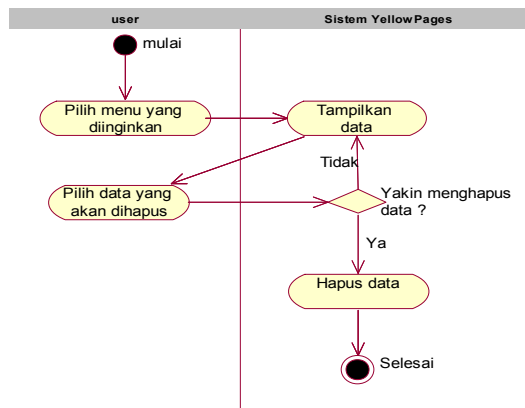
Activity diagram ubah data menggambarkan tahap-tahap yang harus dilalui untuk mengubah data. *Activity diagram* ubah data mencakup pengubahan data pada data individu dan data perusahaan.



Gambar 4.18 *Activity Diagram* Ubah Data

4.5.2.2.5 Activity Diagram Hapus Data

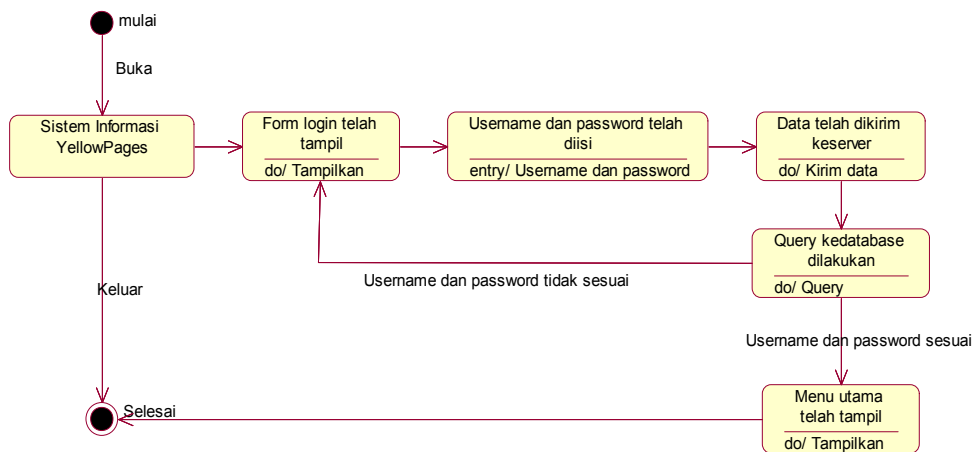
Activity diagram hapus data menggambarkan tahap-tahap yang harus dilalui untuk menghapus data. *Activity diagram* ubah data mencakup penghapusan data pada data individu dan data perusahaan.



Gambar 4.19 Activity Diagram Hapus Data

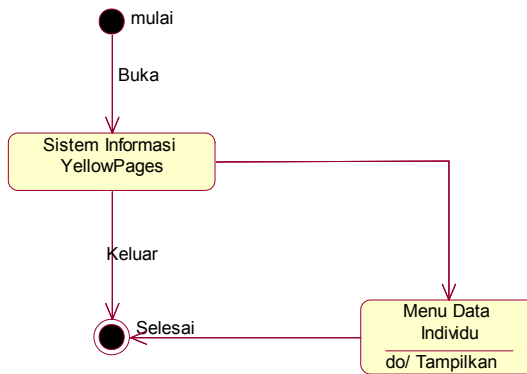
4.5.2.3 Statechart Diagram

4.5.2.3.1 Statechart Diagram login



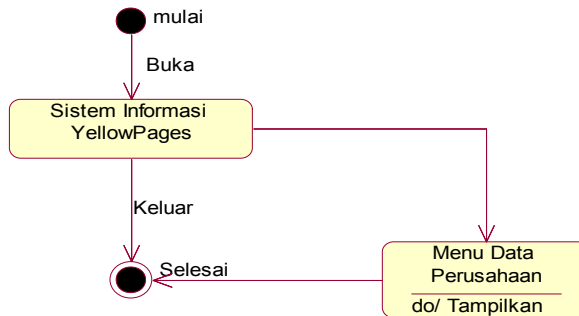
Gambar 4.20 Statechart Diagram Proses Login

4.5.2.3.2 Statechart Diagram Menampilkan Data Individu



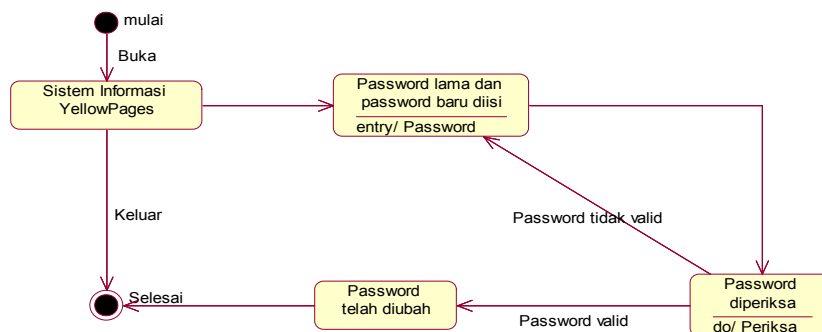
Gambar 4.21 Statechart Diagram Menampilkan data Individu

4.5.2.3.3 Statechart Diagram Menampilkan Data Perusahaan



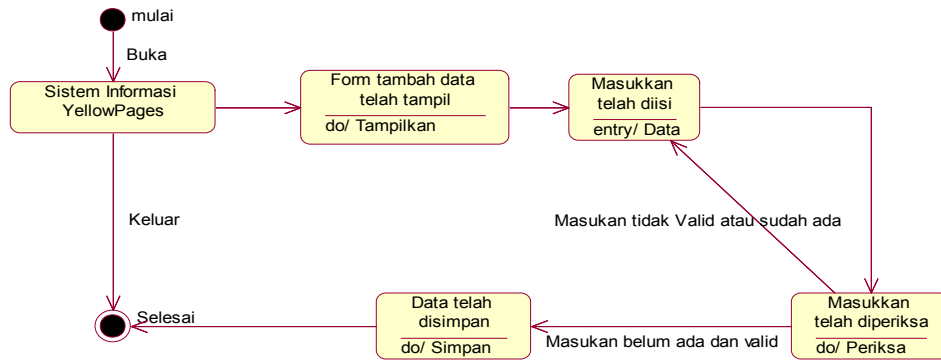
Gambar 4.22 Statechart Diagram Menampilkan data Perusahaan

4.5.2.3.4 Statechart Diagram Mengubah Password



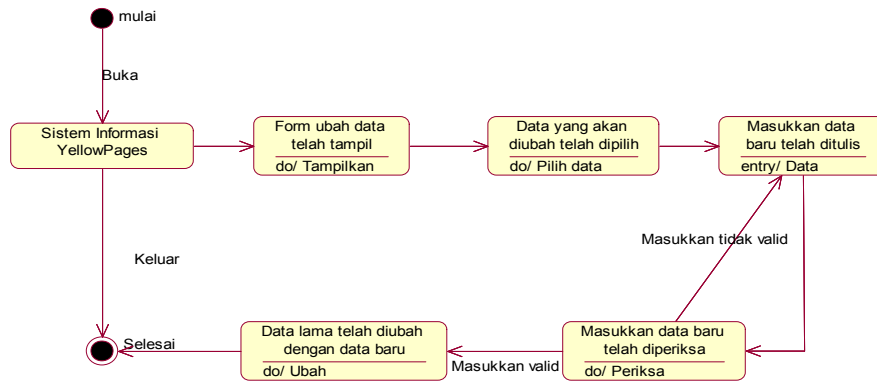
Gambar 4.23 Statechart Diagram Mengubah Password

4.5.2.3.5 Statechart Diagram Menambah Data



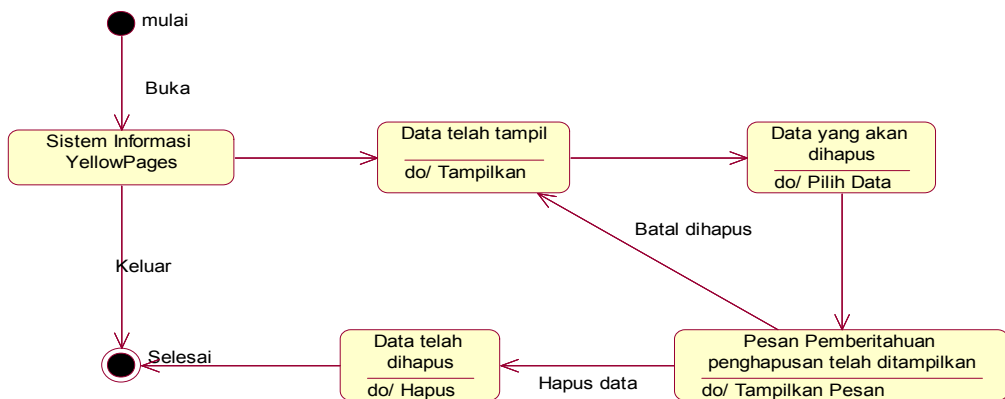
Gambar 4.24 Statechart Diagram Menambah Data

4.5.2.3.6 Statechart Diagram Mengubah Data



Gambar 4.25 Statechart Diagram Mengubah Data

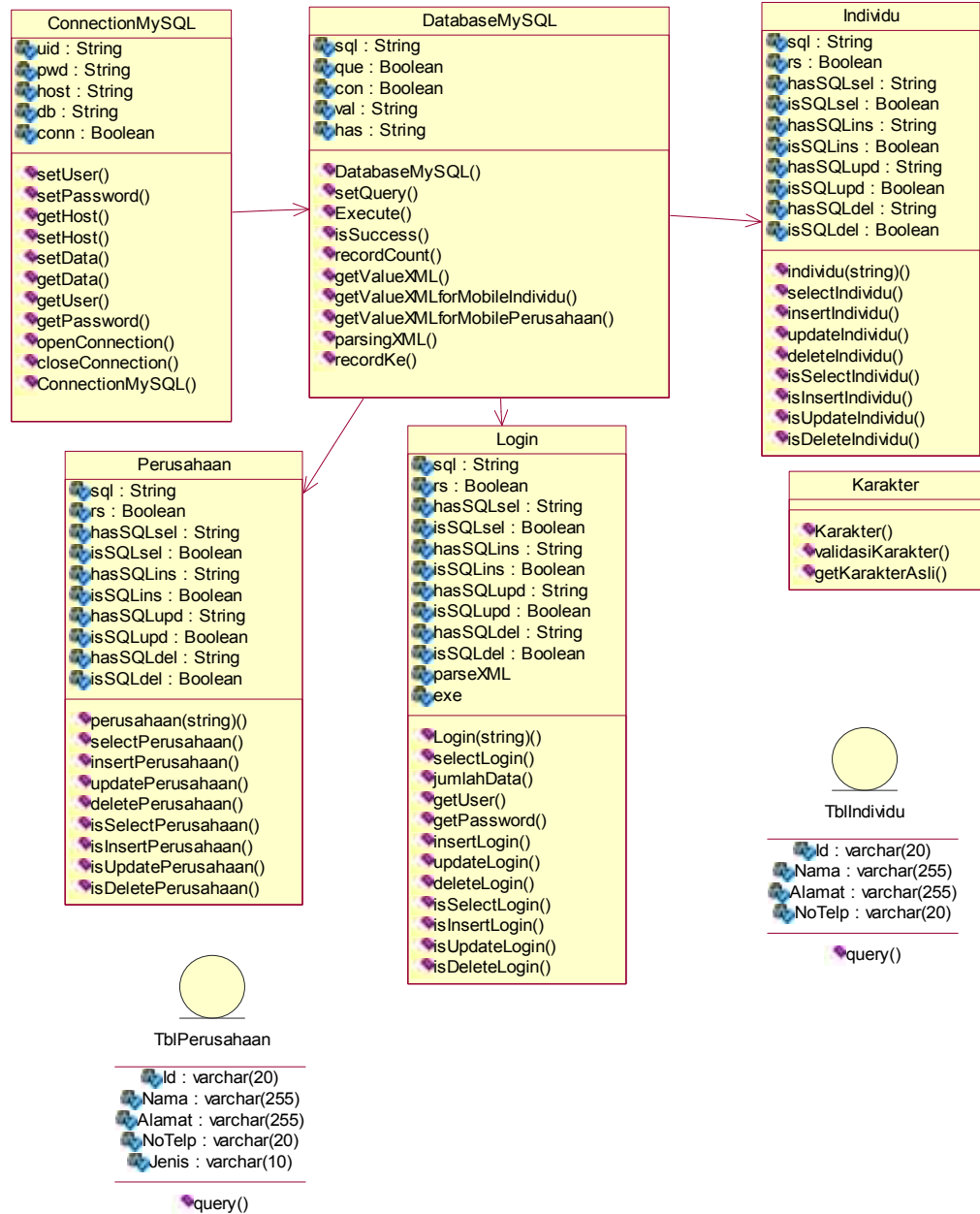
4.5.2.3.7 Statechart Diagram Menghapus Data



Gambar 4.26 Statechart Diagram Menghapus Data

4.5.2.4 Class Diagram Pada Sisi Server

Class diagram yang dijelaskan pada gambar 4.27 adalah class diagram pada server.



Gambar 4.27 Class diagram pada Server

Deskripsi gambar 4.27 tentang *class diagram* pada *Server* dapat dilihat pada tabel

4.7.

Tabel 4.7 Deskripsi *Class diagram* pada *Server*

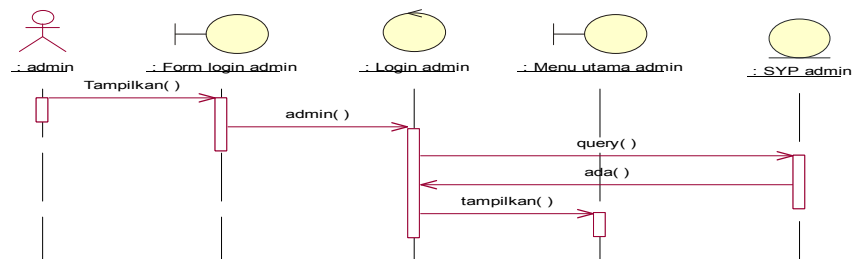
No	Nama <i>Class</i>	Deskripsi
1.	ConnectionMySQL	<i>Class</i> untuk melakukan koneksi <i>database</i>
2.	<i>DatabaseMySQL</i>	<i>Class</i> untuk melakukan query ke <i>database</i>
3.	Individu	<i>Class</i> untuk query <i>select, insert, update, dan delete</i> pada data individu.
4.	Perusahaan	<i>Class</i> untuk query <i>select, insert, update, dan delete</i> pada data perusahaan.
5.	Login	<i>Class</i> untuk query <i>select, insert, update, dan delete</i> pada proses login.
6.	Karakter	<i>Class</i> untuk pengecekan karakter yang di inputkan
7.	TblIndividu	<i>Class</i> untuk <i>database</i> individu
8.	TblPerusahaan	<i>Class</i> untuk <i>database</i> perusahaan

4.5.2.5 Sequence Diagram

Sequence Diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. *Diagram* ini secara khusus berhubungan dengan *use case*. *Sequence diagram* memperlihatkan tahap demi tahap untuk menghasilkan sesuatu didalam *use case*.

4.5.2.5.1 Sequence Diagram Login Administrator

Gambar 4.28 dibawah ini menggambarkan *Sequence diagram* untuk *Login administrator*.

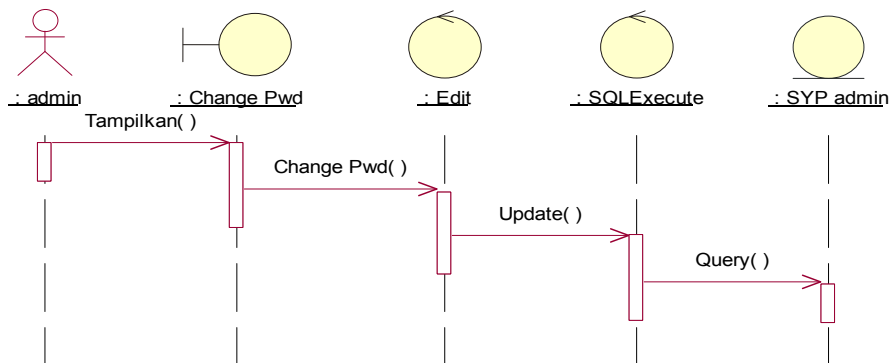


Gambar 4.28 *Sequence diagram login administrator*

Tabel 4.8 Deskripsi *sequence diagram login administrator*

No.	Nama Objek	Deskripsi
1.	form_login_administrator	Antarmuka untuk <i>login administrator</i>
2.	LoginAdministrator	<i>Class</i> untuk <i>query login administrator</i>
3.	menuUtamaAdministrator	Antarmuka menu utama untuk <i>administrator</i>

4.5.2.5.2 Sequence Diagram Ubah Password

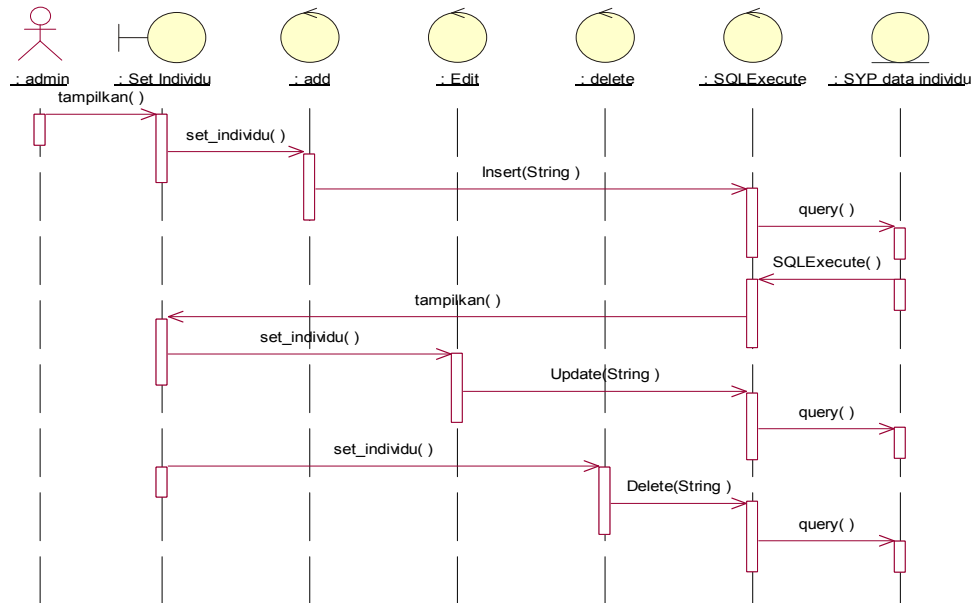


Gambar 4.29 *Sequence diagram* ubah password

Tabel 4.9 Deskripsi *sequence diagram* ubah password administrator

No.	Nama Objek	Deskripsi
1.	change_psw	Antarmuka untuk mengubah password <i>administrator</i>
2.	syp_administrator	<i>Tabel</i> syp_administrator pada <i>database</i>

4.5.2.5.3 Sequence Diagram Pengelolaan Data Individu

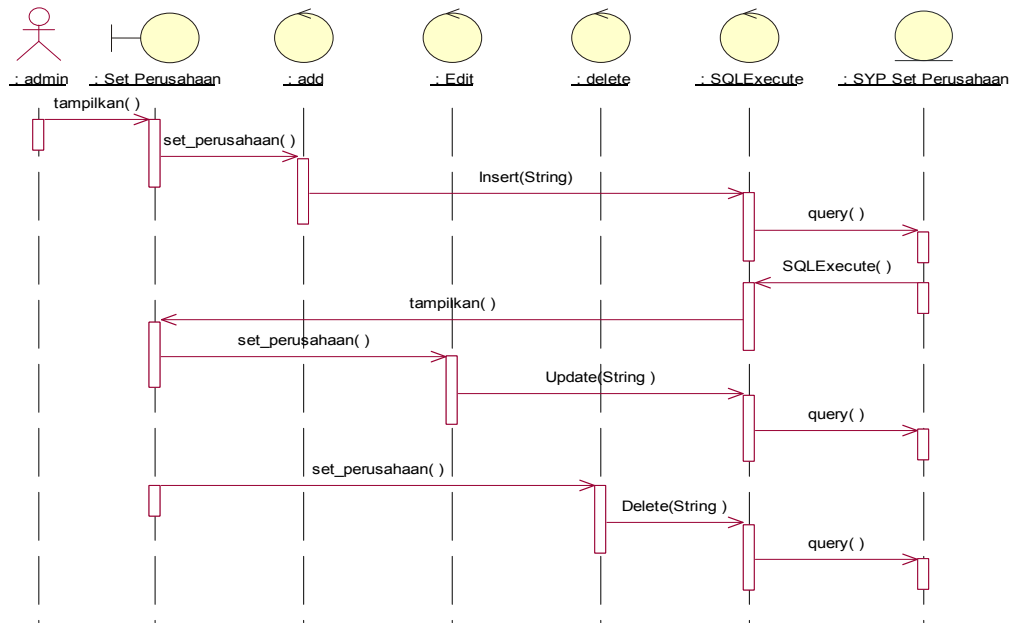


Gambar 4.30 Sequence diagram pengelolaan data individu

Tabel 4.10 Deskripsi sequence diagram pengelolaan data individu

No.	Nama Objek	Deskripsi
1.	<i>Administrator</i>	<i>Administrator yang mengakses sistem</i>
2.	<i>set_individu</i>	<i>Form untuk mengelola data individu</i>
3.	<i>Add</i>	Perintah untuk menambah data individu
4.	<i>Edit</i>	Perintah untuk mengubah data individu
5.	<i>Delete</i>	Perintah untuk menghapus data individu

4.5.2.5.4 Sequence Diagram Pengelolaan Data Perusahaan



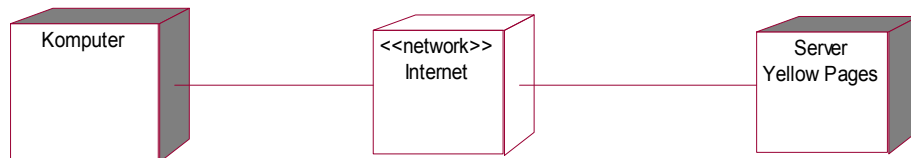
Gambar 4.31 Sequence diagram pengelolaan data perusahaan

Tabel 4.11 Deskripsi sequence diagram pengelolaan data perusahaan

No.	Nama Objek	Deskripsi
1.	<i>Administrator</i>	<i>Administrator</i> yang mengakses sistem
2.	<i>set_perusahaan</i>	<i>Form</i> untuk mengelola data perusahaan
3.	<i>Add</i>	Perintah untuk menambah data perusahaan
4.	<i>Edit</i>	Perintah untuk mengubah data perusahaan
5.	<i>Delete</i>	Perintah untuk menghapus data perusahaan

4.5.2.6 Deployment Diagram

Deployment Diagram ini menjelaskan hubungan antara *software* dan *hardware* yang ada pada sisi *server*. Gambar 4.32 dibawah ini menggambarkan hubungan antara *software* dan *hardware* yang digunakan pada sistem informasi yang dibangun.



Gambar 4.32 *Deployment Diagram* sisi *Server*

Gambar 4.32 Menjelaskan cara kerja bagaimana komputer dapat mengakses sistem. *Komputer* menggunakan jaringan *internet* dan dapat mengakses sistem informasi secara langsung.

4.6 Perancangan Antarmuka Pengguna Sistem

Perancangan antarmuka untuk pengguna sistem pada *handphone* antara lain, splash, menu utama, menu pencarian, pilihan keyword, pencarian individu, pencarian perusahaan dan hasil pencarian.

4.6.1 Perancangan Menu Utama pada *Handphone*

Gambar 4.33 dibawah menjelaskan perancangan menu utama pada *handphone*.



Gambar 4.33 Perancangan menu utama pada *handphone*

Deskripsi gambar 4.33 tentang perancangan antarmuka pada *handphone* dapat dilihat pada tabel 4.12.

Tabel 4.12 Deskripsi perancangan antarmuka nilai pada *handphone*

No	Nama <i>Item</i>	Deskripsi
1.	<i>Yellow pages Mobile</i> Kota Pekanbaru	Nama sistem informasi
2.	cgIndividu	<i>ChoiceGroup</i> untuk pilihan individu.
3.	cgPerusahaan	<i>ChoiceGroup</i> untuk pilihan perusahaan.
4.	cmdKeluar	<i>Command</i> untuk kembali ke menu sebelumnya
5.	cmdOK	<i>Command</i> untuk mengeksekusi pilihan

4.6.2 Perancangan Antarmuka Keluaran Data pada *Handphone*

Gambar 4.34 dibawah ini menjelaskan keluaran data pada *handphone*.



Gambar 4.34 Perancangan keluaran data pada *handphone*

Deskripsi gambar 4.34 tentang perancangan keluaran data pada *handphone* dapat dilihat pada tabel 4.6.

Tabel 4.13 Deskripsi Perancangan Antarmuka Keluaran data pada *handphone*

No	Nama <i>Item</i>	Deskripsi
1.	<i>Yellow pages Mobile Kota Pekanbaru</i>	Nama sistem informasi
2.	Hasil	<i>TextBox</i> yang menampilkan keluaran.
3.	cmdKembali	<i>Command</i> untuk kembali ke <i>form</i> sebelumnya.

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi merupakan kelanjutan dari tahap perancangan sistem yang telah didesain. Implementasi merupakan tahap pembangunan sistem menggunakan perangkat keras dan perangkat lunak yang telah ditetapkan. Implementasi Sistem *Yellow pages* yang dibangun pada sisi *handphone* menggunakan perangkat lunak *JCreator* sebagai *text editor* dan *J2ME Wireless Toolkit 2.2* sebagai *compiler*. Implementasi pada *server* menggunakan *database MySQL* dan bahasa pemrograman *JSP* serta *JavaScript*. Penulisan kode program pada *server* menggunakan *editor Macromedia Dreamweaver MX*.

5.1.1 Lingkungan Implementasi

Lingkungan implementasi sistem ada 2 (dua) yaitu: lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1.1 Perangkat Keras

Web browser Mozilla firefox dijalankan pada sistem operasi *Windows XP Professional* menggunakan komputer *Intel Dual-core 1,46 GHz* dan *1 GB RAM* untuk mengelola *database-nya*. *Script JSP* dan *DBMS* di-*upload* ke salah satu penyedia *hosting* untuk *JSP* yaitu <http://www.orgfree.com>. Sistem yang dibangun untuk *handphone* di-*install*-kan pada *handphone* *Nokia E63*, *Nokia 6300*, dan *Sony Ericson W350i*.

5.1.1.2 Perangkat lunak

Perangkat lunak yang digunakan dalam implementasi ini menggunakan:

1. *JCreator* sebagai *text editor* J2ME dan *J2ME Wireless Toolkit 2.2* sebagai *compiler* yang digunakan dalam membangun Sistem Informasi *Yellow Pages* pada sisi *handphone*.
2. *Java Software Development Kit 1.6* (JSDK 1.6) sebagai *compiler* utama bahasa pemrograman *Java*.
3. Menggunakan *database* MySQL yang terdapat pada aplikasi *XAMPP*.
4. *Macromedia Dreamweaver MX* sebagai *text editor* JSP dan *Apache Tomcat 6.0* sebagai *Web Server*.
5. *Web Browser* yang digunakan menjalankan sistem informasi *Yellow Pages* yang dioperasikan pada *komputer*.
6. *Microsoft Windows XP* sebagai sistem operasi yang digunakan dalam mengimplementasikan perangkat lunak.

5.1.2 Batasan Implementasi

Batasan implementasi dari tugas akhir ini adalah:

1. Menggunakan bahasa pemrograman Java dengan teknologi J2ME untuk membangun sistem informasi *Yellow pages* pada sisi *handphone*.
2. Menggunakan JSP untuk membangun sistem informasi *Yellow Pages* pada sisi *server*.

5.1.3 Hasil Implementasi

Tahap implementasi menghasilkan modul-modul yang digunakan dalam sistem dan antarmuka untuk pengguna.

5.1.3.1 Hasil Implementasi Modul

Sistem yang dibangun terdiri dari dua modul yaitu modul pada *handphone* dan modul pada *server*.

5.1.3.1.1 Modul pada *Handphone*

Modul-modul yang digunakan pada *handphone* untuk semua permintaan informasi ke *server* adalah *setURL()*, *Parsing()*, *hpCariPerusahaanProses()* dan *hpCariIndividuProses()*.

Kode program pada sisi *handphone* untuk *request* data ke *server* dapat dilihat pada lampiran A.

5.1.3.1.2 Modul pada *Server*

Modul-modul yang digunakan untuk melakukan proses *query* pada *server* adalah *class.php*

Kode program pada sisi *server* untuk melakukan koneksi ke *database* dapat dilihat pada lampiran A.

5.1.3.2 Hasil Implementasi Layar

Menu utama Sistem informasi *Yellow Pages* pada sisi *handphone* terdiri dari menu *ChoiceGroup* individu, *ChoiceGroup* perusahaan. Perintah yang digunakan untuk melakukan eksekusi terdiri dari *Command Ok* dan *Command Keluar*. antarmuka menu utama dapat dilihat pada gambar 5.1.



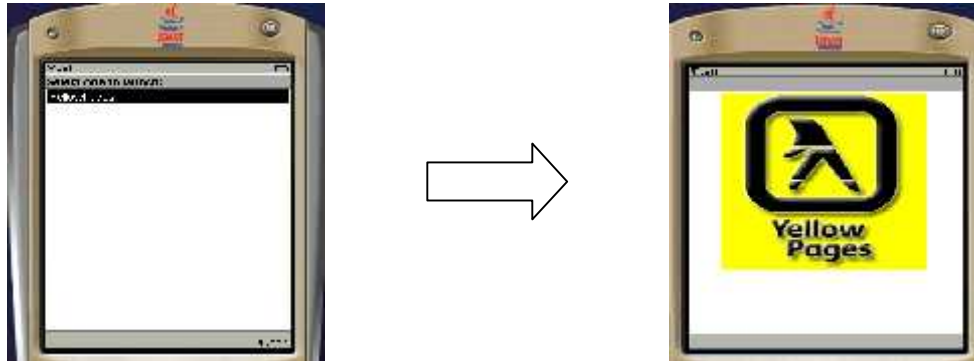
Gambar 5.1 Antarmuka menu utama sistem pada sisi *handphone*.

Tabel 5.1 Deskripsi antarmuka menu utama sistem pada sisi *handphone*.

No.	Objek	Nama	Deskripsi
1.	Individu	cgPilihan(0)	Menampilkan <i>form</i> pencarian data individu.
2.	Perusahaan	cgPilihan(1)	Menampilkan <i>form</i> pencarian data perusahaan.
3.	Keluar	cmdKeluar	Perintah untuk keluar.
4.	Ok	cmdOk	Perintah untuk mengeksekusi menu yang dipilih.

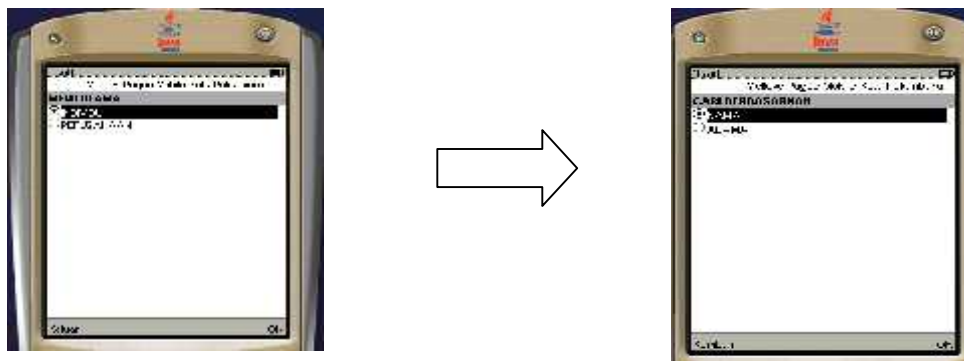
5.1.3.2.1 Implementasi Pada J2ME Wireless Toolkit

5.1.3.2.1.1 Menampilkan Splash



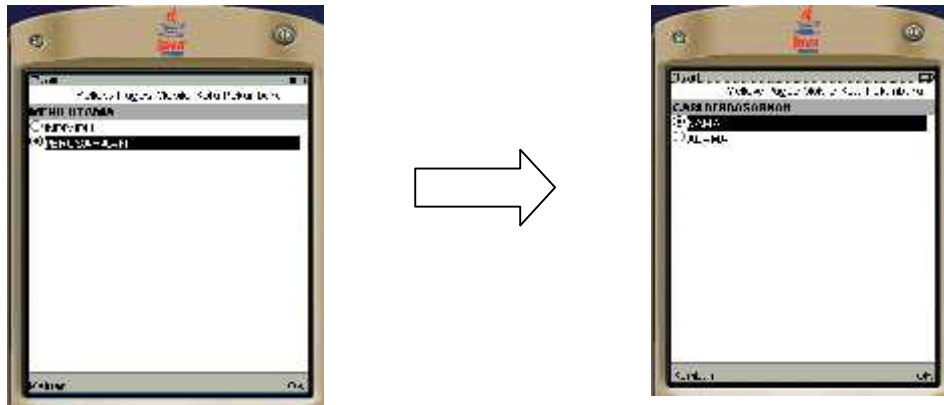
Gambar 5.2 Menampilkan *Splash* (*J2ME Wireless Toolkit*)

5.1.3.2.1.2 Menampilkan Menu Pencarian Data Individu



Gambar 5.3 Menampilkan Menu Pencarian Data Individu (*J2ME Wireless Toolkit*)

5.1.3.2.1.3 Menampilkan Menu Pencarian Data Perusahaan

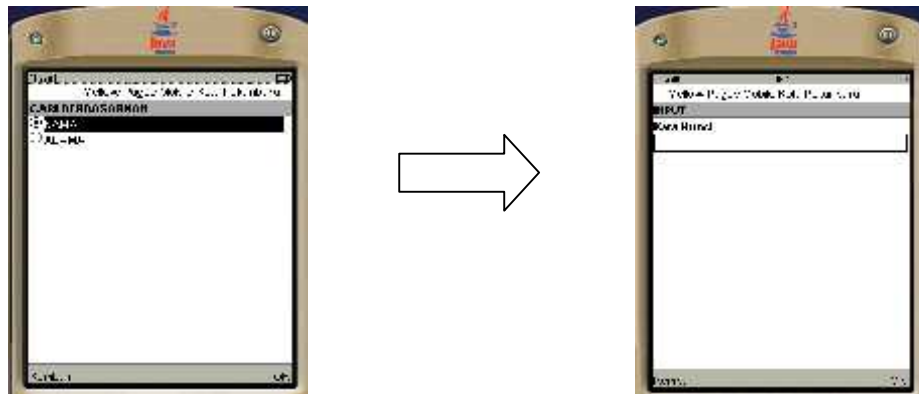


Gambar 5.4 Menampilkan Menu Pencarian Data Perusahaan

(J2ME Wireless Toolkit)

5.1.3.2.1.4 Menampilkan *Form Input Keyword* Berdasarkan Nama Pada Data

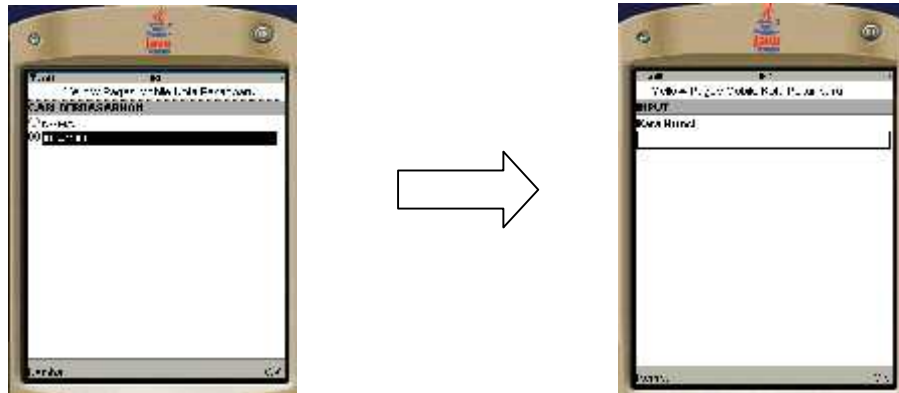
Individu



Gambar 5.5 Menampilkan *Form Input Keyword* Berdasarkan Nama Pada Data

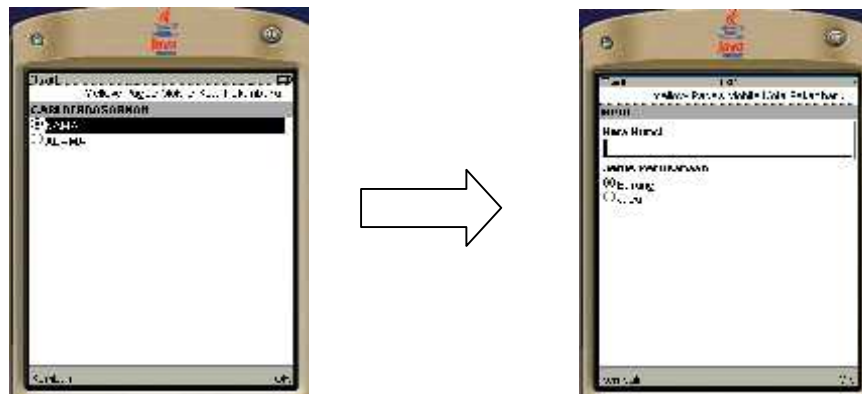
Individu (J2ME Wireless Toolkit)

5.1.3.2.1.5 Menampilkan *Form Input Keyword* Berdasarkan Alamat Pada Data Individu



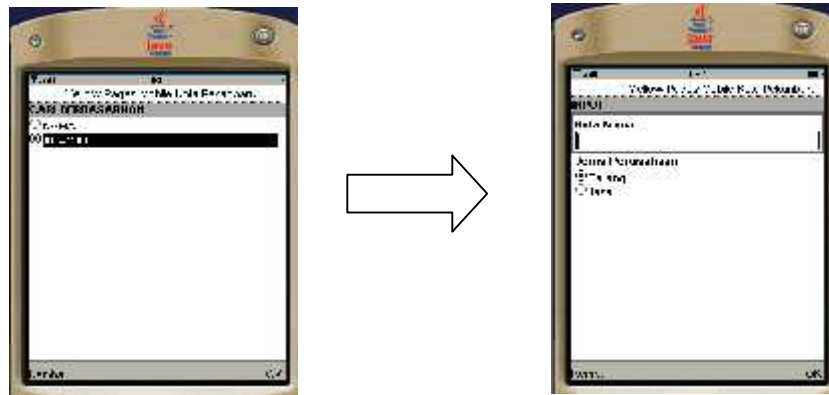
Gambar 5.6 Menampilkan *Form Input Keyword* Berdasarkan Alamat Pada Data Individu (J2ME Wireless Toolkit)

5.1.3.2.1.6 Menampilkan *Form Input Keyword* Berdasarkan Nama Pada Data Perusahaan



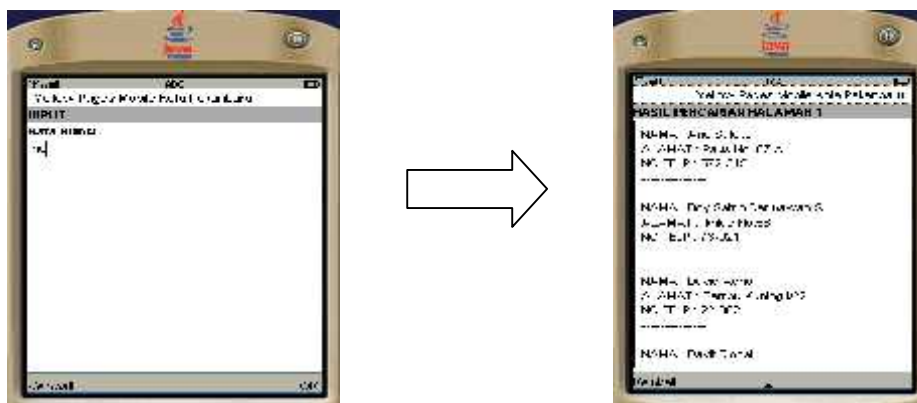
Gambar 5.7 Menampilkan *Form Input Keyword* Berdasarkan Nama Pada Data Perusahaan (J2ME Wireless Toolkit)

5.1.3.2.1.7 Menampilkan *Form Input Keyword* Berdasarkan Alamat Pada Data Perusahaan



Gambar 5.8 Menampilkan *Form Input Keyword* Berdasarkan Alamat Pada Data Perusahaan (*J2ME Wireless Toolkit*)

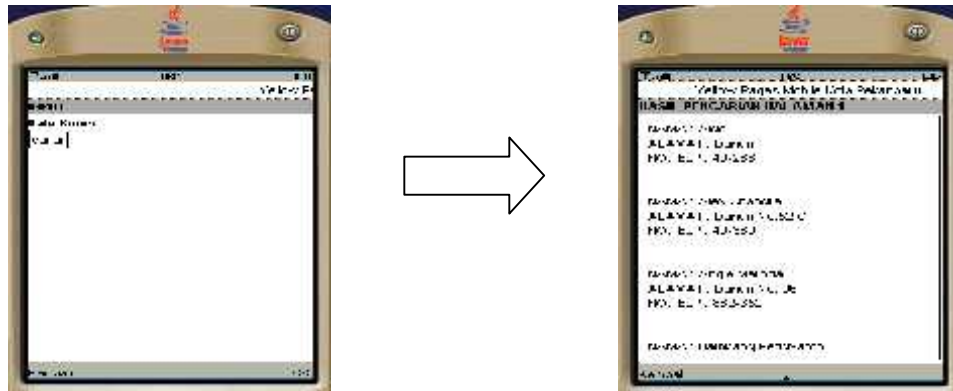
5.1.3.2.1.8 Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data Individu



Gambar 5.9 Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data Individu (*J2ME Wireless Toolkit*)

5.1.3.2.1.9 Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data

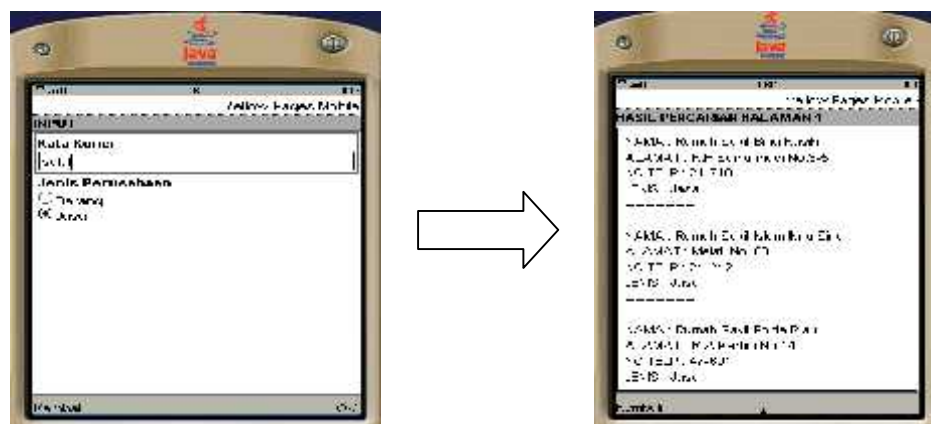
Individu



Gambar 5.10 Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data Individu (*J2ME Wireless Toolkit*)

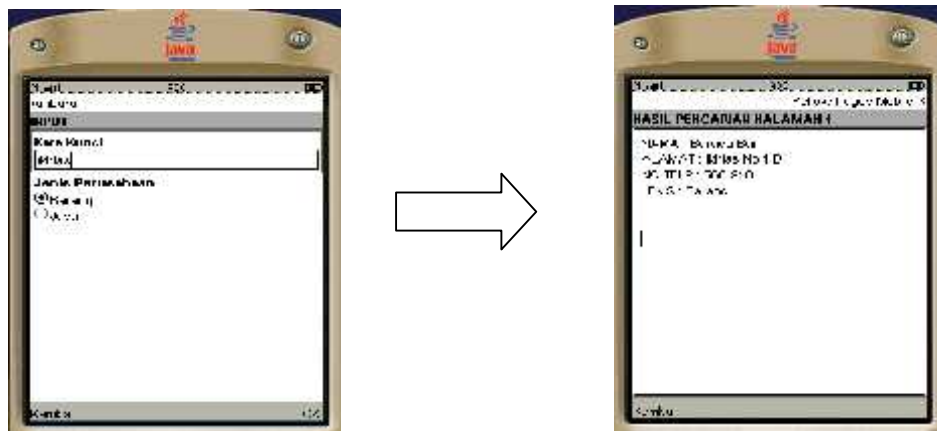
5.1.3.2.1.10 Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data

Perusahaan



Gambar 5.11 Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data Perusahaan (*J2ME Wireless Toolkit*)

5.1.3.2.1.11 Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data Perusahaan



Gambar 5.12 Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data Perusahaan (*J2ME Wireless Toolkit*)

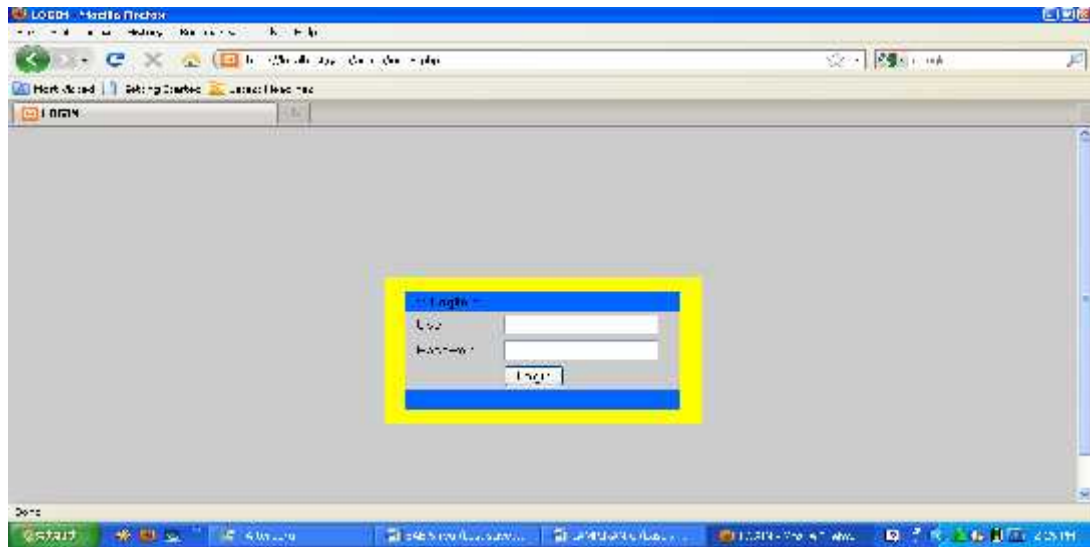
5.1.3.2.2 Implementasi Pada *Handphone*

Setelah sistem tersebut di-*install*-kan pada *handphone* Nokia E63, Nokia 6300 dan sony Ericson W350i maka dilakukan pengaksesan sistem informasi *Yellow pages*, sistem tersebut dapat menampilkan informasi sesuai dengan *database* yang ada pada *server*.

Sistem yang dibangun memiliki kelebihan, yaitu *user* dapat mengakses sistem informasi *Yellow pages* kapanpun dan dimanapun karena menggunakan tambahan media *handphone* untuk mengaksesnya, dengan syarat *handphone* yang digunakan harus mendukung fasilitas *Java* dan GPRS, serta tersedia sinyal GPRS. Sistem yang dibangun juga memiliki kekurangan yaitu *user* tidak dapat menambah data melalui *handphone*.

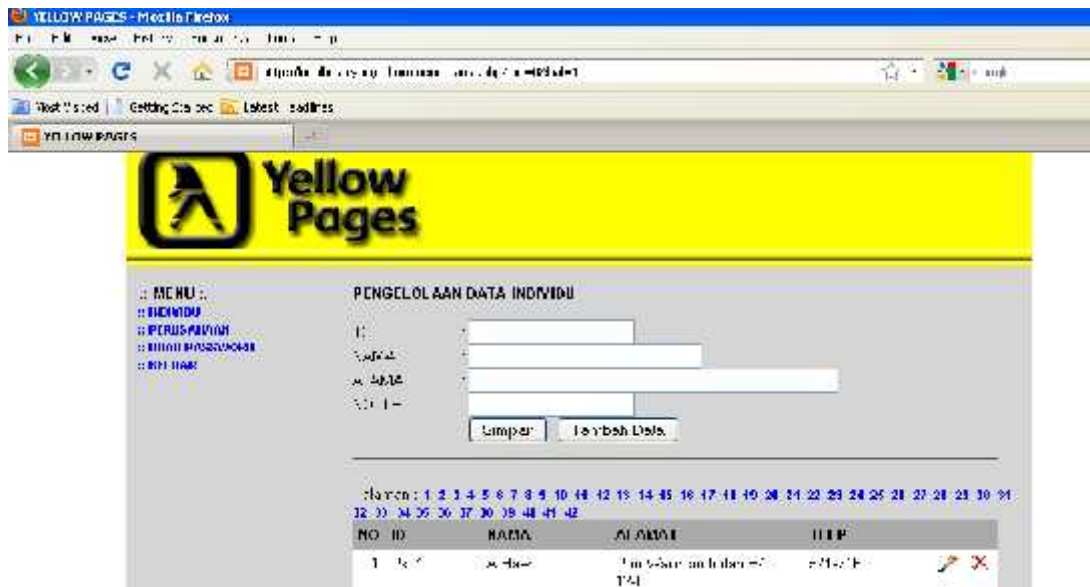
5.1.3.2.3 Implementasi Antarmuka *Administrator*

5.1.3.2.3.1 Antarmuka *Login Administrator*



Gambar 5.13 Antarmuka *Login Administrator*

5.1.3.2.3.2 Antarmuka Pengelolaan Data Individu



Gambar 5.14 Antarmuka Pengelolaan Data Individu

5.1.3.2.3.3 Antarmuka Pengelolaan Data Perusahaan



Gambar 5.15 Antarmuka Pengelolaan Data Perusahaan

5.1.3.2.3.4 Antarmuka Ubah Password Administrator



Gambar 5.16 Antarmuka Ubah Password Administrator

5.2 Pengujian Sistem

Pengujian dilakukan untuk mengetahui apakah sistem yang dibangun sesuai dengan yang diharapkan.

5.2.1 Perangkat Keras Pengujian

Perangkat keras yang digunakan untuk melakukan pengujian adalah Nokia E63, Nokia 6300, dan Sony Ericson W350i.

5.2.2 Identifikasi dan Rencana Pengujian

5.2.2.1 Pengujian Dengan *Handphone*

Tahap pengujian dilakukan untuk mengetahui hasil dari proses sistem yaitu memperlihatkan informasi *yellow pages* yang diakses melalui *handphone*. Hasil pengujian dapat dilihat setelah melakukan koneksi ke *server* dan dapat menampilkan menu utama beserta sub menu dan hasilnya. Kemudian pembuktian dilakukan untuk membuktikan bahwa informasi yang ditampilkan sesuai dengan isi *database*.

Alamat *website* yang digunakan sebagai *server* sistem informasi *Yellow Pages* untuk *handphone* adalah <http://yellowpages.orgfree.com/mobile/>. Tabel 5.2 di bawah ini menjelaskan pengujian-pengujian yang dilakukan menggunakan *handphone* Nokia E63, Nokia 6300 dan Sony Ericson W350i.

Tabel 5.2 Pengujian dengan beberapa *handphone*

Pukul/Tanggal	<i>Hanphone</i>	<i>Provider</i>	Pengujian	Hasil
17.15/ 17 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
18.50/ 17 Oktober 2010	Nokia 6300	Axis	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Gagal
			Pencarian data perusahaan dengan <i>keyword</i> nama	Gagal
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
09.10/ 18 Oktober 2010	Nokia 6300	Axis	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
17.00/ 18 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>

			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
20.00/ 18 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Gagal
00.30/ 19 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
08.00/ 19 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
08.20/ 19 Oktober 2010	Nokia 6300	Axis	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
10.00/ 19 Oktober 2010	Sony Ericson W350i	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>

			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
21.00/ 19 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Gagal
22.30/ 19 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Gagal
07.30/ 20 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
08.00/ 20 Oktober 2010	Nokia 6300	Axis	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
08.20/ 20 Oktober 2010	Sony Ericson W350i	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan <i>keyword</i> alamat	Sesuai <i>database</i>
08.50/ 21 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan <i>keyword</i> nama	Sesuai <i>database</i>
			Pencarian data individu dengan <i>keyword</i> alamat	Sesuai <i>database</i>

			Pencarian data perusahaan dengan keyword nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan keyword alamat	Sesuai <i>database</i>
11.30/ 21 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Gagal
14.00/ 21 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Gagal
18.00/ 21 Oktober 2010	Nokia E63	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan keyword nama	Sesuai <i>database</i>
			Pencarian data individu dengan keyword alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan keyword nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan keyword alamat	Sesuai <i>database</i>
18.30/ 21 Oktober 2010	Sony Ericson W350i	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			Pencarian data individu dengan keyword nama	Sesuai <i>database</i>
			Pencarian data individu dengan keyword alamat	Sesuai <i>database</i>
			Pencarian data perusahaan dengan keyword nama	Sesuai <i>database</i>
			Pencarian data perusahaan dengan keyword alamat	Sesuai <i>database</i>

Hasil dari pengujian dengan menggunakan *Handphone* adalah :

1. Pengujian dilakukan selama 5 hari dengan melakukan beberapa kali pengujian. Jumlah pengujian adalah 61 kali.
2. 7 dari 61 kali pengujian (10%) gagal dilakukan. Gagal disini adalah kegagalan untuk melakukan pengiriman *request* data ke *server*.

3. Kemungkinan kegagalan disebabkan oleh pada waktu pengujian kondisi jaringan GPRS sedang mengalami gangguan, jaringan sibuk, atau *server* dalam kondisi *down*.

Pengujian Sistem Informasi *Yellow pages* dengan menggunakan *Handphone* harus dilakukan berulang-ulang, agar hasil dari pengujian sesuai dengan yang dibutuhkan oleh pengguna dan sesuai dengan yang diinginkan oleh pembuatnya. Maka diperlukan pengujian secara terperinci.

Tabel 5.3 dan tabel 5.4 menjelaskan tahap-tahap pengujian secara rinci yang dilakukan menggunakan *handphone* Nokia E63 untuk menampilkan hasil dari semua proses yang dilakukan.

5.2.2.1.1 Pengujian Menampilkan Hasil Pencarian Data Individu

Tabel 5.3 Pengujian untuk menampilkan hasil pencarian data individu

No.	Deskripsi	Model <i>Handphone</i>	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Pengujian pencarian nama individu dengan 2 (dua) suku kata	Nokia E63	<i>Form</i> pencarian sudah tampil	Masukkan <i>keyword</i> dengan 2(dua) suku kata, kemudian tekan perintah "OK"	<i>Keyword</i> "abdul rahman"	Tampilan hasil nama individu dengan nama "abdul rahman"	Tampilan hasil pencarian data individu dengan nama "abdul rahman"	Diterima
2.	Pengujian pencarian alamat individu dengan 2 (dua) suku kata				<i>Keyword</i> "arifin ahmad"	Tampilan hasil alamat individu dengan alamat "jl.arifin ahmad"	Tampilan hasil pencarian data individu dengan alamat "jl.arifin ahmad"	Diterima
3.	Pengujian dengan data yang tidak ada pada <i>database</i>				<i>Keyword</i> "kambing"	Tampilan pesan data kosong	Tampilan pesan "Data tidak ditemukan. Cari kata kunci lainnya !!"	Diterima

Pengujian secara terperinci pada pencarian data individu berjalan sesuai dengan yang diinginkan. Pencarian data dengan *keyword* nama ataupun alamat individu berjalan dengan baik. Sistem dapat menerima masukan *keyword* dengan menggunakan 2 (dua) suku kata ataupun lebih. Hal ini ditujukan agar pencarian dapat lebih mendetail. Masukan berupa simbol juga dapat diterima oleh sistem. Tetapi sistem akan memberikan pesan bahwa data yang dicari tidak ada didalam *database*.

5.2.2.1.2 Pengujian Menampilkan Hasil Pencarian Data Perusahaan

Tabel 5.4 Pengujian untuk menampilkan hasil pencarian data perusahaan

No.	Deskripsi	Model <i>Handphone</i>	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Pengujian pencarian nama perusahaan dengan 2 (dua) suku kata	Nokia E63	<i>Form</i> pencarian sudah tampil	Masukkan <i>keyword</i> dengan 2(dua) suku kata, kemudian tekan perintah "OK"	<i>Keyword</i> "rumah sakit"	Tampilan hasil nama perusahaan dengan <i>keyword</i> "rumah sakit"	Tampilan hasil pencarian data perusahaan dengan nama "rumah sakit"	Diterima
2.	Pengujian pencarian alamat perusahaan dengan 2 (dua) suku kata				<i>Keyword</i> "ahmad yani"	Tampilan hasil alamat perusahaan dengan alamat "jl.ahmad yani"	Tampilan hasil pencarian data perusahaan dengan alamat "jl.ahmad yani"	Diterima
3.	Pengujian dengan data yang tidak ada pada <i>database</i>			Masukkan <i>keyword</i> nama atau alamat, kemudian tekan perintah "OK"	<i>Keyword</i> "ayam"	Tampilan pesan data kosong	Tampilan pesan "Data tidak ditemukan. Cari kata kunci lainnya !!"	Diterima

Pengujian secara terperinci pada pencarian data perusahaan juga berjalan sesuai dengan yang diinginkan. Pencarian data perusahaan dengan *keyword* nama ataupun alamat sesuai dengan database. Sistem menerima masukan *keyword* dengan menggunakan 2 (dua) suku kata ataupun lebih. Hal ini ditujukan agar pencarian dapat lebih mendetail. Masukan berupa simbol juga dapat diterima oleh sistem. Tetapi sistem akan memberikan pesan bahwa data yang dicari tidak ada didalam *database*.

5.2.3 Analisa Pengujian Sistem

Pengujian yang dilakukan dengan beberapa model *handphone* dan beberapa *provider* telekomunikasi menghasilkan kesimpulan bahwa:

1. Sistem informasi *Yellow Pages* yang telah dibangun pada sisi *handphone* dapat melakukan koneksi ke *server*, menampilkan hasil pencarian data individu dengan *keyword* nama, hasil pencarian data individu dengan *keyword* alamat, hasil pencarian data perusahaan dengan *keyword* nama dan hasil pencarian data perusahaan dengan *keyword* alamat.
2. Seluruh *handphone* yang digunakan untuk pengujian menghasilkan informasi sesuai dengan isi *database*.
3. Sebagian pengujian yang dilakukan mengalami kegagalan untuk melakukan koneksi ke *server* dapat disebabkan beberapa kemungkinan yaitu, pada waktu pengujian kondisi jaringan GPRS sedang mengalami gangguan, jaringan sibuk, atau *server* dalam kondisi *down*.

BAB VI

PENUTUP

6.1 Kesimpulan

Dari penjelasan bab-bab sebelumnya dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem informasi *Yellow Pages* yang dibangun pada sisi *client* dan telah di-*install*-kan pada *handphone* dapat digunakan untuk menampilkan nama dan alamat dari data individu dan perusahaan.
2. Sistem informasi *Yellow Pages* pada sisi *server* yang diakses menggunakan komputer dengan hak akses *administrator* dapat digunakan untuk mengelola data individu dan data perusahaan.
3. Seluruh *handphone* yang digunakan untuk pengujian menghasilkan informasi sesuai dengan isi *database*.
4. Sebagian pengujian yang dilakukan mengalami kegagalan untuk melakukan koneksi ke *server* hal ini disebabkan beberapa kemungkinan yaitu, pada waktu pengujian kondisi jaringan GPRS sedang mengalami gangguan, jaringan sibuk atau *server* dalam kondisi *down*.

6.2 Saran

Beberapa hal yang disarankan dalam pengembangan sistem informasi *Yellow Pages* dengan teknologi J2ME menggunakan jaringan GPRS ini adalah sebagai berikut:

1. Pada pengembangan selanjutnya diharapkan tidak hanya menghasilkan keluaran dalam bentuk teks, tetapi juga dalam bentuk multimedia, misalnya gambar ataupun suara.
2. Pada pengembangan selanjutnya diharapkan dapat melakukan melakukan entri data melalui *handphone*.
3. Pada pengembangan selanjutnya diharapkan dapat diperluas cakupan *database*-nya tidak hanya memberikan layanan informasi *Yellow Pages* di Kota Pekanbaru tetapi di seluruh Indonesia.
4. Pada pengembangan selanjutnya diharapkan tidak hanya menggunakan jaringan GPRS tetapi dapat menggunakan jaringan lainnya, misalnya menggunakan jaringan *Bluetooth* dan *Wi-fi*.

DAFTAR PUSTAKA

- Ariyanto, "*Menguasai Java Programming*", Salemba Infotek, Jakarta, 2009.
- Darmawan, H, Erico, "*Pemrograman Dasar C-Java-C#*", Informatika, Bandung, 2009.
- Indrajani dan Martin, "*Pemrograman Berorientasi Objek dengan Java*", halaman 1-3, PT. Alex Media Komputindo, Jakarta, 2004.
- Putro, Iwan Handoyo, dan Petrus Santoso, dan Stephani Imelda Pella "*Emulasi Aplikasi Pemantauan Ruangan Melalui Handphone Menggunakan Webcam*" Jurnal Teknik Elektro Vol. 5, No. 2, September 2005:102–108 [Online] Available <http://www.petra.ac.id/~puslit/journals/pdf.php?PublishedID=ELK05050208>, diakses 10 Februari 2010.
- Purnama, Rangsang, "*Tuntunan Pemrograman Java Jilid 3*", Prestasi Pustaka, Jakarta, 2005.
- Raharjo, Budi dan Imam Heryanto dan Arif Haryono, "*Tuntunan Pemrograman Java untuk Handphone*", halaman 2-6, Informatika, Bandung, 2007.
- Santoso, Leo Willyanto, dan Sukanto Tedjokusuma, dan Marcel Renaldy Soetanto "*Aplikasi Pelaporan Berita Emergeni Secara Visual dan Tektual Lewat Handphone*" [Online] Available http://fportfolio.petra.ac.id/user_files/03-023/Pelaporan%20Berita.pdf, diakses 28 Agustus 2008.
- Sembiring, Kristianus "*Studi Implementasi dan Perbandingan DES, TDES, AES pada J2ME*" [Online] Available <http://santus.files.wordpress.com/2008/03/makalah1-078.pdf>, diakses 28 Agustus 2008.
- Solahuddin dan Rosa A.S., "*Pemrograman J2ME : Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*" halaman 1-2, Informatika, Bandung, 2006.
- Suhendar dan Hariman Gunadi, "*Visual Modeling Menggunakan UML dan Rational Rose*" halaman 26, 49-56, Informatika, Bandung, 2002.
- Wibisono, Gunawan, "*Teknologi Informasi2: Jaringan dan Manajemen Data*" Modul 1,5 [Online] Available http://www.hotlinkfiles.com/files/327574_t7usn/Modul%201,5.pdf, diakses 28 Agustus 2008.

LAMPIRAN A

MODUL-MODUL PADA SISTEM

A.1 Modul Pada *Handphone*

Modul-modul yang digunakan pada *handphone* untuk semua permintaan informasi ke *server* adalah `setURL()`, `Parsing()`, `hpCariPerusahaanProses()` dan `hpCariIndividuProses()`.

A.1.1 Modul `setURL.java`

```
public class setURL
{
    private String host;
    private String dir;

    public setURL()
    {
        host = "http://localhost/";
        dir = "YPM/mobile/";
        //host = "http://yellowpages.orgfree.com/";
        //dir = "mobile/";
    }

    public String HOST()
    {
        return host;
    }

    public String DIR()
    {
        return dir;
    }
}
```

A.1.2 Modul Parsing.Java

```
import javax.microedition.io.HttpConnection;
import javax.microedition.io.Connector;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.Reader;
import java.io.InputStreamReader;
import java.lang.*;
import org.kxml.*;
import org.kxml.parser.*;

class Parsing {
    private String msg;
    private String isiDataKe[];
    private String isiID[];
    private String isiNAMA[];
    private String isiALAMAT[];
    private String isiNOTELP[];
    private String isiJENIS[];
    private int jumData;
    private int i;
    private int jh;

    private HttpConnection c = null;
    private InputStream is = null;
    private OutputStream os = null;

    public Parsing() {
        this.msg="";
        this.jumData=0;
        this.i=0;
        this.jh=0;
    }

    public void periksaData(String url, String param) throws IOException{
        try {

            c
            =
            (HttpConnection)Connector.open(url,Connector.READ_WRITE);

            c.setRequestMethod(HttpConnection.POST);
            c.setRequestProperty("User-Agent", "Profile/MIDP-1.0
            Configuration/CLDC-1.0");
            c.setRequestProperty("Content-Language", "en-US");
            c.setRequestProperty("Content-Type", "application/x-www-form-
            urlencoded");
            os=c.openOutputStream();
```

```

os.write(param.getBytes());
os.close();

int rc = c.getResponseCode();

if (rc != HttpURLConnection.HTTP_OK)
    throw new IOException("HTTP response code: " + rc);

is = c.openInputStream();

Reader reader = new InputStreamReader(is);
XmlParser parser = new XmlParser(reader);
ParseEvent pe = null;
parser.skip();
pe = parser.read(Xml.START_TAG, null, "data");

this.jumData = Integer.parseInt(pe.getValue("jml"));

boolean trucking = true;
boolean first = true;
int i=0;

this.isiID = new String[this.jumData];
this.isiNAMA = new String[this.jumData];
this.isiALAMAT = new String[this.jumData];
this.isiNOTELP = new String[this.jumData];
this.isiJENIS = new String[this.jumData];

while (trucking) {
    pe = parser.read();
    if (pe.getType() == Xml.START_TAG) {
        String name = pe.getName();
        if (name.equals("row")) {
            i = Integer.parseInt(pe.getValue("num"));
            while ((pe.getType() != Xml.END_TAG) ||
(pe.getName().equals(name) == false)) {
                pe = parser.read();
                if (pe.getType() == Xml.START_TAG &&
pe.getName().equals("id")) {
                    pe = parser.read();
                    this.isiID[i] = pe.getText();
                }
                if (pe.getType() == Xml.START_TAG &&
pe.getName().equals("nama")) {
                    pe = parser.read();
                    this.isiNAMA[i] = pe.getText();
                }
                if (pe.getType() == Xml.START_TAG &&

```

```

pe.getName().equals("alamat")) {
    pe = parser.read();
    this.isiALAMAT[i] = pe.getText();
}
if (pe.getType() == Xml.START_TAG &&
pe.getName().equals("notelp")) {
    pe = parser.read();
    this.isiNOTELP[i] = pe.getText();
}
if (pe.getType() == Xml.START_TAG &&
pe.getName().equals("jenis")) {
    pe = parser.read();
    this.isiJENIS[i] = pe.getText();
}
}
i++;
} else {
while ((pe.getType() != Xml.END_TAG) ||
(pe.getName().equals(name) == false))
    pe = parser.read();
}
}
if (pe.getType() == Xml.END_TAG &&
pe.getName().equals("data"))
    trucking = false;
}
} finally {
    if(is!=null){
        is.close();
    }
}
}

public void setJumData(String url, String param) throws IOException{
    try {

        c
        =
        (HttpURLConnection)Connector.open(url,Connector.READ_WRITE);

        c.setRequestMethod(HttpURLConnection.POST);
        c.setRequestProperty("User-Agent", "Profile/MIDP-1.0
Configuration/CLDC-1.0");
        c.setRequestProperty("Content-Language", "en-US");
        c.setRequestProperty("Content-Type", "application/x-www-form-
urlencoded");
        os=c.openOutputStream();

```

```

        os.write(param.getBytes());
        os.close();

        int rc = c.getResponseCode();

        if (rc != HttpURLConnection.HTTP_OK)
            throw new IOException("HTTP response code: " + rc);

        is = c.openInputStream();

        Reader reader = new InputStreamReader(is);
        XmlParser parser = new XmlParser(reader);
        ParseEvent pe = null;
        parser.skip();
        pe = parser.read(Xml.START_TAG, null, "data");

        this.jumData = Integer.parseInt(pe.getValue("jml"));

    } finally {
        if (is != null) {
            is.close();
        }
    }
}

public int getJumData() {
    return this.jumData;
}

public int getDataKe(int idx) {
    return Integer.parseInt(this.isiDataKe[idx]);
}

public String getId(int idx) {
    if (this.isiID[idx] == null)
        return "-";
    else
        return this.isiID[idx];
}

public String getNama(int idx) {
    if (this.isiNAMA[idx] == null)
        return "-";
    else
        return this.isiNAMA[idx];
}

```

```

public String getAlamat(int idx) {
    if(this.isiALAMAT[idx]==null)
        return "-";
    else
        return this.isiALAMAT[idx];
}

public String getNoTelp(int idx) {
    if(this.isiNOTELP[idx]==null)
        return "-";
    else
        return this.isiNOTELP[idx];
}

public String getJenis(int idx) {
    if(this.isiJENIS[idx]==null)
        return "-";
    else
        return this.isiJENIS[idx];
}

public String getDataIndividu(int hal) {
    int hlm=5;
    int awal=0;
    awal=awal+(hlm*(hal-1));
    int batas=(hlm*hal);

    if(hal==getHalamanKe()){
        batas=getJumData();
    }
    if(getJumData()==0){
        batas=0;
        awal=0;
    }

    for (int i = awal; i < batas; i++) {
        this.msg = this.msg+
            "NAMA : "+getNama(i)+"\n"+
            "ALAMAT : "+getAlamat(i)+"\n"+
            "NO.TELP : "+getNoTelp(i)+
            "\n-----\n\n";
    }
    return this.msg;
}

public String getDataPerusahaan(int hal) {
    int hlm=5;
    int awal=0;
    awal=awal+(hlm*(hal-1));

```



```

        int batas=(hlm*hal);

        if(hal==getHalamanKe()){
            batas=getJumData();
        }
        if(getJumData()==0){
            batas=0;
            awal=0;
        }

        for (int i = awal; i < batas; i++) {
            this.msg = this.msg+
                "NAMA : "+getNama(i)+"\n"+
                "ALAMAT : "+getAlamat(i)+"\n"+
                "NO.TELP : "+getNoTelp(i)+"\n"+
                "JENIS : "+getJenis(i)+
                "\n-----\n\n";
        }
        return this.msg;
    }

    public int getHalamanKe(){
        int hlm=5;
        this.jh=getJumData() / hlm;
        if(getJumData() % hlm != 0){
            jh = jh+1;
        }
        return this.jh;
    }
}

```

A.1.3 Modul hpCariIndividuProses.Java

```

import java.io.InputStream;
import java.io.IOException;

public class hpCariIndividuProses
{
    private String url = "";
    private String param = "";
    private String hq = "";
    private InputStream dis;
    private Parsing prs;
    private String dataHasil;
}

```

```

private int jumData;

public hpCariIndividuProses() { }

public void periksaData(String url, String param, int hal) throws IOException
{
    prs = new Parsing();
    prs.periksaData(url, param);
    this.jumData = prs.getJumData();
    this.dataHasil = prs.getDataIndividu(hal);
}

public String DATA() { return this.dataHasil; }
public int getJumData() { return this.jumData; }
}

```

A.1.3 Modul hpCariPerusahaanProses.Java

```

import java.io.InputStream;
import java.io.IOException;

public class hpCariPerusahaanProses
{
    private String url = "";
    private String param = "";
    private String hq = "";
    private InputStream dis;
    private Parsing prs;
    private String dataHasil;
    private int jumData;

    public hpCariPerusahaanProses() { }

    public void periksaData(String url, String param, int hal) throws IOException
    {
        prs = new Parsing();
        prs.periksaData(url, param);
        this.jumData = prs.getJumData();
        this.dataHasil = prs.getDataPerusahaan(hal);
    }

    public String DATA() { return this.dataHasil; }
    public int getJumData() { return this.jumData; }
}

```

A.2 Modul Pada *Server*

Modul-modul yang digunakan untuk melakukan proses *query* pada *server* adalah class.php

A.2.1 Modul Class.php

```
<?
class ConnectionMySQL {
    var $uid;
    var $pwd;
    var $host;
    var $db;
    var $conn;

    function ConnectionMySQL(){
        $this->host = "localhost";
        $this->uid = "root";
        $this->pwd = "";
        $this->db = "xDB";
    }

    function setHost($host){ $this->host = $host; }
    function setUser($uid){ $this->uid = $uid; }
    function setPassword($pwd){ $this->pwd = $pwd; }
    function setData($db){ $this->db = $db; }

    function getHost(){ return $this->host; }
    function getUser(){ return $this->uid; }
    function getPassword(){ return $this->pwd; }
    function getData(){ return $this->db; }

    function openConnection($host,$uid,$pwd,$db){
        $this->setHost($host);
        $this->setUser($uid);
        $this->setPassword($pwd);
        $this->setData($db);

        $this->conn=mysql_connect($this->getHost(),$this->getUser(),$this-
>getPassword());
        mysql_select_db($this->getData(),$this->conn);
        return $this->conn;
    }
    function closeConnection(){
        mysql_close($this->conn);
        return true;
    }
}
```

```

}
class DatabaseMySQL {
    var $sql;
    var $que;
    var $con;
    var $val;
    var $has;

    function DatabaseMySQL(){
        $this->con = new ConnectionMySQL();
    }
    function setQuery($sql){ $this->sql = $sql; }

    function Execute(){
        $this->con->openConnection($this->con->getHost(),$this->con->getUser(),$this->con->getPassword(),$this->con->getData());
        $this->que=mysql_query($this->sql);
        return $this->que;
        $this->closeConnection();
    }
    function isSuccess(){
        if (!$this->que){
            return false;
        }else{
            return true;
        }
    }
    function recordCount($exe){
        return mysql_num_rows($exe);
    }
    function getValueXML($exc){
        $this->has = "";
        $i=0;
        $n=mysql_num_fields($exc);
        while($s=mysql_fetch_row($exc)){
            $b="";
            for ($j=0;$j<$n;$j++){
                $y = mysql_field_name($exc,$j);
                $b = $b."<".$y.">".$s[$j]."</".$y.">". "\n";
            }
            $this->has = $this->has."<row num=" . $i . ">\n" . $b . "</row>". "\n";
            $i++;
        }
        return "<data jml=" . mysql_num_rows($exc) . ">\n" . $this->has . "</data>";
    }
    function getValueXMLforMobileIndividu($exc,$awal,$batas){
        $this->has = "";
        $i=$awal;

        $k=0;
        $n=mysql_num_rows($exc);

```

```

while($s=mysql_fetch_array($exc)){
    $id[$k]=$s["id"];
    $nama[$k]=$s["nama"];
    $alamat[$k]=$s["alamat"];
    $notelp[$k]=$s["notelp"];
    $k++;
}
for($i=$awal;$i<$batas;$i++){
    $b="";
    $b = $b."<row num=\"" . $i . ">\n";
    $b = $b."<id>". $id[$i]. "</id>". "\n";
    $b = $b."<nama>". $nama[$i]. "</nama>". "\n";
    $b = $b."<alamat>". $alamat[$i]. "</alamat>". "\n";
    $b = $b."<notelp>". $notelp[$i]. "</notelp>". "\n";
    $b = $b."</row>\n";

    $this->has = $this->has.$b;
}
return "<data jml=\"" .mysql_num_rows($exc). ">\n". $this->has. "</data>";
}
function getValueXMLforMobilePerusahaan($exc,$awal,$batas){
    $this->has = "";
    $i=$awal;

    $k=0;
    $n=mysql_num_rows($exc);
    while($s=mysql_fetch_array($exc)){
        $id[$k]=$s["id"];
        $nama[$k]=$s["nama"];
        $alamat[$k]=$s["alamat"];
        $notelp[$k]=$s["notelp"];
        $jenis[$k]=$s["jenis"];
        $k++;
    }
    for($i=$awal;$i<$batas;$i++){
        $b="";
        $b = $b."<row num=\"" . $i . ">\n";
        $b = $b."<id>". $id[$i]. "</id>". "\n";
        $b = $b."<nama>". $nama[$i]. "</nama>". "\n";
        $b = $b."<alamat>". $alamat[$i]. "</alamat>". "\n";
        $b = $b."<notelp>". $notelp[$i]. "</notelp>". "\n";
        $b = $b."<jenis>". $jenis[$i]. "</jenis>". "\n";
        $b = $b."</row>\n";

        $this->has = $this->has.$b;
    }
    return "<data jml=\"" .mysql_num_rows($exc). ">\n". $this->has. "</data>";
}
function parsingXML($val){
    if(class_exists('SimpleXMLElement')){
        $this->xml = new SimpleXMLElement($val);
    }
}

```

```

        return $this->xml;
    }
}
function recordKe($datake,$field,$hpXML){
    return $hpXML->row[$datake]->$field;
}
}

class individu{
    var $sql;
    var $rs;
    var $hasSQLsel;
    var $isSQLsel;
    var $hasSQLins;
    var $isSQLins;
    var $hasSQLlupd;
    var $isSQLlupd;
    var $hasSQLdel;
    var $isSQLdel;

    function individu($sql){
        $this->rs = new DatabaseMySQL();
        $this->sql = $sql;
    }
    function selectIndividu(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLsel = $this->rs->getValueXML($this->rs->Execute());
        $this->isSQLsel = $this->rs->isSuccess();
        return $this->hasSQLsel;
    }
    function insertIndividu(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLins = $this->rs->Execute();
        $this->isSQLins = $this->rs->isSuccess();
        return $this->hasSQLins;
    }
    function updateIndividu(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLlupd = $this->rs->Execute();
        $this->isSQLlupd = $this->rs->isSuccess();
        return $this->hasSQLlupd;
    }
    function deleteIndividu(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLdel = $this->rs->Execute();
        $this->isSQLdel = $this->rs->isSuccess();
        return $this->hasSQLdel;
    }
    function isSelectIndividu(){
        return $this->isSQLsel;
    }
}

```

```

    }
    function isInsertIndividu(){
        return $this->isSQLins;
    }
    function isUpdateIndividu(){
        return $this->isSQLupd;
    }
    function isDeleteIndividu(){
        return $this->isSQLdel;
    }
}
class perusahaan{
    var $sql;
    var $rs;
    var $hasSQLsel;
    var $isSQLsel;
    var $hasSQLins;
    var $isSQLins;
    var $hasSQLupd;
    var $isSQLupd;
    var $hasSQLdel;
    var $isSQLdel;

    function perusahaan($sql){
        $this->rs = new DatabaseMySQL();
        $this->sql = $sql;
    }
    function selectPerusahaan(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLsel = $this->rs->getValueXML($this->rs->Execute());
        $this->isSQLsel = $this->rs->isSuccess();
        return $this->hasSQLsel;
    }
    function insertPerusahaan(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLins = $this->rs->Execute();
        $this->isSQLins = $this->rs->isSuccess();
        return $this->hasSQLins;
    }
    function updatePerusahaan(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLupd = $this->rs->Execute();
        $this->isSQLupd = $this->rs->isSuccess();
        return $this->hasSQLupd;
    }
    function deletePerusahaan(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLdel = $this->rs->Execute();
        $this->isSQLdel = $this->rs->isSuccess();
        return $this->hasSQLdel;
    }
}

```

```

function isSelectPerusahaan(){
    return $this->isSQLsel;
}
function isInsertPerusahaan(){
    return $this->isSQLins;
}
function isUpdatePerusahaan(){
    return $this->isSQLupd;
}
function isDeletePerusahaan(){
    return $this->isSQLdel;
}
}

class Login{
    var $sql;
    var $rs;
    var $hasSQLsel;
    var $isSQLsel;
    var $hasSQLins;
    var $isSQLins;
    var $hasSQLupd;
    var $isSQLupd;
    var $hasSQLdel;
    var $isSQLdel;
    var $parseXML;
    var $exe;

    function Login($sql){
        $this->rs = new DatabaseMySQL();
        $this->sql = $sql;
    }
    function selectLogin(){
        $this->rs->setQuery($this->sql);
        $this->exe = $this->rs->Execute();
        $this->hasSQLsel = $this->rs->getValueXML($this->exe);
        $this->parseXML = $this->rs->parsingXML($this->hasSQLsel);
        $this->isSQLsel = $this->rs->isSuccess();
        return $this->hasSQLsel;
    }
    function jumlahData(){
        return $this->rs->recordCount($this->exe);
    }
    function getUser($idx){
        return $this->rs->recordKe($idx,'id',$this->parseXML);
    }
    function getPassword($idx){
        return $this->rs->recordKe($idx,'pass',$this->parseXML);
    }
    function insertLogin(){
        $this->rs->setQuery($this->sql);
    }
}

```



```

        $this->hasSQLins = $this->rs->Execute();
        $this->isSQLins = $this->rs->isSuccess();
        return $this->hasSQLins;
    }
    function updateLogin(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLupd = $this->rs->Execute();
        $this->isSQLupd = $this->rs->isSuccess();
        return $this->hasSQLupd;
    }
    function deleteLogin(){
        $this->rs->setQuery($this->sql);
        $this->hasSQLdel = $this->rs->Execute();
        $this->isSQLdel = $this->rs->isSuccess();
        return $this->hasSQLdel;
    }
    function isSelectLogin(){
        return $this->isSQLsel;
    }
    function isInsertLogin(){
        return $this->isSQLins;
    }
    function isUpdateLogin(){
        return $this->isSQLupd;
    }
    function isDeleteLogin(){
        return $this->isSQLdel;
    }
}

class Karakter{
    function Karakter(){}

    function validasiKarakter($isi){
        $trans = get_html_translation_table(HTML_ENTITIES);
        $str = "$isi";
        $encoded = strtr($str, $trans);
        $hasil = str_replace("", "", htmlspecialchars($encoded));

        return $hasil;
    }
    function getKarakterAsli($isi){
        $trans = get_html_translation_table(HTML_ENTITIES);
        $trans = array_flip($trans);
        $original = strtr($isi, $trans);
        return $original;
    }
}

$r = new ConnectionMySQL();
$r->setHost("localhost");

```

```
$r->setUser("root");  
$r->setPassword("");  
$r->setData("ypm");  
$r->openConnection($r->getHost(),$r->getUser(),$r->getPassword(),$r->getData());
```

LAMPIRAN B

DAFTAR ISTILAH

<i>Platform independent</i>	: tidak tergantung mesin atau sistem operasi.
<i>Scripting language</i>	: kode-kode bahasa pemrograman.
<i>Full programming language</i>	: bahasa pemrograman yang lengkap.
<i>Web browser</i>	: aplikasi untuk menjelajahi informasi berbasis <i>web</i> .
<i>Java Virtual Machine</i>	: suatu program khusus untuk menerjemahkan dan mengeksekusi setiap baris instruksi pemrograman Java.
<i>Multiplatform</i>	: dapat dioperasikan pada banyak <i>platform</i> .
<i>Interface</i>	: antarmuka.
<i>Client</i>	: pengguna sistem.
<i>Server</i>	: yang menyediakan layanan sistem.
<i>Database</i>	: basis data yang digunakan untuk menyimpan data.
<i>Java database connector</i>	: kelas yang disediakan oleh Java untuk koneksi ke <i>database</i> .
<i>Mobile device</i>	: peralatan komunikasi yang dapat dibawa kemana-mana.
<i>Wireless</i>	: jaringan tanpa menggunakan kabel.

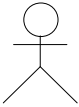
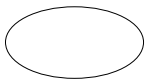




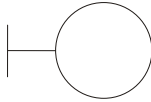

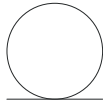
Request : meminta data kepada *server* berdasarkan masukan yang dikirimkan.

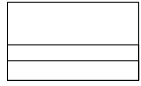
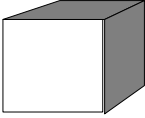
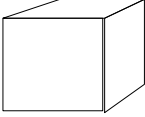
Response : balasan dari *server* akibat *request* dari *client*.

General Packet Radio Service (GPRS) : sistem transmisi berbasis paket untuk GSM.

LAMPIRAN C

DAFTAR SIMBOL

SIMBOL	KETERANGAN SIMBOL
	Pengguna Sistem (Aktor)
	Proses (<i>Use case</i>)
	Aktivitas (<i>Activity</i>)
	Keputusan (<i>decision</i>)
	Status mulai (<i>start state</i>)
	Status selesai (<i>end state</i>)
	Antarmuka (<i>boundary</i>)
	Proses (<i>control</i>)
	Tabel basis data (<i>entity</i>)

SIMBOL	KETERANGAN SIMBOL
	Kelas (<i>class</i>)
	<i>Processor</i>
	Peralatan tambahan (<i>device</i>)

LAMPIRAN D

PETUNJUK PENGINSTALAN APLIKASI PADA *HANDPHONE*

Langkah-langkah dalam menginstal aplikasi sistem informasi yellow pages pada *handphone* adalah sebagai berikut :

1. Download terlebih dahulu aplikasi yellow pages di situs <http://yellowpages.orgfree.com> dengan menggunakan media komputer atau laptop.
2. Setelah selesai mendownload, kirim file yellowpages.jar ke *handphone* anda.
3. Kemudian jalankan file tersebut. Untuk sebahagian *handphone* aplikasi dapat langsung digunakan. Tetapi ada juga *handphone* yang meminta untuk di install terlebih dahulu.

DAFTAR LAMPIRAN

Lampiran	Halaman
A. MODUL-MODUL PADA SISTEM	A-1
B. DAFTAR ISTILAH	B-1
C. DAFTAR SIMBOL	C-1
D. PETUNJUK PENGINSTALAN APLIKASI PADA HANDPHONE	D-1

DAFTAR TABEL

Tabel	Halaman
4.1 Karakteristik Pengguna	IV-9
4.2 <i>Use case diagram</i> (pengaksesan melalui <i>handphone</i>)	IV-10
4.3 Deskripsi <i>Class diagram</i> pada J2ME	IV-14
4.4 Deskripsi <i>sequence diagram</i> menampilkan data individu	IV-15
4.5 Deskripsi <i>sequence diagram</i> menampilkan data Perusahaan	IV-16
4.6 <i>Use case diagram</i> (pengaksesan melalui <i>Komputer</i>)	IV-18
4.7 Deskripsi <i>Class diagram</i> pada Server	IV-25
4.8 Deskripsi <i>sequence diagram login administrator</i>	IV-26
4.9 Deskripsi <i>sequence diagram</i> ubah <i>password administrator</i>	IV-26
4.10 Deskripsi <i>sequence diagram</i> pengelolaan data individu	IV-27
4.11 Deskripsi <i>sequence diagram</i> pengelolaan data perusahaan	IV-28
4.12 Deskripsi perancangan antarmuka nilai pada <i>handphone</i>	IV-30
4.13 Deskripsi Perancangan Antarmuka Keluaran data pada <i>handphone</i>	IV-31
5.1 Deskripsi antarmuka menu utama sistem pada sisi <i>handphone</i>	V-4
5.2 Pengujian dengan beberapa <i>handphone</i>	V-14
5.3 Pengujian untuk menampilkan hasil pencarian data individu	V-19
5.4 Pengujian untuk menampilkan hasil pencarian data individu	V-20

DAFTAR GAMBAR

Gambar	Halaman
2.1 Arsitektur J2SE	II-6
2.2 Arsitektur J2EE.....	II-7
2.3 Arsitektur J2ME.....	II-11
2.4 Siklus hidup J2ME	II-14
2.5 Arsitektur GPRS	II-19
2.6 Contoh <i>use case diagram</i>	II-21
2.7 Contoh <i>activity diagram</i>	II-21
2.8 Contoh <i>statechart diagram</i>	II-22
2.9 Contoh <i>class diagram</i>	II-23
2.10 Contoh <i>sequence diagram</i>	II-24
2.11 Contoh <i>deployment diagram</i>	II-24
3.1 <i>Flowchart</i> langkah-langkah penelitian	III-1
4.1 Blok diagram sistem informasi yellow pages.....	IV-3
4.2 Model Sistem	IV-4
4.3 Alur Proses Pada <i>Handphone</i>	IV-6
4.4 Alur Proses Pada <i>Komputer Administrator</i>	IV-7
4.5 Aliran <i>Use Case Diagram</i> (pengaksesan melalui <i>Handphone</i>).....	IV-10
4.6 <i>Activity Diagram</i> menampilkan data individu.....	IV-11
4.7 <i>Activity Diagram</i> menampilkan data perusahaan	IV-11
4.8 <i>Statechart Diagram</i> menampilkan data individu.....	IV-12
4.9 <i>Statechart Diagram</i> menampilkan data perusahaan	IV-13
4.10 <i>Class Diagram</i> pada J2ME.....	IV-13
4.11 <i>Sequence Diagram</i> menampilkan data individu	IV-15
4.12 <i>Sequence Diagram</i> menampilkan data perusahaan.....	IV-16
4.13 <i>Deployment Diagram</i> sisi <i>handphone</i>	IV-17
4.14 Aliran <i>Use Case Diagram</i> (pengaksesan melalui <i>Komputer</i>)	IV-18
4.15 <i>Activity Diagram</i> Proses <i>Login</i>	IV-19
4.16 <i>Activity Diagram</i> Proses <i>Ubah Password</i>	IV-19

4.17 <i>Activity Diagram</i> Penambahan Data	IV-20
4.18 <i>Activity Diagram</i> Ubah Data	IV-20
4.19 <i>Activity Diagram</i> Hapus Data	IV-21
4.20 <i>Statechart Diagram</i> Proses Login	IV-21
4.21 <i>Statechart Diagram</i> Menampilkan data Individu	IV-22
4.22 <i>Statechart Diagram</i> Menampilkan data Perusahaan	IV-22
4.23 <i>Statechart Diagram</i> Mengubah Password	IV-22
4.24 <i>Statechart Diagram</i> Menambah Data	IV-23
4.25 <i>Statechart Diagram</i> Mengubah Data	IV-23
4.26 <i>Statechart Diagram</i> Menghapus Data	IV-23
4.27 <i>Class diagram</i> pada J2ME	IV-24
4.28 <i>Sequence diagram</i> login administrator	IV-25
4.29 <i>Sequence diagram</i> ubah password	IV-26
4.30 <i>Sequence diagram</i> pengelolaan data individu	IV-27
4.31 <i>Sequence diagram</i> pengelolaan data perusahaan	IV-28
4.32 <i>Deployment Diagram</i> sisi Server	IV-29
4.33 Perancangan menu utama pada <i>handphone</i>	IV-30
4.34 Perancangan keluaran data pada <i>handphone</i>	IV-31
5.1 Antarmuka menu utama sistem pada sisi <i>handphone</i>	V-4
5.2 Menampilkan <i>Splash</i> (J2ME Wireless Toolkit)	V-5
5.3 Menampilkan Menu Pencarian Data Individu (J2ME Wireless Toolkit).....	V-5
5.4 Menampilkan Menu Pencarian Data Perusahaan (J2ME Wireless Toolkit)	V-6
5.5 Menampilkan Form Input Keyword Berdasarkan Nama Pada Data Individu (J2ME Wireless Toolkit)	V-6
5.6 Menampilkan Form Input <i>Keyword</i> Berdasarkan Alamat Pada Data Individu (J2ME Wireless Toolkit)	V-7
5.7 Menampilkan Form <i>Input Keyword</i> Berdasarkan Nama Pada Data Perusahaan (J2ME Wireless Toolkit).....	V-7
5.8 Menampilkan Form <i>Input Keyword</i> Berdasarkan Alamat Pada Data Perusahaan (J2ME Wireless Toolkit).....	V-8

5.9	Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data Individu (J2ME Wireless Toolkit)	V-8
5.10	Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data Individu (J2ME Wireless Toolkit)	V-9
5.11	Menampilkan Hasil Pencarian Berdasarkan Nama Pada Data Perusahaan (J2ME Wireless Toolkit)	V-9
5.12	Menampilkan Hasil Pencarian Berdasarkan Alamat Pada Data Perusahaan (J2ME Wireless Toolkit)	V-10
5.13	Antarmuka <i>Login Administrator</i>	V-11
5.14	Antarmuka Pengelolaan Data Individu	V-11
5.15	Antarmuka Pengelolaan Data Perusahaan	V-12
5.16	Antarmuka Ubah <i>Password Administrator</i>	V-12

DAFTAR SINGKATAN

Singkatan

API	= <i>Application Programming Interface</i>
ASP	= <i>Active Server Pages</i>
BTS	= <i>Base Transceiver Station</i>
CDC	= <i>Connected Device Configuration</i>
CGI	= <i>Comman Gateway Interface</i>
CLDC	= <i>Connected Limited Device Configuration</i>
EJB	= <i>Enterprise Java Beans</i>
GGSN	= <i>Gateway GPRS Support Node</i>
GPRS	= <i>General Packet Radio Service</i>
GSM	= <i>Global System for Mobile Communications</i>
GUI	= <i>Graphical User Interface</i>
HSDPA	= <i>High Speed Download Packet Access</i>
HTML	= <i>Hyper Text Markup Language</i>
JDBC	= <i>Java Database Connectivity</i>
JDK	= <i>Java Developer Kit</i>
JRE	= <i>Java Runtime Environment</i>
JSP	= <i>Java Server Pages</i>
JVM	= <i>Java Virtual Machine</i>
J2EE	= <i>Java 2 Enterprise Edition</i>
J2ME	= <i>Java 2 Micro Edition</i>
J2SE	= <i>Java 2 Standard Edition</i>
KVM	= <i>Kilobyte Virtual Machine</i>
MIDP	= <i>Mobile Information Device Profile</i>
MS	= <i>Mobile Station</i>
OOAD	= <i>Object Oriented Analisys Design</i>
PADP	= <i>Personal Digital Assistant Profile</i>
PAN	= <i>Personal Area Networking</i>
PDN	= <i>Public Data Network</i>

RMI = *Remote Method Invocation*
SGSN = *Serving GPRS Support Node*
SQL = *Structure Query Language*
UMTS = *Universal Mobile Telecommunications System*
WCDMA = *Wideband Code Division Multiple Access*
WWW = *World Wide Web*
XML = *eXtensible Markup Language*

DAFTAR RIWAYAT HIDUP



Penulis dilahirkan di Pekanbaru pada tanggal 12 Maret 1986 dari Ayahanda Ugi Sutardi dan Ibunda Aisyah. Penulis merupakan anak kelima dari lima bersaudara. Pada tahun 1992, penulis masuk Sekolah Dasar Negeri 010 Rumbai dan menyelesaikannya pada tahun 1998. Pekanbaru adalah sebuah Ibukota dari Propinsi Riau. Menamatkan Sekolah Menengah Pertama Negeri 019 Pekanbaru tahun 2001 dan menamatkan Sekolah Menengah Kejuruan Negeri 5 Pekanbaru pada tahun 2004.

Setelah menamatkan pendidikan formal, pada tahun 2004, penulis masuk Universitas Islam Negeri Sultan Syarif Kasim, sebagai mahasiswa pada Jurusan Teknik Informatika hingga menyelesaikan Laporan Tugas Akhir ini.

Selama menjadi mahasiswa, penulis sering mengikuti berbagai seminar IT dan aktifitas mahasiswa lainnya.