

**APLIKASI SISTEM PEMBAYARAN REKENING LISTRIK
MELALUI JARINGAN GPRS**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun Oleh :

AHMAD ZAMAHSYARI HARDIANSYAH

NIM. 0310633003

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS BRAWIJAYA

MALANG

2009

**APLIKASI SISTEM PEMBAYARAN REKENING LISTRIK
MELALUI JARINGAN GPRS**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

UNIVERSITAS BRAWIJAYA



Disusun Oleh :

AHMAD ZAMAHSYARI HARDIANSYAH

NIM. 0310633003

Telah Diperiksa dan Disetujui Oleh:

Dosen Pembimbing

Arief Andv Soebroto, ST., M.kom

NIP. 19720425 199903 1 002

Herman Tolle, ST., MT

NIP. 19740823 200012 1 001

**APLIKASI SISTEM PEMBAYARAN REKENING LISTRIK
MELALUI JARINGAN GPRS**

SKRIPSI

Disusun Oleh :

AHMAD ZAMAHSYARI HARDIANSYAH

NIM. 0310633003

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 21 Oktober 2009

Majelis Penguji

Ali Mustofa, ST., MT

NIP. 19710601 200003 1 001

Adhâarul Muttaqin, ST., MT

NIP. 19760121 200501 1 001

Himawat Aryadita, ST., M.Sc

NIP. 19801018 200801 1 003

Mengetahui,

Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc

NIP. 19710615 199802 1 003

APLIKASI SISTEM PEMBAYARAN REKENING LISTRIK MELALUI JARINGAN GPRS

Ahmad Zamahsyari Hardiansyah (0310633003)

Dosen Pembimbing: Arief Andy Soebroto, ST., Mkom dan Herman Tolle, ST., MT

ABSTRAK

Teknologi WAP (*Wireless Application Protocol*) merupakan sebuah metode yang memungkinkan internet dapat diakses oleh telepon seluler. Dengan menggabungkan pemrograman WML (*Wireless Markup Language*) dengan pemrograman PHP, akan menghasilkan aplikasi WAP yang dinamis, sehingga dapat mengakses *database* dan menampilkan informasinya kedalam telepon seluler *user*. Salah satu aplikasi dengan teknologi WAP adalah *M-Commerce (Mobile Commerce)*, yaitu suatu metode transaksi finansial dengan menggunakan telepon seluler berbasis GPRS. Dengan *M-Commerce* transaksi tersebut dapat dilakukan dimanapun dan kapanpun menggunakan telepon seluler, selama masih terdapat jaringan GPRS.

Sistem pembayaran rekening listrik ini dikembangkan melalui beberapa tahap, yaitu: analisis, desain, implementasi dan pengujian. Sistem ini menyediakan dua buah portal dengan user yang berbeda, yaitu WAP portal yang diakses oleh *user* melalui telepon seluler berbasis GPRS, dan WEB portal yang diakses oleh pihak manajemen untuk melihat laporan yang dibutuhkan, dan mengisi data-data yang berhubungan dengan pembayaran rekening listrik. Sistem pembayaran rekening listrik ini dikembangkan dengan menggunakan bahasa pemrograman WML untuk WAPsitenya, PHP untuk WEBSitenya, MySQL sebagai databasenya, dan web *server* Apache.

Proses pengujian terhadap perangkat lunak sistem pembayaran rekening listrik dilakukan melalui dua tahap, yaitu pengujian algoritma dengan menggunakan teknik pengujian *white box (white box testing)* dan pengujian validasi dengan menggunakan teknik pengujian *black box (black box testing)*. Hasil akhir aplikasi ini berupa sistem pembayaran rekening listrik dengan simulasi sistem pembayaran berupa bank virtual.

Kata kunci: GPRS, WAP, WML, telepon seluler, *M-Commerce*

KATA PENGANTAR

Puji Syukur kehadirat Allah SWT karena dengan berkat rahmat dan karunia serta ridho-Nya penyusunan skripsi dengan judul “Aplikasi Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS” dapat diselesaikan. Penulis menyadari bahwa kajian ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Ayah dan Ibu atas segala jerih payah, cinta kasih sayang, dan kesabarannya dalam memberikan dukungan materiil, doa dan semangat kepada Ananda.
2. Bapak Rudy Yuwono, ST, MSc selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
3. Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Bapak Arief Andy Soebroto, ST., Mkom selaku dosen pembimbing pada penyusunan skripsi ini.
5. Bapak Herman Tolle, ST., MT selaku dosen pembimbing pada penyusunan skripsi ini.
6. Bapak dan Ibu dosen serta karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
7. Teman-teman Teknik Elektro Universitas Brawijaya, khususnya Angkatan 2003 yang telah memberi kenangan yang tak terlupakan.
8. Teman-teman di Bendungan Nawangan, khususnya Elham, Andi Babe, Udi, dan Tante Arif Cobain yang selalu memberi doa dan dukungan selama Ananda di Malang.
9. Dewi Nurlaily Putri atas perhatian dan kasih sayangnya selama ananda menyelesaikan skripsi ini.
10. Firman, Arif Grands, Arif Bud, Bu Messi, dan Elvira yang telah memberi warna dan pelajaran dalam hidup.
11. Serta semua pihak yang tak dapat disebutkan satu persatu yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Tiada yang sempurna di dunia ini, tersadar bahwa skripsi ini sangat jauh dari kesempurnaan. Karenanya, segala kritik dan saran yang sifatnya membangun dari pembaca tentang isi skripsi ini akan diterima dengan senang hati. Akhir kata, penulis berharap, semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, Oktober 2009

Penyusun



DAFTAR ISI

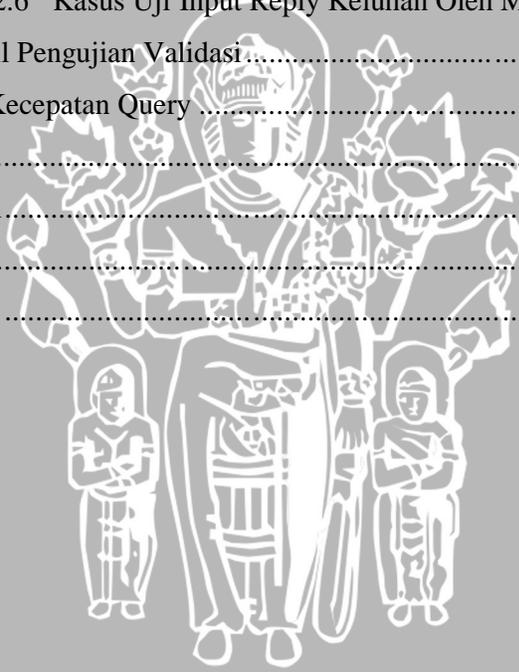
	Halaman
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
I. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
II. TINJAUAN PUSTAKA	5
2.1 Penerapan GPRS Pada Situs Pembayaran Rekening Listrik	5
2.2 Sistem Informasi	6
2.2.1 Konsep Dasar Sistem	6
2.2.2 Konsep Dasar Informasi	6
2.2.3 Konsep Dasar Sistem Informasi	7
2.3 GPRS (<i>General Packet Radio Service</i>).....	8
2.4 WAP (<i>Wireless Application Protocol</i>).....	11
2.5 WML (<i>Wireless Mark-up Language</i>).....	13
2.5.1 Elemen <i>Deck</i> dan <i>Card</i>	15
2.5.2 Elemen <i>Event</i>	15
2.5.3 Elemen <i>Task</i>	16
2.5.4 Elemen <i>Variable</i>	17
2.5.5 Elemen <i>User Input</i>	17
2.5.6 Elemen <i>Anchor</i>	17
2.5.7 Elemen <i>Image</i>	17
2.5.8 Elemen <i>Format Text</i>	18
2.6 PHP	19

2.7	<i>Apache Web Server</i>	20
2.8	Basis Data.....	20
2.9	DFD (<i>Data Flow Diagram</i>).....	26
III.	METODE PENELITIAN	27
3.1	Studi Literatur.....	27
3.2	Analisis Kebutuhan.....	27
3.3	Perancangan.....	28
3.4	Implementasi	28
3.5	Pengujian.....	28
3.6	Pengambilan Kesimpulan dan Saran	29
3.7	Penulisan Laporan.....	29
IV.	PERANCANGAN PERANGKAT LUNAK.....	30
4.1	Analisis Sistem Pembayaran Rekening Listrik	32
4.1.1	Analisis Proses Bisnis.....	32
4.1.2	Analisis Kebutuhan Perangkat Lunak	34
4.1.2.1	Spesifikasi Pengguna Perangkat Lunak	35
4.1.2.2	Spesifikasi Perangkat Lunak	38
4.1.3	Diagram Blok Sistem.....	40
4.1.4	Perancangan Perangkat Lunak	41
4.3.1	Data Flow Diagram	41
4.3.1.1	Context Diagram.....	41
4.3.1.2	DFD Level 1	46
4.3.1.3	DFD Level 2	54
4.3.1.4	DFD Level 3	60
4.3.2	Perancangan Basis Data	73
4.3.2.1	Normalisasi Data	74
4.3.2.1.1	Normalisasi Pertama	74
4.3.2.1.2	Normalisasi Kedua	74
4.3.2.1.3	Normalisasi Ketiga	75
4.3.2.2	Data Object Description	75
4.3.2.2.1	Data Type Definition	75
4.3.2.2.2	Data Field Description	78

4.3.3	Site Map Sistem	80
4.3.4	Perancangan Antarmuka User	81
4.3.4.1	Antarmuka User dan Anggota	81
4.3.4.2	Antarmuka User Admin	85
4.3.4.3	Antarmuka User Manajemen	88
4.3.4.4	Antarmuka User Admin Perangkat Lunak	90
V.	IMPLEMENTASI	91
5.1	Lingkungan Implementasi	92
5.2	Algoritma Pemrograman	92
5.2.1	Proses Koneksi	93
5.2.2	Proses Cek Session	95
5.2.3	Proses Login	96
5.2.4	Proses Input Data	97
5.2.5	Proses Hapus Data	98
5.2.6	Proses Pembayaran	99
5.2.7	Proses Laporan Transaksi	100
5.3	Implementasi Antarmuka	102
5.3.1	Implementasi Antarmuka WAP Anggota dan Pengunjung	102
5.3.1.1	Implementasi Login Anggota	102
5.3.2	Implementasi Antarmuka Admin	103
5.3.2.1	Implementasi Input Data	104
5.3.3	Implementasi Antarmuka Manajemen	104
5.3.4	Implementasi Antarmuka Admin Perangkat Lunak	105
VI.	PENGUJIAN	106
6.1	Pengujian Algoritma	106
6.1.1	Pengujian Proses Login	106
6.1.2	Pengujian Proses Input Data	108
6.1.3	Pengujian Proses Hapus Data	109
6.2	Pengujian Validasi	111
6.2.1	Kasus Uji Validasi Sistem WAP Pembayaran Rekening Listrik	111
6.2.1.1	Kasus Uji Login	111



6.2.1.2	Kasus Uji Logout	113
6.2.1.3	Kasus Uji Daftar Anggota Baru	114
6.2.1.4	Kasus Uji Pembayaran Tagihan Listrik	114
6.2.1.5	Kasus Uji Keluhan	116
6.2.2	Kasus Uji Validasi Sistem WEB Pembayaran Rekening Listrik.....	117
6.2.2.1	Kasus Uji Login Admin	117
6.2.2.2	Kasus Uji Login Manajemen	118
6.2.2.3	Kasus Uji Login Admin Perangkat Lunak	119
6.2.2.4	Kasus Uji Input Data Tagihan Rekening Listrik	120
6.2.2.5	Kasus Uji Lihat Data Laporan Tagihan	121
6.2.2.6	Kasus Uji Input Reply Keluhan Oleh Manajemen	122
6.2.3	Hasil Pengujian Validasi.....	123
6.3	Pengujian Kecepatan Query	124
PENUTUP		128
7.1	Kesimpulan	128
7.2	Saran	129
DAFTAR PUSTAKA		130



DAFTAR GAMBAR

	Halaman
Gambar 2.1	Arsitektur Umum Jaringan GPRS 8
Gambar 2.2	Komponen Arsitektur <i>Wireless Application Protocol</i> 12
Gambar 2.3	Contoh Struktur Basis Data Hierarki 23
Gambar 2.4	Contoh Struktur Basis Data Jaringan 24
Gambar 2.5	Contoh Struktur Basis Data Relasional 25
Gambar 4.1	Diagram Pohon Perancangan Sistem 30
Gambar 4.2	<i>Standart Operational Procedure</i> (SOP) manual 31
Gambar 4.3	<i>Standart Operational Procedure</i> (SOP) elektronik 32
Gambar 4.4	Diagram Blok Sistem Aplikasi Pembayaran Rekening Listrik 40
Gambar 4.5	<i>Context Diagram</i> 42
Gambar 4.6	Data Flow Diagram Level 1 48
Gambar 4.7	DFD Level 2 Proses WAP 55
Gambar 4.8	DFD Level 2 Proses WEB 57
Gambar 4.9	DFD Level 3 Proses Daftar 61
Gambar 4.10	DFD Level 3 Proses Validasi Anggota 62
Gambar 4.11	DFD Level 3 Proses Olah Data Pembayaran 64
Gambar 4.12	DFD Level 3 Proses WEB Validasi Login Admin 67
Gambar 4.13	DFD Level 3 Proses WEB Olah Data Admin 69
Gambar 4.14	DFD Level 3 Proses WEB Olah Data Manajemen 71
Gambar 4.15	DFD Level 3 Proses WEB Olah Data Admin Perangkat Lunak 72
Gambar 4.16	Diagram Entitas Relasional 73
Gambar 4.17	Normalisasi Database 74
Gambar 4.18	<i>Site Map</i> Sistem Antarmuka WAP 80
Gambar 4.19	<i>Site Map</i> Sistem Antarmuka WEB 80
Gambar 4.20	Halaman Index 81
Gambar 4.21	Halaman Daftar Anggota Baru 82
Gambar 4.22	Halaman Login Anggota 82
Gambar 4.23	Halaman Konfirmasi Login 83
Gambar 4.24	Halaman Pembayaran Tagihan Listrik 83



Gambar 4.25	Halaman Simulasi Tagihan	83
Gambar 4.26	Halaman History Tagihan	84
Gambar 4.27	Halaman Informasi Pasang Baru	84
Gambar 4.28	Halaman Informasi Pemadaman	84
Gambar 4.29	Halaman Keluhan Bagi Anggota	85
Gambar 4.30	Halaman Login Admin	85
Gambar 4.31	Halaman Input dan Edit Data	86
Gambar 4.32	Halaman Input Data Pelanggan	86
Gambar 4.33	Halaman Input Data Tagihan	86
Gambar 4.34	Halaman Input Data Pemadaman	87
Gambar 4.35	Halaman Informasi Pasang Baru	87
Gambar 4.36	Halaman Edit Password	88
Gambar 4.37	Halaman Manajemen	88
Gambar 4.38	Halaman Laporan Data Pelanggan	89
Gambar 4.39	Halaman Laporan Data Tagihan	89
Gambar 4.40	Halaman Laporan Keluhan	90
Gambar 4.41	Halaman Edit Password	90
Gambar 4.42	Halaman Admin Perangkat Lunak	91
Gambar 4.43	Halaman Edit Data Admin	91
Gambar 5.1	Diagram Pohon Implementasi	92
Gambar 5.2	Diagram Alir Algoritma Koneksi	93
Gambar 5.3	Diagram Alir Algoritma Cek Session	95
Gambar 5.4	Diagram Alir Algoritma Validasi Login Admin	96
Gambar 5.5	Diagram Alir Algoritma Input Tagihan	97
Gambar 5.6	Diagram Alir Algoritma Hapus Data Tagihan	98
Gambar 5.7	Diagram Alir Algoritma Proses Pembayaran	99
Gambar 5.8	Diagram Alir Algoritma Laporan Pembayaran	101
Gambar 5.9	Implementasi Halaman Index	102
Gambar 5.10	Implementasi Halaman Login Anggota	103
Gambar 5.11	Implementasi Fasilitas Login Admin	103
Gambar 5.12	Implementasi Menu Utama Admin	104
Gambar 5.13	Implementasi Input Data Pelanggan	104



Gambar 5.14	Implementasi Menu Utama Manajemen	105
Gambar 5.15	Implementasi Menu Utama Admin Perangkat Lunak	105
Gambar 6.1	<i>Flow Graph</i> Proses Login	107
Gambar 6.2	<i>Flow Graph</i> Proses Input Data	108
Gambar 6.3	<i>Flow Graph</i> Proses Hapus Data	110
Gambar 6.4	Validasi Proses Login	111
Gambar 6.5	Validasi Proses Login Berhasil	112
Gambar 6.6	Validasi Proses Login Gagal	112
Gambar 6.7	Validasi Proses Logout	113
Gambar 6.8	Kembali Ke Menu Utama	113
Gambar 6.9	Validasi Proses Daftar Anggota	114
Gambar 6.10	Input Nomer Rekening	115
Gambar 6.11	Proses Pembayaran Berhasil	115
Gambar 6.12	Input Keluhan	116
Gambar 6.13	Reply Keluhan	116
Gambar 6.14	Validasi Login Admin	117
Gambar 6.15	Menu Admin	118
Gambar 6.16	Validasi Login Manajemen	119
Gambar 6.17	Menu Admin Manajemen	119
Gambar 6.18	Validasi Login Admin Perangkat Lunak	120
Gambar 6.19	Menu Admin Perangkat Lunak	120
Gambar 6.20	Input Data Tagihan	121
Gambar 6.21	Laporan Data Tagihan	122
Gambar 6.22	Reply Keluhan Oleh Manajemen	122
Gambar 6.23	Hasil Pengujian Query	125
Gambar 6.24	Grafik Rata-rata Waktu Akses Query	126



DAFTAR TABEL

	Halaman
Tabel 4.1 Data Type Definition Tabel Admin_access	75
Tabel 4.2 Data Type Definition Tabel Bank	75
Tabel 4.3 Data Type Definition Tabel Biaya_pemakaian	76
Tabel 4.4 Data Type Definition Tabel Golongan_daya	76
Tabel 4.5 Data Type Definition Tabel Informasi_pemasangan	76
Tabel 4.6 Data Type Definition Tabel Informasi_pemadaman	76
Tabel 4.7 Data Type Definition Tabel Keluhan	77
Tabel 4.8 Data Type Definition Tabel Pelanggan	77
Tabel 4.9 Data Type Definition Tabel Tagihan	77
Tabel 4.10 Data Field Description Basis data listrik_online	78
Tabel 6.1 Kasus Uji Pengujian Proses Login	107
Tabel 6.2 Kasus Uji Pengujian Proses Input Data	109
Tabel 6.3 Kasus Uji Pengujian Proses Hapus Data	110
Tabel 6.4a Kasus Uji Login Valid	111
Tabel 6.4b Kasus Uji Login Gagal	112
Tabel 6.5 Kasus Uji Logout	113
Tabel 6.6 Kasus Uji Daftar Anggota Baru	114
Tabel 6.7 Kasus Uji Pembayaran	114
Tabel 6.8 Kasus Uji Keluhan	116
Tabel 6.9 Kasus Uji Validasi Login Admin	117
Tabel 6.10 Kasus Uji Validasi Login Manajemen	118
Tabel 6.11 Kasus Uji Validasi Login Admin Perangkat Lunak	119
Tabel 6.12 Kasus Uji Input Data Tagihan	120
Tabel 6.13 Kasus Uji Lihat Laporan Data Tagihan	121
Tabel 6.14 Kasus Uji Input Reply Keluhan	122
Tabel 6.15 Hasil Pengujian Sistem WAP	123
Tabel 6.16 Hasil Pengujian Sistem WEB	123
Tabel 6.17 Tabel Pengujian Waktu Akses Query	125
Tabel 6.18 Tabel Perbandingan Waktu Akses Per Record	126

BAB I

PENDAHULUAN

I.1 Latar Belakang

Dewasa ini kebutuhan masyarakat akan listrik semakin meningkat. Hal itu ditunjukkan dengan semakin banyaknya antrian yang terjadi pada loket pembayaran, terutama pada masa jatuh tempo pembayaran tagihan listrik. Kejadian ini dapat mengurangi kenyamanan pelanggan, dengan membuang-buang waktu hanya untuk mengantri. Masyarakat tentunya ingin secara cepat melakukan pembayaran tagihan listrik tanpa harus datang dan mengantri di loket. Dapat disimpulkan bahwa, masyarakat membutuhkan suatu sistem sebagai alat bantu untuk mendukung kenyamanan pelanggan saat melakukan transaksi pembayaran.

Sistem ini akan menerapkan penggunaan media internet yang dapat diakses menggunakan telepon seluler, PDA (Personal Digital Assistance), smartphone dan sebagainya. Hal ini dapat dilakukan dengan adanya wireless internet access yang memungkinkan pengaksesan internet dengan menggunakan alat-alat tersebut.

Perkembangan teknologi ini didukung penuh oleh para produsen telepon seluler, dengan mengembangkan teknologi 2,5G yang menggunakan gelombang radio sebagai pertukaran datanya. Beberapa teknologi 2,5G yang dikembangkan adalah GPRS dan CDMAOne. Penggunaan GPRS memungkinkan pelanggan dapat menciptakan fasa hubungan lebih cepat, karena kecepatan transfer datanya mencapai 115 Kbps, sedangkan CDMAOne hanya 64 Kbps [JUN-06].

Teknologi GPRS (*General Packet Radio Service*) memungkinkan kita mengakses internet dalam keadaan bergerak, dimanapun selama sinyal GPRS tersebut ada, dan terhubung secara permanen [LIN-04:04].

WAP (*Wireless Application Protocol*) merupakan protokol bagi perangkat-perangkat nirkabel yang menyediakan layanan komunikasi data bagi pengguna, baik dalam bentuk yang berhubungan dengan telekomunikasi maupun aplikasi-aplikasi berorientasi internet. Struktur WAP mengadopsi topologi layer-layer yang ada pada *Internet Protocol* (model TCP/IP). Ini terkait dengan tujuan dibuatnya WAP, yaitu memberikan akses *Internet* bagi alat komunikasi *mobile* nirkabel [KOM-03:01].

Untuk menyelesaikan permasalahan tersebut diatas, maka skripsi ini membahas tentang aplikasi pengaksesan pembayaran rekening listrik dengan menggunakan protocol WAP melalui jaringan GPRS. Sistem ini memiliki kemampuan dapat diakses dimana saja, kapan saja, dan dengan biaya yang relatif lebih murah. Protokol WAP digunakan sebagai jembatan untuk pertukaran data antara *user* dengan *server*. Sedangkan jaringan GPRS merupakan media pembawa data *request user* menuju *server*, begitu juga sebaliknya.

Kendala yang ditemui adalah kelancaran koneksi antara *server* WAP dengan telepon seluler. Kelancaran koneksi juga dipengaruhi oleh jenis telepon seluler yang digunakan dan kondisi trafik jaringan GPRS operator yang digunakan.

Aplikasi ini akan dibangun dengan menggunakan PHP, dan WML. PHP sebagai *software* untuk membaca data pada *database*. WML digunakan untuk merancang tampilan aplikasi ini.

I.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijelaskan pada bagian latar belakan g, maka rumusan masalah yang diberikan adalah:

1. Merancang dan membuat diagram *entity relationship* dari Aplikasi Sistem Pembayaran Rekening Listrik
2. Merancang dan membuat aplikasi sistem informasi berbasis WAP dengan menggunakan bahasa pemrograman WML, PHP, dan basisdata MySQL.
3. Merancang dan membuat sebuah aplikasi untuk mempermudah mengakses, memberikan informasi tagihan listrik, simulasi, dan informasi yang terkait dengan PT. PLN
4. Mengimplementasikan dan menguji sistem yang telah dirancang.

I.3 Batasan Masalah

Batasan masalah diperlukan dengan tujuan agar pembahasan yang dilakukan tidak menyimpang jauh dari tujuan. Batasan masalah dari sistem ini adalah:

- Membuat aplikasi untuk mengakses menu-menu yang ada dalam pembayaran rekening listrik, antara lain : informasi transaksi rekening listrik, simulasi tagihan listrik, dan informasi yang terkait dengan PT.PLN

- Menggunakan telepon seluler yang terdapat layanan WAP serta menggunakan jaringan GPRS.
- Bahasa pemrograman yang digunakan PHP dan WML
- Database yang digunakan adalah *database virtual* dengan mengacu kepada PT PLN Malang

I.4 Tujuan

Tujuan akhir dari penulisan ini adalah merancang dan membangun sistem aplikasi pembayaran rekening listrik dengan menggunakan telepon seluler melalui jaringan GPRS dengan layanan WAP.

I.5 Manfaat

1. Bagi penulis

Mengembangkan ilmu pengetahuan yang diperoleh dan mempraktekkan untuk menyelesaikan permasalahan yang timbul dalam keadaan sebenarnya.

2. Bagi masyarakat

mempermudah dan mempercepat proses pembayaran yang terjadi di loket pembayaran sehingga pelanggan merasa nyaman.

I.6 Sistematika Penulisan

Pembahasan Tugas Akhir ini terdiri dari 7 bab yang disusun dengan sistematika penulisan sebagai berikut :

Bab I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan serta sistematika penulisan

Bab II Dasar Teori

Membahas teori-teori yang menunjang dalam pembuatan program aplikasi ini.

Bab III Metodologi

Membahas metodologi yang digunakan dalam penulisan yang terdiri dari studi literatur, perancangan, implementasi, pengujian dan analisa, serta pengambilan kesimpulan dan saran

Bab IV Perancangan

Membahas perancangan program yang sesuai dengan teori yang ada

Bab V Implementasi

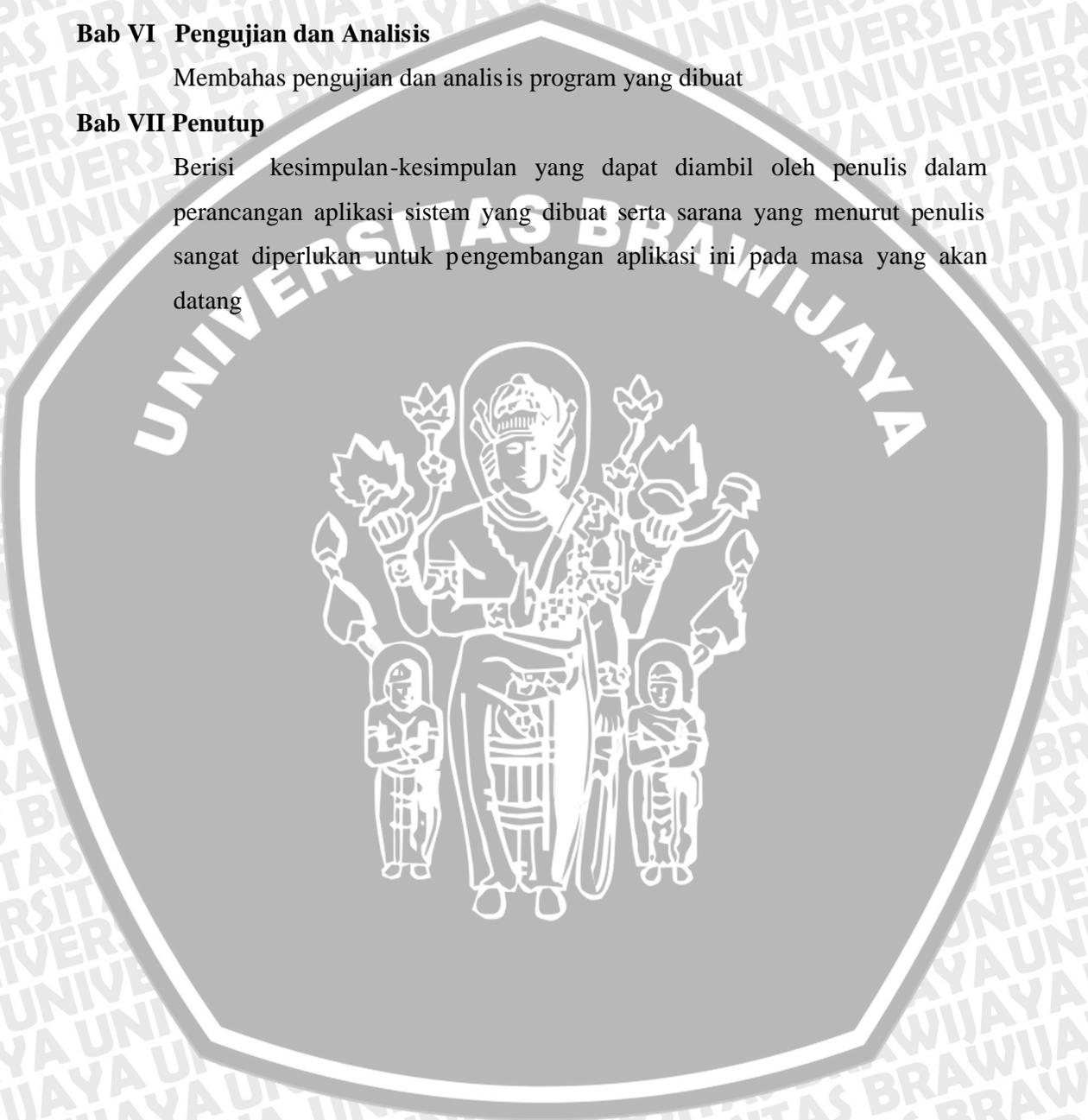
Membahas lingkungan implementasi, batasan implementasi, file-file implementasi, algoritma operasi yang diimplementasikan

Bab VI Pengujian dan Analisis

Membahas pengujian dan analisis program yang dibuat

Bab VII Penutup

Berisi kesimpulan-kesimpulan yang dapat diambil oleh penulis dalam perancangan aplikasi sistem yang dibuat serta sarana yang menurut penulis sangat diperlukan untuk pengembangan aplikasi ini pada masa yang akan datang



BAB II

TINJAUAN PUSTAKA

Pada bab ini membahas kajian pustaka dan dasar teori yang digunakan untuk penulisan skripsi dan pengembangan aplikasi Melalui Jaringan GPRS. Kajian pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan skripsi ini. Kajian pustaka yang dipaparkan adalah mengenai penerapan GPRS sebagai pertukaran data pada situs pembayaran rekening listrik.

Dasar teori yang diperlukan berdasar kajian pustaka untuk penyusunan skripsi ini diantaranya adalah teori tentang sistem informasi, GPRS, WAP, WML, PHP, *Apache Web Server*, basis data.

2.1 Penerapan GPRS pada situs pembayaran rekening listrik

Sebelum ada pengembangan transmisi data lewat GPRS, transmisi data GSM sangat lambat, hal ini dikarenakan kanal radionya yang bersifat tunggal dan berkecepatan rendah, dan diperuntukkan khusus bagi setiap pengguna data selama durasi komunikasi [LIN-04:04].

GPRS menggunakan teknologi *packet switching* memungkinkan semua pengguna dalam sebuah sel dapat berbagi sumber-sumber yang sama. Dengan kata lain pelanggan menggunakan spektrum radio hanya ketika benar-benar mentransmisikan data. Efisiensi penggunaan spektrum pada akhirnya berarti kinerja yang lebih baik dan biaya yang rendah. GPRS dapat mentransmisikan laju data sampai 115 Kbps [LIN-04:04].

GPRS disebut teknologi 2.5 G karena merupakan langkah awal menuju teknologi transfer data kecepatan tinggi lewat jaringan nirkabel (3G). Sehingga sering disebut-sebut sebagai teknologi kunci untuk data bergerak. Secara rinci ada beberapa faktor yang menjadi pertimbangan bahwa GPRS merupakan teknologi kunci untuk data bergerak, yakni [LIN-04:04]:

- mampu memanfaatkan kemampuan cakupan global yang dimiliki GSM (2G)
- memperkaya utiliti investasi untuk perangkat GSM yang sudah ada
- merupakan teknologi jembatan yang bagus menuju generasi ke 3

- berbasis paket data yang lebih efisien dalam penggunaan sumber daya
- memiliki laju data sampai 115 kbps yang berarti dua kali lipat daripada koneksi 'dialup' 56 kbps yang berlaku

Teknologi GPRS memiliki 3 fitur keunggulan, yaitu [JUN-06]:

- a. *Allways Online*. GPRS menghilangkan mekanisme dial kepada pengguna pada saat ingin mengakses data, sehingga dikatakan GPRS selalu online karena transfer data dikirim berupa paket dan tidak bergantung pada waktu koneksi.
- b. *An Upgrade to existing networks (GSM dan TDMA)*. Adopsi sistem GPRS tidak perlu menghilangkan sistem lama karena GPRS dijalankan di atas infrastruktur yang telah ada.
- c. *An Integral part of EDGE and WCDMA*. GPRS merupakan inti dari mekanisme pengiriman paket data untuk teknologi 3G selanjutnya.

Dari kajian tersebut maka judul skripsi yang dikemukakan adalah “Aplikasi Pembayaran Rekening Listrik Melalui Jaringan GPRS”.

2.2 Sistem Informasi

Sistem informasi dapat didefinisikan sebagai kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah *input* menjadi *output* guna mencapai sasaran-sasaran perusahaan [ANO-04:04].

2.2.1 Konsep Dasar Sistem

Definisi sistem yang lebih menekankan pada prosedurnya adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan tertentu [ANO-04:01].

Karakteristik sistem yaitu mempunyai komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah/proses, dan sasaran atau tujuan [ANO-04:01].

2.2.2 Konsep Dasar Informasi

Definisi informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya [ANO-04:02].

Kualitas informasi tergantung pada tiga hal yaitu [ANO-04:02]:

- a. Akurat, berarti informasi harus bebas dari kesalahan dan tidak menyesatkan bagi orang yang menerima informasi tersebut.

- b. Tepat waktu, berarti informasi yang diterima harus tepat pada waktunya, sebab informasi yang usang tidak mempunyai nilai yang baik.
- c. Relevan, berarti informasi harus mempunyai manfaat bagi yang menerima.
- d. Ekonomis, berarti informasi yang dihasilkan mempunyai manfaat yang lebih besar dibandingkan dengan biaya mendapatkannya.

2.2.3 Konsep Dasar Sistem Informasi

Sebuah informasi diperoleh dari sistem informasi (*information system*). Sistem informasi didefinisikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung informasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan [ANO-04:04].

Komponen sistem informasi yang disebut dengan blok bangunan yaitu [ANO-04:04]

- a. Blok masukan, blok masukan harus mewakili data yang masuk ke dalam sistem informasi, termasuk metode dan media untuk memperoleh data yang akan dimasukkan.
- b. Blok model, blok model terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data masukan dan data yang tersimpan dalam basis data untuk menghasilkan keluaran yang diinginkan.
- c. Blok keluaran, blok keluaran dari sistem informasi adalah keluaran berupa informasi yang berkualitas.
- d. Blok teknologi, blok teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi, perangkat lunak, dan perangkat keras.
- e. Blok basis data, blok basis data merupakan kumpulan dari *file* data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat diakses dengan mudah dan cepat.
- f. Blok kendali, pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan bisa langsung diatasi.

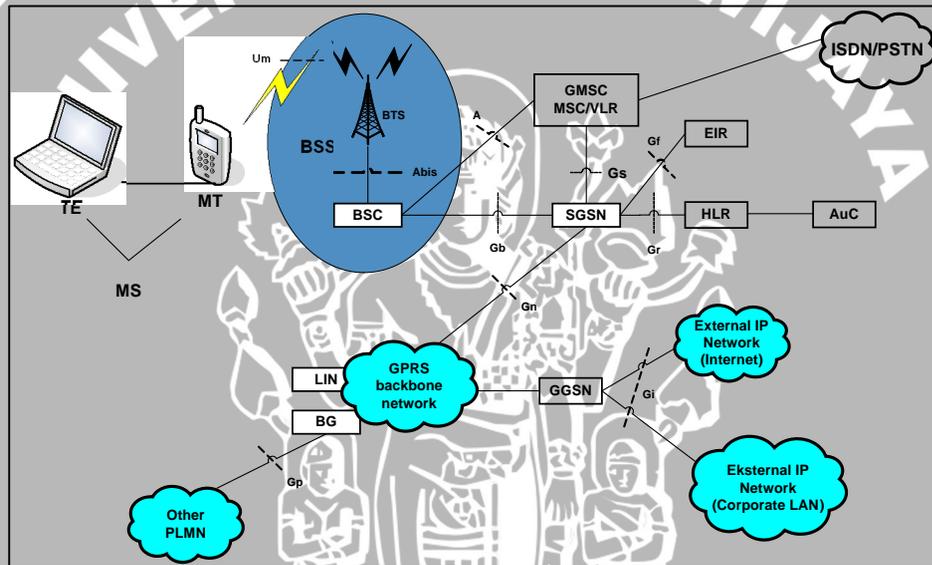
2.3 GPRS (*General Packet Radio Service*)

GSM (*Global System for Mobile Communications*) yang memiliki transmisi 9,6 Kbps tidak cocok digunakan sebagai transmisi data kecepatan tinggi. Terbukti GSM

hanya dapat mentransmisikan data sebesar 160 karakter dalam *Short Message Service* (SMS). Maka digandenglah GPRS salah satu standar komunikasi data *wireless non voice* [MAR-08:06].

Dengan adanya GPRS ini pengguna bisa terus terkoneksi ke Internet. Pengguna tidak perlu *dial up* terus menerus ketika akan melakukan koneksi ke Internet. Tagihan Internet tidak berdasar lama waktu penggunaan Internet namun berdasar banyaknya data yang dikirim atau diterima [LIN-04:04].

Dalam gambar di bawah ini terlihat bahwa jaringan GPRS merupakan bagian dari jaringan GSM (beberapa bagian dalam jaringan GPRS dipakai untuk komunikasi suara). Berikut penjelasan bagian-bagian dalam gambar tersebut [LIN-04:05]:



Gambar 2.1 Arsitektur Umum Jaringan GPRS

Sumber: [LIN-04:06]

- Mobile Station (MS)**

MS dapat dikatakan perangkat selular yang terhubung langsung dengan jaringan GSM, yaitu *SIM (Subscriber Identify Module) Card* dan perangkat keras seperti telepon selular, PDA, perangkat komputer yang terhubung menggunakan jaringan GPRS.
- Base Station System (BSS)**

BSS terdiri dari *BTS (Base Transceiver Station)* dan *BSC (Base Station Controller)*. BSC menyediakan semua fungsi control dan relasi fisik antara MSC (*Mobile Switching Centre*) dan BTS. BTS menangani *interface* radio pada *mobile station*. Sekumpulan BTS dikendalikan oleh sebuah BSC.

- **Gateway Mobile Switching Centre (GMSC)**
GMSC menyediakan fungsi pembatas diantara PLMN (*Public Line Mobile Network*). GMSC mematikan format *traffic* dan pensinyalan PSTN (*Public Switched Telephone Network*) dan mengubahnya menjadi protokol pada jaringan *mobile*.
- **Public Switched Telephone Network (PSTN)**
PSTN merupakan istilah umum untuk layanan dan jaringan telepon.
- **Home Location Register (HLR)**
HLR adalah basis data yang menyimpan data pengguna jaringan GPRS.
- **Visitor Location Register (VLR)**
VLR adalah basis data yang berisi informasi semua MS yang sedang terhubung dengan GPRS.
- **Serving GPRS Support Node (SGSN)**
SGSN adalah komponen utama jaringan GPRS. SGSN akan meneruskan paket data dari/ke MS.
- **Gateway GPRS Support (GGSN)**
GGSN juga merupakan komponen utama jaringan GPRS. GGSN mengubah paket data GSM dari SGSN menjadi paket TCP/IP.
- **Equipment Identity Register (EIR)**
EIR adalah basis data yang berisi data tentang perangkat bergerak.
- **Authentication Center (AuC)**
AuC adalah basis data yang berisi informasi pengguna yang diperbolehkan memakai jaringan GPRS. AuC merupakan bagian dari HLR.
- **GPRS backbone networks**
GPRS *backbone network* adalah Intranet dari jaringan GPRS.
- **Public Land Mobile Network (PLMN)**
PLMN adalah sistem komunikasi *wireless* yang ditujukan di kendaraan atau pejalan kaki.
- **Local Interconnect Network (LIN)**
LIN adalah sistem bus yang digunakan didalam arsitektur jaringan otomotif (yang berhubungan dengan kendaraan).

- **Border Gateway (BG)**

BG adalah *server* yang menjadi perantara antara dua operator.

Secara teori kecepatan pengiriman data GPRS dapat mencapai 171,2 kbps.

Namun dalam implementasinya maksimal hanya mencapai 115 Kbps, hal ini sangat tergantung dari berbagai hal seperti [MAR-08:07]:

- a. Konfigurasi dan Alokasi *time slot* di *level* Radio/BTS
- b. Teknologi *software* yang digunakan
- c. Dukungan ponsel (kelas GPRS yang digunakan)
- d. Cuaca

Kelas GPRS terbagi menjadi dua, yaitu kelas slot dan kelas servis. Kelas slot direpresentasikan dengan angka *downlink* dan *uplink* (data transfer yang mampu didukung oleh ponsel dari dan ke jaringan) angka *downlink/uplink* satu mempunyai kecepatan antara 8-12 Kbps, dua 16-24 Kbps dan seterusnya berkelipatan. Rumus kategori kelas slot ini adalah sbb: GPRS Kelas X (*downlink+uplink*). GPRS Kelas 1 (1+1) GPRS Kelas 2 (2+1), GPRS Kelas 4 (3+1), GPRS Kelas 6 (3+2)/(2+3), GPRS Kelas 8 (4+1), GPRS Kelas 10 (4+1)/(3+2), GPRS Kelas 12 (4+1)/(3+2)/(2+3)/(1+4) [MAR-08:07].

2.4 WAP (*Wireless Application Protocol*)

WAP (*Wireless Application Protocol*) adalah suatu *protocol* aplikasi yang memungkinkan internet dapat diakses oleh ponsel dan perangkat *wireless* lainnya. WAP membawa informasi secara *online* melewati internet langsung menuju ke ponsel atau *client* WAP lainnya. Dengan adanya WAP, berbagai informasi dapat diakses setiap saat hanya dengan menggunakan telepon seluler [MAR-08:08].

Pengembangan aplikasi WAP dilakukan dalam suatu lingkungan kerja yang disebut *Wireless Application Environment* (WAE). Inti dari WAE ini terdiri dari *Wireless Markup Language* (WML) dan pemrograman *script* seperti *Wireless Markup Language Script* (WMLScript) [KOM-03:03].

Untuk menjangkau dunia Internet, sebuah ponsel dengan teknologi WAP harus berjalan *via* WAP Gateway. WAP Gateway ini bertindak sebagai perantara, menghubungkan jaringan *mobile* dengan Internet dengan menerjemahkan *Hypertext Transfer Protocol* (HTTP) menjadi *Wireless Session Protocol* (WSP) [KOM-03:03].

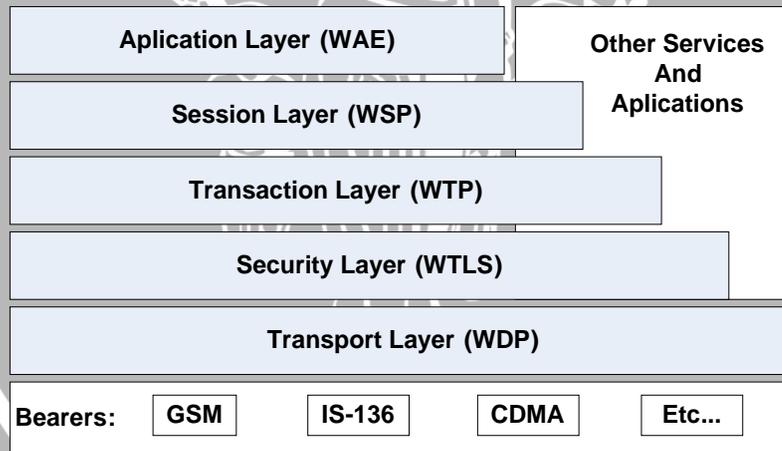
Struktur WAP mengadopsi topologi *layer-layer* yang ada pada *Internet Protocol* (model TCP / IP). Ini terkait dengan tujuan dibuatnya WAP, yaitu memberikan akses Internet bagi alat komunikasi *mobile* nirkabel [KOM-03:01].

Protokol mengatur bagaimana format paket data dan layanan – layanan terhadap paket data pada setiap *layer*; bagaimana suatu *layer* memberikan layanan kepada *layer* yang lain [KOM-03:01].

Protokol dapat didefinisikan sebagai suatu cara atau aturan yang dibakukan untuk melakukan komunikasi antara satu peralatan jaringan (komputer, hub, dll) dengan peralatan jaringan lainnya [SAL-06:02].

Setiap protokol yang dibuat untuk aplikasi jaringan biasanya mengikuti standar untuk mendesainnya. Desain atau standar protokol tersebut dinamakan arsitektur protokol yang ditetapkan secara internasional oleh ISO (*International Standard Organization*). Arsitektur tersebut dinamakan *Open System Reference Standard Architecture* (OSI Reference Model) [SAL-06:02].

Arsitektur WAP tidak jauh berbeda dengan konsep protokol jaringan yang terdiri dari tujuh layer yang berfungsi untuk mendefinisikan tahapan-tahapan dalam desain protokol. Tahapan-tahapan dalam arsitektur dari WAP terdiri atas [MAR-02:05]:



Gambar 2.2 komponen arsitektur *Wireless Application Protocol*

Sumber: [MAR-02:05]

- ❖ **Wireless Application Environment (WAE)**

WAE memiliki fungsi dasar untuk menggabungkan *World Wide Web* (WWW) dengan teknologi telepon seluler. WAE berisi *Micro Browser* yang berfungsi sebagai pendukung antarmuka pemrograman dan layanan telepon, WML, WMLscript [MAR-02:06]

- ❖ **Wireless Session Protocol (WSP)**

WSP berfungsi sebagai pembuka atau mengakhiri suatu koneksi jaringan. WSP terdiri atas dua *session*. Pertama adalah *connection oriented* yaitu koneksi yang menghususkan untuk berinteraksi dengan operasi yang terjadi pada Transaction Layer Protocol (WTP). Kedua adalah koneksi yang berhubungan dengan keamanan data [MAR-02:06].

- ❖ **Wireless Transaction Protocol (WTP)**

WTP berfungsi memeriksa format data, konversi data atau pengkodean yang akan ditransmisikan [MAR-02:06].

- ❖ **Wireless Transport Layer Security (WTLS)**

WTLS adalah protocol untuk keamanan data yang disesuaikan oleh standar industry *Transport Layer Security* yang mendukung *Secure Socket Layer* [MAR-02:07].

- ❖ **Wireless Datagram Protocol (WDP)**

WDP bertugas mentransmisikan data dalam format biner melalui media gateway, serta mendefinisikan pengalamatan jaringan yang akan dikenali oleh bearer [MAR-02:07].

- ❖ **Bearers**

Terdiri dari data *switch*, *short message*, dan data paket yang memiliki fungsi untuk melakukan transfer data dari suatu unit informasi yang berisi alamat dan melakukan *error* dan penundaan transfer hingga proses benar [MAR-02:07].

2.5 WML (*Wireless Mark-up Language*)

WML merupakan bahasa standar yang digunakan untuk membuat suatu situs berbasis WAP. Dengan penggunaan WML maka proses pengiriman data dari WAP *server* menjadi lebih cepat dikarenakan proses konversi pada WAP *Gateway* menjadi lebih cepat dibandingkan bila menggunakan HTML (*Hypertext Mark-up Language*) yang merupakan bahasa standar pada pemrograman suatu WEB [MAR-08-11].

Sebuah halaman WML terdiri dari bagian *header* dan *body*. Dalam *header* terdapat dua hal yang harus dideklarasikan, yaitu deklarasi versi XML yang akan digunakan dan deklarasi *Document Type Definition* (DTD). Deklarasi DTD ini bertujuan agar tipe data dalam dokumen yang dibuat dapat dikenali dengan benar. Deklarasi *header* WML adalah [KOM-03:23]:

```
<?xml version="1.0">
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

File WML terdiri dari *tag-tag*. Semua *tag* WML memiliki atribut sebagai standar : *id*, yang digunakan sebagai referensi unik dalam sebuah *deck*, dan *class*, yang digunakan pada sisi *server* [KOM-03:23].

Bagian *body file* WML diawali dan diakhiri oleh pasangan *tag* `<wml>` dan `</wml>`. Pasangan *tag* ini disebut dengan *deck*. Dalam satu *deck* dapat terdiri dari satu *template* dan satu atau lebih *card*. *Tag template* digunakan sebagai cetak biru *event* semua *card* dalam satu *deck*. Sebuah *card* dapat berisi satu atau lebih komponen sebagai berikut [KOM-03:23]:

- Teks terformat, dapat berupa teks, gambar, dan link.
- Elemen `<input>`, untuk menerima input dari pengguna berupa *string*.
- Elemen `<select>`, user dapat memilih dari daftar pilihan.
- Elemen `<fieldset>`, bertindak sebagai kontainer bagi elemen lain.

Berikut ini contoh yang menunjukkan struktur program WML dengan satu *card* [NUR-03:23].

```
<?xml version="1.0"?>
<!DOCTYPE WML PUBLIC "-//WAPFORUM/DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1">
<wml>
<head>
<meta name="keywords" content="wml WAP Bandung Indonesia"/>
<meta name="description" content="pemrograman WML dan mls"/>
<meta name="author" content="Tyasno Nurhadi"/>
</head>
<card id="card1" title="HALO">
```

```
<p align="center">
```

```
Halo Dunia WAP
```

```
</p>
```

```
</card>
```

```
</wml>
```

Tiga baris pertama dari contoh di atas biasa disebut deklarasi program WML . Ketiganya harus selalu dituliskan di awal *deck* [NUR-03:23].

Baris selanjutnya, yaitu baris `<wml>`, juga harus selalu dituliskan dalam program WML, yang menandai awal dari program WML. Baris berikutnya merupakan *Heading* [NUR-03:24].

Di bawah *heading* barulah pembuat program menuliskan bagian inti dari program WML, dalam unit yang disebut dengan *card*. Dalam contoh diatas *card* diawali oleh baris `<card id="card1" title="HALO">` dan diakhiri dengan baris `</card>` [NUR-03:24].

2.5.1 Elemen *Deck* dan *Card*

WML didesain untuk *bandwidth* kecil, dan perangkat berpenampilan kecil sehingga dipakai konsep setumpuk kartu. Sebuah dokumen WML dianggap sebagai tumpukan kartu, sehingga memungkinkan lebih dari satu kartu bisa di *-download* oleh klien dalam sekali pengambilan data.

Elemen-elemen dalam *Deck* dan *Card* [KOM-03]:

1. Elemen WML

Elemen ini mewakili permulaan dan akhir dari suatu halaman dan mendefinisikan tumpukan kartu, membuka dan menutup semua informasi di dalam *deck*. Cara penulisan elemen ini, dibuka dengan sintaks `<wml>` dan ditutup dengan menuliskan `</wml>`.

2. Elemen *Template*

Elemen *template* ini mendeklarasikan ketetapan untuk semua *card* di dalam *deck*. Cara penulisan sintaksnya dibuka dengan `<template>` dan ditutup oleh `</template>`.

3. Elemen *Card*

Mengindikasikan tampilan yang keluar pada layar klien. Elemen ini mewakili permulaan dan akhir dari *card*. Cara penulisan elemen ini, dibuka dengan `<card>` dan ditutup dengan `</card>`.

4. Elemen *Head*

Elemen *head* memberikan informasi yang berkaitan kedalam *deck*. Penulisannya dibuka dengan `<head>` dan ditutup dengan `</head>`. Didalam elemen *head* terdapat elemen *meta* yang berfungsi untuk pengklarifikasian suatu *deck* ke dalam *search engine*.

2.5.2 Elemen *Event*

Beberapa elemen WML membuat suatu aksi dengan menggunakan elemen - elemen *event*.

Elemen – elemen *event* adalah [KOM-03]:

1. Elemen *do*

Sintaks ini memberikan pilihan pada *user* untuk melakukan *action*. Cara penulisan elemen ini dibuka dengan sintaks `<do>` dan diakhiri dengan sintaks `</do>`.

2. Elemen *Timer*

Elemen *timer* mendeklarasikan waktu pada kartu, dimana waktu nya mulai ketika memasuki *card*, dan berhenti setelah waktu yang ditentukan berakhir atau ketika keluar dari *card* dimana *timer* tersebut aktif.

3. Elemen *Onpick*

Dalam pilihan lebih dari satu, dibutuhkan sintaks *onpick*. Sehingga pengguna dapat memilih atau tidak memilih salah satu dari pilihan-pilihan tersebut. Penulisannya yaitu:

```
<option onpick="alamat">pilihan satu</option>
```

2.5.3 Elemen *Task*

Elemen *do* dapat menspesifikasikan *task* yang akan dijalankan pada klien, ketika pengguna bernavigasi pada klien menuju *deck* atau *card* yang lain.

Elemen – elemen *task* adalah [KOM-03]:

1. Elemen *Go*

Elemen ini akan menavigasikan klien ke lokasi yang telah ditentukan oleh *href*. Cara penulisannya adalah `<go href="alamatnya"/>`

2. Elemen *Prev*

Elemen ini menavigasikan kembali ke halaman sebelumnya. Elemen *prev* mengambil alamat sebelumnya dari fasilitas *history*. Cara penulisannya adalah `<prev/>`

3. Elemen *Refresh*

Elemen ini berfungsi untuk membuka kembali *card* yang terbuka, serta meng-*update* konteks yang muncul jika terjadi perubahan. Cara penulisan elemen ini yaitu `<refresh/>`

4. Elemen *Noop*

Elemen *noop* berarti *no operation*, elemen tidak mengerjakan apa-apa. Fungsi elemen ini jika ingin mengesampingkan efek elemen *template*. Cara penulisan elemen ini yaitu `<noop/>`.

2.5.4 Elemen *Variabel*

User dapat membuat *card* dinamis yang fleksibel, mengganti isi tampilan, dan menavigasi berdasarkan input. Variabel WML dapat digunakan sebagai pengganti *string-string* [KOM-03].

2.5.5 Elemen *User Input*

Elemen – elemen ini membantu mekanisme pengambilan input dari *user*.

Elemen – elemen ini adalah [KOM-03]:

1. Elemen *Input*

Elemen ini menspesifikasi masukan. Sehingga dapat menentukan format masukan yang diisi oleh *user* dan dapat mengisi *text* standar pada *field*.

2. Elemen *Select*

Elemen *select* diperlukan agar *user* dapat memilih pilihan – pilihan yang ada, baik pilihan tunggal ataupun pilihan lebih dari satu. Cara penulisan elemen ini dibuka dengan sintaks `<select>` dan ditutup dengan sintaks `</select>`

3. Elemen *Option*

Elemen ini berada didalam elemen *select*, yang menspesifikasikan pilihan tunggal yang dipilih *user*. Cara penulisan elemen ini dibuka dengan sintaks `<option>` dan ditutup dengan sintaks `</option>`

2.5.6 Elemen *Anchor*

Elemen *anchor* digunakan untuk menspesifikasikan *link*. Elemen ini mengarahkan *user* kepada *link* yang telah ditentukan. Cara penulisan elemen ini dibuka dengan sintaks `<anchor>` dan ditutup dengan sintaks `</anchor>` [KOM-03].

2.5.7 Elemen *Image*

Elemen *image* berguna untuk menampilkan gambar kedalam alur teks pada klien *user*. Gambar pada WML sudah ditentukan, yaitu berformat "wbmp".

Cara penulisan sintaksnya dibuka dengan `` dan ditutup ``. Elemen – elemen yang terdapat pada elemen *image* adalah : *alt*, *src*, *width* dan *height* serta *align* [KOM-03].

2.5.8 Elemen *Format Text*

Elemen yang terdapat pada elemen ini adalah [KOM-03]:

1. *Paragraf*

Elemen ini digunakan untuk memulai paragraf. Penulisan dibuka dengan `<p>` dan ditutup dengan `</p>`.

2. *Emphasis*

Emphasis diperlukan untuk memberi penekanan pada teks. Sintaks penekanan adalah sebagai berikut :

- *i*
Menampilkan jenis huruf miring.
- *b*
Menampilkan jenis huruf tebal.
- *u*
Menampilkan jenis huruf yang bergaris bawah.
- *big*
Menampilkan jenis huruf yang lebih besar.
- *small*
Menampilkan jenis huruf yang lebih kecil.

3. Karakter khusus

Karakter khusus dibutuhkan dalam beberapa penulisan WML. Berikut cara penulisan :

- Kutip dua ("")

Bisa ditulis dengan `"`; atau `"`;

- Ampersand (&)

Bisa ditulis dengan `&`; atau `&`;

- Kutip satu (')

Bisa ditulis dengan `'`; atau `'`;

- Lebih kurang dari (<)

Bisa ditulis dengan `<`; atau `<`;

- Lebih besar dari (>)

Bisa ditulis dengan `>`; atau `>`;

- Spasi

Bisa ditulis dengan ` `; atau ` `;

2.6 PHP

PHP merupakan sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat [SUT-03].

PHP ini mempunyai kelebihan diantaranya [SUT-03]:

1. *Life Cycle* yang singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
2. *Cross Platform*, PHP dapat dikenal di hampir semua web server yang ada, dengan demikian proses *developing* dapat dilakukan dengan menggunakan sistem operasi yang berbeda.
3. PHP bersifat *open source* atau gratis dan bebas pakai, selalu *up to date* sesuai dengan perkembangan.
4. PHP mendukung dan mampu berkomunikasi dengan berbagai jenis basis data meskipun dengan kelengkapan yang berbeda.
5. PHP termasuk bahasa yang *embedded* (bisa ditempel atau diletakkan dalam tag HTML).

PHP mengenal tiga macam representasi data, yaitu *integer*, *floating point* dan *string*. *Integer* menyatakan tipe data bilangan bulat, *floating point* lebih dikenal dengan nama *double* yang selalu dalam bentuk desimal. *String* diawali dengan tanda petik ganda (") atau petik tunggal (') [SUT-03].

Konstanta adalah variabel yang nilainya tetap. Pada PHP telah didefinisikan beberapa konstanta, antara lain `PHP_VERSION`, konstanta `TRUE` dan `FALSE`, `E_ERROR`, `E_WARNING`, dan `E_NOTICE` [SUT-03].

Untuk membuat program WAP yang berbasis PHP, maka setiap halaman di WAP harus ditambahkan dengan *header* pengenalan sebagai berikut [SUT-03]:

```
header('Content-type: text/vnd.wap.wml');
```

Selain itu perlu penambahan pengenalan *script* ke dalam versi penulisan PHP, yaitu dengan memberikan parameter *echo* sebagai berikut [SUT-03]:

```
echo('<?xml version="1.0"?>');  
echo('<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML1.1/EN>');
```

2.7 Apache Web Server

Web server adalah software yang menjadi tulang belakang dari *world wide web* (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Internet Explorer, Mozilla, dan program browser lainnya [AZM-07:01].

Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*) [AZM-07:01].

Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut. Contohnya, bila data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja [AZM-07:01].

Apache ialah *Web server* yang dikembangkan oleh kelompok tenaga sukarela Unix. Kelebihan dari Apache adalah *source code* tersedia dengan bebas, sehingga siapapun dapat mengadaptasikan *server* ini untuk kebutuhan yang spesifik. Dan juga tersedia pula *library* umum yang sangat besar untuk *add-on* Apache. Apache dapat berjalan hampir di semua sistem operasi utama seperti Unix, Windows NT, Linux, Windows 95, dan lain-lain [AZM-07].

2.8 Basis Data

Basis data merupakan komponen utama dalam membangun sebuah sistem yang menyangkut pendokumentasian data ke dalam sebuah *database*. Bentuk basis data adalah sebuah aturan yang mengatasi masalah tersebut. Saat ini basis data memiliki peranan yang sangat penting dalam mengelola data yang ada di dalamnya [NUG -04:70].

Basisdata adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan [NUG-04:71].

Dalam basisdata, data yang ada tidak hanya diletakkan dan disimpan begitu saja dalam sebuah media penyimpanan, akan tetapi dikelola dengan sebuah sistem pengaturan basisdata yang sering disebut dengan *Database Management System* (DBMS) [NUG-04:72].

DBMS akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama serta menunjang keakuratan atau konsistensi data dan sebagainya. Contoh perangkat lunak DBMS adalah: *Oracle, SQL Server 2000/2003, MS Access, MySQL* dan sebagainya [NUG-04:74].

Beberapa alasan mengapa DBMS merupakan komponen penting dalam *Database System* [MUL-07]:

- DBMS memberikan solusi manajemen data yang lebih efisien dan efektif.
- Penggunaan bahasa *query* yang didukung oleh banyak aplikasi memberikan kecepatan akses data.
- Membantu *end-user* dalam mengakses lebih banyak data yang terorganisir secara baik.
- Memberikan solusi yang lebih baik untuk masalah inkonsistensi data.

Seperti yang telah diterangkan diatas bahwa DBMS merupakan program aplikasi yang mengatur proses antara *end-user* dengan kumpulan data yang dimilikinya [MUL-07].

Secara teknis proses basis data dapat diakses dari media *storage* oleh basis data dibedakan atas 2 proses [MUL-07]:

1. Proses Disk Manager

Merupakan proses dimana penyimpanan data terkait dengan bagaimana sesungguhnya data disimpan pada media disk. Pada proses ini apabila direpresentasikan maka kita akan berurusan dengan data yang bersifat data magnetik atau berupa bit-bit data. Proses ini proses ditangani oleh *disk manager*. *Disk manager* merupakan komponen yang biasanya terdapat ditingkat sistem operasi. Komponen ini bertanggung jawab terhadap semua operasi yang terkait dengan *Input Output (I/O)*.

2. Proses File Manager

Merupakan proses yang menggambarkan data secara fungsional kepada *end-user*. *End-user* diberikan suatu representasi data yang tersimpan dalam basis data sebagai suatu kesatuan *file*, *table* atau *record*. Pada level ini *file manager* berperan menjembatani hak *end-user* terhadap sekumpulan data yang dikelola oleh DBMS tersebut.

Fungsi DBMS antara lain : [MUL-07]

- Manajemen kamus data
- Manajemen penyimpanan data
- Transformasi dan penyediaan jasa
- Manajemen keamanan
- Kontrol akses *multi-user*
- Manajemen *back-up* dan *recovery*
- Manajemen integritas data
- Bahasa akses basis data (DDL & DML) dan antarmuka aplikasi
- Antarmuka komunikasi basis data

Dalam basis data dikenal istilah Model basis data. Model basis data adalah kumpulan dari konsepsi basis data yang biasanya mewakili struktur dan relasi data yang terdapat pada suatu basis data. Esensi sebuah Model basis data adalah tempat dimana data atau suatu metodologi untuk menyimpan data. Model basis data tidak dapat dilihat tetapi algoritma yang digunakan oleh model basis data tersebut dapat dilihat [MUL-07].

Ada 2 macam Model basis data : [MUL-07]

1. Model Konseptual

Model Konseptual terfokus kepada representasi basis data secara logika. Model ini lebih memperhatikan tentang apa yang disajikan dibanding dengan bagaimana cara menyajikannya.

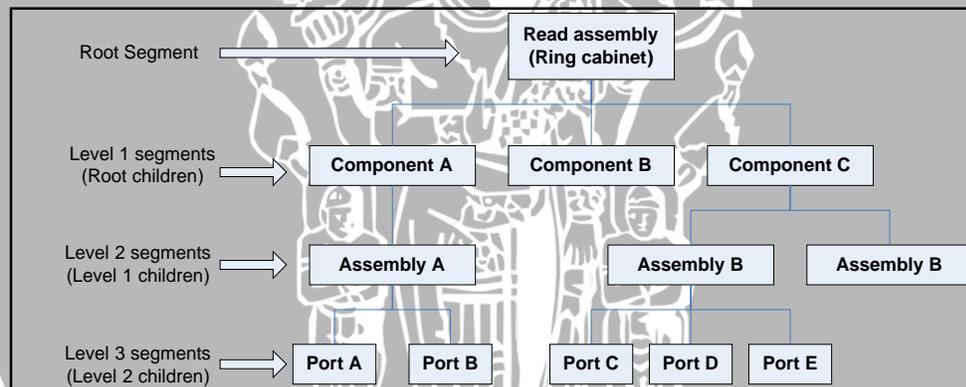
2. Model Implementasi

Ditekankan pada bagaimana cara data disajikan pada basis data atau bagaimana struktur data diimplementasikan

Dari konsep Model basis data implementasi terdapat beberapa konsep basis data yang berkembang antara lain [MUL-07]:

a. Model Basis Data Hierarki (*Hierarchical Database*)

Sistem basis data hierarki merupakan konsep model basis data yang tertua, tidak ada kepastian kapan konsep ini mulai digunakan. Model ini berupa suatu *tree* dengan relasi *Parent Child Relationships* dengan hubungan satu-banyak (1-N).



Gambar 2.3. Contoh Struktur Basis Data Hierarki

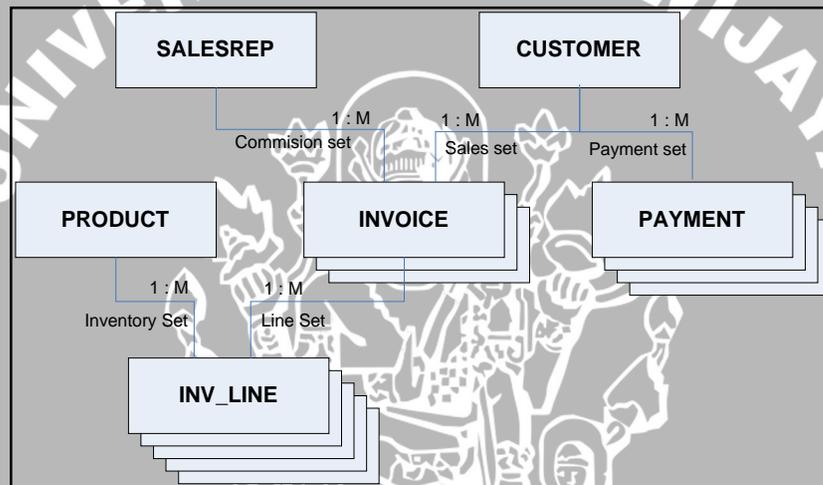
Sumber: [MUL-07:01]

Struktur dasar basis data hierarki :

- Kumpulan *record-record* yang secara logika terorganisir seperti struktur pohon dari atas ke bawah (berbentuk hierarki). Model ini banyak digunakan pada saat awal komputer *database mainframe*. Sistem ini banyak digunakan pada tahun 50-an dan 60-an, yang banyak digunakan oleh bank dan lembaga asuransi pada masa itu.

- Lapisan paling atas bertindak sebagai induk/ *root* dari segmen yang tepat berada di bawahnya dan lapisan bawah tidak bisa memiliki lebih dari satu *root*.
 - Segmen yang berada di bawah dari suatu segmen lainnya merupakan anak dari segmen yang ada di atasnya.
 - Struktur pohon mewakili urutan hierarki dari media penyimpanan pada komputer.
- b. Model Basis Data Jaringan (*Network Database*)

Model basis data ini dikemukakan pada tahun 1969 oleh *CODASYL Consortium*. Pada prinsipnya model basis data jaringan hampir sama dengan basis data hierarki yaitu berupa model *tree*, akan tetapi pada model basis data jaringan *child* dapat memiliki lebih dari satu *parent*.



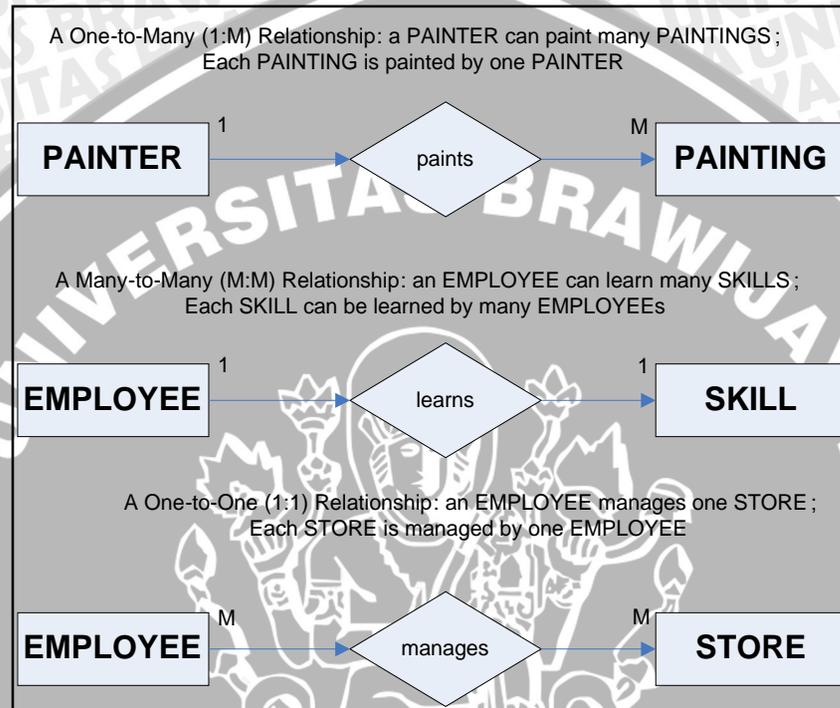
Gambar 2.4. Contoh Struktur Basis Data Jaringan
 Sumber: [MUL-07:02]

Struktur data basis data jaringan :

- *Set* - Sebuah hubungan disebut set. Setiap set terdiri dari paling tidak dua macam *record* : satu *record parent* (induk) dan satu *record child* (anak).
- Satu set mewakili satu hubungan *1:M* antara *parent* dan *child*.

c. Model Basis Data Relasional

Model basis data relasional merupakan model basis data yang dirancang agar memiliki konsistensi informasi dalam bentuk normalisasi basis data, yang secara implementatif dan operasional dikendalikan oleh mesin *Database Management System* (DBMS).



Gambar 2.5. Contoh Struktur Basis Data Relasional

Sumber: [MUL-07:03]

Struktur dasar basis data relasional [MUL-07]:

- *Relational Database Management System* (RDBMS) beroperasi pada lingkungan logika manusia.
- Basis data relasional diasumsikan sebagai sekumpulan tabel -tabel.
- Setiap tabel terdiri dari serangkaian per -potongan baris/kolom
- Tabel-tabel (atau relasi) terhubung satu dengan lainnya menggunakan entitas tertentu yang digunakan secara bersama
- Tipe hubungan seringkali ditunjukkan dalam suatu skema
- Setiap tabel menghasilkan data yang lengkap dan kebebasan struktural

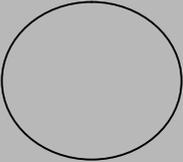
2.9 DFD (Data Flow Diagram)

Diagram aliran data (*data flow diagram* (DFD)) adalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output [KRIS-04:66].

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan alat yang digunakan untuk menggambarkan arus data terstruktur pada metodologi pengembangan sistem yang terstruktur dan jelas [KRIS-04:66].

DFD yang pertama kali digambar adalah level teratas (*top level*) yaitu diagram konteks atau level 0. Dari diagram konteks kemudian akan digambar dengan lebih terinci lagi ke diagram level-n. N dimulai dari 1, 2 sampai tiap-tiap proses tidak dapat digambar lebih rinci lagi.

Simbol-simbol yang digunakan dalam DFD adalah [KRIS-04:66]:

1.  Kesatuan Luar : Merupakan kesatuan (*entity*) di lingkungan luar sistem yang akan memberikan *input* atau *output* dari sistem.
2.  Arus Data : Menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.
3.  Proses : Kegiatan atau kerja yang dilakukan oleh orang atau komputer dari hasil suatu arus data yang masuk untuk arus data yang akan keluar dari proses.
4.  Penyimpanan data : Untuk menggambarkan suatu tempat penyimpanan data.

BAB III

METODE PENELITIAN

Pada bab ini akan dibahas berbagai tahapan penelitian yang digunakan untuk menyelesaikan tugas akhir. Diantaranya adalah studi literatur, metode untuk melakukan perancangan, implementasi dan pengujian. Pada akhir proses penyelesaian akan ditarik kesimpulan dan saran untuk proses perbaikan ke depan.

3.1 Studi Literatur

Mengumpulkan sumber-sumber bacaan yang relevan, dengan tujuan untuk mendapatkan landasan teori yang mendukung pembuatan situs pembayaran rekening listrik melalui jaringan GPRS. Kajian pustaka dan teori-teori tersebut diantaranya adalah:

- a. Sistem Informasi
- b. *General Packet Radio Service* (GPRS)
- c. *Wireless Application Protocol* (WAP)
- d. *Wireless Mark-up Language* (WML)
- e. PHP
- f. *Apache Web Server*
- g. Basis Data
- h. *Data Flow Diagram* (DFD)

3.2 Analisis Kebutuhan

Tahap ini diperlukan untuk menjembatani kebutuhan perangkat lunak dengan perancangan perangkat lunak. Tahap analisis kebutuhan terdiri dari dua tahapan yaitu abstraksi sistem, dan daftar kebutuhan sistem.

Abstraksi sistem merupakan deskripsi dari sistem yang diinginkan. Daftar kebutuhan dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional. Pada tahap ini dibuat *entity relationship* dan *Data Flow Diagram* dari sistem.

3.3 Perancangan

Perancangan akan dilakukan dengan melakukan desain arsitektur untuk mendapatkan arsitektur dari sistem yang akan dibangun. Metode yang digunakan untuk perancangan adalah metode analisis terstruktur dengan menggunakan pemodelan DFD. Pada proses perancangan juga akan dilakukan perancangan dari kebutuhan-kebutuhan yang telah didefinisikan sebelumnya, pembuatan *site-map* dan perancangan *user-interface*.

3.4 Implementasi

Tahap implementasi akan mengubah perancangan yang telah dilakukan menjadi algoritma, *script-script* WML dan PHP. Metode yang digunakan adalah pemrograman WML dan PHP disertai dengan basis data MySQL. Alat bantu yang digunakan adalah *emulator* WAP, XAMPP yang merupakan perpaduan antara Apache, PHP, dan MySQL.

3.5 Pengujian

Pengujian dilakukan untuk menemukan kesalahan aplikasi yang dibuat, sehingga sebelum aplikasi yang dihasilkan digunakan oleh pengguna diharapkan didapatkan sesedikit mungkin atau bahkan tidak ditemukan kesalahan. Pengujian yang dilakukan meliputi pengujian *database* dan DFD menggunakan *software Visible Analyst Database Engineer v7.6.0.0*; pengujian implementasi aplikasi yaitu berupa pengujian dari analisis kebutuhan yang telah dibuat sebelumnya, sehingga dapat diketahui bahwa perangkat lunak dapat memenuhi kebutuhan yang telah ditentukan dan pengujian kecepatan *query* yaitu merupakan pengujian kecepatan sistem dalam mengakses data pada *database*. Metode pengujian yang digunakan adalah metode pengujian *white box* dan *black box*.

3.5 Pengambilan kesimpulan dan saran

Berisi kesimpulan-kesimpulan yang diperoleh dari hasil perancangan, implementasi, dan pengujian untuk menjawab perumusan masalah yang telah disebutkan pada bab pendahuluan serta saran yang menurut penulis sangat diperlukan untuk pengembangan aplikasi ini pada masa yang akan datang.

3.6 Penulisan Laporan

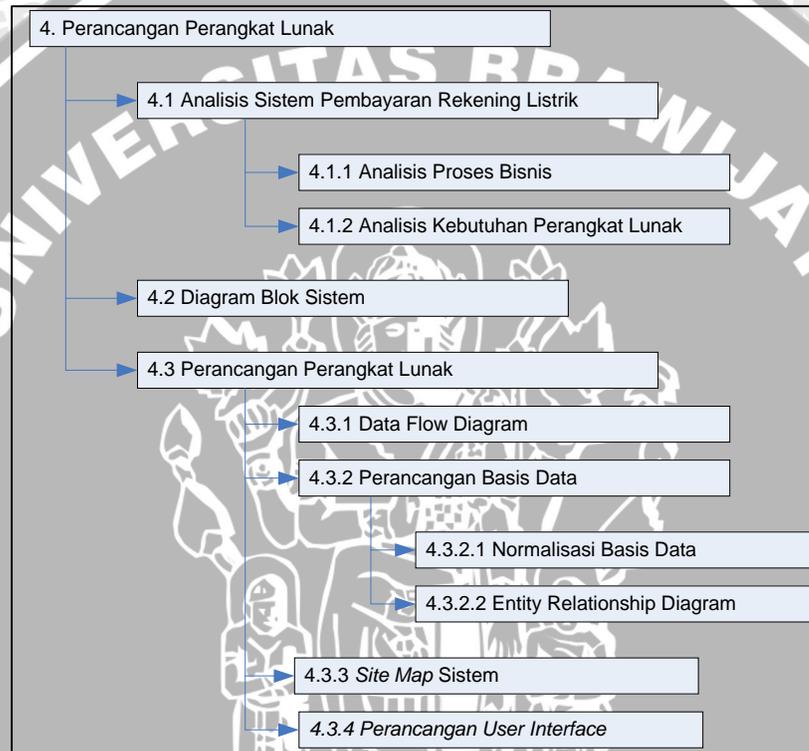
Merupakan kegiatan penulisan laporan terhadap Tugas Akhir yang dibuat yang diharapkan dapat dimanfaatkan bagi pihak-pihak lain dan tidak menutup kemungkinan dapat dilakukan perbaikan dari saran-saran yang diajukan.



BAB IV

PERANCANGAN PERANGKAT LUNAK

Bab ini menjelaskan analisis dan perancangan aplikasi sistem pembayaran Rekening Listrik Melalui Jaringan GPRS. Perancangan yang dilakukan meliputi dua tahap. Pada tahap pertama dilakukan analisa kebutuhan perangkat lunak dan analisis proses bisnis. Diagram pohon perancangan ditunjukkan dalam gambar 4.1

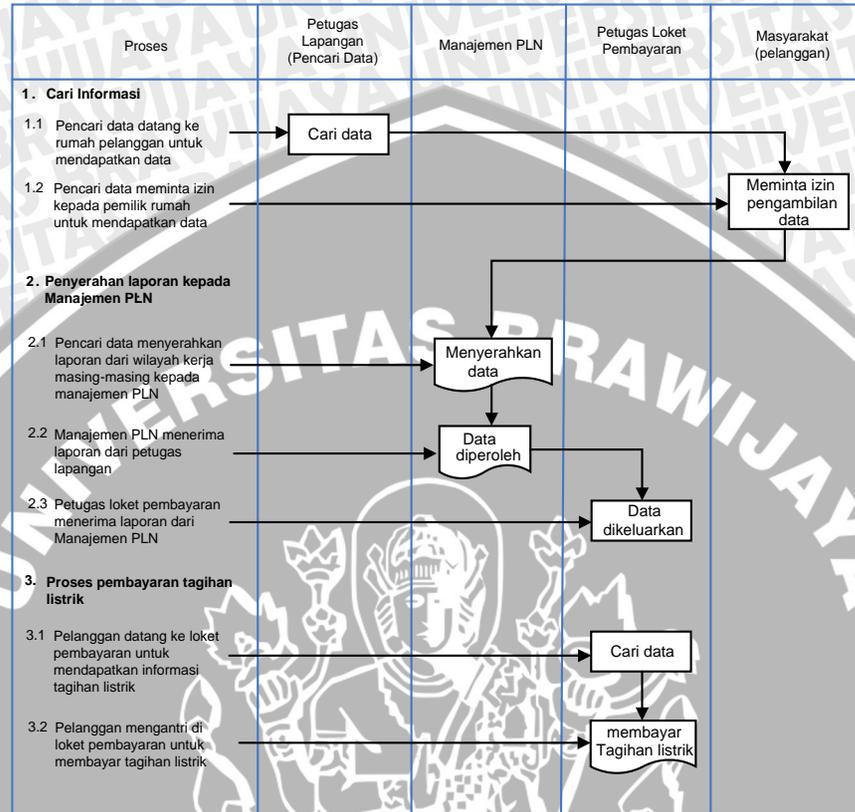


Gambar 4.1 Diagram Pohon Perancangan Sistem

Sumber : *Perancangan*

Standard Operational Procedure (SOP) adalah suatu prosedur standar yang harus dilakukan dalam kegiatan pembayaran tagihan listrik. SOP sistem ini terdiri dari SOP manual dan SOP elektronik.

SOP manual sistem pembayaran tagihan listrik ditunjukkan seperti pada gambar 4.2 dibawah ini:



Gambar 4.2 SOP manual sistem pembayaran

Sumber : PT. PLN kab. Malang

Keterangan:

1. Cari Informasi

- 1.1 Pencari data datang ke rumah pelanggan untuk mendapatkan data yang diperlukan, yaitu berupa angka yang ditunjuk pada KWHmeter .
- 1.2 Pencari data meminta izin kepada pemilik rumah untuk mendapatkan data yang diperlukan.

2. Penyerahan laporan kepada manajemen PLN

- 2.1 Pencari data menyerahkan laporan dari wilayah kerja masing -masing kepada manajemen PLN.
- 2.2 Manajemen PLN menerima laporan dari petugas lapangan, kemudian akan diproses lebih lanjut untuk menentukan besar biaya tagihan pelangg an.

2.3 Petugas loket pembayaran menerima laporan dari manajemen PLN berupa besarnya tagihan pelanggan.

3. Proses pembayaran tagihan listrik

3.1 Pelanggan datang ke loket pembayaran untuk mendapatkan informasi tentang tagihan yang harus dibayarkan.

3.2 Pelanggan mengantri di loket pembayaran untuk membayar tagihan listrik.

4.1. Analisis Sistem Pembayaran Rekening Listrik

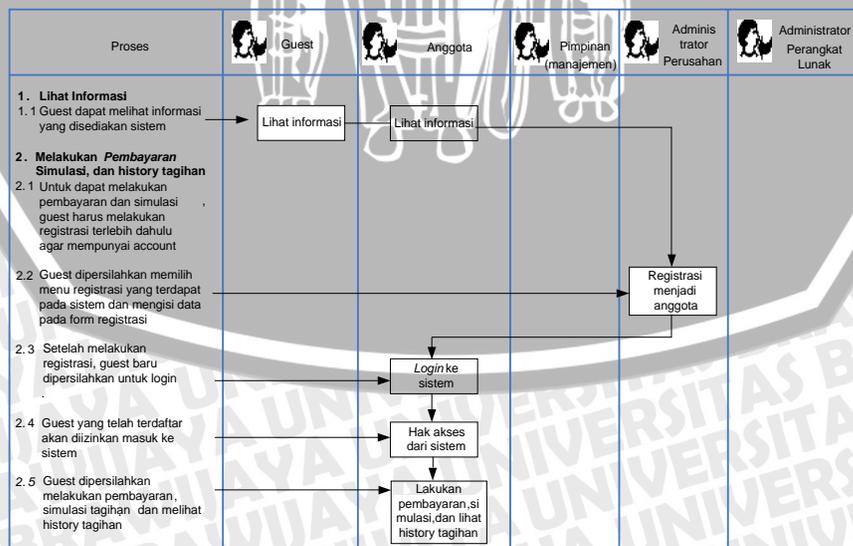
Analisis dibutuhkan untuk menjembatani antara kebutuhan perangkat lunak yang dibuat dengan perancangan, sehingga perangkat lunak yang dihasilkan dapat diimplementasikan.

Proses analisis dalam pembuatan sistem ini meliputi penjelasan cara kerja sistem yang akan dirancang dalam bentuk analisis proses bisnis, dan analisis kebutuhan perangkat lunak.

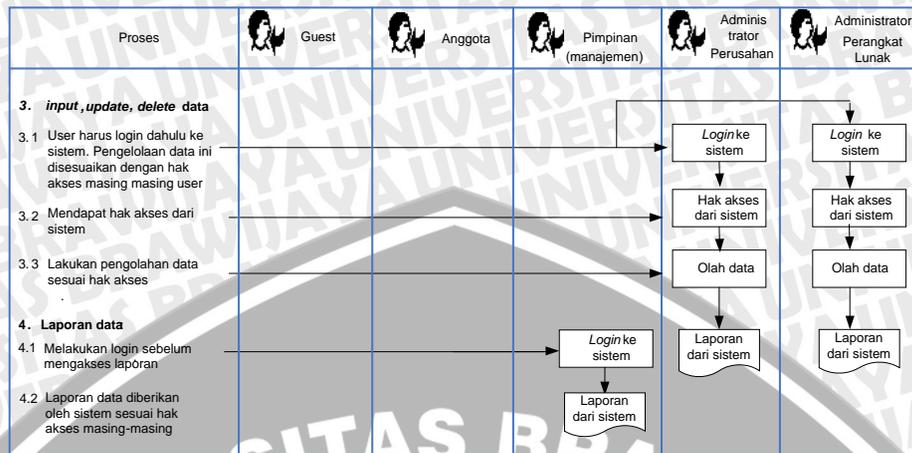
4.1.1 Analisis Proses Bisnis

Analisis proses bisnis dibutuhkan untuk memahami alur kerja dari SOP manual yang telah dibuat sebelumnya. Proses bisnis aplikasi pembayaran rekening listrik berupa SOP elektronik melalui telepon genggam berbasis WAP, agar proses-proses berikutnya sesuai dengan alur kerja yang telah dibuat sebelumnya.

SOP elektronik sistem pembayaran tagihan listrik ditunjukkan seperti pada gambar 4.3 dibawah ini:



[bersambung]



Gambar 4.3 SOP elektronik sistem pembayaran

Sumber : perancangan

Keterangan :

1. Lihat Informasi

- 1.1 *Guest* yang bersifat pengunjung adalah *user* yang akan mendapatkan informasi yang ditampilkan oleh sistem untuk *user* secara umum
- 1.2 *Guest* ini tidak disyaratkan mempunyai kualifikasi khusus, yaitu individu siapa saja yang ingin memperoleh informasi.

2. Melakukan pembayaran dan simulasi

- 2.1 Untuk dapat melakukan pembayaran dan simulasi, *guest* harus melakukan *registrasi* agar terdaftar menjadi anggota, dan mempunyai *account* untuk *login* ke sistem.
- 2.2 *Guest* dipersilahkan memilih menu *Registrasi*, dan mengisi data-data pada form *Registrasi*.
- 2.3 Setelah data diisi lengkap, maka sistem akan menyimpannya dan *guest* bisa mulai *login*.
- 2.4 *Guest* masuk ke dalam sistem dengan menggunakan *username* dan *password* yang diisikan pada saat *registrasi* tadi. Jika *username* dan *password* yang ditulis benar, maka *guest* yang akan memperoleh hak akses dari sistem.
- 2.5 Setelah masuk ke dalam sistem, *guest* dipersilahkan melakukan pembayaran, simulasi tagihan, dan melihat history tagihan sebelumnya.
- 2.6 *Guest* yang akan melakukan *registrasi* (daftar) merupakan individu, bisa siapa saja yang ingin menjadi anggota sistem dan mengakses informasi.

- 2.7 *Guest* yang akan menjadi anggota, tidak disyaratkan mempunyai kualifikasi khusus.
3. *Input, update, dan delete data*
 - 3.1 *User* yang akan melakukan pengelolaan data (meliputi *input, update, delete*) harus *login* terlebih dahulu.
 - 3.2 Setelah mengisikan *username* dan *password* dengan benar, maka *user* akan mendapat hak akses dari sistem.
 - 3.3 Setelah masuk ke dalam sistem, proses *maintenance* data dapat dilakukan. Proses ini disesuaikan dengan hak akses masing-masing.
 - 3.4 *User* yang dapat melakukan *input, update, dan delete* data adalah pegawai PLN dengan kategori manajemen dan administrator.
 - 3.5 Pengelola data bertanggung jawab terhadap data-data yang akan ditampilkan kepada pengunjung situs.
4. Laporan data
 - 4.1 Untuk melihat laporan yang ditampilkan oleh sistem, *user* dengan kategori pimpinan, petugas lapangan, dan administrator harus melakukan *login*.
 - 4.2 Setelah mendapat hak akses, maka *user* dapat mengakses laporan dari sistem sesuai dengan hak aksesnya masing-masing.

4.1.2 Analisis Kebutuhan Perangkat Lunak

Untuk mendapatkan informasi dan membayar tagihan listrik yang dapat diakses melalui Internet, akan dirancang dan dibuat aplikasi penyedia layanan pembayaran rekening listrik berbasis *web*. Aplikasi ini dapat diakses secara *online* melalui *web* menggunakan telepon genggam yang mempunyai fasilitas layanan W AP dan terdapat jaringan GPRS.

Pada saat ini sistem yang digunakan pada PLN untuk melakukan pembayaran tagihan rekening listrik dilakukan secara manual, yaitu datang langsung ke PLN yang bersangkutan.

Berdasarkan hasil survey tersebut, maka untuk menciptakan alternatif sistem pembayaran tagihan secara *online*, didapatkan kebutuhan yang secara spesifik dijelaskan sebagai berikut:

1. Semua *user* (kecuali pengunjung) harus melakukan *login* untuk dapat mengakses sistem.

2. *User* dibagi menjadi dua bagian:
 - a. *User* yang mengakses sistem pembayaran ini melalui *interface web*, yaitu admin perusahaan, manajemen, dan admin perangkat lunak.
 - b. *User* yang mengakses sistem pembayaran ini melalui *interface WAP*, yaitu anggota dan pengunjung.
3. Admin perusahaan dapat memasukkan informasi yang berkaitan dengan PLN.
4. Kepala PLN selaku manajemen dapat mengetahui informasi tagihan listrik, informasi anggota, laporan pembayaran rekening listrik.
5. Admin perangkat lunak dapat memperoleh informasi anggota, memblokir *user*, dan mengganti *password user* (anggota, admin perusahaan, manajemen).
6. Anggota dapat melakukan pembayaran tagihan rekening listrik, melakukan simulasi tagihan listrik, memberikan keluhan kepada manajemen, dan melihat data-data tagihan sebelumnya. Anggota juga dapat memperoleh informasi ID pelanggan, beban tagihan listrik, dan pemakaian listrik.
7. Implementasi sistem ini dibuat dengan bahasa pemrograman web *server-side* yaitu PHP dengan *database* MySQL serta Apache *Web Server* sebagai *server*,
8. Sistem yang dibuat menggunakan *database* yang sudah dinormalisasi sebelumnya.
9. Sistem operasi yang digunakan adalah sistem operasi Windows.
10. Pengaksesan sistem pembayaran rekening listrik oleh anggota dilakukan dengan *browsing* menggunakan telepon genggam yang mempunyai fasilitas layanan WAP dan terdapat jaringan GPRS.
11. Sistem yang dibuat harus terdapat *site map*.

4.1.2.1 Spesifikasi Pengguna Perangkat Lunak

Perangkat lunak tidak lepas dari interaksi pengguna (*user*) dan sistem. Pada sistem ini, *user* dapat digolongkan menjadi lima macam, yaitu: pengunjung, anggota, admin perusahaan, admin perangkat lunak, manajemen. Kelima macam *user* tersebut mempunyai hak akses yang berbeda.

Hak akses dari tiap *user* dapat dijelaskan dengan definisi dan spesifikasi sebagai berikut:

a. Pengunjung

Pengunjung adalah semua pengguna yang membuka aplikasi ini, tetapi hanya memiliki hak untuk mendapatkan informasi yang telah disediakan. Pengunjung tidak perlu melakukan login untuk mengakses aplikasi.

Hak akses yang diberikan adalah sebagai berikut:

Definisi:

1. Pengunjung dapat mengakses Sistem Pembayaran Rekening Listrik dan bila akan melakukan pembayaran harus registrasi menjadi anggota terlebih dahulu. [SRS_101]

Spesifikasi:

- 1.1 Pengunjung hanya dapat memperoleh informasi yang disediakan oleh admin yang meliputi: informasi pemadaman listrik, informasi pasang baru dan informasi lainnya. [SRS_102]
- 1.2 Pengunjung harus melakukan registrasi terlebih dahulu untuk mendapatkan *user id* dan *password* yang dapat digunakan untuk membayar tagihan listrik. [SRS_103]
- 1.3 Apabila pengunjung mengalami masalah saat melakukan registrasi, pengunjung dapat menghubungi admin perusahaan untuk didaftarkan sebagai anggota. [SRS_104]

b. Anggota

Anggota adalah pengguna yang mempunyai hak untuk membayar tagihan listrik melalui WAP, yang melakukan registrasi terlebih dahulu sebelumnya. Untuk lebih jelasnya dapat dijelaskan sebagai berikut:

Definisi:

2. Anggota harus melakukan *login* terlebih dahulu untuk dapat membayar tagihan listrik. [SRS_201]

Spesifikasi:

- 2.1 Anggota memasukkan *user id* dan *password* untuk *login*. [SRS_202]
- 2.2 Anggota dapat melakukan pembayaran, simulasi, dan berkomunikasi dengan admin jika *login* yang diberikan adalah valid. [SRS_203]

2.3 Sistem akan memberikan pesan kesalahan saat *login* yang diberikan tidak valid dan yang bersangkutan akan berstatus sebagai pengunjung. [SRS_204]

2.4 Anggota dapat memperoleh informasi transaksi, setelah melakukan pembayaran tagihan listrik yang bersangkutan. [SRS_205]

c. Admin Perusahaan

Administrator Perusahaan adalah pengguna yang mempunyai hak dalam memasukkan data yang berhubungan dengan pembayaran rekening listrik pada aplikasi ini. Diperlukan login terlebih dahulu sebelum mendapatkan fasilitas-fasilitas administrator.

Definisi:

3. Admin Perusahaan harus dapat mengolah data yaitu menambah, menghapus dan mengubah data yang dilakukan melalui *web*. [SRS_301]

Spesifikasi:

3.1 Admin Perusahaan harus memasukkan login berupa *user id* dan *password* terlebih dahulu sebelum mendapatkan fasilitas-fasilitas administrator. [SRS_302]

3.2 Admin Perusahaan harus dapat menyediakan informasi pembayaran rekening listrik yang meliputi: ID pelanggan, tarif beban listrik, dan besarnya pemakaian listrik tiap bulannya. [SRS_303]

3.3 Admin Perusahaan harus dapat melakukan *update* data. [SRS_304]

d. Admin Perangkat Lunak

Administrator Perangkat Lunak adalah pengguna yang mempunyai hak yang berhubungan dengan *user* pada aplikasi ini.

Definisi:

4. Admin Perangkat Lunak harus dapat mengolah data yang berhubungan dengan *user*, yaitu menghapus, mengubah, ataupun memblokir *user*, yang dilakukan melalui *web*. [SRS_401]

Spesifikasi:

4.1 Admin Perangkat Lunak harus memasukkan login berupa *user id* dan *password* terlebih dahulu, sebelum mendapatkan fasilitas-fasilitas administrator perangkat lunak. [SRS_402]

- 4.2 Admin Perangkat Lunak dapat mengubah atau menghapus data – data anggota apabila diperlukan. [SRS_403]
- 4.3 Admin Perangkat Lunak dapat melakukan blokir terhadap *user* anggota apabila diperlukan. [SRS_403]
- 4.4 Admin Perangkat Lunak dapat mengganti *password* untuk *user* anggota, admin perusahaan, atau manajemen. [SRS_404]

e. Manajemen

Manajemen adalah pengelola perusahaan (*manager*) PLN

Definisi:

5. Manajemen mempunyai kebijakan untuk mengontrol perkembangan perusahaan yaitu menerima hasil laporan dari pengguna aplikasi, yang dilakukan melalui *web*. [SRS_501]

Spesifikasi:

- 5.1 Manajemen harus memasukkan login berupa *user id* dan *password* terlebih dahulu sebelum mendapatkan fasilitas-fasilitas manajemen. [SRS_502]
- 5.2 Manajemen harus menerima hasil laporan dari pengguna aplikasi yaitu laporan identitas anggota, dan pembayaran tagihan rekening listrik. [SRS_503]

4.1.2.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak merupakan hal-hal yang dapat dilakukan oleh perangkat lunak terhadap *user* ataupun perangkat lunak itu sendiri dan kebutuhan perangkat lunak agar dapat diakses oleh *user*.

Definisi:

5. Pengaksesan informasi aplikasi oleh anggota dan pengunjung dilakukan dengan *browsing* ke Internet menggunakan telepon genggam. [SRS_050]

Spesifikasi:

- 5.1 Pengaksesan informasi aplikasi oleh anggota dan pengunjung dilakukan dengan *browsing* menggunakan telepon genggam yang memiliki fasilitas layanan WAP dan terdapat jaringan GPRS. [SRS_051]

Definisi:

6. Pengaksesan informasi aplikasi oleh admin perusahaan, admin perangkat lunak, dan manajemen dilakukan dengan *browsing* ke Internet melalui *web*. [SRS_060]

Spesifikasi:

- 6.1 Pengaksesan informasi aplikasi oleh admin perusahaan, admin perangkat lunak, manajemen dilakukan dengan *browsing* ke Internet melalui *web*. [SRS_061]

Definisi:

7. Sistem yang dibuat harus menggunakan *database* yang akan dirancang dan dinormalisasi sebelumnya. [SRS_070]

Spesifikasi:

- 7.1 Sistem yang dibuat harus menggunakan *database* yang akan dirancang dan dilakukan normalisasi tabel. [SRS_071]

Definisi:

8. Sistem diimplementasikan dengan bahasa pemrograman PHP dan WML, serta basis data MySQL. [SRS_080]

Spesifikasi:

- 8.1 Sistem diimplementasikan menggunakan bahasa pemrograman PHP sebagai penghubung antara sistem dengan *database*. [SRS_081]
- 8.2 Sistem diimplementasikan menggunakan bahasa pemrograman WML untuk memberikan tampilan pada layar telepon genggam. [SRS_082]
- 8.3 Sistem diimplementasikan menggunakan *database* MySQL sebagai sarana penyimpan data. [SRS_083]

Definisi:

9. Sistem operasi yang digunakan adalah sistem operasi Windows . [SRS_090]

Spesifikasi:

- 9.1 Sistem yang akan dibuat menggunakan Microsoft Windows XP SP 2. [SRS_091]

Definisi:

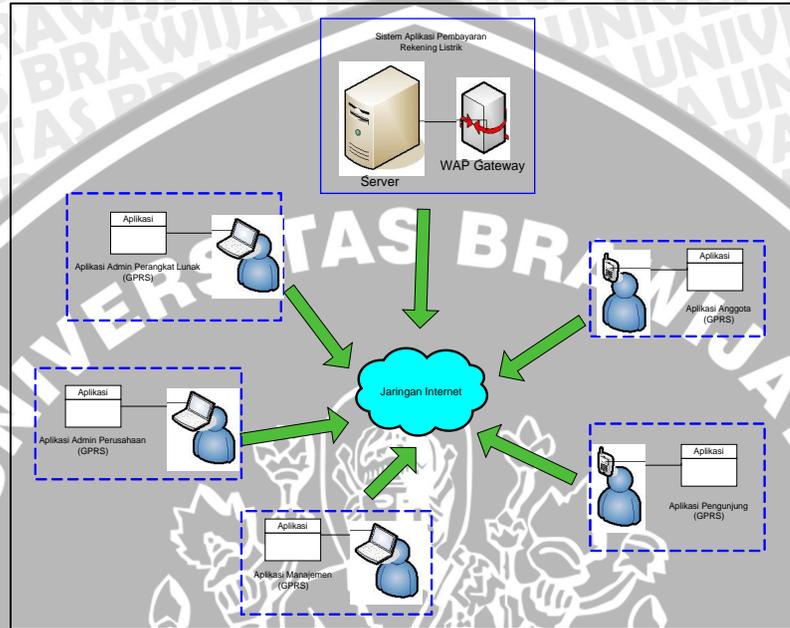
10. Dokumentasi sistem yang dibuat harus terdapat *site map*. [SRS_010]

Spesifikasi:

- 10.1 Site map merupakan penjelasan dari menu yang disediakan oleh sistem, pada *Pengunjung* meliputi: registrasi anggota untuk menjadi anggota, informasi pembayaran, informasi persyaratan keanggotaan, dan profil PLN . [SRS_011]

4.2 Diagram Blok Sistem

Diagram blok sistem menggambarkan setiap blok atau bagian dari sistem aplikasi. Aplikasi Pembayaran Tagihan Listrik ini dirancang untuk dapat dijalankan secara *online*, seperti yang ditunjukkan pada Gambar di bawah ini:



Gambar 4.4 Diagram Blok Sistem Aplikasi Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS

Sumber : Perancangan

Diagram blok sistem aplikasi ini terdiri dari lima blok yang terhubung melalui jaringan Internet. Masing-masing blok dapat dijelaskan sebagai berikut:

- **Sistem Aplikasi Pembayaran Rekening Listrik**
Merupakan bagian inti dari aplikasi ini. Pada bagian ini *server* terhubung dengan *WAP Gateway* sebagai semacam penerjemah informasi dari *server* sebelum diteruskan kepada pengguna Anggota dan Pengunjung dengan menggunakan telepon genggam.
- **Antarmuka Anggota**
Antarmuka Anggota digunakan oleh anggota (*user* Pengunjung yang telah melakukan registrasi) untuk mengakses aplikasi ini. Aplikasi ini diakses melalui telepon genggam dengan menggunakan jaringan GPRS.
- **Antarmuka Pengunjung**

Aplikasi Pengunjung adalah aplikasi yang sebenarnya sama dengan Aplikasi Anggota dalam mengakses aplikasi ini. Yang membedakan adalah Pengunjung merupakan *user* yang belum melakukan registrasi menjadi anggota.

- **Antarmuka Manajemen**

Antarmuka manajemen adalah antarmuka yang digunakan oleh pihak manajemen untuk memeriksa data pada aplikasi ini, yang ditampilkan dalam bentuk laporan. Akses ini diakses langsung ke server tanpa melalui *WAP Gateway*.

- **Antarmuka Administrator**

Antarmuka administrator adalah antarmuka yang digunakan oleh administrator untuk melakukan manajemen data pada aplikasi ini. Akses ini diakses langsung ke server tanpa melalui *WAP Gateway*.

4.3 Perancangan Perangkat Lunak

Berdasarkan data dan informasi sebelumnya, perancangan Sistem Pembayaran Rekening Listrik Berbasis WAP ini meliputi perancangan:

1. DFD
2. Basis data
3. Sistem *Site Map*
4. *Antarmuka User*

4.3.1 Data Flow Diagram

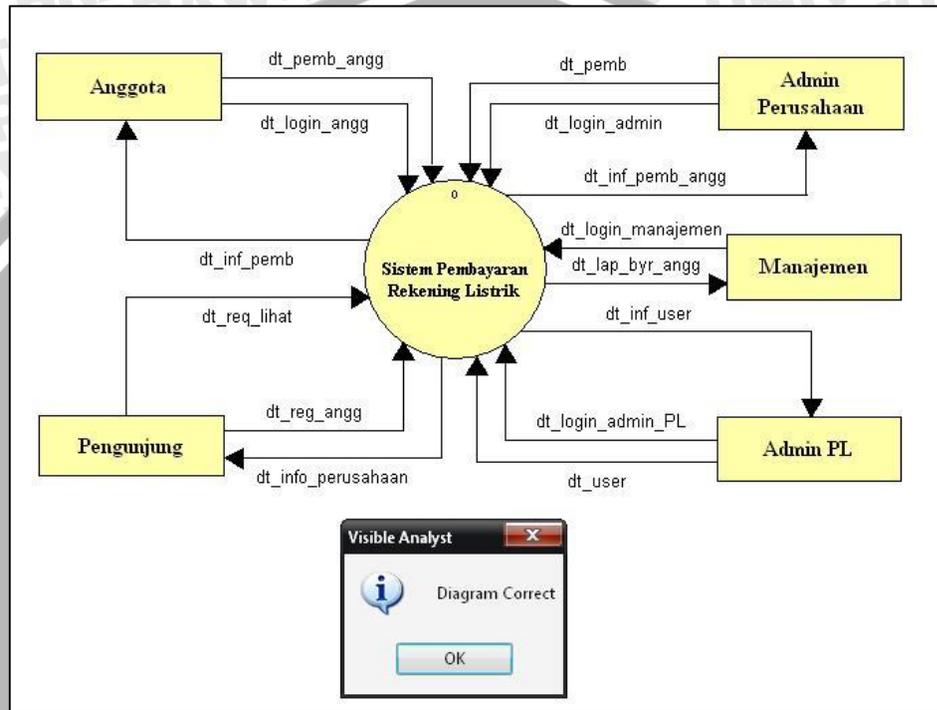
Data Flow Diagram (DFD) adalah suatu diagram yang menggambarkan model komponen sistem yang akan dibuat, dalam hal ini aplikasi sistem pembayaran rekening listrik Melalui Jaringan GPRS.

4.3.1.1 Context Diagram

Context Diagram menjelaskan hubungan sistem dengan lingkungan atau kesatuan luar. Pada sistem ini, diagram konteks yang dibuat melibatkan lima kesatuan luar, yaitu : Anggota, Pengunjung, Admin Perusahaan, Manajemen, Admin Perangkat

Lunak. *Context Diagram* direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

DFD yang pertama kali dibuat adalah DFD level 0 atau yang sering disebut *Context Diagram*. *Context Diagram* Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS ditunjukkan pada Gambar berikut:



Gambar 4.5 *Context Diagram* Aplikasi Sistem Pembayaran Rekening Listrik

Sumber : Perancangan

Berdasarkan Gambar 4.5 proses “Sistem Pembayaran Rekening Listrik” mempunyai data masukan berupa:

1. *dt_login_anggota* [IRS_0110]

dt_login_anggota oleh user (anggota) untuk dilakukan proses validasi login.

Parameter yang digunakan adalah:

- a. *user_anggota* [IRS_0111]

Berisi masukan *nama* yang diberikan oleh anggota. Tipe data dari *nama* adalah *varchar* dengan lebar *field* 50 karakter.

- b. *pass_anggota* [IRS_0112]

Berisi masukan *id_anggota* yang diberikan oleh anggota. Tipe data dari *id_anggota* adalah *varchar* dengan lebar *field* 100 karakter.

2. *dt_login_admin* [IRS_0120]

dt_login_admin diberikan oleh *user* (admin perusahaan) untuk dilakukan proses validasi login.

Parameter yang digunakan adalah:

a. *user_admin* [IRS_0121]

Berisi masukan *username* yang diberikan oleh admin. Tipe data dari *username* adalah *varchar* dengan lebar *field* 50 karakter.

b. *pass_admin* [IRS_0122]

Berisi masukan *password* yang diberikan oleh admin. Tipe data dari *password* adalah *varchar* dengan lebar *field* 100 karakter.

3. *dt_reg_anggota* [IRS_0130]

dt_reg_anggota diberikan oleh *user_pengunjung*. Pengunjung adalah semua pengguna yang membuka aplikasi ini, tetapi hanya memiliki hak untuk mendapatkan informasi yang telah disediakan pada tampilan awal aplikasi, dan perlu melakukan pendaftaran / login untuk mengakses aplikasi, yaitu membayar dan melakukan simulasi tagihan listrik.

Pengunjung dapat melakukan registrasi dengan memasukkan data -data sebagai berikut :

a. *username* [IRS_0131]

Merupakan *username* yang diisi oleh calon anggota yang bersangkutan yang selanjutnya akan digunakan oleh anggota untuk mengisi *login*.

b. *password* [IRS_0132]

Merupakan *password* yang diisi oleh calon anggota yang bersangkutan yang selanjutnya akan digunakan oleh anggota untuk mengisi *login*.

c. *nama_lengkap* [IRS_0133]

Merupakan nama lengkap calon anggota yang selanjutnya akan digunakan sebagai data anggota.

d. *alamat* [IRS_0134]

Merupakan alamat tempat tinggal calon anggota yang selanjutnya akan digunakan sebagai data anggota.

e. *no_tlp* [IRS_0135]

Merupakan nomor telepon calon anggota yang selanjutnya akan digunakan sebagai data anggota.

f. *daya yang digunakan* [IRS_0136]

Merupakan daya beban listrik calon anggota yang nantinya akan digunakan sebagai data anggota

g. *rek_listrik* [IRS_0137]

Merupakan no ID pembayaran rekening listrik calon anggota yang selanjutnya akan digunakan sebagai data anggota.

h. *rek_bank* [IRS_0138]

Merupakan nomor rekening calon anggota yang selanjutnya akan digunakan sebagai data anggota. Nomor rekening ini selanjutnya akan digunakan pada proses pembayaran.

4. *dt_login_manajemen* [IRS_0140]

dt_login_manajemen diberikan oleh *user* (manajemen) untuk dilakukan proses validasi login. Manajemen adalah pengguna yang mempunyai hak *manage* aplikasi.

Parameter yang digunakan adalah :

a. *user_manajemen* [IRS_0141]

Berisi masukan *username* yang diberikan oleh manajemen. Tipe data dari *username* adalah *varchar* dengan lebar *field* 50 karakter.

b. *pass_manajemen* [IRS_0142]

Berisi masukan *password* yang diberikan oleh manajemen. Tipe data dari *password* adalah *varchar* dengan lebar *field* 100 karakter.

5. *data_pembayaran* [IRS_0150]

data_pembayaran diberikan oleh *user* (admin). Admin memasukkan data yang berhubungan dengan pembayaran tagihan rekening listrik.

Parameter yang digunakan adalah:

a. *id_pelanggan* [IRS_0151]

Berisi masukan no ID tagihan listrik anggota dalam melakukan pembayaran tagihan listrik

b. *daya_beban_listrik* [IRS_0152]

Berisi masukan daya beban listrik yang digunakan anggota

c. *KW_meter* [IRS_0153]

Berisi masukan jumlah meteran yang sudah ditempuh dalam kwh meter yang digunakan dalam penentuan jumlah yang harus dibayar anggota

6. *data_pembayaran_anggota* [IRS_0160]

data_pembayaran_anggota diberikan oleh anggota. Anggota memasukkan data tentang pembayaran rekening listrik.

Parameter yang digunakan adalah:

a. *nama_anggota* [IRS_0161]

Berisi masukan nama yang diberikan oleh anggota. Tipe data dari *nama_anggota* adalah *varchar* dengan lebar *field* 20 karakter.

b. *Rek_listrik* [IRS_0162]

Berisi masukan nomor ID tagihan listrik yang diberikan oleh anggota. Tipe data dari *id_pelanggan* adalah *varchar* dengan lebar *field* 50 karakter.

c. *daya_beban_listrik* [IRS_0163]

Berisi masukan *daya_beban_listrik* yang diberikan oleh anggota. Tipe data dari *daya_beban_listrik* adalah *varchar* dengan lebar *field* 20 karakter.

7. *dt_req_lihat* [IRS_0170]

dt_req_lihat diberikan oleh pengunjung. Pengunjung dapat mendapatkan informasi yang telah disediakan untuk pengunjung tanpa harus *login* sebagai anggota.

8. *dt_login_admin_pl* [IRS_0180]

dt_login_admin_pl diberikan oleh *user* (admin perangkat lunak = admin PL) untuk dilakukan proses validasi login. Admin perangkat lunak adalah pengguna yang mempunyai hak mengubah data yang berhubungan dengan hak akses *user*.

Parameter yang digunakan adalah :

a. *user_admin_pl* [IRS_0181]

Berisi masukan *username* yang diberikan oleh admin perangkat lunak. Tipe data dari *username* adalah *varchar* dengan lebar *field* 50 karakter.

b. *pass_admin_pl* [IRS_0182]

Berisi masukan *password* yang diberikan oleh admin perangkat lunak. Tipe data dari *password* adalah *varchar* dengan lebar *field* 100 karakter.

9. *dt_user* [IRS_0190]

dt_user berisi tentang data anggota, admin perusahaan, dan admin perangkat lunak.

Berdasarkan Gambar 4.5 proses “Sistem Pembayaran Rekening Listrik” mempunyai data keluaran berupa:

1. *dt_info_pembayaran* [IRS_1000]

Berisi data-data transaksi pembayaran tagihan listrik anggota bersangkutan.

2. *dt_info_perusahaan* [IRS_1100]

Berisi informasi yang berkaitan dengan perusahaan yang dapat dilihat oleh pengunjung tanpa harus *login* sebagai anggota.

3. *dt_info_pembayaran_anggota* [IRS_1200]

Berisi informasi pembayaran tagihan listrik anggota meliputi : nomor ID tagihan listrik, nama, daya beban listrik, dan informasi pembayaran yang tersimpan dalam *database*.

4. *dt_laporan_pembayaran_anggota* [IRS_1300]

Berisi laporan pembayaran tagihan listrik anggota untuk pihak manajemen, yang meliputi: laporan identitas anggota, laporan data pembayaran, laporan data pembayaran.

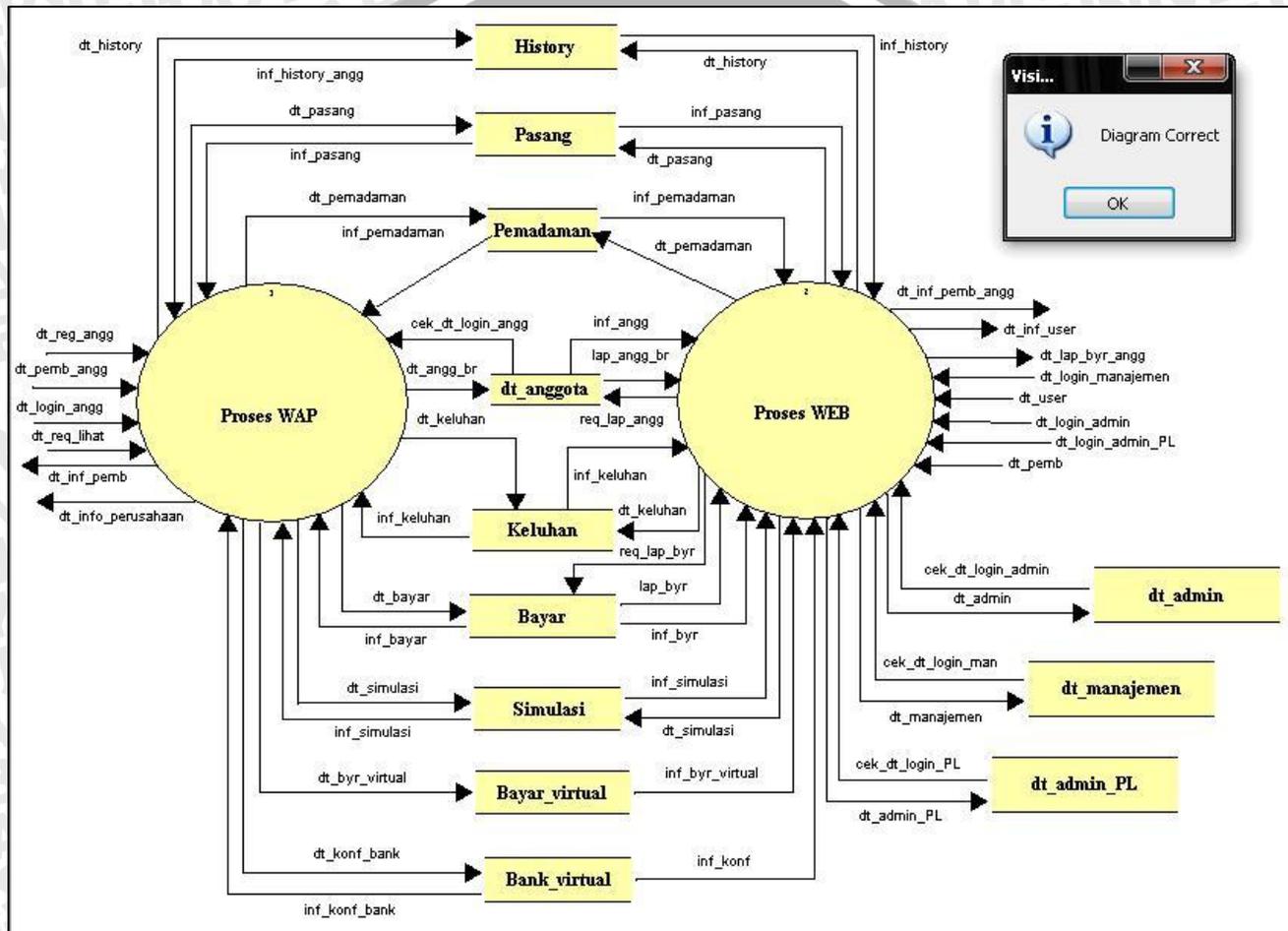
5. *dt_info_user* [IRS_1400]

Berisi informasi tentang *user* pada sistem ini untuk pihak admin perangkat lunak, yang meliputi: data anggota, *user* dan *password* anggota, admin perusahaan, dan manajemen.

Untuk masuk ke dalam sistem tersebut, *user* anggota, admin perusahaan, admin perangkat lunak, dan manajemen harus melakukan proses login terlebih dahulu, sedangkan *user* pengunjung tidak perlu melakukan proses login untuk masuk ke dalam sistem. Jika *login_anggota*, *login_admin*, *login_admin_perangkat_lunak*, *login_manajemen* sudah benar, maka *user* dapat menuju ke proses selanjutnya, apabila terdapat kesalahan, maka *user* akan tetap berada pada proses *login*.

4.3.1.2 DFD Level 1

DFD Level 1 merupakan diagram yang menguraikan proses pada DFD level 0 yang mempunyai rincian proses untuk Sistem Pembayaran Rekening Listrik. DFD level 1 ditunjukkan pada Gambar 4.6



Gambar 4.6 DFD Level 1 Aplikasi Sistem Pembayaran Rekening Listrik

Sumber : Perancangan

Gambar 4.6 merupakan DFD Level 1 dari Sistem Pembayaran Rekening Listrik dimana sistem tersebut dibagi menjadi 2 proses, yaitu : Proses WAP dan Proses *Web*. Dari diagram tersebut, maka dapat dijabarkan lebih detail proses-proses tersebut sebagai berikut:

1. Proses WAP (Proses 1)

Merupakan proses yang terjadi dengan pengaksesan WAP pada sistem pembayaran rekening listrik. *User* yang mengakses WAP adalah anggota dan pengunjung.

Proses ini mempunyai data masukan :

- a. *dt_login_anggota* [IRS_0110]
- b. *dt_reg_anggota* [IRS_0130]
- c. *dt_req_lihat* [IRS_0170]
- d. *data_pembayaran_anggota* [IRS_0160]
- e. *cek_dt_login_anggota* [IRS_0200]
cek_login_anggota berisi data login yang tersimpan dalam *data store dt_anggota* yang digunakan untuk validasi login anggota.
- f. *inf_history_anggota* [IRS_0210]
inf_history_anggota adalah informasi tentang tagihan rekening listrik yang telah dibayarkan sebelumnya.
- g. *inf_pasang* [IRS_0220]
inf_pasang_baru adalah informasi tentang biaya perhitungan pasang baru listrik.
- h. *inf_pemadaman* [IRS_0240]
inf_pemadaman adalah informasi tentang pemadaman terencana yang dilakukan oleh pihak perusahaan
- i. *inf_keluhan* [IRS_0250]
inf_keluhan adalah informasi keluhan yang berkaitan dengan perusahaan .
- j. *inf_simulasi* [IRS_0260]
inf_simulasi adalah informasi tentang perkiraan pembayaran tagihan listrik yang dilakukan oleh anggota, yang meliputi: meteran akhir bulan sebelumnya, meteran yang sudah ditempuh, dan perkiraan biaya yang akan dibayarkan .

k. *inf_byr_anggota* [IRS_0270]

inf_byr_anggota adalah informasi tentang pembayaran tagihan listrik yang dilakukan oleh anggota, yang meliputi: *id_pelanggan*, tanggal pembayaran, nama, daya beban listrik, dan biaya yang harus dibayar.

l. *inf_konf_bank* [IRS_0280]

inf_konf_bank adalah informasi tentang anggota yang dikonfirmasi dengan pihak bank, dalam hal ini berupa bank virtual, yang meliputi: nomor rekening anggota, dan jumlah saldo rekening anggota.

Proses ini mempunyai keluaran:

a. *dt_inf_pembayaran* [IRS_1000]

dt_inf_pembayaran adalah informasi pembayaran tagihan listrik anggota yang bersangkutan meliputi: *id_pelanggan*, nama, alamat, daya beban listrik, *kwh_meter* yang ditempuh.

b. *dt_info_perusahaan* [IRS_1100]

dt_info_perusahaan adalah informasi yang berkaitan dengan perusahaan.

c. *dt_anggota_baru* [IRS_0290]

dt_anggota_baru adalah data pengunjung yang telah melakukan registrasi dan menjadi anggota, yang disimpan pada *data_store dt_anggota*

d. *dt_history* [IRS_0300]

dt_history adalah data tentang tagihan listrik yang telah dibayarkan sebelumnya, yang disimpan pada *data store history*

e. *dt_pasang* [IRS_0310]

dt_pasang adalah data biaya pasang baru listrik.

f. *dt_pemadaman* [IRS_0330]

dt_pemadaman adalah data pemadaman terencana yang dilakukan oleh perusahaan

g. *dt_keluhan* [IRS_0340]

h. *dt_byr_anggota* [IRS_0350]

dt_byr_anggota adalah data tentang pembayaran tagihan listrik anggota bersangkutan

i. *dt_simulasi_anggota* [IRS_0360]

dt_simulasi_anggota adalah data tentang simulasi anggota.

j. *dt_byr_virtual* [IRS_0370]

dt_byr_virtual adalah data tentang pembayaran tagihan listrik yang belum dibayarkan oleh anggota, meliputi: *id_pelanggan*, tanggal pembayaran, nama, daya beban listrik, dan biaya yang harus dibayar oleh anggota.

k. *dt_konf_bank* [IRS_0380]

dt_konf_bank adalah data tentang pembayaran tagihan listrik beserta rekening anggota bersangkutan untuk dikonfirmasi kepada pihak bank, yang berupa *data store bank_virtual*.

Proses ini menggunakan *data store*:

a. *dt_anggota* [IRS_0001]b. *history* [IRS_0002]c. *pasang* [IRS_0003]d. *pemadaman* [IRS_0005]e. *keluhan* [IRS_0006]f. *bayar* [IRS_0007]g. *simulasi* [IRS_0008]h. *bayar_virtual* [IRS_0009]i. *bank_virtual* [IRS_0010]

2. Proses WEB (Proses 2)

Merupakan proses yang terjadi dengan pengaksesan *WEB* pada sistem pembayaran rekening listrik. *User* yang mengakses adalah admin perusahaan, admin perangkat lunak, admin manajemen.

Proses ini mempunyai masukan:

a. *dt_login_admin* [IRS_0120]b. *dt_login_admin_pl* [IRS_0180]c. *dt_login_manajemen* [IRS_0140]d. *dt_user* [IRS_0190]e. *cek_dt_login_admin* [IRS_0390]

cek_dt_login_admin berisi data login yang tersimpan dalam *data store dt_admin* yang digunakan untuk validasi login admin.

- f. *cek_dt_login_admin_pl* [IRS_0400]
cek_dt_login_admin_pl berisi data login yang tersimpan dalam *data store dt_admin_pl* yang digunakan untuk validasi login admin admin.
- g. *cek_login_admin_manajemen* [IRS_0410]
cek_login_admin_manajemen berisi data login yang tersimpan dalam *data store dt_admin_manajemen* yang digunakan untuk validasi login admin manajemen.
- h. *dt_pembayaran* [IRS_0150]
- i. *laporan_anggota_baru* [IRS_0420]
laporan_anggota_baru adalah laporan tentang identitas anggota yang berasal dari *data store dt_anggota*, yang ditujukan untuk manajemen
- j. *inf_anggota* [IRS_0430]
inf_anggota adalah informasi tentang anggota yang terdapat pada *data store dt_anggota*.
- k. *inf_history_anggota* [IRS_0440]
inf_history_anggota adalah informasi tentang history pembayaran anggota yang terdapat di *data store history*
- l. *inf_keluhan* [IRS_0450]
inf_keluhan adalah informasi keluhan yang berhubungan dengan perusahaan yang terdapat di *data store keluhan*.
- m. *inf_byr_anggota* [IRS_0460]
inf_byr_anggota adalah informasi tentang pembayaran tagihan listrik anggota yang terdapat di *data store bayar*.
- n. *lap_bayar* [IRS_0470]
lap_bayar adalah laporan tentang pembayaran tagihan listrik anggota yang berasal dari *data store bayar*, yang ditujukan untuk manajemen
- o. *inf_simulasi_anggota* [IRS_0480]
inf_simulasi_anggota adalah informasi tentang simulasi yang dilakukan oleh anggota yang terdapat di *data store simulasi*
- p. *inf_byr_virtual* [IRS_0490]
inf_byr_virtual adalah informasi tentang pembayaran tagihan listrik yang dilakukan oleh anggota, yang meliputi: *id_pelanggan*, tanggal pembayaran, nama, daya beban listrik, dan biaya yang harus dibayar.

q. *inf_pemadaman* [IRS_0500]

inf_pemadaman adalah informasi tentang pemadaman berkala yang dilakukan oleh perusahaan yang terdapat di *data store pemadaman*.

r. *inf_konf_bank* [IRS_0510]

inf_konf_bank adalah informasi konfirmasi dengan pihak bank, dalam hal ini berupa bank virtual, yang terdapat di *data store bank_virtual*.

Proses ini mempunyai keluaran:

a. *dt_admin* [IRS_0520]

dt_admin adalah data *login admin* berupa *username* dan *password* yang terdapat di *data store dt_admin*

b. *dt_admin_pl* [IRS_0530]

dt_admin_pl adalah data *login admin_pl* berupa *username* dan *password* yang terdapat di *data store dt_admin_pl*

c. *dt_admin_manajemen* [IRS_0540]

dt_admin_manajemen adalah data *login admin manajemen* berupa *username* dan *password* yang terdapat di *data store dt_admin_manajemen*

d. *dt_inf_pembayaran_anggota* [IRS_0550]

inf_pembayaran_anggota adalah informasi pembayaran tagihan listrik anggota meliputi : nomor *id* tagihan listrik, nama, daya beban listrik, dan informasi pembayaran anggota yang bersangkutan.

e. *dt_inf_user* [IRS_1400]

f. *req_lap_anggota* [IRS_0560]

req_lap_anggota adalah permintaan dari *user* manajemen untuk menampilkan laporan anggota

g. *req_lap_byr* [IRS_0570]

req_lap_byr adalah permintaan dari *user* manajemen untuk menampilkan laporan pembayaran

h. *dt_keluhan* [IRS_0580]

dt_lain adalah data-data yang berkaitan dengan perusahaan yang dimasukkan oleh admin ke *data store lain*

i. *dt_history* [IRS_0590]

dt_history adalah data *history* pembayaran yang dimasukkan oleh admin ke *data store history*

j. *dt_simulasi_anggota* [IRS_0600]

dt_simulasi_anggota adalah data tentang simulasi anggota yang dimasukkan oleh admin ke *data store simulasi*.

k. *dt_pasang* [IRS_0610]

dt_pasang adalah data tentang biaya pemasangan listrik baru yang dimasukkan oleh admin ke *data store pasang*

l. *dt_pemadaman* [IRS_0630]

dt_pemadaman adalah data tentang pemadaman berkala yang dimasukkan oleh admin ke *data store pemadaman*.

Proses ini menggunakan *data store*:

a. *dt_anggota* [IRS_0001]

b. *dt_admin* [IRS_0011]

c. *dt_admin_pl* [IRS_0012]

d. *dt_manajemen* [IRS_0013]

e. *history* [IRS_0002]

f. *pasang* [IRS_0003]

g. *pemadaman* [IRS_0005]

h. *keluhan* [IRS_0006]

i. *bayar* [IRS_0007]

j. *simulasi* [IRS_0008]

k. *bayar_virtual* [IRS_0009]

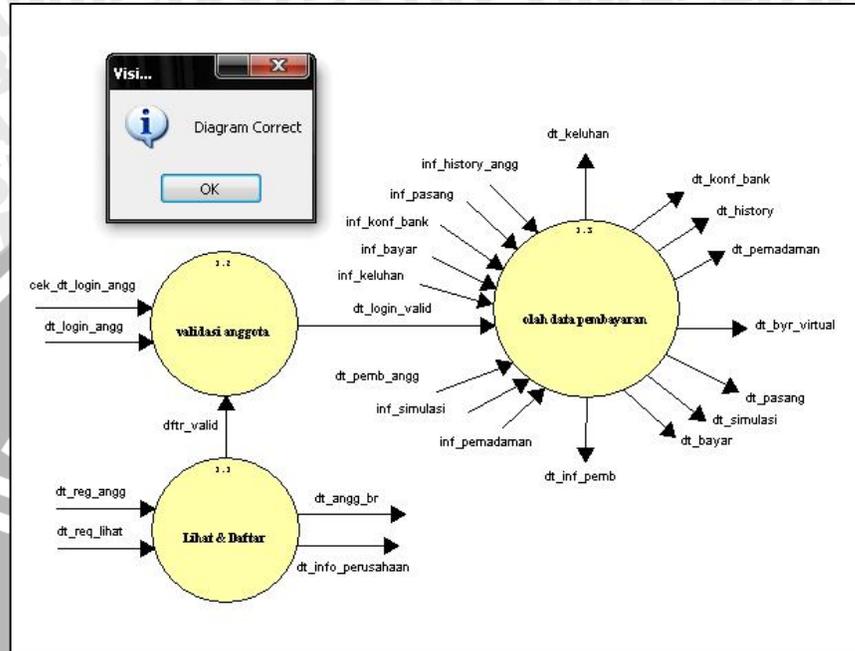
l. *bank_virtual* [IRS_0010]

4.3.1.3 DFD Level 2

DFD Level 2 merupakan diagram yang menguraikan proses pada DFD level 1 yang mempunyai rincian proses untuk Sistem Pembayaran Rekening Listrik. DFD level 2 digambarkan pada Gambar 4.7

a. DFD level 2 Proses WAP

DFD level 2 Proses WAP merupakan penjabaran dari Proses WAP (proses 1) pada DFD level 1



Gambar 4.7 DFD Level 2 Proses WAP Pembayaran Rekening Listrik

Sumber : Perancangan

Gambar 4.7 merupakan DFD level 2 proses pembayaran tagihan listrik dimana proses tersebut terjadi pada Proses WAP dari sistem pembayaran tagihan listrik ini. Proses tersebut dibagi dalam beberapa proses, yaitu: proses daftar, proses validasi anggota, dan proses olah data pembayaran.

1. Proses Daftar (1.1)

Proses ini digunakan untuk melakukan pendaftaran bagi pengunjung yang ingin melakukan pembayaran tagihan listrik.

Proses ini mempunyai data masukan:

- a. *dt_req_anggota* [IRS_0130]
- b. *dt_req_lihat* [IRS_0170]

proses ini mempunyai data keluaran:

- a. *dt_anggota_br* [IRS_0640]
- b. *dtrr_valid* [IRS_0650]

c. *dt_info_perusahaan* [IRS_1100]
proses ini menggunakan *data store*:

dt_anggota [IRS_0001]

2. Proses Validasi Anggota (1.2)

Proses ini dilakukan pertama kali untuk melakukan pembayaran tagihan listrik.

Proses ini mengambil data-data yang diperlukan dari *data store dt_anggota*.

Proses ini mempunyai data masukan:

a. *dt_login_anggota* [IRS_0110]

b. *cek_dt_login_anggota* [IRS_0200]

cek_dt_login_anggota berisi data *login* anggota yang berada di *data store dt_anggota* yang digunakan untuk validasi *login* anggota.

c. *dftr_valid* [IRS_0650]

dftr_valid digunakan pengunjung yang ingin melakukan pembayaran tagihan listrik telah melakukan registrasi dengan benar dan menjadi anggota.

Proses ini memiliki data keluaran:

dt_login_valid [IRS_0660]

Proses ini menggunakan *data store*:

dt_anggota [IRS_0001]

3. Proses Olah Data Pembayaran (1.3)

Proses ini merupakan proses pembayaran, simulasi, dan melihat history pembayaran bagi anggota dalam tampilan WAP.

Proses ini mempunyai data masukan:

a. *dt_login_valid* [IRS_0660]

b. *dt_pemb_anggota* [IRS_0160]

c. *inf_bayar* [IRS_0270]

d. *inf_simulasi* [IRS_0260]

e. *inf_pasang* [IRS_0220]

f. *inf_pemadaman* [IRS_0500]

g. *inf_keluhan* [IRS_0250]

h. *inf_konf_bank* [IRS_0510]

i. *inf_history* [IRS_0210]

Proses ini mempunyai data keluaran:

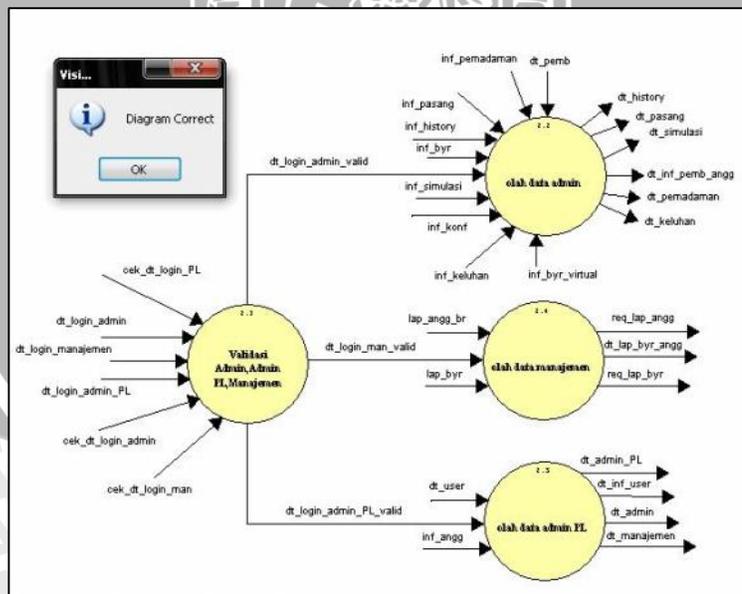
- a. *dt_history* [IRS_0300]
- b. *dt_byr_angg* [IRS_0350]
- c. *dt_simulasi_angg* [IRS_0360]
- d. *dt_pasang* [IRS_0310]
- e. *dt_pemadaman* [IRS_0330]
- f. *dt_keluhan* [IRS_0340]
- g. *dt_byr_virtual* [IRS_0370]
- h. *dt_konf_bank* [IRS_0380]
- i. *dt_inf_pemb* [IRS_1000]

Proses ini menggunakan *data store*:

- a. *history* [IRS_0002]
- b. *bayar* [IRS_0007]
- c. *simulasi* [IRS_0008]
- d. *bayar_virtual* [IRS_0009]
- e. *bank_virtual* [IRS_0010]

b. DFD level 2 Proses WEB

DFD level 2 Proses WEB merupakan penjabaran dari Proses WEB (proses 2) pada DFD level 1



Gambar 4.8 DFD Level 2 Proses WEB Pembayaran Rekening Listrik

Sumber : Perancangan

Gambar 4.8 merupakan DFD level 2 pembayaran tagihan listrik dimana proses tersebut terjadi pada proses WEB dari sistem pembayaran tagihan listrik ini. Proses ini terbagi menjadi beberapa bagian yang lebih spesifik, yaitu: validasi admin, admin perangkat lunak, manajemen, olah data admin, manajemen, dan admin perangkat lunak.

1. Proses Validasi Admin, Admin PL, dan Manajemen (2.1)

Proses ini dilalui pertama kali oleh Admin, Admin PL, dan manajemen dalam sistem WEB pembayaran tagihan listrik ini. Proses ini mengambil data -data yang dibutuhkan dari *data store* *dt_admin*, *dt_admin_pl*, *dt_manajemen*.

Proses ini mempunyai data masukan:

- a. *dt_login_admin* [IRS_0120]
- b. *dt_login_admin_pl* [IRS_0180]
- c. *dt_login_manajemen* [IRS_0140]
- d. *cek_dt_login_admin* [IRS_0670]
- e. *cek_dt_login_admin_pl* [IRS_0680]
- f. *cek_dt_login_manajemen* [IRS_0690]

Proses ini mempunyai data keluaran:

- a. *login_admin_valid* [IRS_0700]
login_admin_valid adalah login admin yang telah tervalidasi setelah menerima data dari *cek_login_admin*.
- b. *login_admin_pl_valid* [IRS_0710]
login_admin_pl_valid adalah login admin yang telah tervalidasi setelah menerima data dari *cek_login_admin_pl*.
- c. *login_manajemen_valid* [IRS_0720]
login_manajemen_valid adalah login admin yang telah tervalidasi setelah menerima data dari *cek_login_manajemen*.

Proses ini menggunakan *data store*:

- a. *dt_admin* [IRS_0011]
- b. *dt_admin_pl* [IRS_0012]
- c. *dt_manajemen* [IRS_0013]

2. Olah Data Admin (2.2)

Proses ini merupakan proses bagi *user* admin untuk memasukkan data

Proses ini mempunyai data masukan:

- a. *login_admin_valid* [IRS_0700]
- b. *inf_history* [IRS_0210]
- c. *inf_byr* [IRS_0270]
- d. *inf_simulasi* [IRS_0260]
- e. *inf_pemadaman* [IRS_0500]
- f. *inf_keluhan* [IRS_0250]
- g. *inf_pasang* [IRS_0220]
- j. *inf_konf* [IRS_0510]
- h. *dt_pembayaran* [IRS_0150]
- i. *inf_byr_virtual* [IRS_0490]

Proses ini mempunyai data keluaran:

- a. *dt_history* [IRS_0590]
- b. *dt_simulasi* [IRS_0600]
- c. *dt_pasang* [IRS_0610]
- d. *dt_pemadaman* [IRS_0630]
- e. *dt_keluhan* [IRS_0580]
- f. *dt_inf_pemb_anggota* [IRS_0550]

Proses ini menggunakan *data store*:

- a. *history* [IRS_0002]
 - b. *simulasi* [IRS_0008]
 - c. *bayar* [IRS_0007]
 - d. *bayar_virtual* [IRS_0009]
 - e. *bank_virtual* [IRS_0010]
3. Olah Data Manajemen (2.3)

Proses ini merupakan proses bagi *user* manajemen untuk melihat laporan tentang pembayaran tagihan listrik ini. Laporan tersebut meliputi: data anggota dari *data store dt_anggota*, data pembayaran tagihan listrik dari *data store bayar*.

Proses ini mempunyai data masukan:

- a. *login_manajemen_valid* [IRS_0720]
- b. *lap_anggota_baru* [IRS_0420]
- c. *lap_byr* [IRS_0470]

Proses ini mempunyai data keluaran:

- a. *req_lap_angg* [IRS_0560]
- b. *dt_lap_byr_angg* [IRS_0470]
- c. *req_lap_byr* [IRS_0570]

Proses ini menggunakan *data store*:

- a. *dt_anggota* [IRS_0001]
 - b. *bayar* [IRS_0007]
4. Olah Data Admin PL (2.4)

Proses ini merupakan proses pengolahan data oleh *user* admin perangkat lunak data masukan dan keluaran dari proses ini berasal dari *data store dt_anggota*, *admin*, dan *manajemen*.

Proses ini mempunyai data masukan:

- a. *dt_login_admin_pl_valid* [IRS_0710]
- b. *dt_user* [IRS_0190]
- c. *inf_anggota* [IRS_0430]

Proses ini mempunyai data keluaran:

- a. *dt_admin_pl* [IRS_0530]
- b. *dt_inf_user* [IRS_1400]
- c. *dt_admin* [IRS_0520]
- d. *dt_manajemen* [IRS_0540]

Proses ini menggunakan *data store*:

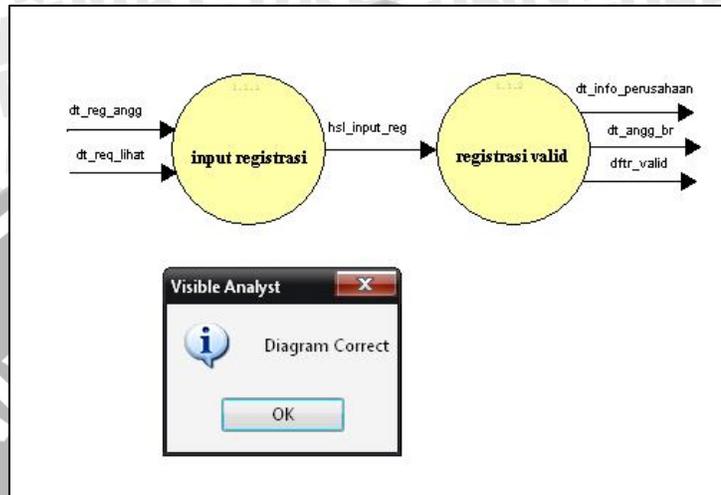
- a. *dt_anggota* [IRS_0001]
- b. *dt_admin* [IRS_0011]
- d. *dt_manajemen* [IRS_0013]

4.3.1.4 DFD Level 3

DFD Level 3 merupakan diagram yang menguraikan proses WAP pada DFD level 2 yang mempunyai rincian proses. Penjabaran lebih rinci proses tersebut akan dijelaskan pada subbab berikut ini.

a. DFD level 3 Proses Daftar

Gambar 4.9 merupakan DFD level 3 proses daftar dari DFD level 2 proses WAP. Proses ini mempunyai rincian yang lebih spesifik lagi yaitu: input registrasi dan registrasi valid.



Gambar 4.9 DFD Level 3 Proses Daftar

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut:

1. Input Registrasi (1.1.1)

Merupakan proses bagi pengunjung yang ingin melakukan pembayaran tagihan listrik dengan mendaftarkan diri sebagai anggota.

Proses ini mempunyai masukan:

- a. *dt_reg_anggota* [IRS_0130]
- b. *dt_req_lihat* [IRS_0170]

Proses ini mempunyai keluaran:

hsl_input_reg [IRS_0730]

2. Registrasi Valid (1.1.2)

Proses ini merupakan validasi *hsl_input_reg* yang merupakan hasil keluaran dari proses sebelumnya. Apabila *hsl_input_reg* valid maka *user* dapat melanjutkan proses berikutnya, bila tidak maka *user* akan diminta memasukkan *reg_anggota* sekali lagi.

Proses ini mempunyai masukan:

hsl_input_reg [IRS_0730]

Proses ini mempunyai keluaran:

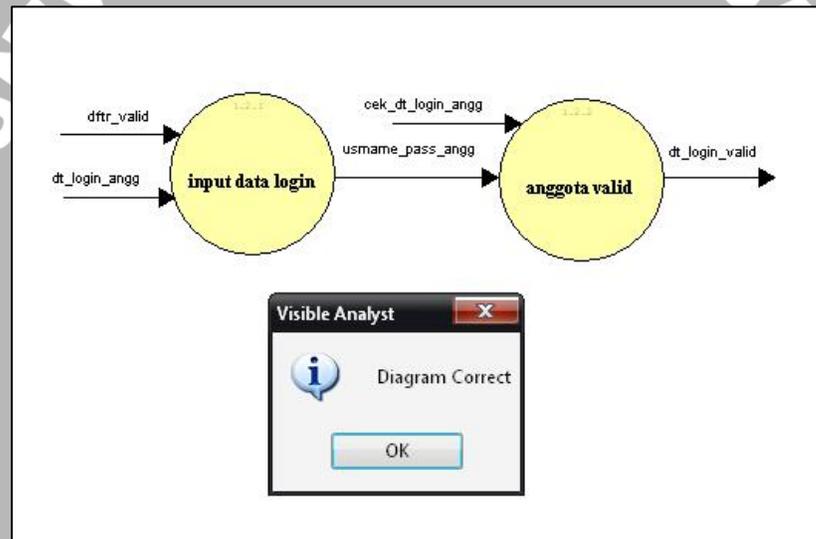
- dt_anggota_br* [IRS_0290]
- dftr_valid* [IRS_0650]
- dt_inf_perusahaan* [IRS_1100]

Proses ini menggunakan *data store*:

dt_anggota [IRS_0001]

b. DFD level 3 Proses Validasi Anggota

DFD level 3 Proses Validasi Anggota merupakan penjabaran dari DFD level 2 proses WAP pembayaran tagihan listrik pada proses validasi anggota (1.2). Penjabaran proses ini dapat dilihat pada gambar 4.10



Gambar 4.10 DFD Level 3 Proses Validasi Anggota

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Input Data Login (1.2.1)

Merupakan proses bagi *user* untuk memasukkan data *login* yang berupa *username* dan *password*.

Proses ini mempunyai data masukan:

- dftr_valid* [IRS_0650]
- dt_login_anggota* [IRS_0110]

Proses ini mempunyai data keluaran:

usname_pass_angg [IRS_0740]

2. Anggota Valid (1.2.2)

Proses ini merupakan pengolahan dari keluaran proses sebelumnya yaitu *username_pass_angg* yang kemudian dicocokkan dengan data yang ada di *data store dt_anggota*. Apabila data valid maka *user* dapat melanjutkan ke proses berikutnya, apabila tidak valid maka akan didapatkan pesan kesalahan dan *user* diminta untuk melakukan proses *login* kembali.

Proses ini mempunyai data masukan:

- a. *username_pass_angg* [IRS_0740]
- b. *cek_dt_login_angg* [IRS_0200]

Proses ini mempunyai data keluaran:

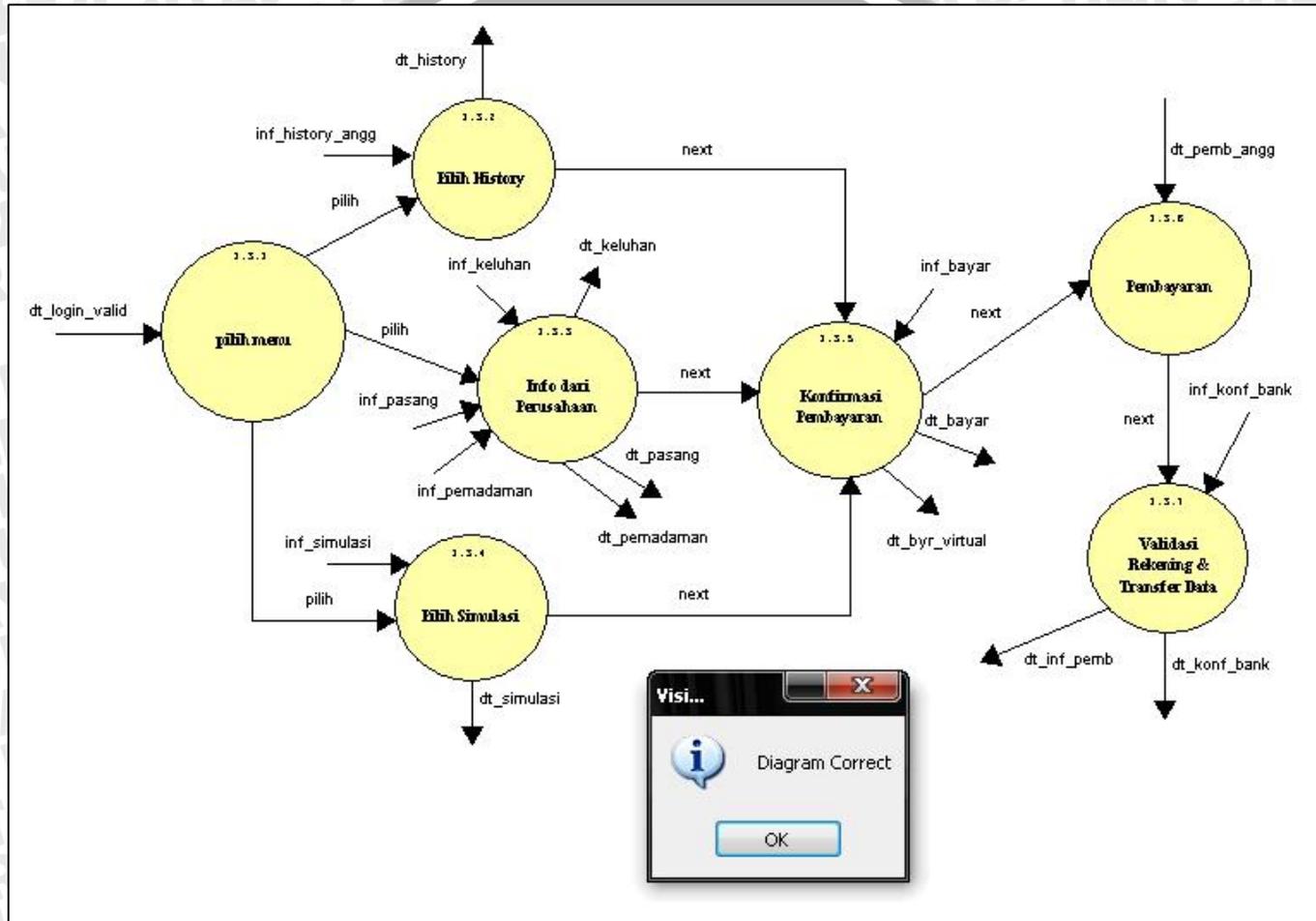
dt_login_valid [IRS_0750]

Proses menggunakan *data store*:

dt_anggota [IRS_0001]

c. DFD level 3 Olah Data Pembayaran

DFD level 3 Olah Data Pembayaran merupakan penjabaran dari DFD level 2 Proses WAP pembayaran tagihan listrik pada proses olah data pembayaran (1.3). Penjabaran dari proses ini ditunjukkan pada gambar 4.11



Gambar 4.11 DFD Level 3 Proses Olah Data Pembayaran

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Pilih Menu (1.3.1)

Merupakan proses pilihan menu yang diinginkan oleh *user* admin untuk menuju ke proses berikutnya.

Proses ini mempunyai data masukan:

dt_login_valid [IRS_0660]

Proses ini mempunyai data keluaran:

pilih [IRS_0760]

2. Pilih History (1.3.2)

Merupakan proses untuk melihat pembayaran yang sudah dilakukan oleh anggota.

Proses ini mempunyai data masukan:

a. *pilih* [IRS_0770]

b. *inf_history_angg* [IRS_0210]

Proses ini mempunyai data keluaran:

a. *dt_history* [IRS_0300]

b. *next* [IRS_0780]

Proses ini menggunakan *data store*:

f. *history* [IRS_0002]

3. Pilih Info Dari Perusahaan (1.3.3)

Merupakan proses untuk melihat menu-menu yang disediakan oleh perusahaan.

Proses ini mempunyai data masukan:

a. *pilih* [IRS_0770]

b. *inf_pasang* [IRS_0220]

c. *inf_pemadaman* [IRS_0500]

d. *inf_keluhan* [IRS_0250]

Proses ini mempunyai data keluaran:

a. *next* [IRS_0780]

b. *dt_pasang* [IRS_0310]

c. *dt_pemadaman* [IRS_0330]

d. *dt_keluhan* [IRS_0340]

Proses ini menggunakan *data store*:

- a. *pasang* [IRS_0003]
- b. *pemadaman* [IRS_0005]
- c. *keluhan* [IRS_0006]

4. Pilih Simulasi (1.3.4)

Merupakan proses untuk melakukan simulasi tagihan listrik dengan memasukkan angka yang tertunjuk pada kwh meter.

Proses ini mempunyai data masukan:

- a. *pilih* [IRS_0760]
- b. *inf_simulasi* [IRS_0260]

proses ini mempunyai data keluaran:

- a. *next* [IRS_0780]
- b. *dt_simulasi* [IRS_0360]

Proses ini menggunakan *data store*:

simulasi [IRS_0008]

5. Konfirmasi Pembayaran (1.3.5)

Merupakan proses konfirmasi pembayaran tagihan listrik oleh anggota.

Proses ini mempunyai data masukan:

- a. *next* [IRS_0780]
- b. *inf_bayar* [IRS_0270]

Proses ini mempunyai data keluaran:

- a. *dt_byr_virtual* [IRS_0370]
- b. *dt_byr* [IRS_0350]
- c. *next* [IRS_0780]

Proses ini menggunakan *data store*:

- a. *bayar* [IRS_0007]
- b. *bayar_virtual* [IRS_0009]

6. Pembayaran (1.3.6)

Proses ini merupakan pembayaran tagihan listrik oleh anggota.

Proses ini mempunyai data masukan:

- a. *next* [IRS_0780]
- b. *dt_pemb_angg* [IRS_0150]

Proses ini mempunyai data keluaran:

next [IRS_0780]

Proses ini menggunakan *data store*:

bayar [IRS_0007]

7. Validasi Rekening dan Transfer Data (1.3.7)

Merupakan proses validasi rekening dan transfer data ke bank yang berupa *data store bank_virtual*.

Proses ini mempunyai data masukan:

a. *next* [IRS_0780]

b. *inf_konf_bank* [IRS_0280]

Proses ini mempunyai data keluaran:

a. *dt_konf_bank* [IRS_0380]

b. *dt_inf_pemb* [IRS_1000]

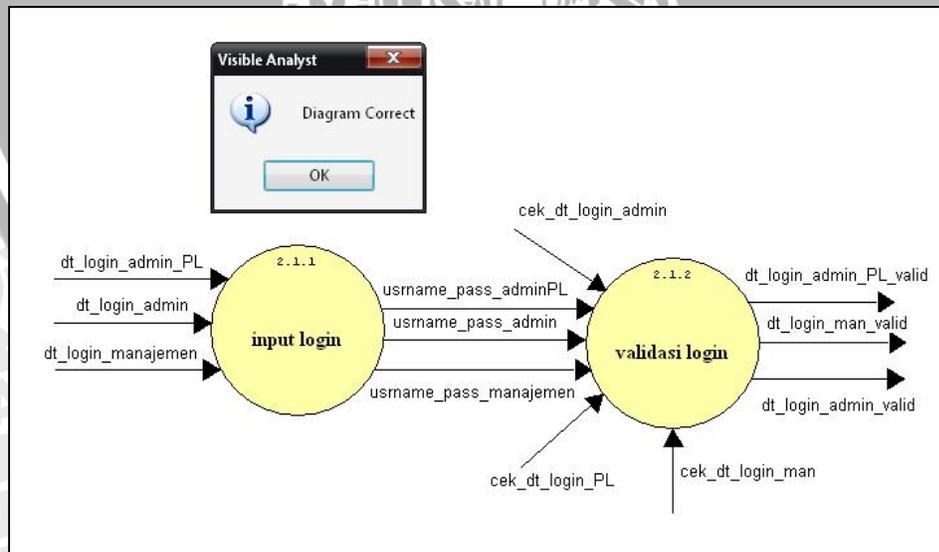
Proses ini menggunakan *data store*

bank_virtual [IRS_0010]

d. DFD level 3 Validasi Login Admin, Admin Perangkat Lunak, manajemen

DFD level 3 validasi login admin, admin perangkat lunak, dan manajemen merupakan penjabaran dari DFD level 2 Proses WEB pembayaran tagihan listrik pada proses validasi admin, admin perangkat lunak, manajemen (2.1).

Penjabaran dari proses ini ditunjukkan pada gambar 4.12



Gambar 4.12 DFD Level 3 Proses WEB validasi login admin, admin PL, manajemen

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Input Login (2.1.1)

Merupakan proses memasukkan data login dari admin, admin perangkat lunak, dan manajemen berupa *username* dan *password*.

Proses ini mempunyai data masukan:

- a. *dt_login_admin* [IRS_0120]
- b. *dt_login_admin_pl* [IRS_0180]
- c. *dt_login_manajemen* [IRS_0670]

Proses ini mempunyai data keluaran:

- a. *usrname_pass_admin* [IRS_0790]
- b. *usrname_pass_admin_pl* [IRS_0800]
- c. *usrname_pass_manajemen* [IRS_0810]

2. Validasi Login (2.1.2)

Proses ini merupakan validasi data keluaran dari proses sebelumnya yang dicocokkan dengan *data store dt_admin, dt_admin_pl, dt_manajemen*. Apabila data valid maka *user* dapat melanjutkan ke proses berikutnya, apabila data tidak valid maka akan muncul pesan kesalahan dan *user* diberikan pilihan untuk melakukan proses *input login* kembali.

Proses ini mempunyai data masukan:

- a. *usrname_pass_admin* [IRS_0790]
- b. *usrname_pass_admin_pl* [IRS_0800]
- c. *usrname_pass_manajemen* [IRS_0810]
- d. *cek_dt_login_admin* [IRS_0390]
- e. *cek_dt_login_admin_pl* [IRS_0400]
- f. *cek_dt_login_manajemen* [IRS_0410]

Proses ini mempunyai data keluaran:

- a. *dt_login_admin_valid* [IRS_0820]
- b. *dt_login_admin_pl_valid* [IRS_0830]
- c. *dt_login_admin_manajemen_valid* [IRS_0840]

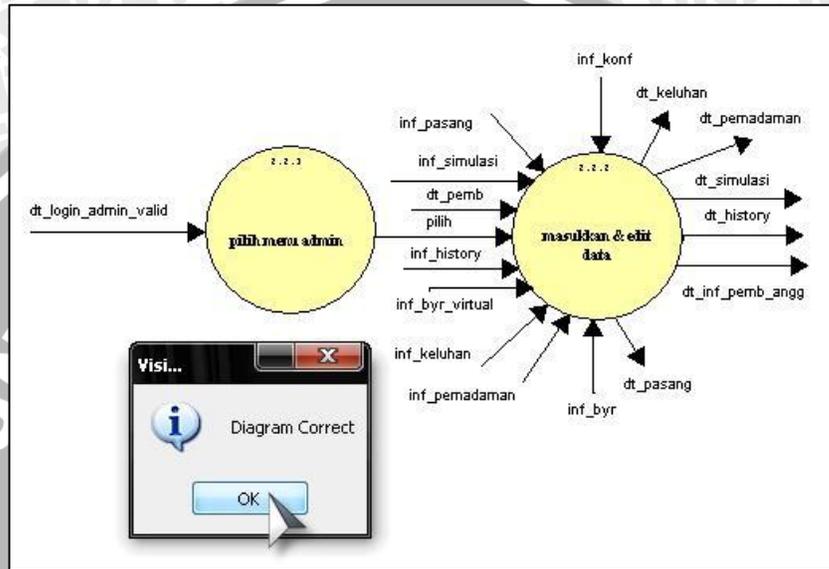
Proses ini menggunakan *data store*:

- a. *dt_admin* [IRS_0011]
- b. *dt_admin_pl* [IRS_0012]

c. *dt_manajemen* [IRS_0013]

e. **DFD Level 3 Olah Data Admin**

DFD level 3 Olah Data Admin merupakan penjabaran dari DFD level 2 proses WEB pembayaran tagihan listrik pada proses olah data admin (2.2). penjabaran dari proses ini ditunjukkan pada gambar 4.13



Gambar 4.13 DFD Level 3 Proses WEB Olah Data Admin

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Pilihan Menu Admin (2.2.1)

Merupakan proses menampilkan menu bagi *user* admin.

Proses ini mempunyai data masukan:

dt_login_admin_valid [IRS_0390]

Proses ini mempunyai data keluaran:

pilih [IRS_0760]

2. Masukkan dan Edit Data (2.2.2)

Merupakan proses memasukkan atau mengedit data bagi *user* admin ke *data store*.

Proses ini mempunyai data masukan:

a. *pilih* [IRS_0760]

b. *inf_simulasi* [IRS_0260]

c. *dt_pemb* [IRS_0150]

- d. *inf_history* [IRS_0210]
- e. *inf_konf* [IRS_0510]
- f. *inf_byr* [IRS_0270]
- g. *inf_byr_virtual* [IRS_0490]
- h. *inf_pasang* [IRS_0220]
- i. *inf_pemadaman* [IRS_0500]
- j. *inf_keluhan* [IRS_0250]

Proses ini mempunyai data keluaran:

- a. *dt_simulasi* [IRS_0600]
- b. *dt_history* [IRS_0590]
- c. *dt_inf_pemb_angg* [IRS_0550]
- d. *dt_pasang* [IRS_0610]
- e. *dt_pemadaman* [IRS_0630]
- f. *dt_keluhan* [IRS_0580]

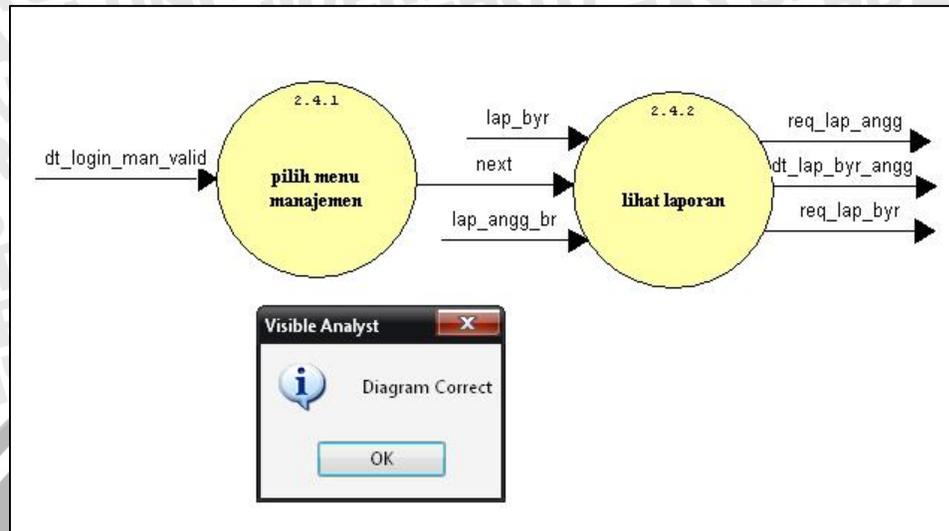
Proses ini menggunakan *data store*:

- a. *history* [IRS_0002]
- b. *simulasi* [IRS_0008]
- c. *bayar* [IRS_0007]
- d. *pasang* [IRS_0003]
- e. *pemadaman* [IRS_0005]
- f. *keluhan* [IRS_0006]
- g. *bayar_virtual* [IRS_0009]
- h. *bank_virtual* [IRS_0010]

f. **DFD Level 3 Olah Data Manajemen**

DFD level 3 olah data manajemen merupakan penjabaran dari DFD level 2 proses WEB pembayaran tagihan listrik pada proses olah data manajemen.

Penjabaran dari proses ini ditunjukkan pada gambar 4.14.



Gambar 4.14 DFD Level 3 Proses WEB Olah Data Manajemen

Sumber : Perancangan

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Pilih Menu Manajemen (2.4.1)

Merupakan proses untuk menampilkan menu bagi *user* manajemen untuk melihat laporan dari pembayaran tagihan listrik anggota.

Proses ini mempunyai data masukan:

dt_login_man_valid [IRS_0840]

Proses ini mempunyai data keluaran:

next [IRS_0780]

2. Lihat Laporan (2.4.2)

Merupakan proses melihat laporan bagi *user* manajemen.

Proses ini mempunyai data masukan:

a. *next* [IRS_0780]

b. *lap_byr* [IRS_0470]

c. *lap_angg_br* [IRS_0420]

Proses ini mempunyai data keluaran:

a. *req_lap_angg* [IRS_0560]

b. *dt_lap_byr_angg* [IRS_0470]

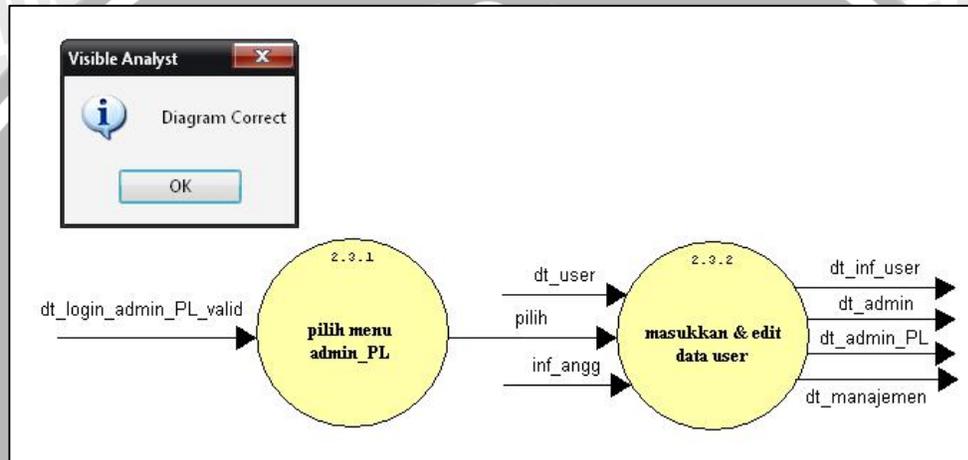
c. *req_lap_byr* [IRS_0570]

Proses ini menggunakan *data store*:

- dt_anggota* [IRS_0001]
- bayar* [IRS_0007]

g. DFD Level 3 Olah Data Admin Perangkat Lunak

DFD level 3 olah data admin perangkat lunak merupakan penjabaran dari proses WEB DFD level 2 pembayaran tagihan listrik pada proses oleh data admin perangkat lunak. Penjabaran proses ini ditunjukkan pada gambar 4.15



Gambar 4.15 DFD Level 3 Proses WEB Olah Data Admin Perangkat Lunak

Sumber : *Perancangan*

Penjabaran dari proses tersebut dapat dijelaskan sebagai berikut

1. Pilih Menu Admin PL (2.3.1)

Merupakan proses menampilkan menu bagi *user* admin perangkat lunak.

Proses ini mempunyai data masukan:

dt_login_admin_pl_valid [IRS_0710]

Proses ini mempunyai data keluaran:

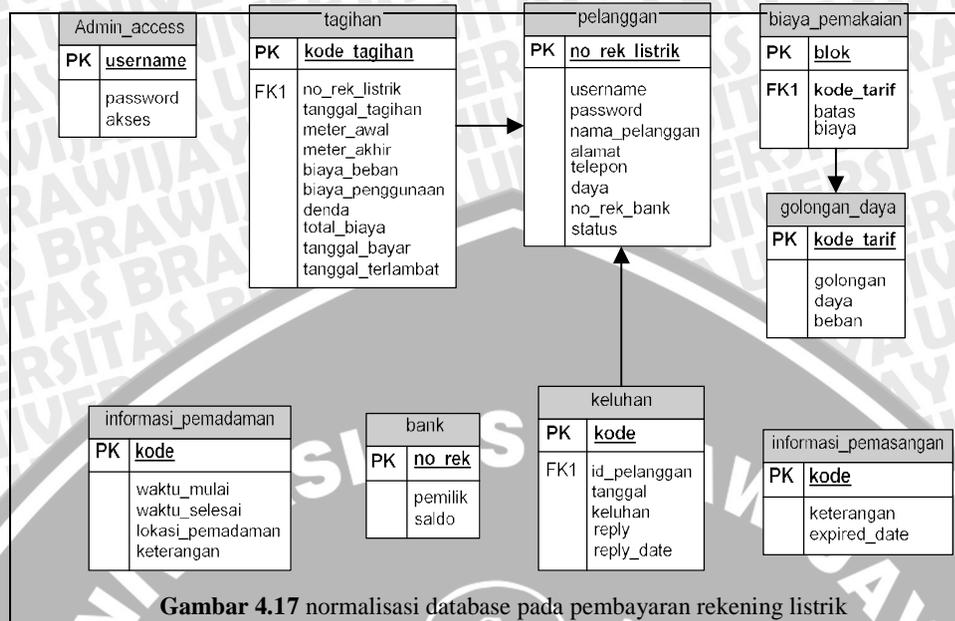
pilih [IRS_0760]

2. Masukkan dan Edit Data User (2.3.2)

Merupakan proses untuk memasukkan dan mengedit data *user* ke *data store*.

Proses ini mempunyai data masukan:

- pilih* [IRS_0760]
- dt_user* [IRS_0190]
- inf_anggota* [IRS_0430]



Gambar 4.17 normalisasi database pada pembayaran rekening listrik

Sumber : Perancangan

4.3.2.1 Normalisasi Data

Normalisasi adalah sebuah cara memecah atau memisahkan sebuah tabel yang mengalami redundansi data ke dalam dua entitas (tabel) atau lebih [BUN -04:84]. Proses normalisasi data membuat basisdata lebih mudah untuk diperlihara.

4.3.2.1.1 Normalisasi Pertama

Normalisasi pertama dilakukan setelah mengetahui kelemahan yang ada pada tabel awal. Secara umum tujuan dilakukannya normalisasi pertama adalah sebagai berikut:

- ❖ mendefinisikan atribut yang menjadi kunci bagi tabel tersebut
- ❖ tidak adanya perulangan kolom dalam suatu tabe l
- ❖ semua atribut selain kunci, bergantung pada kunci

4.3.2.1.2 Normalisasi Kedua

Normalisasi kedua dilakukan apabila setelah normasilasi tabel pertama dilakukan masih terdapat redundansi data. Secara umum tujuan dilakukannya normalisasi kedua adalah sebagai berikut:

- ❖ memastikan bentuk normal pertama telah terpenuhi
- ❖ memastikan tidak ada ketergantungan parsial

4.3.2.1.3 Normalisasi Ketiga

Normalisasi ketiga dilakukan apabila masih terdapat penumpukan data. Secara umum tujuan dilakukannya normalisasi ketiga adalah sebagai berikut:

- ❖ memastikan bentuk normal kedua telah terpenuhi
- ❖ memastikan tidak ada ketergantungan transitif

4.3.2.2 Data Object Description

Data object description menjelaskan secara rinci mengenai atribut-atribut yang dimiliki oleh masing-masing tabel yang ada pada basis data sesuai dengan *entity relationship diagram*. *Data object description* dapat dipisahkan menjadi dua bagian, yaitu:

1. *Data type definition* yaitu menjelaskan tipe data yang digunakan oleh atribut pada masing-masing tabel yang ada di basisdata listrik_online.
2. *Data field description* yaitu menjelaskan keterangan setiap atribut pada masing-masing tabel yang ada di basisdata listrik_online.

4.3.2.2.1 Data Type Definition

Data type definition menjelaskan secara rinci mengenai tipe data yang digunakan setiap kolom pada masing-masing tabel basis data listrik_online.

Tabel-tabel yang terdapat dalam basis data listrik_online adalah :

✓ Tabel admin_access

Tabel 4.1 Tabel *data type definition* dari tabel admin_access

Table Name	No	Field Name	Type	Key	Null	Width
admin_access	01	username	varchar	PK	Not null	50
	02	password	varchar			100
	03	akses	enum			

Sumber : Perancangan

✓ Tabel bank

Tabel 4.2 Tabel *data type definition* dari tabel bank

Table Name	No	Field Name	Type	Key	Null	Width
bank	01	no_rek	varchar	PK	Not null	50
	02	pin	varchar			100
	03	pemilik	varchar			50
	04	saldo	bigint			20

Sumber : Perancangan

✓ **Biaya_pemakaian**

Tabel 4.3 Tabel *data type definition* dari tabel biaya_pemakaian

Table Name	No.	Field Name	Type	Key	Null	Width
biaya_pemakaian	01	blok	varchar	PK	not null	50
	02	kode_tarif	varchar			50
	03	batas	varchar			50
	04	biaya	int			11

Sumber : Perancangan

✓ **Tabel golongan_daya**

Tabel 4.4 Tabel *data type definition* dari tabel biaya_pemakaian

Table Name	No.	Field Name	Type	Key	Null	Width
golongan_daya	01	kode_tarif	varchar	PK	not null	50
	02	golongan	varchar			50
	03	daya	varchar			50
	04	beban	int			11

Sumber : Perancangan

✓ **Informasi_pemasangan**

Tabel 4.5 Tabel *data type definition* dari tabel informasi_pemasangan

Table Name	No	Field Name	Type	Key	Null	Width
informasi_pemasangan	01	kode	int	PK	not null	11
	02	keterangan	text			
	03	expired_date	date			

Sumber : Perancangan

✓ **Informasi_pemadaman**

Tabel 4.6 Tabel *data type definition* dari tabel informasi_pemadaman

Table Name	No	Field Name	Type	Key	Null	Width
informasi_pemadaman	01	kode	bigint	PK	not null	20
	02	waktu_mulai	datetime			
	03	waktu_selesai	datetime			
	04	lokasi_pemadaman	varchar			255
	05	keterangan	text			

Sumber : Perancangan

✓ **Keluhan**

Tabel 4.7 Tabel *data type definition* dari tabel keluhan

Table Name	No	Field Name	Type	Key	Null	Width
------------	----	------------	------	-----	------	-------

keluhan	01	kode	bigint	PK	not null	20
	02	id_pelanggan	varchar			20
	03	tanggal	datetime			
	04	keluhan	text			
	05	reply	text			
	06	reply_date	datetime			

Sumber : Perancangan

✓ Pelanggan

Tabel 4.8 Tabel *data type definition* dari tabel pelanggan

Table Name	No	Field Name	Type	Key	Null	Width
pelanggan	01	username	varchar			50
	02	password	varchar			100
	03	nama_pelanggan	varchar			100
	04	alamat	varchar			100
	05	telepon	varchar			20
	06	daya	varchar			20
	07	no_rek_listrik	varchar	PK	not null	50
	08	no_rekening_bank	varchar			20
	09	status	enum			

Sumber : Perancangan

✓ Tagihan

tabel 4.9 Tabel *data type definition* dari tabel tagihan

Table Name	No	Field Name	Type	Key	Null	Width
tagihan	01	kode_tagihan	bigint	PK	not null	20
	02	no_rekening_listrik	varchar			20
	03	tanggal_tagihan	date			
	04	meter_awal	int			11
	05	meter_akhir	int			11
	06	biaya_beban	bigint			20
	07	biaya_penggunaan	bigint			20
	08	denda	bigint			11
	09	total_biaya	bigint			20
	10	tanggal_bayar	date			
	11	tanggal_terlambat	date			

Sumber : Perancangan

4.3.2.2.2 Data field Description

Data field description menjelaskan keterangan seluruh kolom (*field*) dalam masing-masing tabel basis data listrik_online. *Data field description* basis data listrik_online ditunjukkan pada Tabel 4.10.

tabel 4.10 Tabel data field description basisdata listrik_online

Table Name	No	Field Name	Field Description
admin_access	01	username	id admin
	02	password	password admin
	03	akses	hak akses admin
bank	01	no_rek	nomer rekening bank
	02	pin	pin bank
	03	pemilik	nama pemilik
	04	saldo	saldo tabungan
biaya_pemakaian	01	blok	blok listrik
	02	kode_tarif	kode tarif
	03	batas	batas blok
	04	biaya	biaya tagihan
golongan_daya	01	kode_tarif	kode tarif
	02	golongan	golongan daya
	03	daya	daya listrik
	04	beban	tarif beban listrik
info_pemasangan	01	kode	kode pemasangan
	02	keterangan	informasi pemasangan
	03	expired_date	batas informasi pemasangan
info_pemadaman	01	kode	kode pemadaman
	02	waktu_mulai	waktu mulai pemadaman
	03	waktu_selesai	waktu selesai pemadaman
	04	lokasi_pemadaman	tempat dilakukan pemadaman
	05	keterangan	informasi pemadaman
keluhan	01	kode	kode keluhan
	02	id_pelanggan	id pelanggan
	03	tanggal	tanggal mengisi keluhan
	04	keluhan	isi keluhan
	05	reply	balasan keluhan dari admin
	06	reply_date	Tanggal balasan keluhan
pelanggan	01	username	id pelanggan
	02	password	password pelanggan
	03	nama_pelanggan	nama pelanggan
	04	alamat	alamat pelanggan
	05	telepon	telepon pelanggan

	06	daya	daya beban listrik pelanggan
	07	no_rek_listrik	nomer rekening listrik pelanggan
	08	no_rekening_bank	nomer rekening bank pelanggan
	09	status	status pelanggan
tagihan	01	kode_tagihan	kode tagihan
	02	no_rekening_listrik	nomer rekening listrik
	03	tanggal_tagihan	tanggal tagihan
	04	meter_awal	KWH meter awal
	05	meter_akhir	KWH meter akhir
	06	biaya_beban	biaya beban listrik sesuai dengan daya
	07	biaya_penggunaan	biaya penggunaan listrik
	08	denda	denda apabila terjadi keterlambatan
	09	total_biaya	total biaya tagihan listrik
	10	tanggal_bayar	tanggal pembayaran
	11	tanggal_terlambat	tanggal terlambat

Sumber : Perancangan

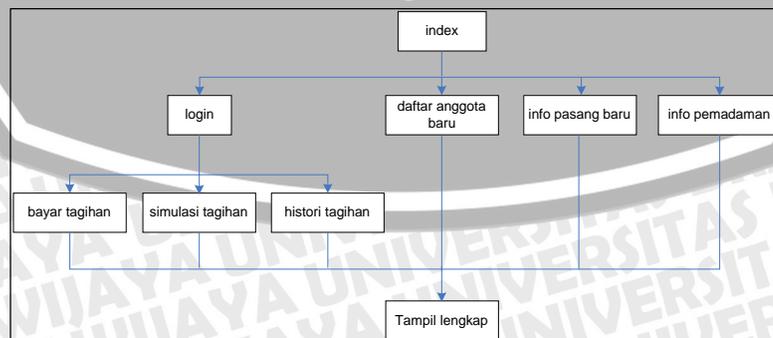
4.3.3 Site Map Sistem

Site map menggambarkan menu-menu yang akan diterapkan pada sistem yang akan dibuat, sehingga dapat diketahui alur pengaksesan dari sistem yang akan dibuat.

Halaman situs WAP pembayaran rekening listrik ini diproses oleh *server*. Halaman ini berisi tujuh *link* yang dapat diakses menuju halaman yang berbeda. *User* anggota yang telah melakukan *login* dan tervalidasi dapat mengakses keseluruhan sistem sesuai dengan batasan yang diberikan oleh sistem.

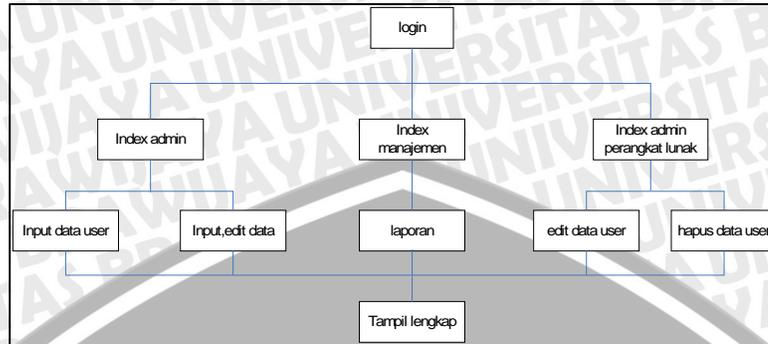
Halaman WAP digunakan oleh *user* anggota dan pengunjung. Halaman WEB digunakan oleh *user* admin dan manajemen untuk memasukkan dan mengubah data, dan melihat laporan-laporan yang diberikan oleh sistem.

Site map sistem yang dibuat dapat ditunjukkan pada gambar dibawah ini:



Gambar 4.18 *site map* sistem pembayaran rekening listrik (antarmuka WAP)

Sumber : Perancangan



Gambar 4.19 site map sistem pembayaran rekening listrik (antarmuka WEB)

Sumber : Perancangan

4.3.4 Perancangan Antarmuka User

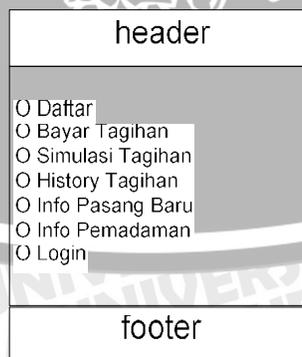
Antarmuka merupakan bagian yang menghubungkan antara sistem dengan *user*. Dengan demikian suatu perancangan antarmuka yang baik akan memudahkan *user* untuk berinteraksi dengan sistem yang dibuat.

4.3.4.1 Antarmuka User Pengunjung dan Anggota

User pengunjung merupakan *user* yang mempunyai hak akses paling terbatas karena untuk mengakses sistem ini, *user* pengunjung tidak memerlukan *login*.

a. Tampilan Index Pengunjung dan Anggota

Menu ini merupakan halaman yang pertama kali diakses *user* pada saat alamat URL situs ini diakses melalui telepon genggam. Pada halaman ini terdapat delapan *link*, yaitu: Daftar Anggota, Bayar Tagihan, Simulasi Tagihan, History Tagihan, Info Pasang Baru, Info Pemadaman, Keluhan dan Login. *Link* yang dapat diakses oleh *user* pengunjung adalah Daftar Anggota, Info Pasang Baru, dan Info Pemadaman.



Gambar 4.20 Halaman Index Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS

Sumber : Perancangan

b. Tampilan Daftar

Pada halaman ini, *user* pengunjung dapat mendaftarkan diri menjadi anggota agar dapat melakukan pembayaran listrik. *Field-field* yang harus diisi adalah: "username", "password", "nama", "alamat", "no.tlp", "daya", "no. Rekening Listrik", dan "no. Rekening Bank". Apabila semua data telah dimasukkan dengan lengkap pada *form* pendaftaran ini, maka data akan tersimpan pada *data store dt_anggota*. Dan apabila data yang diisi tidak lengkap, *username* atau nomor rekening listrik yang dimasukkan telah digunakan, maka proses pendaftaran akan diulangi kembali.

header	
<input type="text"/>	O Username : []
<input type="password"/>	O Password : []
<input type="text"/>	O Nama : []
<input type="text"/>	O Alamat : []
<input type="text"/>	O No.tlp : []
<input type="text"/>	O Daya : []
<input type="text"/>	O Rekening Listrik : []
<input type="text"/>	O Rekening Bank : []
DAFTAR	
footer	

Gambar 4.21 Halaman Daftar Anggota Baru Pengunjung

Sumber : Perancangan

c. Tampilan Login Anggota

Menu ini merupakan halaman yang ditampilkan bagi *user* yang menekan *link Login* pada halaman *index*. Apabila akan melakukan pembayaran tagihan listrik, maka *field* "Username" dan "Password" harus diisi terlebih dahulu untuk dapat menuju halaman berikutnya. Proses *login* akan gagal, dan akan memunculkan halaman kegagalan proses *login* apabila *field* "Username" dan "Password" diisi dengan data yang salah, atau data yang belum terdaftar pada sistem.

header	
O Username	: []
O Password	: []
LOGIN	
footer	

Gambar 4.22 Halaman Login Anggota
Sumber : *Perancangan*

d. Tampilan Konfirmasi Login Anggota

Menu ini merupakan halaman yang ditampilkan apabila *user* melakukan *login* dengan benar. Disediakan dua *link*: "Pesan Tiket" untuk melakukan proses pemesanan tiket, dan *link* "Batal Tiket" untuk melakukan proses pembatalan pemesanan tiket.

header	
Text Informasi...	
O Daftar	
O Bayar Tagihan	
O Simulasi Tagihan	
O History Tagihan	
O Info Pasang Baru	
O Info Pemadaman	
O Logout	
footer	

Gambar 4.23 Halaman Konfirmasi Login
Sumber : *Perancangan*

e. Tampilan Pembayaran Tagihan Listrik

Menu ini merupakan halaman yang ditampilkan bagi *user* yang menekan *link* "Bayar Tagihan" pada halaman sebelumnya (Konfirmasi *Login*). Tersedia *field* yang harus diisi, yaitu: "No. Rekening Bank".

header	
Text Informasi...	
O No.rekening	: []
BAYAR	
footer	

Gambar 4.24 Halaman Pembayaran Tagihan Listrik
Sumber : *Perancangan*

f. Tampilan Simulasi Tagihan

Halaman ini ditampilkan setelah *user* anggota memilih simulasi tagihan. Pada halaman ini *user* akan ditunjukkan tampilan informasi rekening listrik dan masukan penggunaan kWh meter.

header	
Text Informasi...	
Penggunaan kWh meter	: []
SIMULASI	
footer	

Gambar 4.25 Halaman Simulasi Tagihan
Sumber : Perancangan

g. Tampilan History Tagihan

Menu ini merupakan halaman yang ditampilkan bagi *user* setelah memilih history tagihan. Tersedia dua *field* yang harus diisi, yaitu "tanggal awal", dan "tanggal akhir". *Link* "Bayar sekarang" akan muncul apabila *user* anggota belum membayar tagihan listrik. Tulisan tersebut akan hilang apabila *user* anggota telah membayar tagihan listrik.

header	
Text Informasi...	
Tanggal	: [] s d []
LIHAT HISTORY	
Text Informasi...	
footer	

Gambar 4.26 Halaman History Tagihan
Sumber : Perancangan

h. Tampilan Info Pasang Baru

Halaman ini menampilkan informasi tentang pemasangan listrik. *User* akan menerima informasi dari admin, apabila tidak ada informasi terbaru tentang pemasangan listrik, maka akan muncul pesan "tidak ada informasi".

header	
Text Informasi...	
footer	

Gambar 4.27 Halaman Info Pasang Baru

Sumber : Perancangan

i. Tampilan Info Pemadaman

Halaman ini menampilkan informasi tentang pemadaman listrik yang akan dilakukan oleh pihak PLN. Apabila tidak ada informasi terbaru tentang pemadaman dari pihak PLN, maka akan muncul pesan “tidak ada informasi”.

header
Text Informasi...
footer

Gambar 4.28 Halaman Informasi Pemadaman
Sumber : Perancangan

j. Tampilan Keluhan

Halaman ini merupakan interaksi antara *user* anggota dengan pihak PLN. *User* anggota dapat mengetikkan keluhan dan dapat melihat tanggapan dari pihak PLN. *Field* yang harus diisi oleh *user* anggota adalah form “keluhan” .

header
Text Informasi...
Keluhan : []
KIRIM KELUHAN
Text Informasi...
footer

Gambar 4.29 Halaman Keluhan Bagi Anggota
Sumber : Perancangan

4.3.4.2 Antarmuka User Admin

User admin merupakan *user* yang dapat bertindak sebagai pengelola data yang ada dari masing-masing *data store* yang berhubungan dengan data-data pembayaran.

a. Tampilan Login Admin, Admin Perangkat Lunak, dan Manajemen

Menu ini merupakan halaman *login* bagi *user* admin, admin perangkat lunak, dan manajemen. Admin, admin perangkat lunak, dan manajemen mempunyai nama *user* dan *password* yang berbeda, yang akan menuju ke halaman yang berbeda pula.

header	
Sign in to Web Admin	
Username	: <input type="text"/>
Password	: <input type="password"/>
<input type="button" value="Sign in"/>	
footer	

Gambar 4.30 Halaman Login Admin, Admin Perangkat Lunak, dan Manajemen
 Sumber : *Perancangan*

b. Tampilan Menu Utama Admin

Menu ini merupakan menu utama untuk *user* admin. Berisi *link-link* yang dapat dipilih untuk memasukkan data baru atau mengedit data yang sudah ada sebelumnya.

header		
<input type="button" value="Pelanggan"/>	<input type="button" value="Tagihan"/>	<input type="button" value="Info Pemadaman"/>
<input type="button" value="Info Pasang Baru"/>	<input type="button" value="Ubah Password"/>	<input type="button" value="Log Out"/>
focter		

Gambar 4.31 Halaman Masukkan dan Edit Data
 Sumber : *Perancangan*

c. Tampilan Form Masukkan Data Pelanggan

Form ini digunakan untuk memasukkan data tentang pelanggan oleh admin.

header	
Username	<input type="text"/>
Password	<input type="password"/>
Nama	<input type="text"/>
Alamat	<input type="text"/>
No.tlp	<input type="text"/>
Daya	<input type="text"/>
Rek. Listrik	<input type="text"/>
Rek. Bank	<input type="text"/>
Status	<input type="text"/>
<input type="button" value="Simpan"/>	
Informasi Text...	
footer	

Gambar 4.32 Halaman Masukkan Data Pelanggan
 Sumber : *Perancangan*

d. Tampilan Form Masukkan Data Tagihan

Form ini digunakan untuk memasukkan data tentang tagihan oleh admin.

header	
Tanggal Tagihan	<input type="text"/>
No.Rek. Listrik	<input type="text"/>
Meter Awal	<input type="text"/>
Meter Akhir	<input type="text"/>
Tanggal Terlambat Bayar	<input type="text"/>
<input type="button" value="Simpan"/>	
Informasi Text...	
footer	

Gambar 4.33 Halaman Masukkan Data Tagihan
Sumber : Perancangan

e. Tampilan Form Masukkan Data Pemadaman

Form ini digunakan untuk memasukkan data tentang pemadaman oleh admin.

header	
Waktu Mulai	Tanggal : <input type="text"/> ,Jam : <input type="text"/>
Waktu Selesai	Tanggal : <input type="text"/> ,Jam : <input type="text"/>
Lokasi Pemadaman	<input type="text"/>
Keterangan	<input type="text"/>
Information Text...	<input type="button" value="Simpan"/>
footer	

Gambar 4.34 Halaman Masukkan Data Pemadaman
Sumber : Perancangan

f. Tampilan Form Masukan Pasang Baru

Form ini digunakan untuk mengubah atau menghapus data tentang pasang baru listrik oleh admin.

header	
Form Informasi Pemasangan	
Keterangan	<input type="text"/>
Expired Date	<input type="text"/>
<input type="button" value="Simpan"/>	
Informasi Text...	
footer	

Gambar 4.35 Halaman informasi pasang baru
Sumber : Perancangan

g. Tampilan Form Edit Password

Form ini digunakan untuk mengubah atau menghapus password admin.

header	
Information Text...	
Old Password	<input type="text"/>
New Password	<input type="text"/>
<input type="button" value="Ubah Password"/>	
footer	

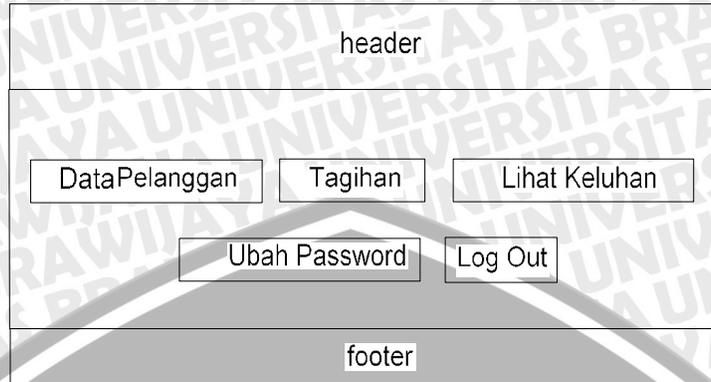
Gambar 4.36 Halaman Edit Password
Sumber : Perancangan

4.3.4.3 Antarmuka User Manajemen

User manajemen merupakan *user* dengan hak akses sebagai pihak manajer. Manajemen dapat melihat laporan-laporan dari sistem pemesanan tiket bus malam ini.

a. Tampilan Menu Utama Manajemen

Menu ini merupakan menu utama bagi *user* manajemen setelah melakukan *login* sebagai manajemen. Berisi *link-link* yang dapat dipilih untuk melihat laporan-laporan yang ada.



Gambar 4.37 Halaman Manajemen
 Sumber : Perancangan

b. Tampilan Lihat Laporan Data Pelanggan

Merupakan tampilan laporan tentang data pelanggan yang dapat dilihat oleh manajemen.



Gambar 4.38 Halaman Lihat Laporan Data Pelanggan
 Sumber : Perancangan

c. Tampilan Lihat Laporan Data Tagihan

Merupakan tampilan laporan tentang data tagihan yang dapat dilihat oleh manajemen.

header											
Kode	No.Rek.Listrik	Tgl tagihan	Meter awal	Meter Akhir	Biaya beban	Biaya penggunaan	denda	Total biaya	Tanggal bayar	Tanggal terlambat	status

footer

Gambar 4.39 Halaman Lihat Laporan Data Tagihan

Sumber : *Perancangan*

d. Tampilan Lihat Laporan Keluhan

Merupakan tampilan laporan tentang keluhan *user* anggota yang dapat dilihat oleh manajemen.



header			
No.Rek.Listrik	nama	tanggal	keluhan

footer

Gambar 4.40 Halaman Laporan Keluhan

Sumber : *Perancangan*

e. Tampilan Form Edit Password

Form ini digunakan untuk mengubah atau menghapus password manajemen.

header	
Information Text...	
Old Password	<input type="text"/>
New Password	<input type="text"/>
<input type="button" value="Ubah Password"/>	
footer	

Gambar 4.41 Halaman Edit Password
 Sumber : Perancangan

4.3.4.4 Antarmuka User Admin Perangkat Lunak

User admin perangkat lunak merupakan admin yang bertindak sebagai pengelola data pada *data store* yang berhubungan dengan hak akses *user* atau registrasi anggota.

a. Tampilan Menu Utama Admin Perangkat Lunak

Menu ini merupakan menu utama bagi *user* admin perangkat lunak setelah melakukan *login* sebagai admin perangkat lunak. Berisi *link-link* yang dapat dipilih untuk mengelola data *user*.

header		
<input type="button" value="User"/>	<input type="button" value="Ubah Password"/>	<input type="button" value="Log Out"/>
footer		

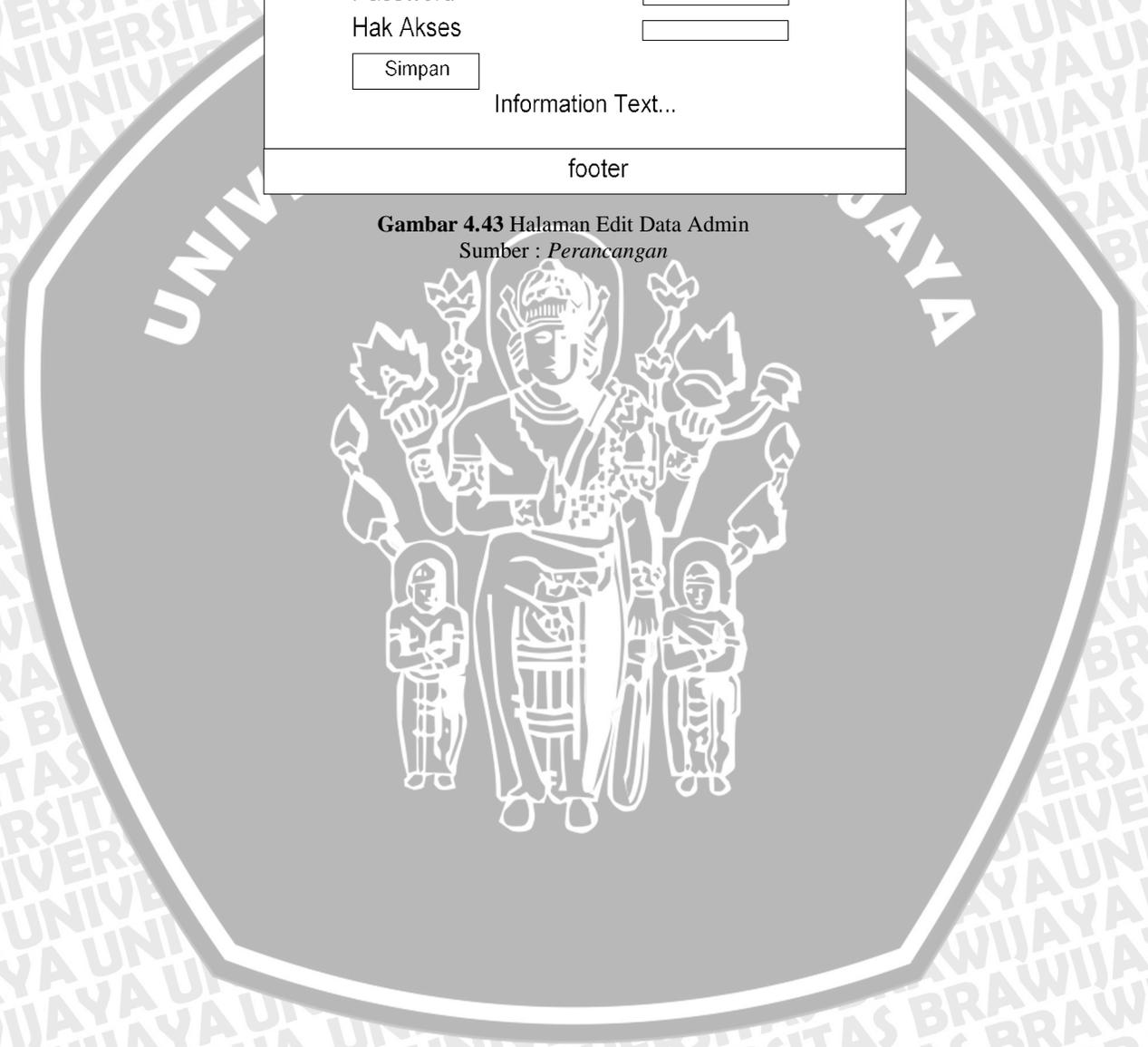
Gambar 4.42 Halaman Menu Utama Admin Perangkat Lunak
 Sumber : Perancangan

b. Tampilan Form Edit Data Admin, Manajemen, Admin Perangkat Lunak

Form ini digunakan untuk mengubah *username* admin atau *password* admin atau keduanya, oleh admin perangkat lunak.

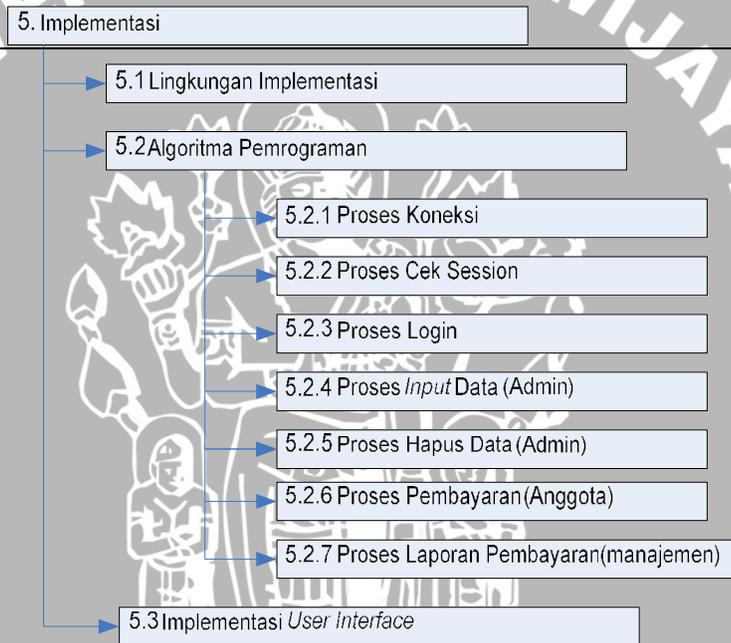
header	
Username	<input type="text"/>
Password	<input type="password"/>
Hak Akses	<input type="text"/>
<input type="button" value="Simpan"/>	
Information Text...	
footer	

Gambar 4.43 Halaman Edit Data Admin
Sumber : Perancangan



BAB V IMPLEMENTASI

Bab ini membahas implementasi Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS dari proses analisis dan perancangan yang dibuat sebelumnya ke dalam kode (*coding*). Implementasi diawali dengan perancangan algoritma program, dilanjutkan dengan penerapan algoritma program ke dalam bahasa pemrograman dan implementasi *user interface*. Diagram pohon implementasi ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Diagram Pohon Implementasi Sistem
Sumber: *Implementasi*

5.1 Lingkungan Implementasi

Sistem diimplementasikan dengan menggunakan bahasa pemrograman web PHP, *database* MySQL dan *Apache* sebagai *web server*. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut :

1. Konfigurasi server

Spesifikasi *hardware* minimal:

- CPU Intel Pentium III 800 MHz

- Memori 256 Mb SD RAM

Spesifikasi *software* :

- Microsoft Windows XP Profesional Version 2002 Service Pack 2
- PHP Versi 5.2.5
- MySQL Versi 5.0.5
- Apache *Web Server* Versi 2.2.2
- Emulator WAP M3Gate Versi 0.5

2. Konfigurasi *browser*

- Tipe ponsel : Nokia E63
- Menggunakan jaringan GPRS dengan operator Indosat IM3

Selain implementasi yang sudah disebutkan sebelumnya, untuk memudahkan implementasi khususnya dalam pembuatan algoritma dan kode program digunakan beberapa *tool* pendukung sebagai berikut:

1. Microsoft Visio 2003.

Merupakan tool untuk mendisain algoritma dalam bentuk *flow chart*.

2. Adobe Photoshop CS2

Digunakan untuk mendisain tampilan halaman web dalam bentuk format image.

3. Macromedia Dreamweaver CS3

Digunakan untuk mengedit tampilan web dan menambahkan kode program agar web yang dihasilkan lebih dinamis

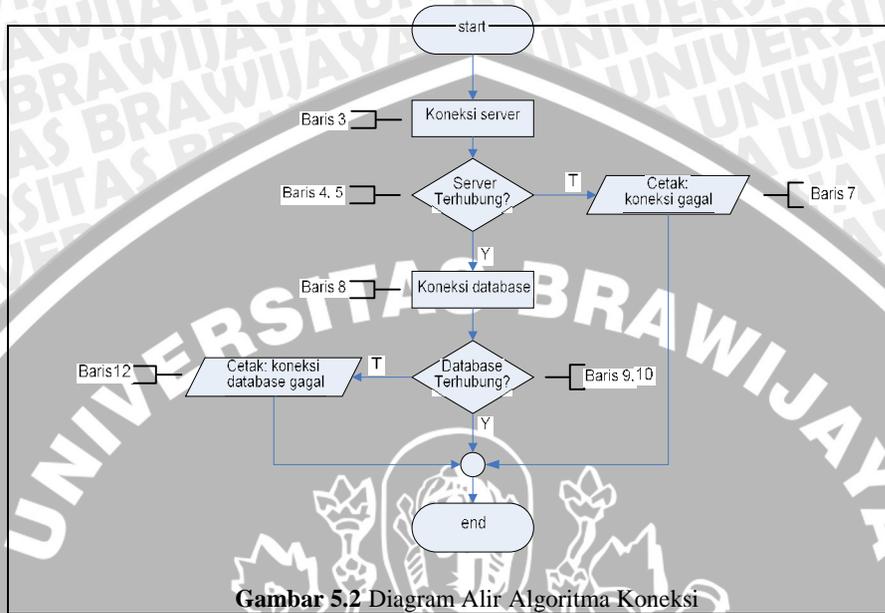
5.2 Algoritma Pemrograman

Berikut dijelaskan representasi algoritma berupa diagram alir dan *pseudocode* setiap proses. Proses-proses tersebut adalah proses koneksi, proses cek *session*, proses *login*, proses *input* data, proses hapus data, proses pembayaran, proses laporan pembayaran.

5.2.1 Proses Koneksi

Proses koneksi dimulai dengan permintaan koneksi pada server . Apabila berhasil koneksi maka akan diteruskan dengan permintaan untuk me milih *database* dalam hal ini

adalah *database* listrik_online. Proses koneksi tersebut ditunjukkan pada *flowchart* dalam Gambar 5.2.



Gambar 5.2 Diagram Alir Algoritma Koneksi
Sumber: *Implementasi*

Pseudocode dari proses koneksi adalah sebagai berikut:

```

1 |  PROCEDURE koneksi
2 |      Prosedur pengecekan koneksi dengan server. Dan
      prosedur pemilihan database yg akan digunakan jika
      proses koneksi berhasil
3 |  CONNECT TO server
4 |  IF server connected
5 |      SELECT database
6 |  ELSE
7 |      PRINT "koneksi gagal"
8 |  CONNECT TO database
9 |  IF database connected
10 |     SELECT table
11 |  ELSE
12 |     PRINT "koneksi database gagal"
13 |  ENDIF
14 |  ENDPROC
  
```

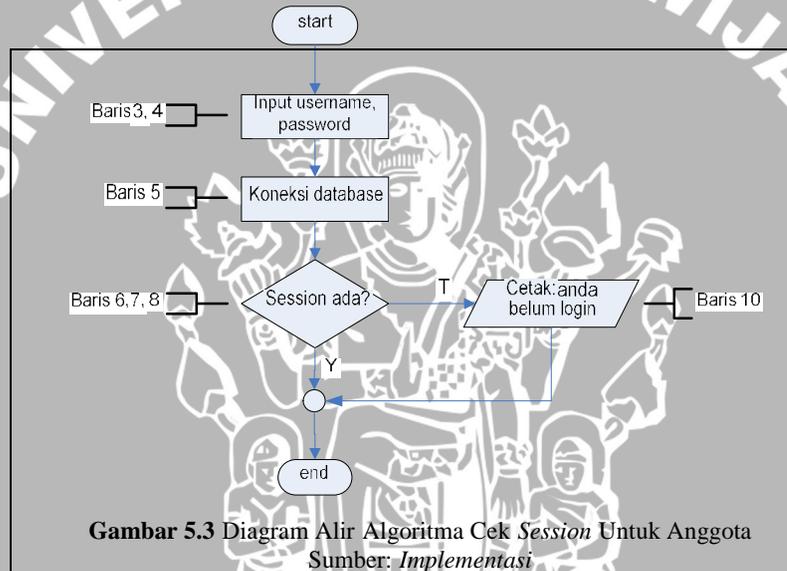
Penjelasan dari *pseudocode* proses koneksi adalah:

Baris 3 menjelaskan tentang prosedur pengecekan koneksi dengan *server*. Baris 4, 5, 6, dan 7 menjelaskan tentang kondisi koneksi dengan *server*, apabila terhubung maka akan dilanjutkan ke pemilihan database, apabila gagal maka akan muncul pesan.

Baris 8 menjelaskan tentang prosedur pemilihan database yang akan digunakan. Baris 9, 10, 11, 12, menjelaskan tentang kondisi koneksi dengan database, apabila terhubung maka akan dilanjutkan ke pemilihan tabel yang akan digunakan, apabila gagal maka akan muncul pesan.

5.2.2 Proses Cek *Session*

Proses Cek *Session* digunakan untuk membatasi *user* dalam mengakses halaman yang tersedia yaitu dimulai dengan proses mulai *session*. Proses cek *session* pada sistem ini terbagi empat, yaitu proses cek *session user admin*, admin perangkat lunak, manajemen, dan anggota. Proses - proses tersebut ditunjukkan pada *flowchart* dalam Gambar 5.3.



Pseudocode dari proses cek *session* adalah sebagai berikut:

```

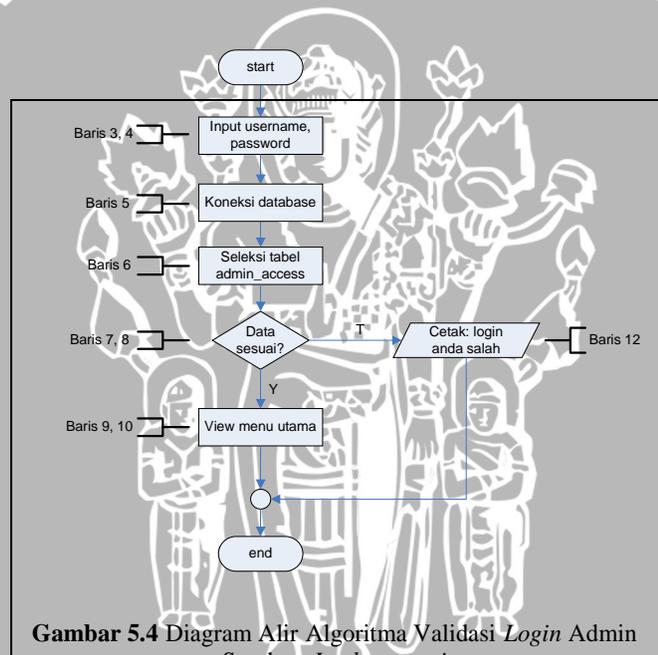
1 | PROCEDURE session
2 |     Prosedur pengecekan untuk membatasi user dalam
3 |     mengakses halaman yang tersedia
4 |     TYPE username, password IS STRING
5 |     INPUT user_value and password_value
6 |     CONNECT TO database
7 |     IF (username=user_value and
8 |     password=password_value) THEN
9 |     SESSION REGISTER as user_value and
10 |    password_value
11 | ELSE
12 |     PRINT "Anda belum login!!!"
13 | ENDIF
14 | ENDPROC
  
```

Penjelasan dari *pseudocode* proses cek session adalah:

Baris 3 dan 4 menjelaskan tentang prosedur pengecekan hak akses *user* dengan masukan *username* dan *password*. Baris 5 menjelaskan tentang koneksi dengan database. Baris 6, 7, 8 menjelaskan tentang kondisi dari proses cek *session*, apabila *username* dan *password* ada dalam database maka akan dilanjutkan ke proses berikutnya, apabila gagal maka akan muncul pesan.

5.2.3 Proses Login

Pada proses *login* terdapat empat login, yaitu *login* admin, *login* admin perangkat lunak, *login* manajemen dan *login* user. Proses *login* admin sama dengan apa yang terjadi pada proses *login* admin perangkat lunak, *login* manajemen, dan *login* anggota. Proses *login* admin ditunjukkan pada *flowchart* dalam Gambar 5.4.



Gambar 5.4 Diagram Alir Algoritma Validasi Login Admin

Sumber: Implementasi

Pseudocode dari algoritma proses login admin adalah sebagai berikut :

```

1|  PROCEDURE login
2|    Prosedur pengecekan login serta untuk menampilkan
3|    menu yang dapat diakses anggota
4|    TYPE username, passwd IS STRING
5|    INPUT user_value and passwd_value
6|    CONNECT TO database
7|    SELECT username,password FROM database
8|    IF (username=user_value and
    password=passwd_value) THEN
    SESSION REGISTER as user_value and
  
```

```

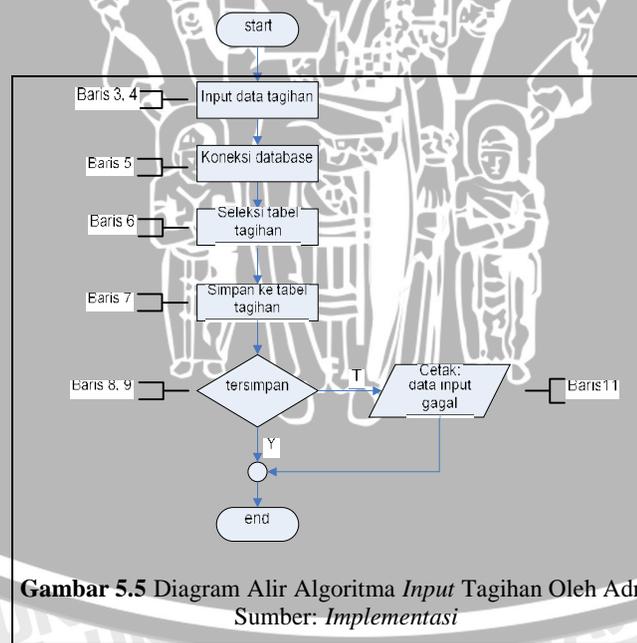
password_value
9 | PRINT "Selamat Datang"
10 | VIEW menu utama
11 | ELSE
12 | PRINT "Login Anda Salah"
13 | ENDIF
14 | ENDPROC
    
```

Penjelasan dari *pseudocode* proses login adalah:

Baris 3, dan 4 menjelaskan tentang proses *input* data *login*. Baris 5 menjelaskan tentang koneksi dengan database. Baris 6 menjelaskan tentang seleksi tab el database yang sesuai dengan *input* data *login*. Baris 7, 8, 9, 10, 11, 12 menjelaskan tentang kondisi pengecekan data *input* dengan database, apabila benar maka akan muncul pesan dan menu utama admin, apabila salah maka akan muncul pesan "Login Anda Salah".

5.2.4 Proses Input Data

Proses ini adalah proses yang dilakukan oleh admin yang merupakan proses untuk memasukkan data-data ke dalam *database*. Proses *input* tagihan ditunjukkan pada *flowchart* dalam Gambar 5.5.



Gambar 5.5 Diagram Alir Algoritma *Input* Tagihan Oleh Admin
 Sumber: *Implementasi*

Pseudocode dari algoritma proses input data adalah sebagai berikut :

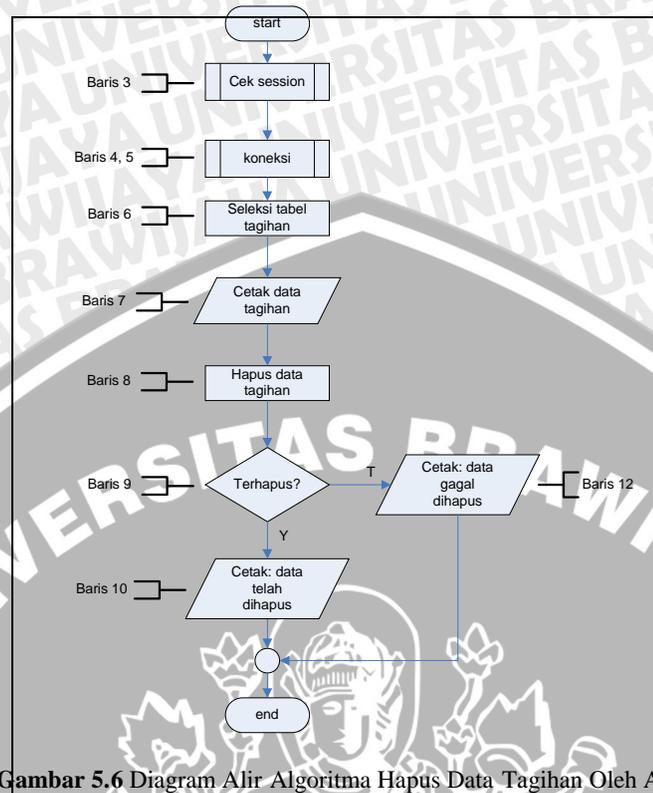
```
1|  PROCEDURE input data
2|      prosedur untuk memasukkan data kedalam
      database
3|  TYPE tanggal, rekening, isi_data IS STRING
4|  INPUT rekening value and isi_data_value
5|  CONNECT TO database
6|  SELECT tanggal, rekening, isi_data FROM database
7|  SQL INSERT INTO database VALUES (tanggal,
      rekening_value and isi_data_value)
8|      IF (sukses) THEN
9|          simpan
10|     ELSE
11|         PRINT "Data Input Gagal"
12|     ENDIF
13| ENDPROC
```

Penjelasan dari *pseudocode* proses *input* data tagihan adalah:

Baris 3, dan 4 menjelaskan tentang proses *input* data tagihan. Baris 5 menjelaskan tentang koneksi database. Baris 6 menjelaskan tentang seleksi tabel database yang sesuai dengan data tagihan, yaitu tabel tagihan. Baris 7 menjelaskan tentang proses penyimpanan data ke tabel tagihan. Baris 8, 9, 10, 11 menjelaskan tentang kondisi proses *input* data tagihan, apabila data berhasil disimpan, maka akan muncul pesan, apabila gagal maka akan muncul pesan "Data Input Gagal".

5.2.5 Proses Hapus Data

Proses ini adalah proses yang dilakukan oleh admin yang merupakan proses untuk menghapus data-data tagihan pelanggan dari *database*. Proses hapus data bus ditunjukkan pada *flowchart* dalam Gambar 5.6.



Gambar 5.6 Diagram Alir Algoritma Hapus Data Tagihan Oleh Admin
 Sumber: Implementasi

Pseudocode dari algoritma proses hapus data adalah sebagai berikut :

```

1 | PROCEDURE hapus data
2 |     prosedur untuk menghapus data dari database
3 | START session
4 | CONNECT TO server
5 | CONNECT TO database
6 | SELECT tabel tagihan FROM database
7 | PRINT data tagihan
8 | SQL DELETE FROM database WHERE id = (GET (id) )
9 | IF (delete succes) THEN
10 |     PRINT "data telah dihapus";
11 | ELSE
12 |     PRINT "Data gagal dihapus";
13 | ENDIF
14 | ENDPROC
  
```

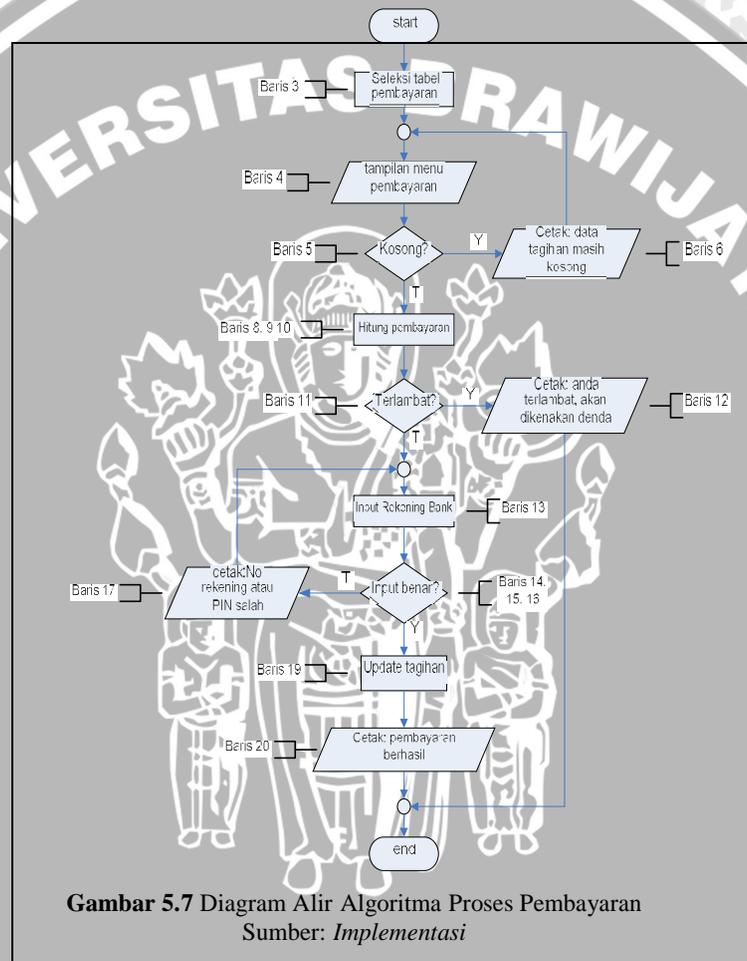
Penjelasan dari pseudocode proses hapus data tagihan adalah:

Baris 3, 4, dan 5 menjelaskan tentang proses cek *session*, koneksi *server* dan koneksi database. Baris 6 menjelaskan tentang seleksi tabel tagihan dari database. Baris 7 menampilkan data tagihan yang akan dihapus. Baris 8 menjelaskan tentang proses menghapus data dari database. Baris 9, 10, 11, 12 menjelaskan tentang kondisi

proses hapus data oleh admin, apabila sukses maka akan muncul pesan, apabila gagal maka akan muncul pesan “Data Gagal Dihapus”.

5.2.6 Proses Pembayaran

Proses ini adalah proses yang dilakukan oleh anggota yang merupakan proses pembayaran tagihan listrik yang sebelumnya telah diisi oleh admin. Proses transaksi pembayaran tagihan tersebut ditunjukkan pada *flowchart* dalam Gambar 5.7.



Pseudocode dari algoritma proses transaksi pembayaran adalah sebagai berikut :

```

1 | PROCEDURE pembayaran tagihan listrik
2 |   prosedur proses pembayaran tagihan listrik
   oleh user anggota
3 |   SELECT tabel pembayaran FROM database
4 |   VIEW menu pembayaran
5 |   IF empty
6 |     echo "Data tagihan masih kosong "
7 |   Else
8 |     variabel sql = "select * from tagihan where
   no_rekening_listrik="
  
```

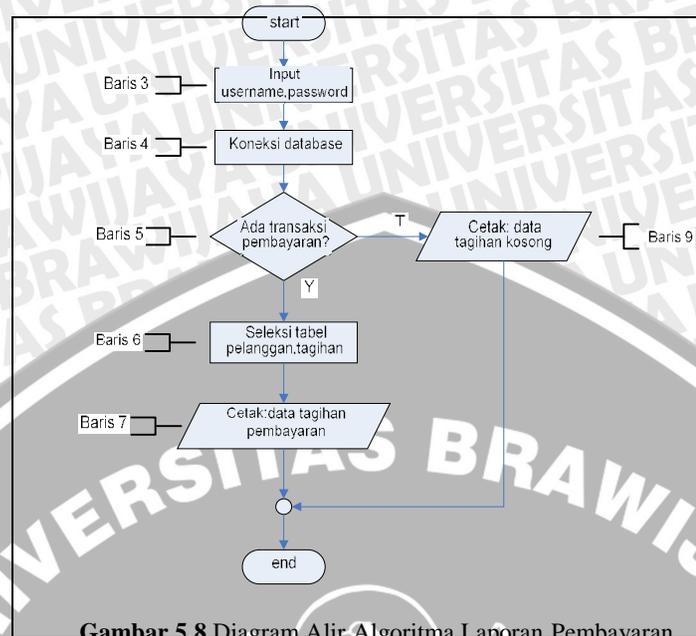
```
9|      variabel data=variabel db->fetch()
10|     variabel tanggal_bayar=date("Y-m-d")
11|     IF( variabel tanggal_bayar > variabel
12|       data['tanggal_terlambat'])
13|       echo "Anda terlambat, Anda akan dikenakan
14|         denda!!!"
15|     SQL INSERT INTO database VALUES (nomer rekening)
16|     IF variabel pin=md5([pin])
17|       variabel sql2= "select * from bank where
18|         no_rek='variabel_POST[nomer_rekening]' and
19|         pin='variabelpin'"
20|       variabel db->getNumRows() < 1
21|         echo "Nomer Rekening atau Pin Salah!!!"
22|     Else
23|       variabel sqlBayar="update tagihan"
24|       echo "Pembayaran berhasil, Sisa Saldo : "
25|     END IF
26|   END PROC
```

Penjelasan dari *pseudocode* proses pembayaran adalah:

Baris 3 menjelaskan tentang seleksi tabel pembayaran dari database. Baris 4 menampilkan menu pembayaran anggota. Baris 5, dan 6 menjelaskan tentang kondisi proses pembayaran, apabila data tagihan masih kosong, maka akan muncul pesan. Baris 8, 9, 10, 11, 12 menjelaskan tentang perhitungan pembayaran, apabila terjadi keterlambatan pembayaran, maka akan muncul pesan. Baris 13 menjelaskan tentang *input* data pembayaran, yaitu rekening anggota. Baris 14, 15, 16, 17, menjelaskan tentang kondisi proses pengecekan pin dan rekening, apabila terjadi kesalahan, maka akan muncul pesan. Baris 19, dan 20 menjelaskan tentang proses update saldo dan akan muncul pesan bahwa pembayaran berhasil.

5.2.7 Proses Laporan Transaksi

Proses ini adalah proses yang dilakukan oleh manajemen yang merupakan proses untuk melihat laporan transaksi yang terjadi. Proses laporan data tagihan tersebut ditunjukkan pada *flowchart* dalam Gambar 5.8.



Gambar 5.8 Diagram Alir Algoritma Laporan Pembayaran

Sumber: Implementasi

Pseudocode dari algoritma proses Laporan Pembayaran adalah sebagai berikut :

```

1 |  PROCEDURE laporan pembayaran
2 |      prosedur untuk melihat laporan pembayaran dari
3 |      database
4 |  TYPE id    IS STRING
5 |  CONNECT TO database
6 |  IF (GET (id) )
7 |      SQL SELECT FROM database WHERE id = (GET (id) )
8 |      PRINT "data tagihan";
9 |  ELSE
10 |      PRINT "Data tagihan kosong";
11 |  ENDIF
12 |  ENDPROC
  
```

Penjelasan dari *pseudocode* proses laporan pembayaran adalah:

Baris 3 menjelaskan tentang proses *input* data *login*, yaitu *username* dan *password*.

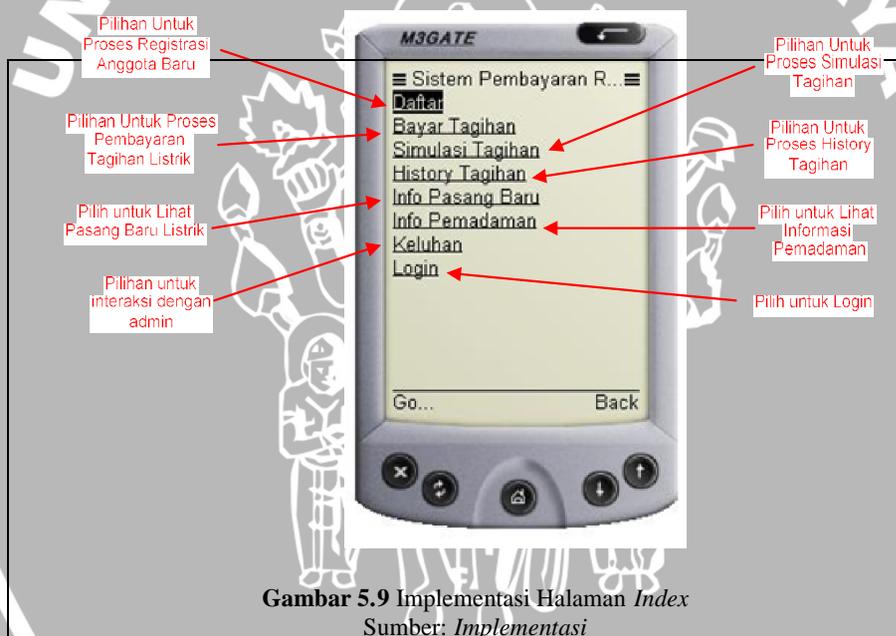
Baris 4 menjelaskan tentang proses koneksi database. Baris 5, 6, dan 7 menjelaskan tentang kondisi dari proses laporan pembayaran, apabila *input* data *login* benar, maka akan dilanjutkan seleksi tabel *pelanggan*, *tagihan*, dan laporan akan ditampilkan, apabila belum terjadi proses pembayaran maka akan muncul pesan "Data tagihan kosong".

5.3 Implementasi Antarmuka

Implementasi antarmuka terdiri dari implementasi antarmuka untuk pengunjung, anggota dan implementasi antarmuka untuk admin dan manajemen.

5.3.1 Implementasi Antarmuka WAP Anggota dan Pengunjung

Pertama kali *user* akan masuk ke halaman menu utama yang berisi tampilan gambaran tentang Sistem Pembayaran Tagihan Listrik. *User* pengunjung dapat memilih tiga *field* yaitu: Daftar Anggota Baru, Lihat Informasi Pasang Baru, Lihat Informasi Pemadaman. Sedangkan *field* sisanya hanya dapat digunakan oleh anggota karena memerlukan *username* dan *password* untuk melakukan *login*. Untuk implementasi halaman *index* Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Implementasi Halaman Index

Sumber: Implementasi

5.3.1.1 Implementasi Login Anggota

Anggota yang ingin memilih *field* Bayar Tagihan, Simulasi, History dan Keluhan harus melalui proses validasi *login*. Halaman *login* mempunyai dua *field* untuk diisi yaitu: Username dan Password. Untuk implementasi fasilitas *login* anggota ditunjukkan dalam Gambar 5.10.

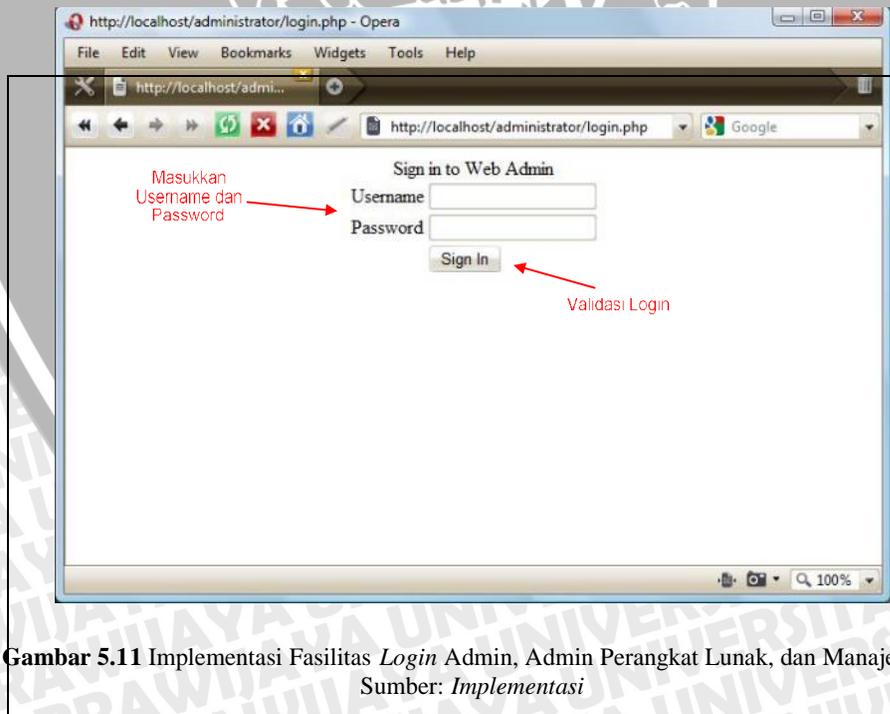


Gambar 5.10 Implementasi Halaman *Login* Anggota

Sumber: *Implementasi*

5.3.2 Implementasi Antarmuka Admin

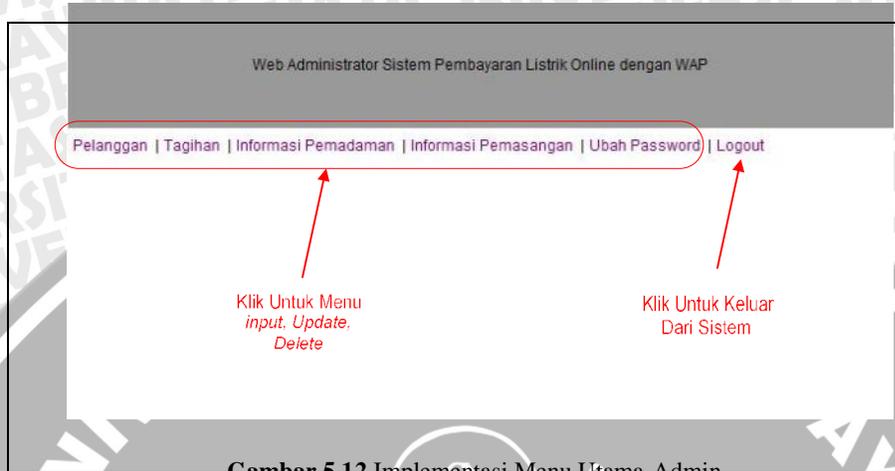
Admin, admin perangkat lunak, dan manajemen yang menggunakan aplikasi pada Sistem Pembayaran Rekening Listrik ini harus melalui proses validasi *login*. Halaman *login* mempunyai dua *field* untuk *username* dan *password*. Implementasi fasilitas *login* admin, admin perangkat lunak, dan manajemen ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Implementasi Fasilitas *Login* Admin, Admin Perangkat Lunak, dan Manajemen

Sumber: *Implementasi*

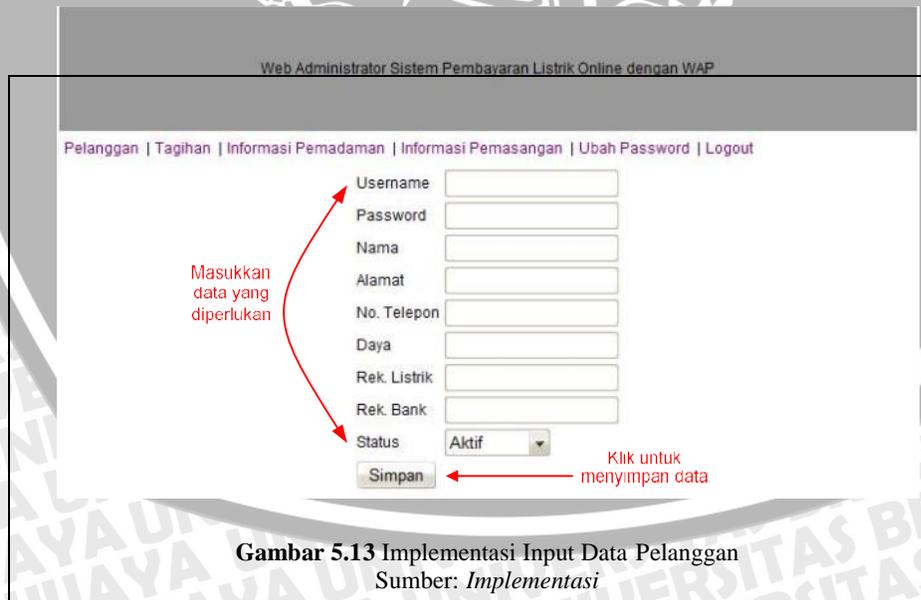
Apabila login admin valid, maka akan masuk ke halaman menu utama admin yang mana dalam halaman tersebut terdapat pilihan untuk proses input data dan edit data. Implementasi menu utama admin ditunjukkan dalam Gambar 5. 12.



Gambar 5.12 Implementasi Menu Utama Admin
Sumber: *Implementasi*

5.3.2.1 Implementasi Input Data

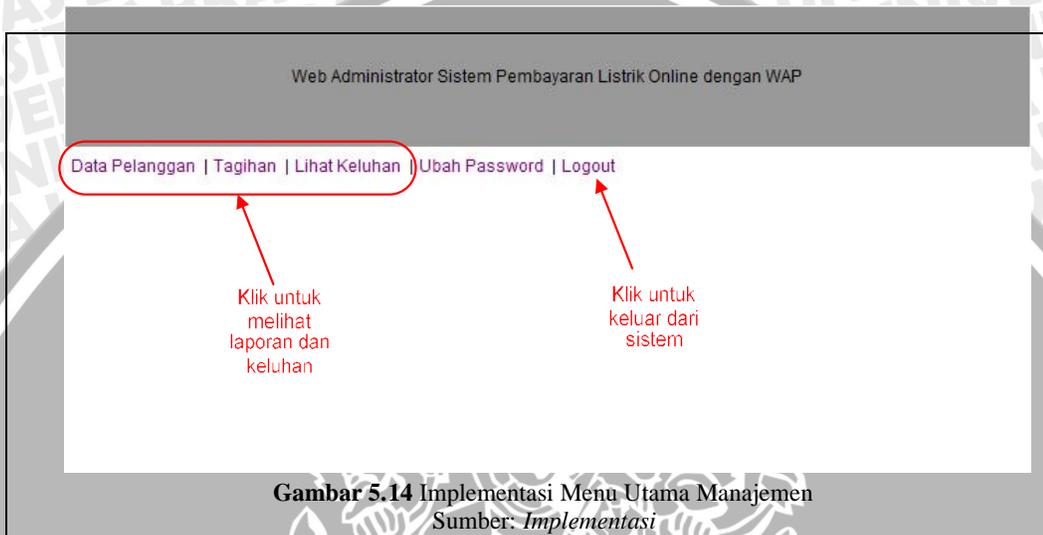
Admin dapat melakukan proses input data pelanggan. Proses input ini serupa dengan proses input data tagihan, informasi pemadaman, dan informasi pasang baru. Implementasi input data pelanggan ditunjukkan dalam Gambar 5.13.



Gambar 5.13 Implementasi Input Data Pelanggan
Sumber: *Implementasi*

5.3.3 Implementasi Antarmuka Manajemen

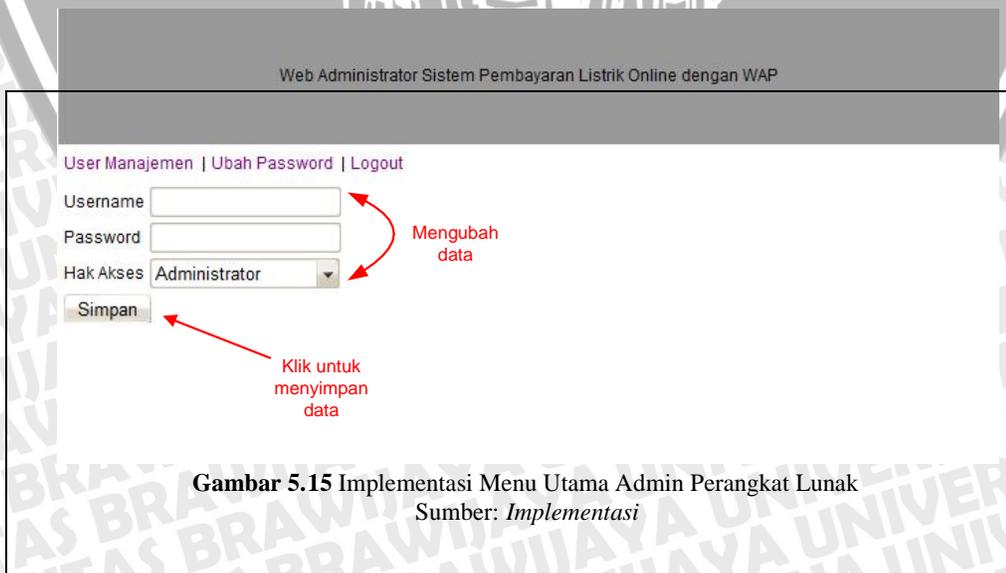
Apabila melakukan login sebagai manajemen, maka akan masuk ke antarmuka untuk manajemen. Pada menu utama ini, manajemen dapat memilih untuk melihat laporan yang telah disediakan. Implementasi menu utama manajemen ditunjukkan dalam Gambar 5.14.



Gambar 5.14 Implementasi Menu Utama Manajemen
Sumber: *Implementasi*

5.3.4 Implementasi Antarmuka Admin Perangkat Lunak

Apabila melakukan login sebagai admin perangkat lunak, maka akan masuk ke antarmuka untuk admin perangkat lunak. Pada menu utama ini, admin perangkat lunak dapat memilih proses input atau proses edit data yang telah disediakan. Implementasi menu utama admin perangkat lunak ditunjukkan dalam Gambar 5.15.



Gambar 5.15 Implementasi Menu Utama Admin Perangkat Lunak
Sumber: *Implementasi*

BAB VI PENGUJIAN

Bab ini membahas proses pengujian terhadap perangkat lunak Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS yang telah dibangun. Proses pengujian dilakukan melalui dua tahapan (strategi) yaitu pengujian algoritma dan pengujian validasi. Pengujian algoritma menggunakan teknik pengujian *white box* (*white box Testing*). Pengujian validasi menggunakan teknik pengujian *black box* (*black box Testing*).

6.1 Pengujian Algoritma

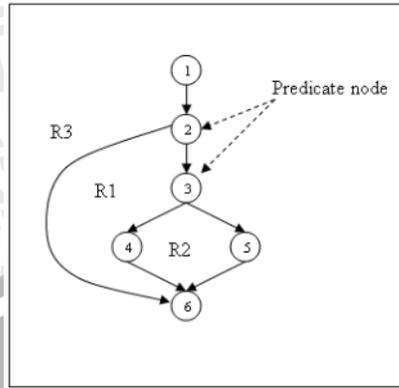
Teknik pengujian yang dilakukan adalah teknik pengujian kotak putih (*white box testing*) dengan teknik pengujian jalur dasar (*basis path testing*). Pada pengujian jalur dasar akan dilakukan pembuatan *programme design language* (PDL), pembuatan grafik alir (*flow graph*), menentukan kompleksitas siklomatis, menentukan jalur independen dan melakukan kasus uji.

6.1.1 Pengujian Proses Login

1. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi

```
PROCEDURE login
    Prosedur pengecekan login serta untuk
    menampilkan menu yang dapat diakses
    anggota
TYPE username, passwd IS STRING
INPUT user_value and passwd_value
CONNECT TO database
SELECT username,password FROM database
IF (username=user_value and
    password=password_value) THEN
    SESSION REGISTER as user_value and
    password_value
    PRINT "Selamat Datang"
    VIEW menu utama
ELSE
    PRINT "Login Anda Salah"
ENDIF
ENDPROC
```

2. Pembuatan grafik alir (*flow graph*) yang sesuai



Gambar 6.1 Flow Graph Proses login
Sumber : [Pengujian]

3. Penentuan kompleksitas siklomatis dari grafik alir resultan

$$V(G) = 3 \text{ region}$$

$$V(G) = 7 \text{ edge} - 6 \text{ node} + 2 = 3$$

$$V(G) = 2 \text{ predicate node} + 1 = 3$$

4. Penentuan jalur independen

Jalur 1 : 1 - 2 - 6

Jalur 2 : 1 - 2 - 3 - 5 - 6

Jalur 3 : 1 - 2 - 3 - 4 - 6

5. Melakukan kasus uji

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.1.

Tabel 6.1 Kasus uji untuk pengujian proses login

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	username = null or username != username value	User gagal login dan keluar pesan error	User gagal login
2	password = null or password != password value	User gagal login dan keluar pesan error	User gagal login dan keluar pesan error
3	username = username value and password = password value	User sukses login dan tampilkan menu pembayaran	User sukses login dan tampilkan menu pembayaran

Sumber : [Pengujian]

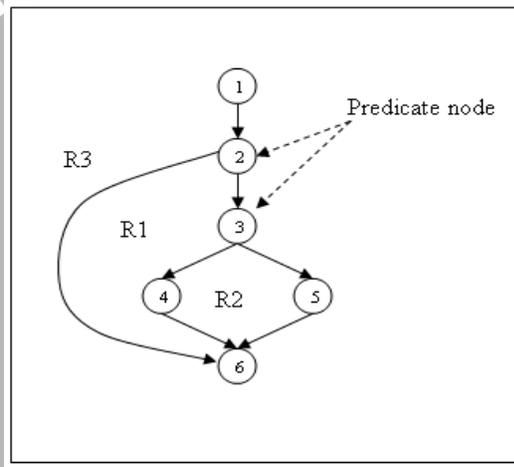
6.1.2 Pengujian Proses Input Data

1. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi

```

PROCEDURE input data
    prosedur untuk memasukkan data kedalam
    database
TYPE tanggal, rekening, isi_data IS STRING
INPUT rekening value and isi_data_value
CONNECT TO database
SELECT tanggal, rekening, isi_data FROM database
SQL INSERT INTO database VALUES (tanggal,
rekening_value and isi_data_value)
    IF (sukses) THEN
        simpan
    ELSE
        PRINT "Data Input Gagal"
ENDIF
ENDPROC
    
```

2. Pembuatan grafik alir yang sesuai



Gambar 6.2 Flow Graph Proses Input Data
Sumber : [Pengujian]

3. Penentuan kompleksitas siklomatis dari grafik alir resultan

$$V(G) = 3 \text{ region}$$

$$V(G) = 7 \text{ edge} - 6 \text{ node} + 2 = 3$$

$$V(G) = 2 \text{ predicate node} + 1 = 3$$

4. Penentuan jalur independen

Jalur 1 : 1 – 2 – 6

Jalur 2 : 1 – 2 – 3 – 4 – 6

Jalur 3 : 1 – 2 – 3 – 5 – 6

5. Melakukan kasus uji

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.2.

Tabel 6.2 Kasus uji untuk pengujian proses input data

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Data value = null	Data yang dimasukkan kosong	Data yang dimasukkan kosong
2	Id_data value != null and data value != null and query sql "gagal"	Data gagal disimpan	Data gagal disimpan
3	Id_data value != null and data value != null and query sql "sukses"	Data berhasil disimpan	Data berhasil disimpan

Sumber : [Pengujian]

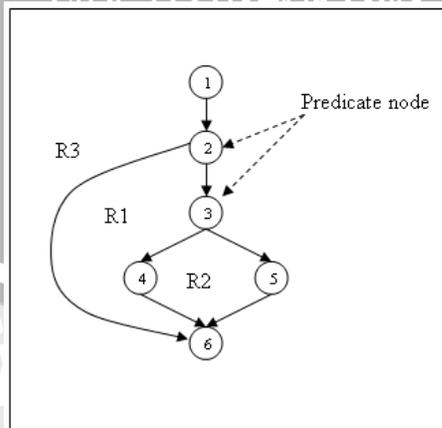
6.1.3 Pengujian Proses Hapus Data

1. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi

```

PROCEDURE hapus data
    prosedur untuk menghapus data dari database
START session
CONNECT TO server
CONNECT TO database
SELECT tabel tagihan FROM database
PRINT data tagihan
SQL DELETE FROM database WHERE id = (GET (id) )
IF (delete sukses) THEN
    PRINT "data telah dihapus";
ELSE
    PRINT "Data gagal dihapus";
ENDIF
ENDPROC
    
```

2. Pembuatan grafik alir yang sesuai



Gambar 6.3 Flow Graph Proses Hapus Data

Sumber : [Pengujian]

3. Penentuan kompleksitas siklomatis dari grafik alir resultan

$$V(G) = 3 \text{ region}$$

$$V(G) = 7 \text{ edge} - 6 \text{ node} + 2 = 3$$

$$V(G) = 2 \text{ predicate node} + 1 = 3$$

4. Penentuan jalur independen

$$\text{Jalur 1 : } 1 - 2 - 6$$

$$\text{Jalur 2 : } 1 - 2 - 3 - 4 - 6$$

$$\text{Jalur 3 : } 1 - 2 - 3 - 5 - 6$$

5. Melakukan kasus uji

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.3.

Table 6.3 Kasus uji untuk pengujian proses hapus data

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Id_data value = null	Obyek list dengan jumlah elemen 0	Obyek list dengan jumlah elemen 0
2	SQL delete "gagal"	Data gagal dihapus	Data gagal dihapus
3	SQL delete "sukses"	Data berhasil dihapus	Data berhasil dihapus

Sumber : [Pengujian]

6.2 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Untuk mendemonstrasikan setiap fungsi dan mengecek apakah terjadi kesalahan, pengujian validasi menggunakan metode pengujian *Black Box*.

6.2.1 Kasus Uji Validasi Sistem WAP Pembayaran Rekening Listrik

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem WAP pada *handphone*, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan kasus uji masing-masing. Melalui WAP browser aplikasi ini dapat diakses dengan alamat (<http://iix.bayusenoadji.com/listrik>).

6.2.1.1 Kasus Uji Login

Kasus uji login ditunjukkan dalam Tabel 6.4a dan Tabel 6.4b.

Tabel 6.4a Kasus uji login dimana *username* dan *password* ada dalam sistem

Nama kasus uji	Kasus uji login
Obyek uji	Kebutuhan login ke sistem
Tujuan pengujian	Menguji bahwa hanya pengguna yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu. Pengguna sistem

	yang tidak login hanya akan menjadi <i>user</i> pengunjung.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memasukkan <i>username</i> dan <i>password</i> pada form login (<i>username</i> = "zama", <i>password</i> = "zama"). 3. Menekan tombol login.
Hasil yang diharapkan	Pengguna berhasil login pada sistem.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.4 Validasi Proses Login

Sumber : [Pengujian]



Gambar 6.5 Validasi Proses Login Berhasil

Sumber : [Pengujian]

Tabel 6.4b Kasus uji login dimana *username* dan *password* tidak ada dalam sistem

Nama kasus uji	Kasus uji login dimana <i>username</i> dan <i>password</i> tidak ada dalam sistem
Obyek uji	Kebutuhan login ke sistem
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa login tidak berhasil.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memasukkan <i>username</i> dan <i>password</i> pada form login (<i>username</i>="zamma", <i>password</i> = "zama"). 3. Menekan tombol login.
Hasil yang diharapkan	Menampilkan pesan login tidak berhasil.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.6 Validasi Proses Login Gagal
Sumber : [Pengujian]

6.2.1.2 Kasus Uji Logout

Kasus uji logout ditunjukkan dalam Tabel 6.5.

Tabel 6.5 Kasus uji logout

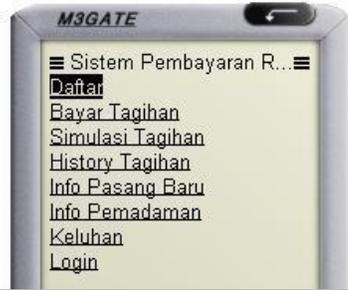
Nama kasus uji	Kasus uji logout
Obyek uji	Kebutuhan logout dari sistem
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas bagi <i>user</i> yang telah login untuk keluar dari sistem
Prosedur uji	1. Sistem telah berjalan dan <i>user</i> telah <i>login</i> 2. <i>User</i> menekan tombol logout
Hasil yang diharapkan	<i>User</i> berhasil keluar dari sistem

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.7 Validasi Proses Logout
Sumber : [Pengujian]



Gambar 6.8 User Kembali Ke Menu Utama
Sumber : [Pengujian]

6.2.1.3 Kasus Uji Daftar Anggota Baru

Kasus uji daftar anggota baru yang dapat dilakukan oleh *user* pengunjung ditunjukkan dalam Tabel 6.6.

Tabel 6.6 Kasus Uji Daftar Anggota Baru

Nama kasus uji	Kasus Daftar Anggota Baru
Obyek uji	Kebutuhan Daftar Anggota
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas bagi <i>user</i> pengunjung yang ingin menjadi anggota.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan 2. <i>User</i> menekan link “Daftar” pada halaman utama. 3. <i>User</i> mengisi data yang diperlukan termasuk <i>username</i> dan <i>password</i>. 4. Setelah data diisi lengkap, <i>user</i> menekan link “option → daftar” 5. Muncul pesan “registrasi sukses”
Hasil yang diharapkan	<i>User</i> pengunjug berhasil mendaftarkan diri menjadi anggota.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.9 Validasi Proses Daftar Anggota
Sumber : [Pengujian]

6.2.1.4 Kasus Uji Pembayaran Tagihan Listrik

Kasus uji pembayaran tagihan listrik dilakukan oleh user yang telah mendaftar sebagai anggota ditunjukkan dalam tabel 6.7.

Tabel 6.7 Kasus uji pembayaran tagihan listrik

Nama kasus uji	Kasus uji Pembayaran Tagihan Listrik oleh anggota.
Obyek uji	Kebutuhan Pembayaran Tagihan Listrik
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan menu pembayaran dan user dapat melakukan pembayaran tagihan listrik.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan <i>user</i> telah login. 2. <i>User</i> menekan link “Bayar Tagihan” pada halaman utama. 3. <i>User</i> menekan link “tanggal tagihan”. 4. <i>User</i> memasukkan nomer rekening. 5. <i>User</i> menekan tombol “option → bayar”. 6. Proses pembayaran berhasil dengan muncul pesan
Hasil yang diharapkan	Sistem menampilkan pesan keberhasilan pembayaran.

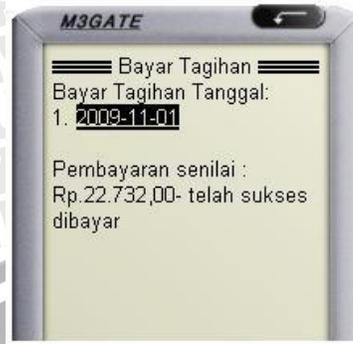
Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.10 Input Nomer Rekening dan Pin

Sumber : [Pengujian]



Gambar 6.11 Proses Pembayaran Berhasil
Sumber : [Pengujian]

6.2.1.5 Kasus Uji Keluhan

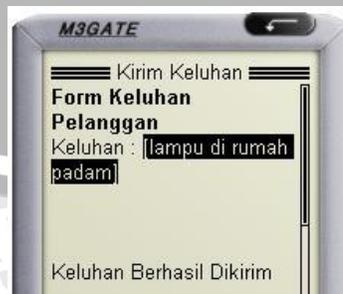
Kasus uji keluhan yang dapat dilakukan oleh *user* yang telah mendaftar sebagai anggota ditunjukkan dalam tabel 6.8

Tabel 6.8 Kasus uji pembatalan keluhan oleh anggota

Nama kasus uji	Kasus uji Keluhan Jasa oleh anggota.
Obyek uji	Kebutuhan Keluhan Jasa
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan menu keluhan jasa dan user dapat melakukan interaksi dengan admin.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan <i>user</i> telah login. 2. <i>User</i> menekan link “keluhan” pada halaman utama. 3. <i>User</i> mengisi keluhan. 4. <i>User</i> menekan link “option → Kirim Keluhan” 5. Keluhan Berhasil
Hasil yang diharapkan	Sistem berhasil menyampaikan keluhan <i>user</i> ke admin, dan menampilkan balasan dari admin.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:



Gambar 6.12 Input Keluhan
Sumber : [Pengujian]



Gambar 6.13 Reply Keluhan Admin
Sumber : [Pengujian]

6.2.2 Kasus Uji Validasi Sistem WEB Pembayaran Rekening Listrik

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem WEB pada komputer, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan kasus uji masing-masing. Melalui WEB browser aplikasi untuk admin dapat diakses dengan alamat (<http://iix.bayusenoadji.com/administrator>).

6.2.2.1 Kasus Uji Login Admin

Kasus uji validasi login admin yang hanya dapat dilakukan oleh *user* admin ditunjukkan dalam tabel 6.9.

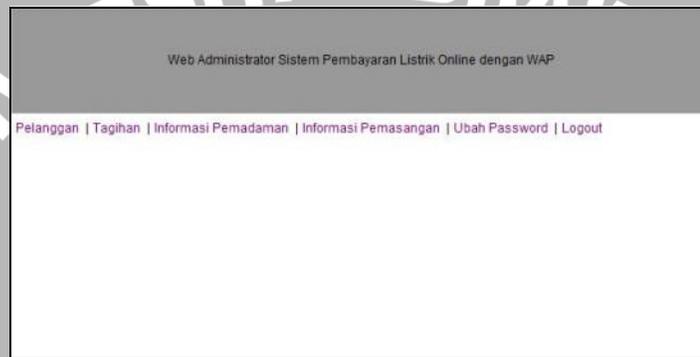
Tabel 6.9 Kasus Uji Validasi Login Admin

Nama kasus uji	Kasus uji Validasi Login Admin
Obyek uji	Kebutuhan login ke sistem sebagai admin.
Tujuan pengujian	Menguji bahwa hanya pengguna yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu. Pengguna sistem yang tidak login sebagai admin maka tidak dapat mengakses fasilitas admin.
Prosedur uji	<ol style="list-style-type: none"> 1. <i>User</i> admin memasukkan <i>username</i> dan <i>password</i> untuk masuk ke dalam sistem. 2. Sistem telah berjalan dan login sebagai admin. 3. Halaman <i>input</i> data baru dan <i>edit</i> data yang telah ada dari sistem pembayaran rekening listrik tampil.
Hasil yang diharapkan	Sistem menampilkan halaman edit dan input data yang hanya dapat diakses jika login sebagai admin.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Gambar 6.14 Validasi Login Admin
Sumber : [Pengujian]



Gambar 6.15 Menu Admin
Sumber : [Pengujian]

6.2.2.2 Kasus Uji Login Manajemen

Kasus uji validasi login manajemen yang dapat dilakukan oleh *user* manajemen ditunjukkan dalam tabel 6.10.

Tabel 6.10 Kasus Uji Validasi Login Manajemen

Nama kasus uji	Kasus uji Validasi Login Manajemen
Obyek uji	Kebutuhan login ke sistem sebagai manajemen .
Tujuan pengujian	Menguji bahwa hanya pengguna yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu. Pengguna sistem yang tidak login sebagai manajemen maka tidak dapat mengakses laporan sistem pembayaran tagihan listrik .
Prosedur uji	<ol style="list-style-type: none"> 1. <i>User</i> manajemen memasukkan <i>username</i> dan <i>password</i> untuk masuk ke dalam sistem. 2. Sistem telah berjalan dan login sebagai manajemen. 3. Halaman lihat laporan baik laporan pembayaran, maupun data anggota.
Hasil yang diharapkan	Sistem menampilkan halaman lihat laporan sistem

pembayaran tagihan listrik yang hanya dapat diakses jika login sebagai manajemen.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Gambar 6.16 Validasi Login Manajemen
Sumber : [Pengujian]

Gambar 6.17 Menu Admin Manajemen
Sumber : [Pengujian]

6.2.2.3 Kasus Uji Login Admin Perangkat Lunak (PL)

Kasus uji validasi login admin perangkat lunak yang dapat dilakukan oleh *user* admin perangkat lunak ditunjukkan dalam tabel 6.11.

Tabel 6.11 Kasus Uji Validasi Login Admin Perangkat Lunak

Nama kasus uji	Kasus uji Validasi Login Admin Perangkat Lunak
Obyek uji	Kebutuhan login ke sistem sebagai admin perangkat lunak.
Tujuan pengujian	Menguji bahwa hanya pengguna yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu. Pengguna sistem yang tidak login sebagai admin perangkat lunak maka tidak dapat mengakses laporan sistem pembayaran tagihan

	listrik.
Prosedur uji	<ol style="list-style-type: none"> 4. <i>User</i> admin PL memasukkan <i>username</i> dan <i>password</i> untuk masuk ke dalam sistem. 5. Sistem telah berjalan dan login sebagai admin PL. 6. Halaman edit <i>username</i> dan <i>password</i> setiap <i>user</i>, baik <i>user</i> admin maupun <i>user</i> manajemen.
Hasil yang diharapkan	Sistem menampilkan halaman edit <i>username</i> dan <i>password</i> setiap <i>user</i> yang hanya dapat diakses jika login sebagai admin PL.

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Gambar 6.18 Validasi Login Admin Perangkat Lunak
Sumber : [Pengujian]

Gambar 6.19 Menu Admin Perangkat Lunak
Sumber : [Pengujian]

6.2.2.4 Kasus Uji Input Data Tagihan Rekening Listrik

Kasus uji input data pembayaran yang dilakukan oleh admin ditunjukkan dalam tabel 6.12.

Tabel 6.12 Kasus Uji Input Data Tagihan

Nama kasus uji	Kasus uji Input Data Tagihan Rekening Listrik
Obyek uji	Kebutuhan input data tagihan rekening listrik
Tujuan pengujian	Menguji fungsi <i>input</i> data oleh admin pada sistem pembayaran rekening listrik.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem berjalan dan <i>login</i> sebagai admin

	<ol style="list-style-type: none"> 2. Tampil halaman <i>input</i> data pembayaran sesuai <i>user</i> anggota. 3. Admin melakukan <i>input</i> pada data <i>tagihan</i>. 4. Menyimpan hasil <i>input</i> data tersebut.
Hasil yang diharapkan	Sistem dapat melakukan proses <i>input</i> data, dalam hal ini data <i>tagihan</i> dan menyimpan perubahan tersebut ke dalam <i>database</i> .

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Gambar 6.20 Input Data Tagihan
Sumber : [Pengujian]

6.2.2.5 Kasus Uji Lihat Laporan Data Tagihan

Kasus uji lihat laporan data tagihan yang dilakukan oleh manajemen ditunjukkan dalam tabel 6.13.

Tabel 6.13 Kasus Uji Lihat Laporan Data Tagihan

Nama kasus uji	Kasus uji Lihat Data Tagihan
Obyek uji	Kebutuhan menampilkan laporan data tagihan <i>user</i> anggota
Tujuan pengujian	Menguji fungsi lihat laporan bagi <i>user</i> manajemen pada sistem pembayaran rekening listrik.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem berjalan dan <i>login</i> sebagai manajemen 2. Tampil halaman lihat laporan 3. <i>User</i> manajemen memilih laporan apa saja yang ingin ditampilkan, dalam hal ini laporan data tagihan. 4. Menampilkan laporan yang diinginkan oleh <i>user</i> manajemen.
Hasil yang diharapkan	Sistem dapat melakukan proses menampilkan laporan sesuai permintaan

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Web Administrator Sistem Pembayaran Listrik Online dengan WAP

Data Pelanggan | Tagihan | Lihat Keluhan | Ubah Password | Logout

Data Tagihan

kode tagihan	no rekening listrik	tanggal tagihan	meter awal	meter akhir	biaya bebas	biaya penggunaan	denda	total biaya	tanggal bayar	tanggal terlambat	status
1	000000000012	2009-06-01	0	100	66440	45400	2000	113840	2009-07-25	2009-06-25	Sudah dibayar
2	000000000012	2009-07-01	100	250	66440	70150	0	136590	2009-07-25	2009-07-25	Sudah dibayar
3	000000000023	2009-06-01	0	200	18000	92600	2000	112600	2009-07-26	2009-06-25	Sudah dibayar
4	000000000023	2009-07-01	200	350	18000	67850	2000	87850	2009-07-26	2009-07-25	Sudah dibayar
5	000000000020	2009-06-01	0	200	39130	94800	2000	135930	2009-07-26	2009-06-25	Sudah dibayar
6	000000000020	2009-07-01	200	350	39130	70050	2000	111180	2009-07-26	2009-07-25	Sudah dibayar
9	000000000023	2009-09-01	0	45	18000	16625	0	34625	2009-09-01	2009-09-03	Sudah dibayar

Gambar 6.21 Laporan Data Tagihan
Sumber : [Pengujian]

6.2.2.6 Kasus Uji *Input Reply* Keluhan Oleh Manajemen

Kasus uji *reply* keluhan yang dilakukan oleh manajemen ditunjukkan dalam tabel 6.14.

Tabel 6.14 Kasus Uji *Input Reply* keluhan

Nama kasus uji	Kasus uji input reply keluhan oleh manajemen
Obyek uji	Kebutuhan berinteraksi dengan <i>user</i> anggota
Tujuan pengujian	Menguji fungsi <i>input reply</i> keluhan <i>user</i> manajemen pada sistem pembayaran rekening listrik.
Prosedur uji	<ol style="list-style-type: none"> 5. Sistem berjalan dan <i>login</i> sebagai manajemen 6. Tampil halaman keluhan 7. <i>User</i> manajemen memilih keluhan yang ingin dibalas 8. Menampilkan keluhan bagi <i>user</i> manajemen.
Hasil yang diharapkan	Sistem dapat melakukan proses <i>reply</i> keluhan anggota

Sumber : [Pengujian]

Penjelasan diatas dapat ditunjukkan pada gambar di bawah ini:

Web Administrator Sistem Pembayaran Listrik Online dengan WAP

Data Pelanggan | Tagihan | Lihat Keluhan | Ubah Password | Logout

Form Reply

Reply akan segera

Data Keluhan

Rakening Listrik	id_pelanggan	Nama	tanggal	keluhan	reply	reply_date	Action
000000000020	000000000020	Aqila Zahra	2009-07-26 10:37:15	Listrik Sering Dipadamkan di daerah Griya Permata Alam	OK, segera kami proses secepatnya	2009-08-05 18:32:48	
000000000012	000000000012	Baju Seno Adj	2009-08-05 18:14:47	test	Segera di proses	2009-08-05 18:32:36	
000000000023	000000000023	Zama	2009-08-12 23:33:52	tes	akan segera dikirim petugas kesana	2009-09-01 01:27:51	
000000000023	000000000023	Zama	2009-08-12 23:35:08	tes	akan segera dikirim petugas kesana	2009-09-01 01:27:51	
000000000023	000000000023	Zama	2009-09-01 01:25:41	lampu di rumah padam	akan segera dikirim petugas kesana	2009-09-01 01:27:51	

Gambar 6.22 Reply Keluhan Oleh Manajemen
Sumber : [Pengujian]

6.2.3 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan pada sistem WAP dan sistem WEB Pembayaran Rekening Listrik sesuai dengan prosedur pengujian pada sub pokok bahasan 6.2.1 dan 6.2.2, didapatkan hasil seperti ditunjukkan pada Tabel 6. 15 dan 6.16.

Tabel 6.15 Hasil Pengujian Validasi Sistem WAP Pembayaran Tagihan Listrik

No	Nama kasus uji	Hasil yang diharapkan	Hasil yang didapat	Status
1	Kasus uji <i>login</i>	Pengguna berhasil login pada sistem.	Pengguna berhasil login pada sistem.	Valid
2	Kasus uji login dimana <i>username</i> dan <i>password</i> tidak ada dalam sistem	Menampilkan pesan login tidak berhasil.	Menampilkan pesan login tidak berhasil.	Valid
3	Kasus uji <i>logout</i>	Pengguna berhasil keluar dari sistem.	Pengguna berhasil keluar dari sistem.	Valid
4	Kasus uji daftar anggota baru	User pengunjung berhasil mendaftarkan diri menjadi anggota.	User pengunjung berhasil mendaftarkan diri menjadi anggota.	Valid
5	Kasus uji Pembayaran Tagihan Listrik	Sistem menampilkan pesan keberhasilan pembayaran.	Sistem menampilkan pesan keberhasilan pembayaran.	Valid
6	Kasus uji Keluhan	Sistem berhasil menyampaikan keluhan <i>user</i> ke admin, dan menampilkan balasan dari admin.	Sistem berhasil menyampaikan keluhan <i>user</i> ke admin, dan menampilkan balasan dari admin.	Valid

Sumber : [Pengujian]

Tabel 6.16 Hasil Pengujian Validasi Sistem WEB Pembayaran Tagihan Listrik

No	Nama kasus uji	Hasil yang diharapkan	Hasil yang didapat	Status
1	Kasus uji admin	Sistem menampilkan halaman edit dan input data yang hanya dapat diakses jika login sebagai admin.	Sistem menampilkan halaman edit dan input data yang hanya dapat diakses jika login sebagai admin.	Valid
2	Kasus uji login manajemen	Sistem menampilkan halaman lihat laporan sistem pembayaran tagihan listrik yang hanya dapat diakses jika login sebagai manajemen.	Sistem menampilkan halaman lihat laporan sistem pembayaran tagihan listrik yang hanya dapat diakses jika login sebagai manajemen.	Valid
3	Kasus uji input data tagihan listrik	Sistem dapat melakukan proses <i>input</i> data, dalam hal ini data tagihan dan menyimpan perubahan tersebut ke dalam <i>database</i>	Sistem dapat melakukan proses <i>input</i> data, dalam hal ini data tagihan dan menyimpan perubahan tersebut ke dalam <i>database</i>	Valid
4	Kasus uji lihat laporan data tagihan	Sistem dapat melakukan proses menampilkan laporan sesuai permintaan.	Sistem dapat melakukan proses menampilkan laporan sesuai permintaan.	Valid
5	Kasus uji reply keluhan oleh <i>user</i> manajemen	Sistem dapat melakukan proses <i>reply</i> keluhan anggota	Sistem dapat melakukan proses <i>reply</i> keluhan anggota	Valid

Sumber : [Pengujian]

Dari 11 kasus uji yang ditunjukkan dalam Tabel 6.20 dan Tabel 6.21 dapat disimpulkan bahwa sistem yang dibangun tidak terdapat kesalahan dan terjamin validitasnya.

6.3 Pengujian Kecepatan Query

a. Tujuan

Pengujian dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan sebuah *query* pada *database* Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS. Pengujian kecepatan *query* ini menggunakan contoh *query* transaksi pembayaran yang dilakukan oleh *user* anggota berdasarkan tabel tagihan.

Pengujian dilakukan untuk mendapatkan perbandingan waktu *query* yang dilakukan terhadap jumlah data yang berbeda pada *database* Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS.

b. Prosedur Pengujian

Mengakses `test_query.php`. Halaman ini dibuat khusus untuk melakukan pengujian waktu akses *query*.

Tabel bayar diberi masukan data sebanyak 100, 300, 500, 700, 900, dan 1100 *data entry* dalam *database* Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS. Melakukan perhitungan untuk mendapatkan nilai waktu akses *query* yang dilakukan terhadap jumlah data yang berbeda dalam *database* Sistem Pembayaran Rekening Listrik.

c. Hasil Pengujian dan Analisis

Hasil pengujian waktu *query* transaksi pembayaran dengan jumlah data `table pembayaran` sebanyak 100, 300, 500, 700, 900, dan 1100 *data entry* ditunjukkan dalam Gambar 6.8.

<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 100 record ← -100 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000485897064209 detik Loop ke : 2 Waktu Akses : 0.000328063964844 detik Loop ke : 3 Waktu Akses : 0.000319004058838 detik Loop ke : 4 Waktu Akses : 0.000289916992188 detik Loop ke : 5 Waktu Akses : 0.00028395652771 detik</p> <p>Total rata-rata waktu akses = 0.000341367721558 detik</p>	<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 300 record ← -300 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000650882720947 detik Loop ke : 2 Waktu Akses : 0.000335931777954 detik Loop ke : 3 Waktu Akses : 0.000308036804199 detik Loop ke : 4 Waktu Akses : 0.000275135040283 detik Loop ke : 5 Waktu Akses : 0.000272035598755 detik</p> <p>Total rata-rata waktu akses = 0.000368404388428 detik</p>
<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 500 record ← -500 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000397920608521 detik Loop ke : 2 Waktu Akses : 0.000336170196533 detik Loop ke : 3 Waktu Akses : 0.000581979751587 detik Loop ke : 4 Waktu Akses : 0.000325202941895 detik Loop ke : 5 Waktu Akses : 0.000304937362671 detik</p> <p>Total rata-rata waktu akses = 0.000389242172241 detik</p>	<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 700 record ← -700 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000381946563721 detik Loop ke : 2 Waktu Akses : 0.000350952148438 detik Loop ke : 3 Waktu Akses : 0.000296115875244 detik Loop ke : 4 Waktu Akses : 0.000284910202026 detik Loop ke : 5 Waktu Akses : 0.000717878341675 detik</p> <p>Total rata-rata waktu akses = 0.000406360626221 detik</p>
<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 900 record ← -900 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000385999679565 detik Loop ke : 2 Waktu Akses : 0.000452041625977 detik Loop ke : 3 Waktu Akses : 0.00032114982605 detik Loop ke : 4 Waktu Akses : 0.000302076339722 detik Loop ke : 5 Waktu Akses : 0.000651836395264 detik</p> <p>Total rata-rata waktu akses = 0.000422620773315 detik</p>	<p>Hasil Query</p> <p>Loop : 5 Jumlah Data : 1100 record ← -1100 data</p> <hr/> <p>Loop ke : 1 Waktu Akses : 0.000499963760376 detik Loop ke : 2 Waktu Akses : 0.000332832336426 detik Loop ke : 3 Waktu Akses : 0.000339031219482 detik Loop ke : 4 Waktu Akses : 0.000741004943848 detik Loop ke : 5 Waktu Akses : 0.000324964523315 detik</p> <p>Total rata-rata waktu akses = 0.000447559256682 detik</p>

Gambar 6.23 Hasil Pengujian Akses Query

Sumber : Pengujian

- Rata-rata pengujian waktu akses query transaksi pembayaran dengan jumlah data Tabel bayar sebanyak 100, 300, 500, 700, 900, dan 1100 data entry ditunjukkan dalam Tabel 6.17.

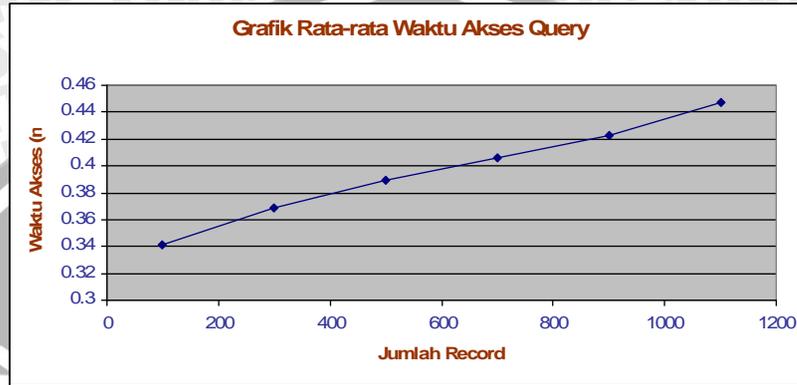
Tabel 6.17 Tabel Pengujian Waktu Akses Query

Pengujian	Jumlah record	Waktu akses (ms)
	Tabel bayar	
1	100	0.341368
2	300	0.368404
3	500	0.389242
4	700	0.406361
5	900	0.422621

6	1100	0.447559
---	------	----------

Sumber : *Pengujian*

- Grafik perbandingan waktu akses *query* pada transaksi pembayaran dengan jumlah data yang berbeda diperlihatkan di dalam Gambar 6. 24.



Gambar 6.24 Grafik Rata-Rata Waktu Akses *Query*
Sumber : *Pengujian*

- Dari pengujian waktu akses *query* dapat diketahui waktu akses per *record* untuk *query* transaksi pembayaran, merupakan hasil bagi antara waktu akses dengan jumlah *record*. Perbandingan waktu akses per *record* dari *query* transaksi pembayaran ditunjukkan dalam Tabel 6.18.

Tabel 6.18 Tabel Perbandingan Waktu Akses Per *Record*

Pengujian	Jumlah record	Waktu akses (ms)	Waktu akses per record (ms)
1	100	0.341368	0.00341368
2	300	0.368404	0.001228013
3	500	0.389242	0.000778484
4	700	0.406361	0.000580516
5	900	0.422621	0.000469579
6	1100	0.447559	0.000406872
Rata-rata waktu akses query per record (ms)			0.001146191

Sumber : *Pengujian*

Berdasarkan hasil pengujian pada Tabel 6. 18 waktu akses *query* untuk *query* transaksi pembayaran dengan jumlah data Tabel bayar sebanyak 100, 300, 500, 700,

900, dan 1100 data *entry*, dapat diketahui bahwa rata-rata waktu akses per *record* sebesar 0,001146191 ms. Dari hasil pengujian dan analisis dapat diketahui bahwa waktu akses rata-rata *query* transaksi pembayaran dengan jumlah data yang ditampilkan semakin banyak, maka waktu akses akan meningkat secara linear.



BAB VII PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapat dari perancangan, implementasi dan pengujian “Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS” antara lain:

1. Aplikasi Sistem Pembayaran Rekening Listrik dapat melakukan koneksi dengan *server* secara *online*.
2. Pada pengujian DFD (*Data Flow Diagram*) menggunakan *Visible Analyst 7.6* dapat disimpulkan bahwa DFD yang dibuat tidak ditemukan kesalahan aliran data dengan ditampilkannya “*diagram correct*” oleh *tool* tersebut.
3. Pada pengujian sistem secara *online* menunjukkan sistem dapat diimplementasikan sesuai analisis kebutuhan yang telah disebutkan pada bab perancangan sebelumnya .
4. Berdasarkan hasil pengujian waktu akses *query* untuk *query* transaksi pembayaran dengan 100, 300, 500, 700, 900, dan 1100 data *entry*, dapat diketahui bahwa rata-rata waktu akses per *record* sebesar 0,001146191 ms.
5. Berdasarkan hasil pengujian dan analisis dapat diketahui bahwa waktu akses rata-rata *query* transaksi pembayaran dengan jumlah data yang ditampilkan semakin banyak, maka waktu akses akan meningkat secara linear.
6. Aplikasi yang telah dihasilkan dapat dimanfaatkan sebagai salah satu alternatif media pembayaran rekening listrik, yang secara umum mempermudah bagi para pelanggan karena sistem tersebut dapat diakses dari mana saja dan kapan saja dengan menggunakan telepon genggam yang mempunyai fasilitas WAP dengan menggunakan jaringan GPRS.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan “Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS”:

1. Sistem Pembayaran Rekening Listrik Melalui Jaringan GPRS dapat dikembangkan lebih luas, tidak hanya dalam lingkup PT.PLN APJ Malang saja.
2. Untuk informasi transaksi pembayaran yang dilakukan oleh anggota dapat dikirimkan melalui sms sebagai tanda bukti pembayaran.
3. Sistem ini dapat dikembangkan lebih lanjut dengan menggunakan teknologi WAP 2.0 yang mempunyai fitur-fitur yang lebih lengkap dan tampilan yang lebih menarik.
4. Menambahkan fasilitas keamanan dalam proses pembayaran pada sistem ini.



DAFTAR PUSTAKA

- [KOM-03] Komputer, Wahana, 2003, "Pengembangan Program WAP", Semarang: Andi Yogyakarta.
- [NUR-03] Nurhadi, Tyasno, 2003, "Pemrograman WML dan WMLS", Yogyakarta: Andi Yogyakarta.
- [MAR-02] Martin, Filda Virmansyah, 2002, "Membuat Halaman Aplikasi WAP", Jakarta: PT Elex Media Komputindo.
- [SUT-04] Sutanta, Edhy, 2004, "Sistem Basis Data", Yogyakarta: Graha Ilmu.
- [NUG-04] Nugroho, Bunafit, 2004, "Database Relasional dengan MySQL", Yogyakarta: Andi Yogyakarta.
- [LIN-04] Lintaka, L, 2004, "Keamanan Jaringan GPRS".
<http://budi.insan.co.id/courses/ec5010/projects/lintaka-report.pdf>
- [ANO-08] www.ilkom.unsri.ac.id/dosen/hartini/materi/I_PengantarSI.pdf
- [AZM-07] azmifauzan.net/wp-content/uploads/2007/10/web-server.pdf
- [ANO-05] www.malangkab.go.id/kabmalang/galeri-ti/TeknologiGPRS.pdf
- [NUG-05] Nugroho, Bunafit, 2005, "Pengembangan Program WAP dengan WML & PHP", Yogyakarta: Gava Media.
- [ASF-06] Asfi, Marsani, 2006, "Interaksi WAP aplikasi dengan MySQL".
- [JUN-06] Juniantoro, Debri D.T, 2006, "Pengaksesan Sistem Informasi Kemahasiswaan Jurusan Teknik Elektro Universitas Brawijaya Melalui Jaringan GPRS dengan Layanan WAP".
- [MAR-08] Mardiansyah, Arif. 2008, "Pembuatan Sistem Informasi PENS ITS Yang Dapat Diakses Melalui WAP".