

PROTOTYPE APLIKASI INFO REKENING LISTRIK PLN (IRT) BERBASIS *MOBILE*

(Studi Kasus: PT. PLN Persero Distribusi Jakarta Raya dan Tangerang)



Oleh :

INDAH FAUZIAH

206091004057

**PROGRAM STUDI TEHNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDYATULLAH
JAKARTA
2011 M / 1432 H**

PROTOTYPE APLIKASI INFO REKENING LISTRIK PLN

BERBASIS *MOBILE*

(Studi Kasus : PT. PLN Persero Distribusi Jakarta Raya dan Tangerang)

Indah Fauziah

206091004057

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
SYARIF HIDAYATULLAH
JAKARTA
2011 M/1433H**

PROTOTYPE APLIKASI INFO REKENING LISTRIK PLN

BERBASIS *MOBILE*

(Studi Kasus : PT.PLN Persero Distribusi Jakarta Raya dan Tangerang)

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

Sarjana Komputer

Fakultas Sains dan Teknologi

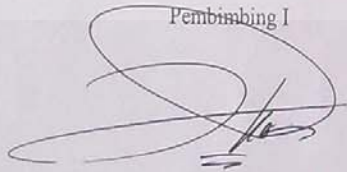
Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh :

Indah Fauziah
206091004057

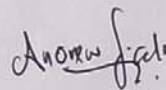
Menyetujui,

Pembimbing I



Herlino Nanang, MT
NIP. 19730402 200112 2 001

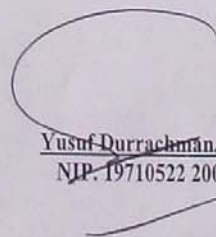
Pembimbing II



Andrew Fiade, M.Kom
NIP. 19820811 200912 1 004

Mengetahui,

Ketua Program Studi Teknik Informatika



Yusuf Durrachman, M.Sc, MIT
NIP. 19710522 200604 1 002

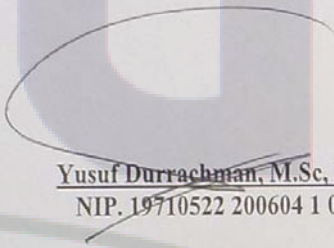
PENGESAHAN UJIAN

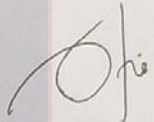
Skripsi yang berjudul "Prototype Aplikasi Info Rekening Listrik PLN berbasis *Mobile* (Studi Kasus : PT. PLN Persero Distribusi Jakarta Raya dan Tangerang)" yang ditulis oleh Indah Fauziah, NIM 206091004057 telah diuji dan dinyatakan lulus dalam sidang *Munaqosyah* Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada hari Kamis 01 Desember 2011. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) Program Studi Teknik Informatika.

Menyetujui,

Penguji I

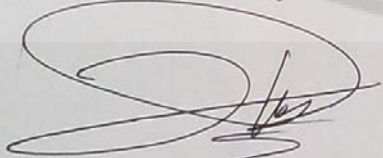
Penguji II

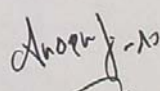

Yusuf Durrachman, M.Sc, MIT
NIP. 19710522 200604 1 002


Arini, MT, M.Eng
NIP. 19760131 200912 2 001

Pembimbing I

Pembimbing II



Herlino Nanang, MT
NIP. 19721223 200710 2 004


Andrew Fiade, M.Kom
NIP. 19760131 200912 2 001

Mengetahui,

Dekan Fakultas Sains dan Teknologi

Ketua Program Studi Teknik
Informatika


Dr. Syopiansyah Jaya Putra, M.Sis
NIP. 19680117 200112 1 001


Yusuf Durrachman, M.Sc, MIT
NIP. 19710522 200604 1 002

PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI DAN BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Jakarta, Desember 2011

Indah Fauziah
206091004057

ABSTRAK

Indah Fauziah, Prototype Aplikasi Info Rekening Listrik PLN (IRT) berbasis *Mobile* (Studi Kasus : PT. PLN Persero Distribusi Jakarta Raya dan Tangerang), dibawah bimbingan **Herlino Nanang** dan **Andrew Fiade**.

Perkembangan teknologi informasi dan komunikasi saat ini telah menghasilkan pelayanan-pelayanan baru termasuk melalui *mobile device* seperti *handphone*, *PDA phone*, *Smartphone*, sampai *pocket pc*. Tujuan penulisan skripsi ini yaitu Mengembangkan sistem pelayanan informasi rekening listrik PLN berbasis *mobile*, untuk memudahkan pelanggan dalam memperoleh informasi yang dibutuhkan khususnya informasi rekening listrik. PT. PLN Persero Disribusi Jakarta Raya dan Tangerang adalah perusahaan Distribusi tenaga listrik wilayah Jakarta dan Tangerang. Oleh karena itu aplikasi ini dapat berguna untuk mempermudah PLN dalam memberikan informasi rekening listrik kepada pelanggan khususnya di wilayah Jakarta dan Tangerang. Metode penelitian yang digunakan yaitu metode pengumpulan data, metode pengembangan sistem. Metode pengumpulan data yang digunakan yaitu studi lapangan, studi pustaka, studi *literature* sejenis. Sedangkan untuk metode pengembangan sistem menggunakan metode pengembangan sistem prototype yang meliputi Tahap mendengarkan pelanggan, tahap membangun dan memperbaiki prototype, tahap pengujian prototype. Prototype Aplikasi Info Rekening Listrik PLN ini dijabarkan menggunakan diagram UML. Kesimpulan dari prototype aplikasi info rekening listrik PLN ini adalah dapat memudahkan pelanggan dalam memperoleh informasi seputar rekening listrik serta menunjang media pelayanan pelanggan melalui perangkat *mobile*. Diharapkan aplikasi ini dapat diintegrasikan oleh pihak PLN serta dikembangkan lebih lanjut dengan *operating system* Android.

V Bab + 101 Halaman + Daftar Pustaka + Lampiran

Kata Kunci : Aplikasi, informasi, *Mobile Device*, Metode Prototype, Pelanggan.

KATA PENGANTAR

Bismillahirrahmaanirrahim,

Syukur alhamdulillah penulis haturkan kehadiran Allah SWT atas rahmat dan karuniaNya, selanjutnya shalawat dan salam penulis sampaikan kepada junjungan kita Nabi besar Muhammad SAW beserta sahabat dan keluarganya yang telah membawa kita dari alam kegelapan kealam yang berilmu pengetahuan sehingga penulis dapat menyelesaikan Skripsi ini dengan judul **“Prototype Info Rekening Listrik PLN berbasis *Mobile* (Studi kasus: PT. PLN Persero Distribusi Jakarta Raya dan Tangerang)”**. Yang merupakan syarat dalam menyelesaikan Program Studi S1 pada program studi Teknik Informatika, Fakultas Sains dan Teknologi di Universitas Islam Negeri (UIN) Syarif Hidayatullah Jakarta.

Selama penulisan Skripsi ini, penulis banyak mendapat bantuan dan bimbingan dari berbagai pihak, untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. DR. Syopiansyah Jaya Putra, M.Sis, selaku Dekan Fakultas Sains dan Teknologi.
2. Yusuf Durrachman, M.IT, selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi dan Viva Arifin, ST, M.MSI, selaku Sekretaris Program Studi Teknik Informatika Fakultas Sains dan Teknologi.

3. Bapak Herlino Nanang, MT selaku Dosen Pembimbing I dan Bapak Andrew Fiade, M.Kom selaku Dosen Pembimbing II. Atas waktu dan perhatian yang telah diluangkan, serta arahan dan bimbingan terhadap kesulitan yang penulis hadapi selama mengerjakan skripsi ini.
4. Seluruh dosen dan staf karyawan Program Studi Teknik Informatika Universitas Islam Negeri (UIN) Syarif Hidayatullah Jakarta yang telah membimbing penulis selama masa kuliah.
5. Pimpinan dan seluruh karyawan/i PT. PLN Persero Distribusi Jakarta Raya dan Tangerang, khususnya bagian IT yang telah banyak memberikan informasi kepada penulis.
6. Ibu dan Bapak tercinta, untuk ketulusan doa serta dukungan baik secara moril dan materil serta kasih sayang yang diberikan selama ini.
7. Kakak dan Adik-adik tercinta, untuk doa dan dukungannya yang selalu menyertai.
8. Terimakasih juga untuk Wahyudin Ramadhan, yang telah memberikan semangat dan motivasi kepada penulis selama ini.
9. Seluruh teman-teman angkatan 2006 Teknik Informatika yang tidak bisa disebutkan satu per satu.
10. Dan seluruh pihak yang telah membantu penulis dalam menyelesaikan Skripsi.

Akhirnya penulis menyadari sepenuhnya bahwa Skripsi ini jauh dari sempurna, baik isi, pemilihan bahasa maupun sistematika penulisannya, namun penulis mengharapkan bantuan berupa saran dan kritik yang

membangun dari semua pihak demi kesempurnaan dan mutu penulisan Skripsi ini.

Akhir kata semoga Skripsi ini dapat bermanfaat bagi kita semua serta dapat mempercepat mahasiswa di dalam proses pembuatan Skripsi.

Amin... Ya rabbal 'Alamin

Jakarta, Desember 2011

Penulis

Indah fauziah

DAFTAR ISI

Bab	Isi	
Halaman		
LEMBAR SAMPUL		i
LEMBAR JUDUL		ii
LEMBAR PENGESAHAN SKRIPSI		iii
LEMBAR PENGESAHAN UJIAN		iv
LEMBAR PERNYATAAN		v
ABSTRAK		vi
KATA PENGANTAR		vii
DAFTAR ISI		x
DAFTAR TABEL		xiv
DAFTAR GAMBAR		xv
DAFTAR LAMPIRAN		xviii
 BAB I	 PENDAHULUAN	 1
	1.1. Latar Belakang	1
	1.2. Perumusan Masalah	2
	1.3. Batasan Masalah	2
	1.4. Tujuan dan Manfaat Penelitian	3
	1.4.1. Tujuan Penelitian	3
	1.4.2. Manfaat Penelitian	3
	1.5. Metode Penelitian	4
	1.5.1. Metode Pengumpulan Data	4
	1.5.2. Metode Pengembangan sistem	5
	1.6. Sistematika Penulisan	7
 BAB II	 LANDASAN TEORI	 9
	2.1. Sistem Informasi	9

2.1.1 Konsep Dasar Sistem.....	9
2.2. Pengertian Data dan Informasi	11
2.3. Pengertian Sistem Informasi.....	12
2.4. Pengertian Perancangan Sistem	13
2.5. Aplikasi	14
2.6. Pelayanan	15
2.7. <i>Mobile Application</i>	15
2.8. Teknologi <i>Mobile</i>	16
2.9. <i>Internet Mobile Phone</i>	17
2.10. Pemograman Java2 <i>Micro Edition</i> (J2ME)	19
2.10.1. Pembahasan Umum Java	19
2.10.2. Pengertian Java2 <i>Micro Edition</i>	22
2.11. MySQL	31
2.12. XML (<i>Extensible Markup Language</i>)	34
2.12.1 Keketatan XML	35
2.13. PHP	36
2.14. <i>Internet</i>	39
2.14.1. TCP/IP	40
2.14.2. <i>Web Server</i>	41
2.15. <i>Unified Modelling Language</i> (UML)	42
2.15.1. Pengertian UML	42
2.15.2. UML Diagram	43
2.15.3. Notasi UML.....	45
2.16. Studi Sejenis	49
2.17. Simbol-Simbol UML	65
BAB III METODOLOGI PENELITIAN	70
3.1 Metode Pengumpulan Data	70
3.1.1. Studi Lapangan.	70
3.1.3. Studi Pustaka	71
3.2 Metode Pengembangan Sistem	72

3.2.1.	Mendengarkan pelanggan	73
3.2.2.	Membangun Memperbaiki prototype	74
3.2.2.1	Perancangan Sistem	74
3.2.2.2	Pengkodean Sistem	75
3.2.3.	Pengujian Prototype	76
BAB IV	PEMBAHASAN	77
4.1.	Gambaran Umum Perusahaan	77
4.1.1.	Sejarah.....	77
4.1.2.	Visi dan Misi PLN DisJaya	77
4.1.3.	Struktur Organisasi.....	78
4.2.	Pengembangan Sistem	79
4.2.1.	Mendengarkan Pelanggan	79
4.2.1.1.	Analisis sistem yang Berjalan	79
4.2.1.2.	Identifikasi masalah.....	79
4.2.1.3.	Usulan Penyelesaian Masalah	80
4.2.2.	Membangun Memperbaiki Prototype	80
4.2.2.1.	Desain Aplikasi	80
4.2.2.2.	Desain Basis Data	104
4.2.2.3.	Desain <i>Interface</i>	106
4.2.2.4.	Desain Alur Komunikasi Data	112
4.2.2.5.	Pengkodean Sistem	113
4.2.3.	Pengujian Prototype	115
4.2.3.1.	Spesifikasi Ponsel yang digunakan	115
4.2.3.2.	<i>Transfer</i> Aplikasi dari PC kedalam Ponsel	116
4.2.3.3.	Pengujian Aplikasi pada Ponsel Nokia E63	116
4.2.3.4.	Evaluasi Sistem	118

BAB V	PENUTUP	119
	5.1. Kesimpulan	119
	5.2. Saran	119
DAFTAR PUSTAKA		120
LAMPIRAN		123



DAFTAR TABEL

Tabel	Isi	Halaman
2.1	<i>Class Diagram</i>	65
2.2	<i>Componen Diagram</i>	66
2.3	<i>Deployment Diagram</i>	66
2.4	<i>Use Case Diagram</i>	67
2.5	<i>Sequence Diagram</i>	67
2.6	<i>Collaboration Diagram</i>	68
2.7	<i>Activity Diagram</i>	69
4.1	Penentuan Actor	81
4.2	Narasi <i>Use Case Register</i>	82
4.3	Narasi <i>Use Case Login</i>	83
4.4	Narasi <i>Use Case Profile</i>	84
4.5	Narasi <i>Use Case Info Rekening</i>	84
4.6	Narasi <i>Use Case Info Token Prabayar</i>	85
4.7	Narasi <i>Use Case Info Kantor PLN</i>	86
4.8	Narasi <i>Use Case Info Daftar Bank</i>	86
4.9	Narasi <i>Use Case Keluar</i>	87
4.10	Tabel <i>User</i>	104
4.11	Tabel Rekening	105
4.12	Tabel Token	105
4.13	Spesifikasi Ponsel yang digunakan	115
4.14	Hasil Pengujian <i>Black Box</i>	117

DAFTAR GAMBAR

Gambar	Isi	Halaman
2.1	Arsitektur J2ME	23
2.2	Bagian – Bagian Di Dalam <i>Platform</i> J2ME	24
2.3	Arsitektur J2ME pada CLDC.....	26
2.4	MIDP <i>User Interface</i>	28
2.5	Notasi <i>Actor</i>	45
2.6	Notasi <i>Class</i>	45
2.7	Notasi <i>Use Case</i>	46
2.8	Notasi <i>Interaction</i>	46
2.9	Notasi <i>Interface</i>	46
2.10	Notasi <i>Package</i>	47
2.11	Notasi <i>Note</i>	47
2.12	Notasi <i>Dependency</i>	48
2.13	Notasi <i>Association</i>	49
2.14	Notasi <i>Generalization</i>	49
2.15	Notasi <i>Realization</i>	49
3.1	Fase-Fase RAD (<i>Rapid Application Development</i>)	73
4.1	Struktur Organisasi PLN.....	78
4.2	Perancangan <i>Use Case</i>	81
4.3	<i>Activity Diagram Register</i>	88
4.4	<i>Activity Diagram Login</i>	89
4.5	<i>Activity Diagram Profile</i>	90
4.6	<i>Activity Diagram</i> Info Rekening	91
4.7	<i>Activity Diagram</i> Info Token Prabayar	92
4.8	<i>Activity Diagram</i> Info Kantor PLN	93
4.9	<i>Activity Diagram</i> Info Daftar Bank	94
4.10	<i>Activity Diagram</i> Keluar	95
4.11	<i>Sequence Diagram Register</i>	96

4.12	<i>Sequence Diagram Login</i>	97
4.13	<i>Sequence Diagram Profile</i>	98
4.14	<i>Sequence Diagram Info Rekening</i>	99
4.15	<i>Sequence Diagram Info Token Prabayar</i>	100
4.16	<i>Sequence Diagram Info Kantor PLN</i>	101
4.17	<i>Sequence Diagram Info Daftar Bank</i>	102
4.18	<i>Sequence Diagram Keluar</i>	103
4.19	<i>Class Diagram Aplikasi</i>	104
4.20	<i>Design User Interface Form Splash Screen</i>	106
4.21	<i>Design User Interface Form Login</i>	106
4.22	<i>Design User Interface Form Home</i>	107
4.23	<i>Design User Interface Form Menu</i>	107
4.24	<i>Design User Interface Form Register</i>	107
4.25	<i>Design User Interface Form Register Pasca Bayar</i>	108
4.26	<i>Design User Interface Form Register Prabayar</i>	108
4.27	<i>Design User Interface Form Profile Pasca Bayar</i>	108
4.28	<i>Design User Interface Form Profile Prabayar</i>	109
4.29	<i>Design User Interface Form Info Rekening</i>	109
4.30	<i>Design User Interface Form Info Token Prabayar</i>	109
4.31	<i>Design User Interface Form Info Kantor PLN</i>	110
4.32	<i>Design User Interface Form Info Kantor PLN Jakarta dan Tangerang</i>	110
4.33	<i>Design User Interface Form Info Alamat Kantor PLN Jakarta dan Tangerang</i>	111
4.34	<i>Design User Interface Form Info Daftar Bank</i>	111
4.35	<i>Alur Komunikasi Data</i>	112

DAFTAR LAMPIRAN

- Lampiran I. Hasil Tampilan Aplikasi Pada Emulator dan Ponsel Nokia E63
- Lampiran II. Proses Kontruksi Program
- Lampiran III. Proses Transfer Aplikasi Dari PC Kedalam Ponsel dan Instalasinya
- Lampiran IV. *Source Code*
- Lampiran V. Wawancara dan Kuesioner
- Lampiran VI. Surat Keterangan Penelitian

BAB I PENDAHULUAN

1.1 Latar Belakang

PT. PLN Persero Distribusi Jakarta Raya dan Tangerang adalah perusahaan milik Negara yang bergerak dibidang pendistribusian tenaga listrik khusus wilayah Jakarta dan Tangerang.

PT. PLN Persero Distribusi Jakarta Raya dan Tangerang mempunyai banyak pelanggan di wilayah DKI Jakarta dan sekitarnya, mulai dari kalangan ibu rumah tangga sampai kalangan VVIP. Banyak diantaranya yang membutuhkan informasi mengenai pelayanan PLN. Namun, PLN belum bisa melayani seluruh kebutuhan informasi yang dibutuhkan oleh pelanggan khususnya mengenai info rekening listrik. Untuk memenuhi kebutuhan pelanggan akan informasi mengenai tagihan rekening listrik, maka perlu dibentuk suatu aplikasi informasi khususnya mengenai info rekening listrik.

Aplikasi informasi rekening listrik PLN ini merupakan Aplikasi yang berisi informasi tagihan listrik, dan informasi tambahan yang bisa diketahui oleh pelanggan melalui *mobile device*, seperti *handphone*, *PDA phone*, *Smartphone*, sampai *pocket pc*. Aplikasi informasi rekening listrik PLN ini dapat membantu pelanggan dalam memperoleh informasi yang dibutuhkan secara *realtime* sehingga pelanggan akan lebih mudah mendapatkan informasi yang dicari secara rinci.

Sehubungan dengan hal diatas, penulis mengembangkan aplikasi ringan yang menyediakan informasi standar PLN, berbasis *platform Smartphone*. Dengan ini penulis merancang aplikasi dengan judul **“Prototype Aplikasi Info Rekening Listrik PLN Berbasis *Mobile* (Studi Kasus: PT. PLN Persero Distribusi Jakarta Raya dan Tangerang)”**.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas ada beberapa masalah yang penulis rumuskan :

1. Bagaimana mengembangkan sistem pelayanan informasi rekening listrik berbasis *mobile*?
2. Bagaimana merancang aplikasi informasi yang dapat memudahkan pelanggan untuk memperoleh informasi seputar rekening listrik?

1.3 Batasan Masalah

Mengoptimalkan pembahasan masalah maka akan dibatasi tulisan pada ruang lingkup:

1. Pengelolaan aplikasi ini hanya menampilkan simulasi informasi rekening listrik.
2. Untuk pengujian dan implementasi aplikasi ini, penulis menggunakan ponsel merk Nokia tipe E63.
3. Merancang aplikasi yang dapat membantu penyampaian informasi dengan menggunakan empat diagram yang terdapat pada UML.

4. Data pelanggan yang ditampilkan hanya berupa data terstruktur yang diberikan oleh pihak PLN.
5. Aplikasi tersebut diimplementasikan pada telepon selular yang mendukung teknologi java MIDP 2.0

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Tujuan penelitian tugas akhir ini adalah:

1. Mengembangkan sistem pelayanan informasi rekening listrik berbasis *mobile*.
2. Memberikan kemudahan kepada pelanggan PLN untuk mendapatkan informasi rekening listrik secara *realtime*.

1.4.2 Manfaat Penelitian

1. Bagi Penulis

- a. Menerapkan ilmu-ilmu dalam merancang sebuah sistem yang telah di peroleh selama kuliah.
- b. Menambah wawasan dalam perancangan prototype aplikasi info rekening listrik PLN.
- c. Melatih kemampuan dalam menganalisa, mengolah masalah sehingga dapat memberikan solusi tepat terhadap masalah yang ada pada lembaga/instansi, maupun pemerintah atau swasta.

2. Bagi Instansi

1. Sebagai nilai tambah instansi dalam pengembangan pelayanan pelanggan.

3. Bagi Universitas

1. Sebagai masukan untuk mengevaluasi sejauh mana universitas mencetak tenaga-tenaga terampil, kreatif, dinamis dan profesional untuk dunia kerja.
2. Mengetahui sejauh mana kemampuan mahasiswa dalam menguasai materi yang telah diberikan oleh universitas.
3. Agar universitas dapat lebih memberikan apa yang dibutuhkan oleh mahasiswa di dalam perkembangan teknologi yang sangat pesat seperti sekarang ini.

1.5 Metode Penelitian

Metode penelitian yang digunakan dalam penulisan skripsi ini adalah:

1.5.1 Metode Pengumpulan Data

1. Studi Lapangan

a. Observasi

Yaitu pengamatan yang dilakukan secara langsung untuk mendapatkan data-data yang akan diolah kedalam prototype aplikasi info rekening listrik PLN berbasis *mobile*.

b. Wawancara

Yaitu dengan melakukan pengumpulan data-data yang dimiliki instansi untuk mendapatkan informasi, dengan cara bertanya langsung kepada responden.

c. Kuesioner

Peneliti akan membagikan kuesioner kepada pelanggan untuk menggali kebutuhan pelanggan untuk diambil hasil-hasil pendapatnya sebagai bahan dalam analisis.

2. Studi Pustaka

Dalam hal ini teori-teori yang berhubungan dengan pengumpulan data dan penelitian sejenis yang pernah dikembangkan untuk sebagai alat banding. Penjelasan dari masing-masing data tersebut diambil dari buku-buku, studi sejenis maupun dari *internet*.

1.5.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan dalam penulisan skripsi ini adalah metode Prototype. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Model prototype ini mampu menawarkan pendekatan yang terbaik dalam hal kepastian terhadap efisiensi algoritma, kemampuan penyesuaian diri dari sebuah sistem operasi

atau bentuk-bentuk yang harus dilakukan oleh interaksi manusia dengan mesin (Pressman, 2002:40).

Prototyping adalah proses iterative dalam pengembangan sistem dimana *requirement* diubah ke dalam sistem yang bekerja (*working system*) yang secara terus menerus diperbaiki melalui kerjasama antara *user* dan analis.

Pengembangan dari perancangan sistem tersebut dalam pelaksanaannya penulis lakukan menggunakan tiga tahap siklus pengembangan model prototype yaitu (Pressman, 2002:42):

1. Mendengarkan pelanggan

Merupakan tahap pertama dalam merancang sebuah sistem. Pada tahap ini akan menentukan informasi-informasi yang dibutuhkan oleh pelanggan agar tercipta sebuah aplikasi sehingga mengarah pada tujuan dibuatnya aplikasi tersebut.

2. Membangun Memperbaiki Prototype

Dalam tahap ini dilakukan perancangan dan pengkodean untuk sistem yang diusulkan yang mana tahapnya meliputi: perancangan proses-proses yang akan terjadi didalam sistem, perancangan diagram UML yang akan digunakan, perancangan antar muka keluaran serta dilakukan tahap pengkodean terhadap rancangan-rancangan yang telah didefinisikan, kelengkapan *software* dan *hardware*.

3. Pengujian Prototype

Pada tahap ini akan dilakukan pengujian terhadap sistem yang telah disusun dan melakukan pengenalan terhadap sistem yang telah diujikan serta mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan

1.6 Sistematika Penulisan

Adapun sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini akan diuraikan latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat, metodologi penelitian, sistematika penulisan, daftar pustaka.

BAB II : LANDASAN TEORI

Dalam bab ini akan diuraikan teori, perancangan, aplikasi, *Mobile Application*, *Php* dan *Mysql*, *UML*, *J2ME* dan kajian-kajian yang diperlukan dalam penulisan penelitian ini.

BAB III : METODOLOGI PENELITIAN

Bab ini membahas metode yang digunakan penulis. Dalam melakukan pencarian data maupun pengembangan sistem yang dilakukan pada penelitian ini, studi *literature* sejenis, serta metode pengembangan sistem.

BAB IV : PEMBAHASAN

Bab ini membahas tentang analisis, gambaran umum perusahaan, desain sistem, pengujian, serta implementasi sistem yang dibangun, dengan mengacu kepada pengembangan sistem yaitu prototype.

BAB V : PENUTUP

Bab ini merupakan bab akhir yang memuat kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

2.1.1 Konsep Dasar Sistem

Menurut lucas (1993: 5) sistem merupakan suatu himpunan atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain terpadu.

Berbagai ahli telah mengajukan beberapa daftar sebagai dasar teori sistem umum diantaranya :

- a. Komponen-komponen sistem saling berhubungan dan tergantung satu sama lain
- b. Sebuah sistem harus dipandang sebagai satu keseluruhan
- c. Sistem adalah pengejar sasaran dalam beberapa cara. Komponen yang saling berinteraksi mencapai beberapa sasaran yaitu sebuah keseimbangan pencapaian tujuan.
- d. Sistem mempunyai input dan output
- e. Semua sistem mengubah input menjadi output
- f. Sistem menunjukkan adanya entropi. Entropi menunjukkan keadaan sistem tertutup (tidak ada input diluar sistem) dimana tiap elemen bergerak menuju ke keadaan tidak terorganisasi dan ketidakmampuan untuk mendapatkan serta mengolah input sehingga sistem tersebut tidak sanggup menghasilkan output.

- g. Sistem harus mempunyai suatu cara untuk mengatur komponen-komponen yang saling berinteraksi agar tujuan sistem tercapai
- h. Sistem biasanya terdiri dari subsistem yang lebih kecil
- i. Tiap unit khusus menangani tugas yang khusus
- j. Sistem biasanya menunjukkan hal yang sama yaitu beberapa keadaan akhir yang dapat dicapai dari berbagai jalur yang berlainan.

Menurut Jogiyanto (1999: 3) pada dasarnya sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu:

a. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi membentuk satu kesatuan.

b. Batas sistem

Suatu daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan lain.

c. Lingkungan luar sistem

Apapun diluar batas sistem yang mempengaruhi operasi sistem, baik yang menguntungkan maupun yang merugikan sistem.

d. Penghubung sistem

Suatu media penghubung yang memungkinkan suatu sumberdaya mengalir antara satu subsistem dengan subsistem lainnya.

e. Masukan sistem

Energi yang dimasukkan kedalam sistem berupa masukan perawatan dan masukan sinyal.

f. Keluaran sistem

Hasil dari energi yang diolah akan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah sistem

Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan lainnya menjadi keluaran berupa bahan jadi.

h. Sasaran dan tujuan

Sistem yang berguna pasti mempunyai tujuan dan sasaran yang sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan di sistem.

2.2 Pengertian Data dan Informasi

1. Data dan Informasi

Jogianto (2005: 9) Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan lain akan membuat sejumlah data kembali.

2. Data

Menurut Purwono (2002: 22) Data bisa disusun dalam suatu urutan (disebut *sort*). Dalam berbagai kriteria ukuran, meski hanya menggunakan tumpukan data yang sama (disebut *file*).

Data adalah kumpulan kejadian yang diangkat dari suatu kenyataan yang dapat berupa angka-angka, huruf-huruf, simbol-simbol khusus atau gabungan darinya (Jogiyanto, 2005:2).

3. Informasi

Menurut Witarto (2004: 9) informasi adalah rangkaian data yang mempunyai sifat sementara, tergantung dengan waktu, mampu member kejutan pada penerima. Intensitas dan lamanya kejutan dari informasi disebut nilai informasi. Informasi yang tidak mempunyai, biasanya karena rangkaian data yang tidak lengkap atau kadarluasa.

4. Hubungan Data dan Informasi

Hubungan data dan informasi model dan selanjutnya membentuk suatu siklus, John Burch berpendapat siklus ini dinamakan dengan siklus informasi.

2.3 Pengertian Sistem Informasi

Menurut Witarto (2004: 8) Suatu sistem yang dinyatakan sebagai sistem informasi adalah jika didalamnya terdapat informasi yang bermanfaat bagi sistem organisasi, atau jika datanya pernah ada dimasukkan, namun sudah kadarluasa. Sistem informasi yang baik adalah jika *user*-nya rajin

memasukkan dan memeriksa data dari waktu ke waktu sehingga pimpinan dalam sebuah organisasi mudah dalam mengakses informasi tentang kinerja sistem organisasi, melalui keberadaan sistem informasi serta didasarkan pada data yang akurat dan mutakhir.

Menurut Jogiyanto (2005:11) Sistem informasi diartikan sebagai suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem informasi didalam sebuah organisasi menampung dan mengolah data yang menghasilkan suatu informasi yang berguna, yang mendukung sebuah organisasi dan karyawannya, pelanggan, pemasok dan mitra kerja (Whitten, *et al*, 2004:12)

2.4 Pengertian Perancangan Sistem

Menurut whitten, *et al* (2004:23) Perancangan sistem informasi didefinisikan sebagai sebuah pekerjaan yang terfokus pada spesifikasi dari solusi basis komputer yang terinci.

Menurut Pressman (2002: 401), perancangan perangkat lunak adalah satu-satunya cara diman kita dapat secara akurat menterjemahkan kebutuhan pelanggan kedalam produk atau sistem perangkat lunak yang berfungsi sebagai dasar bagi semua rekayasa perangkat lunak dan kemudian diikuti oleh suatu pemeliharaan perangkat lunak.

Menurut Jogianto (2005: 211), tahap desain adalah perancangan komponen-komponen sistem informasi dengan tujuan untuk dikomunikasikan dengan *user* bukan pemograman.

Perancangan (*design*) adalah pendekatan yang digunakan dalam bidang rekayasa dan bidang-bidang lainnya yang digunakan untuk menspesifikasi bagaimana membuat atau melakukan sesuatu (Wahid, 2005:17).

Menurut Kamus Besar Bahasa Indonesia Perancangan (desain) biasa diterjemahkan sebagai seni terapan, arsitektur, dan berbagai pencapaian kreatif lainnya. Dalam sebuah kalimat, kata “desain” bisa digunakan baik sebagai kata benda maupun kata kerja. Sebagai kata kerja, “desain” memiliki arti “proses untuk membuat dan menciptakan objek baru”. Sebagai kata benda, “desain” digunakan untuk menyebut hasil akhir dari sebuah proses kreatif, baik itu terwujud sebuah rencana, proposal atau berbentuk objek nyata.

2.5 Aplikasi

Aplikasi (program) merupakan tugas khusus yang akan dijalankan oleh *enterprise* baik secara otomatis maupun semi otomatis (Prahasta, 2005:195).

Aplikasi adalah suatu program yang siap pakai. Program yang diciptakan untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi adalah program proses kata dan *web browser* (Erfan, 2009).

2.6 Pelayanan

Hasyim (2005: 44), layanan adalah kegiatan atau urutan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik, dan menyediakan kepuasan pelanggan.

Dalam kamus besar Bahasa Indonesia dijelaskan pelayanan sebagai usaha melayani kebutuhan orang lain. Sedangkan melayani adalah membantu menyiapkan (mengurus) apa yang diperlukan seseorang.

2.7 Mobile Application

Mobile application merupakan aplikasi yang cara aksesnya menggunakan perangkat bergerak (*mobile device*) seperti (*handphone*), *smartphone* dan *PDphone* (Suryana, 2008).

Aplikasi yang menggunakan perangkat bergerak semakin meningkat dan beragam saat ini seperti aplikasi *mobile* berbasis java berbasis *android* ataupun berbasis *Flash lite*.

1. Ponsel

Telepon seluler atau lebih dikenal dengan nama ponsel atau *handphone* saat ini menjadi salah satu alat komunikasi populer, karena perkembangan alat komunikasi yang satu ini belakangan berubah secara signifikan. Ini dibuktikan dengan kemampuan ponsel dijejali dengan berbagai macam *software* dan *hardware* multimedia seperti kamera, pemutar musik, kemampuan berselancar di *internet* dan lain-lain (Rudi, 2009).

Perkembangan *handphone* atau ponsel telah memasuki babak baru, yaitu babak multimedia. Kemunculan babak baru yang dipicu oleh luasnya perkembangan teknologi *internet* menjadikan fitur ponsel tidak hanya sekedar untuk sms atau sebagai telepon (Sugiarto, 2007).

2.8 Teknologi Mobile

Menurut Sharon, *mobile device* adalah ponsel, telepon seluler, komunikasi gengam yang terhubung dengan jaringan nirkabel, yang memungkinkan pengguna untuk membuat panggilan, mengirim pesan teks, dan menjalankan aplikasi.

Menurut Sharon perkembangan teknologi *mobile* terdapat beberapa generasi atau hanya disingkat (G), yaitu :

1. Generasi pertama (1G)

Generasi pertama dari ponsel dan bertipe analog (*analog devices*). menggunakan sinyal analog untuk mentransmisikan data. Memiliki keterbatasan dalam mentransmisikan datanya.

2. Generasi kedua (2G)

Merupakan pengembangan dari generasi pertama. pada generasi kedua ini, ponsel bertipe digital. Fitur yang dimiliki adalah *voice + SMS+ circuit switched data*. Pada generasi ini, menggunakan sistem standar GSM, iDEN, CDMA, TDMA.

3. Generasi 2.5 (2.5G)

Pengembangan dari teknologi sebelumnya. Penambahan fitur adalah GPRS, HSCSD, WiDEN. Selain itu, adapula penambahan berupa teknologi *transfer* data seperti EDGE, CDMA2000 1xRTT.

4. Generasi Ketiga (3G)

Merupakan generasi ketiga dari teknologi *mobile*. Memiliki fitur *Broadband data + voice + streaming video*. Pada generasi ini menggunakan sistem standar W-CDMA (UMTS, FOMA), 1xEVDO

5. Generasi keempat (4G)

Menggunakan akses data *broadband* kecepatan tinggi, dan *visual centric information*. Menrasmisikan data 100 Mbps ketika bergerak dan 1 Gbps ketika diam.

Menurut elron, janousek, joos (2009:3) perangkat *mobile* mengalami sebuah revolusi yang menarik hari ini karena ada peningkatan cepat dalam jaringan pelanggan serta bertambahnya desainer yang berpengalaman, yang hasil dalam antar muka pengguna yang lebih baik dan inovasi perangkat keras seperti layar sentuh. Seperti perubahan perangkat *mobile*, harapan konsumen perangkat *mobile* mereka meningkat juga.

2.9 Internet Mobile Phone

Internet bergerak atau *mobile internet* adalah hasil konvergensi dari teknologi *fixed internet* dan teknologi telepon bergerak (*mobile telephone*). Perkembangan teknologi *internet* bergerak berjalan seiring dengan

kemajuan teknologi *transfer* data diatas jaringan *wireless* (Setiawan, 2008 :67).

Internet bergerak pada saat ini masih pada tahap pengenalan. Perkembangan *mobile internet* pada dasarnya tergantung pada 3 faktor utama:

1. Ketersediaan aplikasi dan konten
2. Ketersediaan perangkat pendukung dengan harga terjangkau
3. Kemudahan penggunaan dan kemampuan koneksi

Elemen penting yang akan menjadi kelebihan *internet* bergerak dibandingkan dengan *internet* konvensional, terutama dilihat dari sisi aplikasi-aplikasinya adalah:

1. Personalisasi (*Personalization*)

Personalisasi dari aplikasi *wireless* dibangun agar ketika pengguna *login* ke *internet* maka *preference* pengguna akan teraktivasi, berdasarkan kebutuhan dari pengguna dan disesuaikan pula dengan lokasi pengguna.

2. Selalu *Online* (*Always On*)

Kondisi selalu *Online* adalah dampak dari penggunaan teknologi paket data (GPRS) yang menghilangkan prosedur *dial up* untuk akses *internet*.

3. Mobilitas pengguna (*Mobility*)

Pengguna akan dapat menikmati aplikasi akses *internet* bergerak yang mudah, yaitu dengan usaha sedikit memberi hasil yang besar.

Hal ini mendukung mobilitas pengguna saat menggunakan perangkat *mobile* nya.

2.10 Pemograman Java2 Micro Edition (J2ME)

2.10.1 Pembahasan Umum Java

Java adalah bahasa pemograman yang berorientasi objek, java dikembangkan dari bahasa C++ sehingga sintaksnya seperti bahasa C++ (Supardi, 2005:1). Java juga dikenal sebagai suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemograman, dan sekaligus suatu *platform*. Sebagai bahasa pemograman, java dikenal sebagai bahasa pemograman tingkat tinggi. Java mudah dipelajari, terutama lagi programmer yang telah mengenal C/C++.

Menurut Budi raharjo, et. al. (2007:1), java adalah bahasa pemograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekanya seperti Patrick Naughton, Chris warth, Ed Frank, dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama *Sun Microsystem* pada tahun 1991. Bahasa pemograman ini pada awalnya diinisialisasi dengan nama Oak, tetapi karena Oak sendiri merupakan nama dari bahasa pemograman *computer* yang sudah ada, maka sun menggubahnya menjadi java.

Sun kemudian meluncurkan *browser* dari java yang disebut *Hot java* yang mampu menjalankan *Applet*. Setelah itu teknologi

java diadopsi oleh Netscape yang memungkinkan program java dijalankan di *browser* Netscape yang kemudian diikuti *Internet Explorer* dan perusahaan *Microsoft*. Karena keunikan dan kelebihanannya, teknologi java mulai menarik banyak *vendor* lain seperti IBM, Oracle, Symantec, Inprise (dahulu Borland Inc.) dan perusahaan-perusahaan *mobile* seperti Nokia, Siemens, SonyEricsson, Motorola dan Samsung juga mengadopsi teknologi java. Sun merilis versi awal java secara resmi pada tahun 1996 yang kemudian terus berkembang hingga muncul JDK (*Java Development Kit*) 1.1, kemudian JDK1.2 yang mulai disebut sebagai versi java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utamanya adalah adanya *swing* yang merupakan teknologi GUI (*Graphical User Interface*) yang mampu menghasilkan *window* yang *portable*.

Alasan utama pembentukan bahasa java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan berbagai macam perangkat *electronic*, seperti *Microwave oven* dan *remote control*, sehingga java harus bersifat *portable* atau yang sering disebut dengan *platform independent* (tidak tergantung pada *platform*). Itulah yang menyebabkan dalam dunia pemrograman java, dikenal adanya istilah '*write once, run everywhere*', yang berarti kode program yang ditulis sekali, namun dapat dijalankan dibawah

platform manapun, tanpa harus melakukan perubahan kode program (Budi raharjo, et. al, 2007:2).

Menurut M.Salahudin dan Rosa A.S (2006:1), java2 adalah generasi kedua dari java *platform* (generasi awalnya adalah *Java Development Kit*). Java berdiri diatas 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. *Platform* java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin java, dan sebuah *compiler*, *debugger*, dan kelas lain yang dipaket dalam *Java Development Kit* (JDK). Agar sebuah program java dapat dijalankan, maka *file* dengan ekstensi “.java” harus dikompilasi menjadi *file bytecode*. Untuk menjalankan *ByteCode* tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program java, hanya menjalankan, tidak untuk membuat kode baru lagi dan JRE berisi JVM dan *Library* java yang digunakan.

API adalah *Application Programming Interface* yaitu sebuah *layer* yang berisi *class-class* yang sudah didefinisikan dan antarmuka pemograman yang akan membantu para pengembang aplikasi dalam perancangan sebuah aplikasi. API memampukan para pengembang untuk dapat mengakses fungsi-fungsi sistem

operasi yang diijinkan melalui bahasa java. Pada saat ini dikenal ada tiga buah API dari java, yaitu (Andi, 2010:3):

Sun Microsystems telah mendefinisikan tiga buah edisi dari java2, yaitu sebagai berikut:

1. *Java2 Standard Edition* (J2SE), yang digunakan untuk mengembangkan aplikasi-aplikasi *desktop* dan *applet* (aplikasi java yang dijalankan di dalam *browser web*).
2. *Java2 Enterprise Edition* (J2EE), merupakan *super set* dari J2SE yang memperbolehkan kita untuk mengembangkan aplikasi-aplikasi berskala besar (*enterprise*), yaitu dengan melakukan pembuatan aplikasi-aplikasi di sisi *server* dengan menggunakan EJBs (*Enterprise JavaBeans*), aplikasi web dengan menggunakan *servlet* dan JSP (*Java Server Pages*) dan teknologi seperti CORBA (*Common Object Request Broken Architecture*) dan XML (*Extensible Markup language*).
3. *Java2 Micro Edition* (J2ME), merupakan *subset* dari J2SE yang digunakan untuk menangani pemograman didalam perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh.

2.10.2 Pengertian *Java2 Micro Edition*(J2ME)

Java2 Micro Edition atau biasa disebut *J2ME* adalah lingkungan pengembangan yang di desain untuk meletakkan

perangkat lunak *Java* pada barang elektronik beserta perangkat pendukungnya (Salahudin & Rosa, 2006:5).

Menurut Budi Raharjo, et.al. (2007, 3), J2ME merupakan sebuah kombinasi yang terbentuk antara sekumpulan *interface java* yang sering disebut dengan *java API (Application programming Interface)* dengan JVM (*Java Virtual Machine*) yang di desain khusus untuk alat dengan ruang yang terbatas. Kombinasi tersebut kemudian digunakan untuk melakukan pembuatan aplikasi-aplikasi yang dapat berjalan diatas alat (dalam hal ini *mobile devices*).

J2ME membawa *java* ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop* yang biasanya lebih kecil dibandingkan perangkat komputer *desktop*. J2ME biasa digunakan pada telepon selular, *pager*, *personal digital assistants* (PDA's) dan sejenisnya.

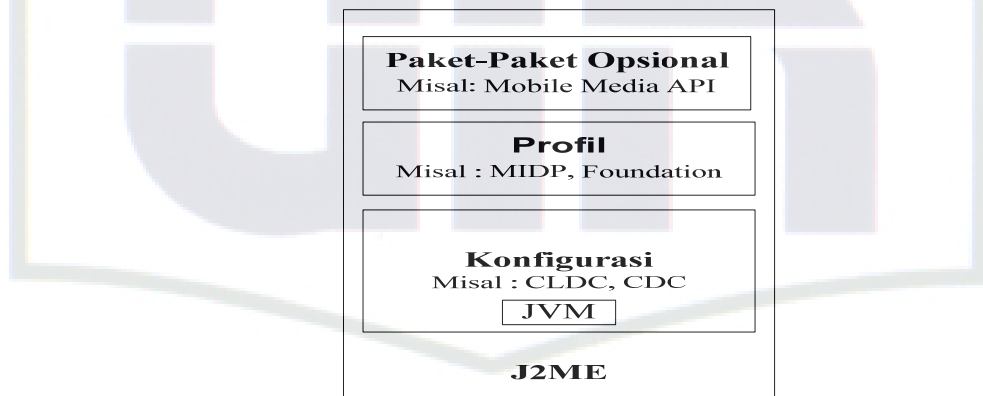
M. Shalahuddin dan Rosa A.S (2006:5), J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar berikut:

Profil	
Configuration	Kumpulan Library
	JVM
Sistem Operasi	

Gambar 2.1 Arsitektur J2ME

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada *handphone*. J2ME sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi *merk handphone*, maupun kemampuan *handphone* dan dukungannya terhadap J2ME.

J2ME sendiri pada dasarnya terdiri dari tiga buah bagian, yaitu konfigurasi, *profile*, dan paket-paket opsional, seperti yang ditunjukkan oleh gambar berikut (Budi raharjo, et. al, 2007:4).



Gambar 2.2 Bagian-bagian Di dalam *Platform* J2ME

1. Konfigurasi

Konfigurasi merupakan bagian yang berisi JVM dan beberapa *Library* kelas lainnya, dan yang dimaksud dengan JVM disini bukanlah JVM tradisional seperti terdapat pada J2SE, melainkan JVM yang sudah didesain secara khusus untuk alat.

Terdapat dua buah konfigurasi yang disediakan oleh sun *Microsystems*, yaitu CLDC (*Connected Limited Device Configuration*) dan CDC (*Connected Device Configuration*).

Target alat dari konfigurasi CLDC adalah alat-alat kecil. Sedangkan CDC merupakan *superset* dari CLDC sehingga semua kelas yang didefinisikan di dalam CLDC akan ada juga di dalam CDC.

2. Profil

Profil merupakan bagian perluasan dari konfigurasi. Artinya, selain sekumpulan kelas yang terdapat konfigurasi, terdapat juga kelas-kelas spesifik yang didefinisikan lagi dalam profil. Dengan kata lain, profil akan membantu secara fungsional yaitu dengan menyediakan kelas-kelas yang tidak terdapat di level konfigurasi.

Adapun profil yang sangat populer penggunaannya adalah profil yang disediakan oleh Sun Microsystems, yaitu yang dinamakan dengan MIDP (*Mobile Information Device Profil*).

3. Paket-Paket opsional

Paket-paket opsional merupakan paket-paket tambahan yang dibutuhkan oleh aplikasi sehingga pada saat proses *deployment* paket-paket tersebut perlu didistribusikan juga sebagai bagian dari aplikasi bersangkutan.

4. *Connected Limited Device Configuration (CLDC)*

Connected limited Device Configuration (CLDC) adalah sebuah konfigurasi yang terdapat di dalam J2ME untuk alat-alat yang memiliki keterbatasan ruang memori atau RAM (kurang dari 512 KB) dan pada umumnya dioperasikan dengan menggunakan baterai, serta memiliki *bandwith* yang kecil (Raharjo, et.al, 2007:6).

Connected Limited device Configuration (CLDC) adalah perangkat besar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon seluler, pager, dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*, kelas dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, dan JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*). Berikut ini merupakan gambar arsitektur J2ME yang menunjukkan posisi CLDC (Shalahuddin & Rosa, 2006:8).

MIDP	
Configuration	Kumpulan Library
	KVM
Sistem Operasi	

Gambar 2.3 Arsitektur J2ME pada CLDC

5. *Connected Device Configuration (CDC)*

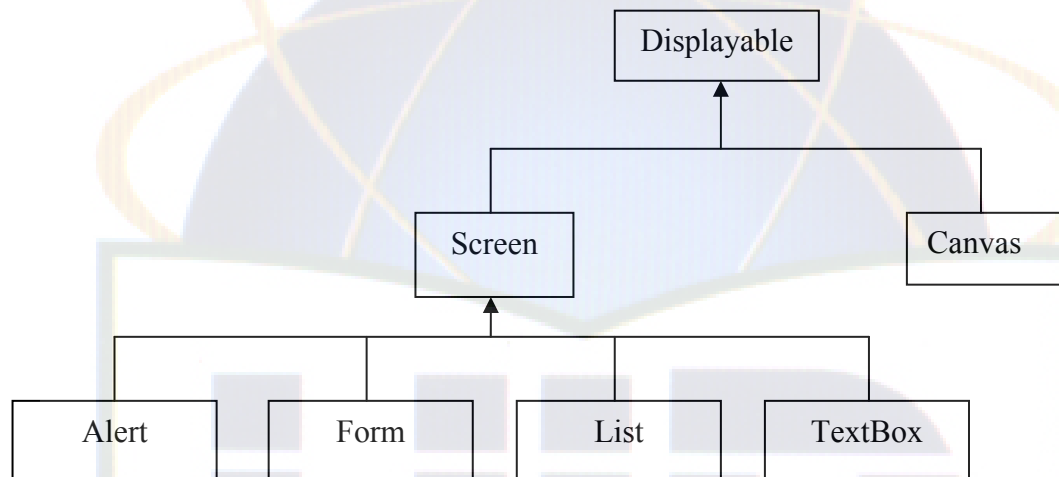
Connected Device Configuration (CDC) adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada java yang memiliki standarisasi. CDC terdiri dari *Virtual Machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform* (Shalahuddin & Rosa, 2006:8).

6. *Mobile Information Device Profile (MIDP)*

Mobile Information Device Profile (MIDP) adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan dan penyimpanan persisten (Shalahuddin & Rosa, 2006: 9).

MIDP *user interface* API memiliki API level tinggi dan level rendah. API level rendah berbasiskan penggunaan dari kelas abstrak *Canvas*, sedangkan kelas API level tinggi antara lain *Alert*, *Form*, *List*, dan *TextBox* yang merupakan ekstensi dari kelas abstrak *screen*. API level rendah lebih memberikan kemudahan kepada pengembang untuk memodifikasi sesuai dengan kehendaknya, sedangkan API level tinggi biasanya hanya

memberikan pengaksesan yang terbatas. Berikut ini merupakan gambar dari arsitektur antar muka MIDP.



Gambar 2.4 MIDP User Interface

7. MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan sub kelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk (Shalahuddin & Rosa, 2006: 11).

Menurut Budi Raharjo, et. al. (20007:8), MIDlet adalah sebutan untuk aplikasi-aplikasi yang dibuat didalam *handphone* dengan menggunakan profil MIDP (*Mobile Information Device Profile*).

Menurut Ivan Michael siregar (2007:21), Midlet adalah aplikasi yang dibangun pada *class* Midlet. *Application Manager* berkomunikasi dengan MIDlet melalui *method* yang ada pada *class* tersebut. Komunikasi tersebut dua arah, sehingga *Application manager* bisa melakukan *pause* sebuah MIDlet (contohnya pada saat *user* diijinkan untuk menerima panggilan telepon) kemudian dijalankan kembali.

Terdapat tiga buah *method* yang harus diimplementasikan oleh setiap MIDlet. Dengan kata lain, setiap MIDlet yang kita buat harus memiliki ketiga buah *method*. Adapun *method-method* tersebut sebagai berikut (Raharjo, et.al, 2007:23).

1. *startApp()*
2. *pauseApp()*
3. *destroyApp()*

Setiap Midlet dapat berada dalam salah satu keadaan (*state*) berikut: *pause*, *active* maupun *destroy*.

Pada saat pembuatan MIDlet baru, mula-mula MIDlet akan berada dalam keadaan *paused*. Apabila proses pembuatan MIDlet gagal akan mengakibatkan kesalahan (menimbulkan eksepsi), maka MIDlet akan langsung berada dalam keadaan *destroy*. Namun apabila proses pembuatan MIDlet berjalan dengan baik, maka setelah MIDlet dijalankan, maka AMS (*Application Management Software*) secara otomatis akan mengeksekusi

method starApp() dan hal ini akan mengubah MIDlet untuk berada dalam keadaan *active*. MIDlet yang berada dalam keadaan *active* dapat di ubah kembali menjadi keadaan *paused* melalui pemanggilan *method pauseApp()* atau diubah menjadi keadaan *destroyed* melalui pemanggilan *destroyApp()*. Sebagai contoh, pada saat Midlet dijalankan dan kemudian dihentikan oleh *user*, maka MIDlet akan mengalami perubahan keadaan, yaitu dari *active* menjadi *destroyed*.

Dalam implementasinya, MIDlet Memiliki struktur direktori sebagai berikut :

a. src

Menyimpan *Souce code* untuk MIDlet dan kelas lain yang diperlukan.

b. res

Menyimpan Sumber daya yang dibutuhkan oleh MIDlet, seperti misalnya gambar *icon*.

c. lib

Menyimpan file *.JAR atau ZIP yang berisi *library* tambahan yang dibutuhkan oleh MIDlet

d. bin

Menyimpan file *.JAR, *.JAD, dan file *manifest* yang berisi muatan komponen MIDlet.

8. J2ME Wireless Toolkit 2.2

J2ME Wireless Toolkit adalah sekumpulan *tool* yang digunakan untuk mengembangkan aplikasi-aplikasi dalam *handphone* dan *wireless device* lainnya (Raharjo, et.al, 2007:9).

M.Shalahuddin dan Rosa A.S (2006:15), *J2ME wireless toolkit* adalah kakas yang menyediakan lingkungan emulator, dokumentasi beserta contoh-contoh aplikasi java untuk perangkat kecil (*small device*). *J2ME WTK* berbasiskan pada CLDC dan MIDP. *J2ME WTK* adalah program yang meniru kerja *handphone* yang mendukung MIDP atau yang biasa disebut emulator. Oleh karena itu, belum tentu MIDlet yang berjalan di emulator juga berjalan pada *handphone* yang sebenarnya, karena bergantung pada kemampuan dan kapasitas *handphone* yang digunakan.

2.11 MySQL

Menurut Sitinduan (2003, 45) mengemukakan bahwa “MySQL adalah sistem manajemen *database* relasi (*relation database manajemen system*) yang bersifat terbuka (*open source*)”.

MySQL didistribusikan secara gratis dibawah lisensi GPL (*General Public License*), dimana setiap orang bebas menggunakan MySQL, namun boleh dijadikan produk turunan bersifat *closed source* atau komersial.

MySQL merupakan *server database* yang kecil dan mudah digunakan namun memiliki kehandalan dan performa yang tinggi.

1. Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Open Source

MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance Tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Jenis Kolom

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan *clien* menggunakan protokol TCP/IP, Unix soket (UNIX), atau *Named Pipes (NT)*.

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, Bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *API (Application Programming Interface)*.

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur Tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

Adapun keunggulan dari MySQL adalah:

1. Mampu menangani jutaan *user* dalam waktu yang bersamaan.
2. Mampu menampung lebih dari 50.000.000 *record*
3. Sangat cepat mengeksekusi perintah
4. MySQL menggunakan SQL dan bersifat *free* (gratis).
5. MySQL dapat berjalan di berbagai *platform*, antara lain Linux, Windows, dan lain sebagainya
6. Memiliki *user privilege system* yang mudah dan efisien.

2.12 XML (*Extensible Markup Language*)

Extensible Markup Language (XML) adalah sebuah tipe bahasa baru yang dikembangkan untuk *web* yang berbeda dengan tipe bahasa *scripting* atau pemrograman lain yang terdapat sebelumnya. XML tidak dikonsentrasikan untuk pemrosesan dan penampilan data tetapi terutama

dimaksudkan untuk memberitahu komputer apa arti sesungguhnya dari data yang disertakan. XML digunakan untuk mendefinisikan apa arti dari data dan bukan bagaimana data ditampilkan maka XML dengan mudah menggunakan data yang sama pada *platform* yang berbeda (Husni, 2007:97).

Manfaat *real* dari XML adalah untuk mendeskripsikan data. XML digunakan dengan cara menyerupai HTML, tetapi fakta menunjukkan perbedaan besar antara keduanya yaitu: (Husni, 2007:98)

1. HTML digunakan untuk mendeskripsikan data yang diformat.
2. XML digunakan untuk mendeskripsikan apa arti sebenarnya dari data.

2.12.1. Keketatan XML

Point lain yang diambil dari keketatan XML adalah keketatan pada saat penulisan kode. Ide keseluruhan dari XML adalah bahwa XML bersifat *platform independent* pada tempatnya berjalan. Kode yang sama dapat berjalan pada PC, Mac dan *mobile phone* bahkan pada *toaster*. Karena XML sesungguhnya tidak melakukan apapun (XML hanya sebuah bahasa untuk mendefinisikan data), XML diserahkan kepada *software developer* untuk membuat *software* yang menggunakan data ini pada *platform* tertentu. ini berarti bahwa semua kode XML distrukturkan dalam cara yang sama, sehingga *software* dapat dengan mudah dikembangkan (Husni, 2007:100).

2.13 PHP

1. Sejarah PHP

Menurut Dodit (2008, 17) mengemukakan bahwa, “PHP merupakan kependekan dari kata *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *general purpose licences (GPL)*”.

PHP merupakan salah satu bahasa pemrograman *server side scripting*. Maksud dari *server side scripting* adalah perintah (*script*) hanya diproses di *server web* dan hasilnya dikembalikan dalam bentuk *tag-tag HTML*. Oleh karena itu *script* program yang ditulis tidak akan tampil bila akan di lihat melalui menu *source code*.

Pemrograman PHP dapat berdiri sendiri ataupun disisipkan di antara kode html sehingga dapat ditampilkan bersama dengan kode-kode html tersebut dengan syarat *web server* harus *support* dengan php. Dalam pemakaian pemrograman PHP ditambahkan dengan mengapit program tersebut di antara tanda `<? dan ?>` atau `<?PHP dan ?>`. *File* html yang telah dibubuhi pemrograman php harus diganti ekstensi-nya menjadi *.php* sesuai dengan konfigurasi pada *web server*.

Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan sistem *database* di dalam *web*. Sistem *database* yang dapat didukung oleh PHP antara lain : MySQL, Oracle, Sybase, PostgreSQL dan SQL

lite. PHP dapat berjalan di berbagai sistem operasi seperti windows 98/NT, UNIX/LINUX, Solaris maupun Macintosh.

Dalam pembuatan *website* yang dinamis, PHP berfungsi menampilkan secara mandiri, diperlukan beberapa perangkat lunak tambahan yang harus di-*install*.

PHP tidak bisa bekerja dan digunakan secara mandiri, diperlukan beberapa perangkat lunak tambahan yang harus di *install* antara lain :

a. Apache-tomcat-6.0.16

Merupakan *web server* yang digunakan oleh PHP, berfungsi menampilkan hasil proses *script* PHP ke komputer *browser* dalam bentuk tag PHP.

b. PHP MyAdmin

Php MyAdmin adalah kakas untuk pengelolaan *database* yang berbasis *web*. PHP MyAdmin bukan merupakan suatu keharusan manipulasi data bisa juga digantikan dengan kakas lain, misalnya *MySQL Server 2000*. Namun dengan PHPMYAdmin, pengelolaan atau manipulasi *database* menjadi lebih mudah.

Php MyAdmin juga merupakan suatu alat bantu *open source* yang ditulis dalam PHP yang digunakan untuk menangani administrasi basis data *Mysql* yang diakses melalui *web browser* (*internet explorer, firefox* dan *opera*, dll). Fasilitas yang tersedia saat ini dapat membuat dan menghapus *database*, membuat,

menghapus dan menambah tabel, menghapus, mengedit dan menambah *field*, melakukan berbagai macam perintah SQL, mengatur kunci pada *field*, mengatur akses (*privileges*), mengekspor data ke berbagai format dan tersedia dalam 50 bahasa.

PHP merupakan kependekan dari kata *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *general porpose licences (GPL)*.

Pemrograman PHP sangat cocok dikembangkan dalam lingkungan *web*, karena PHP bisa diletakkan pada *script* HTML atau sebaliknya. PHP dikhususkan untuk pengembangan *web* dinamis. Maksudnya, PHP mampu menghasilkan *website* yang secara terus-menerus hasilnya bisa berubah-ubah sesuai dengan pola yang diberikan.

PHP tergolong juga sebagai bahasa pemrograman yang berbasis *server (server side scripting)*. Ini berarti bahwa semua *script* PHP diletakkan di *server* dan di terjemahkan oleh *web server* terlebih dahulu, kemudian hasil terjemahan itu dikirim ke *browser client*. Tentu hal tersebut berbeda dengan *JavaScript*. Kode program *JavaScript* harus di-*download* terlebih dahulu di komputer *client*, selanjutnya diterjemahkan oleh *browser internet*. Oleh karena itu, kode program *JavaScript* selalu tampak di

halaman *web* bersangkutan, jika dilakukan penyimpanan terhadap *file web*.

Secara teknologi, bahasa pemrograman PHP memiliki kesamaan dengan bahasa *ASP (Active Server Page)*, Cold Fusion, *JSP (Java Server Page)*, ataupun Perl.

2.14 Internet

Internet adalah jaringan yang kompleks dan cepat meluas bagi komputer yang saling terjalin. Kesepakatan mengenai seperangkat *protocol* bersama memungkinkan komputer manapun dalam *network* (jaringan) untuk berkomunikasi dengan komputer lain, dalam jaringan dengan menggunakan berbagai jenis cara yang berbeda. secara skematik, *Local Area Network* (LAN) untuk organisasi atau area geografis tertentu bergabung bersama dalam jaringan semakin besar untuk memberikan cakupan lebih global. Hal ini memungkinkan tugas bersama bisa dilakukan di seluruh jaringan, misalnya: mengirim/menerima suatu *email*, *surfing web* untuk mencari informasi (*teks* atau *graphics*) yang terdapat dalam *database*. *Internet* memungkinkan individu atau organisasi terhubung ke *web server* yang menampung sumber-sumber informasi dan membuat sumber-sumber ini tersedia bagi pengguna yang meminta.

Internet mampu mendukung berbagai perangkat digital selain dari komputer. Akibatnya, hal itu juga mengakibatkan struktur sentral bagi *distributed education* yang memungkinkan konektifitas *internet* untuk

videoconverencing, untuk percakapan telepon *Voice Over-Internet* (VOIP), dan berbagai aplikasi perangkat lunak seperti GPS (sistem posisi geografis) atau pengalaman *Virtual Reality* (Mulyanto, 2009:113).

2.14.1 TCP / IP

Pada dasarnya, komputer-komputer yang sedang terhubung dalam jaringan merupakan computer yang beragam, baik jenis dari pabriknya maupun sistem operasi yang menjalankannya. Untuk itu diperlukan sebuah bahasa standar yang dapat digunakan komputer-komputer tersebut untuk saling berkomunikasi. Bahasa tersebut dapat berupa sebuah *protocol* TCP/IP (*Transmission, Control Protocol/Internet Protocol*) yang digunakan untuk berkomunikasi. TCP/IP dapat terdiri dari lima lapisan (*layer*) yang dapat dihubungkan dengan arsitektur OSI (*Open System Interconnection*).

TCP merupakan *protocol* yang berada pada layer *transport* yang menyediakan layanan halaman terhadap komputer-komputer yang terhubung. Layanan yang pertama dikenal dengan *conection oriented*, yaitu melakukan pembentukan sebuah koneksi antar dua aplikasi yang akan berkomunikasi menggunakan TCP. Layanan kedua disebut dengan *reliable*, yaitu mendeteksi ada atau tidaknya kesalahan *packet* dan transmisi data. Sedangkan layanan yang terakhir adalah *byte stream service*, yaitu memastikan bahwa *packet* yang dikirim telah sampai kepada komputer tujuan.

IP akan membawa data yang berasal dari data *layer* dan kemudian membungkus pengiriman data dalam *packet* dengan alamat pengirim dan tujuannya.

Komputer-komputer tersebut harus memiliki identitas agar IP dapat mengenali komputer tersebut. Identitas ini dikenal dengan IP *address* atau alamat IP yang berupa empat bagian. Masing-masing bagian terdiri dari 8 *byte*, artinya nilainya berupa angka 0–255. Bagian-bagian tersebut dipisahkan dengan tanda titik. Contohnya, 172.16.32.198 (Mulyanto, 2009, 114-115).

2.14.2 Web Server

Secara umum *web server* terbagi menjadi dua tipe yaitu *web server* secara *offline* atau *local* dan *web server* secara *online* atau *web server* yang terkoneksi dengan *internet*. *Web server* ini dapat juga diartikan sebagai rumahnya *website* atau *situs*. Setiap *situs* yang ada di *internet* pastilah berjalan diatas *web server*. Jadi peranan *web server* ini sangatlah penting dalam pembuatan *situs*.

Sedangkan secara teori *web server* adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dengan mengirim kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya dalam bentuk halaman HTML (Kurniawan, 2009, 2).

2.15 Unified Modelling Language (UML)

2.15.1 Pengertian UML

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasi, dan membangun sistem perangkat lunak. UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artifak-artifak dari sistem. UML menggunakan notasi grafis untuk menyatakan suatu desain. Pemodelan dengan UML berarti menggambarkan yang ada dalam dunia nyata ke dalam bentuk yang dapat lebih dipahami dengan menggunakan notasi standar UML (Hariyanto, 2004, 259).

Menurut (Munawar, 2005:17) UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemrograman visual yang memungkinkan bagi pengembang sistem untuk membuat cetak baru atas visi mereka dalam bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan.

2.15.2 UML *Diagram*

UML menyediakan 9 jenis *diagram* yang akan dikelompokkan berdasarkan sifatnya, statis atau dinamis. Ke 9 jenis *diagram* tersebut UML adalah : (Hermawan, 2004, 23).

1. *Diagram* Kelas

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. *Diagram* ini umum dijumpai pada pemodelan sistem.

2. *Diagram* Object

Diagram ini memperlihatkan objek-objek serta relasi-relasi antar objek. *Diagram* objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai dari *diagram* kelas.

3. *Use-Case Diagram*

Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). *Diagram* ini terutama sangat penting untuk organisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. *Sequence Diagram*

Diagram sequence (*diagram* urutan) adalah *diagram* interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

5. *Collaboration Diagram*

Diagram kolaborasi adalah *diagram* interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

6. *Statechart Diagram*

Diagram state ini memperlihatkan *state-state* pada sistem, memuat *state*, transisi, *event*, serta aktivitas. *Diagram* ini penting terutama untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

7. *Activity Diagram*

Diagram aktifitas ini adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktifitas lainnya dalam suatu sistem. *Diagram* ini penting terutama dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagram*

Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak komponen-komponen yang telah ada sebelumnya. *Diagram* ini berhubungan dengan *diagram* kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka (*interface*), serta kolaborasi-kolaborasi.

9. *Deployment Diagram*

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. *Diagram* ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada di dalamnya. *Deployment diagram* berhubungan erat dengan *diagram* komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen.

2.15.3 Notasi UML

Berikut ini adalah notasi-notasi yang ada pada UML: (Hermawan, 2004:14).

1. Aktor (*Actor*)

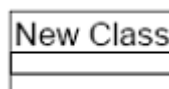
Actor adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan.



Gambar 2.5 Notasi *Actor*

2. Kelas (*class*)

Kelas seperti juga objek adalah sesuatu yang membungkus informasi dan perilaku dalam dirinya.



Gambar 2.6 Notasi *Class*

3. *Use Case*

Use case adalah peringkat tertinggi dari fungsionalitas yang dimiliki sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang akan menggunakan/memanfaatkan sistem. *Use case* menjelaskan suatu urutan kegiatan yang dilakukan oleh *actor* dan sistem untuk mencapai suatu tujuan tertentu walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.



Gambar 2.7 Notasi *Use case*

4. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek.



Gambar 2.8 Notasi *Interaction*

5. *Interface*

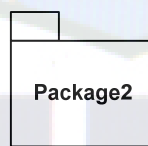
Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan dalam operasi dalam *class*.



Gambar 2.9 Notasi *Interface*

6. *Package*

Package adalah *container* atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari suatu sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari suatu model yang sedang dibangun.



Gambar 2.10 Notasi *Package*

7. *Note*

Note dibangun untuk membangun keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam *model*. *Note* ini bias ditempelkan kesemua elemen notasi yang lain.



Gambar 2.11 Notasi *Note*

8. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada suatu elemen memberikan pengaruh pada elemen lain. Elemen yang ada dibagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa ada tanda panah.



Gambar 2.12 Notasi *Dependency*

9. *Association*

Association menggambarkan navigasi antar *class* (*navigation*), berapa banyak objek lain yang bisa berhubungan dengan suatu objek (*multiplicity* antar *class*), dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).



Gambar 2.13 Notasi *Association*

10. *Generalization*

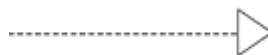
Generalization menunjukkan hubungan antar elemen yang lebih umum ke elemen yang lebih spesifik (*sub class*).



Gambar 2.14 Notasi *Generalization*

11. *Realization*

Realization menunjukkan hubungan bahwa elemen yang ada dibagian tanpa panah akan merelisasikan apa yang dinyatakan oleh elemen yang ada dibagian depan panah.



Gambar 2.15 Notasi *Realization*

2.16 Studi Sejenis

Studi literatur atau kajian literatur merupakan salah satu kegiatan penelitian yang mencakup: memilih teori-teori hasil penelitian, mengidentifikasi literatur, dan menganalisa dokumen, serta menerapkan hasil analisa sebagai landasan teori bagi penyelesaian masalah dalam penelitian yang dilakukan. Teori yang tepat disini adalah teori-teori yang bersesuaian dengan ruang lingkup masalah (Subana dan Sudrajat, 2005).

Pada studi literatur sejenis ini penulis mencari informasi melalui buku-buku, *internet*, makalah dan hasil penelitian dan literatur - literatur yang berhubungan dengan aplikasi yang akan dikembangkan, yaitu prototype aplikasi info rekening listrik PLN berbasis *mobile*.

Dalam hal ini proyek-proyek yang berhubungan dan atau penelitian sejenis yang pernah dikembangkan untuk sebagai alat banding dalam penelitian yang akan dilakukan penulis. Berikut ini adalah studi literatur sejenis yang berhubungan dengan penelitian skripsi ini yaitu :

Prototype Aplikasi *Commercial Traffic information* berbasis *Web* yang disusun oleh Muhammad Nazaruddin 2008 UIN Syarif Hidayatullah Jakarta. Penelitian ini merancang aplikasi sistem informasi yang dapat merepresentasikan data penayangan iklan secara *on-line* berbasis WAP untuk diimplementasikan pada stasiun televisi TRANS TV. Tujuan aplikasi ini untuk klien yaitu agar lebih mudah mengakses dan memonitori secara langsung penayangan iklan secara *realtime*, lebih efisien dalam melihat informasi penayangan iklan, sehingga menghemat waktu dan tenaga dan

mengoptimalkan sumber daya ada. Penelitian ini menggunakan metode SDLC (*System Development Life Cycle*) dengan model *waterfall*. Kelebihan dari aplikasi ini dapat memudahkan *customer* dalam melakukan pencarian informasi *log tape* penayangan iklan secara mudah dan cepat serta dapat menciptakan kondisi kerja yang efisien. Kekurangan aplikasi ini, aplikasi belum diimplementasikan secara *public*, serta kurangnya sistem keamanan data-data *user*, maupun data-data *server*.

Pembuatan Prototype *E-Banking via Internet* dengan Notifikasi SMS, yang disusun oleh Hendry Yanuard 2004 Universitas Diponogoro. *E-Banking via internet* adalah suatu sistem dimana *user* dapat mendapatkan semua *service* yang disediakan suatu bank melalui *internet*. Tujuan aplikasi ini adalah merancang suatu aplikasi *E-Banking* dengan berbagai fasilitas yang memberikan kemudahan sekaligus keamanan bagi para nasabah. Kelebihan aplikasi ini yaitu prototype *e-banking* ini dapat menangani transaksi-transaksi dasar banking, yaitu *transfer* rekening, pembayaran tagihan secara *e-lektronik* dan pembelian secara elektronik. Kekurangannya adalah fasilitas yang disediakan dalam sistem masih sangat minimal dan masih banyak pengembangan.

Prototype Aplikasi *M-Ticketing* pada PT Kereta Api Indonesia dengan Menggunakan Teknologi Java 2 *Micro Edition* (J2ME) yang disusun oleh Febiyanto Januar 2005 Universitas Kristen Petra. Aplikasi *mobile ticketing* yaitu aplikasi yang memungkinkan pengguna telepon seluler untuk melakukan reservasi tiket kereta api tanpa harus melakukan antri di stasiun

kereta. Tujuan aplikasi ini adalah membuat aplikasi reservasi tiket kereta api berbasis *M-Commerce*, dimana calon penumpang dapat melakukan reservasi tiket *online* melalui perangkat *wireless*, tanpa harus susah-susah ke stasiun kereta api. Penelitian ini menggunakan metode RAD. Kelebihannya aplikasi ini dapat mempermudah pelanggan dalam melakukan reservasi tiket kereta api. Kekurangannya *handphone* yang digunakan harus mendukung java.

Membangun Prototype *Display* Antrian Pelanggan dengan Menggunakan Teks dan Suara Berbasis Jaringan *Wired* LAN, yang di susun oleh Tri Daryanto dan TW Wisjhnuadji tahun 2008 Universitas Budi Luhur. Aplikasi ini dirancang hanya pada jaringan *Wired* LAN (*Local Area Network*). Aplikasi ini dapat berjalan pada *bandwith* yang cukup rendah yaitu 10 kbps, sehingga apabila terjadi *traffic* yang cukup tinggi sekalipun prototype ini dapat berjalan dengan baik. Kedepan prototype ini dapat di analisa dan diperbandingkan dengan menggunakan *wireless* (jaringan tanpa kebel) LAN maupun Dengan Kabel *Wired*, untuk dapat dilihat kinerja *prototype display* antrian yang kami rancang. Skenario rancangan *testbad* jaringan yang digunakan *display* antrian yang menggunakan tampilan *teks* dan suara dan cetak nomer urut antrian sehingga nasabah tidak perlu lagi mengantri secara konvensional. Dalam rancangan *display* antrian ini menggunakan jaringan yang berbasis *client-server*, dikarenakan jaringan pada bank bersifat 24 jam sehingga tidak memerlukan sumber daya yang lain dalam membangun *display* antrian. *Display* antrian yang dibangun

dapat digunakan bisa lebih dari 1 loket atau *teller* yang berada dalam satu ruangan.

Perancangan dan Pembuatan Prototype Aplikasi SMS Untuk Akses Informasi Anggota Perusahaan Multilevel yang disusun oleh Heryanto 2004 Universitas Kristen Petra. Dengan memadukan teknologi telepon selular dan *Short Message Service* (SMS) dengan sistem perdagangan multilevel, maka dalam tugas akhir ini dibuat aplikasi untuk mengakses informasi-informasi keanggotaan sebuah perusahaan multilevel melalui SMS. Informasi-informasi yang dapat diakses antara lain informasi mengenai jumlah bonus, *down line level* pertama, harga produk, dan daftar produk. Tujuan aplikasi ini adalah untuk membuat program yang dapat memberikan informasi kepada para anggota suatu perusahaan multilevel melalui fasilitas sms secara cepat dan akurat sesuai dengan permintaan para anggota. Penelitian ini menggunakan metode *water fall*. Kelebihan aplikasi ini program ini dirancang otomatis sehingga tidak diperlukan operator, Kekurangannya adalah metode pengiriman informasi hanya berupa sms.

Analisis dan Perancangan Prototype Aplikasi *E-Commerce* yang disusun oleh Heni Jusuf 2008 Universitas Nasional. *E-commerce* merupakan cara berbelanja atau berdagang secara *online* atau *direct selling* yang memanfaatkan fasilitas *internet* dimana terdapat *website* yang dapat menyediakan layanan “*get and deliver*“. *E-commerce* akan merubah semua kegiatan *marketing* dan juga sekaligus memangkas biaya-biaya operasional untuk kegiatan *trading* (perdagangan). Tujuan analisis dan perancangan

prototype ini adalah untuk melakukan transaksi penjualan secara *online*. Studi literatur ini menggunakan arsitektur dan konfigurasi sistem *client/server*. Kelebihannya yaitu Meningkatkan pendapatan dengan menggunakan *online channel* yang biayanya lebih murah serta mengurangi biaya-biaya yang berhubungan dengan kertas, seperti biaya pos surat, pencetakan, *report*, dan sebagainya.

Prototipe Perangkat Lunak Studi dan Implementasi Sistem Antrian yang disusun oleh Sukenda Roosaleh Laksono 2009 Universitas Widyatama. Prototipe perangkat lunak studi sebuah sistem antrian yang berupa prototipe perangkat lunak implementasi antrian. Dalam membangun dan merancang prototipe perangkat lunak implementasi antrian akan mengasumsikan bahwa sebuah kasus sistem antrian yang memiliki satu *server* dan satu antrian dalam sistem antrian dengan disiplin antrian FCFS (*First Come First Serve*). Penelitian ini menggunakan metode OMT (*Object Modelling Technique*). Kelebihannya maka pihak manajemen sistem antrian bisa menentukan keputusan bahwa sistem antrian yang dapat memenuhi kepuasan konsumen didasarkan atas informasi hasil analisa dari perangkat lunak *pro-que*. Kekurangannya pada tahap implementasi perangkat lunak *pro-que* masih menggunakan asumsi dengan kasus *server* dan antrian.

Pembuatan Prototipe Alat Pencampur Cat Berbasis MCS-51, yang disusun oleh Ronald Indarajaya tahun 2002 Universitas Kristen Petra. Penelitian ini membuat suatu simulasi yang berupa prototipe alat pencampur cat otomatis. Prototipe ini terdiri atas sebuah konveyo runtuk menggerakkan

kontainer, solenoida untuk membuka dan menutup valve pada tangki, sensor *infrared* LED sebagai *proximity switch*, dan motor DC untuk mengangkat dan menurunkan timbangan A dan timbangan B, menggerakkan lengan Z, dan mengaduk cat. Sistem kerja dari protipe ini adalah mengisi kontainer dengan cat yang terdapat pada tangki A dan tangki B. Metode yang dipakai untuk mendapatkan perbandingan warna cat tertentu itu, yaitu dengan menimbang berat masing-masing warna cat dengan suatu *transduser LVDT*, sesuai dengan *setting point* yang di-*input*-kan. Tujuannya adalah pembuatan sistem yang mampu mencampurkan 2 warna cat dengan perbandingan tertentu secara tertentu, untuk aplikasi pada dunia industri. Kelebihannya prototype alat pencampur warna mampu mencampurkan dua warna dengan pertimbangan tertentu, kekurangannya pembuatan sistem *scada*, sehingga seluruh proses sistem belum dipantau menggunakan PC.

Analisis dan Implementasi Informasi pembayaran Rekening Air berbasis *Mobile* (*studi kasus*: Perusahaan daerah air minum Tirta Kerta Raharja Kabupaten Tangerang) yang disusun oleh Syukrina Irfiyanda tahun 2009 UIN Syarif Hidayatullah Jakarta. Penelitian ini membuat sistem layanan informasi tagihan listrik air untuk mengatasi masalah yang dihadapi oleh perusahaan dalam memberikan pelayanan informasi dan pembayaran tagihan rekening air kepada pelanggan. Tujuannya adalah untuk memberikan solusi dengan mengimplementasikan sistem informasi pembayaran rekening air berbasis *mobile*. Metode pengembangan sistem yang digunakan adalah metode sekuensial linier. Kelebihan studi literature

ini yaitu sistem informasi pembayaran rekening air berbasis *mobile* ini memberikan pelayanan informasi dan pembayaran tagihan rekening air kepada pelanggan tiap bulannya serta dapat menampung saran dan kritik. Kekurangannya sistem informasi pembayaran rekening air berbasis *mobile* ini belum diterapkan di semua PDAM.

Aplikasi Panduan Kata dalam Mencari Ayat Al-Qur'an Juz 30 Berbasis Java *Mobile*, Disusun oleh Faruq Tataran 2010 UIN Syarif Hidayatullah Jakarta. Penelitian ini membuat aplikasi yang bertujuan untuk Memberikan pengetahuan kepada masyarakat tentang Panduan Kata dalam Mencari Ayat Al-Qur'an Juz 30 berbasis *mobile*, pembuatan aplikasi ini tidak terlepas dari sudah banyaknya perangkat *mobile* yang sudah mendukung aplikasi Java. Metodologi penelitian yang digunakan menggunakan metode Pengembangan RAD yang dibuat oleh James Martin. Aplikasi ini di bangun menggunakan bahasa pemograman Java dan menggunakan MySQL untuk *database* nya. Kelebihan studi literatur ini adalah memberikan informasi berupa potongan-potongan ayat al-quran dalam bentuk latin, aplikasi ini mudah digunakan atau *user friendly* dan mudah digunakan kapan saja dan dimana saja. Kekurangan aplikasi ini sukar mengalami *hang* atau berhenti disaat pemrosesan karena terlalu banyaknya data sehingga memerlukan waktu yang lama dalam proses pencarian, kurang adanya visual suara, aplikasi ini hanya berupa teks.

Aplikasi *Translation* Pada Telepon Seluler Menggunakan J2ME, disusun oleh Satya Bestari 2010 UIN Syarif Hidayatullah Jakarta. Penelitian

ini merancang aplikasi *translation* pada telepon seluler menggunakan J2ME. Aplikasi ini merupakan aplikasi pencarian suatu kata dan menemukan terjemahan dalam bahasa Inggris dan Jepang dengan menggunakan *handphone* sebagai sarannya. Aplikasi ini mampu berjalan atau dapat digunakan pada *handphone* yang berbasis Java MIDP 2.0. Aplikasi ini dapat menghemat penggunaan biaya atau pulsa dalam pencarian kata serta menghasilkan sebuah aplikasi yang *open source*. Metodologi penelitian yang digunakan menggunakan Metode Pengembangan RAD yang dibuat oleh James Martin. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan menggunakan MySQL untuk *database* nya. Kelebihan memiliki kemampuan untuk menambah perbendaharaan kata dalam bahasa asing, kelemahannya hanya bisa digunakan pada *handphone* yang mempunyai kemampuan memori yang cukup besar.

Virtual Reality Prototype Perumahan Berbasis Multimedia 3 Dimensi (Studi kasus: PT Unggul Cemerlang) disusun oleh Muhammad Djupri 2010 UIN Syarif Hidayatullah Jakarta. Penelitian ini yaitu membahas aplikasi yang dibuat untuk presentasi bangunan 3 dimensi yang menampilkan suatu produk perumahan dengan melihat dari dalam bangunan seperti kamar tidur, ruang tamu dan ruang lain di dalam rumah serta sudut pandang yang diinginkan, agar calon konsumen puas dan hasilnya sesuai dengan yang sudah dipresentasikan dengan bangunan 3 dimensi. Tujuannya adalah untuk mengurangi biaya promosi produk perumahan. Metode penelitian yang digunakan adalah metode pengembangan aplikasi multimedia. Kelebihan

aplikasi ini dapat membantu memasarkan produk perumahan. Kekurangannya harus menggunakan spesifikasi *computer* diatas rata-rata agar kinerja aplikasi berjalan baik.

Prototype Aplikasi Media Pembelajaran Rumah Adat Berbasis Multimedia (Studi Kasus : SMP Yayasan Miftahul Jannah (YMJ) Ciputata Timur Tangerang) disusun oleh Imm Imamudin 2010 UIN Syarif Hidayatullah Jakarta. Penelitian ini merancang aplikasi media pembelajaran interaktif bagi siswa mengenai rumah adat yang ada di pulau jawa. Tujuan dibuatnya aplikasi ini adalah membuat aplikasi interaktif sebagai media pembelajaran rumah adat berbasis multimedia untuk membantu pembelajran dalam mengenal budaya Indonesia. Metode penelitian yang digunakan adalah metode pengembangan aplikasi multimedia. Kelebihan aplikasi ini adalah mengenalkan prototype rumah adat secara 3 dimensi dari masing-masing provinsi di pulau jawa. Kekurangannya adalah rumah adat yang disajikan hanya rumah adat yang ada di daerah pulau jawa.

Perancangan dan Pembuatan Prototype Robot Inspeksi Kereta Api, disusun oleh Choirun Niam 2011 Universitas Diponogoro. Penelitian ini, memaparkan tentang perancangan dan pembuatan prototype robot inspeksi rel kereta api. Monitoring suatu rel kereta api menggunakan robot berbasi camera. Tujuan aplikasi ini adalah membuat sistem penggerak motor dengan kendali jarak jauh via *internet* serta memberikan informasi data berupa gambar. Kelebihan studi *literature* ini robot inspeksi kereta api dapat dikontrol dan dikendalikan melalui jaringan *internet* dengan menggunakan

software team viewer. Dimana proses kendali dilakukan disisi *client* dengan menggunakan program hasil desain *software visual studio* yang dilengkapi dengan tampilan kamera. Kekurangannya yaitu robot yang dijalankan dengan kecepatan lebih dari 60 Rpm sehingga gambar yang dihasilkan oleh kamera tidak jelas.

Pembuatan Prototype Pintu Otomatis satu arah berbasis Mikrokontroler ATmega 8535 Menggunakan Double IR, disusun oleh Hendra Maryanto 2010 Universitas Sebelas Maret. Prototipe pintu otomatis satu arah ini dapat memberikan kemudahan untuk membuka dan menutup pintu sehingga dapat menghemat waktu dan tenaga. Tujuan dari tugas akhir ini adalah membuat sebuah prototipe pintu otomatis satu arah. Kelebihan dari studi literature ini adalah prototipe pintu otomatis satu arah berbasis mikrokontroler ATmega 8535 menggunakan *double* IR sehingga pintu dapat terbuka setelah sensor pertama aktif kemudian bisa tertutup kembali setelah sensor kedua aktif. Kekurangannya yaitu pintu otomatis hanya berlaku untuk satu arah saja serta dapat memperkecil arah dan jangkauan sensor PIR.

Pembuatan prototype aplikasi pengiriman informasi kurs valuta asing dengan menggunakan Sms push. Disusun oleh Gunawan Jimmy 2010 Universitas Teknologi Sepuluh November. Tugas Akhir ini dibuat untuk mencoba memecahkan masalah tersebut dengan membuat aplikasi yang dapat mengirimkan informasi kurs secara otomatis untuk setiap jangka waktu tertentu yang telah ditentukan. Untuk implementasinya, digunakan

metode Push pada fasilitas *Short Messaging Service (SMS)* di ponsel. Dengan makin bertambahnya pengguna ponsel yang telah dilengkapi teknologi *Short Messaging Service (SMS)*, maka usaha untuk mendapatkan informasi kurs akan menjadi lebih mudah dan praktis. Dengan aplikasi ini, pengguna dapat mengetahui informasi kurs valuta asing terbaru melalui ponsel mereka, sehingga akan bermanfaat bagi mereka-mereka yang ingin membeli atau menjual produk-produk tertentu yang melibatkan penggunaan mata uang asing yang berbeda.

Prototype sistem pengaman kendaraan bermotor menggunakan gps dan sms. Disusun oleh Firdaus Aji Akbar 2011 Universitas Teknologi Sepuluh November Tugas Akhir ini akan dibuat prototipe sistem pengaman kendaraan bermotor menggunakan gps dan sms. Pada alat ini, pengamanan dilakukan oleh handphone pemilik. Handphone pemilik mengirim sms ke Siemens C55, dan mengaktifkan relay, alarm, dan memanfaatkan teknologi GPS (Global Positioning System) untuk menerima koordinat posisi. Saat relay aktif, maka kendaraan bermotor berhenti bekerja. Ketika GPS receiver menerima data, data tersebut akan dikirim ke mikrokontroler dan akan ditampilkan di LCD dan mengirimkan data ke handphone pemilik untuk memberikan informasi tentang lokasi keberadaan kendaraan. Dari Tugas Akhir ini, dihasilkan sistem pengaman kendaraan bermotor dengan sistem pengiriman bolak balik. Dengan adanya pembuatan sistem yang dirancang, bisa lebih mengurangi kejadian pencurian kendaraan bermotor.

Perancangan Prototype Antena Microstrip untuk aplikasi WiMAX pada Pita frekuensi 5.8 Ghz. Disusun oleh Budiono 2011 Universitas Teknologi Sepuluh November. Pada tugas akhir ini telah dirancang Antena Microstrip berupa Antena Microstrip Tunggal dan *Antena Microstrip Array* yang ditujukan untuk teknologi WiMAX dan mengimplementasikannya ke dalam bentuk prototype. Perancangan menggunakan *software Ansoft High Frequency Structural Simulator (HFSS) v911* dan menganalisis hasil simulasi parameter yang dihasilkan, yaitu koefisien refleksi (S11), *Voltage Standing Wave Ratio (VSWR)*, pola radiasi dua dimensi dan pola radiasi tiga dimensi. Penelitian ini bertujuan untuk merancang dan membuat prototype Antena Microstrip Tunggal dan *Antena Microstrip Array* untuk aplikasi Generasi Lanjut Sistem Komunikasi Nirkabel yang beroperasi pada frekuensi 5,8 GHz. Kelebihan penelitian ini VSWR yang diperoleh pada simulasi perancangan antena dengan software Ansoft HFSS v11 telah memenuhi standar ideal ($VSWR \leq 1.5$).

Pengembangan prototype perangkat lunak *inter-office mail dynamic courier system (IOMDCS)* dengan Teknologi sms. Disusun oleh Hidayat Arief Saiful 2009 Universitas Teknologi Sepuluh November. Penelitian ini akan mengembangkan penelitian-penelitian sebelumnya yaitu aplikasi *Inter-office Mail Courier System* dan pengembangan *model Dynamic Pickup and Delivery With Time Windows*. Sehingga tercipta penelitian *Inter-Office Mail Dynamic Courier System* sebagai implementasi lanjutan yang berdasar pada penelitian sebelumnya. Sebagai *support system tool*, digunakan teknologi

SMS sebagai alat penyaluran informasi *rute* baru kepada kurir secara cepat, tepat, dan hemat dalam pengembangan prototype aplikasi ini. Selain itu, aplikasi ini terkandung sistem informasi secara intranet dalam rangka mempermudah penyaluran informasi dua arah dari/kepada *client* maupun *Inter-Office Mail*

Analisa dan perancangan prototype sistem informasi penempatan spot iklan pada stasiun radio di PT. Kubus Media Komunikasi. Disusun oleh Kamajaya, Bagus 2009 Universitas Teknologi Sepuluh November. PT Kubus Media Komunikasi merupakan salah satu perusahaan yang menjalankan kegiatan jasa biro iklan. Pergeseran *fungsi public relation* di perusahaan – perusahaan produsen menyebabkan makin terfokusnya bidang kerja biro iklan, dalam hal ini PT Kubus Media Komunikasi makin memperkuat posisinya sebagai media *buyer agency* khususnya media radio. Semakin meningkatnya jumlah media plan yang harus dilayani menyebabkan keterlambatan pemrosesan administrasi dan keterlambatan proses konfirmasi antara beberapa bagian di perusahaan sehingga membuat pelayanan kepada pelanggan kurang cepat dan rawan terhadap persaingan dengan biro iklan lainnya. Untuk mengatasi kelemahan tersebut perusahaan memerlukan suatu sistem pemrosesan informasi baru yang dapat diandalkan. Penelitian ini dilakukan dengan melalui empat tahap yaitu tahap perencanaan penelitian, tahap analisa, tahap disain sistem dan prototyping. Hasil dari penelitian ini diharapkan adalah sebuah prototype sistem

informasi pemrosesan penempatan iklan di media radio dan dokumen analisa sistemnya.

Pembuatan prototype aplikasi *wireless* menggunakan teknologi java pada sistem informasi akademik politeknik Caltex Riau. Disusun oleh Sari, Juni Nurma 2007 Universitas Teknologi Sepuluh November. Aplikasi *wireless* dirancang untuk dapat mengakses Sistem Informasi Akademik Pada penelitian ini, Sistem Informasi akademik yang akan diakses melalui telepon selular adalah registrasi, informasi nilai, informasi indeks prestasi, informasi jadwal dan informasi dosen. Aplikasi ini merupakan aplikasi *client* dan *server*, sebagai *client* adalah telepon selular dengan MIDP dan J2ME dan sebagai *server* adalah *servelt* atau *JSP container* yang menyediakan informasi-informasi yang dibutuhkan. Di dalam analisis dan perancangan ini akan dilakukan pendekatan-pendekatan untuk mengumpulkan informasi dan data yang diperlukan, yaitu dengan melakukan wawancara dengan administrasi akademik, *survey* untuk mengetahui kondisi di lingkungan mahasiswa PCR, mengenai pemakaian telepon selular, tipe telepon selular yang digunakan, *survey* untuk mengetahui pendapat manajemen apabila aplikasi ini diimplementasi, wawancara dengan administrator SIAK serta studi literatur tentang Sistem Informasi Manajemen PCR modul akademik Dari pendekatan-pendekatan tersebut dilakukan analisa sehingga diketahui kebutuhan user untuk mengetahui informasi-informasi penting yang dapat diakses melalui telephone selular. Baru kemudian dilakukan perancangan sistem dan

estimasi biaya apabila diimplementasikan. Dengan dilakukannya penelitian ini diharapkan menghasilkan suatu rancangan system yang membantui proses registrasi dan dapat menampilkan informasi-informasi akademik dengan biaya yang minimal dan manfaat yang besar. Dan diharapkan dapat sebagai acuan untuk merancang aplikasi wireless untuk Sistem Informasi yang lain, yaitu: Sistem Informasi Kepegawaian, Perpustakaan, Inventory dan Keuangan.

Rancang bangun sistem informasi akademik dengan metode prototype di MA Ponpes Amanatul Ummah. Disusun oleh Hidayat Mochammad Yasin 2011 Universitas Teknologi Sepuluh November. Proses penilaian siswa di Madrasah Aliah Akselerasi (MAA) di pondok pesantren Amanatul Ummah adalah proses yang penting. Semakin banyaknya data dan lokasi MAA yang terdiri dari 2 lokasi berjauhan maka diperlukan aplikasi khusus untuk otomatisasi proses ini. Aplikasi yang sudah ada saat ini tidak dapat diimplementasikan di MAA karena perbedaan periode pendidikan di MAA. Tugas akhir ini membahas rancangan dan pembuatan aplikasi system penilaian siswa. Rancangan aplikasi akan dibuat menggunakan UML yang terdiri dari *domain model*, *use case*, *activity diagram*, *diagram kelas*, basis data, dan antarmuka. Aplikasi akan dibuat dengan menggunakan metode prototipe. Hasil penelitian ini adalah disain dan aplikasi sistem informasi pengolahan data nilai siswa. Adanya sistem ini diharapkan proses pengolahan data nilai siswa dapat dilaksanakan dengan lebih efisien.

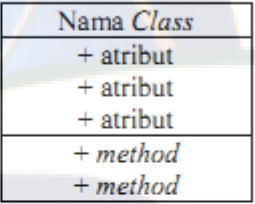



Dari beberapa penelitian sejenis yang dijadikan sebagai perbandingan dengan penelitian ini, terdapat beberapa kekurangan yang kemudian dijadikan sebagai acuan untuk pengembangan yang lebih baik pada aplikasi yang dirancang ini. Dari hasil studi penelitian sejenis yang dilakukan, maka dapat disimpulkan beberapa kelebihan yang ada pada penelitian ini, yaitu sebagai berikut:

1. Prototype Aplikasi yang dirancang, berguna untuk membantu pelanggan dalam memperoleh informasi seputar tagihan rekening listrik melalui *mobile device*.
2. Pengembangan aplikasi bersifat *mobile* sehingga bisa diakses oleh pengguna dari mana saja dan kapan saja.
3. Prototype Aplikasi ini menggunakan bahasa pemrograman java dan PHP serta *database mysql* yang bersifat *multi flatform*.

2.17 Simbol – Simbol UML

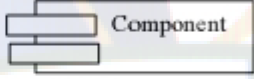

1. Class Diagram

Tabel 2.1 *Class Diagram*
(Sumber Grady Booch)

Nama Komponen	Keterangan	Simbol
Class	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .	
Association	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).	
Composition	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
Dependency	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.	




2. Componen Diagram

Tabel 2.2 *Componen Diagram*
(Sumber Grady Booch)

Nama Komponen	Keterangan	Simbol
<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya.	
<i>Dependency</i>	Sebuah <i>Dependency</i> digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung.	

3. Deployment Diagram

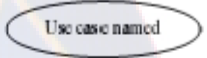


Tabel 2.3 *Deployment Diagram*
(Sumber Grady Booch)

Nama Komponen	Keterangan	Simbol
<i>Component</i>	Pada <i>deployment</i> diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.	
<i>Node</i>	Node menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.	
<i>Association</i>	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen-komponen <i>hardware</i> .	

4. Use Case Diagram

Tabel 2.4 Use Case Diagram

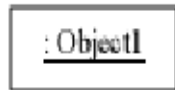

(Sumber Grady Booch)



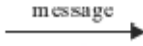
Nama Komponen	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>use case</i> dituliskan didalam elips tersebut.	
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .	
<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>Actor</i> dengan <i>Use Case</i> .	

5. Sequence Diagram

Tabel 2.5 Sequence Diagram

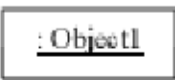

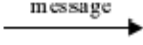
(Sumber Grady Booch)

Nama Komponen	Keterangan	Simbol
<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor</i> Use Case Diagram.	

Nama Komponen	Keterangan	Simbol
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.	
<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object</i> - <i>object</i> .	











6. Collaboration Diagram

Tabel 2.6 Collaboration Diagram
(Sumber Grady Booch)

Nama Komponen	Keterangan	Simbol
<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama pada <i>Actor Use Case Diagram</i> .	
<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek.	

7. Activity Diagram

Tabel 2.7 Activity Diagram
(Sumber Grady Booch)

Simbol	Keterangan
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk mengambil keputusan
	<i>Fork</i> ; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Rake</i> ; Menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Untuk melengkapi dan mendukung materi dalam uraian pembahasan, dibutuhkan sejumlah data atau informasi yang berkaitan dengan penelitian karya ilmiah ini. Oleh karena itu, penulis melakukan penelitian guna mendapatkan data yang *real* sesuai referensi yang diperlukan. Esensial dalam metode ini bertujuan untuk melengkapi proses penelitian selanjutnya. Metode ini meliputi penelitian lapangan (observasi, wawancara dan kuesioner) dan studi pustaka. Berikut ini adalah rincian metode pengumpulan data yang dibutuhkan :

3.1.1 Studi Lapangan

Studi lapangan dilakukan dengan dua cara, yaitu:

1. Wawancara (*Interview*)

Metode ini dilakukan dengan mengadakan tanya jawab langsung dengan bagian IT yang ada di PLN Distribusi Jakarta Raya dan Tangerang pada hari senin tanggal 5 September 2011 pukul 13.30 WIB. Dalam wawancara yang dilakukan, penulis mengumpulkan beberapa data dan informasi yang berhubungan dengan perancangan aplikasi informasi rekening listrik PLN berbasis *mobile*.

2. Pengamatan (*Observasi*)

Guna mengumpulkan informasi mengenai kebutuhan sistem penulis melakukan pengumpulan data dengan cara observasi di tempat penelitian, yang dalam hal ini adalah PT. PLN Persero Distribusi Jakarta Raya dan Tangerang, Jl.M.I.R.Rais No.1 Jakarta Pusat 10110. Penulis mengamati langsung ke lapangan pada hari Senin, 12 September 2011. Hal ini perlu dilakukan agar penulis dapat melakukan analisis terhadap sistem yang telah berjalan.

3. Kuesioner

Kuesioner adalah dokumen berisi daftar pertanyaan yang digunakan untuk mendapatkan informasi dari pelanggan. Dengan kuesioner peneliti berharap dapat mengetahui bahwa aplikasi sudah sesuai dengan yang diinginkan.

3.1.2 Studi Pustaka

Pada tahapan pengumpulan data dengan cara studi pustaka, penulis mencari referensi-referensi yang relevan dengan objek yang akan diteliti. Pada metode studi pustaka ini, penulis membandingkan dua puluh tiga karya ilmiah yang sejenis sebagai pembanding dalam prototype aplikasi info rekening listrik PLN berbasis *mobile*. Batasan yang penulis lakukan dalam mengevaluasi penelitian-penelitian terdahulu terfokus pada dalam prototype aplikasi info rekening listrik PLN berbasis *mobile*. Penulis akan menjelaskan mengenai tujuan

sistem yang dibuat, metodologi yang digunakan, kelebihan dan kekurangan dari beberapa karya ilmiah, Penjelasan ini dapat dilihat pada bab II.

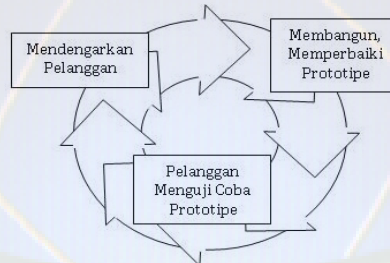
3.2 Metode Pengembangan Sistem

Dalam penelitian ini metode pengembangan sistem yang digunakan adalah model Prototype. Adapun alasan penulis menggunakan metodologi ini karena lebih menghemat waktu dalam pengembangan sistem, pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan, penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkan (Ladjamuddin, 2006:26).

Model Prototype penulis gunakan untuk mengatasi ketidakserasian antara pelanggan dan pengembang, maka harus dibutuhkan kerjasama yang baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalam menyelesaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan

Model ini dimulai dengan mengumpulkan kebutuhan dimana pengembang dan pelanggan bertemu dan mengidentifikasi objektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan

keharusan dan kemudian dilakukan perancangan kilat (Pressman, 2002:40).
Dibawah ini adalah gambar model prototype paradigma menurut Roger Pressman.



Gambar 3.1 Model Prototype Paradigma

Pengembangan sistem tersebut dalam pelaksanaanya penulis lakukan menggunakan tiga tahap siklus pengembangan model Prototype yaitu:

3.2.1 Mendengarkan Pelanggan

Pada tahap ini penulis dimulai dengan pengumpulan kebutuhan. Pada tahap ini penulis melakukan analisis kebutuhan dan mengidentifikasi segala kebutuhan. Dari analisis tersebut dapat ditetapkan tujuan perancangan, pengajuan usulan solusi yang dapat diterima.

Tahap yang dilakukan antara lain:

1. Mengalisa sistem yang berjalan pada tempat penelitian.
2. Melakukan indentifikasi masalah yang terjadi pada proses pelayanan pelanggan PLN.
3. Usulan penyelesaian masalah, analisis disini adalah dengan meninjau dari sisi kebutuhan, pihak yang terlibat, faktor pendukung dan kendala yang dihadapi.

3.2.2 Membangun Memperbaiki Prototype

Setelah mengetahui definisi aplikasi yang akan dikembangkan maka tahapan berikutnya dibagi menjadi dua bagian diantaranya:

3.2.2.1 Perancangan sistem

Perancangan disini dimaksudkan untuk membuat pemodelan sementara terhadap aplikasi baru yang dapat membantu sistem yang berjalan saat ini di PLN Distribusi Jakarta Raya dan Tangerang dan pembuatan program untuk sistem aplikasi. *Design* yang dimaksudkan meliputi perancangan aplikasi dan perancangan *database*.

1. Desain Aplikasi

Untuk perancangan Aplikasi, didesign menggunakan *Unified Modelling Language (UML)*. Hal ini dilakukan untuk memudahkan pengembangan sistem, selain itu penggunaan UML lebih cocok digunakan dalam perancangan aplikasi bersifat *object oriented*. Perancangan aplikasi yang penulis lakukan dengan menggunakan *tools* UML ini meliputi :

- a. Identifikasi *Actor*
- b. Perancangan *Use Case* Diagram
- c. Perancangan *Use Case Scenario*
- d. Perancangan *Activity* Diagram

e. Perancangan *Sequence Diagram*

f. Perancangan *Class diagram*

2. Desain Basis Data

Pada desain basis data ini, data-data yang digunakan dalam suatu aplikasi akan disimpan dalam *database*. Desain basis data dilakukan dengan merancang tabel-tabel atau *record store* yang digunakan untuk menyimpan data. Pada tahapan ini penulis lakukan :

- a. Penerjemahan *class diagram* ke dalam basis data.
- b. Menampilkan struktur basis data.

3. Desain *Interface*

Pada tahap ini, penulis melakukan perancangan terhadap *user interface* dari aplikasi ini. Perancangan yang dilakukan meliputi halaman-halaman yang ada dalam aplikasi.

4. Desain Alur Komunikasi Data

Pada tahap ini dijelaskan desain alur komunikasi data, dimana terdapat penjelasan mengenai proses permintaan data dari *client* ke *server*.

3.2.2.2 Pengkodean sistem

Pada tahapan ini dilakukan tahap pembuatan program yang telah dirancang sebelumnya menggunakan netbeans IDE 6.0 untuk membuat *user interface* dan *source*

code dan *emulator J2ME wireless toolkit 2.5* untuk menjalankan program dalam PC dan *My SQL* untuk *Database*. Pada tahapan ini juga penulis mencantumkan kelengkapan *software* dan *hardware* yang digunakan untuk membuat aplikasi ini.

3.2.3 Pengujian Prototype

Pada tahap ini dilakukan beberapa tahapan, antara lain :

1. Spesifikasi ponsel yang digunakan.
2. *Transfer* aplikasi dari PC ke ponsel.
3. Pengujian Aplikasi pada Ponsel Nokia E63.

Pengujian aplikasi dilakukan pada ponsel dengan menguji seluruh fitur yang terdapat pada aplikasi informasi rekening listrik PLN berbasis *mobile* menggunakan teknik *Black Box Testing*.

4. Evaluasi Sistem

Pada tahap ini dilakukan evaluasi untuk mendapatkan tanggapan dari pelanggan untuk mengetahui apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan, Evaluasi sistem ditentukan dengan melakukan kuesioner.

BAB IV

PEMBAHASAN

4.1 Gambaran Umum Perusahaan

4.1.1 Sejarah Perusahaan

Perjalanan berdirinya PLN Distribusi Jakarta Raya dan Tangerang dimulai sejak tahun 1897, yaitu dengan digarapnya pembangkit tenaga listrik (PLTU) oleh salah satu perusahaan Belanda (NV NIGM) yang berlokasi di Gambir. Pada tahun 1947 perusahaan ini berganti nama menjadi NV OGEM. Sesuai Keputusan Menteri PU dan Tenaga No. U 16/9/I per 30 Desember 1953, NV OGEM diambil alih oleh Pemerintah Indonesia. Serah terima antara NV OGEM dengan pemerintahan Indonesia dilakukan pada 1 Januari 1954, yang menandai pula awal mula pengelolaan listrik ke Perusahaan Listrik Jakarta dengan wilayah kerja yang meliputi Jakarta Raya serta Ranting Kebayoran dan Tangerang.

4.1.2 Visi dan Misi PLN Distribusi Jakarta dan Tangerang

1. Visi

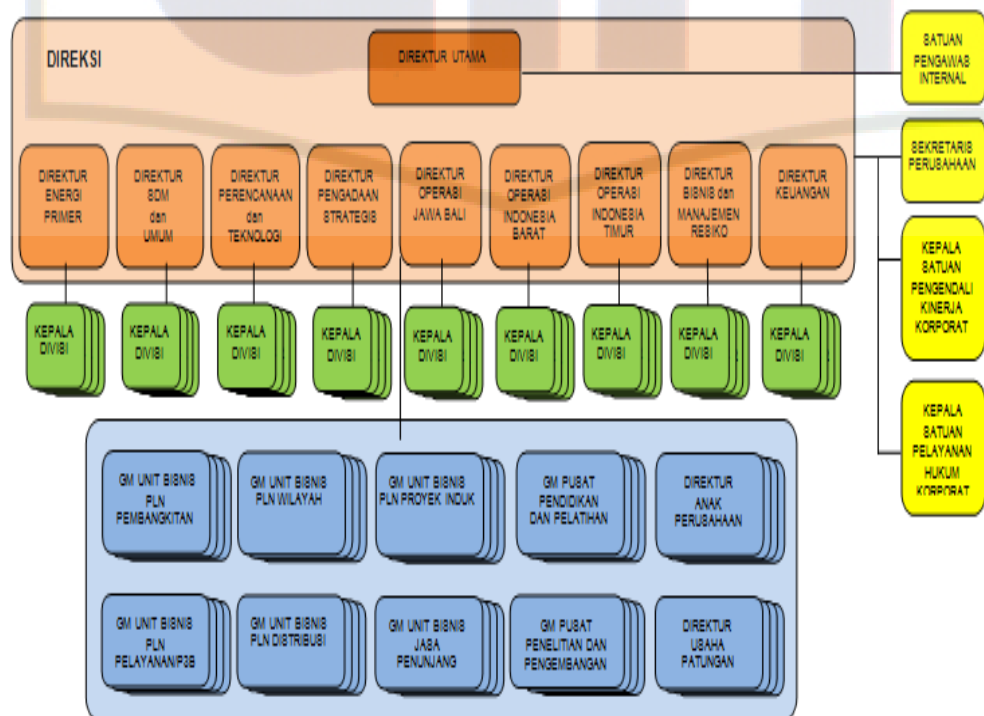
Menjadi perusahaan distribusi tenaga listrik yang handal, tangguh dan berkembang.

2. Misi

- a) Melaksanakan bisnis distribusi tenaga listrik yang berorientasi kepada pelanggan, karyawan dan pemilik.
- b) Meningkatkan profesionalisme Sumber Daya Manusia.
- c) Menjadikan bisnis tenaga listrik sebagai sarana pendorong pertumbuhan ekonomi nasional.
- d) Melaksanakan usaha sesuai kaidah bisnis.

4.1.3 Struktur Organisasi

Adapun struktur organisasinya sebagai berikut:



Gambar 4.1 Struktur Organisasi PLN

4.2 Metode Pengembangan Sistem

Metode yang digunakan untuk melakukan prototype aplikasi info rekening listrik PLN berbasis *mobile* adalah metode pengembangan sistem dengan model Prototype. Berikut adalah tiga tahap pengembangan model prototype menurut Roger Pressman.

4.2.1 Mendengarkan Pelanggan

Pada tahap ini penulis dimulai dengan pengumpulan kebutuhan. Pada tahap ini penulis melakukan analisis kebutuhan dan mengidentifikasi segala kebutuhan.

4.2.1.1 Analisis Sistem yang Berjalan

Sistem yang berjalan saat ini di PLN Distribusi Jakarta Raya dan Tangerang masih menggunakan sistem manual yaitu sistem *call center* 123, dimana sistem *call center* 123 melayani *service* pelanggan melalui saluran telepon seperti informasi rekening listrik PLN.

4.2.1.2 Identifikasi Masalah

Berdasarkan hasil analisis pada sistem berlangsung, terdapat masalah yang dihadapi yaitu banyaknya Pelanggan PLN yang mengakses informasi rekening listrik pada Awal bulan melalui saluran *telephone Call Center* PLN 123, kondisi ini tidak sebanding dengan petugas operator telepon yang ada.

4.2.1.3 Usulan Penyelesaian Masalah

Dengan melihat permasalahan yang ada maka di butuhkan suatu sistem yang dapat membantu mengatasi masalah *customer service*, serta dapat membantu pelanggan untuk mengetahui rekening listrik dengan praktis dan mudah.

4.2.2 Membangun Memperbaiki Prototype

Pada tahapan desain ini, penulis akan merancang sistem yang akan dibangun. Adapun rancangan sistem yang penulis buat meliputi rancangan penentuan *aktor*, perancangan *use case*, perancangan *use case scenario* atau narasi *use case*, *activity diagram*, *Sequence diagram*, *class diagram*, desain *database*, desain *interface* dan desain alur sistem.

4.2.2.1 Desain Aplikasi

1. Penentuan *Aktor*

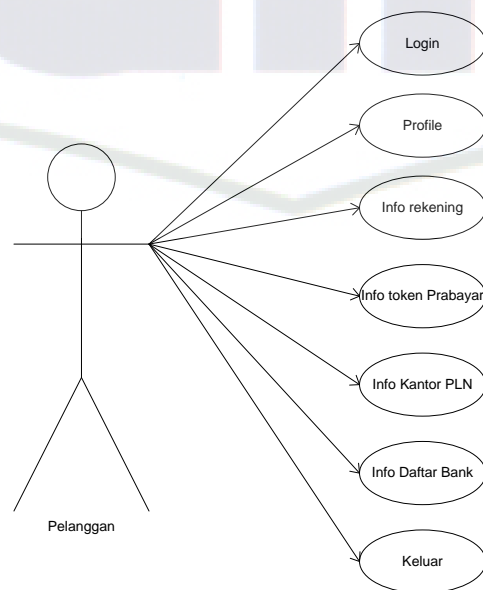
Adapun *aktor* yang terlibat pada sistem ini adalah *user* atau pelanggan. Pelanggan merupakan *user* yang telah terdaftar dalam sistem. Seorang *user* dapat *meregister* diri mereka pada aplikasi info rekening listrik berbasis *mobile*, sehingga dapat melihat konten yang ada dalam aplikasi tersebut.

Tabel 4.1 Penentuan Aktor

NO	Aktor	Keterangan
1.	User	Adapun hak-hak yang bisa dilakukan oleh <i>aktor user</i> adalah registrasi, <i>login</i> , melihat informasi rekening, melihat informasi token prabayar, melihat <i>profile</i> , melihat informasi kantor PLN dan melihat informasi daftar Bank.

2. Perancangan Use case

Aplikasi info Rekening Listrik PLN



Gambar 4.2 Perancangan Use Case

3. Perancangan Use case Scenario

Usecase scenario merupakan penjelasan yang lebih terperinci mengenai masing-masing *usecase* yang terjadi dalam sistem. *Usecase diagram* terdiri dari :

1. Nama *Usecase* adalah nama *usecase* yang akan dideskripsikan.
2. *Aktor* yang terlibat.
3. *Description* adalah deskripsi *Usecase*.
4. *Trigger* adalah tujuan dari *usecase*.
5. *Pre condition* adalah syarat penting bagi *usecase* untuk memulai.
6. *Basic flow* adalah kegiatan yang dilakukan oleh *usecase*.
7. *Post condition* adalah kegiatan setelah *usecase* selesai dikerjakan.

Berikut ini adalah *usecase scenario* dari *diagram usecase* di atas:

a. Use case Register

Tabel 4.2 Narasi *Use case Register*

<i>Use case Name</i>	<i>Register</i>
<i>Aktor</i>	<i>User</i>
<i>Description</i>	<i>Use case</i> ini digunakan untuk melakukan registrasi sebagai <i>user</i> .
<i>Precondition</i>	<i>User</i> belum menjadi <i>user</i> dan telah berada pada halaman login.
<i>Trigger</i>	Setiap <i>user</i> yang ingin memiliki hak akses penuh menggunakan sistem harus memiliki akun.

<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik registrasi. 3. Mengisi <i>form</i> registrasi.	2. Menampilkan <i>form</i> registrasi. 4. Menampilkan <i>message</i> : validasi sukses.
<i>Postcondition</i>	<i>User</i> dapat melakukan <i>login</i> ke dalam sistem dengan <i>username</i> and <i>password</i> yang telah di <i>register</i> sebelumnya.	

b. Use case Login

Tabel 4.3 Narasi *Use case Login*

<i>Use case Name</i>	<i>Login</i>	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>user</i> melakukan <i>login</i> ke dalam sistem.	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> berada pada halaman <i>login</i> .	
<i>Trigger</i>	Saat <i>user</i> ingin mengakses informasi mereka secara penuh di dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. <i>Input username</i> dan <i>password</i> di <i>form login</i>	. 3. Cek <i>username</i> dan

	2. Klik <i>login</i> .	<i>password</i> .
		4. Menampilkan halaman utama.
<i>Postcondition</i>	Dapat melakukan pemilihan menu pada masing-masing status keanggotaan <i>user</i> .	

c. Use case Profile

Tabel 4.4 Narasi *Use case Profile*

<i>Use case Name</i>	<i>Profile</i>	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>aktor</i> pada saat ingin melihat <i>profile</i> .	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .	
<i>Trigger</i>	Saat <i>aktor</i> ingin melihat profil dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik menu. 3. Klik profil.	2. Menampilkan halaman <i>menu</i> . 4. Menampilkan halaman informasi profil.
<i>Postcondition</i>	Isi <i>profile</i> dapat terlihat oleh aktor.	

d. Use case Info Rekening

Tabel 4.5 Narasi *Use case* Info Rekening

<i>Use case Name</i>	Info Rekening	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>aktor</i> pada saat ingin melihat informasi rekening.	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .	
<i>Trigger</i>	Saat <i>aktor</i> ingin melihat informasi rekening dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik menu. 3. Klik informasi rekening.	2. Menampilkan halaman <i>menu</i> 4. Menampilkan halaman informasi rekening.
<i>Postcondition</i>	Isi informasi rekening dapat terlihat oleh aktor.	

e. Use case Info Token Prabayar

Tabel 4.6 Narasi *Use case* Info Token Prabayar

<i>Use case Name</i>	Info Token Prabayar
<i>Aktor</i>	<i>User</i>
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>aktor</i> pada saat ingin melihat info token.
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .

<i>Trigger</i>	Saat <i>aktor</i> ingin melihat info token dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik menu. 3. Klik info token.	2. Menampilkan halaman <i>menu</i> . 4. Menampilkan halaman info token prabayar.
<i>Postcondition</i>	Isi info token prabayar dapat terlihat oleh aktor.	

f. Use case Info Kantor PLN

Tabel 4.7 Narasi *Use case* Info Kantor PLN

<i>Use case Name</i>	Info Kantor PLN	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>aktor</i> pada saat ingin melihat daftar alamat kantor PLN.	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .	
<i>Trigger</i>	Saat <i>aktor</i> ingin melihat daftar alamat kantor PLN dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik menu. 3. Klik kantor PLN.	2. Menampilkan halaman <i>menu</i> . 4. Menampilkan info daftar kantor PLN.
<i>Postcondition</i>	Isi daftar alamat kantor PLN dapat terlihat oleh aktor.	

g. Use case Info Daftar Bank

Tabel 4.8 Narasi *Use case* Info Daftar Bank

<i>Use case Name</i>	Info Daftar Bank	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan <i>aktor</i> pada saat ingin melihat daftar bank.	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .	
<i>Trigger</i>	Saat <i>aktor</i> ingin melihat daftar bank dalam aplikasi.	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>
	1. Klik menu. 3. Klik Daftar Bank.	2. Menampilkan halaman <i>menu</i> . 4. Menampilkan daftar bank.
<i>Postcondition</i>	Isi informasi daftar bank dapat terlihat oleh aktor.	

h. Use case Keluar

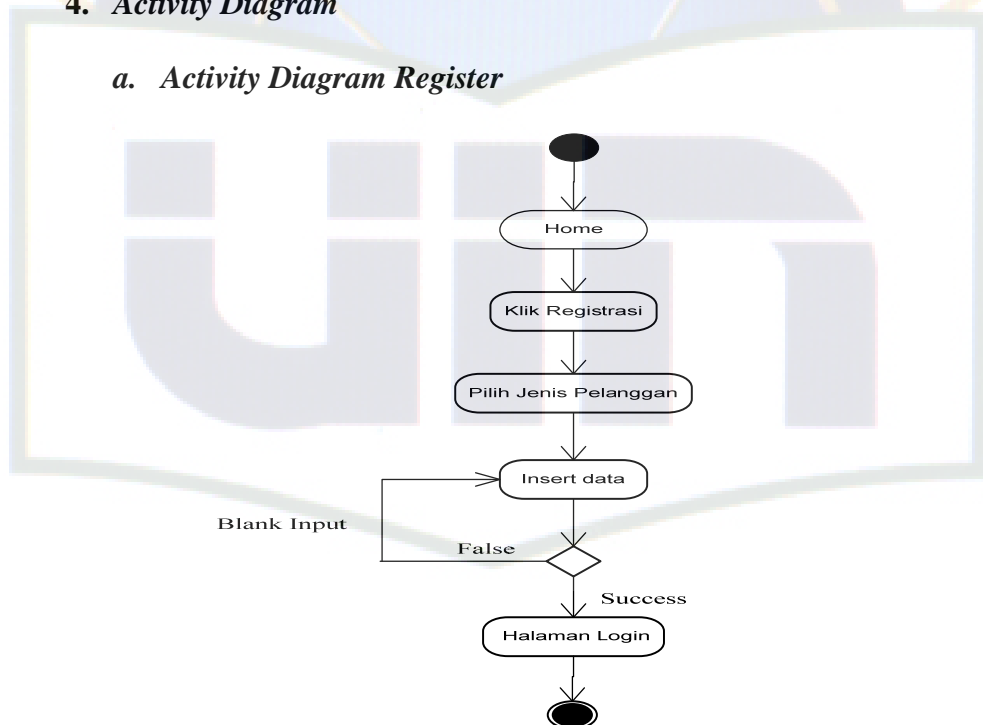
Tabel 4.9 Narasi *Use case* keluar

<i>Use case Name</i>	<i>keluar</i>	
<i>Aktor</i>	<i>User</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan proses kejadian saat <i>aktor</i> ingin keluar dari aplikasi.	
<i>Precondition</i>	<i>Use case</i> ini dilakukan setelah <i>aktor</i> melakukan <i>login</i> .	
<i>Trigger</i>	Saat <i>aktor</i> ingin keluar dari aplikasi .	
<i>Basic flow</i>	<i>Aktor action</i>	<i>System Response</i>

	1. Klik keluar.	2. menampilkan halaman <i>login</i> .
<i>Postcondition</i>	Aktor keluar dari aplikasi dan menampilkan halaman <i>login</i>	

4. Activity Diagram

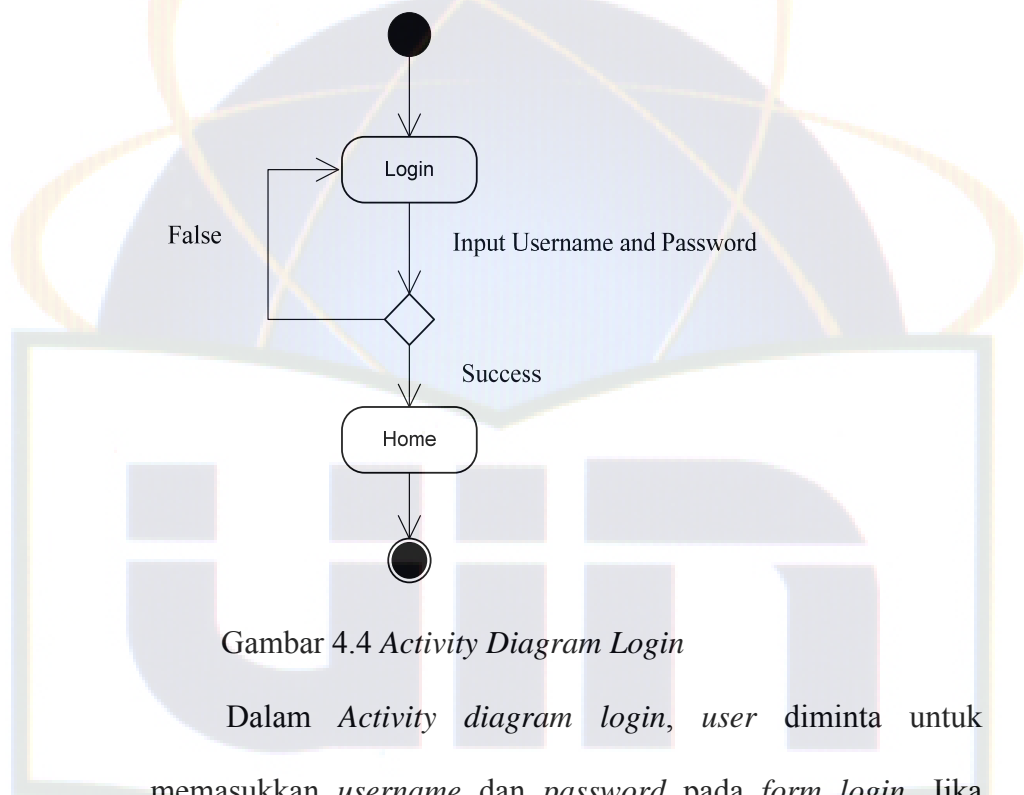
a. Activity Diagram Register



Gambar 4.3 Activity Diagram Register

Dalam *activity diagram register* ini menjelaskan urutan aktifitas yang harus dilakukan oleh pengguna yang ingin melakukan registrasi pada sistem aplikasi ini. Adapun langkah-langkah yang harus dilakukan oleh pengguna meliputi pengisian data formulir pendaftaran, setelah berhasil melakukan pendaftaran, barulah pengguna bisa *login* ke dalam aplikasi ini.

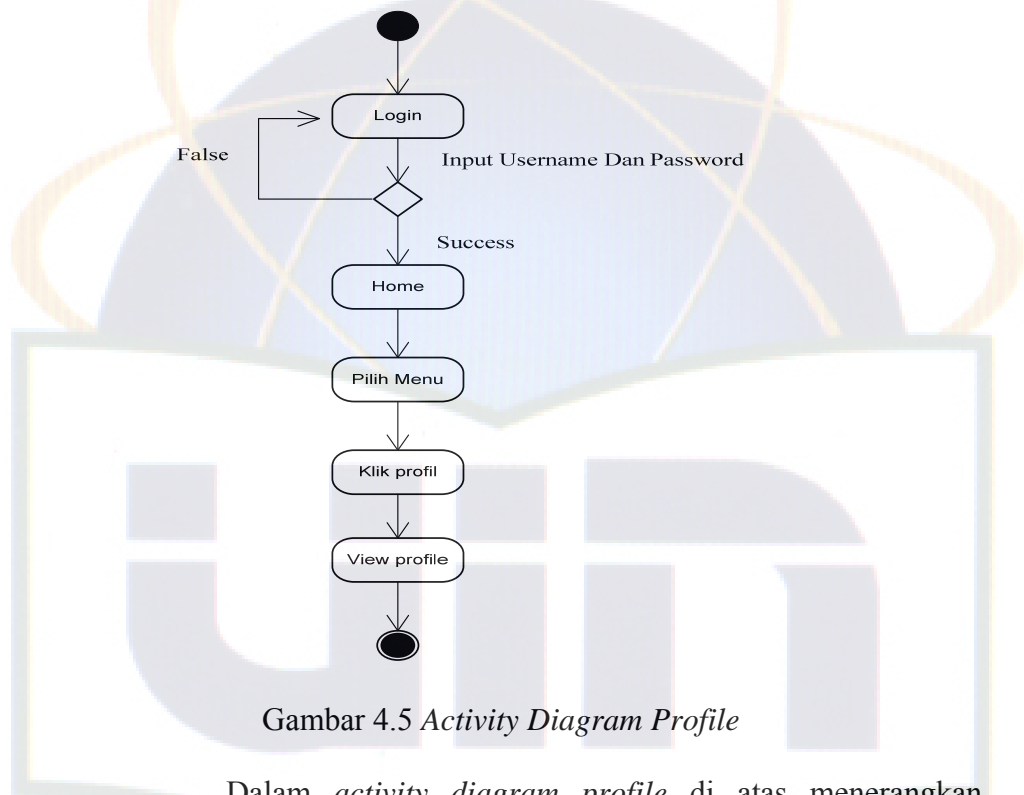
b. Activity Diagram Login



Gambar 4.4 Activity Diagram Login

Dalam Activity diagram login, user diminta untuk memasukkan username dan password pada form login. Jika username dan password yang user masukan salah, maka sistem akan menuju kembali ke form login dengan pesan username dan password salah. Apabila username dan password benar maka user akan masuk kehalaman home.

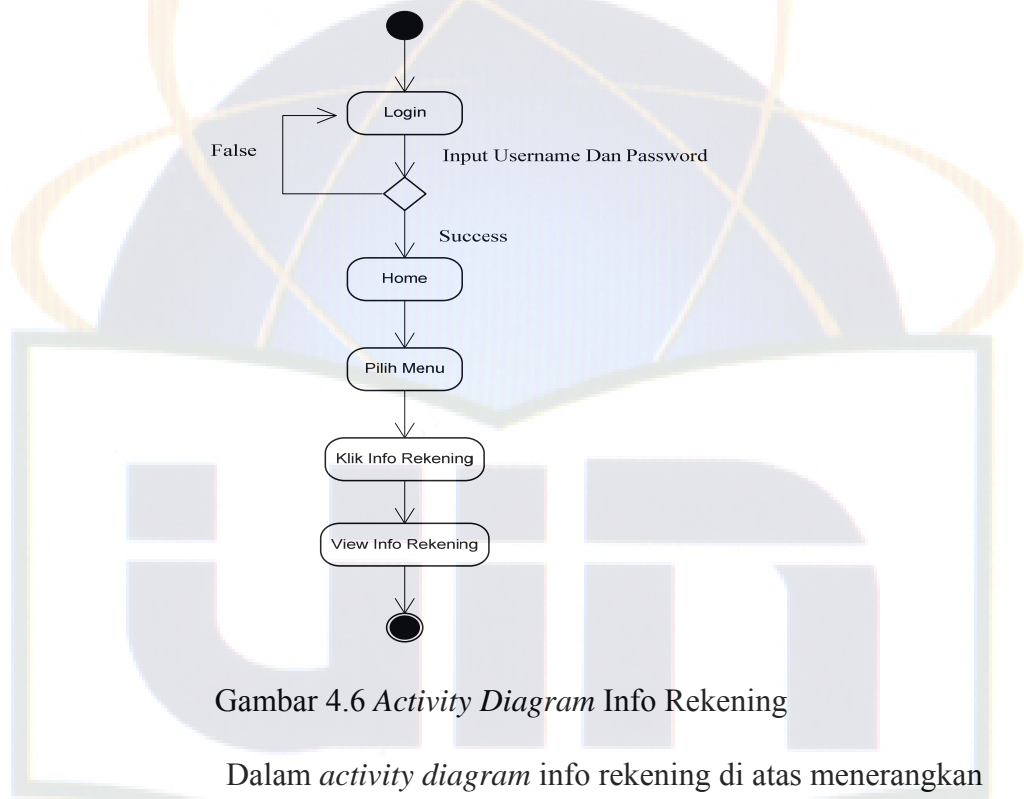
c. Activity Diagram Profile



Gambar 4.5 Activity Diagram Profile

Dalam *activity diagram profile* di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk melihat profil. Sebelumnya *user* harus melakukan *login* terlebih dahulu kedalam sistem dengan menginput *username* dan *password*, setelah berhasil melakukan *login* maka sistem akan menampilkan halaman *home*. Kemudian klik menu, untuk melihat daftar yang ada di menu lalu pilih *profile*, maka sistem akan menampilkan halaman *profile user*. Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram profile* di atas.

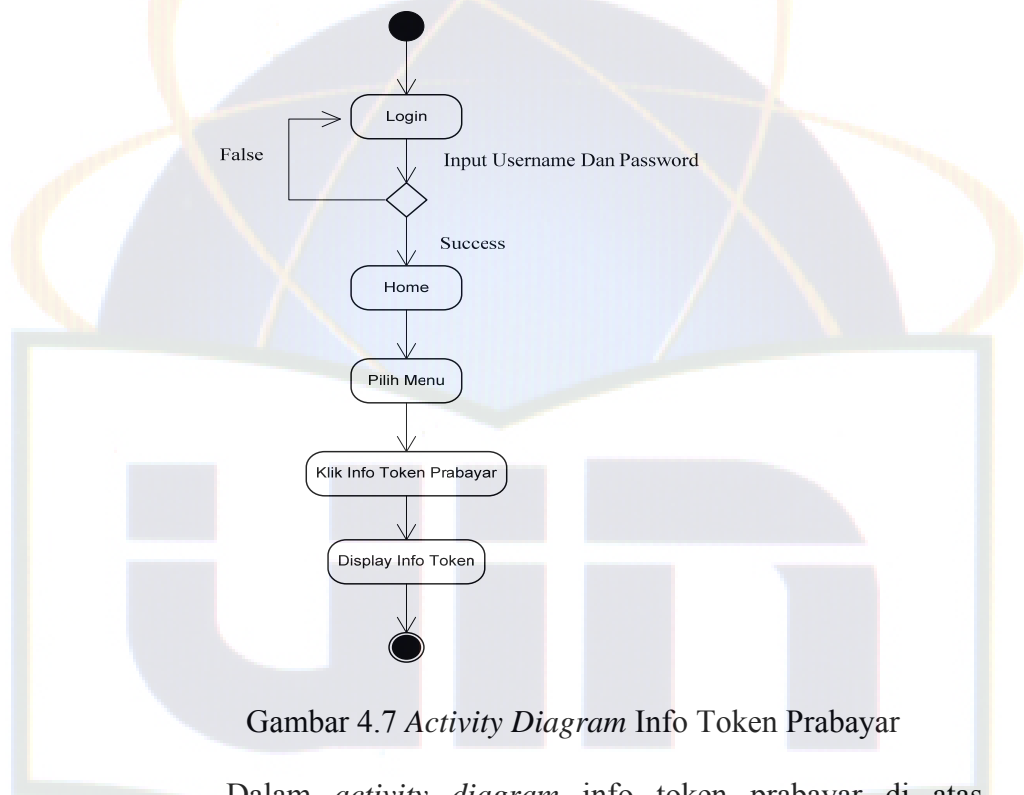
d. Activity Diagram Info Rekening



Gambar 4.6 Activity Diagram Info Rekening

Dalam *activity diagram* info rekening di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk melihat info rekening. Sebelumnya *user* harus melakukan *login* terlebih dahulu kedalam sistem dengan menginput *username* dan *password*, setelah berhasil melakukan *login* maka sistem akan menampilkan halaman *home*. Kemudian klik menu, untuk melihat daftar yang ada di menu lalu pilih info rekening, maka sistem akan menampilkan halaman informasi rekening. Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram* info rekening di atas.

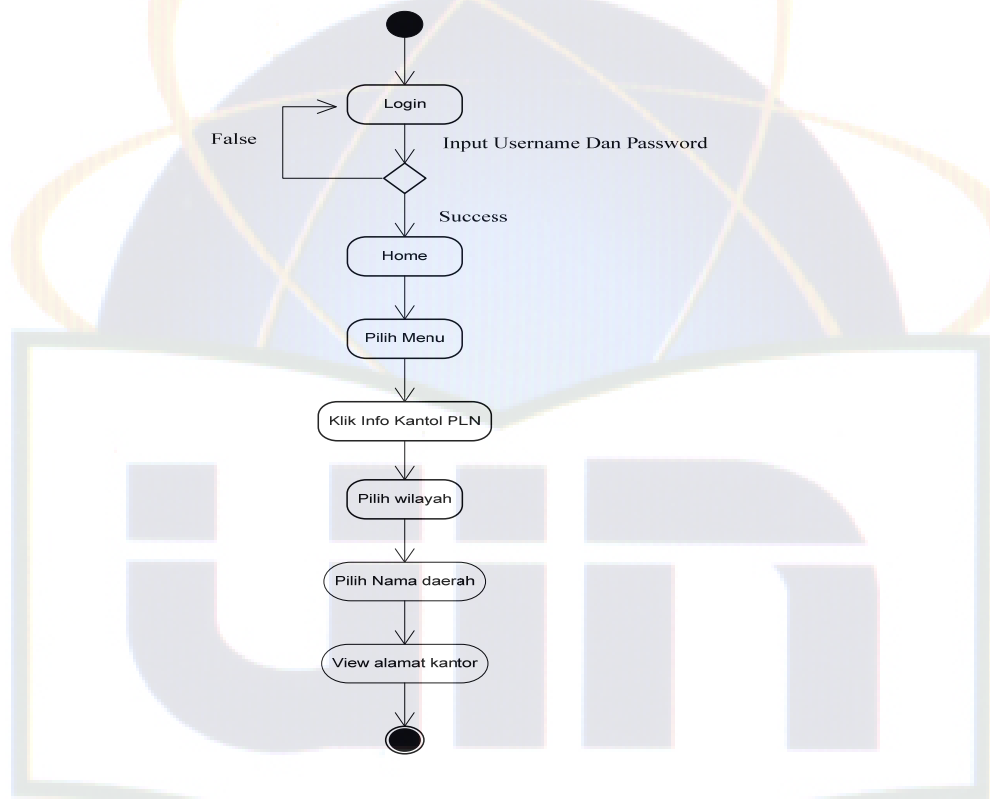
e. Activity Diagram Info Token Prabayar



Gambar 4.7 Activity Diagram Info Token Prabayar

Dalam *activity diagram* info token prabayar di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk melihat informasi keterangan pembelian token terakhir. Sebelumnya *user* harus melakukan *login* terlebih dahulu kedalam sistem dengan menginput *username* dan *password*, setelah berhasil melakukan *login* maka sistem akan menampilkan halaman *home*. Kemudian klik menu, untuk melihat daftar yang ada di menu lalu pilih info token, maka sistem akan menampilkan halaman informasi token. Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram* info token prabayar di atas.

f. *Activity Diagram* Info Kantor PLN

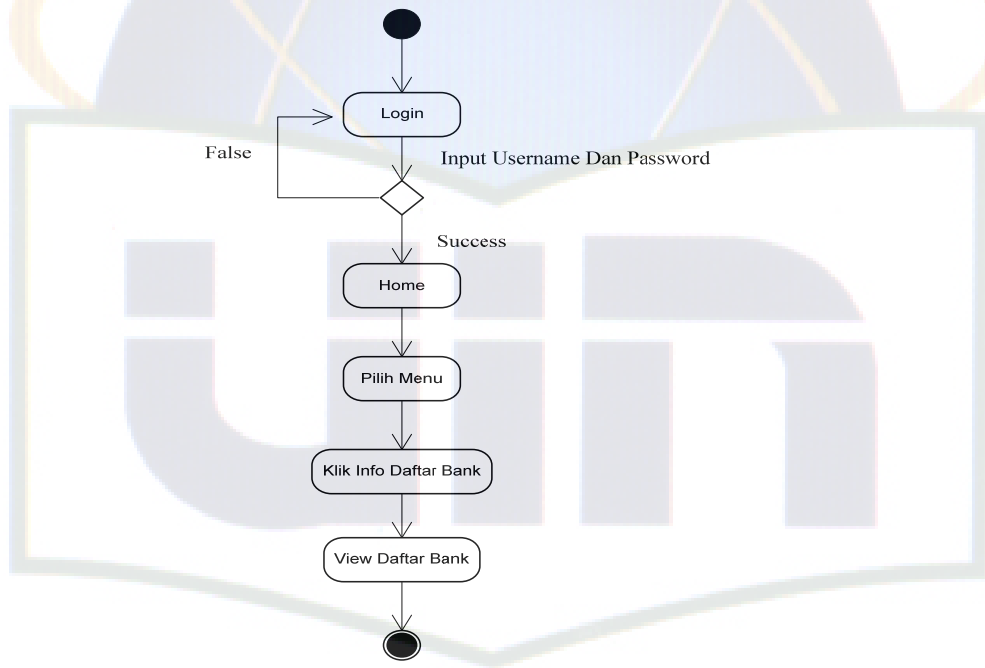


Gambar 4.8 *Activity Diagram* Info Kantor PLN

Dalam *activity diagram* info kantor PLN di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk melihat informasi alamat kantor PLN. Sebelumnya *user* harus melakukan *login* terlebih dahulu kedalam sistem dengan menginput *username* dan *password*, setelah berhasil melakukan *login* maka sistem akan menampilkan halaman *home*. Kemudian klik menu, untuk melihat daftar yang ada di menu lalu pilih kantor PLN, setelah itu pilih wilayah mana yang ingin dilihat kemudian pilih daerahnya, kemudian sistem akan menampilkan halaman informasi kantor PLN berdasarkan wilayah dan daerah

yang dipilih. Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram* info kantor PLN di atas.

g. Activity Diagram Info Daftar Bank

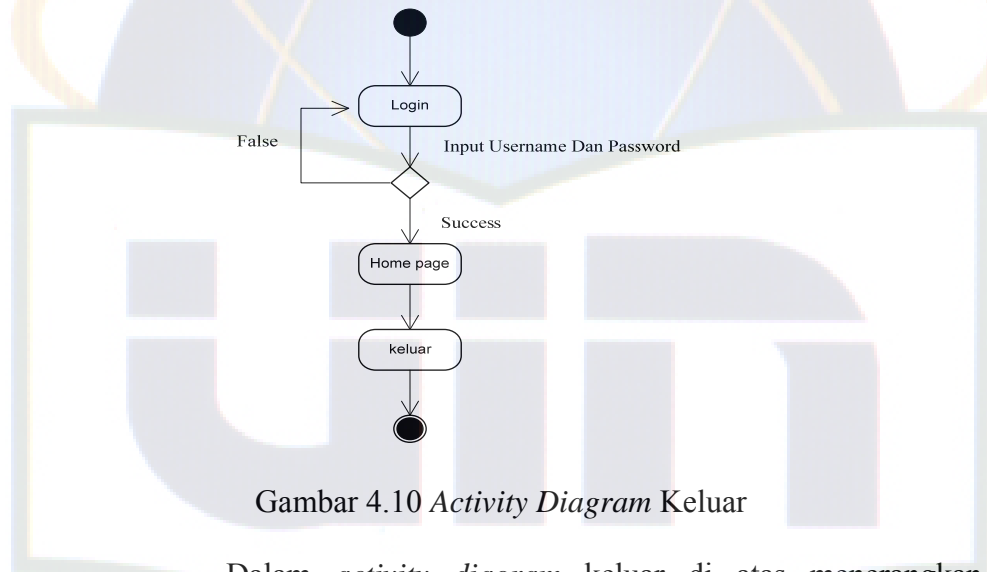


Gambar 4.9 Activity Diagram Info Daftar Bank

Dalam *activity diagram* info daftar Bank di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk melihat daftar Bank. Sebelumnya *user* harus melakukan *login* terlebih dahulu kedalam sistem dengan menginput *username* dan *password*, setelah berhasil melakukan *login* maka sistem akan menampilkan halaman *home*. Kemudian klik menu, untuk melihat daftar yang ada di menu lalu pilih info daftar Bank, maka sistem akan menampilkan halaman daftar Bank.

Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram* Info daftar Bank di atas.

h. Activity Diagram Keluar

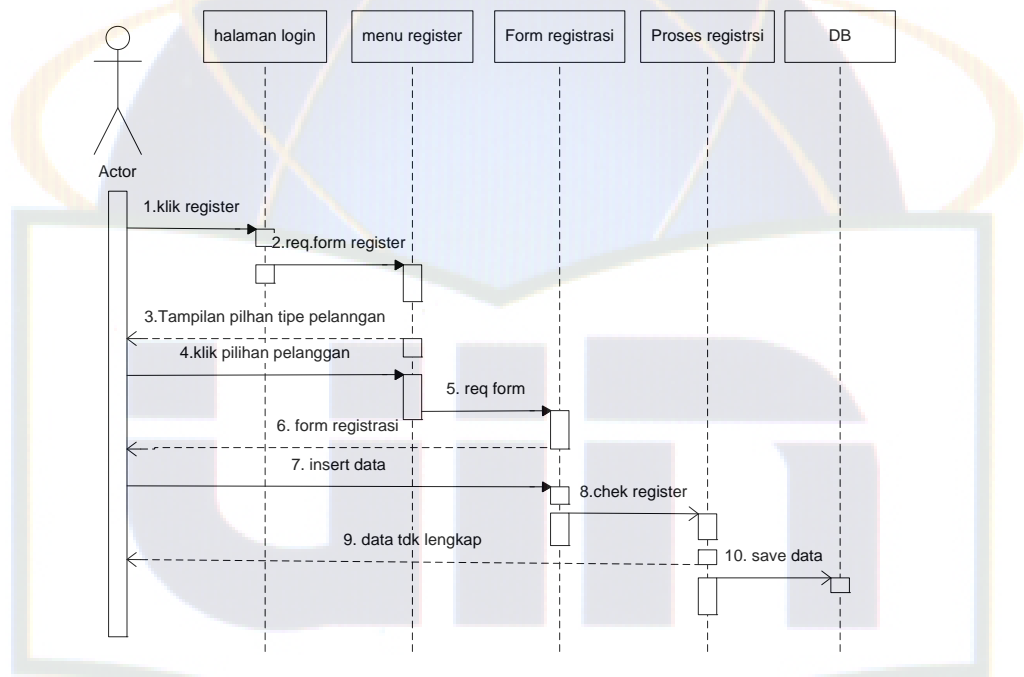


Gambar 4.10 *Activity Diagram* Keluar

Dalam *activity diagram* keluar di atas menerangkan serangkain aktifitas yang dilakukan oleh *user* untuk keluar dari aplikasi. Untuk lebih jelas langkah-langkah aktifitas ini bisa dilihat pada gambar *activity diagram view* keluar di atas.

5. Sequence Diagram

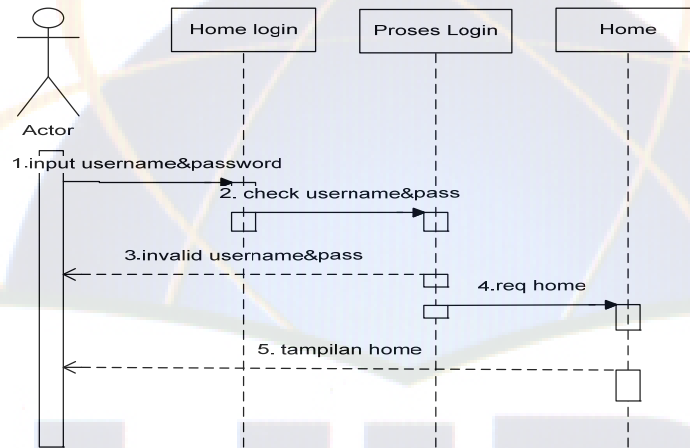
a. Sequence Diagram Register



Gambar 4.11 *Sequence Diagram Register*

Pada *sequence diagram register*, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses registrasi dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek *menu register* yang ditemani oleh kelas *home login*, *menu register*, *form registrasi*, proses registrasi, DB. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram register* diatas.

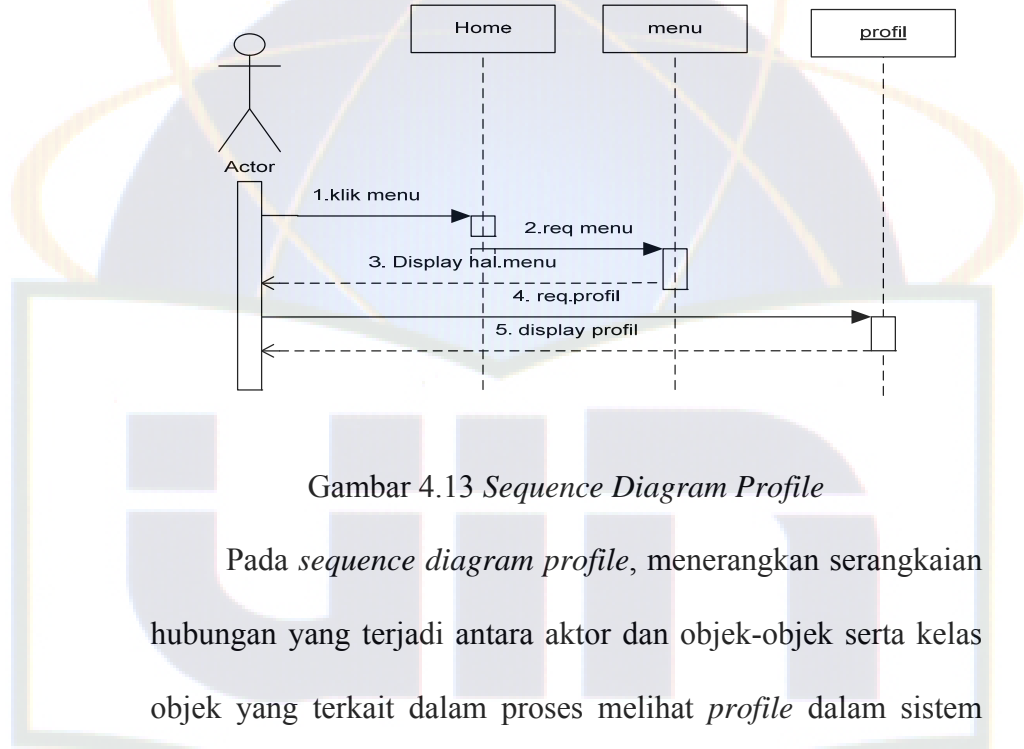
b. *Sequence Diagram Login*



Gambar 4.12 *Sequence Diagram Login*

Pada *sequence diagram login*, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait pada saat *login* terjadi. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek *user* yang ditemani oleh beberapa kelas sistem mulai dari *home login*, *proses login*. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram login* diatas.

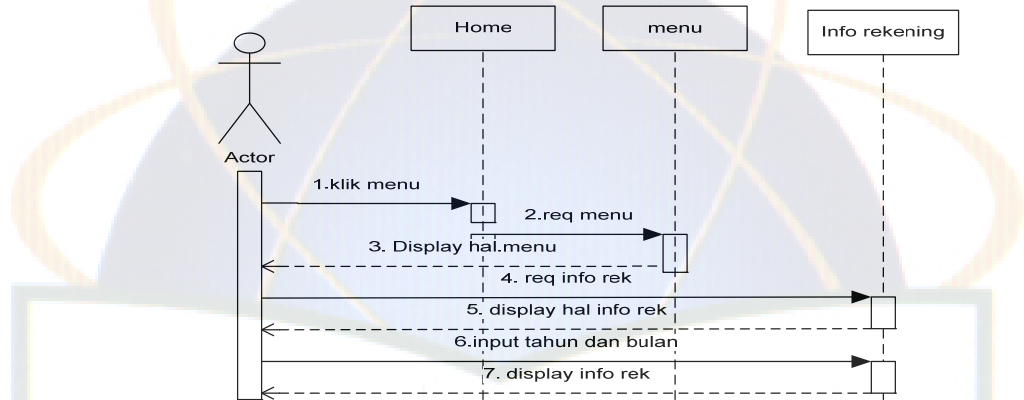
c. *Sequence Diagram Profile*



Gambar 4.13 *Sequence Diagram Profile*

Pada *sequence diagram profile*, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses melihat *profile* dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek profil yang ditemani oleh kelas *home* dan kelas *menu*. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram profile* diatas.

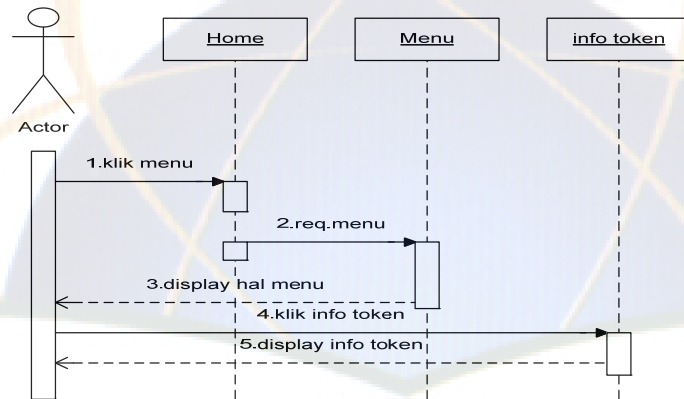
d. *Sequence Diagram* Info Rekening



Gambar 4.14 *Sequence Diagram* Info Rekening

Pada *sequence diagram* info rekening, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses *view* info rekening dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek info rekening yang ditemani oleh kelas *home* dan kelas *menu*. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram* info rekening diatas.

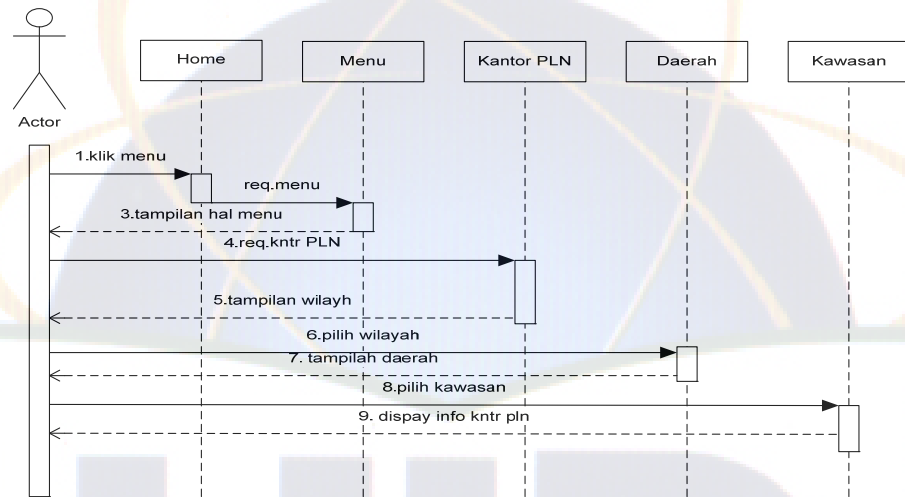
e. *Sequence Diagram Info Token Prabayar*



Gambar 4.15 *Sequence Diagram Info Token Prabayar*

Pada *sequence diagram* info token prabayar, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses *view* info token dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek info token yang ditemani oleh kelas *home* dan kelas menu. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram* info token prabayar diatas.

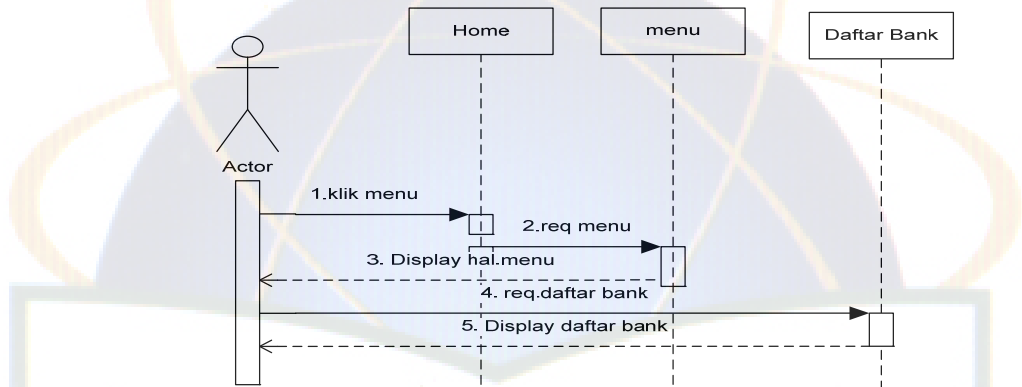
f. *Sequence Diagram* Info Kantor PLN



Gambar 4.16 *Sequence Diagram* Info Kantor PLN

Pada *sequence diagram* Info Kantor PLN, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses info Kantor PLN dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek Kantor PLN yang ditemani oleh *home*, *menu*, *daerah* dan *kawasan*. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram info Kantor PLN* diatas.

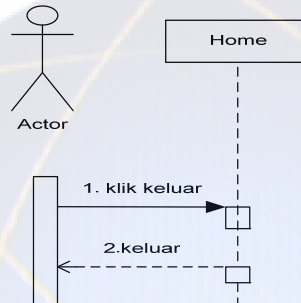
g. Sequence Diagram Info Daftar Bank



Gambar 4.17 *Sequence Diagram* Info Daftar Bank

Pada *sequence diagram* info daftar bank, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses melihat info daftar bank dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user* dan objek yang berinteraksi dengan aktor adalah objek daftar bank yang ditemani oleh kelas home dan menu. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram* info daftar bank diatas.

h. *Sequence Diagram* Keluar



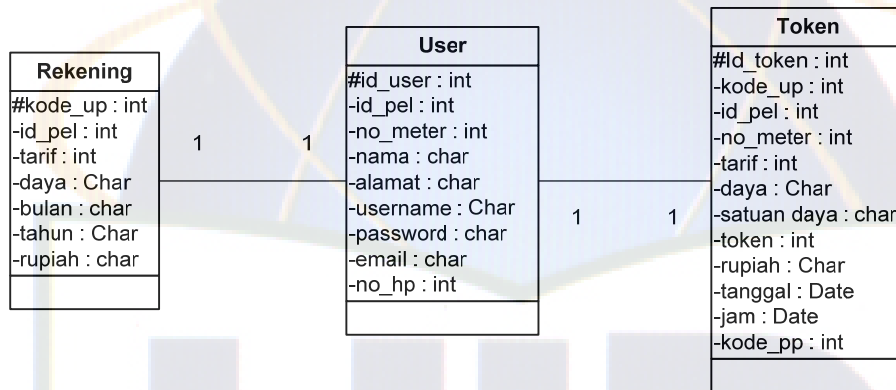
Gambar 4.18 *Sequence Diagram* Keluar

Pada *sequence diagram* keluar, menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek yang terkait dalam proses keluar dalam sistem aplikasi ini. Dalam *diagram* ini aktor yang terlibat adalah *user*. Sedangkan yang berinteraksi dengan aktor tersebut adalah *home login*. Adapun interaksi antara aktor dan objek beserta kelas tersebut dapat dilihat pada gambar *sequence diagram* keluar diatas.

6. *Class Diagram*

Class diagram ini digunakan untuk menggambarkan kumpulan dari *class* dan hubungannya. *Diagram* ini merupakan *diagram* yang paling umum ditemukan dalam pemodelan sistem berorientasi *object*. *Class* menggambarkan keadaan suatu sistem, sekaligus layanan untuk memanipulasi keadaan metode atau fungsi sehingga *class* memiliki tiga area pokok, yaitu: nama, atribut, dan metode. Selaian itu setiap *class* yang ada dapat

menjadi sebuah *form* saat pembuatan program. *Class diagram* sistem yang diusulkan dapat dilihat pada gambar berikut.



Gambar 4.19 *Class diagram*

4.2.2.2 Desain Basis Data

Pada *database* ini semua jenis data yang terlibat dalam proses yang terjadi, didefinisikan dan dikumpulkan dalam bentuk penyajian sebagai berikut :

1. Tabel *User*

Tabel Name : *user*

Primary Key : *id_user*

Foreign Key : *id_pel, no_meter*

Tabel 4.10 Tabel User

No.	Field	Type	Extra
1.	id_user	Int(10)	Auto_increment
2.	id_pel	Int(10)	
3.	No_meter	Int(25)	
4.	nama	Varchar(100)	
5.	alamat	Varchar(100)	

6.	username	Varchar(100)	
7.	password	Varchar(100)	
8.	email	Varchar(100)	
9.	No_hp	Int (25)	

2. Tabel Rekening

Tabel Name : Rekening

Primary Key : kode_up

Foreign Key : id_pel

Tabel 4.11 Tabel Rekening

No.	Field	Type	Extra
1.	Kode_up	Int(25)	Auto_increment
2.	id_pel	Int(10)	
3.	Tarif	Varchar(25)	
4.	Daya	Varchar(10)	
5.	Bulan	Varchar(100)	
6.	Tahun	Varchar(100)	
7.	rupiah	int(100)	

3. Tabel Token

Tabel Name : Token

Primary Key : id_token

Foreign Key : no_meter, id_pel

Tabel 4.12 Tabel Token

No.	Field	Type	Extra
1.	Id_token	Int(50)	Auto_increment

2.	kode_up	Int(25)	
3.	id_pel	Int(10)	
4.	No_meter	Int(25)	
5.	tarif	int(100)	
6.	daya	Varchar(100)	
7.	Satuan daya	Varchar(10)	
8.	token	Varchar(100)	
9.	rupiah	int (100)	
10.	tanggal	date	
11.	jam	date	
12.	Kode_pp	int	

4.2.2.3 Desain Interface

1. Design User Interface Form splash screen

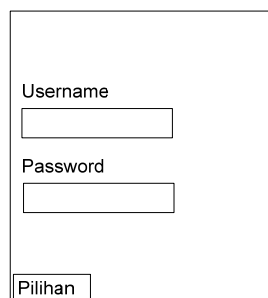


Gambar splash screen

Info Rekening dan Token

Gambar 4.20 Design User Interface Form splash screen

2. Design User Interface Form Login



Username

Password

Pilihan

Gambar 4.21 Design User Interface Form Login

3. *Design User Interface Form Home*

The diagram shows a rectangular container representing the Home form. Inside, there is a smaller rectangle labeled "Gambar splash screen". Below this, the text "Info Rekening dan Token" is displayed. At the bottom left of the container is a button labeled "Pilihan".

Gambar 4.22 *Design User Interface Form Home*

4. *Design User Interface Form Menu*

The diagram shows a rectangular container representing the Menu form. At the top, the word "Menu" is centered. Below it is a vertical list of five buttons: "profile", "Info Rekening", "Info Token Prabayar", "Info Kantor PLN", and "Info Daftar Bank". At the bottom left of the container is a button labeled "Kembali".

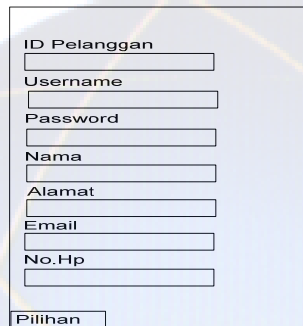
Gambar 4.23 *Design User Interface Form Menu*

5. *Design User Interface Form Register*

The diagram shows a rectangular container representing the Register form. Inside, there are two buttons: "Pasca Bayar" and "Prabayar". At the bottom left of the container is a button labeled "Kembali".

Gambar 4.24 *Design User Interface Form Register*

6. *Design User Interface Form Register Pasca Bayar*



ID Pelanggan

Username

Password

Nama

Alamat

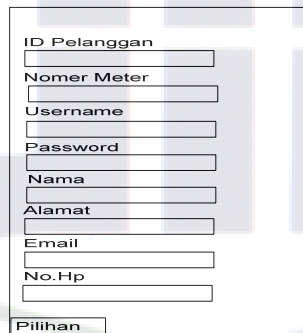
Email

No.Hp

Pilihan

Gambar 4.25 *Design User Interface Form Register PascaBayar*

7. *Design User Interface Form register Pra Bayar*



ID Pelanggan

Nomer Meter

Username

Password

Nama

Alamat

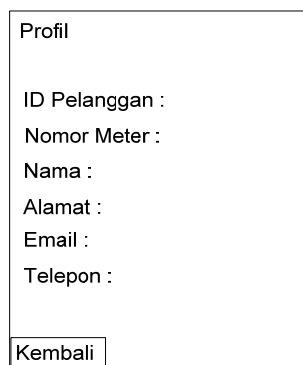
Email

No.Hp

Pilihan

Gambar 4.26 *Design User Interface Form Register Pra Bayar*

8. *Design User Interface Form Profile Pasca Bayar*



Profil

ID Pelanggan :
Nomor Meter :
Nama :
Alamat :
Email :
Telepon :

Kembali

Gambar 4.27 *Design User Interface Form Profile Pasca Bayar*

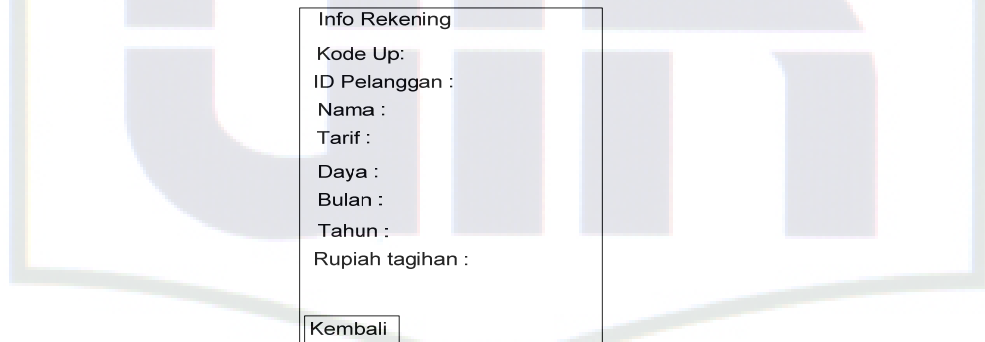
9. Design User Interface Form Profile Prabayar



Profil
Pelanggan Prepaid
ID Pelanggan :
Nomor Meter :
Nama :
Alamat :
Email :
Telepon :
Kembali

Gambar 4.28 *Design User Interface Form Profile Prabayar*

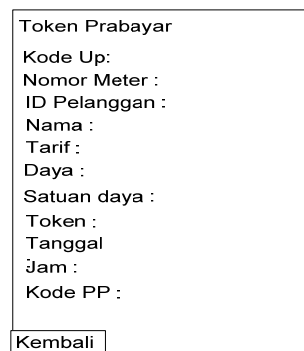
10. Design User Interface Form Info Rekening



Info Rekening
Kode Up:
ID Pelanggan :
Nama :
Tarif :
Daya :
Bulan :
Tahun :
Rupiah tagihan :
Kembali

Gambar 4.29 *Design User Interface Form Info Rekening*

11. Design User Interface Form Info Token Prabayar

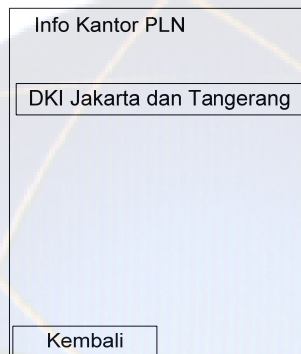


Token Prabayar
Kode Up:
Nomor Meter :
ID Pelanggan :
Nama :
Tarif :
Daya :
Satuan daya :
Token :
Tanggal
Jam :
Kode PP :
Kembali

Gambar 4.30 *Design User Interface Form Info Token*

Prabayar

12. Design User Interface Form Info Kantor PLN



Info Kantor PLN

DKI Jakarta dan Tangerang

Kembali

Gambar 4.31 *Design User Interface Form Info Kantor PLN*

13. Design User Interface Form Info Kantor PLN Jakarta dan Tangerang



DKI Jakarta dan Tangerang

Daftar nama Wilayah

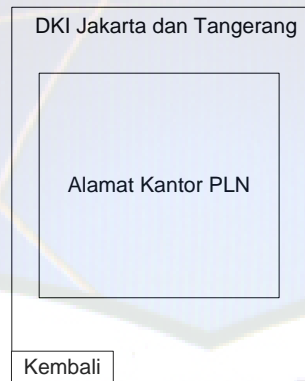
Kembali

Gambar 4.32 *Design User Interface Form Info Kantor PLN*

Jakarta dan Tangerang

14. *Design User Interface Form Info Alamat Kantor PLN*

Jakarta dan Tangerang



DKI Jakarta dan Tangerang

Alamat Kantor PLN

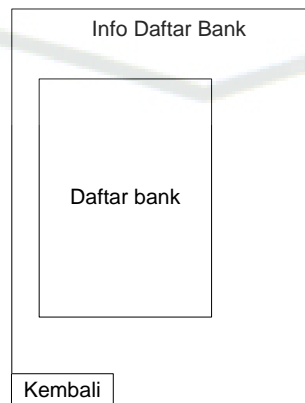
Kembali

The image shows a user interface design for a form titled 'DKI Jakarta dan Tangerang'. It features a light blue rectangular area labeled 'Alamat Kantor PLN' and a 'Kembali' button at the bottom.

Gambar 4.33 *Design User Interface Form Info Alamat*

Kantor PLN Jakarta Raya dan Tangerang

15. *Design User Interface Form Info Daftar Bank*



Info Daftar Bank

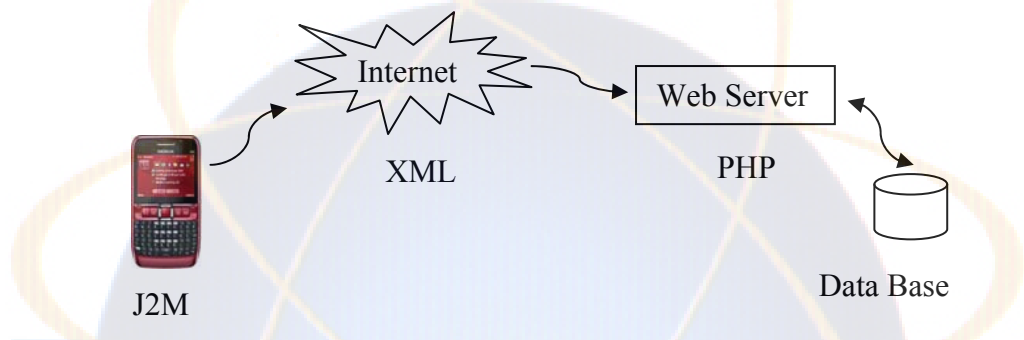
Daftar bank

Kembali

The image shows a user interface design for a form titled 'Info Daftar Bank'. It features a light blue rectangular area labeled 'Daftar bank' and a 'Kembali' button at the bottom.

Gambar 4.34 *Design User Interface Form Info Daftar Bank*

4.2.2.4 Desain Alur Komunikasi Data



Gambar 4.35 Alur komunikasi data

Pada Gambar diatas menjelaskan alur komunikasi data aplikasi info rekening listrik PLN, terlebih dahulu *user* menginstal aplikasi tersebut pada *handphone* yang berbasis *java*. Aplikasi tersebut dikirim melalui *bluetooth*, kabel data atau *card reader*. Proses instalasi terdapat pada halaman lampiran.

Alur Komunikasi data dimulai ketika *client* ingin *merequest* informasi yang dibutuhkan. Maka, program yang ada di *handphone* akan mengirimkan data yang berformat XML ke *server* melalui *web service*, dimana XML merupakan sebuah bahasa untuk mendefinisikan atau menterjemahkan data. Kemudian *server* akan menerima *request* dan akan memproses serta menampilkan informasi yang dibutuhkan oleh *client*.

4.2.2.5 Pengkodean Sistem

1. Implementasi Coding

Pada tahap ini, penulis melakukan tahap pengkodean terhadap hasil rancangan yang sudah didefinisikan sebelumnya untuk dijadikan program aplikasi. pembuatan aplikasi menggunakan beberapa *tools* atau *software* antara lain J2SE JDK versi 1.6.0 sebagai *platform* java, *emulator java wireless toolkit* versi 2.5.2 untuk menjalankan aplikasi di PC, NetBeans IDE versi 6.0 untuk membuat *user interface* dan *source code*, *Dreamweaver CS3* untuk membuat *source code* php dan *Adobe Photoshop CS3* yang digunakan sebagai *software* tambahan dan untuk merancang tampilan *splash screen* dan *icon*. Untuk proses konstruksi program dan *source code* program dapat dilihat selengkapnya pada lampiran.

2. Kelengkapan Software dan Hardware

I. Kelengkapan Perangkat Keras (*Hardware*)

- a. Satu buah PC untuk membuat program dengan spesifikasi sebagai berikut:
 - 1) *Processor* 2.26 Ghz
 - 2) *RAM / Memory* 768 Mb
 - 3) *Hardisk* 20 GB

4) *Keyboard* dan *mouse*

5) Monitor dengan resolusi minimal 1024 x 768 pixel.

b. Sebuah kabel data, *Bluetooth* dan *card Reader* untuk mentrasfer file aplikasi dari PC ke *mobile device*.

c. Ponsel atau *mobile device* yang telah mendukung java MIDP versi 2.0. Dalam hal ini penulis menggunakan perangkat ponsel Nokia E63 untuk pengujian aplikasi.

II. Kelengkapan Perangkat Lunak (Software)

a. J2SE JDK Versi 1.6.0 sebagai *platform* java yang membuat *wireless toolkit* dapat berjalan.

b. J2ME *wireless toolkit* versi 2.5.2 sebagai emulator untuk aplikasi MIDlet.

c. Netbeans IDE versi 6.0 sebagai *integrated Development Environment* (IDE) untuk merancang *user interface* dan *source code*.

d. Xampp versi 1.6.6a untuk pengkoneksian *Database*.

e. *Dreamweaver CS3* sebagai *software* tambahan untuk merancang *source code* PHP.

f. Sistem operasi Windows Seven Ultimate yang berjalan pada PC (*personal computer*).

4.2.3 Pengujian Prototype

Pada tahap ini, dilakukan pengujian prototype, sebelumnya penulis menganalisis spesifikasi kebutuhan ponsel yang dapat digunakan untuk menerapkan aplikasi ini, pengujian aplikasi pada ponsel dengan syarat-syarat yang telah terpenuhi, yaitu Nokia E63, untuk melakukan kinerja aplikasi pada ponsel tersebut.

4.2.3.1 Spesifikasi Ponsel yang digunakan

Ponsel yang digunakan untuk menjalankan aplikasi ini harus memiliki spesifikasi minimal sebagai berikut :

1. Mendukung aplikasi java MIDP 2.0
2. Memiliki memori minimal 256 Kb atau lebih untuk proses instalasi.

Dalam pengimplementasian aplikasi yang dilakukan oleh penulis, ponsel yang digunakan adalah Nokia E63. Ponsel tersebut telah memenuhi kebutuhan spesifikasi minimal ponsel yang digunakan yang telah disebutkan di atas.

Berikut ini adalah tabel spesifikasi ponsel yang digunakan dalam pengujian aplikasi.

Tabel 4.13 Spesifikasi Ponsel yang digunakan

<i>Merk</i> dan tipe Ponsel	Nokia E63
Jaringan	Symbian series 60
Pengiriman data	GPRS 10
<i>Browser</i>	<i>Web browser. Wap 2.0</i>

Java	Java MIDP 2.0
Konektifitas	<i>Bluetooth</i> , USB, Wifi
Memori <i>Eksternal</i>	2 Gb
Memori <i>Internal</i>	22

4.2.3.2 Transfer aplikasi dari PC kedalam ponsel

Untuk mendistribusikan aplikasi ke dalam ponsel saat ini dapat menggunakan beberapa *tools* diantaranya *bluetooth*, *card reader* dan kabel data. Dalam hal ini, penulis menggunakan kabel data untuk *transfer* aplikasi ke dalam ponsel Nokia E63.

Hasil dari proses instalasi serta hasil seluruh tampilan aplikasi pada ponsel dapat dilihat selengkapnya pada lampiran.

4.2.3.3 Pengujian aplikasi pada ponsel Nokia E63

Pada bagian ini akan dibahas pengujian terhadap perancangan aplikasi Info rekening listrik PLN Berbasis *Mobile* yang telah dibuat. Pengujian aplikasi dengan menggunakan teknik pengujian “*Black Box testing*”. Cara pengujiannya hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan yang diinginkan.

Setelah dilakukan *transfer* aplikasi dari PC ke dalam ponsel dan instalasi program maka akan dilakukan pengujian program pada ponsel. Pengujian dilakukan dengan menguji

seluruh fitur yang ada pada aplikasi ini. Fitur-fitur yang di uji seperti fitur *login*, Registrasi, *Profile*, Info Rekening, Info Token Prabayar, Info Alamat Kantor PLN dan Info Daftar Bank.

Tabel 4.14 Hasil Pengujian *Black Box*

No	Nama tes	Hasil yang diharapkan	Hasil	Keterangan
1.	<i>Login</i>	<i>User</i> dapat masuk kedalam sistem	OK	Lihat lampiran 1, No.2
2.	Registrasi	<i>User</i> berhasil melakukan registrasi	OK	Lihat lampiran 1, No.3-5
3.	<i>Profile</i>	<i>User</i> dapat melihat informasi <i>profile</i> pada aplikasi	OK	Lihat lampiran 1, No.8-9
4.	Info rekening	<i>User</i> dapat melihat informasi rekening dalam aplikasi	OK	Lihat lampiran 1, No.10
5.	Info Token Prabayar	<i>User</i> dapat melihat informasi pembelian token terakhir dalam aplikasi	OK	Lihat lampiran 1, No.11
6	Alamat Kantor PLN	<i>User</i> dapat melihat informasi rekening dalam aplikasi	OK	Lihat lampiran 1, No.12
7.	Daftar nama Bank	<i>User</i> dapat melihat daftar Bank dalam aplikasi	OK	Lihat lampiran 1, No.13
8.	Keluar	<i>User</i> dapat keluar dari aplikasi	OK	Lihat lampiran 1, No.14

4.2.3.4 Evaluasi sistem

Pada tahap ini dilakukan evaluasi untuk mendapatkan tanggapan dari pelanggan untuk mengetahui apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Setelah melakukan kuesioner dan melihat hasil maka dapat disimpulkan bahwa aplikasi info rekening listrik PLN berbasis *mobile* telah sesuai dengan yang diharapkan. Hasil kuesioner dapat dilihat pada lampiran V.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan penelitian ini penulis dapat menyimpulkan, bahwasanya:

1. Aplikasi info rekening listrik PLN berbasis *mobile* ini, dapat memudahkan pelanggan dalam memperoleh informasi seputar rekening listrik, dapat dilihat di lampiran V pada tabel kuesioner.
2. Aplikasi ini menunjang media pelayanan pelanggan yang ada melalui perangkat *mobile*, dapat dilihat di lampiran V pada tabel kuesioner.
3. Aplikasi ini dirancang dengan metode pengembangan sistem model prototype dengan perancangan aplikasi dan alur proses sistem yang menggunakan diagram UML, dapat dilihat pada bab IV.

5.2 Saran

Setelah menganalisa dan mengimplementasikan sistem ini, didapatkan saran untuk pengembangan lebih lanjut mengenai aplikasi secara teknis dan bisnis yaitu :

1. Aplikasi info rekening berbasis *mobile* dapat dikembangkan lebih lanjut dengan menggunakan *operating system* android serta ditambahkan fitur-fitur pelayanan pelanggan lainnya seperti adanya fitur informasi tarif daya listrik dan fitur *search* informasi dalam aplikasi tersebut.

DAFTAR PUSTAKA

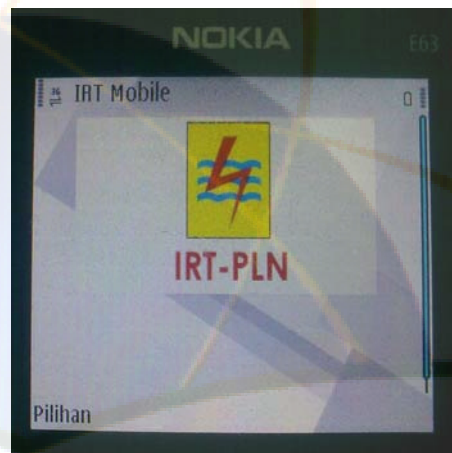
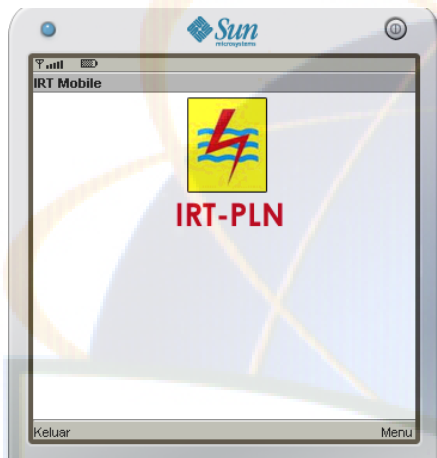
- Elrom, Elad, Janousek, Scott & Joss, Thomas. 2009. *Advanced Flash on Devices Mobile Development with Flash Lite and Flash 10*. Berkeley: Friends of ED.
- Faruq Tataran. 2010. *Aplikasi Panduan Kata dalam Mencari Ayat Al-Quran Juz 30 Berbasis Java Mobile*. UIN Syarif Hidayatullah Jakarta.
- Hidayat Mochammad Yasin. 2011. *Rancang Bangun Sistem Informasi Akademik dengan Metode Prototype di MA Ponpes Amanatul Ummah*. Universitas Teknologi Sepuluh November.
- Hermawan, Julius. 2004. *Analisa Desain dan Pemograman Berorientasi Objek dengan UML di Visual Basic Net*. Yogyakarta : Andi.
- Husni. 2007. *Pemograman Database Berbasis Web*. Yogyakarta: Graha Ilmu.
- Imm Imamudin. 2010. *Prototype Aplikasi Media Pembelajaran Rumah Adat Berbasis Multimedia*. UIN Syarif Hidayatullah Jakarta.
- Jogianto, M.H. 2005. *Pengenalan Komputer, Dasar Ilmu Komputer, pemograman Sistem dan Intelegensi Buatan*. Yogyakarta: Andi.
- Kadir, Abdul. 2003. *Pengenalan sistem informasi*. Yogyakarta: Andi.
- Ladjamudin, Al-bahra bin. 2006. *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- Muhammad Djupri. 2010. *Virtual Reality Prototype Perumahan Berbasis Multimedia 3 Dimensi*. UIN Syarif Hidayatullah Jakarta.

- Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak Pendekatan praktisi*". Yogyakarta : Andi.
- Paranginangin, Kasiman. 2006. *Aplikasi web dengan PHP dan MySQL*. Yogyakarta: Andi
- Purwono, Edi. 2002. *Apa yang Harus Diketahui Oleh Sistem Analis*. Yogyakarta: Andi
- Raharjo, Budi, et.al. 2007. *Tuntunan Pemograman Java untuk Handphone*. Bandung : Informatika Bandung.
- Rudi, R. 2009. *Game ponsel paling populer*. Yogyakarta: media.kom
- Sari, Juni Nurma. 2007. *Pembuatan prototype aplikasi wireless menggunakan teknologi java pada sistem informasi akademik politeknik Caltex Riau*. Universitas Teknologi Sepuluh November
- Satya Bestari. 2010. *Aplikasi Translation Pada Telepon Seluler Menggunakan J2ME*. UIN Syarif Hidayatullah Jakarta.
- Shalahuddin,M dan A.S, Rosa. 2006. *Pemograman J2ME: Belajar Cepat Pemograman perangkat Mobile*. Bandung :Informatika Bandung.
- Setiawan, Dee. 2008. *Koneksi Internet Menggunakan Perangkat GSM + CDMA*. Yogyakarta: Andi.
- Siregar, Ivan Michael. 2007. *Membangun Aplikasi Chat Lewat GPRS dengan J2ME Menggunakan Netbeans IDE 5.0*. Yogyakarta: Gava Media.

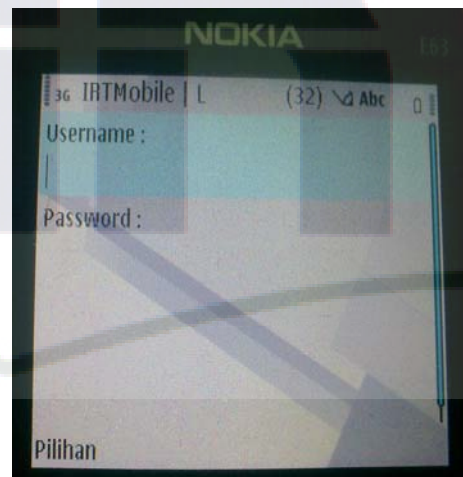
- Subana, H.M dan Sudrajat. 2005. *Dasar-Dasar Penelitian Ilmiah*. Bandung: CV. Pustaka Setia.
- Sugiarto, A.2007. *Fotografi Posel Murah, Mudah, Indah*. Jakarta: PT Gramedia Pustaka Ptama.
- Supardi, Yunair. 2005. *36 jam Belajar Komputer Pemograman J2SE SDK 1.4*. Jakarta: PT.Elex Media Komputindo.
- Suprianto, Dodit. 2008. *Buku Pintar Pemrograman PHP*. Bandung: OASE Media
- Suryana, M. 2008. *Arsitektur Mobile Application dengan Platform Proprietary*. [Http://informatika.lipi.go.id/publikasi/arsitektur-mobile-application-dengan-platfrom-propietary.html](http://informatika.lipi.go.id/publikasi/arsitektur-mobile-application-dengan-platfrom-propietary.html), 8 juni 2011, 15.00 WIB.
- Syukrina Irfiyanda. 2009. *Analisis dan Implementasi Informasi pembayaran Rekening Air berbasis Mobile*. UIN Syarif Hidayatullah Jakarta.
- Wahid, Fathul. 2005. *Kamus Istilah Teknologi Informasi*. Yogyakarta: Andi.
- Witarto. 2004. *Memahami sistem informasi pendekatan praktis rekayasa informasi melalui kasus-kasus sistem informasi di sekitar kita*. Bandung: informatika.
- Whitten, J.L. Bentley, L.D. & Dittman, K.C. *Metode Desain dan Analisis Sistem*. Yogyakarta: Andi



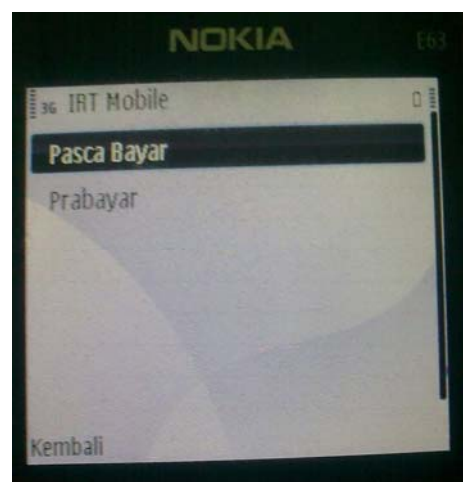
1. Tampilan Splash Screen IRT



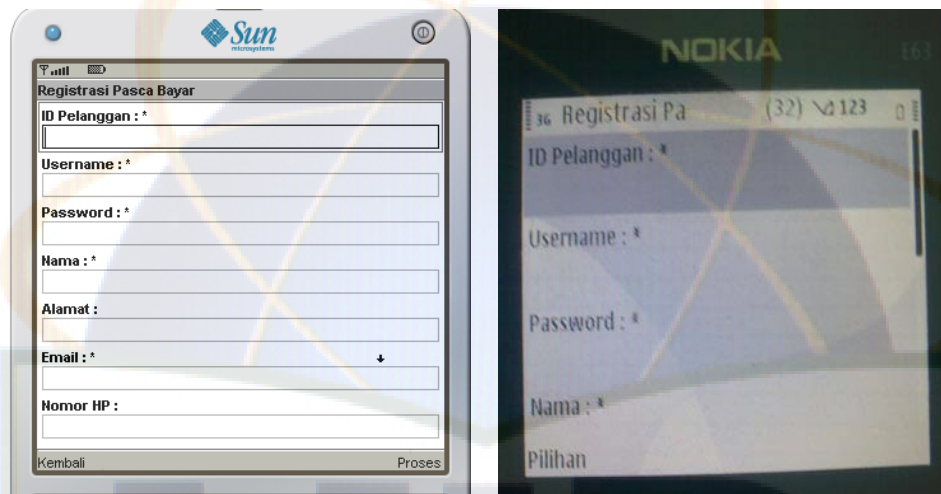
2. Tampilan Halaman Login



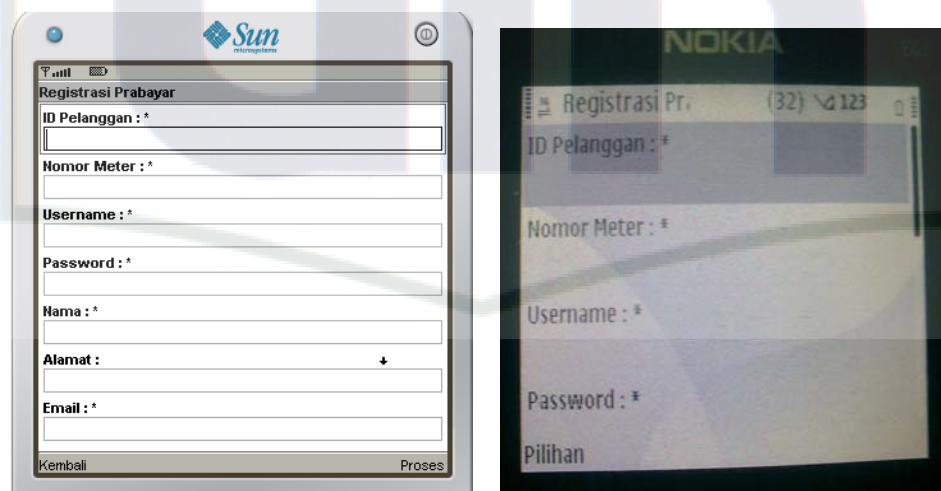
3. Tampilan Halaman Register



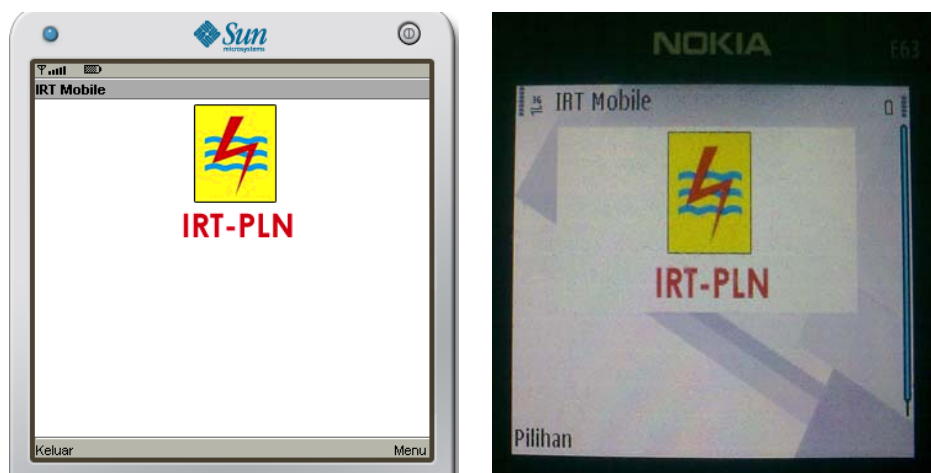
4. Tampilan Halaman Register Pascabayar



5. Tampilan Halaman Register Prabayar



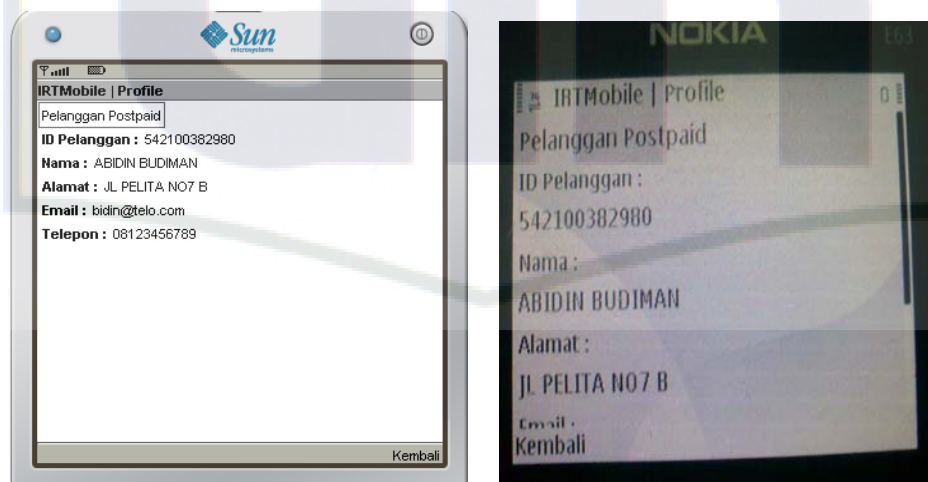
6. Tampilan Halaman Home



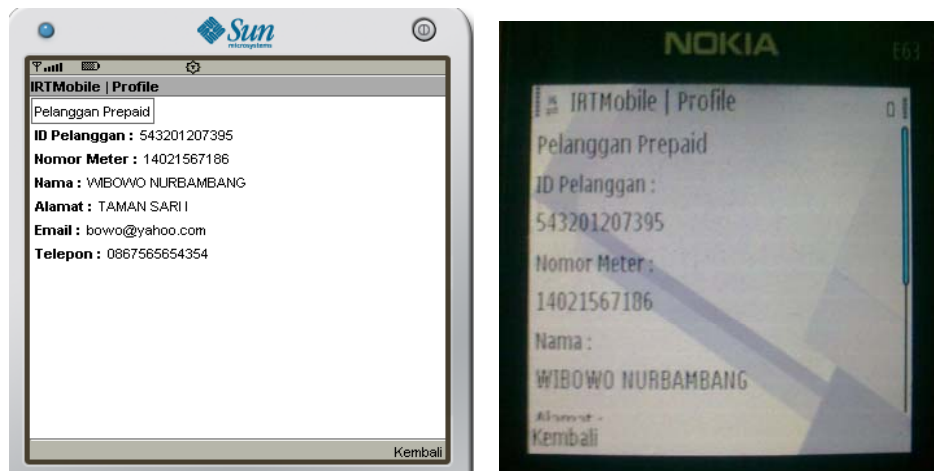
7. Tampilan Halaman Menu



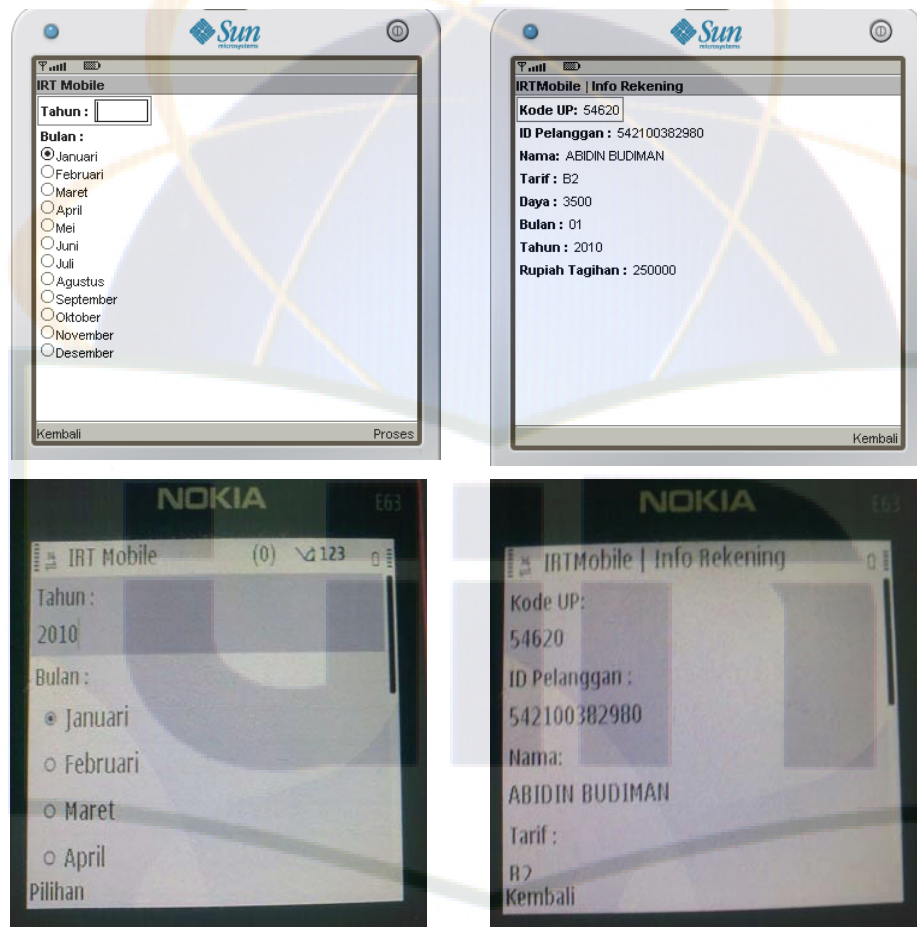
8. Tampilan Halaman Profile Pascabayar



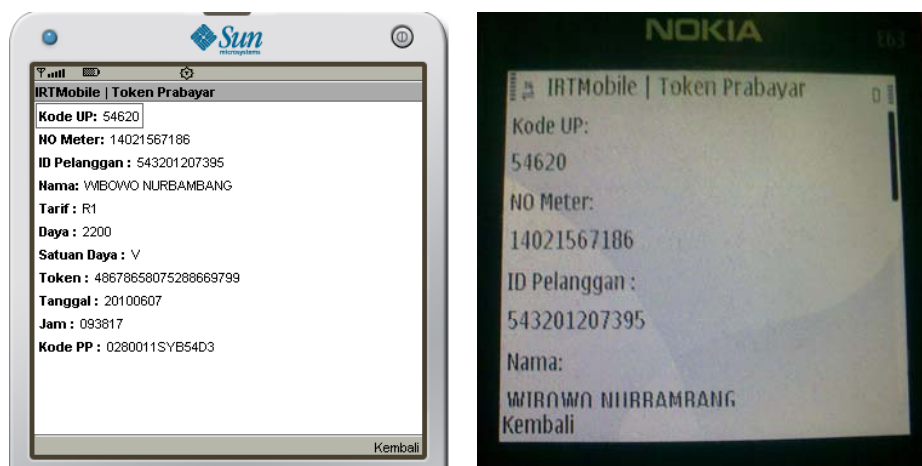
9. Tampilan Halaman Profile Prabayar



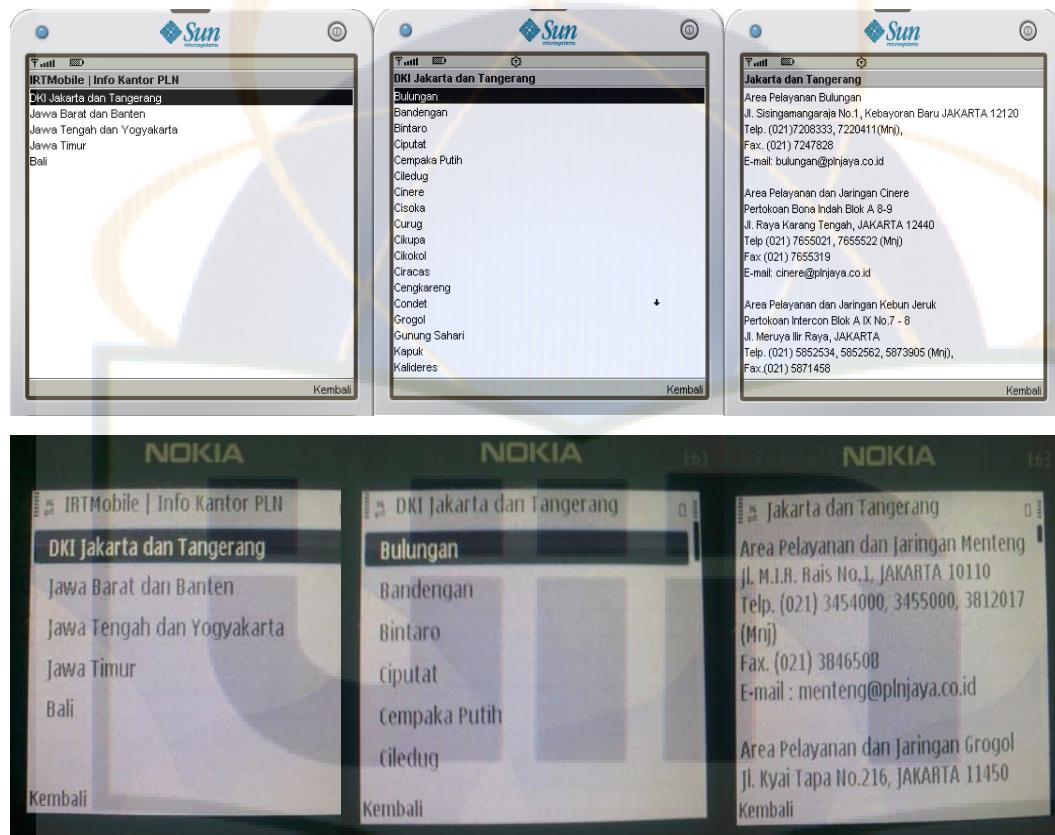
10. Tampilan Halaman Info Rekening



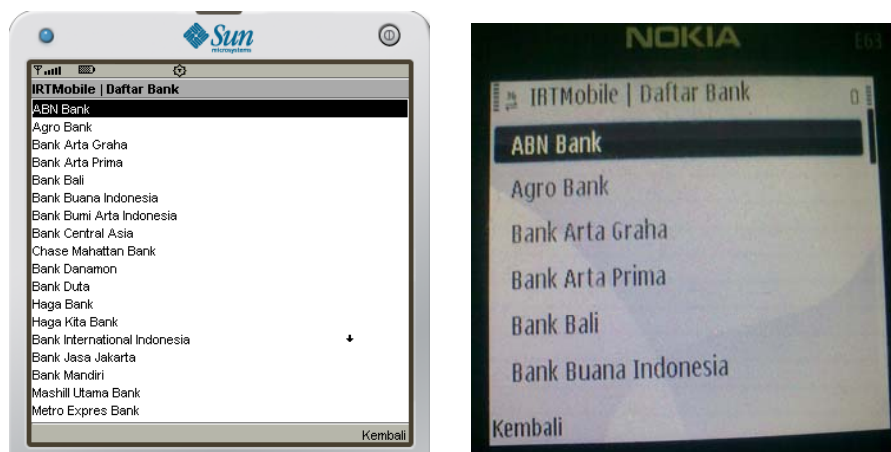
11. Tampilan Halaman Info Token Prabayar



12. Tampilan Halaman Info Kantor PLN



13. Tampilan Halaman Info Daftar Bank





LAMPIRAN II
Proses Kontruksi Program

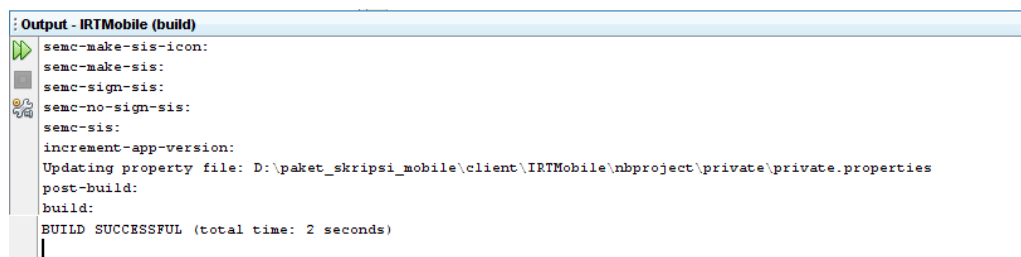
Proses kontruksi program akan diuraikan sebagai berikut :

1. Instalasi J2SE SDK versi 1.6.0
2. Instalasi J2ME *Wireless Toolkit* versi 2.5
3. Instalasi Netbeans IDE versi 6.0
4. Mengaktifkan *software* Netbeans 6.0

Prosesnya klik tombol *start* -> program -> netbeans 6.0 -> netbeans IDE, kemudian pilih menu *file* -> *new project*, pilih *categories mobile* -> *mobile application* -> *next*, kemudian isikan *project name* “IRT” dan pilih *project location*, dan tandai pilihan *set as main project* dan klik tombol *next*. Kemudian pilih *emulator platform J2ME Wireless Toolkit* versi 2.5, dan pilih *device default color phone*, pilih *device configuration* CLCD-1.0 dan pilih *device profile* MIDP-2.0, kemudian klik tombol *finish*.

5. Menjalankan program aplikasi dalam PC

caranya, pilih menu *build* -> *build main project* atau tekan tombol f11 pada *keyboard*, proses *build* dapat dilihat pada gambar berikut



```
Output - IRTMobile (build)
semc-make-sis-icon:
semc-make-sis:
semc-sign-sis:
semc-no-sign-sis:
semc-sis:
increment-app-version:
Updating property file: D:\paket_skripsi_mobile\client\IRTMobile\nbproject\private\private.properties
post-build:
build:
BUILD SUCCESSFUL (total time: 2 seconds)
```

Tampilan dalam proses *build*

setelah melakukan proses *build*, lalu pilih menu *run*, *run main project* atau tekan tombol f6 pada *keyboard* proses *run* dapat dilihat pada gambar berikut,

```
Output - IRTMobile (run)
jar:
pre-run:
cldc-run:
Copying 1 file to D:\paket_skripsi_mobile\client\IRTMobile\dist\mbrum53172
Copying 1 file to D:\paket_skripsi_mobile\client\IRTMobile\dist\mbrum53172
Jad URL for OTA execution: http://localhost:8082/serviet/org.netbeans.modules.mobility.project.jam.JAMServlet/D43A/paket_skripsi_mobile/client/IRTMobile/dist//IRTSmartphone.jad
Starting emulator in execution mode
Running with storage root C:\Users\user\j2se\2.5.2\appdb\QertyDevice
Running with locale: English_United States.1252
Running in the identified_third_party security domain
```

Tampilan setelah berhasil di *run*

setelah proses *run*, aplikasi akan muncul pada emulator J2ME WTK 2.5 yg dapat dilihat selengkapnya pada lampiran 1.

6. Program yang sudah dibuat akan dilakukan proses *build* sudah otomatis menjadi *file*. jar yang nantinya akan di distribusikan ke dalam ponsel. *file* .jar tersebut terdapat pada direktori irt8/dist/irt.jar

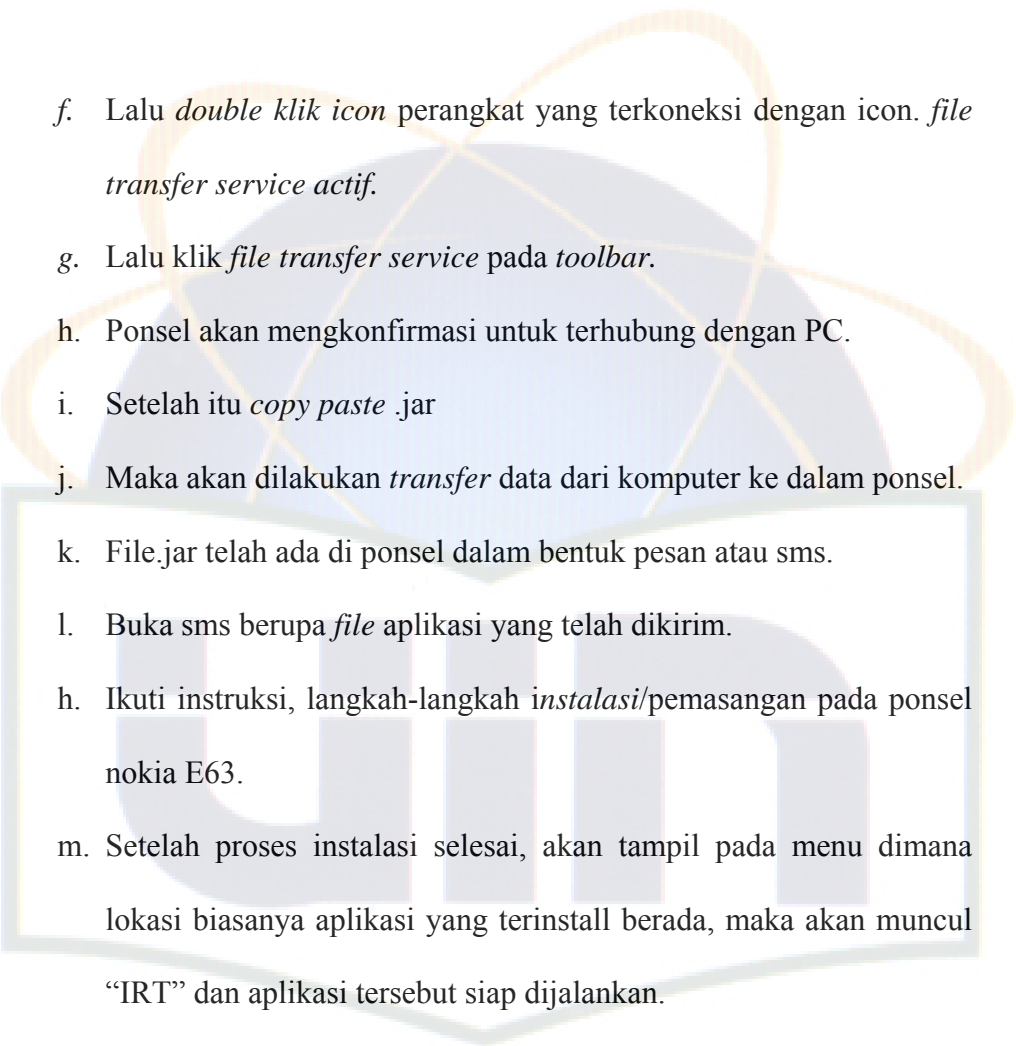


1. Berikut ini adalah langkah-langkah dalam melakukan *transfer* aplikasi dari PC ke dalam ponsel Nokia E63 menggunakan kabel data.

- a. Instalasi program nokia PC Suite.
- b. Pasang kabel data pada *port usb* pada PC.
- c. Pasang kabel data pada *port* kabel data yang ada pada ponsel.
- d. Jalankan program Nokia PC Suite.
- e. Pilih menu "*install app*" -> pilih *file* "*irt*" pada *direktori file* tersebut -> klik menu *install* yang bergambar tanda panah.
- f. Ikuti instruksi, langkah-langkah instalasi / pemasangan pada ponsel Nokia E63.
- g. Setelah proses instalasi selesai, akan tampil pada menu dimana lokasi biasanya aplikasi yang *terinstall* berada, maka akan muncul "IRT" dan aplikasi tersebut siap dijalankan.

2. Berikut ini adalah langkah-langkah dalam melakukan *transfer* aplikasi dari PC ke dalam ponsel Nokia E63 menggunakan *Bluetooth*.

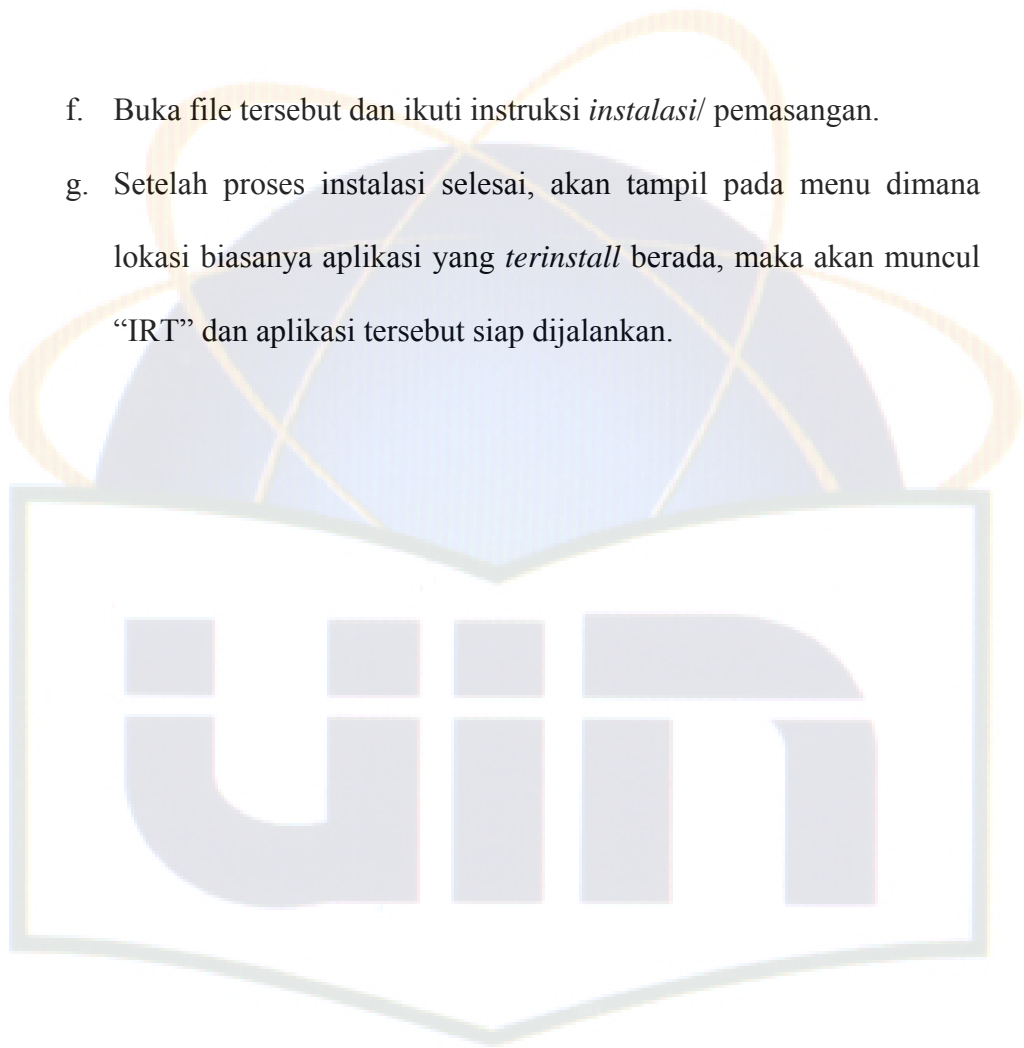
- a. Instalasi *driver Bluetooth*.
- b. Pasang *bluetooth* pada *port usb serial computer*.
- c. Aktifkan koneksi *bluetooth* ponsel.
- d. Jalankan aplikasi *bluesoil* pada Komputer.
- e. Lalu klik *center ball* untuk mencari perangkat *bluetooth* yang akan dikoneksikan.

- 
- f. Lalu *double klik icon* perangkat yang terkoneksi dengan icon. *file transfer service aktif*.
 - g. Lalu klik *file transfer service* pada *toolbar*.
 - h. Ponsel akan mengkonfirmasi untuk terhubung dengan PC.
 - i. Setelah itu *copy paste .jar*
 - j. Maka akan dilakukan *transfer* data dari komputer ke dalam ponsel.
 - k. File.jar telah ada di ponsel dalam bentuk pesan atau sms.
 - l. Buka sms berupa *file* aplikasi yang telah dikirim.
 - h. Ikuti instruksi, langkah-langkah *instalasi/pemasangan* pada ponsel nokia E63.
 - m. Setelah proses instalasi selesai, akan tampil pada menu dimana lokasi biasanya aplikasi yang terinstall berada, maka akan muncul “IRT” dan aplikasi tersebut siap dijalankan.

3. Berikut ini adalah langkah-langkah dalam melakukan *transfer* aplikasi dari PC ke dalam ponsel Nokia E63 menggunakan *card reader*

- a. Lepaskan kartu *memory eksternal* pada ponsel sesuai prosedur.
- b. Masukkan kedalam media *card reader* sesuai pada jenis *memory*.
- c. *Copy paste file irt .jar* ke dalam ponsel.
- d. Lepaskan kartu memori dan pasang kembali kedalam ponsel setelah *file* tersebut sudah terdapat pada memori.
- e. Jalankan aplikasi *file manager* yang sudah disediakan oleh ponsel atau aplikasi sejenis seperti *explore*, dsb.

- f. Buka file tersebut dan ikuti instruksi *instalasi*/ pemasangan.
- g. Setelah proses instalasi selesai, akan tampil pada menu dimana lokasi biasanya aplikasi yang *terinstall* berada, maka akan muncul “IRT” dan aplikasi tersebut siap dijalankan.





LAMPIRAN IV

Source Code

Config.php

```
<?php

/* File Konfigurasi
Database Mysql */
$mysql_host
= "localhost";
$mysql_user
= "root";
$mysql_password
= "";
$mysql_database
= "irtserver";

mysql_connect($mysql_host, $mysql_user, $mysql_password)
;
mysql_select_db($mysql_database);
error_reporting(0);
?>

getParseLogin.php
<?php
include_once("config.php");

$get_username = $_GET['u'];
$get_password = $_GET['p'];

$sql_query="SELECT * FROM
users WHERE uname =
'".$get_username."' AND salt =
'".$get_password."' LIMIT 1 ";

$result_sql_query=mysql_query(
$sql_query);

while($rows=mysql_fetch_array(
$result_sql_query)){
    $get_id
    = $rows['id'];
    $get_id_pel
    = $rows['id_pel'];
    $get_no_meter
    = $rows['no_meter'];
    $get_nama
    = $rows['nama'];
    $get_alamat
    = $rows['alamat'];
    $get_uname
    = $rows['uname'];
    $get_salt
    = $rows['salt'];
    $get_email
    = $rows['email'];

    $get_no_hp
    = $rows['no_hp'];
    $get_blocked
    = $rows['blocked'];
    $get_forgot
    = $rows['forgot'];
}
mysql_close();

// write xml users
echo "<?xml
version=\\"1.0\\"?>";
if ($get_id!="") {
    echo "<userlogin>";
    echo
    "<ketemu>1</ketemu>";
    echo
    "<id>".$get_id."</id>";
    echo
    "<id_pel>".$get_id_pel."</id_pel>";
    echo
    "<no_meter>".$get_no_meter."</no_meter>";
    echo
    "<nama>".$get_nama."</nama>";
    echo
    "<alamat>".$get_alamat."</alamat>";
    echo
    "<uname>".$get_uname."</uname>";
    echo
    "<salt>".$get_salt."</salt>";
    echo
    "<email>".$get_email."</email>";
    echo
    "<no_hp>".$get_no_hp."</no_hp>";
    echo
    "<blocked>".$get_blocked."</blocked>";
    echo
    "<forgot>".$get_forgot."</forgot>";
    echo "</userlogin>";
} else {
    echo "<userlogin>";
    echo
    "<ketemu>0</ketemu>";
    echo
    "<id>".$get_id."</id>";
    echo
    "<id_pel>".$get_id_pel."</id_pel>";
```

```

        echo
        "<no_meter>".$get_no_meter."</no_meter>";
        echo
        "<nama>".$get_nama."</nama>";
        echo
        "<alamat>".$get_alamat."</alamat>";
        echo
        "<uname>".$get_uname."</uname>";
        echo
        "<salt>".$get_salt."</salt>";
        echo
        "<email>".$get_email."</email>";
        echo
        "<no_hp>".$get_no_hp."</no_hp>";
        echo
        "<blocked>".$get_blocked."</blocked>";
        echo
        "<forgot>".$get_forgot."</forgot>";
        echo "</userlogin>";
    }
?>
getRekening.php
<?php

    $id_pel    = $_GET['idpel'];
    $get_bulan = $_GET['bulan'];
    $get_tahun = $_GET['tahun'];

    include_once("function/config.php");

    include_once("function/function_users.php");

    //$get_var_id_pel =
    $_POST['var_id_pel'];
    $users =
    select_users_by_id_pel($id_pel);

    $current_id = $users[0][0];
    $current_id_pel =
    $users[0][1];
    $current_no_meter =
    $users[0][2];
    $current_nama =
    $users[0][3];

    $current_alamat =
    $users[0][4];
    $current_uname =
    $users[0][5];
    $current_salt =
    $users[0][6];
    $current_email =
    $users[0][7];
    $current_no_hp =
    $users[0][8];
    $doc = new DOMDocument();
    $doc->formatOutput = true;

    $postpaid = $doc-
    >createElement( "postpaid" );
    $doc-
    >appendChild($postpaid);

    $kodeup= $doc-
    >createElement( "kodeup" );
    $kodeup->appendChild($doc-
    >createTextNode("54620"));
    $postpaid-
    >appendChild($kodeup);

    $idpel= $doc->createElement(
    "idpel" );
    $idpel->appendChild($doc-
    >createTextNode($_GET['idpel']
    ));
    $postpaid-
    >appendChild($idpel);

    $nama= $doc->createElement(
    "nama" );
    $nama->appendChild($doc-
    >createTextNode($current_nama
    ));
    $postpaid-
    >appendChild($nama);

    $tarif = $doc-
    >createElement( "tarif" );
    $tarif->appendChild($doc-
    >createTextNode("B2"));
    $postpaid-
    >appendChild($tarif);

    $daya = $doc->createElement(
    "daya" );
    $daya->appendChild($doc-
    >createTextNode("3500"));
    $postpaid-
    >appendChild($daya);

```

```

    $bulan= $doc->createElement(
"bulan" );
    $bulan->appendChild($doc-
>createTextNode($get_bulan));
    $postpaid-
>appendChild($bulan);

    $tahun = $doc-
>createElement( "tahun" );
    $tahun->appendChild($doc-
>createTextNode($get_tahun));
    $postpaid-
>appendChild($tahun);

    $rupiah= $doc-
>createElement( "rupiah" );
    $rupiah->appendChild($doc-
>createTextNode("250000"));
    $postpaid-
>appendChild($rupiah);

    echo $doc->saveXML();

?>
getVoucherLast.php
<?php

    $get_nometer      =
$_GET['nometer'];
    $get_bulan = $_GET['bulan'];
    $get_tahun = $_GET['tahun'];

include_once("function/config.
php");

include_once("function/functio
n_users.php");
    //$get_var_id_pel =
$_POST['var_id_pel'];
    //$users =
select_users_by_id_pel($id_pel
);
    $users =
select_users_by_no_meter($get_
nometer);

    $current_id = $users[0][0];
    $current_id_pel =
$users[0][1];
    $current_no_meter =
$users[0][2];
    $current_nama =
$users[0][3];
    $current_alamat =
$users[0][4];

    $current_uname =
$users[0][5];
    $current_salt =
$users[0][6];
    $current_email =
$users[0][7];
    $current_no_hp =
$users[0][8];

    echo "<prepaid>";
    echo
"<kodeup>54620</kodeup>";
    echo
"<id_pel>".$current_id_pel."</
id_pel>";
    echo
"<no_meter>".$current_no_meter
."</no_meter>";
    echo
"<nama>".$current_nama."</nama
>";
    echo "<tarif>R1</tarif>";
    echo "<daya>2200</daya>";
    echo
"<satuan_daya>V</satuan_daya>"
;

    echo
"<token>48678658075288669799</
token>";
    echo
"<rupiah>98400</rupiah>";
    echo
"<tanggal>20100607</tanggal>";
    echo "<jam>093817</jam>";
    echo
"<kodepp>0280011SYB54D3</kodep
p>";
    echo "</prepaid>";

?>

registerIRT.php
<?php
include_once("config.php");
include_once("function/config.
php");
include_once("function/functio
n_users.php");

$get_type = $_GET['type'];
$get_idpel = $_GET['idpel'];
$get_nometer =
$_GET['nometer'];
$get_nama = $_GET['nama'];
$get_alamat = $_GET['alamat'];

```

```

$get_email = $_GET['email'];
$get_nohp = $_GET['nohp'];
if ($current_id_pel=="") {
    $get_username = $_GET['username'];
    $get_password = $_GET['password'];
    $berhasil = 1;

    // urlRegisterPospaid =
    "http://127.0.0.1/j2me/registrPostpaid.php?idpel="+get_req_postpaid_idpel+"&nama="+get_req_postpaid_nama+"&alamat="+get_req_postpaid_alamat+"&email="+get_req_postpaid_email+"&nohp="+get_req_postpaid_nohp+"&username="+get_req_postpaid_username+"&password="+get_req_postpaid_password+"";

    $sql_query="INSERT INTO users (id,id_pel,no_meter,nama,alamat,uname,salt,email,no_hp,blocked,forgot)VALUES (NULL, '$get_idpel', '0', '$get_nama', '$get_alamat', '$get_username', '$get_password', '$get_email', '$get_nohp', '0', '0')";

    mysql_connect($mysql_host, $mysql_user, $mysql_password)or die("cannot connect");
    mysql_query($sql_query,$con);
    $con = mysql_connect($mysql_host, $mysql_user, $mysql_password);
    mysql_select_db($mysql_databases)or die("cannot select DB");
    } else {
        $berhasil = 0;
    }

    if ($get_type=="1"){ //postpaid
        if (($get_type=="") && ($get_idpel=="") && ($get_nama=="") && ($get_nometer=="") && ($get_alamat=="") && ($get_email=="") && ($get_nohp=="") && ($get_username=="") && ($get_password=="") ) {
            $berhasil=0;
        } else {
            //cek tidak boleh ada idpel sama

            $users = select_users_by_id_pel($get_idpel);

            $current_id_pel = $users[0][1];
        }
    } else {
        if ($get_type=="2"){ //prepaid
            if (($get_type=="") && ($get_idpel=="") && ($get_nama=="") && ($get_alamat=="") && ($get_email=="") && ($get_nohp=="") && ($get_username=="") && ($get_password=="") ) {
                $berhasil=0;
            } else {
                $users_a = select_users_by_id_pel($get_idpel);
            }
        }
    }
}

```

```

        $current_id_pel_a =
        $users[0][1];

        $users =
        select_users_by_no_meter($get_
        nometer);

        $current_id_pel =
        $users[0][1];

        if
        (($current_id_pel=="") &&
        ($current_id_pel_a=="")) {

            $berhasil = 1;

            $sql_query="INSERT INTO
            users
            (id,id_pel,no_meter,nama,alama
            t,uname,salt ,email,no_hp
            ,blocked ,forgot)VALUES (NULL
            , '$get_idpel',
            '$get_nometer', '$get_nama',
            '$get_alamat',
            '$get_username',
            '$get_password', '$get_email',
            '$get_nohp', '0', '0')";

            mysql_query($sql_query,$
            con);

            } else {

                $berhasil = 0;

            }

        }

        //mysql_query($sql_query);

        echo "<?xml
        version=\"1.0\"?>";
        //if
        (!mysql_query($sql_query,$con)
        )
        if ($berhasil=="0")
        {
            echo "<register>";

            echo
            "<status_register>0</status_re
            gister>";

            echo
            "</register>";
            } else
            if ($berhasil=="1")
            {
                echo "<register>";
                echo
                "<status_register>1</status_re
                gister>";

                echo
                "</register>";
            }
        }
    }
}
?>

```

function_users.php

```

<?php

function select_users(){
    $sql="SELECT * FROM
    users" ;
    $resultQuery=mysql_query
    ($sql);
    while
    ($rows=mysql_fetch_row($result
    Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_id($var_id){
    $sql="SELECT * FROM
    users WHERE id = '"
    .mysql_real_escape_string(trim
    ($var_id)). "' ";
    $resultQuery=mysql_query
    ($sql);
    while
    ($rows=mysql_fetch_row($result
    Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_id_pel($var_id
_pel){
    $sql="SELECT * FROM
    users WHERE id_pel = '"
    .mysql_real_escape_string(trim
    ($var_id_pel)). "' ";

```

```

        $resultQuery=mysql_query
($sql);
        while
($rows=mysql_fetch_row($result
Query)){
            $data[] = $rows;
        }
        return $data;
    }
function
select_users_by_no_meter($var_
no_meter){
    $sql="SELECT * FROM
users WHERE no_meter = '"
.mysql_real_escape_string(trim
($var_no_meter)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_nama($var_nama
){
    $sql="SELECT * FROM
users WHERE nama = '"
.mysql_real_escape_string(trim
($var_nama)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_alamat($var_al
amat){
    $sql="SELECT * FROM
users WHERE alamat = '"
.mysql_real_escape_string(trim
($var_alamat)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
}

        return $data;
    }
function
select_users_by_uname($var_una
me){
    $sql="SELECT * FROM
users WHERE uname = '"
.mysql_real_escape_string(trim
($var_uname)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_salt($var_salt
){
    $sql="SELECT * FROM
users WHERE salt = '"
.mysql_real_escape_string(trim
($var_salt)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_email($var_ema
il){
    $sql="SELECT * FROM
users WHERE email = '"
.mysql_real_escape_string(trim
($var_email)). "' ";
    $resultQuery=mysql_query
($sql);
    while
($rows=mysql_fetch_row($result
Query)){
        $data[] = $rows;
    }
    return $data;
}

function
select_users_by_no_hp($var_no_
hp){

```



```

        $sql="SELECT * FROM
users WHERE no_hp = '"
.mysql_real_escape_string(trim
($var_no_hp)). "' ";
        $resultQuery=mysql_query
($sql);
        while
($rows=mysql_fetch_row($result
Query)){
            $data[] = $rows;
        }
        return $data;
    }
    function
select_users_by_blocked($var_b
locked){
        $sql="SELECT * FROM
users WHERE blocked = '"
.mysql_real_escape_string(trim
($var_blocked)). "' ";
        $resultQuery=mysql_query
($sql);
        while
($rows=mysql_fetch_row($result
Query)){
            $data[] = $rows;
        }
        return $data;
    }
    function
select_users_by_forgot($var_fo
rgot){
        $sql="SELECT * FROM
users WHERE forgot = '"
.mysql_real_escape_string(trim
($var_forgot)). "' ";
        $resultQuery=mysql_query
($sql);
        while
($rows=mysql_fetch_row($result
Query)){
            $data[] = $rows;
        }
        return $data;
    }

    function
delete_users_by_id($var_id){
        $sql="DELETE FROM users
WHERE id = '"
.mysql_real_escape_string(trim
($var_id)). " LIMIT 1";
        $resultQuery=mysql_query
($sql);

        if($resultQuery){
            $returnDetete_users=1; } else
            { $returnDetete_users=0; }
            return
            $returnDetete_users;
        }

        function
insert_users($var_id_pel,$var_
no_meter,$var_nama,$var_alamat
,$var_uname,$var_salt,$var_ema
il,$var_no_hp,$var_blocked,$va
r_forgot){
            $sql="INSERT INTO users
(id,id_pel,no_meter,nama,alama
t,uname,salt,email,no_hp,block
ed,forgot) VALUES (NULL,'"
.mysql_real_escape_string(trim
($var_id_pel)). "',''"
.mysql_real_escape_string(trim
($var_no_meter)). "',''"
.mysql_real_escape_string(trim
($var_nama)). "',''"
.mysql_real_escape_string(trim
($var_alamat)). "',''"
.mysql_real_escape_string(trim
($var_uname)). "',''"
.mysql_real_escape_string(trim
($var_salt)). "',''"
.mysql_real_escape_string(trim
($var_email)). "',''"
.mysql_real_escape_string(trim
($var_no_hp)). "',''"
.mysql_real_escape_string(trim
($var_blocked)). "',''"
.mysql_real_escape_string(trim
($var_forgot)). "')";
            $resultQuery=mysql_query
($sql);
            if($resultQuery){
                $returnInsert_users=1; } else
                { $returnInsert_users=0; }
                return
                $returnInsert_users;
            }

            function
update_users($var_id,$var_id_p
el,$var_no_meter,$var_nama,$va
r_alamat,$var_uname,$var_salt,
$var_email,$var_no_hp,$var_blo
cked,$var_forgot){
                $sql="UPDATE users SET
id_pel = '"
.mysql_real_escape_string(trim
($var_id_pel)). "',' no_meter =
'"

```



```

.mysql_real_escape_string(trim
($var_no_meter)). "', nama =
'"
.mysql_real_escape_string(trim
($var_nama)). "', alamat = '"
.mysql_real_escape_string(trim
($var_alamat)). "', uname = '"
.mysql_real_escape_string(trim
($var_uname)). "', salt = '"
.mysql_real_escape_string(trim
($var_salt)). "', email = '"
.mysql_real_escape_string(trim
($var_email)). "', no_hp = '"
.mysql_real_escape_string(trim
($var_no_hp)). "', blocked =
'"
.mysql_real_escape_string(trim
($var_blocked)). "', forgot =
'"
.mysql_real_escape_string(trim
($var_forgot)). "' WHERE id =
'"
.mysql_real_escape_string(trim
($var_id)). "' LIMIT 1";
$resultQuery=mysql_query
($sql);
if($resultQuery){
$returnUpdate_users=1; } else
{ $returnUpdate_users=0; }
return
$returnUpdate_users;
}
?>

```

infoRekening Token.Java

```

package IRT;
import
java.io.DataInputStream;
import
java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import
java.io.InputStreamReader;
import
javax.microedition.io.Connector;
import
javax.microedition.io.HttpConnection;
import
javax.microedition.midlet.*;
import
javax.microedition.lcdui.*;
import
org.kxml2.io.KXmlParser;

```

```

import
org.netbeans.microedition.lcdui
i.SimpleTableModel;
import
org.netbeans.microedition.lcdui
i.SplashScreen;
import
org.netbeans.microedition.lcdui
i.TableItem;
import
org.xmlpull.v1.XmlPullParser;

/**
 * @author asti
 */
public class InfoRekeningToken
extends MIDlet implements
CommandListener {
    public configuration
objConf; // Create Object
Configuration
    private boolean
midletPaused = false;
    private StringItem
varXML_login_ketemu;
    private StringItem
varXML_login_id;
    private StringItem
varXML_login_idpel;
    private StringItem
varXML_login_meter;
    private StringItem
varXML_login_nama;
    private StringItem
varXML_login_alamat;
    private StringItem
varXML_login_uname;
    private StringItem
varXML_login_no_hp;
    private StringItem
varXML_login_salt;
    private StringItem
varXML_login_forgot;
    private StringItem
varXML_login_email;
    private StringItem
varXML_login_blocked;
    //<editor-fold
defaultstate="collapsed"
desc=" Generated Fields ">
    private Command
exitCommand;
    private Command
buttonLogin_LupaPassword;
    private Command
buttonLogin_Submit;

```

```

        private Command
buttonLogin_Registrasi;
        private Command
buttonLogin_Keluar;
        private Command
buttonRegistrasi_JenisMeter;
        private Command
buttonRegister_KembaliPrabayar
;
        private Command
buttonRegisterPascaBayar_Kemba
li;
        private Command
buttonRegisterPascaBayar_Prose
s;
        private Command
buttonHome_Keluar;
        private Command
buttonHome_Menu;
        private Command
buttonRegister_ProsesPrabayar;
        private Command
buttonMenu_Kembali;
        private Command
buttonInfoRekeningPostpaid_Kem
bali;
        private Command
buttonInfoRekening_Proses;
        private Command
buttonInfoRekening_Kembali;
        private Command
buttonKantorPLN;
        private Command
buttonPLNDKI_Kembali;
        private Command
butonPLNMenteng_Kembali;
        private Command Menteng;
        private Command
screenCommand;
        private Command
buttonProfile_Kembali;
        private Command
buttonRequestToken_Kembali;
        private Command
buttonBank_Kembali;
        private Command
buttonTDL_Kembali;
        private Command
buttonTDLR_kembali;
        private Command
buttonTDL_R_Kembali;
        private Command
buttonTDL_R_Proses;
        private Command
buttonTDL_P_Proses;

        private Command
buttonTDL_P_Kembali;
        private Command
buttonTDL_S_Proses;
        private Command
buttonTDL_B_Kembali;
        private Command
buttonTDL_B_Proses;
        private Command
buttonTDL_S_Kembali;
        private Command
buttonTDL_I_Proses;
        private Command
buttonTDL_I_Kembali;
        private Command
screenCommand3;
        private Command
screenCommand2;
        private Command
screenCommand1;
        private Command
screenCommand4;
        private Form formLogin;
        private TextField
var_loginUsername;
        private TextField
var_loginPassword;
        private SplashScreen
splashScreen;
        private List
RegistrasiPostpaidPrepaid;
        private Form
formRegistrasiPascaBayar;
        private TextField
var_registrasi_Password_PascaB
ayar;
        private TextField
var_registrasi_Nama_PascaBayar
;
        private TextField
var_registrasi_Alamat_PascaBay
ar;
        private TextField
var_registrasi_Email_PascaBaya
r;
        private TextField
var_registrasi_IDPel_PascaBaya
r;
        private TextField
var_registrasi_Username_PascaB
ayar;
        private Spacer spacer1;
        private TextField
var_registrasi_NoHP_PascaBayar
;

```

```

        private StringItem
stringItem1;
        private Form
formRegistrasiPrabayar;
        private TextField
var_registrasi_Username_Prabay
ar;
        private TextField
var_registrasi_IDPel_Prabayar;
        private TextField
var_registrasi_Email_Prabayar;
        private TextField
var_registrasi_Alamat_Prabayar
;
        private TextField
var_registrasi>Nama_Prabayar;
        private TextField
var_registrasi>Password_Prabay
ar;
        private TextField
var_registrasi_NoMeter_Prabaya
r;
        private TextField
var_registrasi_NoHP_Prabayar;
        private Spacer spacer;
        private StringItem
stringItem;
        private Alert
alertLoginFailed;
        private Form formHome;
        private ImageItem
imageItem;
        private Alert
alertRegisterFailed;
        private Alert
alertRegisterSuccess;
        private Alert
alertLoginSuccess;
        private List menuHome;
        private Form
formRekening_Input;
        private ChoiceGroup
var_RekeningInput_Bulan;
        private TextField
var_RekeningInput_Tahun;
        private Form formRekening;
        private List
formKantorPLN_Disjaya;
        private List
formKantorPLN;
        private Form
formKantorPLN_Detail_Jakarta;
        private StringItem
stringItem6;
        private List formBank;
        private List formInfoTDL;

        private Form
formHomeProfile;
        private Alert
alertProfileNotFound;
        private Form
formHomePrepaid_TokenTerakhir;
        private Form
formInfoTDL_R;
        private ChoiceGroup
var_inputTDL_Daya_R;
        private TextField
var_inputTDL_KWH_R;
        private Form
formInfoTDL_B;
        private TextField
var_inputTDL_KWH_B;
        private ChoiceGroup
var_inputTDL_Daya_B;
        private Form formHomeTDL;
        private Form
formInfoTDL_S;
        private ChoiceGroup
var_inputTDL_Daya_S;
        private TextField
var_inputTDL_KWH_S;
        private Form
formInfoTDL_P;
        private TextField
var_inputTDL_KWH_P;
        private ChoiceGroup
var_inputTDL_Daya_P;
        private Alert
alertPelangganPostpaid;
        private List
formKantorPLN_Disjabar;
        private Form
formInfoTDL_I;
        private TextField
var_inputTDL_KWH_I;
        private ChoiceGroup
var_inputTDL_Daya_I;
        private Alert
alertPelangganPrepaid;
        private List
formKantorPLN_Bali;
        private List
formKantorPLN_Disjatim;
        private List
formKantorPLN_Disjateng;
        private Form
formKantorPLN_Detail_Bali;
        private StringItem
stringItem2;
        private Form
formKantorPLN_Detail_Jabar;

```

```

        private StringItem
stringItem4;
        private Form
formKantorPLN_Detail_Jateng;
        private StringItem
stringItem3;
        private Form
formKantorPLN_Detail_Jatim;
        private StringItem
stringItem5;
        private Image
imgSplashScreen;
        private SimpleTableModel
tableModel1;
        //</editor-fold>
        private String urlLogin;
        private StringItem
varXML_status_register;
        private DataOutputStream
dos_ParseRegister;
        private DataInputStream
dis_ParseRegister;
        private String
urlRegisterPospaid;
        private String
urlRegisterPrepaid;
        private StringItem
var_RekeningPostpaid_kdup;
        private StringItem
var_RekeningPostpaid_idpel;
        private StringItem
var_RekeningPostpaid_nama;
        private StringItem
var_RekeningPostpaid_tarif;
        private StringItem
var_RekeningPostpaid_daya;
        private StringItem
var_RekeningPostpaid_bulan;
        private StringItem
var_RekeningPostpaid_tahun;
        private StringItem
var_RekeningPostpaid_rupiah;
        private String
urlRequestInfoRekening;
        private StringItem
var_GetVoucherLast_kdup;
        private StringItem
var_GetVoucherLast_id_pel;
        private StringItem
var_GetVoucherLast_no_meter;
        private StringItem
var_GetVoucherLast_nama;
        private StringItem
var_GetVoucherLast_tarif;
        private StringItem
var_GetVoucherLast_daya;

        private StringItem
var_GetVoucherLast_satuan_daya
;
        private StringItem
var_GetVoucherLast_token;
        private StringItem
var_GetVoucherLast_rupiah;
        private StringItem
var_GetVoucherLast_tanggal;
        private StringItem
var_GetVoucherLast_jam;
        private StringItem
var_GetVoucherLast_kodepp;
        private String
urlGetVoucherLast;
        private String urlProfile;
        private String
urlTDL_RumahTangga;
        private StringItem
varXML_tdl_kwh;
        private StringItem
varXML_tdl_tarif_daya;
        private StringItem
varXML_tdl_jam_nyala;
        private StringItem
varXML_tdl_rpkwh;
        private StringItem
varXML_tdl_rp_ppj;
        private StringItem
varXML_tdl_rp_mat;
        private StringItem
varXML_tdl_rp_tag;
        private String
urlTDL_Bisnis;
        private String
urlTDL_Pemerintah;
        private String
urlTDL_Sosial;
        private String
urlTDL_Industri;

        //</editor-fold>

        /**
         * The InfoRekeningToken
constructor.
         */
        public InfoRekeningToken()
        {
            public Display getDisplay () {
                return
                Display.getDisplay(this);
            }
        }
        /**
         * Exits MIDlet.
         */

```

```

        public void exitMIDlet() {
            switchDisplayable
            (null, null);
            destroyApp(true);
            notifyDestroyed();
        }
        /**
         * Called when MIDlet is
         started.
         * Checks whether the
         MIDlet have been already
         started and initialize/starts
         or resumes the MIDlet.
         */
        public void startApp() {
            if (midletPaused) {
                resumeMIDlet ();
            } else {
                initialize ();
                startMIDlet ();
            }
            midletPaused = false;
        }
        /**
         * Called when MIDlet is
         paused.
         */
        public void pauseApp() {
            midletPaused = true;
        }
        /**
         * Called to signal the
         MIDlet to terminate.
         * @param unconditional if
         true, then the MIDlet has to
         be unconditionally terminated
         and all resources has to be
         released.
         */
        public void
        destroyApp(boolean
        unconditional) {
        }

        /* service untuk
        mengenerate XML login */
        public void
        ParseLogin(InputStream
        isxml_ParseLogin) throws
        IOException {
            KXmlParser
            KXMLLoginParser = new
            KXmlParser();
            try {
                KXMLLoginParser.setInput(new
                InputStreamReader(isxml_ParseL
                ogin));
                KXMLLoginParser.next();
                KXMLLoginParser.require(XmlPul
                lParser.START_TAG, null,
                "userlogin");
                while
                (KXMLLoginParser.nextTag() !=
                XmlPullParser.END_TAG) {
                    KXMLLoginParser.require(XmlPul
                    lParser.START_TAG, null,
                    "ketemu");
                    varXML_login_ketemu = new
                    StringItem("Found :
                    ",KXMLLoginParser.nextText());
                    KXMLLoginParser.require(XmlPul
                    lParser.END_TAG, null,
                    "ketemu");
                    KXMLLoginParser.nextTag();
                    //System.out.print("cuk=
                    "+varXML_login_ketemu.toSring
                    ());
                    KXMLLoginParser.require(XmlPul
                    lParser.START_TAG, null,
                    "id");
                    varXML_login_id = new
                    StringItem("PLN Pin :
                    ",KXMLLoginParser.nextText());
                    KXMLLoginParser.require(XmlPul
                    lParser.END_TAG, null, "id");
                    KXMLLoginParser.nextTag();
                    KXMLLoginParser.require(XmlPul
                    lParser.START_TAG, null,
                    "id_pel");
                    varXML_login_idpel = new
                    StringItem("ID Pelanggan :
                    ",KXMLLoginParser.nextText());
                    KXMLLoginParser.require(XmlPul
                    lParser.END_TAG, null,
                    "id_pel");
                }
            }
        }
    }

```

```

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"no_meter");

varXML_login_meter = new
StringItem("Nomor Meter :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"no_meter");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"nama");

varXML_login_nama = new
StringItem("Nama :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"nama");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"alamat");

varXML_login_alamat = new
StringItem("Alamat :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"alamat");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"uname");

varXML_login_uname = new
StringItem("Username :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"uname");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"salt");

varXML_login_salt = new
StringItem("Password :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"salt");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"email");

varXML_login_email = new
StringItem("Email :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"email");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"no_hp");

varXML_login_no_hp = new
StringItem("Telepon :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
"no_hp");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
"blocked");

varXML_login_blocked = new
StringItem("Blocked :
",KXMLLoginParser.nextText());

```



```

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
    "blocked");

KXMLLoginParser.nextTag();

KXMLLoginParser.require(XmlPullParser.START_TAG, null,
    "forgot");

varXML_login_forgot = new
StringItem("Forgot :
",KXMLLoginParser.nextText());

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
    "forgot");
    }

KXMLLoginParser.require(XmlPullParser.END_TAG, null,
    "userlogin");
    } catch (Exception e)
    {
        e.printStackTrace();
    }
}

/* Fungsi untuk memanggil
ParseLogin */
public void
getXML_ParseLogin(String
urlXML_ParseLogin) throws
IOException {
    HttpURLConnection conn;
    DataInputStream
dis_ParseLogin;
    DataOutputStream
dos_ParseLogin;
    try {
        conn =
        (HttpURLConnection)
Connector.open(urlXML_ParseLogin);

conn.setRequestMethod(HttpURLConnection.GET);

conn.setRequestProperty("Content-Type", "text/xml");
        dos_ParseLogin =
        new
        DataOutputStream(conn.openData
        OutputStream());

dis_ParseLogin =
        new
        DataInputStream(conn.openDataI
        nputStream());
        if
        (conn.getResponseCode() ==
        HttpURLConnection.HTTP_OK) {
            System.out.println("begin
            parsing xml data");

            ParseLogin(dis_ParseLogin);
            /** Mengambil data XML dari
            server */

            System.out.println("end
            parsing xml data");
            if
            (varXML_login_ketemu.getText()
            .equals("1")){
                switchDisplayable(getAlertLogi
                nSuccess(),getFormHome()); //
                User Logged
            } else { /*
            User Not Found */
                switchDisplayable(getAlertLogi
                nFailed(),getFormLogin()); //
                User Not Logged
            }
        } else {
            /** Koneksi
            Error */

            System.out.println("Koneksi
            Error");
        }
    } catch (Exception e)
    {
        System.out.println("Error: " +
        e); /** Error */
    }
}

public void
getXML_ParseProfile(String
urlXML_ParseProfile) throws
IOException {
    HttpURLConnection conn;
    DataInputStream
dis_ParseProfile;
    DataOutputStream
dos_ParseProfile;
    try {

```

```

        conn =
(HttpConnection)
Connector.open(urlXML_ParsePro
file);

conn.setRequestMethod(HttpConn
ection.GET);

conn.setRequestProperty("Conte
nt-Type", "text/xml");

        dos_ParseProfile =
new
DataOutputStream(conn.openData
OutputStream());

        dis_ParseProfile =
new
DataInputStream(conn.openDataI
nputStream());

        if
(conn.getResponseCode() ==
HttpConnection.HTTP_OK) {

ParseLogin(dis_ParseProfile);
/** Mengambil data XML dari
server */

        if
(varXML_login_ketemu.getText()
.equals("1")){

getFormHomeProfile().deleteAll
());

//
getFormHomeProfile().append(va
r_id);

        if
(varXML_login_meter.getText().
equals("0")){

getFormHomeProfile().append("P
elanggan Postpaid");

getFormHomeProfile().append(va
rXML_login_idpel);

        } else
{

getFormHomeProfile().append("P
elanggan Prepaid");

getFormHomeProfile().append(va
rXML_login_idpel);

getFormHomeProfile().append(va
rXML_login_meter);

        }

getFormHomeProfile().append(va
rXML_login_nama);

getFormHomeProfile().append(va
rXML_login_alamat);

getFormHomeProfile().append(va
rXML_login_email);

getFormHomeProfile().append(va
rXML_login_no_hp);

switchDisplayable(null,
getFormHomeProfile());
        } else { /*
User Not Found */

switchDisplayable(null,
getAlertProfileNotFound()); //
User Not Logged

        } else {
        /** Koneksi
Error */

System.out.println("Koneksi
Error");

        } catch (Exception e)
{

System.out.println("Error: " +
e); /** Error */

        }

        /* service untuk
mengenerate XML
KXMLParseRegister_Postpaid */
        public void
ParseRegister(InputStream
isxml_ParseLogin) throws
IOException {

            KXmlParser
KXMLParseRegister = new
KXmlParser();

            try {

KXMLParseRegister.setInput(new
InputStreamReader(isxml_ParseL
ogin));

KXMLParseRegister.next();

```



```

KXMLParseRegister.require(XmlPullParser.START_TAG, null,
"register");
        while
(KXMLParseRegister.nextTag()
!= XmlPullParser.END_TAG) {
KXMLParseRegister.require(XmlPullParser.START_TAG, null,
"status_register");

varXML_status_register = new
StringItem("Status :
",KXMLParseRegister.nextText()
);

KXMLParseRegister.require(XmlPullParser.END_TAG, null,
"status_register");
        }

KXMLParseRegister.require(XmlPullParser.END_TAG, null,
"register");
        } catch (Exception e)
{
e.printStackTrace();
        }

        /* Fungsi untuk memanggil
ParseRegister */
        public void
getXML_ParseRegister(String
urlXML_ParseRegister_Postpaid)
throws IOException {
            HttpURLConnection conn;
            DataInputStream
dis_ParseLogin;
            DataOutputStream
dos_ParseLogin;
            try {
                conn =
(HttpURLConnection)
Connector.open(urlXML_ParseReg
ister_Postpaid);

conn.setRequestMethod(HttpConn
ection.GET);

conn.setRequestProperty("Conte
nt-Type", "text/xml");
                dos_ParseRegister
= new

```

```

DataOutputStream(conn.openData
OutputStream());
                dis_ParseRegister
= new
DataInputStream(conn.openDataI
nputStream());
                if
(conn.getResponseCode() ==
HttpURLConnection.HTTP_OK) {

ParseRegister(dis_ParseRegiste
r); /** Mengambil data XML
dari server */
                if
(varXML_status_register.getTex
t().equals("1")){

System.out.println("hasil
login =
"+varXML_status_register.getTe
xt().toString());

switchDisplayable(
getAlertRegisterSuccess(),
getFormLogin()); // User
Logged
                } else { /*
User Not Found */

switchDisplayable(getAlertRegi
sterFailed(), getFormLogin());
// User Not Logged
                }
            } else {
                /** Koneksi
Error */

System.out.println("Koneksi
Error");
            }
        } catch (Exception e)
{
System.out.println("Error: " +
e); /** Error */
        }
        }
        /* service untuk
mengenerate XML
ParseGetRekening */
        public void
ParseGetRekening(InputStream
isxml_ParseGetRekening) throws
IOException {

```

```

        KXmlParser
KXMLParseGetRekening = new
KXmlParser();
        try {

KXMLParseGetRekening.setInput(
new
InputStreamReader(isxml_ParseG
etRekening));

KXMLParseGetRekening.next();

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"postpaid");
                while
(KXMLParseGetRekening.nextTag(
) != XmlPullParser.END_TAG) {

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"kodeup");

var_RekeningPostpaid_kdup =
new StringItem("Kode UP:
",KXMLParseGetRekening.nextTex
t());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"kodeup");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"idpel");

var_RekeningPostpaid_idpel =
new StringItem("ID Pelanggan :
",KXMLParseGetRekening.nextTex
t());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"idpel");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"nama");

var_RekeningPostpaid_nama =
new StringItem("Nama:
",KXMLParseGetRekening.nextTex
t());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"nama");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"tarif");

var_RekeningPostpaid_tarif =
new StringItem("Tarif :
",KXMLParseGetRekening.nextTex
t());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"tarif");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"daya");

var_RekeningPostpaid_daya =
new StringItem("Daya :
",KXMLParseGetRekening.nextTex
t());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"daya");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"bulan");

var_RekeningPostpaid_bulan =
new StringItem("Bulan :

```

```

",KXMLParseGetRekening.nextText());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"bulan");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"tahun");

var_RekeningPostpaid_tahun =
new StringItem("Tahun :
",KXMLParseGetRekening.nextText());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"tahun");

KXMLParseGetRekening.nextTag()
;

KXMLParseGetRekening.require(X
mlPullParser.START_TAG, null,
"rupiah");

var_RekeningPostpaid_rupiah =
new StringItem("Rupiah Tagihan :
",KXMLParseGetRekening.nextText());

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"rupiah");

    }

KXMLParseGetRekening.require(X
mlPullParser.END_TAG, null,
"postpaid");
    } catch (Exception e)
    {
        e.printStackTrace();
    }
}

/* Fungsi untuk memanggil
ParseGetRekening */

    public void
getXML_ParseGetRekening(String
urlXML) throws IOException {
    HttpConnection conn;
    DataInputStream
dis_getXML_ParseGetRekening;
    DataOutputStream
dos_getXML_ParseGetRekening;
    try {
        conn =
        (HttpConnection)
Connector.open(urlXML);

        conn.setRequestMethod(HttpConn
ection.GET);

        conn.setRequestProperty("Conte
nt-Type", "text/xml");

        dos_getXML_ParseGetRekening =
        new
        DataOutputStream(conn.openData
OutputStream());

        dis_getXML_ParseGetRekening =
        new
        DataInputStream(conn.openDataI
nputStream());
        if
        (conn.getResponseCode() ==
HttpConnection.HTTP_OK) {

            ParseGetRekening(dis_getXML_Pa
rseGetRekening);

            /* Mengambil
            data XML dari server */

            System.out.println("Mengambil
            data ParseGetRekening XML dari
            server");

            //System.out.println(var_GetVo
            ucherLast_token.toString());

            getFormRekening().deleteAll();

            getFormRekening().append(var_R
            ekeningPostpaid_kdup);

            getFormRekening().append(var_R
            ekeningPostpaid_idpel);

            getFormRekening().append(var_R
            ekeningPostpaid_nama);
        }
    }
}

```

```

getFormRekening().append(var_RekeningPostpaid_tarif);

getFormRekening().append(var_RekeningPostpaid_daya);

getFormRekening().append(var_RekeningPostpaid_bulan);

getFormRekening().append(var_RekeningPostpaid_tahun);

getFormRekening().append(var_RekeningPostpaid_rupiah);

switchDisplayable(null,
getFormRekening());

        } else {
            /** Koneksi
Error */

System.out.println("Koneksi
Error");
        } catch (Exception e)
{

System.out.println("Error: " +
e); /** Error */
        }

        /* service untuk
mengenerate XML GetVoucherLast
*/

        public void
ParseGetVoucherLast(InputStream isxml_ParseGetVoucherLast)
throws IOException {
            KXmlParser
KXMLParseGetVoucherLast = new
KXmlParser();
            try {

KXMLParseGetVoucherLast.setInput(
new
InputStreamReader(isxml_ParseGetVoucherLast));

KXMLParseGetVoucherLast.next();

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "prepaid");
                while
(KXMLParseGetVoucherLast.nextTag() != XmlPullParser.END_TAG)
{

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "kodeup");

var_GetVoucherLast_kdup = new
StringItem("Kode UP:
",KXMLParseGetVoucherLast.nextText());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"kodeup");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "id_pel");

var_GetVoucherLast_id_pel =
new StringItem("ID Pelanggan :
",KXMLParseGetVoucherLast.nextText());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"id_pel");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "no_meter");

var_GetVoucherLast_no_meter =
new StringItem("NO Meter:
",KXMLParseGetVoucherLast.nextText());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"no_meter");

```

```

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "nama");

var_GetVoucherLast_nama = new
StringItem("Nama:
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "nama");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "tarif");

var_GetVoucherLast_tarif = new
StringItem("Tarif :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "tarif");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "daya");

var_GetVoucherLast_daya = new
StringItem("Daya :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "daya");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "satuan_daya");

var_GetVoucherLast_satuan_daya
= new StringItem("Satuan Daya
:
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "satuan_daya");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "token");

var_GetVoucherLast_token = new
StringItem("Token :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "token");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require(
    XmlPullParser.START_TAG,
    null, "rupiah");

var_GetVoucherLast_rupiah =
new StringItem("Rupiah
Pembelian. :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require(
    XmlPullParser.END_TAG, null,
    "rupiah");

KXMLParseGetVoucherLast.nextTag();

KXMLParseGetVoucherLast.require

```

```

e(XmlPullParser.START_TAG,
null, "tanggal");

var_GetVoucherLast_tanggal =
new StringItem("Tanggal :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"tanggal");

KXMLParseGetVoucherLast.nextTa
g();

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "jam");

var_GetVoucherLast_jam = new
StringItem("Jam :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"jam");

KXMLParseGetVoucherLast.nextTa
g();

KXMLParseGetVoucherLast.require
e(XmlPullParser.START_TAG,
null, "kodepp");

var_GetVoucherLast_kodepp =
new StringItem("Kode PP :
",KXMLParseGetVoucherLast.next
Text());

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"kodepp");

}

KXMLParseGetVoucherLast.require
e(XmlPullParser.END_TAG, null,
"prepaid");
} catch (Exception e)
{
e.printStackTrace();
}

}

}

/* Fungsi untuk memanggil
ParseGetVoucherLast */
public void
getXML_ParseGetVoucherLast(Str
ing urlXML) throws IOException
{
    HttpURLConnection conn;
    DataInputStream
dis_getXML_ParseGetVoucherLast
;
    DataOutputStream
dos_getXML_ParseGetVoucherLast
;

    try {
        conn =
        (HttpURLConnection)
Connector.open(urlXML);

        conn.setRequestMethod(HttpConn
ection.GET);

        conn.setRequestProperty("Conte
nt-Type", "text/xml");

        dos_getXML_ParseGetVoucherLast
= new
DataOutputStream(conn.openData
OutputStream());

        dis_getXML_ParseGetVoucherLast
= new
DataInputStream(conn.openDataI
nputStream());
        if
        (conn.getResponseCode() ==
HttpURLConnection.HTTP_OK) {

            ParseGetVoucherLast(dis_getXML
_ParseGetVoucherLast);
            /* Mengambil
            data XML dari server */

            System.out.println("Mengambil
            data XML dari server");

            //System.out.println(var_GetVo
            ucherLast_token.toString());

            getFormHomePrepaid_TokenTerakh
            ir().deleteAll();

            getFormHomePrepaid_TokenTerakh
            ir().append(var_GetVoucherLast
            _kdup);
        }
    }
}

```



```

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_no_meter);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_id_pel);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_nama);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_tarif);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_daya);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_satuan_daya);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_token);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_tanggal);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_jam);

getFormHomePrepaid_TokenTerakh
ir().append(var_GetVoucherLast
_kodepp);

switchDisplayable(null,
getFormHomePrepaid_TokenTerakh
ir());

        } else {
            /** Koneksi
Error */

System.out.println("Koneksi
Error");
        }
    } catch (Exception e)
    {
        System.out.println("Error: " +
e); /** Error */
    }
}

Configuration.java
package IRT;
public class configuration {
    public static
String getIRT_IP() {String
IRT_IP = "127.0.0.1"; return
IRT_IP; } // IP LAPTOP
    //public static String
getIRT_IP() {String IRT_IP =
"202.162.220.59"; return
IRT_IP; } //IP INTERNET
    public static String
getIRT_DBPostpaid_IP() {String
IRTDDBPostpaid_IP =
"127.0.0.1"; return
IRTDDBPostpaid_IP; }
    //public static String
getIRT_DBPostpaid_IP() {String
IRTDDBPostpaid_IP =
"202.162.220.59"; return
IRTDDBPostpaid_IP; }
    public static String
getIRT_DBPrepaid_IP() {String
IRTDDBPrepaid_IP = "127.0.0.1";
return IRTDBPrepaid_IP; } //
IP LAPTOP
    //public static String
getIRT_DBPrepaid_IP() {String
IRTDDBPrepaid_IP =
"202.162.220.59"; return
IRTDDBPrepaid_IP; } //IP
INTERNET
    public static String
getIRT_Path() {String IRT_Path
= "irtserver8"; return
IRT_Path; } //INTERNET-NOKIA
    public static String
getIRT_FileLogin() {String
IRT_FileLogin =
"getParseLogin.php"; return
IRT_FileLogin; }
    public static String
getIRT_FileRegister() {String
IRT_FileRegister =
"registerIRT.php"; return
IRT_FileRegister; }
    public static String
getIRT_FileSimulasiTDL()
{String IRT_FileSimulasiTDL =
"getServiceTDL.php"; return
IRT_FileSimulasiTDL; }

```

```

        public static
String
getIRT_DBPostpaid_Path()
{String IRTDBPostpaid_Path =
"irtserver8"; return
IRTDBPostpaid_Path;
} //INTERNET-NOKIA
        public static String
getIRT_DBPostpaid_FileGetReken
ing() {String
IRTDBPostpaid_FileGetRekening
= "getRekening.php"; return
IRTDBPostpaid_FileGetRekening;
}

        public static String
getIRT_DBPrepaid_Path()
{String IRTDBPrepaid_Path =
"irtserver8"; return
IRTDBPrepaid_Path; }
        //INTERNET-NOKIA
        public static String
getIRT_DBPrepaid_FileGetLastTo
ken() {String
IRTDBPrepaid_FileGetLastToken
= "getVoucherLast.php"; return
IRTDBPrepaid_FileGetLastToken;
}

```





LAMPIRAN V

Wawancara dan Kuesioner

WAWANCARA

Responden : Kustan

Penanya : Indah Fauziah

Waktu : Senin, 29 November 2010

Tujuan : Mengetahui sistem *call center* 123 PLN Distribusi Jakarta
Raya dan Tangerang

1. Apa sajakah tugas *call center* 123 PLN dalam melayani pelanggan?

Jawab :

Tugas *call center* 123 PLN adalah melayani pengaduan-pengaduan masyarakat tentang listrik, memberikan informasi pelayanan-pelayanan yang ada di PLN dan memberikan informasi tagihan rekening listrik.

2. Apa sajakah yang sering di pertanyakan oleh pelanggan PLN?

Jawab :

Yang sering dipertanyakan oleh pelanggan PLN adalah mengenai informasi tagihan rekening listrik.

3. Selain melalui *call center* 123 media apa saja yang bisa pelanggan gunakan untuk mengetahui informasi tagihan rekening listrik?

Jawab :

Melalui media website PLN. Dengan adanya *website* pun belum efisien karena pelanggan masih sering menghubungi *call center* 123 untuk mencari informasi yang dibutuhkan oleh pelanggan.

4. Apa kendala yang dihadapi PLN dalam melayani pelanggan melalui *call center* 123?

Jawab :

Kendala yang ada yaitu jumlah operator yang ada tidak sebanding dengan banyaknya jumlah pelanggan yang menghubungi *call center*.

5. Upaya-upaya apa sajakah yang telah dilakukan oleh PLN untuk meningkatkan pelayanan konsumen?

Jawab :

Upaya yang dilakukan yaitu dengan peningkatan pelayanan melalui website PLN .

6. Menurut bapak apakah perlu tambahan sistem baru yang dapat membantu pelanggan dalam memperoleh informasi rekening?

Jawab :

Perlu, untuk lebih memudahkan pelanggan dalam memperoleh informasi rekening listrik serta menambah citra perusahaan terhadap pelayanan pelanggan.

TESTING KUESIONER
PROTOTYPE INFO REKENING LISTRIK PLN BERBASIS *MOBILE*

Nama :
Alamat :

1. Menurut Anda, Apakah pelayanan informasi pelanggan sekarang ini sudah cukup membantu?
☐ Sudah
☐ Belum
2. Apakah pelanggan mengalami kesulitan mencari informasi seputar rekening listrik dengan media pelayanan pelanggan yang sudah ada?
☐ Ya
☐ Tidak
3. Menurut anda apakah pelayanan pelanggan selama ini sudah cukup baik?
☐ Sudah
☐ Belum
4. Menurut anda, apa saja kekurangan dari pelayanan yang sudah ada?
☐ Jaringan Layanan susah dihubungi
☐ Lama dalam pemrosesan
☐ Tidak tahu
5. Menurut anda, Apakah perlu dibutuhkan suatu sistem informasi baru dalam pelayanan pelanggan berbasis mobile?
☐ Ya
☐ Tidak
6. Menurut anda, apakah aplikasi info rekening listrik PLN berbasis *mobile* dapat memberikan kemudahan bagi pelanggan dalam memperoleh informasi?
☐ Ya
☐ Tidak

7. Apakah anda mengharapkan perusahaan PLN perlu untuk membuat aplikasi ini sebagai sarana informasi?

☐ Ya

☐ Tidak

8. Informasi apa saja yang perlu ditampilkan dalam aplikasi tersebut?

☐ Informasi tagihan rekening listrik

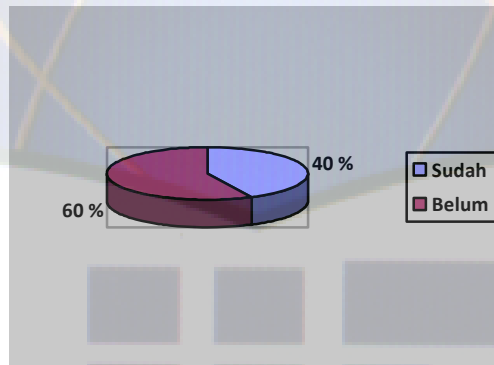
☐ Informasi rekening listrik

☐ Informasi token

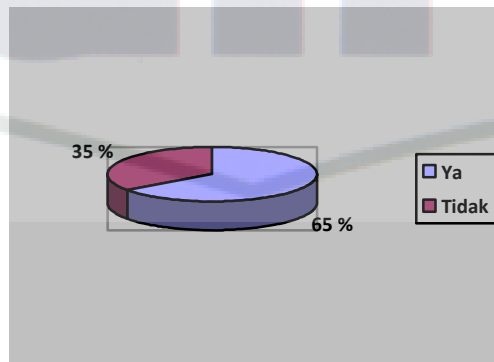
☐ Informasi Layanan lainnya

HASIL TESTING KUESIONER PROTOTYPE INFO REKENING LISTRIK PLN BERBASIS *MOBILE*

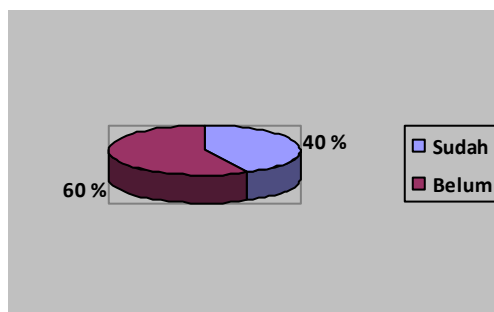
1. Menurut Anda, Apakah pelayanan informasi pelanggan sekarang ini sudah cukup membantu?



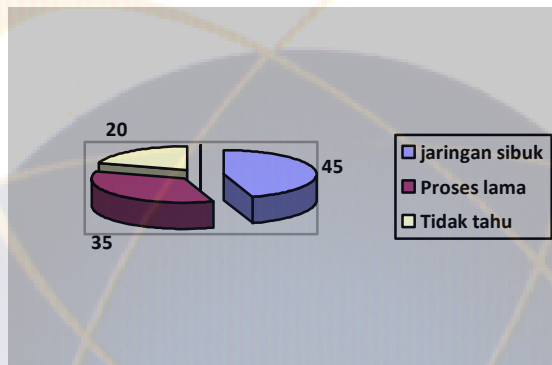
2. Apakah pelanggan mengalami kesulitan mencari informasi seputar rekening listrik dengan media pelayanan pelanggan yang sudah ada?



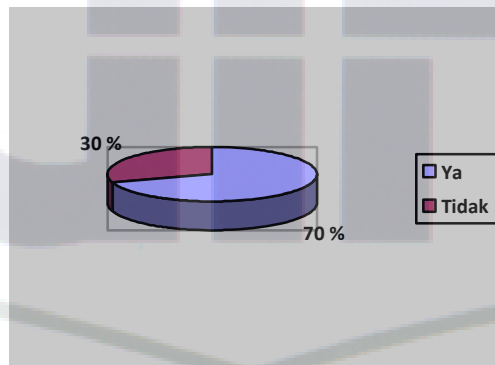
3. Menurut anda apakah pelayanan pelanggan selama ini sudah cukup baik?



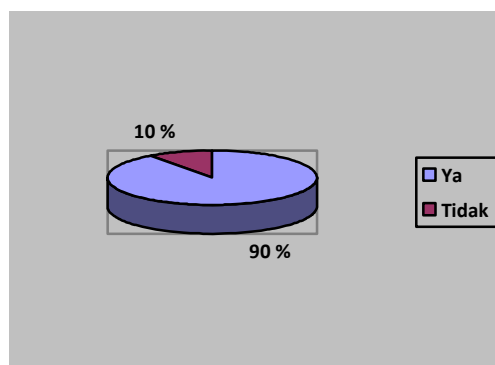
4. Menurut anda, apa saja kekurangan dari pelayanan yang sudah ada?



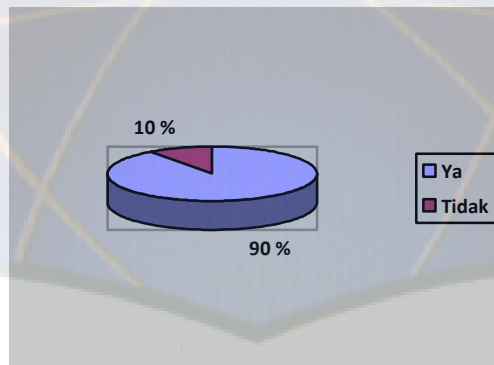
5. Menurut anda, Apakah perlu dibutuhkan suatu sistem informasi baru dalam pelayanan pelanggan berbasis mobile?



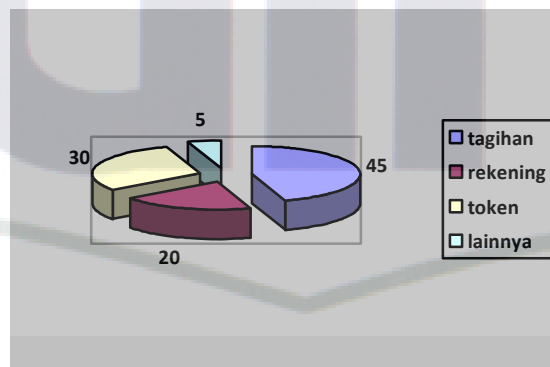
6. Menurut anda, apakah aplikasi info rekening listrik PLN berbasis *mobile* dapat memberikan kemudahan bagi pelanggan dalam memperoleh informasi?



7. Apakah anda mengharapkan perusahaan PLN perlu untuk membuat aplikasi ini sebagai sarana informasi?



8. Informasi apa saja yang perlu ditampilkan dalam aplikasi tersebut?

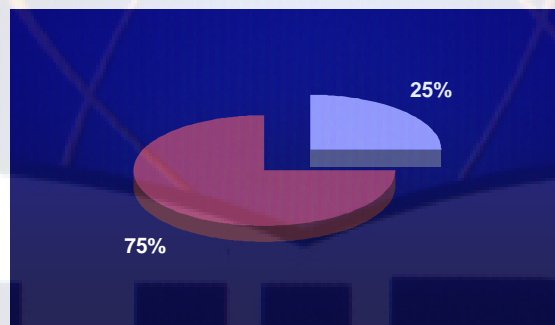


**HASIL KUESIONER TESTING PROROTYPE APLIKASI INFO
REKENING LISTRIK BERBASIS *MOBILE***

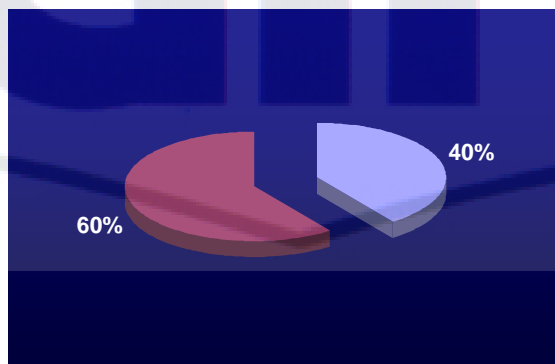
No.	Keterangan	Jumlah	Persentase
1.	Bagaimana pendapat anda tentang prototype aplikasi info rekening listrik PLN ini? a. Sangat bagus b. Cukup bagus c. Kurang bagus d. Tidak bagus	5 15 0 0	25% 75% 0% 0%
2.	Seberapa besarkah pengaruh dari aplikasi ini dapat membantu anda dalam mencari informasi rekening tagihan listrik PLN? a. Sangat Membantu b. Cukup Membantu c. Kurang Membantu d. Tidak Membantu	8 12 0 0	40% 60% 0% 0%
3.	Dari aplikasi ini bagian mana yang paling sering anda butuhkan? a. Info Rekening Listrik b. Info Token c. Info Tagihan Rekening Listrik d. Lain-lain	3 4 10 3	15% 20% 50% 15%
4.	Bagaimana kelengkapan informasi dari aplikasi ini? a. Sangat lengkap b. Cukup lengkap c. Kurang lengkap d. Tidak lengkap	5 11 4 0	25% 55% 20% 0%
5.	Apakah anda mengalami kesulitan dalam mengakses aplikasi ini? a. Ya b. Tidak	5 15	25% 75%
6.	Menurut anda apakah aplikasi info rekening listrik PLN berbasis <i>mobile</i> mampu menunjang sarana pelayanan pelanggan khususnya tentang informasi rekening listrik? a. Ya b. Tidak	18 2	90% 10%
7.	Apakah aplikasi ini memudahkan anda dalam memperoleh info rekening listrik?		

a. Ya	15	75%
b. Tidak	5	25%

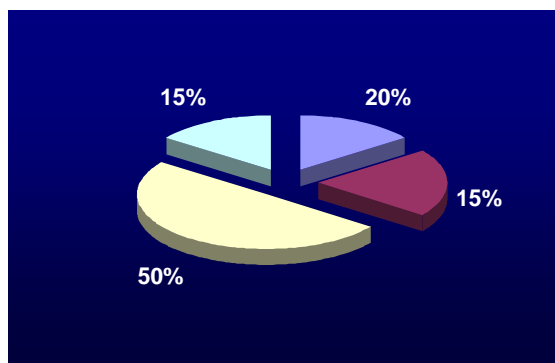
1. Bagaimana pendapat anda tentang prototype aplikasi info rekening listrik PLN ini?



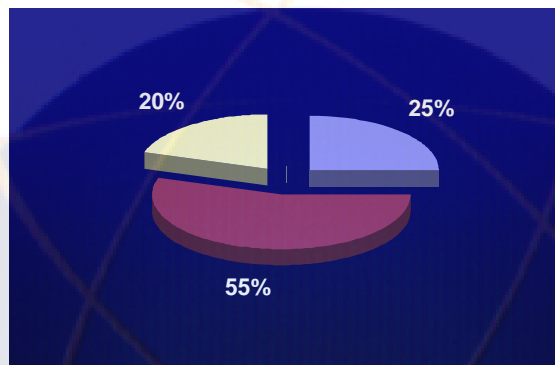
2. Seberapa besarkah pengaruh dari aplikasi ini dapat membantu anda dalam mencari informasi rekening tagihan listrik PLN?



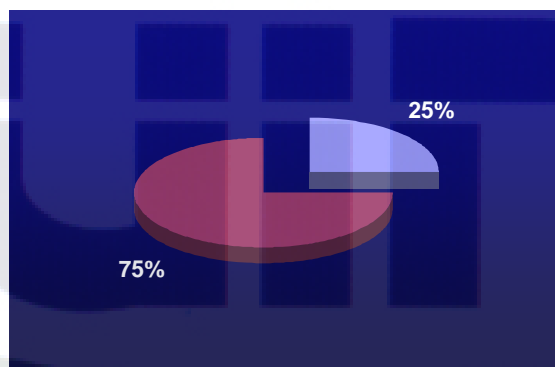
3. Dari aplikasi ini bagian mana yang paling sering anda butuhkan?



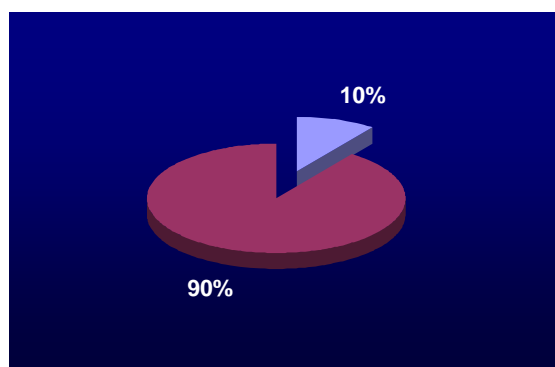
4. Bagaimana kelengkapan informasi dari aplikasi ini?



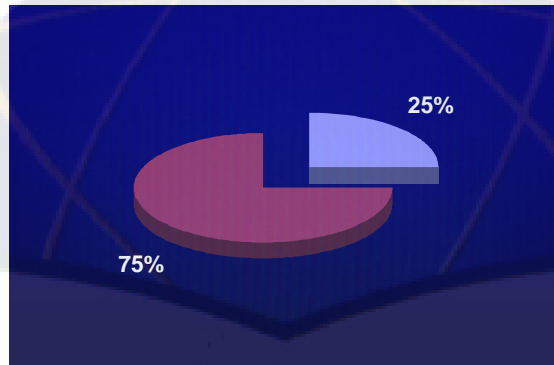
5. Apakah anda mengalami kesulitan dalam mengakses aplikasi ini?



6. Menurut anda apakah aplikasi info rekening listrik PLN berbasis mobile mampu menunjang sarana pelayanan pelanggan khususnya tentang informasi rekening listrik?



7. Apakah aplikasi ini memudahkan anda dalam memperoleh info rekening listrik?





LAMPIRAN VI

Surat Keterangan Penelitian



**PT. PLN (PERSERO)
DISTRIBUSI JAKARTA RAYA DAN TANGERANG
SUB BIDANG APLIKASI SISTEM TEKNOLOGI INFORMASI (ASTI)**

SURAT KETERANGAN RISET TUGAS AKHIR

NO: 11/KP/TI/11/2011

Dengan ini PT. PLN (persero) Distribusi Jakarta dan Tangerang, Bidang Perencanaan, Sub Bidang Aplikasi Teknologi Informasi menerangkan bahwa :

Nama : **INDAH FAUZIAH**
NIM : **206091004057**
Jurusan : **S1 – TEKNIK INFORMATIKA**
Universitas : **UIN Syarif Hidayatullah Jakarta**

Telah melaksanakan kerja praktek di PT. PLN (persero) Distribusi Jakarta Raya dan Tangerang, Bidang Perencanaan, Sub Bidang Aplikasi Teknologi Informasi (ASTI) selama 3 (tiga) minggu terhitung mulai tanggal 5 September 2011 hingga 25 Oktober 2011.

Demikian Surat Keterangan Riset Tugas Akhir ini dibuat dengan sebenarnya untuk digunakan seperlunya.

Mengetahui,
Pembimbing Lapangan

Kustan Setiawan, ST
Asisten Analist IT