

**PENGEMBANGAN MEDIA TRANSAKSI MENGGUNAKAN SMS
(SHORT MESSAGE SERVICE) UNTUK BELANJA ONLINE**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh :

ANGGAR RISKINANTO

NIM. 0001063230 - 63

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2007

**PENGEMBANGAN MEDIA TRANSAKSI MENGGUNAKAN SMS
(SHORT MESSAGE SERVICE) UNTUK BELANJA ONLINE**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh :

ANGGAR RISKINANTO

NIM. 0001063230 - 63

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2007

**PENGEMBANGAN MEDIA TRANSAKSI MENGGUNAKAN SMS
(SHORT MESSAGE SERVICE) UNTUK BELANJA ONLINE**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

ANGGAR RISKINANTO

0001063230- 63

DOSEN PEMBIMBING

Ir. Muhammad Aswin
NIP. 131 879 045

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

repository.ub.ac.id

**PENGEMBANGAN MEDIA TRANSAKSI MENGGUNAKAN SMS
(SHORT MESSAGE SERVICE) UNTUK BELANJA ONLINE**

Disusun oleh:

ANGGAR RISKINANTO

0001063230- 63

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 6 Agustus 2007

DOSEN PENGUJI

Suprpto, ST, MT
NIP. 132 149 320

Ir. Sutrisno
NIP. 131 653 479

Ir. Primantara HT
NIP. 132 090 390

Arief Andy Subroto, ST, M.Kom
NIP. 132 231 567

Mengetahui

Ketua Jurusan Teknik Elektro,

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah SWT semata, sholawat serta salam semoga terlimpahkan untuk Nabi Muhammad SAW, beserta para sahabat yang setia mengikuti jejak beliau. Berkat hidayah dan inayah Allah SWT, Skripsi berjudul "PENGEMBANGAN MEDIA TRANSAKSI MENGGUNAKAN SMS (*SHORT MESSAGE SERVICE*) UNTUK BELANJA *ONLINE*" dapat diselesaikan.

Ucapan terima kasih yang mendalam penulis sampaikan kepada pihak-pihak yang terlibat secara langsung maupun tidak langsung, dan telah memberikan dukungan kepada penulis sehingga penyusunan skripsi ini dapat selesai pada waktunya. Atas dukungan dan bimbingannya, pada kesempatan yang berbahagia ini, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Ir. Heru Nurwarsito, MKom, selaku Ketua Jurusan Teknik Elektro.
2. Bapak Rudy Yuwono, ST, Msc selaku Sekretaris Jurusan Teknik Elektro.
3. Bapak Ir. Muhammad Aswin selaku Dosen Pembimbing I dan Bapak Ir. Heru Nurwarsito, MKom selaku Dosen Pembimbing II dan Ketua Kelompok Dosen Keahlian (KKDK) konsentrasi Teknik Informatika dan Komputer yang telah meluangkan waktu, tenaga dan pikiran untuk membimbing penulis dalam penyusunan Skripsi ini.
4. Bapak Ir. Soetrisno, selaku Dosen Wali yang telah membimbing penulis.
5. Seluruh Dosen beserta Staf Administrasi Jurusan Teknik Elektro Universitas Brawijaya.
6. Bapak dan Ibu tercinta yang telah memberikan motivasi, dukungan moral dan material.
7. Adik-adikku tersayang Wiwin dan Ayu.
8. Eko, Farid, Indriawan, dan Yudha sahabat yang telah memberikan doa dan semangat pada penulis.
9. Andi, Rian, Rahmat, Asih, Dian, Lia, Bhibin, Iman, dan Nanang teman penyemangat untuk lulus dengan cepat.
10. Semua Anak Elektro Angkatan 2000 baik reguler maupun ekstensi.
11. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah membantu dan mendorong dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karenanya penulis sangat mengharapkan masukan berupa kritik ataupun saran yang membangun untuk penyempurnaan pada tahap selanjutnya. Semoga skripsi ini memberi manfaat bagi semua pihak sesuai dengan fungsinya.

Malang, 1 Agustus 2007

Penulis



ABSTRAK

Anggar Riskinanto. 2007. : Pengembangan Media Transaksi Menggunakan SMS (*Short Message Service*) Untuk Belanja *Online*. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Pembimbing: Ir. Muhammad Aswin. dan Ir. Heru Nurwarsito, MKom.

Saat ini penggunaan media transaksi pada toko *online* masih terbatas pada media Internet. Hal ini membatasi daerah cakupan pelayanan toko *online* itu sendiri. Disebabkan tidak semua daerah di Indonesia mempunyai akses terhadap layanan Internet. Untuk mengatasi keterbatasan media transaksi, maka dikembangkan media baru menggunakan media SMS selain media Internet yang ada. Diharapkan dengan adanya pengembangan media transaksi ini, daerah cakupan toko *online* dalam menangani transaksi akan bertambah luas.

Perancangan dan implementasi tugas akhir menggunakan osCommerce sebagai aplikasi toko *online*. OsCommerce menggunakan basis data MySQL dalam penyimpanan data. Gnokii digunakan untuk menangani SMS yang diterima maupun akan dikirim. Gnokii menggunakan basis data MySQL untuk menyimpan SMS. Aplikasi antarmuka untuk menjembatani SMS sehingga bisa digunakan sebagai masukan transaksi pada osCommerce menggunakan bahasa pemrograman skrip Perl. Proses pengolahan masukan SMS agar bisa diubah menjadi sebuah *query* basis data menggunakan fasilitas ekspresi reguler yang ada pada Perl.

Hasil pengujian dari pengembangan sistem menghasilkan waktu rata-rata untuk mengirim transaksi hingga menerima konfirmasi transaksi menggunakan SMS sebesar 2464,2 milidetik atau sekitar 24 detik.

Kata kunci: Belanja *Online*, Toko *Online*, SMS, SMS Gateway, Ekspresi Reguler

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	i
KATA PENGANTAR	iii
ABSTRAK	v
DAFTAR ISI	vi
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiv
I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
II LANDASAN TEORI	
2.1 SMS (<i>Short Message Service</i>)	5
2.1.1 Mekanisme Kerja SMS	6
2.1.2 SMS Gateway	8
2.1.3 Perangkat Komunikasi	9
2.1.4 Media Komunikasi	10
2.1.5 Gnokii	13
2.1.6 Tinjauan Pustaka Gnokii.....	14
2.2 Sistem Informasi	15
2.2.1 Komponen Sistem Informasi	15
2.2.2 Arsitektur Sistem Informasi	16
2.2.3 Diagram Aliran Data (<i>Data Flow Diagram</i>)	17
2.3 Basis Data	18
2.3.1 DBMS (<i>Database Management System</i>)	18
2.3.2 Bahasa Basis Data	19

2.3.3	Diagram E-R (<i>Entity-Relationship</i>)	19
2.3.4	MySQL	20
2.4	Internet	21
2.4.1	<i>E-Commerce</i>	22
2.4.2	Keamanan Transaksi dalam E-Commerce.....	22
2.4.3	osCommerce	25
2.4.4	Tinjauan Pustaka osCommerce.....	26
2.5	Perl	27
2.5.1	Tipe Data	28
2.5.2	Operasi Berkas	28
2.5.3	Operasi String	30
2.5.4	Pengenalan dan Pencocokan Pola	33
2.5.5	Koneksi Basis Data dengan Basis Data MySQL	36
2.6	<i>Web Server</i>	38
2.6.1	Apache	39
2.7	Ubuntu.....	40
2.7.1	Tinjauan Pustaka Ubuntu.....	40
III	METODOLOGI	
3.1	Studi Literatur	42
3.2	Perancangan Sistem	42
3.3	Realisasi Sistem	43
3.4	Pengujian dan Analisis Sistem	43
3.5	Pengambilan Kesimpulan dan Saran	43
IV	PERANCANGAN	
4.1	Prinsip Kerja osCommerce	44
4.2	Transaksi Barang pada osCommerce	45
4.3	Basis Data pada osCommerce	46
4.4	Diagram Aliran Data pada Aplikasi Antarmuka	47
4.4.1	Diagram Aliran Data Level 0	47
4.4.2	Diagram Aliran Data Level 1	48
4.4.3	Diagram Aliran Data Level 2	49
4.5	Diagram E-R (<i>Entity-Relationship</i>) Transaksi Barang pada osCommerce	50

4.6	Diagram Alir Transaksi Barang pada osCommerce	51
4.7	Prinsip Kerja Pengembangan Sistem Transaksi	52
4.8	Diagram Blok Pengembangan Sistem Transaksi	53
4.9	Perancangan Isi osCommerce	53
4.9.1	Perancangan Barang	55
4.9.2	Perancangan Jenis Pembayaran	56
4.10	Perancangan SMS <i>Gateway</i>	56
4.10.1	Perancangan Format Masukan SMS <i>Gateway</i>	58
4.11	Perancangan Aplikasi Antarmuka	59
4.11.1	Diagram Alir Aplikasi Antarmuka	60
4.11.2	Perancangan Pengubahan SMS ke <i>Query</i> Basis Data	62
4.11.3	Perancangan Koneksi Basis Data	63
4.11.4	Perancangan Laporan Hasil Transaksi	63
4.12	Perancangan Keamanan Sistem	64
V IMPLEMENTASI		
5.1	Implementasi LAMP (Linux/Apache/MySQL/PHP)	65
5.2	Implementasi Aplikasi WarungLINUX	66
5.2.1	Implementasi Basis Data WarungLINUX	66
5.2.2	Implementasi Administrasi	67
5.2.3	Implementasi Registrasi	75
5.2.4	Implementasi <i>Login</i>	77
5.2.5	Implementasi Pencarian Barang	78
5.2.6	Implementasi Pembelian Barang	80
5.3	Implementasi Gnokii	84
5.3.1	Implementasi Perangkat Keras	85
5.3.2	Implementasi Basis Data	86
5.4	Implementasi Aplikasi Antarmuka	87
5.4.1	Koneksi Basis Data	87
5.4.2	Pencacahan SMS	88
5.4.3	Pemrosesan Kata Kunci pada Media SMS	89
VI PENGUJIAN DAN ANALISIS		
6.1	Pengujian Administrasi WarungLINUX Melalui Media Internet	91



6.1.1	Pengujian Akses Administrasi	91
6.1.2	Pengujian Penambahan Kategori/Barang	94
6.1.3	Pengujian Pengubahan Kategori/Barang	98
6.1.4	Pengujian Penghapusan Kategori/Barang	100
6.2	Pengujian Transaksi WarungLINUX Melalui Media Internet	102
6.2.1	Pengujian Registrasi Pelanggan	102
6.2.2	Pengujian <i>Login</i> Pelanggan	104
6.2.3	Pengujian Pemilihan Barang Oleh Pelanggan	106
6.2.4	Pengujian Pembelian Barang Oleh Pelanggan	108
6.2.5	Pengujian Batasan Nilai Transaksi WarungLINUX.....	110
6.3	Pengujian Perangkat Keras SMS <i>Gateway</i>	112
6.3.1	Pengujian Koneksi Kabel Data	112
6.3.2	Pengujian Koneksi Ponsel Menggunakan Kabel Data	114
6.4	Pengujian Pengiriman dan Penerimaan SMS Melalui SMS <i>Gateway</i>	115
6.5	Pengujian Basis Data SMS <i>Gateway</i>	117
6.6	Pengujian Transaksi WarungLINUX Melalui Media SMS	119
6.7	Pengujian Konfirmasi Hasil Transaksi WarungLINUX Melalui Media SMS ...	122
6.8	Pengujian Kata Kunci pada Media SMS.....	124
6.9	Pengujian Sistem Keseluruhan	125
6.10	Analisis Pengembangan Sistem Transaksi	126
6.10.1	Kelebihan Pengembangan Sistem Transaksi	127
6.10.2	Kekurangan Pengembangan Sistem Transaksi	127
VII	PENUTUP	
7.1	Kesimpulan	128
7.2	Saran	128
VIII	DAFTAR PUSTAKA	DP-1
IX	LAMPIRAN	
	Lampiran A Instalasi dan Konfigurasi Apache	L-1
	Lampiran B Instalasi dan Konfigurasi MySQL	L-2
	Lampiran C Instalasi PHP	L-3
	Lampiran D Instalasi osCommerce	L-4

DAFTAR TABEL

	Halaman
Tabel 2.1 Daftar Ponsel yang Didukung oleh Gnokii	13
Tabel 2.2 Simbol-Simbol yang Digunakan Dalam DFD	17
Tabel 2.3 Penentu format pada sprintf()	32
Tabel 2.4 Metode DBI untuk Mengembalikan Data	38
Tabel 4.1 Daftar Tabel pada Basis Data yang Digunakan oleh osCommerce	46
Tabel 4.2 Daftar Peringkat 10 Besar Distro Menurut www.distrowatch.com pada Mei 2007	54
Tabel 4.3 Struktur Tabel <code>inbox</code> pada Basis Data SMS	58
Tabel 4.4 Struktur Tabel <code>outbox</code> pada Basis Data SMS	58
Tabel 6.1 Hasil Pengujian Pengiriman SMS Dari Ponsel Pengirim ke Komputer <i>Server</i> 116	
Tabel 6.2 Hasil Pengujian Pengiriman SMS Dari Komputer <i>Server</i> ke Ponsel Penerima	116
Tabel 6.3 Hasil Pengujian Waktu Penerimaan dan Pengiriman SMS	118
Tabel 6.4 Tabel Hasil Pengujian Waktu Transaksi	126

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh Penggunaan SMS Pada Ponsel	5
Gambar 2.2 Mekanisme Intra-Operator SMS	7
Gambar 2.3 Mekanisme Inter-Operator SMS	7
Gambar 2.4 Ilustrasi SMS <i>Gateway</i>	9
Gambar 2.5 Komponen Sistem Informasi	16
Gambar 2.6 Notasi Dalam Diagram E-R	20
Gambar 2.7 Tahapan Operasi Pada Berkas	29
Gambar 2.8 Struktur DBI dan ODBC	36
Gambar 4.1 Diagram Aliran Data Level 0	48
Gambar 4.2 Diagram Aliran Data Level 1 – Pemrosesan SMS	48
Gambar 4.3 Diagram Aliran Data Level 1 – Pemrosesan Transaksi	49
Gambar 4.4 Diagram Aliran Data Level 1 – Pemrosesan Data SMS	50
Gambar 4.5 Diagram E-R Pada osCommerce	51
Gambar 4.6 Diagram Alir Transaksi Barang Aplikasi osCommerce	52
Gambar 4.7 Diagram Blok Pengembangan Sistem Transaksi	53
Gambar 4.8 Halaman Depan WarungLINUX	55
Gambar 4.9 Katalog Barang WarungLINUX	55
Gambar 4.10 Diagram Perangkat Keras Pada Gnokii	56
Gambar 4.11 Kabel Data DKU-5	57
Gambar 4.12 Diagram Alir Aplikasi Antarmuka	60
Gambar 4.13 Fungsi Cacah <i>Field</i>	61
Gambar 4.14 Fungsi Konfirmasi Kesalahan	61
Gambar 4.15 Fungsi Laporan Transaksi	62
Gambar 4.16 Diagram Alir Pencacahan SMS	63
Gambar 5.1 Daftar <i>User</i> MySQL	67
Gambar 5.2 Halaman Administration	67
Gambar 5.3 Halaman My Store	68
Gambar 5.4 Halaman Kategori	69
Gambar 5.5 Halaman Produk	70

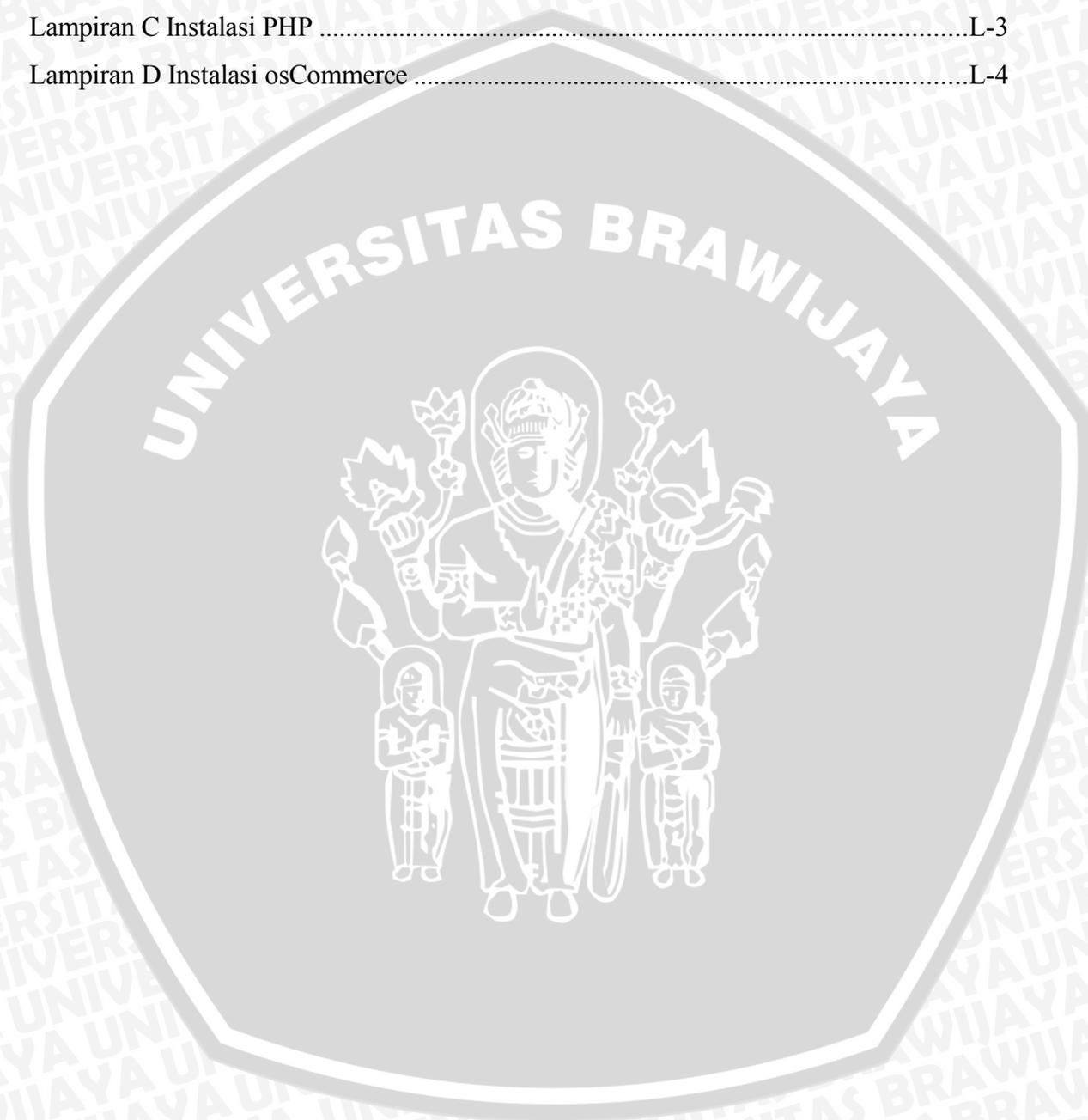
Gambar 5.6	Halaman Payment Modules	71
Gambar 5.7	Halaman Customers	71
Gambar 5.8	Halaman Orders	72
Gambar 5.9	Halaman Countries	72
Gambar 5.10	Halaman New Currencies	73
Gambar 5.11	Halaman Reports	74
Gambar 5.12	Halaman Tools	75
Gambar 5.13	Halaman Registrasi dan <i>Login</i>	76
Gambar 5.14	Halaman Registrasi	77
Gambar 5.15	Proses <i>Login</i> pada Halaman Depan	77
Gambar 5.16	Halaman <i>Login</i>	78
Gambar 5.17	Pencarian Barang melalui Quick Find	80
Gambar 5.18	Halaman Advanced Search	80
Gambar 5.19	Halaman Produk	81
Gambar 5.20	Halaman Cart Content	81
Gambar 5.21	Halaman Delivery Information	82
Gambar 5.22	Halaman Payment Information	83
Gambar 5.23	Halaman Order Information	83
Gambar 5.24	Halaman Order Confirmation	84
Gambar 5.25	Halaman Pemberitahuan Pemrosesan Pesanan	84
Gambar 5.26	Hasil Perintah <code>gnokii -identify</code>	85
Gambar 5.27	Isi Tabel inbox pada Basis Data sms	87
Gambar 5.28	Pola Regex untuk Pencacahan SMS	88
Gambar 6.1	Blok Diagram Lingkungan Pengujian Sistem	90
Gambar 6.2	Halaman Administrasi Aplikasi WarungLINUX	92
Gambar 6.3	Halaman Configuration pada Aplikasi WarungLINUX	92
Gambar 6.4	Halaman Catalog pada Aplikasi WarungLINUX	93
Gambar 6.5	Halaman Modules pada Aplikasi WarungLINUX	93
Gambar 6.6	Halaman Customers pada Aplikasi WarungLINUX	94
Gambar 6.7	Halaman <i>Preview</i> Barang	96
Gambar 6.8	Tabel categories dan categories_description Sebelum dan Sesudah Diisi Data	96



Gambar 6.9 Tabel products Sebelum dan Sesudah Diisi Data	96
Gambar 6.10 Halaman Barang yang Diubah	99
Gambar 6.11 Tabel products Sebelum dan Sesudah Diubah Datanya	99
Gambar 6.12 Halaman Barang yang Akan Dihapus	101
Gambar 6.13 Tabel products Sebelum dan Sesudah Dihapus Datanya	101
Gambar 6.14 Halaman My Account Information	103
Gambar 6.15 Tabel customers Sebelum dan Sesudah Diisi Data	103
Gambar 6.16 Halaman Utama Sebelum Pelanggan <i>Login</i>	105
Gambar 6.17 Halaman Utama Setelah Pelanggan <i>Login</i>	105
Gambar 6.18 Halaman Cart Contents	107
Gambar 6.19 Tabel customers_basket Sebelum dan Sesudah Diisi Data	107
Gambar 6.20 Tabel orders_products dan orders Sebelum dan Sesudah Diisi Data	109
Gambar 6.21 Halaman Pemesanan Barang dengan Jumlah Barang Sebanyak 1000 Buah	111
Gambar 6.22 Peringatan Ketika Jumlah Barang yang Dipesan Melebihi Stok	111
Gambar 6.23 Hasil Perintah <code>dmescg grep usb</code>	113
Gambar 6.24 Hasil Perintah <code>lsusb</code>	113
Gambar 6.25 Hasil perintah <code>gnokii --identify</code>	114
Gambar 6.26 Tabel inbox Sebelum dan Sesudah Menerima SMS	118
Gambar 6.27 Tabel outbox Sebelum dan Sesudah Mengirim SMS	118
Gambar 6.28 Halaman Cart Contents Sebelum SMS Diterima	120
Gambar 6.29 Halaman Cart Contents Setelah SMS Diterima	121
Gambar 6.30 Tabel inbox Sebelum dan Sesudah Menerima SMS	121
Gambar 6.31 Tabel customers_basket Sebelum dan Sesudah SMS Diterima	121
Gambar 6.32 Tabel outbox Sebelum dan Sesudah Mengirim SMS Konfirmasi	123
Gambar 6.33 Program Enkripsi Password Pada Media SMS	124
Gambar 6.34 Isi Tabel customers pada Basis Data warunglinux	124

DAFTAR LAMPIRAN

	Halaman
Lampiran A Instalasi dan Konfigurasi Apache	L-1
Lampiran B Instalasi dan Konfigurasi MySQL	L-2
Lampiran C Instalasi PHP	L-3
Lampiran D Instalasi osCommerce	L-4



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era teknologi informasi yang berkembang pesat pada saat ini, semua pihak dituntut untuk mendapatkan dan menyediakan informasi secara cepat. Untuk itu Internet sebagai salah satu mediana sangat berguna untuk melakukan hal itu. Perdagangan merupakan suatu bidang yang paling merasakan keuntungan dari penggunaan media Internet. Dengan Internet, tempat yang tadinya memaksa antara penjual dengan pembeli untuk bertemu dapat dihilangkan.

Seiring dengan semakin meluasnya penggunaan Internet, maka perlahan namun pasti penjual mulai menggunakan Internet sebagai media dalam melakukan kegiatan transaksi barang. Salah satu solusi dalam penggunaan aplikasi toko *online* adalah osCommerce. OsCommerce merupakan aplikasi administrasi untuk transaksi yang biasanya terdapat pada sebuah toko. Terdapat kelemahan pada aplikasi ini, yaitu aksesnya yang hanya menggunakan media Internet. Menurut Sean Michael Kerner [KER-04:1], *“osCommerce akan bekerja pada semua mesin yang dapat menjalankan PHP (4.x+) dan mempunyai akses ke sebuah basis data MySQL (3.x+). Menurut pengalaman saya dengan osCommerce, maksud sebenarnya adalah sebuah setup LAMP (Linux/Apache/MySQL/PHP) yang tipikal”*. Dengan kata lain aplikasi osCommerce ditujukan untuk perdagangan berbasis *web*. Hal ini berpengaruh pada perdagangan yang akan mengurangi jumlah kegiatannya.

Untuk itu diperlukan suatu pengembangan sistem informasi yang telah ada. Pengembangan tersebut bertujuan untuk memperluas media akses yang awalnya hanya menggunakan Internet ditambah menggunakan media SMS. Aplikasi ini diharapkan dapat mengatasi permasalahan mengenai akses media yang terbatas. Sehingga siapapun dapat melakukan transaksi dimanapun ia berada.

Pada sistem osCommerce standar, media transaksi hanya bisa dilakukan melalui Internet. Dengan dikembangkannya sistem, maka media SMS juga dapat dimanfaatkan untuk media transaksi. Pengambilan SMS menggunakan SMS Gateway Gnokii. Kemudian digunakan suatu program antarmuka yang dikembangkan menggunakan bahasa pemrograman Perl. Perl digunakan untuk mengubah masukan yang berupa SMS menjadi sebuah *query* basis data. Sehingga dapat dimasukkan ke dalam basis data milik

osCommerce. Diharapkan tidak ada bedanya hasil transaksi yang menggunakan media Internet maupun SMS.

1.2 Rumusan Masalah

Pada saat ini, sebagian besar toko-toko *online* yang ada telah mengimplementasi sistem informasi pada produk-produk mereka. Namun itu semua hanya terbatas pada media Internet. Sehingga dengan adanya pengembangan sistem informasi ini, maka tidak akan terbatas pada media Internet tapi juga diperluas menggunakan media SMS. Untuk itu rumusan masalah ditekankan pada :

1. Media transaksi barang tidak hanya terbatas pada media Internet, tapi juga diperluas dengan menggunakan media SMS.
2. Mengkonfigurasi telepon selular agar dapat berfungsi sebagai SMS *modem*.
3. Pesan SMS yang diterima dapat diubah menjadi *query* basis data.
4. Konsumen dapat melihat daftar barang yang telah ia pesan dan kemudian dapat menambah atau mengurangi belanjaan yang telah ia lakukan melalui media SMS.
5. Konsumen mendapatkan laporan apabila transaksi yang dilakukannya berhasil atau tidak melalui media SMS.

1.3 Batasan Masalah

Dalam penyusunan tugas akhir ini akan dibataskan pada:

1. Sistem informasi ini menyediakan informasi mengenai produk-produk toko *online* tersebut.
2. Sistem informasi membahas mengenai pemesanan barang dan konsumen dapat melihat informasi pemesanan. Kemudian ia dapat mengubahnya. Dan akhirnya menerima laporan transaksi.
3. Sistem informasi bersifat *semi client server*. Dimana terdapat implementasi untuk SMS Gateway pada dunia nyata. Sedangkan simulasi untuk aplikasi antarmuka dan basis data.
4. Sistem informasi menggunakan sistem operasi GNU/Linux.
5. Sistem informasi menggunakan basis data MySQL.
6. Sistem informasi menggunakan bahasa pemrograman skrip Perl untuk aplikasi antarmuka antara SMS Gateway Gnokii dengan aplikasi administrasi transaksi *e-commerce* osCommerce.

7. Sistem informasi menggunakan osCommerce sebagai administrasi transaksi *e-commerce*.
8. Sistem informasi menggunakan SMS Gateway Gnokii sebagai SMS *engine* yang mengatur komunikasi menggunakan SMS.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah mengembangkan transaksi yang ada pada toko *online* dengan menambahkan media SMS disamping media Internet untuk memudahkan pemesanan barang yang dilakukan oleh konsumen.

1.5 Manfaat

Manfaat yang didapat dari tugas akhir ini adalah :

1. Memperluas area cakupan pemesanan barang dengan diterapkannya aplikasi.
2. Meminimalisasi keterlibatan penjual dalam menangani kegiatan transaksi barang pada tokonya.
3. Mengatasi keterbatasan media Internet sebagai media pemesanan barang.

1.6 Sistematika Pembahasan

Dalam penyusunan tugas akhir ini, maka untuk mempermudah pembahasan masalah, maka sistematika pembahasan yang ada adalah :

BABI : PENDAHULUAN

Dalam bab ini membahas mengenai tinjauan umum, latar belakang masalah, pokok masalah, batasan masalah, maksud dan tujuan, metode pengumpulan data dan sistematika penulisan.

BAB II : DASAR TEORI

Dalam bab ini berisikan tentang tinjauan umum sistem informasi, *e-commerce*, sistem transaksi barang di toko *online*, dan sistem SMS Gateway.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini membahas mengenai metodologi dalam pembuatan aplikasi.

BAB IV : PERANCANGAN

Dalam bab ini dijelaskan tentang definisi sistem, sistem perangkat lunak, sistem perangkat keras, diagram alir sistem, hubungan antar tabel, perancangan masukan, perancangan keluaran dan rancangan tabel.

BAB V : IMPLEMENTASI

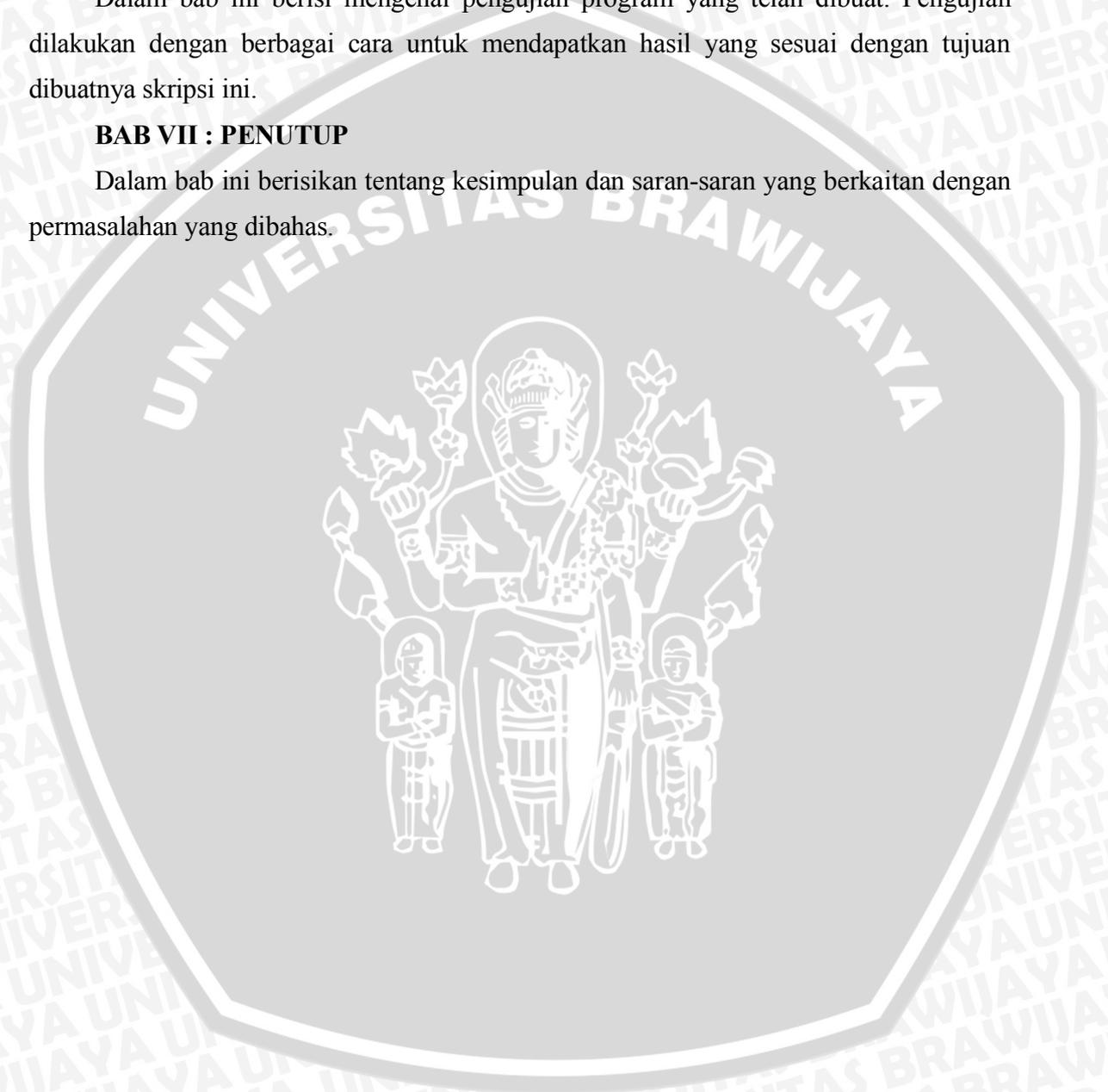
Dalam bab ini membahas mengenai spesifikasi program, algoritma pembuatan program antar muka berbasis SMS.

BAB VI : PENGUJIAN DAN ANALISIS

Dalam bab ini berisi mengenai pengujian program yang telah dibuat. Pengujian dilakukan dengan berbagai cara untuk mendapatkan hasil yang sesuai dengan tujuan dibuatnya skripsi ini.

BAB VII : PENUTUP

Dalam bab ini berisikan tentang kesimpulan dan saran-saran yang berkaitan dengan permasalahan yang dibahas.



BAB II

DASAR TEORI

2.1 SMS (*Short Message Service*)

SMS (*Short Message Service*) merupakan sebuah layanan pada ponsel yang memungkinkan pengiriman dan penerimaan pesan pendek (lebih dikenal sebagai pesan teks) diantara ponsel [ANI-06:1]. Kegunaan SMS yang lain adalah untuk pemesanan *ringtone*, *wallpaper*, dan mengikuti kompetisi.

SMS sendiri adalah sebuah teknologi. Teknologi yang memungkinkan untuk menerima maupun mengirim pesan antar ponsel. Teknologi baru ini pertama kali diperkenalkan pada tahun 1992 di Eropa oleh ETSI (*European Telecommunication Standards Institute*), dan pada awalnya menjadi suatu standar untuk telepon nirkabel yang berbasis GSM (*Global System for Mobile Communications*). Namun, teknologi lain seperti CDMA dan TDMA pun memasukkan SMS ini sebagai fitur standar mereka.

Sebagaimana namanya, SMS berarti layanan pesan pendek, maka besar data yang dapat ditampung oleh SMS ini sangat terbatas. Untuk satu SMS yang dikirimkan, hanya dapat menampung paling banyak 140 *byte* atau 1120 bit. Bila diubah ke dalam bentuk karakter, maka untuk satu SMS hanya dapat berisi paling banyak 160 karakter untuk karakter latin, dan 70 karakter untuk karakter non-latin seperti Cina maupun Jepang.



Gambar 2.1. Contoh Penggunaan SMS Pada Ponsel.
Sumber: [ANI-06:1]

Lalu bagaimana dengan telepon yang mampu mengirim SMS lebih dari 160 karakter dengan sekali kirim? Ponsel yang bisa mengirim SMS sebanyak itu, pada dasarnya bukan berarti SMS memiliki batasan menjadi lebih dari 160 karakter. Namun, ketika ponsel mengirimkan SMS yang memiliki karakter lebih dari 160 karakter, ponsel akan memecah SMS itu menjadi SMS-SMS kecil sebesar 160 karakter, kemudian ponsel penerima akan menggabung SMS-SMS tersebut menjadi SMS utuh. SMS ini sering disebut sebagai *concatenated SMS*.

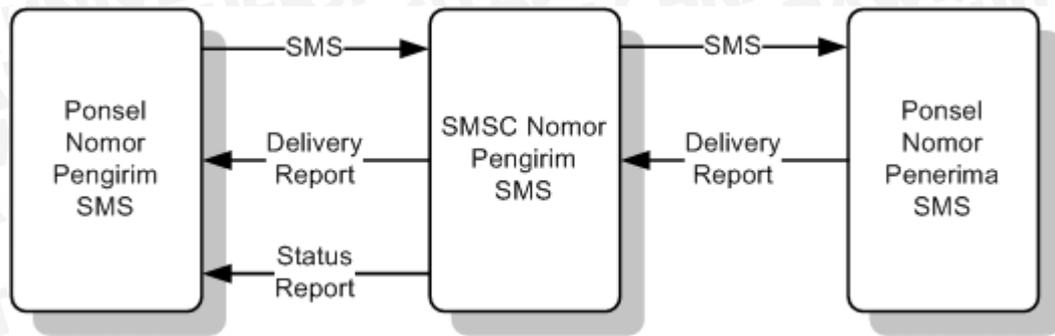
2.1.1 Mekanisme Kerja SMS

Ketika SMS dikirimkan ke suatu nomor tertentu, SMS yang dikirimkan tidak akan langsung dikirimkan ke nomor tersebut, namun akan masuk terlebih dahulu ke SMS Center (SMSC) operator telepon yang digunakan. SMS Center sendiri dapat diartikan sebagai sebuah *server* yang bertanggung jawab pada proses pengiriman SMS dalam suatu operator. SMS yang dikirimkan dari suatu ponsel akan masuk SMSC ini, kemudian baru diteruskan ke nomor tujuan SMS tersebut. Bila nomor yang dituju ternyata sedang tidak aktif/*offline*, SMSC ini akan menyimpan SMS tersebut untuk sementara waktu, hingga nomor tujuan aktif kembali. Lamanya waktu penyimpanan SMS, sangat tergantung dari lamanya waktu yang telah ditetapkan oleh operator untuk menyimpan SMS tersebut. Nomor yang telah menerima SMS akan mengirimkan laporan SMSC bahwa SMS telah diterima. Laporan tersebut kemudian akan diteruskan kembali ke nomor pengirim SMS.

Secara garis besar, mekanisme kerja pengiriman SMS dapat dibagi menjadi tiga macam [YUN-06:73-74], yaitu:

1. Pengiriman SMS dalam satu operator atau sering diistilahkan dengan *Intra-Operator SMS*.

Gambaran mekanisme pengiriman SMS ini dapat dilihat pada gambar berikut:

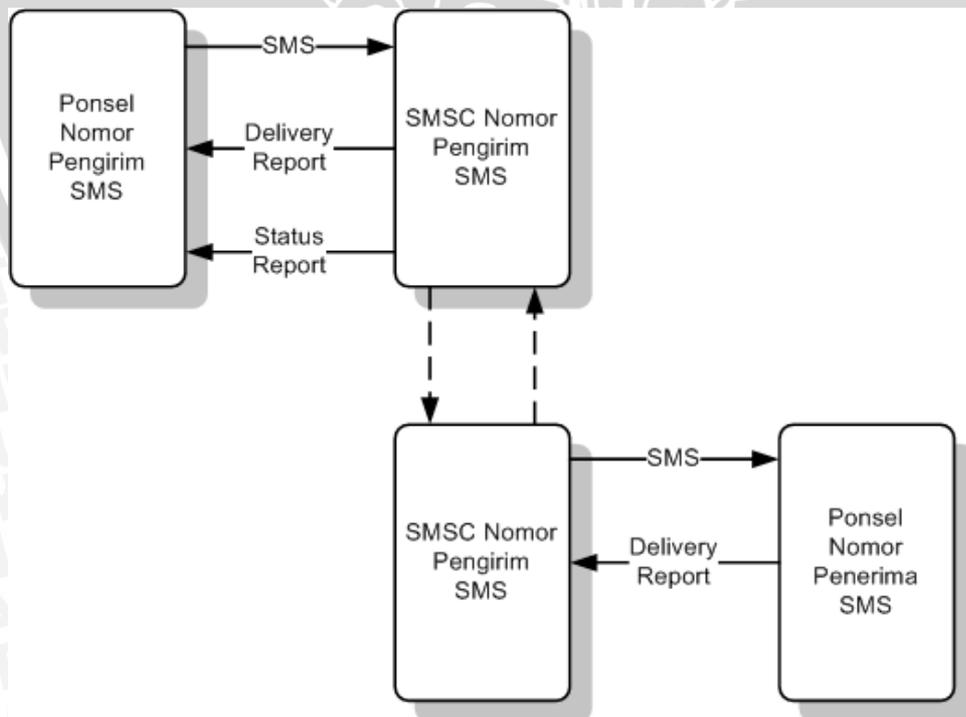


Gambar 2.2. Mekanisme Intra-Operator SMS.
 Sumber: [YUN-06:73]

Dalam Gambar 2.2, SMS dikirimkan oleh nomor pengirim akan dimasukkan terlebih dahulu ke dalam SMSC operator nomor pengirim, kemudian SMSC tersebut akan mengirimkan sebuah *delivery report* yang menyatakan bahwa SMS telah diterima ke SMSC. SMSC kemudian meneruskan *report* tersebut ke nomor pengirim SMS, disertai *status report* dari pengiriman SMS tersebut.

2. Pengiriman SMS antar operator yang berbeda atau *Inter-Operator SMS*.

Berbeda dengan mekanisme *intra-operator* di atas. Pada mekanisme ini, SMS yang dikirimkan akan melalui dua buah SMSC. Perbedaan tersebut ditunjukkan oleh gambar berikut:



Gambar 2.3. Mekanisme Inter-Operator SMS.
 Sumber: [YUN-06:74]

Dalam Gambar 2.3, setelah masuk ke SMSC operator pengirim, SMS yang dikirimkan akan diteruskan oleh SMSC operator pengirim, ke SMSC operator penerima SMS, kemudian baru diteruskan ke nomor tujuan. *Delivery report* yang dihasilkan pun akan melalui jalur tersebut, agar dapat sampai ke nomor pengirim SMS. Dalam mekanisme ini, terlihat adanya komunikasi tidak langsung antara dua operator berbeda. Komunikasi tersebut dapat berjalan, setelah terjadi sebuah kesepakatan kerja sama antaroperator tersebut. Tidak adanya sebuah kesepakatan kerjasama antaroperator, dapat menyebabkan SMS tidak sampai pada nomor tujuan tersebut.

3. Pengiriman SMS dari operator suatu negara ke negara lain.

Proses pengiriman SMS pada mekanisme ini sering disebut dengan istilah SMS Internasional. Mekanisme yang terjadi tidak jauh berbeda dengan mekanisme dengan mekanisme pada *inter-operator SMS*. Perbedaannya hanya ada pada SMSC nomor penerima, yang merupakan SMSC operator luar negeri, dan penambahan kode negara pada nomor tujuan.

2.1.2 SMS Gateway

Istilah *gateway*, bila dilihat pada kamus bahasa Inggris-Indonesia diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat diartikan juga sebagai jembatan penghubung antar satu sistem dengan sistem lain yang berbeda, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian, SMS *gateway* dapat diartikan sebagai suatu penghubung untuk lalu lintas data-data SMS, baik yang dikirimkan, maupun yang diterima [YUN-06:75].

Pada dasarnya, sebuah SMS Gateway menggunakan salah satu atau lebih *modem-modem* GSM atau sebuah jaringan koneksi langsung (HTTP(S), SMPP, CIMS, dan lain sebagainya) dengan pusat SMS untuk mengirim dan menerima pesan-pesan ke dan dari jaringan bergerak.

Aplikasi perangkat lunak secara umum berantar muka dengan SMS Gateway menggunakan *file* teks ASCII, XML/SOAP atau tabel basis data. Sebuah SMS Gateway yang baik akan menyediakan pada pengembang dengan komponen ActiveX atau sebuah API untuk berinteraksi dengan *gateway* untuk membaca atau menulis pesan-pesan SMS, sehingga mengurangi waktu pengembangan untuk mengimplementasikan fitur pesan bergerak dalam aplikasi perangkat lunak.



Gambar 2.4. Ilustrasi SMS Gateway.
Sumber: [YUN-06:75]

Namun seiring perkembangan teknologi komputer, baik dari segi *hardware* maupun *software*, dan perkembangan teknologi komunikasi, SMS gateway tidak lagi dimaksudkan seperti dalam ilustrasi di atas. Dewasa ini, masyarakat lebih mengartikan SMS gateway sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel) dengan perangkat komputer mereka, yang menjadikan aktifitas SMS menjadi lebih mudah dan menyenangkan. Pengertian SMS gateway lebih mengarah pada sebuah program yang mengomunikasikan antara sistem operasi komputer, dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS [YUN-06:76]. Salah satu komunikasi yang terjadi, dapat dilakukan dengan mengirimkan perintah AT pada perangkat komunikasi tersebut, kemudian hasil operasinya dikirimkan kembali ke komputer. Dibutuhkan suatu antarmuka baik dalam bentuk aplikasi maupun halaman *web* untuk membaca SMS yang masuk, atau mengirim SMS tersebut.

2.1.3 Perangkat Komunikasi

Perangkat komunikasi yang dimaksud adalah perangkat yang dapat digunakan untuk mengirim atau menerima SMS. Perangkat-perangkat tersebut dapat berupa:

1. Telepon seluler (ponsel).

Apapun merk dan tipe ponsel yang digunakan, bisa dipastikan memiliki fitur untuk SMS, baik yang menggunakan teknologi GSM maupun CDMA. Namun bukan berarti ponsel-ponsel tersebut dapat digunakan sebagai piranti SMS gateway. Agar dapat terhubung ke komputer, ponsel harus memiliki dukungan konektifitas ke perangkat lain, baik dengan kabel, *bluetooth* maupun infra merah.

2. GSM *modem*.

GSM *modem* adalah sebuah *modem* nirkabel yang bekerja dengan jaringan GSM. Fungsi dari GSM *modem* ini hampir sama dengan *modem* biasa, bedanya terletak pada media yang digunakan untuk transfer data. Bila *modem* biasa menggunakan kabel telepon untuk transfer data, GSM *modem* menggunakan gelombang radio sebagai medianya.

GSM *modem* yang digunakan dapat berupa PC Card/PCMCIA Card, maupun berupa eksternal *device* yang menggunakan kabel serial atau USB untuk koneksi ke komputer.

Sebagaimana namanya, GSM *modem* memerlukan sebuah SIM *card* GSM untuk mengoperasikannya. GSM *modem* ini dapat digunakan untuk operasi standar SMS (baca, kirim, hapus), memonitor kekuatan sinyal, operasi-operasi *phonebook*, dan melihat status *charging battery* menggunakan perintah-perintah AT.

Kelebihan penggunaan GSM *modem* adalah tidak perlu takut terhadap keadaan *low battery* seperti yang dialami pada ponsel. Karena pada GSM *modem* menggunakan daya listrik langsung dari listrik jaringan rumah (terdapat *adapter* AC ke DC pada rangkaiannya).

3. GPRS *modem*.

GPRS *modem* memiliki fungsi yang mirip dengan GSM *modem*. Perbedaan paling mendasar pada GPRS *modem* adalah adanya tambahan dukungan untuk teknologi GPRS untuk transmisi datanya. Kecepatan proses SMS pada GPRS *modem* lebih cepat dibandingkan GSM *modem*, bila proses transfer SMS menggunakan GPRS. Selain itu, GPRS *modem* sangat bermanfaat untuk mengirim atau menerima MMS.

2.1.4 Media Komunikasi

Media komunikasi yang dimaksud adalah media yang dapat digunakan untuk mengirim data dari perangkat komunikasi ke komputer atau sebaliknya. Pada ponsel, istilah yang sering digunakan adalah *connectivity* atau konektifitas. Media-media tersebut diantaranya adalah:

1. Kabel Data.

Ada banyak jenis kabel data yang dapat digunakan untuk menghubungkan sebuah

ponsel ke komputer, hal ini tergantung merk dan tipe ponsel yang akan digunakan. Karena masing-masing merk dan tipe ponsel memiliki konektor kabel yang berbeda satu sama lain, sehingga diperlukan kejelian dalam memilih kabel yang akan digunakan.

Kabel-kabel data tersebut, umumnya dapat digunakan untuk:

- Mengirim atau menerima SMS.
- *Upload ringtone* dan gambar.
- Mengelola data *phonebook*.

Sedangkan fungsi-fungsi tambahan yang tidak dimiliki oleh setiap jenis kabel, misalnya untuk:

- Koneksi ke Internet.
- Mengirim dan menerima faksimili dan *e-mail*.
- Mengakses *Network Monitor*.
- *Unlocker* ponsel.

2. *Bluetooth*.

Bluetooth adalah sebuah standar spesifikasi yang mendefinisikan bagaimana antar perangkat (telepon, komputer, PDA, atau perangkat digital lain) dapat saling berkomunikasi, dan berhubungan tanpa melalui kabel (nirkabel).

Gagasan ini pertama kali dikeluarkan oleh Ericsson pada tahun 1994 yang melihat tidak efisiennya komunikasi antara *handphone* dengan PDA (*Personal Digital Assistant*). Karena gagasan tersebut dianggap baik, maka Ericsson, IBM, Intel, Nokia, dan Toshiba membentuk *Bluetooth Special Interest Group* (BSIG) pada Februari 1998. Tiga bulan kemudian, yaitu pada bulan Mei, *Bluetooth* diperkenalkan pada publik. Setelah peluncuran perdana standar *Bluetooth* pada Juli 1999, makin banyak perusahaan yang bergabung menjadi anggota SIG, di antaranya adalah Motorola, Microsoft, Lucent, dan 3Com.

Nama *Bluetooth* sendiri diambil dari nama panggilan seorang raja Viking, Harald Bland II (941-980). Bland adalah raja yang telah berhasil menyatukan Denmark dibawah bendera Kristiani, dengan Norwegia yang berhasil ditaklukkannya. Salah satu alasan penggunaan nama *Bluetooth* ini adalah karena keberhasilan standar dalam mempersatukan antara PC dengan peralatan telekomunikasi.

Media yang dipakai dalam transmisi data adalah gelombang radio yang berjalan pada frekuensi ISM 2,4 Ghz. ISM adalah singkatan dari *Industrial Scientific and Medical*, frekuensi bebas yang memang diperuntukkan untuk keperluan industri, sains, dan medis. Jarak maksimal yang mungkin untuk komunikasi antar perangkat ini adalah 100 meter. Agar sebuah PC dapat berkomunikasi dengan perangkat lain yang memiliki *Bluetooth*, diperlukan sebuah *bluetooth adapter* atau sering disebut juga sebagai *bluetooth dongle*. Perangkat ini biasanya adalah perangkat eksternal sebuah PC, sehingga mudah dilepas dan dipasang kembali, dengan konektor bertipe USB.

3. IrDA.

Sesuai dengan namanya, IrDA (Infrared Data Association) adalah sebuah organisasi nonprofit yang dibentuk untuk membuat sebuah standar teknologi transmisi data tanpa kabel, dengan sinar infra merah sebagai media transmisinya. Untuk selanjutnya, penggunaan istilah IrDA sendiri dipelopori oleh HP, IBM, dan Sharp pada Juni 1993, yang bertujuan mempromosikan sebuah standar teknologi industri untuk berkomunikasi. Beberapa bulan kemudian, sebanyak 50 perusahaan menyetujui sebuah *standard methode* komunikasi data menggunakan transmisi infra merah. Sejak itu, anggota IrDA berkembang menjadi 130 perusahaan, dan bertambah terus hingga saat ini.

Prinsip kerja dari perangkat inframerah ini adalah sebuah *transceiver* IrDA akan mengirimkan pulsa infra merah dalam bentuk kerucut dengan sudut antara 15 hingga 30 derajat dari titik tengah. Secara umum, jarak maksimal antara perangkat IrDA agar dapat berkomunikasi adalah sepanjang 1 meter tanpa penghalang. Beberapa perangkat yang dijual di pasaran, ada yang hanya mampu bekerja kurang dari 1 meter, tetapi ada juga yang dapat bekerja pada jarak beberapa meter. Sedangkan gelombang inframerah yang digunakan oleh IrDA adalah dengan panjang gelombang antara 0,85 hingga 0,90 meter.

Ada tiga macam kategori kecepatan transmisi yang dimiliki oleh IrDA, yaitu:

- SIR (*Serial Infra Red*), kecepatan yang didukung oleh *port* rs-232 adalah 9600 bps, 19,2 Kbps, 38,4 Kbps, 57,6 Kbps, dan 115,2 Kbps. Hampir semua perangkat menggunakan kecepatan 9600 bps sebagai kecepatan transmisinya (*baud rate*).

- MIR (Medium Infra Red), memiliki kecepatan transmisi berkisar antara 576 Mbps dan 1,152 Mbps.
- FIR (Fast Infra Red), memiliki kecepatan transmisi berkisar 4 Mbps.

2.1.5 Gnokii

Gnokii adalah nama sebuah proyek yang ditujukan untuk membangun sebuah *tool* dan *driver* yang dapat menangani ponsel di dalam Linux. Pada awalnya, Gnokii hanya ditujukan pada ponsel Nokia. Namun pada perkembangannya, Gnokii juga mulai mendukung ponsel selain Nokia. *Tool* yang dimiliki Gnokii mencakup *driver modem*, *tool* untuk konsol, aplikasi yang berbasis grafis dan banyak lagi. Dengan lisensi GNU/GPL, menjamin kebebasan menggunakan *tool* ini tanpa perlu takut masalah legalitas dan biaya mahal yang harus dikeluarkan, agar dapat menggunakan atau memanfaatkan SMS Gateway.

Sampai saat ini, Gnokii mendukung ponsel dan tipe dari merk berikut:

Tabel 2.1. Daftar Ponsel yang Didukung oleh Gnokii.

Merk	Tipe
Nokia Seri 3810	3110, 3810, 8110, 8100i
Nokia Seri 6110	3210, 3310, 3330, 3360, 3410, 5110, 5190, 6110, 6130, 6150, 6190, 8210, 8250, 8290
Nokia Seri 7110	6210, 6250, 7110, 7190
Nokia Seri 6510	5100, 6100, 6310, 6310i, 6360, 6510, 6610, 8310, 6800, 6170
Nokia Seri 640 dan 2110	
Nokia Seri 6160	5120, 6160
Seri AT1	Bosch 908/909 Motorola Timeport P7389i (L series), C350, Nokia 6210/ 7110/8210/ 6310(i)/6510/9210/7650/3650, Siemens S25/SL45i/C55/M55/S55, Sony Ericsson T68i

Sumber: [YUN-06:98-99]

Model lain yang dapat bekerja dengan AT *command*, seperti GSM *modem* maupun GPRS *modem*. Untuk ponsel dengan teknologi CDMA, kemungkinan juga dapat digunakan.

Kemampuan yang dapat ditangani oleh Gnokii, di antaranya adalah:

1. Membaca, menghapus, dan mengirim SMS.
2. Membuat atau menghapus *folder* SMS.
3. Membaca, menulis, dan menghapus isi *phonebook*.
4. Membaca, menulis, dan menghapus catatan kalender pada ponsel.
5. Operasi memanggil atau menerima telepon.
6. Penanganan *setting* pada ponsel.
7. Penanganan *feature* WAP ponsel.
8. Penanganan keamanan, logo dan *ringtone* untuk ponsel.

2.1.6 Tinjauan Pustaka Gnokii

Gnokii merupakan sebuah aplikasi SMS Gateway yang memungkinkan administrasi ponsel dalam Linux. Selain Gnokii, terdapat solusi lainnya yaitu Gammu dan Kannel. Gammu merupakan pengembangan dari Gnokii. Sehingga aplikasi tersebut mendukung banyak ponsel Nokia. Sedangkan Kannel adalah sebuah aplikasi SMS Gateway dan WAP Gateway. Kannel mempunyai kelebihan dapat menangani beberapa SMS Center sekaligus. Namun Kannel hanya mendukung beberapa merek ponsel saja [AOU-07:01].

Keunggulan aplikasi Gnokii dibanding dengan aplikasi SMS Gateway yang lain, misalnya Gammu atau Kannel adalah:

1. Kompatibel dengan ponsel merek Nokia [YUN-06:98]
Aplikasi Gnokii pada awalnya hanya dikembangkan untuk ponsel Nokia. Dibandingkan dengan aplikasi Kannel yang lebih mendukung penggunaan modem GSM. Sehingga ponsel merk Nokia mendapat porsi terbesar dalam hal dukungan.
2. Kemudahan dalam konfigurasi aplikasi [YUN-06:105]
Dalam mengkonfigurasi aplikasi Gnokii, pengguna hanya perlu untuk mengubah isi *file* `.gnokiirc` yang terdapat pada *folder* pengguna.
3. Dukungan basis data [YUN-06:103]
SMS yang diterima maupun yang akan dikirim oleh aplikasi Gnokii disimpan dalam sebuah basis data. Basis data yang dapat digunakan agar aplikasi Gnokii dapat berfungsi dengan baik adalah PostgreSQL atau MySQL.

2.2 Sistem Informasi

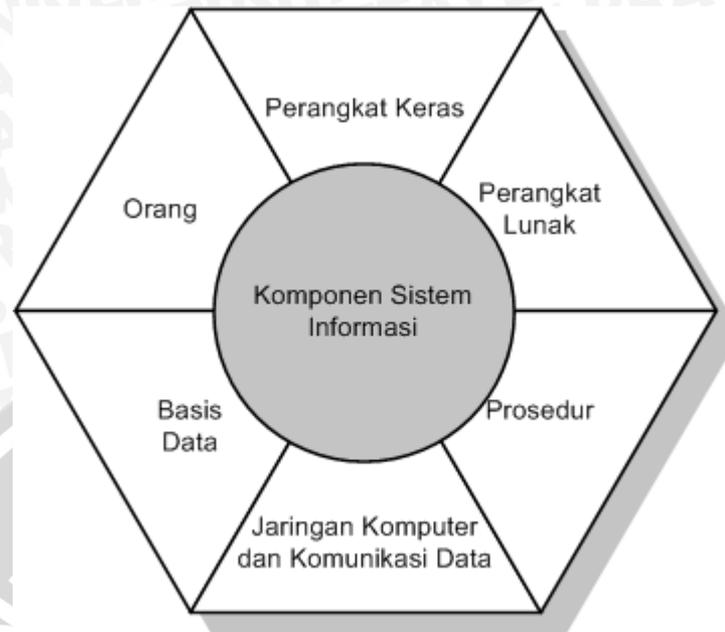
Menurut Hall [HAL-01:14], sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. Sedangkan menurut Alter [ALT-92:17] sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Dari definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

2.2.1 Komponen Sistem Informasi

Menurut Kadir [KAD-03:70], dalam suatu sistem informasi terdapat komponen-komponen seperti:

1. Perangkat keras (*hardware*): mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program: yaitu sekumpulan instruksi yang memungkinkan perangkat keras untuk memproses data.
3. Prosedur: sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang: semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Basis Data (*database*): sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data: sistem penghubung yang memungkinkan sumber-sumber (*resources*) dipakai secara bersama oleh sejumlah pemakai.



Gambar 2.5. Komponen Sistem Informasi.
Sumber: [KAD-03:71]

Dalam prakteknya, tidak semua sistem informasi mencakup keseluruhan komponen-komponen tersebut. Sebagai contoh, sistem informasi pribadi hanya melibatkan sebuah pemakai sebuah komputer tidak melibatkan fasilitas jaringan dan komunikasi. Namun, sistem informasi grup kerja (*workgroup information system*) yang melibatkan sejumlah orang dan sejumlah komputer, memerlukan sarana jaringan dan komunikasi.

2.2.2 Arsitektur Sistem Informasi

Arsitektur sistem informasi adalah suatu pemetaan atau rencana kebutuhan-kebutuhan informasi di dalam suatu organisasi. Arsitektur ini berguna sebagai penuntun bagi operasi sekarang atau menjadi cetak-biru (*blueprint*) untuk arahan masa mendatang. Tujuan dari arsitektur ini adalah agar bagian teknologi informasi dapat memenuhi kebutuhan-kebutuhan bisnis strategis organisasi. Oleh karena itu, arsitektur sistem informasi memadukan kebutuhan informasi, komponen sistem informasi, dan teknologi pendukung.

Sebuah arsitektur sistem informasi yang detail berisi perencanaan yang digunakan untuk menjawab pertanyaan-pertanyaan berikut [ALT-92:26]:

1. Data apa yang akan dikumpulkan?
2. Di mana dan bagaimana data akan dikumpulkan?
3. Di mana data akan disimpan?
4. Aplikasi-aplikasi (program) apa yang akan menggunakan data dan bagaimana aplikasi-aplikasi tersebut dihubungkan sebagai sebuah sistem yang utuh?

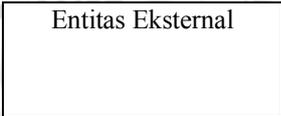
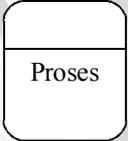
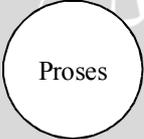
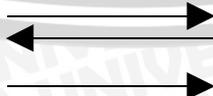
Arsitektur sistem informasi menggunakan arsitektur teknologi yang dapat dibedakan menjadi tiga macam, yaitu tersentralisasi (*centralized*), desentralisasi (*decentralized*), dan *client/server*.

2.2.3 Diagram Aliran Data (*Data Flow Diagram*)

Diagram Aliran Data (*Data Flow Diagram*) adalah sebuah representasi aliran data pada suatu sistem informasi [ANI-07:01]. *Data Flow Diagram* (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 mengalir atau dimana data tersebut akan disimpan [ANN-07:01].

Simbol-simbol yang digunakan dalam menggambarkan DFD diperlihatkan dalam tabel berikut:

Tabel 2.2. Simbol-Simbol yang Digunakan Dalam DFD.

Gane/Sarson	Yourdon/De Marco	Keterangan
 <p>Entitas Eksternal</p>	 <p>Entitas Eksternal</p>	Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
 <p>Proses</p>	 <p>Proses</p>	Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi
<p>Aliran Data</p> 	<p>Aliran Data</p> 	Aliran data dengan arah khusus dari sumber ke tujuan
<p>Data Store</p> 	<p>Data Store</p> 	Penyimpanan data atau tempat data yang direferensi oleh proses.

Sumber: [ANN-07:02]

2.3 Basis Data

Basis data (*database*) menurut Kadir adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi [KAD-03:254]. Sehingga basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas.

2.3.1 DBMS (*Database Management System*)

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS (*database management system*). DBMS adalah perangkat lunak sistem yang memungkinkan pemakai membuat memelihara, mengontrol, dan mengakses basis data dengan cara praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda.

Umumnya DBMS menyediakan fitur-fitur sebagai berikut:

1. Independensi data-program
Karena basis data ditangani oleh DBMS, program dapat ditulis sehingga tidak tergantung pada struktur data dalam basis data. Dengan kata lain, program tidak akan terpengaruh sekiranya bentuk fisik data diubah.
2. Keamanan
Keamanan dimaksudkan untuk mencegah pengaksesan data oleh pihak yang tidak berwenang.
3. Integritas
Hal ini ditunjukkan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.
4. Konkurensi
Konkurensi memungkinkan data dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.
5. Pemulihan (*recovery*)
DBMS menyediakan mekanisme untuk mengembalikan basis data ke keadaan semula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

6. Katalog sistem

Katalog sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

7. Perangkat produktifitas

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktifitas, DBMS menyediakan sejumlah perangkat produktifitas seperti pembangkit *query* dan pembangkit laporan.

2.3.2 Bahasa Basis Data

Semua DBMS mempunyai dua macam bahasa yang digunakan untuk mengelola dan mengorganisasikan data, yaitu:

1. Bahasa Definisi Data (*Data Definition Language* atau DDL)

DDL adalah perintah-perintah yang biasa digunakan oleh administrator basis data untuk mendefinisikan skema basis data dan juga sub-skema. Hasil kompilasi dari pernyataan-pernyataan DDL disimpan dalam berkas-berkas khusus yang disebut katalog sistem. Katalog sistem ini mengandung metadata, yaitu data yang menjelaskan objek-objek dalam basis data. Isi metadata adalah definisi rekaman-rekaman, *item* data, dan objek lain yang berguna bagi pemakai atau diperlukan oleh DBMS.

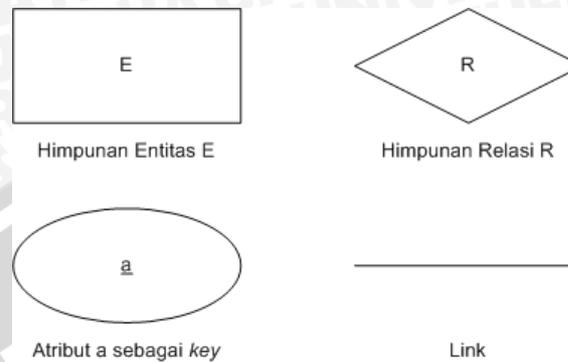
2. Bahasa Manipulasi Data (*Data Manipulation Data* atau DML)

DML adalah perintah-perintah yang digunakan untuk melakukan hal seperti mengambil, menambah, mengubah, dan atau menghapus data pada basis data. DML terbagi menjadi dua kelompok, yaitu prosedural dan non-prosedural. DML prosedural adalah perintah-perintah yang memungkinkan data apa saja yang diperlukan dan bagaimana cara mendapatkannya. Sedangkan DML non-prosedural adalah perintah-perintah yang memungkinkan pemakai menentukan data apa saja yang diperlukan, tanpa perlu menyebutkan cara mendapatkannya.

2.3.3 Diagram E-R (*Entity-Relationship*)

Diagram E-R (*Entity-Relationship*) merupakan suatu diagram sistematis yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi oleh atribut-atribut yang mempresentasikan seluruh fakta dari 'dunia nyata'

yang ditinjau [FAT-02:79]. Diagram E-R dapat digambarkan menggunakan beberapa notasi simbolik sebagai berikut:



Gambar 2.6. Notasi Dalam Diagram E-R.
Sumber: [FAT-02:80]

- Persegi panjang, menyatakan Himpunan Entitas.
- Lingkaran/Elips, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digaris bawah).
- Belah Ketupat, menyatakan Himpunan Relasi.
- Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.
- Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).

2.3.4 MySQL

MySQL adalah sebuah *server* basis data buatan T.c.X Data Konsultan AB, Swedia. Basis data MySQL banyak digunakan di Internet karena kehandalannya. MySQL tidak membutuhkan ruang *harddisk* yang besar untuk aplikasinya dan mudah digunakan untuk *server* basis data, sangat ideal untuk aplikasi yang kecil dan menengah. Keistimewaan MySQL dijelaskan seperti di bawah ini [IRE-02:5]:

1. Dukungan standar

MySQL mendukung level masukan ANSI SQL-92 dan ODBC level 0-2 SQL standar.

2. Dukungan bahasa
Server basis data *mysqld* dapat menampilkan pesan *error* dalam bahasa Czech, Belanda, Inggris, Estonia, Perancis, Jerman, Hongaria, Italia, Norwegia, dan lain-lain. MySQL yang menggunakan ISO-8859-1 karakter tertentu untuk data dan pengurutan. Karakter tersebut dapat dipilih ketika mengkompilasi *source*.
3. Bahasa pemrograman API untuk mengakses basis data
Aplikasi basis data MySQL dapat ditulis dengan bahasa tertentu seperti C, Perl, PHP, dan lain-lain.
4. Tabel yang besar
MySQL menyimpan masing-masing tabel dalam basis data seperti file, terpisah dalam direktori basis data. Ukuran maksimum tabel berkisar antara 4 GB dan sistem operasi mendekati ukuran maksimum.
5. Kecepatan, kekuatan dan kemudahan penggunaan
MySQL lebih cepat tiga atau empat kali lipat dari basis data komersial yang lain. MySQL sangat mudah untuk dikendalikan dan tidak membutuhkan *administrator* basis data terlatih untuk meng-*install* MySQL.

2.4 Internet

Internet merupakan jaringan terbesar yang menghubungkan jaringan komputer yang tersebar di seluruh penjuru dunia dan tidak terikat pada satu organisasipun. Siapa saja dapat bergabung pada Internet. Dengan menggunakan jaringan ini, sebuah organisasi dapat melakukan pertukaran informasi secara internal ataupun melakukan pertukaran secara eksternal dengan organisasi-organisasi yang lain. Dalam hal ini, jaringan tersusun atas berbagai jenis komputer dan sistem operasi.

Agar dapat terhubung dengan Internet, seorang pemakai dapat mengakses komputer pada perusahaan yang telah terkoneksi ke Internet atau perlu menjadi pelanggan dari sebuah ISP (*Internet Service Provider*). Alternatif lain, cukup melakukan pengaksesan pada warung-warung Internet (*warnet*). ISP adalah organisasi komersial yang bergerak dalam bidang penyediaan jasa akses ke Internet.

2.4.1 E-Commerce

E-Commerce digunakan untuk mendukung kegiatan pembelian dan penjualan, pemasaran produk, jasa, dan informasi melalui Internet atau Extranet. *E-commerce* umumnya dikelompokkan menjadi dua buah kategori: *business-to-business* (B2B) dan *business-to-consumer* (B2C). Pada perkembangan terakhir juga muncul jenis hubungan yang disebut *consumer-to-consumer* (C2C) dan *consumer-to-business* (C2B).

2.4.2 Keamanan Transaksi dalam E-Commerce

Keamanan merupakan hal mutlak dalam kegiatan transaksi dalam *e-commerce*. Menurut Ir. Dany Nugraha K. [NUG-97], terdapat dua metode dalam mengamankan transaksi dalam *e-commerce*, yaitu:

1. Metode Enkripsi.

Metode enkripsi atau yang lebih dikenal dengan kriptografi (*cryptograph*) adalah metode penyandian suatu pesan atau data yang terkirim melalui jaringan publik dengan menggunakan kunci-kunci (*keys*) tertentu. Beberapa teknologi enkripsi yang cukup populer adalah:

- Kombinasi *Public Key* dan *Private Key*.

Pada teknologi *public key* dan *private key*, penyandian dilakukan dengan menggunakan kombinasi antara *public* dan *private key*. *Public Key* merupakan kunci yang dikenal oleh umum, sedangkan *Private Key* merupakan kunci yang hanya dikenal oleh si pemilikinya.

- *Certificate Authority/Digital Signature*

Pada penggunaan *public key* di atas masih dimungkinkan adanya pencurian atau pemalsuan *public key*. Oleh karenanya diperlukan adanya keterlibatan pihak ketiga yang dapat dipercaya (yang menjamin keabsahan dari suatu *public key*), yaitu *Certification Authority* (CA). CA inilah yang akan memberikan sertifikasi atas *public key*. Sertifikasi yang diberikan kepada *public key* seseorang ini dikenal sebagai *Digital Signature*.

Ibarat suatu surat atau bon, dianggap sah apabila sudah dibubuhi stempel atau cap dari suatu perusahaan, begitu pula dengan *public key*. Apabila sudah memiliki *Digital Signature* atau CA, *public key* tersebut benar-benar asli dan sah.

- *Secure Electronic Transaction* (SET)

SET pertama kali diperkenalkan oleh RSA Data Security, suatu lembaga

independen yang mengeluarkan berbagai standarisasi dalam hal Internet Security. Teknologi yang digunakan dalam SET merupakan gabungan antara teknologi enkripsi *public key/private key* dengan teknologi *digital signature*. Saat ini SET dijadikan standar protokol untuk pembayaran dengan kartu kredit dalam *e-commerce*. Pada SET, enkripsi *public key* menggunakan enkripsi 56 bit sampai dengan 1024 bit, sehingga tingkat kombinasi enkripsinya pun sangat tinggi. Sehingga amat sulit untuk melakukan pembongkaran atas enkripsi ini.

RSA menyatakan bahwa dibutuhkan waktu 500 tahun untuk bisa membuka enkripsi ini. SET didukung oleh sebagian besar perusahaan penerbit kartu kredit, seperti Visa dan Mastercard.

2. Metode VPN (*Virtual Private Network*).

Virtual Private Network atau Jaringan Pribadi Maya sesungguhnya sama dengan Jaringan Pribadi (*Private Network/PN*) pada umumnya, di mana satu jaringan komputer suatu lembaga atau perusahaan di suatu daerah atau negara terhubung dengan jaringan komputer dari satu grup perusahaan yang sama di daerah atau negara lain. Perbedaannya hanyalah pada media penghubung antar jaringan. Pada PN, media penghubungnya masih merupakan milik perusahaan/grup itu sendiri. Sedangkan dalam VPN, media penghubungnya adalah jaringan publik seperti Internet.

Dalam VPN, karena media penghubung antar jaringannya adalah jaringan publik, diperlukan pengamanan dan pembatasan-pembatasan. Pengamanan diperlukan untuk menjaga agar tidak sebarang orang dari jaringan publik dapat masuk ke jaringan pribadi. Yang dikecualikan hanyalah orang-orang yang terdaftar atau terotentifikasi terlebih dahulu yang dapat masuk ke jaringan pribadi. Pembatasan diperlukan untuk menjaga agar tidak semua orang atau *user* dari jaringan pribadi dapat mengakses jaringan publik (Internet).

Secara garis besar, ada dua cara membentuk VPN, yaitu :

- *Tunneling*

Sesuai dengan arti *tunnel* atau lorong, dalam membentuk suatu VPN ini dibuat suatu *tunnel* di dalam jaringan publik untuk menghubungkan antara jaringan yang satu dan jaringan lain dari suatu grup atau perusahaan yang ingin membangun VPN tersebut. Seluruh komunikasi data antarjaringan pribadi akan melalui *tunnel* ini, sehingga orang atau *user* dari jaringan publik yang tidak memiliki izin untuk masuk tidak akan mampu

untuk menyadap, mengacak atau mencuri data yang melintasi *tunnel* ini. Ada beberapa metode *tunelling* yang umum dipakai, di antaranya adalah *IPX To IP Tunnelling*, atau *PPP To IP Tunnelling*.

IPX To IP tunnelling biasa digunakan dalam jaringan VPN Novell Netware. Jadi dua jaringan Novell yang terpisah akan tetap dapat saling melakukan komunikasi data melalui jaringan publik Internet melalui *tunnel* ini tanpa khawatir akan adanya gangguan pihak ke-3 yang ingin mengganggu atau mencuri data. Pada *IPX To IP tunnelling*, paket data dengan protokol IPX (standar protokol Novell) akan dibungkus (*encapsulated*) terlebih dahulu oleh protokol IP (standar protokol Internet) sehingga dapat melewati *tunnel* ini pada jaringan publik Internet. Sama halnya untuk *PPP To IP tunnelling*, di mana PPP protokol di-*encapsulated* oleh protokol IP.

- *Firewall*

Sebagaimana layaknya suatu dinding, *firewall* akan bertindak sebagai pelindung atau pembatas terhadap orang-orang yang tidak berhak untuk mengakses jaringan kita. Umumnya dua jaringan yang terpisah yang menggunakan *firewall* yang sejenis, atau seorang *remote user* yang terhubung ke jaringan dengan menggunakan *software client* yang terenkripsi akan membentuk suatu VPN, meskipun media penghubung dari kedua jaringan tersebut atau penghubung antara *remote user* dengan jaringan tersebut adalah jaringan publik seperti Internet.

Suatu jaringan yang terhubung ke Internet pasti memiliki *IP address* (alamat Internet) khusus untuk masing-masing komputer yang terhubung dalam jaringan tersebut. Apabila jaringan ini tidak terlindungi oleh *tunnel* atau *firewall*, *IP address* tadi akan dengan mudahnya dikenali atau dilacak oleh pihak-pihak yang tidak diinginkan. Akibatnya data yang terdapat dalam komputer yang terhubung ke jaringan tadi akan dapat dicuri atau diubah. Dengan adanya pelindung seperti *firewall*, kita bisa menyembunyikan (*hide*) *address* tadi sehingga tidak dapat dilacak oleh pihak-pihak yang tidak diinginkan.

Kemampuan yang dimiliki oleh *firewall*, antara lain adalah:

- *IP Hiding/Mapping*. Kemampuan ini mengakibatkan *IP address* dalam jaringan dipetakan atau ditranslasikan ke suatu *IP address* baru. Dengan demikian *IP address* dalam jaringan tidak akan dikenali di Internet.
- *Privilege Limitation*. Dengan kemampuan ini kita dapat membatasi para *user* dalam jaringan sesuai dengan otorisasi atau hak yang diberikan kepadanya.

- *Outside Limitation*. Dengan kemampuan ini kita dapat membatasi para *user* dalam jaringan untuk hanya mengakses ke alamat-alamat tertentu di Internet di luar dari jaringan kita.
- *Inside Limitation*. Kadang-kadang kita masih memperbolehkan orang luar untuk mengakses informasi yang tersedia dalam salah satu komputer (misalnya *Web Server*) dalam jaringan kita. Selain itu, tidak diperbolehkan, atau memang sama sekali tidak dizinkan untuk mengakses seluruh komputer yang terhubung ke jaringan kita.
- *Password and Encrypted Authentication*. Beberapa *user* di luar jaringan memang diizinkan untuk masuk ke jaringan kita untuk mengakses data dan sebagainya, dengan terlebih dahulu harus memasukkan *password* khusus yang sudah terenkripsi.

2.4.3 osCommerce

osCommerce adalah suatu program perangkat lunak manajemen *e-commerce* dan toko *online* [ANO-06:1]. osCommerce dapat digunakan pada semua *server* web yang mempunyai PHP dan MySQL terpasang. Ia berlisensi GPL (GNU General Public License). OsCommerce mempunyai baik sebuah katalog dimuka dan alat administrasi dibelakang, yang bisa diinstal secara mudah dan dikonfigurasi melalui prosedur instalasi berbentuk web.

osCommerce mengkombinasikan solusi Open Source untuk menyediakan sebuah *platform* bebas dan terbuka, yang meliputi bahasa *scripting* web PHP yang kuat dan *server* basis data MySQL. Tanpa kebutuhan khusus, osCommerce dapat berjalan pada *server* web dengan PHP 4.1+ yang diaktifkan pada lingkungan Linux, Solaris, BSD, Mac OS X, dan Microsoft Windows.

Fitur-fitur dari osCommerce adalah:

1. Kompatibel dengan PHP 4 dan MySQL –portabel secara mudah ke PHP v5.
2. Kompatibel dengan *browser* yang umum.
3. Berbahasa ragam seperti Inggris, Jerman dan Spanyol untuk standarnya.
4. Instalasi berdasarkan web secara otomatis.
5. Produk dan kategori dengan jumlah tak terbatas.
6. *Setup* otomatis dari sistem pembayaran *e-commerce*: Authorize.net,

2Checkout.com, iPayment.de, PayPal.com, PsiGate.com (sudah tidak berjalan), SECPay.com, TrustCommerce.com; banyak sistem *e-commerce* tambahan lainnya, pemrosesan pembayaran *off-line* (penggunaan kartu kredit yang tidak aman) dan *on-line*.

7. Dukungan mata uang beragam melalui update manual dari server kurs pertukaran mata uang asing.
8. Mendukung gambar dinamis (satu gambar per produk).
9. Mendukung produk fisik maupun produk yang di-*download*.
10. Modul administrasi berdasarkan *web*.
11. Basis data yang bisa di-*backup* dan *restore*.
12. Kereta belanja yang permanen dan sementara.
13. Transaksi aman dengan dukungan SSL (Secure Socket Layer).
14. Banyak pilihan pengiriman termasuk berat, harga, atau tujuan.
15. Integrasi kuota pengiriman *real time* dengan UPS, USPS.
16. Pilihan pengiriman berdasarkan zona seperti pilihan pengiriman gratis sebagai bagian dari *Order Totals Module*.
17. Fungsional pajak yang penuh, termasuk pajak tambahan, zona berbeda untuk produk yang berbeda.

2.4.4 Tinjauan Pustaka osCommerce

osCommerce merupakan aplikasi *e-commerce* yang berdasarkan sistem CMS. osCommerce mempunyai fungsi-fungsi standar, mudah digunakan, dan mempunyai desain yang telah disediakan. Solusi lainnya adalah CRE Loaded dan Zen Cart. CRE Loaded adalah sebuah aplikasi *e-commerce* pengembangan dari osCommerce. CRE Loaded merupakan sebuah kombinasi dari sistem *template* osCommerce dan inovasi osCommerce yang paling populer kemudian dibangun ke dalam sebuah *engine*. Sedangkan Zen Cart merupakan aplikasi *e-commerce* yang menggunakan struktur dan solusi program yang berbeda dari lainnya. Zen Cart merupakan solusi aplikasi *e-commerce* bagi yang menginginkan sebuah aplikasi yang sangat mendasar untuk membuat toko *online* [AON-07:01].

Keunggulan osCommerce dibandingkan dengan aplikasi *e-commerce* lainnya, seperti CRE Loaded, atau Zen Cart adalah:

1. Kemudahan dalam instalasi aplikasi [ANS-07:01]
Proses instalasi aplikasi osCommerce menggunakan *web browser (web-based)*. Sehingga relatif mudah bagi pengguna.
2. Kemudahan dalam konfigurasi [AON-07:01]
Konfigurasi dalam aplikasi osCommerce terbagi menjadi beberapa bagian atau modul. Selain itu, bagian konfigurasi atau administrasi terpisah dari aplikasi utama osCommerce. Sehingga memudahkan administrator dalam mengkonfigurasi toko *online*.
3. Tersedia modul-modul tambahan [KOE-04:01]
Aplikasi osCommerce dapat ditambahkan modul-modul baru untuk menambah fiturnya. Modul-modul tersebut tersedia dalam *website* resmi osCommerce.

2.5 Perl

Perl adalah suatu bahasa pemrograman tingkat tinggi yang pada awalnya ditulis oleh Larry Wall dan sekarang dikembangkan oleh ribuan orang. Semantik dari bahasa pemrograman Perl banyak didasarkan pada bahasa pemrograman C, sementara juga banyak mewariskan fitur-fitur terbaik dari sed, awk, *shell* Unix, dan banyak bahasa pemrograman lainnya.

Perl merupakan sebuah bahasa yang sangat kuat dalam mengolah proses, file dan manipulasi teks. Tidak seperti utilitas Unix kebanyakan, Perl tidak membatasi ukuran dari data masukan, tetapi bergantung pada memori yang tersedia. Perl dapat mengolah semua file sebagai sebuah *string* tunggal. Rekursi mempunyai kedalaman yang tak terhingga. Dan tabel yang digunakan oleh *hash* (seringkali disebut *array* asosiatif) berkembang seperlunya untuk mencegah penurunan performa. Perl dapat menggunakan teknik penyamaan pola untuk memindai sejumlah besar data dengan cepat. Ini menjadikannya sangat berguna dalam utilitas sistem, *tool* perangkat lunak, tugas manajemen sistem, akses basis data, pemrograman grafis, jaringan, dan pemrograman web.

Kekuatan ini yang menjadikannya menarik bagi penulis skrip CGI, administrator sistem, matematikawan, wartawan, dan hampir semua orang yang memerlukan untuk menulis aplikasi dan utilitas dengan cepat.

Perl pada awalnya dikembangkan pada lingkungan Unix, tapi telah menjadi sebuah alat pengembangan lintas *platform*. Perl berjalan pada *mainframe* IBM; AS/400;

Windows NT, 98, 95; Novell Netware, superkomputer Cray; VMS Digital; Tandem Guardian; HP MPE/ix; Mac OS; dan semua turunan Unix, termasuk Linux. Sebagai tambahan, Perl telah di-*porting* ke banyak sistem operasi yang lebih kecil, termasuk BeOS, RISCOS Acorn, dan bahkan mesin seperti Amiga.

2.5.1 Tipe Data

Tidak seperti bahasa pemrograman lain seperti C/C++ atau Pascal, Perl hanya memiliki tiga jenis tipe data, yaitu:

1. Skalar
2. *Array*, dan
3. *Array* asosiatif

Tipe data skalar dapat berupa *string* (deretan karakter) atau bilangan. *Array* menyatakan suatu tipe data yang dapat menampung sejumlah nilai tipe skalar, sedangkan *array* asosiatif merupakan suatu jenis *array* khusus. *Array* asosiatif (sering disebut *hash*) dibedakan terhadap *array* berdasarkan subskripnya. Pada *array* biasa, subskrip berupa angka. Sedangkan pada *array* asosiatif, subskrip yang digunakan dapat berupa *string*.

2.5.2 Operasi Berkas

Dalam melakukan suatu operasi pada berkas, seringkali kita harus melewati tiga tahapan [KAD-02:278]. Tiga tahapan itu adalah membuka berkas, melaksanakan pemrosesan terhadap berkas, dan menutup berkas. Tahapan tersebut kira-kira sama dengan kegiatan membaca atau menulis buku. Pertama, membuka buku. Kedua, membaca isinya atau menulis ke dalam buku. Terakhir, menutup buku tersebut kembali.



Gambar 2.7. Tahapan Operasi Pada Berkas.

Sumber: [KAD-02:278]

Untuk membuka berkas Perl menggunakan fungsi `open()`. Salah satu bentuk pemanggilan fungsi `open()` adalah dengan bentuk seperti berikut:

```
open(penangan_berkas, nama_berkas);
```

Dalam hal ini:

- `penangan_berkas` adalah penangan berkas (sering juga disebut variabel *berkas*) yang nantinya dipakai pada fungsi-fungsi lain dalam pemrosesan berkas sebagai penghubung antara proses Perl dengan berkas fisik.
- `nama_berkas` menyatakan nama berkas yang akan dibuka.

Nama variabel `berkas` atau penangan `berkas` dapat berupa sekumpulan huruf, angka, atau garis-bawah (`_`), dengan syarat karakter pertama berupa huruf. Umumnya (menurut kebiasaan), nama penangan berkas menggunakan huruf kapital. Contoh:

```
BERKAS_MASUKAN  
MASUKAN  
BERKAS_KELUARAN  
BERKAS_1
```

Fungsi `open()` sesungguhnya memberikan nilai balik berupa:

nilai BUKAN-NOL jika berkas berhasil dibuka, atau nilai NOL jika terjadi suatu kegagalan dalam membuka berkas.

Ketika menutup berkas Perl menyediakan fungsi `close()`. Salah satu bentuk pemakaiannya berupa:

```
close(penangan_berkas);
```

Dalam hal ini `penangan_berkas` adalah penangan berkas yang sama seperti yang digunakan pada fungsi `open()`.

2.5.3 Operasi String

Pada Perl, operator-operator yang terkait dengan string meliputi:

1. Operator titik(`.`), yang berguna untuk menggabungkan string.
2. Operator `.=`, yang berguna untuk menambahkan string ke suatu variabel.
3. Operator `x`, yang berguna untuk mengulang string.

Operator `x` digunakan dalam ekspresi dengan bentuk sebagai berikut:

```
string x_jumlah
```

Hasil ekspresi di atas berupa string yang diulang sebanyak jumlah kali.

Berbagai fungsi yang berhubungan dengan string, yaitu:

1. Fungsi `lc()` dan `lcfirst()`

Fungsi `lc()` berguna untuk memperoleh string yang seluruhnya berupa huruf kecil.

Bentuk pemakaiannya:

```
lc(string)
```

Adapun fungsi `lcfirst()` berguna untuk mendapatkan string dengan huruf pertama yang akan dikonversi menjadi huruf kecil. Bentuk pemakaiannya:

```
lcfirst(string)
```

2. Fungsi `uc()` dan `ucfirst()`

Fungsi `uc()` berguna untuk memperoleh string yang seluruh hurufnya berupa huruf kapital. Bentuk pemakaiannya:

```
uc(string)
```

Adapun fungsi `ucfirst()` berguna untuk mendapatkan string dengan huruf pertama akan dikonversi menjadi huruf kapital. Bentuk pemakaiannya:

```
ucfirst(string)
```

3. Fungsi `substr()`

Fungsi ini berguna untuk memperoleh substring dari suatu string. Bentuk

pemakaiannya:

```
substr(string, ofset, jum)
substr(string, ofset)
```

Pada bentuk pertama, hasil berupa substring yang berasal dari string dimulai dari posisi ofset dengan panjang berupa jum karakter. Pada bentuk kedua, hasil berupa substring dimulai dari posisi ofset hingga karakter terakhir pada string.

4. Fungsi index() dan rindex()

Fungsi index() berguna untuk memperoleh posisi suatu substring di dalam suatu string. Bentuk pemanggilannya:

```
index(string, substring)
index(string, substring, posisi)
```

Pada bentuk pertama, nilai balik fungsi berupa posisi pada string tempat substring ditemukan pertama kali. Pada bentuk kedua, posisi menentukan posisi awal pada string yang digunakan untuk mencari substring. Bila substring tidak ditemukan pada string, nilai balik fungsi index() berupa -1.

Adapun fungsi rindex() mempunyai kegunaan semacam index(), tetapi pencarian dimulai bukan dari kiri melainkan dari kanan. Dengan kata lain, rindex() berfungsi untuk memperoleh kemungkinan terakhir posisi substring pada string.

5. Fungsi ord() dan chr()

Fungsi ord() berguna untuk memperoleh nilai ASCII dari karakter pertama yang terdapat pada argumennya. Bentuk pemanggilannya:

```
ord(string)
ord
```

Bila tidak ada argumen yang diberikan, string pada variabel \$_ yang dievaluasi.

Fungsi chr() merupakan kebalikan dari fungsi ord(). Bentuk pemanggilannya berupa:

```
chr(numerik)
```

Argumen berupa nilai ASCII dan nilai baliknya berupa karakter yang nilai ASCII-nya sesuai dengan argumen.

6. Fungsi sprintf()

Fungsi sprintf() mempunyai kegunaan seperti printf() tetapi hasilnya dalam bentuk string. Bentuk pemakaian fungsi ini:

```
sprintf(string_format, argumen)
```

string_format menentukan format keluaran argumen-argumennya. Pada bagian

ini bisa menggunakan satu atau beberapa penentu format yang beberapa diantaranya dapat dilihat di tabel di bawah.

Tabel 2.3. Penentu format pada sprintf().

Penentu Format	Keterangan
%s	Untuk menampilkan string
%b	Untuk menampilkan data dalam bentuk biner
%u	Untuk menampilkan data bilangan bulat tak bertanda
%d atau %i	Untuk menampilkan bilangan bulat dalam bentuk desimal (basis 10)
%b	Untuk menampilkan bilangan bulat dalam bentuk biner (basis 2)
%o	Untuk menampilkan bilangan bulat dalam bentuk oktal (basis 8)
%x	Untuk menampilkan bilangan bulat dalam bentuk heksadesimal (basis 16) dengan notasi yang digunakan untuk huruf berupa huruf kecil (a, b, c, d, e, dan f)
%X	Untuk menampilkan bilangan bulat dalam bentuk heksadesimal (basis 16) dengan notasi yang digunakan untuk huruf berupa huruf kapital (A, B, C, D, E, dan F)
%f	Untuk menampilkan bilangan dalam bentuk real dengan notasi dddd.dddddd
%e atau %E	Untuk menampilkan bilangan dalam bentuk notasi ilmiah
%g	Menampilkan bilangan dalam bentuk notasi %f atau %e tergantung oleh kepresisian data (digit nol yang tidak berarti tidak ditampilkan)

Sumber: [KAD-02:60-61]

7. Fungsi q//

Fungsi q// (seringkali juga disebut operator) memiliki bentuk seperti:

```
q/string/
```

```
q(string)
```

Fungsi ini identik dengan penulisan string yang diletakkan dalam tanda petik tunggal:

```
'string'
```

Fungsi q// tidak bisa melakukan interpolasi variabel. Dengan demikian yang diletakkan dalam q tidak bisa berupa suatu variabel.

1. Fungsi qq//

Fungsi `qq//` memiliki bentuk seperti berikut:

```
qq/string/
```

```
qq(string)
```

Fungsi ini identik dengan penulisan string yang diletakkan dalam tanda petik ganda:

```
"string"
```

Fungsi `qq//` bisa melakukan interpolasi variabel. Dengan demikian yang diletakkan dalam `qq` bisa berupa suatu variabel.

2. Fungsi `qw//`

Fungsi `qw//` memiliki bentuk seperti berikut:

```
qw/string/
```

```
qw(string)
```

Hasilnya berupa sebuah list yang berisi kata-kata dalam string. Dalam hal ini, spasi yang dijadikan pedoman sebagai pemisah antar kata. Fungsi `qw//` tidak bisa melakukan interpolasi variabel. Dengan demikian yang diletakkan dalam `qw` tidak bisa berupa suatu variabel.

3. Fungsi `qx//`

Fungsi `qx//` memiliki bentuk seperti berikut:

```
qx/string/
```

```
qx(string)
```

Bentuk diatas identik dengan penulisan string yang diletakkan dalam tanda *backquote*:

```
`string`
```

Fungsi `qx//` tidak bisa melakukan interpolasi variabel. Dengan demikian yang diletakkan dalam `qx` tidak bisa berupa suatu variabel.

4. Fungsi `quotmeta()`

Fungsi ini memiliki bentuk sebagai berikut:

```
quotmeta(string)
```

Bentuk diatas mempunyai nilai balik berupa string dengan seluruh metakarakter (karakter-karakter yang mempunyai makna khusus) diberi *backslash*.

2.5.4 Pengenalan dan Pencocokan Pola

Yang dimaksud dengan pola adalah sederetan karakter yang dicari dalam sebuah string. Misalnya, hendak bermaksud mencari substring "hai" di dalam suatu string, maka

“hai” berkedudukan sebagai pola. Namun tentu saja sebuah pola bisa tidak sesederhana itu, melainkan melibatkan tanda-tanda khusus.

Pencocokan pola dapat dilakukan dengan menggunakan ekspresi reguler (*regular expression*). Sebagai gambaran, jika bermaksud mencari nama seseorang yang berawalan a, maka pencarian dapat dilakukan dengan menggunakan ekspresi reguler.

Pada bahasa pemrograman Perl, ada lima operator yang berhubungan dengan ekspresi reguler (regex). Yakni `m//`, `qr//`, `s///`, `=~`, dan `!~`. Dua operator pertama (`m//` dan `qr//`) adalah operator yang dipakai untuk mengapit dan mengompilasi sebuah pola regex. Operator ketiga dan keempat (`=~` dan `!~`) digunakan untuk pencocokan atau substitusi regex dengan sebuah string. Sementara operator terakhir (`s///`) digunakan untuk substitusi regex.

Perlu diketahui bahwa pola regex tidak ditulis dalam bentuk string biasa, misalnya “ka[km]u”. Melainkan harus dikutip oleh salah satu dari kedua operator pengutip regex yang telah disebutkan sebelumnya. Misalnya `m/ka[km]u/` atau `qr/ka/[km]u/`. Karena operator-operator pengapit regex tersebut sekaligus berfungsi untuk mengompilasi pola regex sehingga dapat dipakai.

Selain keenam operator tersebut, ada fungsi-fungsi lain yang berhubungan dengan regex, yaitu `quotemeta()`, `split()`, dan `grep()`. Berikut merupakan penjelasan lebih detail mengenai operator pencocokan, yaitu:

1. Pencocokan dengan Operator `m//`

Operator pencocokan `m//` berfungsi untuk mencocokkan pola dalam suatu string. Operator ini melakukan pencarian pola terhadap variabel `$_`. Bentuk pemakaiannya:

```
m/pola/
```

Hasilnya berupa benar jika `pola` terdapat pada variabel `$_` dan salah jika `pola` tidak ditemukan pada variabel `$_`. Penulisan seperti:

```
m/Jawa Timur/
```

juga dapat hanya ditulis menjadi:

```
/Jawa Timur/
```

2. Pencocokan dengan Operator `=~` dan `!~`

Untuk memeriksa apakah suatu pola berada dalam string, Perl menyediakan operator `=~`. Bentuk ekspresi yang menggunakan operator ini:

```
string =~ pola
```

Ekspresi ini menghasilkan nilai benar jika pola ditemukan pada string dan salah jika pola tidak ditemukan pada string.

Jika operator `=~` adalah operator “cocok dengan”, Perl menyediakan operator yang berfungsi berkebalikan dengannya berupa `!~`. Operator ini dapat dibaca sebagai “operator tidak cocok dengan”. Bentuk pemakaiannya:

```
string !~ pola
```

Hasil ekspresi berupa salah jika pola cocok dengan string dan benar jika pola tidak cocok dengan string.

3. Pencocokan dengan Operator `qr//`

Operator `qr//` berfungsi sama dengan operator `m//`. Perbedaan diantara keduanya adalah pada operator `qr//`, pola regex akan disimpan dahulu ke dalam sebuah variabel. Sedangkan pada operator `m//` makan akan langsung diproses. Bentuk pemakaiannya sama seperti operator `m//`, yaitu:

```
qr/pola/
```

Hasil akan benar bila pola terdapat pada variabel `$_` dan salah bila tidak ada pola pada variabel `$_`.

4. Pencocokan dengan Operator `s///`

Operator `s///` merupakan sebuah operator substitusi. Bila pada operator `m//` hanya akan mencari pola regex, maka operator `s///` juga dapat mengganti pola regex yang telah dicari. Bentuk pemakaian operator ini adalah:

```
s/pola/pengganti/
```

Hasil yang terjadi adalah bagian yang cocok dengan pola akan diganti dengan pengganti.

5. Pencocokan dengan fungsi `split`

Fungsi `split` merupakan bagian dari regex. Fungsi ini berguna untuk memecah sebuah string berdasarkan sebuah pola. Pola ini bisa berupa tabulasi, titik-dua, spasi kosong, atau data apapun juga. Bentuk pemakaiannya berupa:

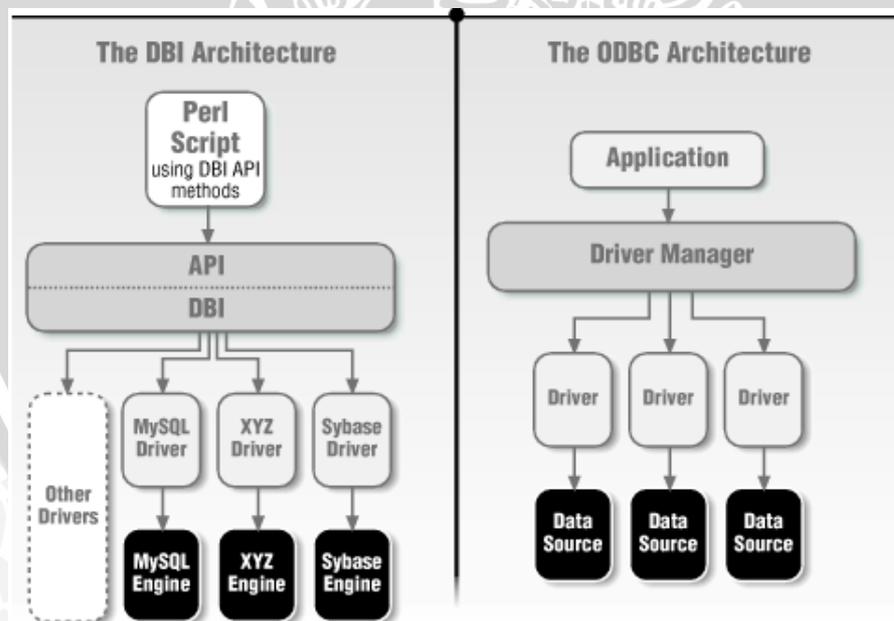
```
split/pola/$string
```

Hasilnya akan berupa beberapa string baru yang merupakan pemisahan dari string yang menggunakan pola.

2.5.5 Koneksi Basis Data dengan Basis Data MySQL

Terdapat dua buah *framework* standar dalam berkomunikasi dengan sebuah server SQL, yaitu DBI (*DataBase Interface*) dan ODBC (*Open DataBase Connectivity*). Dahulunya, DBI merupakan standar Unix dan ODBC merupakan standar Win32, tapi perbedaan ini kabur karena sekarang ODBC telah tersedia dalam dunia Unix dan DBI telah di-*port* ke Win32. Semakin kabur dalam baris perintah adalah paket DBD:ODBC, sebuah modul DBD menggunakan ODBC yang berasal dari dalam *framework* DBI.

DBI dan ODBC sangat mirip dalam maksud dan eksekusi. Baik DBI maupun ODBC dapat dianggap sebagai “*middleware*”. Mereka membentuk sebuah *layer* abstraksi yang memungkinkan *programmer* untuk menulis kode menggunakan pemanggilan DBI/ODBC generik, tanpa mengetahui API spesifik dari suatu basis data tertentu. Modul DBI memanggil sebuah *driver* DBI untuk melakukan hal ini. ODBC Manager memanggil driver data sumber ODBC yang spesifik. Gambar di bawah menunjukkan arsitektur DBI dan ODBC.



Gambar 2.8. Struktur DBI dan ODBC.
Sumber: [BLA-01:45]

Pengaksesan DBI dalam Perl cukup menggunakan langkah-langkah sebagai berikut [BLA-01:46]:

1. Langkah Pertama. Memuat modul Perl yang diperlukan.

Pertama kali dalam menggunakan DBI adalah menggunakan perintah:

```
use DBI;
```

2. Langkah Kedua. Koneksi ke basis data dan menerima sebuah penanganan koneksi.

Memasukkan perintah DBI untuk membangun koneksi DBI pada sebuah basis data MySQL dan mengembalikan penanganan sebuah basis data yang menggunakan perintah:

```
# connect using to the database named $database using given
```

```
# username and password, return a database handle
```

```
$database = "sysadm";
```

```
$dbh = DBI->connect("DBI:mysql:$database", $username, $pw);
```

```
die "Unable to connect: $DBI::errstr\n" unless (defined $dbh);
```

DBI akan memuat sebuah *driver* level rendah (DBD::mysql) terlebih dahulu sebelum berhubungan dengan server. Kemudian menguji jika kode `connect()` berhasil sebelum melanjutkan. DBI menyediakan opsi `RaiseError` dan `PrintError` untuk `connect()`, ketika menginginkan DBI untuk melakukan tes atau secara otomatis mengeluarkan komplain mengenai kesalahan ketika terjadi.

3. Langkah Ketiga. Mengirim perintah SQL ke *server*.

Dengan modul Perl yang telah dimuat dan sebuah koneksi ke *server* basis data pada tempatnya, maka hanya perlu memasukkan perintah SQL yang diperlukan.

Berikut adalah contoh metode DBI dalam mengirimkan perintah:

```
$results=$dbh->do(q{UPDATE hosts
```

```
    SET bldg = 'Main'
```

```
    WHERE name = 'bendir'});
```

```
die "Unable to perform update:$DBI::errstr\n" unless (defined $results);
```

4. Langkah Keempat. Mengeluarkan hasil SELECT.

Ketika kita mengirimkan sebuah pernyataan SELECT ke *server* menggunakan `execute()`, kita menggunakan sebuah mekanisme yang memungkinkan kita mengeluarkan hasil satu baris dalam satu waktu.

Dalam DBI, kita memanggil satu dari metode-metode yang ada dalam Tabel 2.4 untuk mengembalikan data dari kumpulan hasil.

Tabel 2.4. Metode DBI untuk Mengembalikan Data.

Nama	Pengembalian	Pengembalian Bila Tidak Ada Baris Lagi
<code>fetchrow_arrayref()</code>	Sebuah referensi <i>array</i> ke sebuah <i>array</i> anonim yang merupakan kolom-kolom dari baris berikutnya dalam sebuah kumpulan hasil	Tidak terdefinisi
<code>fetchrow_array()</code>	Sebuah <i>array</i> dengan nilai yang merupakan kolom-kolom dari baris berikutnya dalam sebuah kumpulan hasil	Sebuah daftar kosong
<code>fetchrow_hashref()</code>	Sebuah referensi <i>hash</i> pada sebuah <i>hash</i> anonim yang merupakan nama dan nilai kolom yang merupakan nilai dari kolom-kolom dari baris berikutnya dalam sebuah kumpulan hasil	Tidak terdefinisi
<code>fetchall_arrayref()</code>	Sebuah referensi dari sebuah <i>array</i> dari struktur data <i>array</i>	Sebuah referensi pada sebuah <i>array</i> kosong

Sumber: [BLA-01:46]

5. Langkah Kelima. Menutup koneksi ke *server*.

Dalam DBI cukup dengan memasukkan perintah:

```
# tells server you will not need more data from statement handle
# (optional, since we're just about to disconnect)
$sth->finish;
# disconnects handle from database
$dbh->disconnect;
```

2.6 **Web Server**

Web server adalah sebuah program *server* yang berada di belakang WWW (*World Wide Web*). Ia menerima permintaan dari *client*, seperti *web browser*. Jika ia mendapatkannya, maka ia memprosesnya dan mengembalikan beberapa data. Data ini biasanya berupa halaman terformat dengan teks dan gambar. *Browser* kemudian mengolah data ini dengan kemampuan terbaiknya dan memperlihatkannya pada *user*. *Web server* dalam konsepnya sangat sederhana, mereka hanya menunggu dan mengolah permintaan yang datang.



Web server berkomunikasi dengan *browser* atau *client* lainnya menggunakan HTTP (*Hyper Text Transfer Protocol*), yang merupakan protokol sederhana menstandarkan cara bagaimana permintaan dikirim dan diproses. Ini memungkinkan bervariasinya *client* untuk berkomunikasi dengan semua *server vendor* tanpa permasalahan kompatibilitas.

2.6.1 Apache

Apache merupakan turunan dari *web server* yang dikeluarkan oleh NSCA, yaitu NSCA HTTPd sekitar tahun 1995. Pada dasarnya, Apache merupakan sebuah *patch* (dari kata “*A PatCHy*”) dan pengganti dari NCSA HTTPd. Apache *web server* merupakan tulang punggung dari *World Wide Web* (WWW). *Web server* menunggu permintaan dari *client* yang menggunakan *browser*, seperti Netscape Navigator, Internet Explorer, Mozilla, lynx, dan lain-lain. *Web server* dalam berkomunikasi dengan kliennya menggunakan protokol HTTP (*Hyper Text Transfer Protocol*). Saat ini Apache banyak digunakan sebagai *web server* untuk portal-portal besar. Adapun pertimbangan dalam memilih Apache adalah [IRE-02:3]:

1. Apache termasuk dalam kategori perangkat lunak bebas.
2. Instalasi Apache sangat mudah.
3. Mampu beroperasi pada banyak *platform* sistem operasi, seperti AUX 3.1, BSDX 2.0, FreeBSD 2.1, HP-UX 9.07, IRIX 5.3, Linux, SolarisX86, Windows, dan lain-lain.
4. Mudah dalam pengkonfigurasiannya karena Apache hanya mempunyai empat buah *file* konfigurasi.
5. Apache mudah dalam penambahan *peripheral* lainnya ke dalam *platform web server*, misalnya menambah modul.

Adapun beberapa ciri khas dari Apache, yaitu:

1. Apache sangat cepat dalam merespon *client* melebihi *server* NCSA.
2. Apache menyediakan fitur untuk *multihomed* dan *virtual server*.
3. Apache mempunyai level-level pengamanan.
4. Apache mempunyai komponen dasar terbanyak di antara *web server* lainnya. Sehingga termasuk *web server* terlengkap.
5. Performansi dan konsumsi sumber daya dari *web server* Apache tidak terlalu

banyak, yaitu sekitar 20 MB untuk file-file dasar dan setiap *daemon*-nya hanya memerlukan sekitar 950 kB memori per *child*.

6. Mendukung transaksi yang aman menggunakan SSL (*Secure Socket Layer*).
7. Banyak dukungan melalui *web*.
8. Kompatibilitas yang tinggi.

2.7 Ubuntu

Distro Ubuntu merupakan salah satu distro Linux yang banyak digunakan oleh masyarakat. Ubuntu adalah suatu sistem operasi bebas dan *open source* yang menggunakan Debian sebagai fondasinya dan dirilis secara berkala (setiap enam bulan), fokus utama sistem operasi Ubuntu adalah para pengguna dan kemudahan penggunaan (sesuai dengan "Just Work"TM) dan pada setiap rilis Ubuntu akan memberikan perbaikan keamanan selama 18 bulan.

Ubuntu menyertakan lingkungan *desktop* Gnome terbaru di setiap rilis dan juga menyertakan beragam pilihan perangkat lunak untuk *server* dan *desktop* yang semuanya dikemas ke dalam satu CD [AOM-07:01].

2.7.1 Tinjauan Pustaka Ubuntu

Ubuntu merupakan distro Linux yang didasarkan oleh distro Debian GNU/Linux. Ubuntu disponsori oleh Canonical Ltd. Distro Linux lainnya yang cukup populer digunakan adalah Fedora, openSUSE, dan Mandriva. Distro Fedora merupakan distro pengembangan dari distro Red Hat Linux. Sehingga pengguna juga bisa mendapatkan dukungan dari komunitas Red Hat Linux. Distro openSUSE merupakan distro pengembangan dari SUSE. Distro ini disponsori oleh Novell. OpenSUSE terkenal akan aplikasi *control panel* YaST. Mandriva dahulunya bernama Mandrake merupakan salah satu distro paling *user friendly* ketika Linux masih jarang dilirik oleh pengguna komputer. Mandrake terkenal akan konfigurasi grafik yang mudah digunakan dan kemampuan mengubah partisi NTFS [NOP-07:30-37].

Keunggulan distro Ubuntu dibandingkan dengan distro Linux lain, misalnya Fedora, openSuSE, atau Mandriva adalah:

1. Kemudahan dalam instalasi aplikasi [ANM-07:01].

Ubuntu menggunakan program Synaptics dalam menangani instalasi aplikasinya.

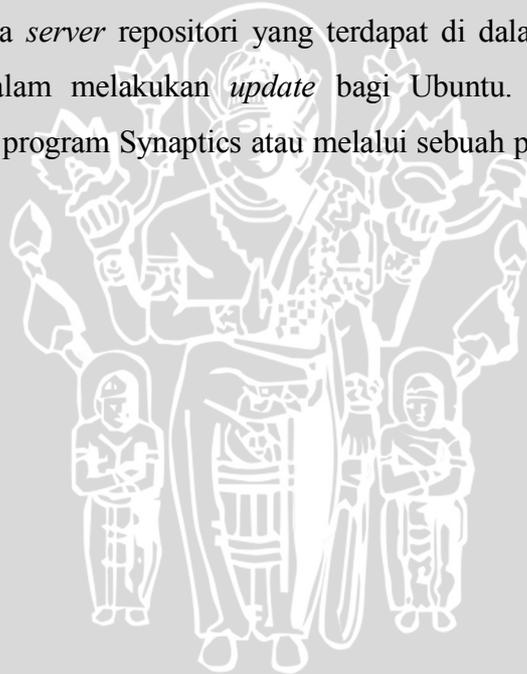
Program ini juga dapat meng-*update* aplikasi-aplikasi yang ada dalam Ubuntu melalui akses Internet. Selain melalui akses Internet, pengguna juga dapat melakukan instalasi melalui DVD Repositori yang disediakan oleh komunitas pengguna Ubuntu .

2. Kemudahan dalam akses hak *root* [AOI-07:01].

Akses *root* dalam Ubuntu menggunakan perintah *sudo*. Berbeda dengan dengan distro lain seperti Fedora atau Mandriva yang menggunakan perintah *su* yang membuat pengguna berada pada lingkungan *root*. Penggunaan perintah *sudo* bertujuan untuk meminimalisasi penggunaan lingkungan *root* dalam Ubuntu. Selain itu untuk mempermudah dalam administrasi sistem.

3. Kemudahan dalam *update* aplikasi maupun distro [ANU-07:01].

Dengan banyaknya *server* repositori yang terdapat di dalam negeri, maka akan mempermudah dalam melakukan *update* bagi Ubuntu. Proses *update* dapat dilakukan melalui program Synaptics atau melalui sebuah perintah *command line* sederhana.



BAB III

METODOLOGI PENELITIAN

Dalam mencapai tujuan dari penulisan tugas akhir, diperlukan adanya metodologi. Metodologi yang digunakan dalam penulisan tugas akhir antara lain adalah :

3.1 Studi Literatur

Untuk mendapatkan teori-teori pendukung serta pemecahan masalah yang dihadapi oleh penulis, maka diperlukan studi literatur. Literatur ini dapat berupa media cetak maupun elektronik. Studi literatur dalam penulisan tugas akhir ini meliputi:

1. Melakukan kajian pustaka mengenai basis data pada perangkat lunak osCommerce.
2. Melakukan kajian pustaka mengenai konfigurasi telepon selular yang digunakan sebagai modem SMS.
3. Melakukan kajian pustaka mengenai SMS Gateway Gnokii.
4. Melakukan kajian pustaka mengenai bahasa pemrograman Perl.
5. Melakukan kajian pustaka mengenai ekspresi reguler pada bahasa pemrograman Perl.
6. Melakukan kajian pustaka mengenai DBI Framework untuk menghubungkan Perl dengan basis data MySQL melalui DBI.

3.2 Perancangan Sistem

Untuk memudahkan dalam penulisan tugas akhir, maka diperlukan sebuah rancangan sistem. Dimana untuk memberikan solusi secara umum terhadap permasalahan yang dihadapi. Perancangan dilakukan dengan melakukan analisis terhadap sistem keseluruhan yang nantinya akan dibuat blok diagram secara keseluruhan. Perancangan dilakukan pada blok aplikasi yang berada antara SMS Gateway dengan aplikasi osCommerce. Perancangan juga dilakukan berdasarkan basis data yang terdapat pada aplikasi osCommerce. Perancangan didasarkan pada teori-teori yang telah ada yang kemudian akan diterapkan pada sistem yang akan dibuat.

3.3 Realisasi Sistem

Merupakan hasil dari semua penyelesaian masalah yang berupa program. Pada tahap ini diharapkan tujuan dari tugas akhir tercapai. Realisasi sistem merupakan pembuatan aplikasi toko *online* yang menggunakan osCommerce Version 2.2 Milestone 2 Update 051112 . Selain itu juga pembuatan SMS Gateway menggunakan Gnokii Version 0.6.14 . Kedua aplikasi tersebut menggunakan basis data MySQL 5.0.38. Basis data yang ada pada Gnokii akan disesuaikan agar bisa diterima oleh basis data osCommerce. Untuk itu diperlukan sebuah aplikasi antarmuka. Pembuatan aplikasi antarmuka menggunakan bahasa pemrograman Perl v5.8.8. Pembuatan aplikasi berasal dari perancangan yang telah dilakukan. Aplikasi yang telah dibuat akan dipasang di *server* bersama-sama dengan aplikasi lainnya. *Server* menggunakan sistem operasi GNU/Linux Ubuntu 7.04.

3.4 Pengujian dan Analisis Sistem

Tujuan dari pengujian adalah untuk memastikan bahwa sistem yang dibuat telah berjalan, yaitu dengan menguji sistem per blok dan keseluruhan sistem. Hasil yang diharapkan berupa suatu kesimpulan mengenai aplikasi antarmuka yang dibuat.

Pengujian selanjutnya adalah menerapkan sistem informasi tersebut dalam keadaan sesungguhnya. Pengujian dilakukan terhadap permasalahan yang mungkin dapat terjadi ketika proses transaksi berlangsung. Pengujian kemudian dilakukan terhadap waktu yang dibutuhkan hingga berhasil ketika transaksi menggunakan media SMS. Pengujian juga dilakukan dengan membandingkan transaksi pada kondisi menggunakan media Internet dengan menggunakan media SMS. Pengujian juga dilakukan terhadap hasil transaksi yang telah terjadi.

3.5 Pengambilan Kesimpulan dan Saran

Bab terakhir merupakan pengambilan kesimpulan dari aplikasi yang telah dibuat. Pengambilan kesimpulan ini berdasarkan kesesuaian antara teori dengan praktek yang telah didapat.

Bagian terakhir dari penulisan tugas akhir adalah saran yang bertujuan untuk memperbaiki kesalahan atau kekurangan yang telah terjadi selama penyusunan tugas akhir. Serta bermaksud untuk menyempurnakan penulisan tugas akhir.

BAB IV

PERANCANGAN

Dalam bab ini akan dijelaskan mengenai perancangan *Pengembangan Media Transaksi Menggunakan SMS (Short Message Service) Untuk Belanja Online* beserta prinsip kerja dari keseluruhan sistem. Perancangan sistem meliputi perancangan penggunaan perangkat keras dan perangkat lunak. Perancangan perangkat keras terdiri atas perancangan penggunaan SMS Gateway dengan ponsel yang berfungsi sebagai *modem*. Sedangkan perancangan perangkat lunak dibuat dengan menggunakan bahasa pemrograman skrip Perl.

4.1 Prinsip Kerja osCommerce

osCommerce merupakan suatu aplikasi administrasi untuk kegiatan *e-commerce*. Aplikasi ini dikembangkan menyerupai sebuah aplikasi CMS (*Content Management System*) sehingga mudah dalam pengelolaannya. Aplikasi ini merupakan program yang berupa *web-based*. Dikembangkan menggunakan bahasa PHP dan menggunakan MySQL untuk basis datanya.

Seperti pada aplikasi CMS kebanyakan, terdapat beberapa *user* yang dapat mengakses aplikasi ini. Terdapat tiga *user* pada aplikasi ini yaitu pengunjung, pelanggan, dan administrator. Pengunjung adalah *user* yang tidak perlu mendaftar untuk mengakses aplikasi ini. Mereka dapat melihat koleksi barang yang ada di katalog. Tetapi mereka tidak bisa melakukan pemesanan barang yang terdapat pada katalog. Untuk memesan barang, pengunjung harus mendaftar. Setelah mendaftar, maka *user* pengunjung berubah menjadi *user* pelanggan.

Pelanggan merupakan *user* yang telah mendaftarkan diri pada aplikasi ini. Pelanggan mempunyai hak seperti pengunjung. Hanya ia mempunyai kelebihan yaitu dapat melakukan pemesanan barang. Atau melakukan pembatalan atas pemesanan yang telah dilakukannya. Pendaftaran dilakukan langsung pada aplikasi. Kemudian ia harus mengecek *e-mail* untuk menerima konfirmasi pendaftarannya. Setelah mengkonfirmasi, ia bisa mulai melakukan transaksi barang yang terdapat pada aplikasi ini. Untuk mulai melakukan kegiatan transaksi ia harus *login* pada aplikasi terlebih dahulu. Apabila pelanggan tidak *login*, maka ia akan dianggap sebagai pengunjung biasa.

Yang terakhir adalah *user administrator*. Administrator adalah *user* yang mengelola aplikasi agar tetap berjalan. *User* ini mempunyai kewenangan tertinggi terhadap aplikasi, bahkan terhadap *user* di bawahnya. Administrator berwenang untuk membuat basis data yang akan digunakan pada aplikasi ini, hingga menentukan *banner* yang akan ditampilkan. Administrator juga berwenang untuk menghapus *user* yang telah mendaftar pada aplikasi ini. Dapat diandaikan administrator sebagai pemilik sekaligus pengelola toko. Ia yang menggunakan aplikasi osCommerce sebagai administrasi tokonya dan ia juga yang mengelola toko tersebut melalui aplikasi osCommerce.

4.2 Transaksi Barang Pada osCommerce

Untuk melakukan transaksi pada osCommerce seperti pemesanan, seorang pengunjung harus melakukan registrasi terlebih dahulu. Registrasi dilakukan melalui menu MyAccount yang terdapat pada deretan menu pada aplikasi osCommerce. Setelah melakukan registrasi, pelanggan dapat mulai melakukan kegiatan transaksi. Pelanggan dapat memilih barang-barang yang diperlihatkan dalam katalog. Di dalam katalog terdapat informasi mengenai barang yang dijual. Barang yang telah dipilih kemudian akan dimasukkan ke dalam sebuah 'keranjang'. Istilah 'keranjang' merupakan kata lain dari penggunaan sebuah penyimpanan data sementara untuk barang-barang yang akan dibeli. Di keranjang inilah terdapat barang-barang yang berasal dari katalog yang telah dipilih oleh pelanggan untuk dibeli. Barang-barang yang telah dimasukkan ke dalam keranjang dapat dilihat pada menu Cart Contents.

Pada menu Cart Contents pelanggan dapat mengubah isi dari keranjang. Setelah yakin dengan barang yang akan dibeli, pelanggan dapat memilih menu Checkout. Setelah menu memilih menu Checkout, maka pelanggan akan diperlihatkan keranjang yang berisi barang-barang yang telah dipilih sebelumnya. Setelah setuju dengan pilihannya, maka pelanggan memilih proceed untuk membuka halaman pembayaran. Setelah setuju dengan pembayaran yang akan dilakukan, maka halaman pengiriman terbuka. Disini terdapat alamat pengiriman. Alamat inilah tujuan dimana paket akan dikirim. Setelah setuju dengan alamat pengiriman, maka pelanggan akan dibawa ke halaman awal dari aplikasi osCommerce. Pelanggan dapat memilih menu Logout untuk keluar dari osCommerce. Atau dapat kembali melakukan transaksi barang.

4.3 Basis Data Pada osCommerce

Basis data pada osCommerce merupakan tempat menyimpan segala informasi yang ada pada aplikasi ini. Dari mulai informasi *user* hingga mengenai *banner* iklan. Informasi-informasi tersebut disimpan ke dalam banyak tabel. Basis data yang digunakan untuk aplikasi osCommerce telah disediakan pada *file* instalasinya. Pada *file* tersebut telah dibuatkan tabel-tabel yang diperlukan untuk membuat sebuah aplikasi toko *online*. Ada sebanyak 37 tabel yang berfungsi menyimpan informasi-informasi tersebut. Dari kesemuanya, terbagi menjadi 21 kelompok. Pengelompokan ini berdasarkan tujuan dari dibuatnya tabel-tabel tersebut.

Tabel 4.1. Daftar Tabel pada Basis Data yang Digunakan oleh osCommerce.

Kelompok	Nama Tabel
Order	orders
	order_products
	orders_products_attributes
Pelanggan	customer
	customer_info
	customers_basket
	customers_basket_attributes
Produk	products
	products_to_categories
Atribut Produk	products_attributes
	products_option
	products_option_value
	products_option_value_to_products_option
Bahasa	languages
Status Order	orders_status
Buku Alamat	address_book
Konfigurasi	configuration
	configuration_group
Iklan	banners
	banners_id
Sesi	sessions
Who's Online	whos_online

Counter	counter
Counter History	counter_history
Mata Uang	currencies
Barang Spesial	specials
Manufaktur	manufacturers
	manufactures_info
Negara	countries
Format Alamat	address_format
Zona	zones
	geo_zones
	zones_to_geo_zones
Pajak	tax_rates
	tax_class
Review	reviews
	reviews_id

Sumber: [ANM-06:01]

Ada beberapa kelompok tabel yang berhubungan langsung dengan transaksi osCommerce. Mereka adalah kelompok Order, Pelanggan, dan Produk. Sedangkan kelompok yang berhubungan tidak langsung dengan transaksi adalah Atribut Produk, Buku Alamat, Manufaktur, Negara, Format Alamat, Zona, Pajak, Status Order. Sedangkan kelompok yang tidak berhubungan dengan transaksi adalah Bahasa, Konfigurasi, Iklan, Sesi, Who's Online, Counter, Mata Uang, Barang Spesial, dan Review.

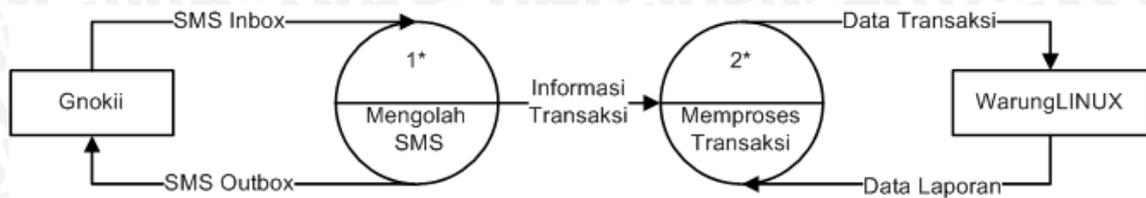
4.4 Diagram Aliran Data pada Aplikasi Antarmuka

Diagram Aliran Data (*Data Flow Diagram*) digunakan untuk membagi suatu sistem yang rumit dalam perencanaan pembuatan sebuah aplikasi menjadi beberapa sub sistem yang lebih kecil. *Data Flow Diagram* (DFD) dibagi-bagi menjadi beberapa level. Setiap level dijelaskan oleh level berikutnya. Salah satu keuntungan penggunaan DFD adalah membantu memudahkan pemakai untuk mengerti sistem yang akan dikerjakan.

4.4.1 Diagram Aliran Data Level 0

DFD Level 0 Merupakan penggambaran dari sistem secara keseluruhan. DFD Level 0 tidak menjelaskan secara detail dari proses yang terjadi. Proses itu akan

dijelaskan secara bertahap pada level selanjutnya. DFD level 0 dijelaskan dalam Gambar 4.1.



Gambar 4.1. Diagram Aliran Data Level 0.
Sumber: [Perancangan]

Aplikasi antarmuka pertama kali akan membaca informasi yang berupa SMS dari aplikasi Gnokii. Isi dari SMS ini berupa *username* dan *password* milik pelanggan WarungLINUX. Selain itu juga terdapat data transaksi yang akan diolah oleh WarungLINUX.

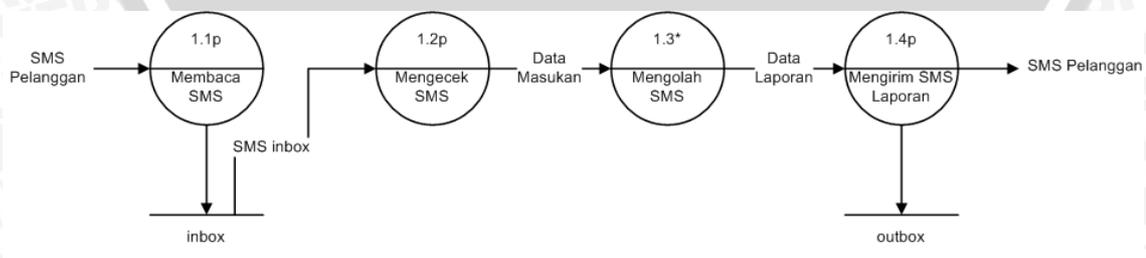
Proses pengolahan SMS dijelaskan lebih lanjut pada DFD level 1. Ketika SMS berhasil diolah, maka akan dilakukan proses lebih lanjut. Pemrosesan tersebut adalah memproses data transaksi yang terdapat pada SMS. Setelah proses ini berlangsung, maka akan diberikan laporan ke dalam SMS.

4.4.2 Diagram Aliran Data Level 1

DFD level 0 yang dijelaskan pada sub bab diatas dapat dijelaskan lebih detail menjadi beberapa DFD level 1, yang berupa:

1. DFD Level 1 – Pemrosesan SMS

DFD level 1 untuk Pemrosesan SMS ditunjukkan dalam Gambar 4.2.



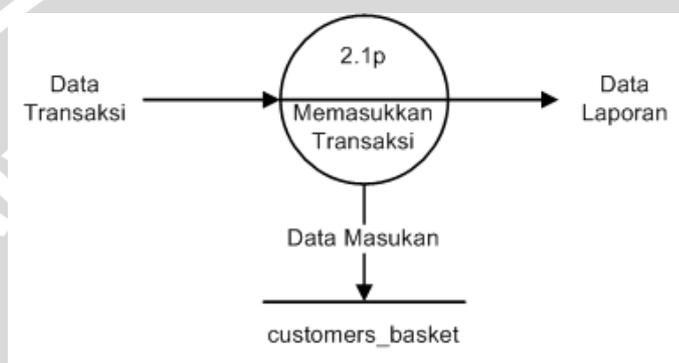
Gambar 4.2. Diagram Aliran Data Level 1 – Pemrosesan SMS.
Sumber: [Perancangan]

Gambar diatas menjelaskan mengenai proses yang dilakukan oleh aplikasi antarmuka terhadap SMS yang diterima. Pertama kali aplikasi akan membaca SMS yang

berada pada tabel `inbox` pada basis data milik Gnokii. Setelah itu aplikasi akan mengecek SMS dengan mencocokkannya dengan format regex yang telah ditentukan sebelumnya. Apabila valid, maka aplikasi akan mengolah SMS tersebut. Ketika SMS berhasil diolah, maka proses berikutnya adalah mengirim laporan ke pelanggan dengan memasukkan SMS ke dalam tabel `outbox` milik Gnokii.

2. DFD Level 1 – Pemrosesan Transaksi

DFD level 1 untuk Pemrosesan Pelanggan diperlihatkan dalam Gambar 4.3.



Gambar 4.3. Diagram Aliran Data Level 1 – Pemrosesan Transaksi.
Sumber: [Perancangan]

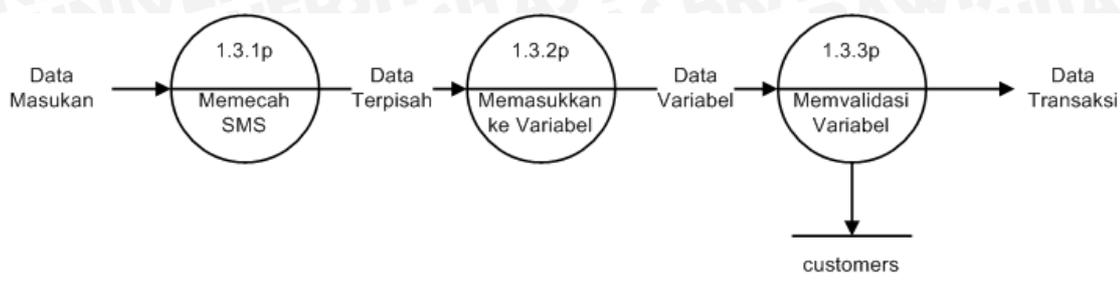
Proses dalam Gambar 4.3 menunjukkan mengenai proses dalam menangani data masukan. Data masukan yang berasal dari SMS berisi data mengenai barang yang akan dipesan oleh pelanggan. Data ini akan dimasukkan ke dalam tabel `customers_basket` milik aplikasi WarungLINUX. Setelah data dimasukkan ke dalam tabel, maka akan diberikan laporan dalam bentuk SMS.

4.4.3 Diagram Aliran Data Level 2

DFD level 1 dapat dijabarkan lebih detail menjadi DFD level 2. Hal ini dijelaskan sebagai berikut:

1. DFD Level 2 – Pemrosesan Data SMS

DFD Level 2 untuk Pemrosesan Data SMS diperlihatkan dalam Gambar 4.4.



Gambar 4.4. Diagram Aliran Data Level 2 – Pemrosesan Data SMS.
Sumber: [Perancangan]

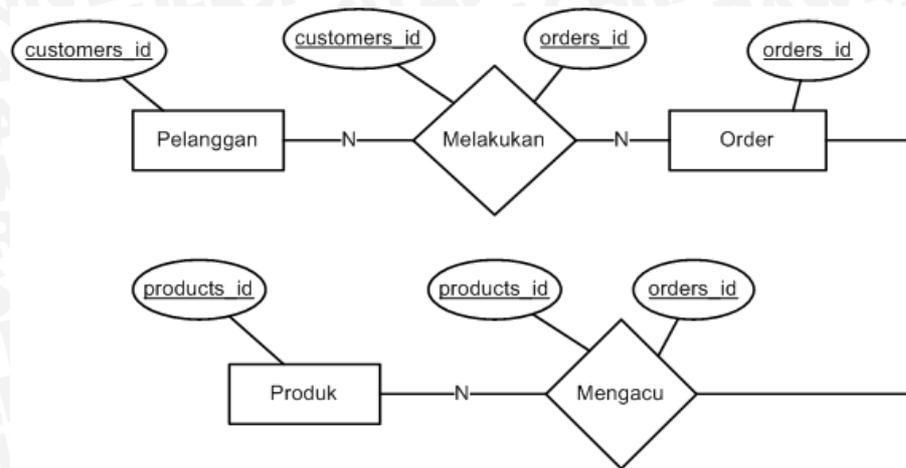
Dalam gambar diatas diperlihatkan mengenai pemrosesan SMS yang diterima. Data masukan yang berupa SMS pertama kali dipecah berdasarkan format regex yang telah ditentukan sebelumnya. Setelah terpisah, maka dimasukkan ke dalam variabel-variabel. Variabel-variabel ini kemudian divalidasi terhadap tabel *customers* milik WarungLINUX. Setelah divalidasi, maka variabel-variabel tersebut akan menjadi data transaksi bagi aplikasi WarungLINUX.

4.5 Diagram E-R (*Entity-Relationship*) Transaksi Barang pada osCommerce

Diagram E-R digunakan untuk membantu dalam perancangan basis data untuk osCommerce. Diagram tersebut menggambarkan entitas-entitas secara umum yang ada pada osCommerce. Sehingga perancangan basis data menggunakan diagram E-R tidak mewakili sepenuhnya terhadap struktur nyata dari sistem yang akan dikembangkan.

Diagram E-R seperti yang diperlihatkan dalam Gambar 4.5 secara umum menggambarkan hubungan antar entitas satu dengan yang lainnya. Dengan memperlihatkan hubungan antara atribut yang dijadikan *key* maka bisa terlihat hubungan diantaranya.

Setiap tabel pada osCommerce mempunyai atribut kunci yang memiliki ketergantungan fungsional terhadap yang lainnya. Misalnya pada tabel Pelanggan dengan *customers_id* sebagai *primary key*. Maka atribut *customers_id* mempunyai ketergantungan fungsional terhadap nama, alamat *e-mail*, jenis kelamin, *password*, dan lain-lain. Penggunaan *customers_id* sebagai *primary key* karena diperlukan suatu *key* yang berfungsi untuk membedakan pelanggan satu dengan yang lainnya, sehingga terjamin keakuratan datanya.

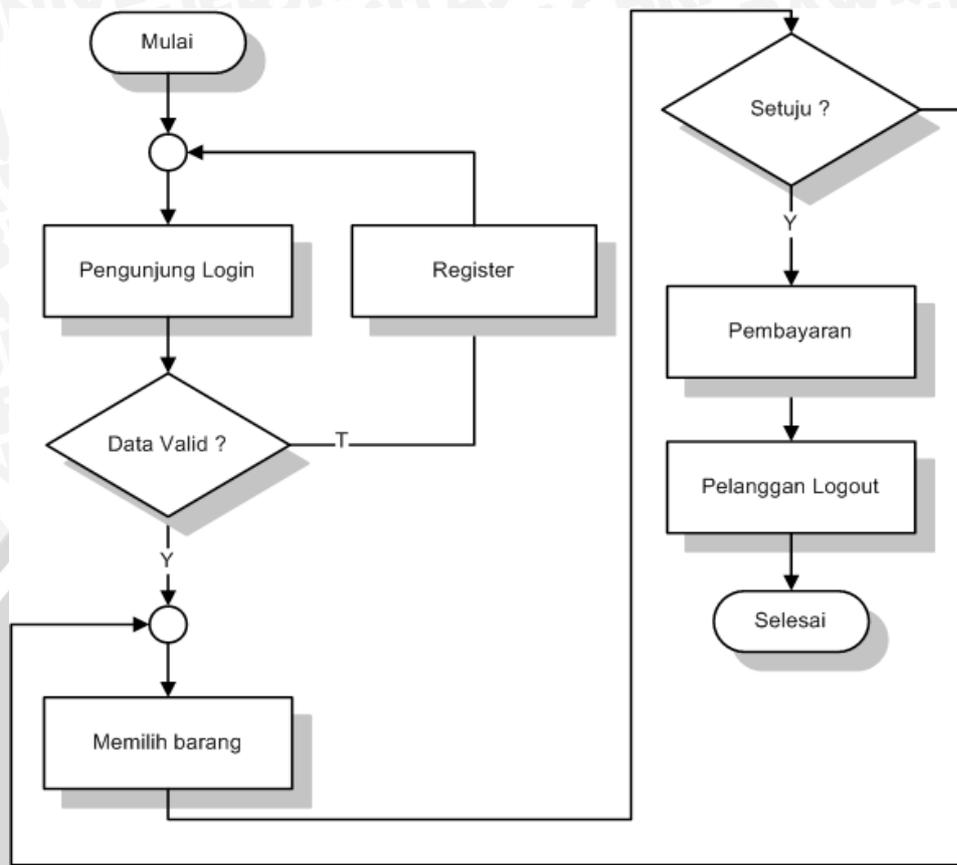


Gambar 4.5. Diagram E-R Pada osCommerce.
 Sumber: [Perancangan]

4.6 Diagram Alir Transaksi Barang Pada osCommerce

Diagram alir di bawah menunjukkan secara garis besar transaksi yang ada pada aplikasi osCommerce. Dimulai dengan *login* pengunjung menjadi pelanggan. Jika data *login* tidak valid, berarti pengunjung belum mendaftarkan dirinya. Pengunjung kemudian mendaftarkan dirinya agar *login* berhasil. Jika *login* berhasil, maka pengunjung diubah statusnya sebagai pelanggan. Kemudian pelanggan dapat memilih barang yang diinginkan. Jika pelanggan telah setuju dengan barang yang telah ia pilih, pelanggan akan dibawa ke menu untuk melakukan pembayaran. Setelah pelanggan berhasil melakukan pembayaran, maka pelanggan dapat memilih kembali barang yang diinginkan. Terakhir, pelanggan *logout* untuk keluar dari aplikasi osCommerce.





Gambar 4.6. Diagram Alir Transaksi Barang Aplikasi osCommerce.
Sumber: [Perancangan]

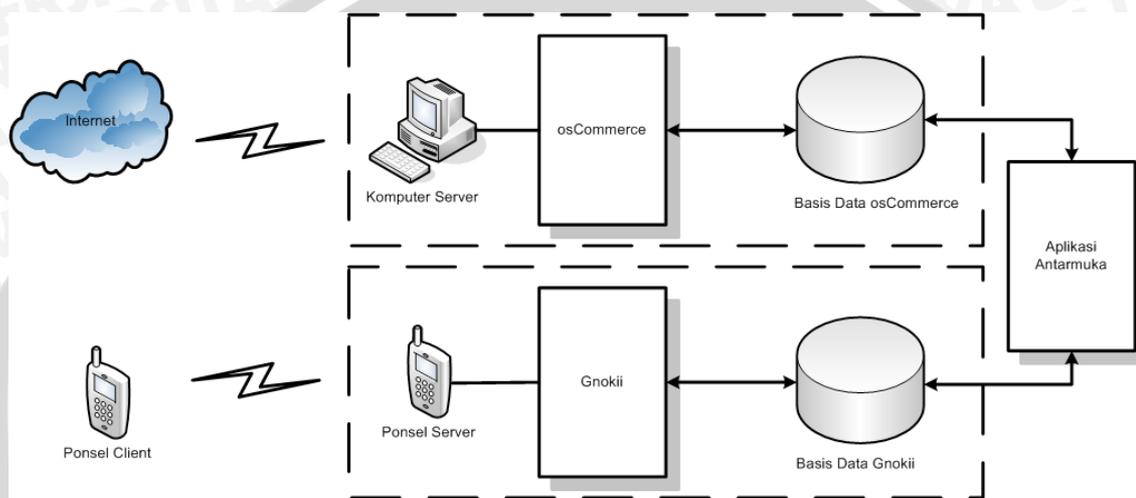
4.7 Prinsip Kerja Pengembangan Sistem Transaksi

Pada prinsipnya kerja dari sistem yang ada adalah transaksi pembelian dengan menggunakan media Internet. Untuk itu dikembangkan dengan menambahkan media SMS. Ini bertujuan untuk memperluas area transaksi barang yang telah ada. Dengan ditambahkannya media SMS diharapkan transaksi tidak tergantung lagi terhadap akses Internet yang belum mencakup daerah yang cukup terpencil.

Pembeli dapat menggunakan SMS untuk melakukan transaksi pembelian barang. Hal ini dilakukan dengan cara mengirimkan SMS ke nomor yang telah ditentukan oleh penjual. SMS yang dikirim menggunakan format tertentu yang telah ditentukan sebelumnya. SMS yang telah masuk akan diolah oleh aplikasi antarmuka. SMS ini nantinya akan diubah menjadi sebuah *query* basis data. *Query* basis data ini akan dimasukkan ke dalam basis data milik aplikasi osCommerce. Sehingga tidak ada bedanya transaksi menggunakan media Internet maupun media SMS. Setelah transaksi telah berhasil dilakukan, aplikasi antarmuka akan mengirim SMS kepada pembeli untuk memberikan laporan bahwa transaksi telah berhasil dilakukan.

4.8 Diagram Blok Pengembangan Sistem Transaksi

Blok diagram rangkaian merupakan salah satu bagian terpenting dalam perancangan sistem ini, karena dapat diketahui prinsip kerja keseluruhan aplikasi. Keseluruhan blok diagram sistem tersebut menunjukkan gambaran umum sistem yang dapat difungsikan atau sistem yang bekerja sesuai dengan perancangan. Blok diagram sistem transaksi secara keseluruhan dapat dilihat dalam Gambar 4.7.



Gambar 4.7. Diagram Blok Pengembangan Sistem Transaksi.
Sumber: [Perancangan]

Pada sistem informasi awal, media transaksi hanya menggunakan Internet. Dimana pengolahan transaksi menggunakan aplikasi osCommerce. Data transaksi yang dilakukan oleh pelanggan dimasukkan ke dalam basis data milik aplikasi osCommerce. Untuk itu dikembangkan suatu media alternatif dengan menggunakan SMS. SMS yang dikirim oleh pelanggan akan diterima melalui SMS *Gateway* menggunakan aplikasi Gnokii. SMS yang diterima oleh Gnokii disimpan ke dalam basis data milik aplikasi Gnokii. Aplikasi antarmuka berguna untuk mengubah data SMS yang berada pada aplikasi Gnokii ke dalam basis data pada aplikasi osCommerce. Bahasa pemrograman Perl digunakan untuk pembuatan aplikasi antarmuka ini. Ketika transaksi berhasil diolah oleh aplikasi antarmuka, maka aplikasi antarmuka akan mengirim laporan bahwa transaksi berhasil dilakukan ke ponsel pelanggan melalui SMS.

4.9 Perancangan Isi osCommerce

Toko *online* pada tugas akhir menggunakan WarungLINUX sebagai nama toko. Tujuan WarungLINUX adalah menawarkan distro-distro Linux versi terakhir yang

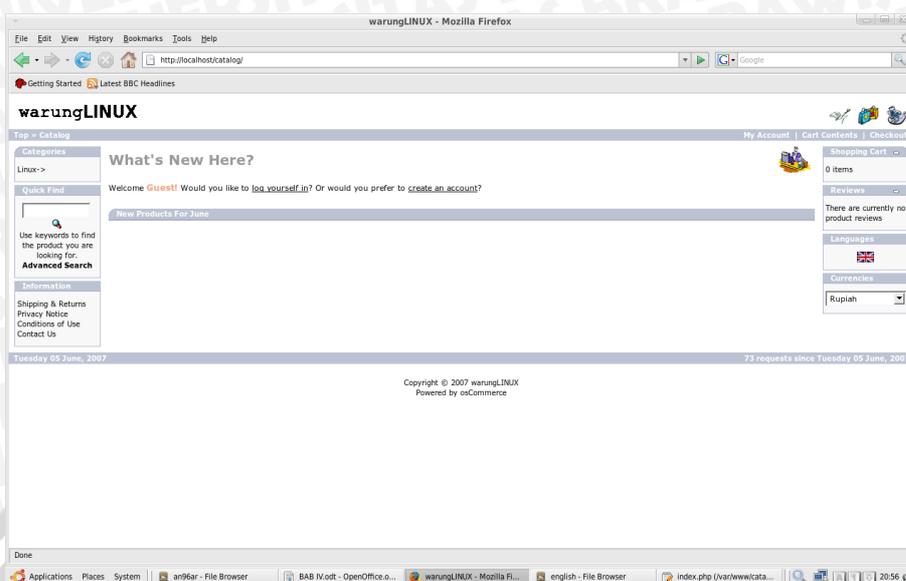
tersedia. Barang-barang yang terdapat pada WarungLINUX merupakan distro Linux peringkat 10 tertinggi yang terdapat pada website www.distrowatch.com pada bulan Mei 2007.

Tabel 4.2. Daftar Peringkat 10 Besar Distro Menurut www.distrowatch.com pada Mei 2007.

Peringkat	Nama Distro	Alamat Situs
1	Ubuntu	http://www.ubuntu.com/
2	openSUSE	http://www.opensuse.org/
3	PCLinuxOS	http://www.pclinuxos.com/
4	Fedora	http://fedoraproject.org/
5	MEPIS	http://www.mepis.org/
6	Debian	http://www.debian.org
7	Sabayon	http://www.sabayonlinux.org/
8	Mandriva	http://www.mandrivalinux.com/
9	Linux Mint	http://linuxmint.com/
10	Damn Small Linux	http://www.damnsmalllinux.org/

Sumber: [www.distrowatch.com]

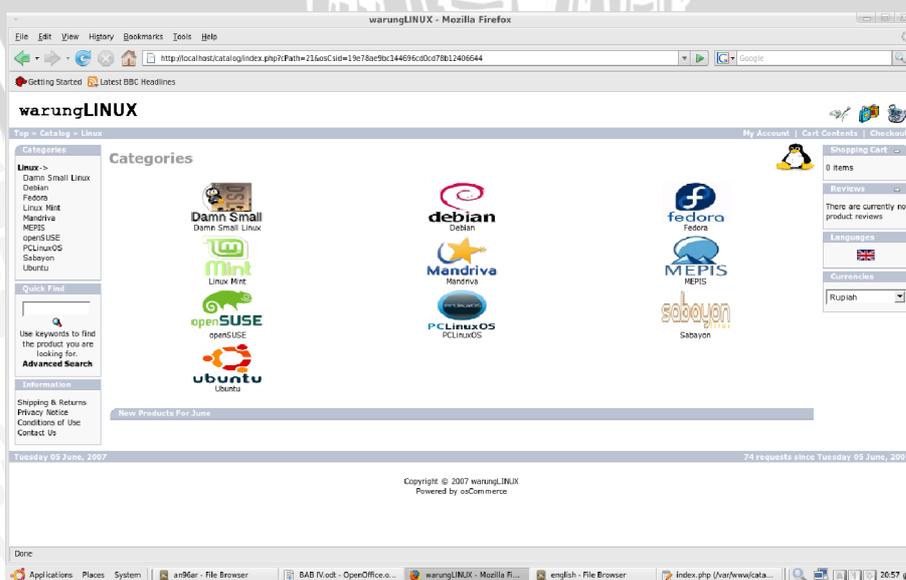
Distro-distro tersebut dipaketkan ke dalam media CD dan DVD. Pemilihan media DVD didasarkan pada kenyataan merebaknya penggunaan DVD-ROM oleh pengguna komputer di Indonesia. Media DVD juga hemat tempat. Dibandingkan dengan media CD, maka 5 media CD dapat dimuat ke dalam sebuah media DVD. Sehingga diharapkan dapat meminimalisasi biaya pengiriman yang didasarkan pada berat yang dilakukan oleh jasa pengiriman.



Gambar 4.8. Halaman Depan WarungLINUX.
Sumber: [Perancangan]

4.9.1 Perancangan Barang

Dengan banyaknya barang yang ditawarkan oleh WarungLINUX, maka perlu dibuat pemilahan. Pemilahan ini bertujuan untuk memudahkan pencarian barang yang diinginkan oleh pengunjung. Pemisahan kategori didasarkan pada nama distro tersebut. Jika sebuah distro mempunyai varian yang berbeda, maka akan dimasukkan ke dalam kategori baru. Apabila suatu distro tersedia dalam beberapa versi media, maka akan dimasukkan dalam kategori yang sama. Perbedaan kategori dalam katalog hanya pada nama distro.



Gambar 4.9. Katalog Barang WarungLINUX.
Sumber: [Perancangan]



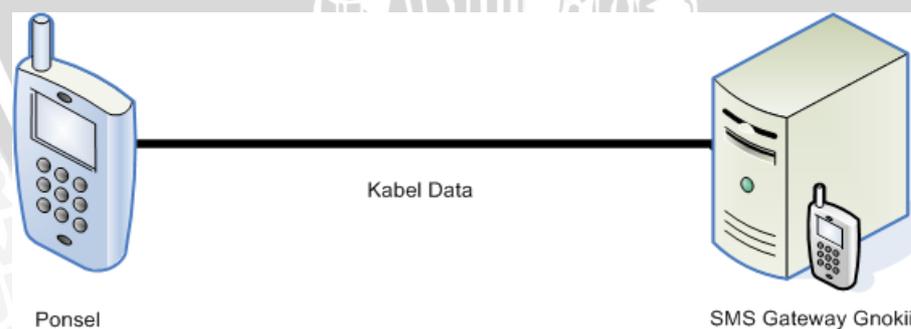
4.9.2 Perancangan Jenis Pembayaran

Pembayaran yang dapat diterima oleh WarungLINUX adalah melalui COD (*Cash On Delivery*) bila pelanggan berada pada kota yang sama dengan penjual atau transfer melalui rekening bank bila pelanggan berada di lain kota dengan penjual. Kartu kredit tidak digunakan sebagai bentuk pembayaran dalam aplikasi ini. Hal ini disebabkan belum meluasnya penggunaan kartu kredit sebagai pembayaran *online* di Indonesia. Selain itu juga untuk mengurangi resiko dari kejahatan yang terjadi melalui pencurian kartu kredit (*carding*). Meski pembayaran melalui rekening merupakan pembayaran yang lebih membutuhkan waktu dan tenaga, sampai saat ini masih merupakan pembayaran yang paling aman untuk transaksi *online*.

Rekening bank yang diusahakan tersedia untuk pembayaran merupakan rekening dari bank-bank besar yang ada di Indonesia. Terutama bank-bank yang mempunyai fasilitas ATM dengan jumlah banyak dan dapat dijangkau dengan mudah. Hal ini untuk memudahkan apabila pelanggan akan melakukan pembayaran ATM. Sehingga pelanggan tidak perlu melakukan transfer antar bank.

4.10 Perancangan SMS Gateway

Aplikasi SMS Gateway yang digunakan dalam tugas akhir ini adalah Gnokii. Gnokii dipilih karena kemudahan pengoperasiannya. Selain itu Gnokii sangat kompatibel dengan ponsel merek Nokia.



Gambar 4.10. Diagram Perangkat Keras Pada Gnokii.
Sumber: [Perancangan]

Tugas akhir menggunakan ponsel Nokia seri 3100 yang berfungsi sebagai *modem* yang akan menerima dan mengirim SMS. Ponsel ini dihubungkan melalui sebuah kabel data seri DKU-5. DKU-5 menggunakan port USB (*Universal Serial Bus*) sebagai konektor pada komputer.



Gambar 4.11. Kabel Data DKU-5.
Sumber: [ANO-07]

Selain aplikasi inti Gnokii, terdapat juga *daemon* dengan nama `gnokiid`. `gnokiid` yang dijalankan akan menampilkan semua SMS yang masuk ke dalam layar monitor. Namun sayangnya SMS-SMS itu tidak disimpan.

Karena itu Gnokii menyediakan sebuah *daemon* yang untuk menangani SMS yang masuk, yaitu `smsd`. `smsd` tidak hanya menampilkan SMS yang masuk, tetapi menyimpannya ke dalam sebuah basis data. Basis data yang digunakan adalah basis data MySQL atau PostgreSQL. SMS tersebut juga dapat disimpan ke dalam sebuah *file* teks.

Pada tugas akhir ini, basis data yang digunakan untuk menyimpan SMS adalah MySQL. Hal ini dengan pertimbangan kepraktisan karena aplikasi osCommerce juga menggunakan MySQL sebagai basis datanya. Agar `smsd` dapat berjalan dengan lancar, maka dibutuhkan dua tabel dalam sebuah basis data, dengan struktur yang telah ditentukan oleh `smsd` seperti yang ditunjukkan dalam Tabel 4.3 dan 4.4. *Default* nama basis data yang digunakan adalah `sms`. Pada basis data ini akan terisi dua buah tabel yang mempunyai nama `inbox` dan `outbox`. Tabel `inbox` digunakan `smsd` untuk menyimpan SMS yang masuk ke dalam ponsel, sedangkan tabel `outbox` digunakan untuk menyimpan semua SMS yang akan atau telah dikirimkan ke nomor tujuan. Selain itu, tabel `outbox` juga digunakan untuk validasi proses pengiriman SMS, artinya SMS dapat dikirimkan hanya dengan memasukkan data ke dalam tabel `outbox` tersebut. Format pengiriman SMS melalui tabel `outbox` adalah:

```
MySQL> INSERT INTO outbox(number, text)
```

Dimana `number` adalah field untuk nomor tujuan, sedangkan `text` adalah isi pesan yang akan dikirim.

Tabel 4.3. Struktur Tabel `inbox` pada Basis Data SMS.

Nama Kolom	Tipe Data	Lebar
<code>id</code>	<code>integer</code>	10
<code>number</code>	<code>varchar</code>	20
<code>smsdate</code>	<code>datetime</code>	
<code>insertdate</code>	<code>timestamp</code>	14
<code>text</code>	<code>varchar</code>	160
<code>phone</code>	<code>tinyint</code>	4
<code>processed</code>	<code>tinyint</code>	4

Sumber: [Perancangan]

Tabel 4.4. Struktur Tabel `outbox` pada Basis Data SMS.

Nama Kolom	Tipe Data	Lebar
<code>id</code>	<code>int</code>	10
<code>number</code>	<code>varchar</code>	20
<code>processed_date</code>	<code>timestamp</code>	14
<code>insertdate</code>	<code>timestamp</code>	14
<code>text</code>	<code>varchar</code>	160
<code>phone</code>	<code>tinyint</code>	4
<code>processed</code>	<code>tinyint</code>	4
<code>error</code>	<code>tinyint</code>	4
<code>dreport</code>	<code>tinyint</code>	4
<code>not_before</code>	<code>time</code>	
<code>not_after</code>	<code>time</code>	

Sumber: [Perancangan]

4.10.1 Perancangan Format Masukan SMS Gateway

Masukan yang berupa SMS sangat berbeda dibanding dengan masukan yang biasa terdapat pada program-program komputer. Karena SMS terbatas hanya 160 karakter, maka diperlukan suatu masukan yang dapat memuat masukan-masukan yang diperlukan untuk menyerupai masukan untuk sebuah transaksi pada aplikasi biasa. Masukan ini juga

tidak boleh mempunyai bentuk yang panjang, karena akan mengurangi alokasi karakter yang tersedia.

Berdasarkan masukan diatas, maka SMS masukan berupa suatu deretan kata yang mempunyai format:

```
#user_name #password *id_barang =jumlah_barang
```

Dimana `user_name` adalah user name dari pelanggan yang telah mendaftarkan di WarungLINUX. Sedangkan `password` adalah password dari user name dari pelanggan yang juga telah mendaftarkan diri. `id_barang` merupakan nomor identitas yang ada pada setiap produk yang ada pada katalog WarungLINUX. `jumlah` merupakan banyaknya paket yang dipilih yang sesuai dengan `id_barang` pada kata sebelumnya.

Untuk pemilihan dengan lebih dari satu jenis barang, maka format yang digunakan merupakan format yang sama, dengan penambahan diakhir dengan `*id_barang =jumlah_barang`. Sehingga format masukan SMS akan menjadi:

```
#user_name #password *id_barang =jumlah_barang *id_barang  
=jumlah_barang
```

Begitu seterusnya, dengan penambahan kata-kata `*id_barang =jumlah_barang` untuk setiap jenis barang berbeda berikutnya.

4.11 Perancangan Aplikasi Antarmuka

Aplikasi antarmuka merupakan suatu program yang bertujuan untuk mengubah SMS yang dikirim oleh pelanggan agar bisa dimasukkan ke dalam basis data osCommerce. Aplikasi dikembangkan dengan menggunakan bahasa pemrograman Perl. Implementasi aplikasi ini berupa perangkat lunak yang dapat diterapkan pada aplikasi osCommerce.

Aplikasi antarmuka menggunakan basis data MySQL sebagai masukannya yaitu basis data milik aplikasi Gnokii. Data yang ada pada basis data Gnokii diolah sehingga dapat dimasukkan ke dalam basis data yang dimiliki oleh osCommerce, yang juga menggunakan MySQL sebagai basis datanya. Aplikasi antarmuka menggunakan Framework DBI sebagai media untuk melakukan koneksi dengan basis data MySQL yang terdapat pada sistem operasi.

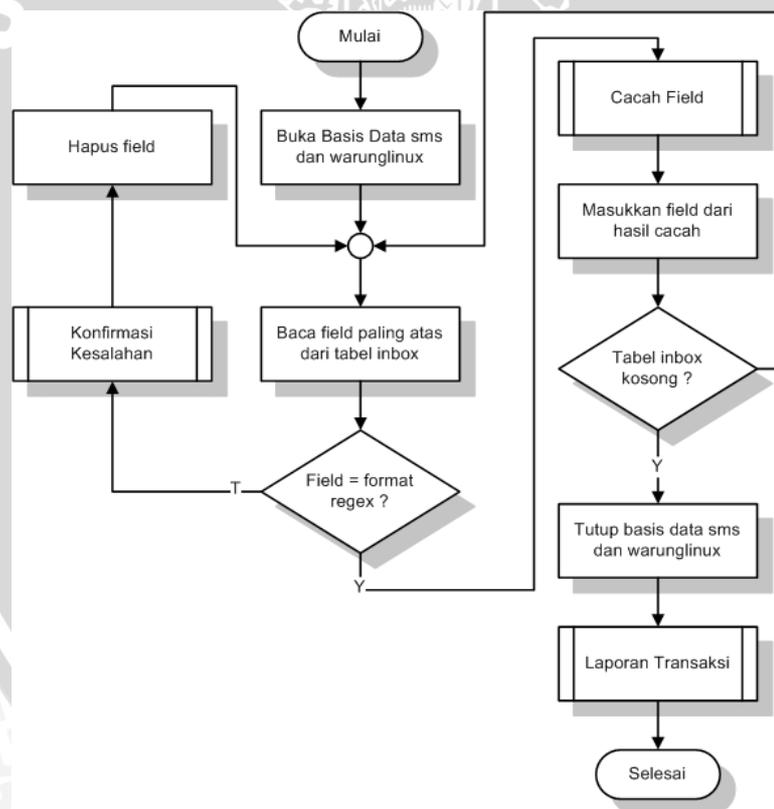
Aplikasi antarmuka menggunakan basis data `sms` dan `warunglinux` dalam kerjanya. Basis data `sms` berisi informasi mengenai SMS yang telah diterima dan yang akan dikirim maupun yang telah dikirim. Sedangkan basis data `warunglinux` merupakan basis data yang digunakan oleh osCommerce dalam menampung data. Isi dari basis data

sms akan dimasukkan ke dalam basis data warunlinux. Tentu saja melalui beberapa proses terlebih dahulu. Karena format yang ada pada basis data sms sangat berbeda dengan basis data warunlinux. Untuk itu digunakan pemrograman regex.

Pemrograman regex menggunakan bahasa pemrograman Perl untuk melakukannya. Perl akan mencacah isi SMS sehingga dapat diterima oleh basis data warunlinux. Diharapkan *field* yang dimasukkan melalui aplikasi antarmuka akan sama dengan *field* yang dimasukkan oleh osCommerce.

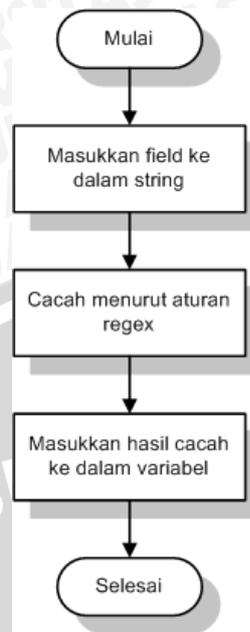
4.11.1 Diagram Alir Aplikasi Antarmuka

Diagram alir membantu untuk memudahkan dalam perancangan dan pembuatan aplikasi antarmuka. Pada diagram alir dibawah terdapat 3 fungsi, yaitu fungsi cacah *field*, fungsi konfirmasi kesalahan, dan fungsi laporan transaksi.

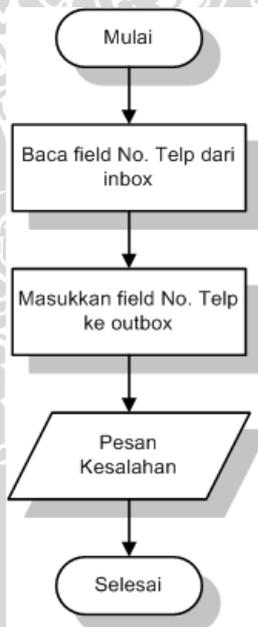


Gambar 4.12. Diagram Alir Aplikasi Antarmuka.

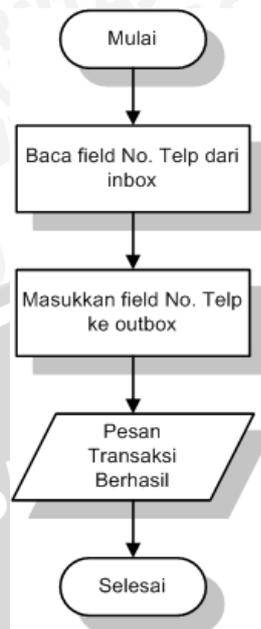
Sumber: [Perancangan]



Gambar 4.13. Fungsi Cacah *Field*.
 Sumber: [Perancangan]



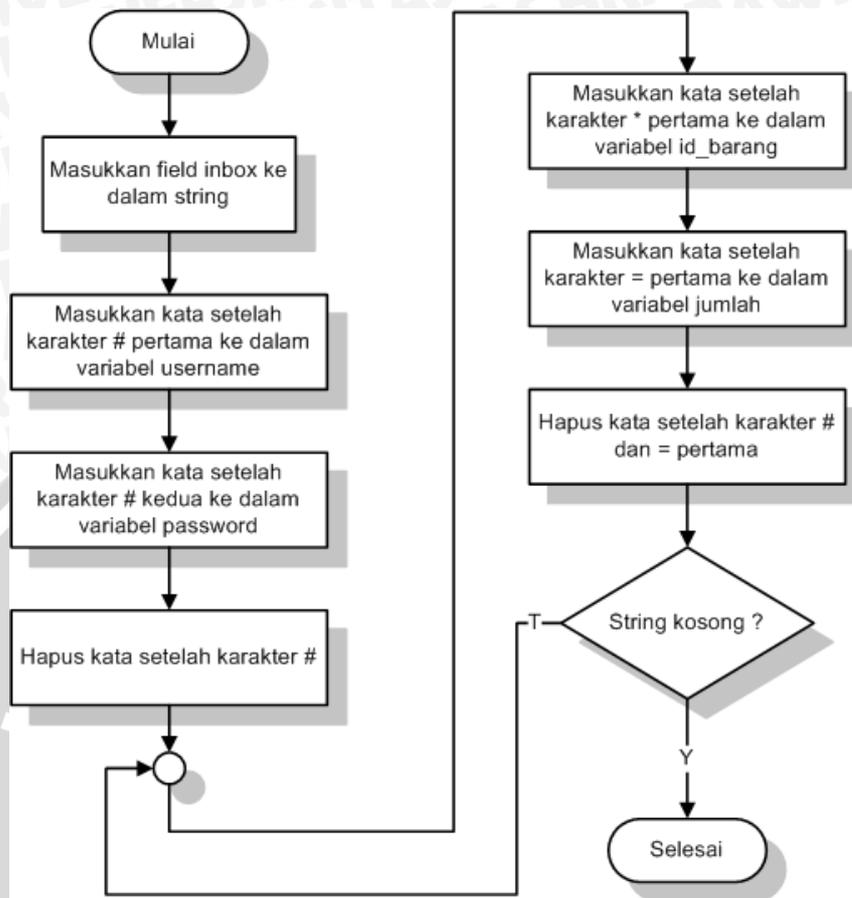
Gambar 4.14. Fungsi Konfirmasi Kesalahan.
 Sumber: [Perancangan]



Gambar 4.15. Fungsi Laporan Transaksi.
Sumber: [Perancangan]

4.11.2 Perancangan Pengubahan SMS ke *Query* Basis Data

Hasil keluaran dari Gnokii adalah berupa SMS yang tersimpan dalam basis data. SMS ini mempunyai format seperti yang telah ditentukan pada sub-bab diatas. SMS tersebut tidak bisa langsung dimasukkan ke dalam basis data milik osCommerce, karena dalam SMS tersebut terdapat beberapa informasi yang akan dimasukkan ke dalam basis data osCommerce. Untuk itu diperlukan pengubahan format dari bentuk SMS yang telah ditentukan sebelumnya menjadi sebuah *query* basis data yang akan dimasukkan ke dalam beberapa *field* basis data osCommerce. Pengubahan SMS ini menggunakan bahasa pemrograman Perl dengan menggunakan ekspresi reguler (regex). Rumusan regex yang digunakan untuk mencacah SMS menjadi beberapa bagian diperlihatkan dalam Gambar 4.16.



Gambar 4.16. Diagram Alir Pencacahan SMS.
Sumber: [Perancangan]

4.11.3 Perancangan Koneksi Basis Data

Koneksi pada basis data pada bahasa Perl menggunakan DBI. Listing dari pengaksesan basis data adalah:

```

use DBI;
$db_handle=DBI->connect(dbi:$data_source,$user,$password)
||die$DBI::errstr;

```

Dalam hal ini:

- \$data_source adalah nama basis data.
- \$user adalah user yang diperbolehkan mengakses basis data.
- \$password adalah kata kunci milik user yang akan mengakses basis data.

4.11.4 Perancangan Laporan Hasil Transaksi

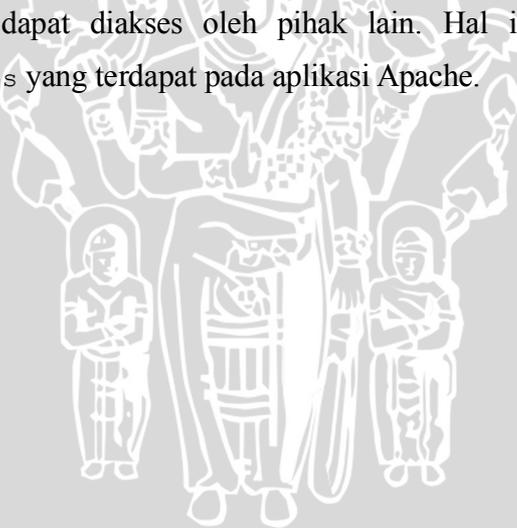
Hasil transaksi dengan menggunakan media SMS akan dilaporkan menggunakan media SMS. Laporan yang dikirimkan dapat terbagi menjadi dua jenis. Apabila transaksi

tidak berhasil dilakukan. Misalnya karena salah format masukan SMS, maka aplikasi akan mengirimkan laporan kepada pengirim. Isi laporan memberitahukan bahwa masukan salah dan perlu diulang. Begitu juga bila transaksi berhasil dilakukan. Isi laporan berupa pemberitahuan bahwa transaksi telah berhasil dilakukan. Selain itu diakhir pemberitahuan, terdapat biaya yang harus dikeluarkan oleh pembeli.

4.12 Perancangan Keamanan Sistem

Perancangan keamanan pada sistem dilakukan dengan mengatur hak user yang diperbolehkan untuk menggunakan basis data yang terdapat pada sistem. Terdapat dua basis data yang digunakan dalam sistem ini, yaitu basis data `sms` yang merupakan milik Gnokii dan basis data `warunglinux` yang merupakan milik osCommerce. Kedua basis data ini dibatasi dalam akses *account*-nya. *Account user* yang diperbolehkan secara *default* adalah `root`. Sehingga *user* lain tidak dapat mengakses basis data tersebut.

Langkah keamanan lainnya juga dengan memproteksi *folder* admin pada aplikasi osCommerce agar tidak dapat diakses oleh pihak lain. Hal ini dilakukan dengan memodifikasi file `.htaccess` yang terdapat pada aplikasi Apache.



BAB V

IMPLEMENTASI

Dalam bab ini akan dijelaskan mengenai implementasi *Pengembangan Media Transaksi Menggunakan SMS (Short Message Service) Untuk Belanja Online*. Implementasi sistem meliputi implementasi penggunaan perangkat keras dan perangkat lunak. Implementasi perangkat keras terdiri atas penggunaan ponsel yang berfungsi sebagai *modem* pada *SMS Gateway*. Sedangkan implementasi perangkat lunak adalah pembuatan aplikasi antarmuka menggunakan bahasa pemrograman skrip Perl.

5.1 Implementasi LAMP (Linux/Apache/MySQL/PHP)

Penggunaan solusi LAMP (Linux/Apache/MySQL/PHP) didasarkan pada pertimbangan legalitas. Karena semua komponen LAMP menggunakan lisensi GPL (GNU Public License). Sehingga diharapkan penggunaannya tidak akan melanggar hukum yang berlaku. Selain itu solusi LAMP merupakan solusi yang relatif murah dibandingkan menggunakan solusi lain.

Sistem operasi tempat dimana semua aplikasi ditempatkan adalah menggunakan GNU/Linux. Distro GNU/Linux yang digunakan adalah Ubuntu Linux versi 7.04 Feisty Fawn menggunakan kernel 2.6.20-15-generic. Penggunaan Ubuntu Linux didasarkan pada pertimbangan instalasi program yang mudah. Ubuntu Linux juga merupakan turunan dari distro Debian yang terkenal atas kehandalannya sebagai solusi *server*. Sehingga diharapkan akan mudah dalam konfigurasi nantinya.

Ubuntu Linux ditempatkan pada partisi yang dijelaskan lebih detail sebagai berikut:

- sda6 sebagai partisi / (*root*).
- sda7 sebagai partisi home.
- sda8 sebagai partisi swap.

Pemisahan partisi home dengan / (*root*) adalah dengan alasan kemudahan konfigurasi apabila GNU/Linux akan di-*install* ulang.

Selain sistem operasi, diperlukan juga aplikasi-aplikasi pendukung agar tujuan dari pembuatan tugas akhir tercapai. Di bawah dijelaskan secara detail mengenai aplikasi yang digunakan dalam pengerjaan tugas akhir:

- *Web Server* yang digunakan menggunakan Apache 2.2.3.
- Basis data yang digunakan adalah MySQL 5.0.38.
- Bahasa pemrograman yang digunakan adalah PHP 5.2.1.

5.2 Implementasi Aplikasi WarungLINUX

WarungLINUX menggunakan aplikasi osCommerce sebagai aplikasi untuk administrasi dalam menangani transaksi pada toko *online*. OsCommerce yang digunakan adalah osCommerce 2.2 Milestone 2 Update 051112. Aplikasi osCommerce ditempatkan di dalam *folder* `/var/www` dengan menggunakan nama `catalog`.

5.2.1 Implementasi Basis Data WarungLINUX

Seluruh aplikasi WarungLINUX menggunakan basis data MySQL. Pada *file* instalasi, osCommerce telah menyediakan *file* konfigurasi awal basis data yang diperlukan. *File* ini terdapat pada *folder* `/catalog/install` dengan menggunakan nama `oscommerce.sql`. *File* ini nantinya digunakan dalam salah satu tahapan instalasi osCommerce.

Basis data yang ada perlu dikonfigurasi ulang dengan alasan keamanan. Agar terhindar dari penggunaan oleh pengguna yang tidak berwenang yang dapat merusak struktur dari basis data yang telah dibuat. Untuk itu, konfigurasi yang dilakukan untuk basis data untuk WarungLINUX adalah:

- Membuat basis data dengan nama `warunglinux`.
- Memberikan *password* yaitu rahasia pada *user* `root` yang mempunyai hak akses penuh.
- Membuat *user* baru dengan nama `an96ar` dengan *password* `riskey`. *User* `an96ar` mempunyai hak setingkat dengan `root`.
- Menghapus semua *user* yang ada sehingga hanya tersisa *user* seperti yang diperlihatkan dalam Gambar 5.1.

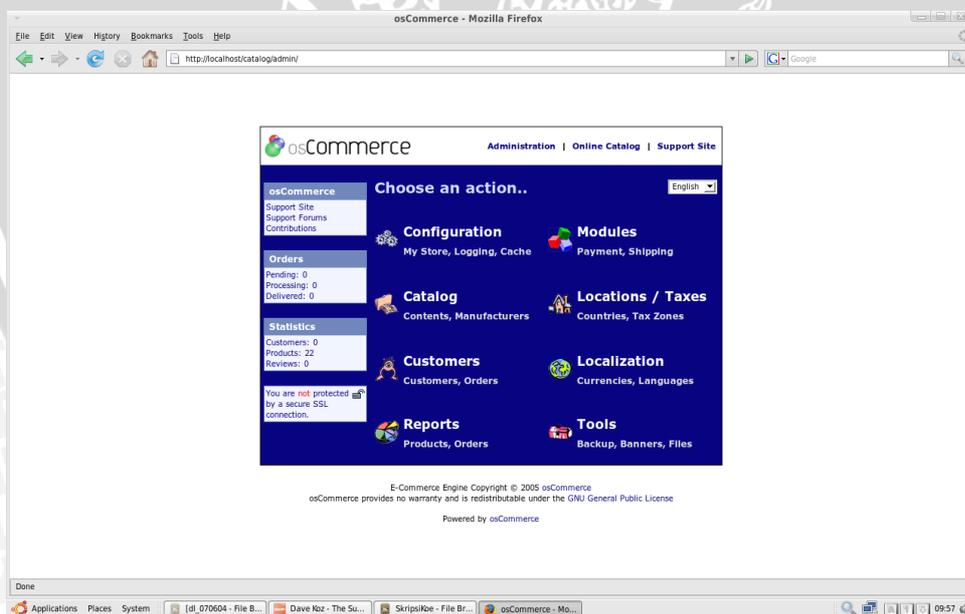
The screenshot shows a MySQL Query window with the following SQL query: `SELECT * FROM `user` LIMIT 0,1000`. The results are displayed in a table with columns: Host, User, Password, and Select.

Host	User	Password	Select
localhost	root	*3E5287812B7D1F947439AC45E73935377A3ADEF7	Y
darkbox	root		Y
127.0.0.1	root		Y
localhost	debian-sys-maint	*45367F7442C51D4B877F2726E93F94438E97C0CD	Y
%	an96ar	*9D670CA2E0F61F4C15E09AB9CD648B2CC91216	Y

Gambar 5.1. Daftar *User* MySQL.
Sumber: [Implementasi]

5.2.2 Implementasi Administrasi

Untuk melakukan proses administrasi pada WarungLINUX, seseorang perlu mengakses halaman Administration. Halaman ini dapat diakses melalui alamat <http://localhost/catalog/admin>. Pada halaman ini semua proses administrasi yang diperlukan terdapat di sini.

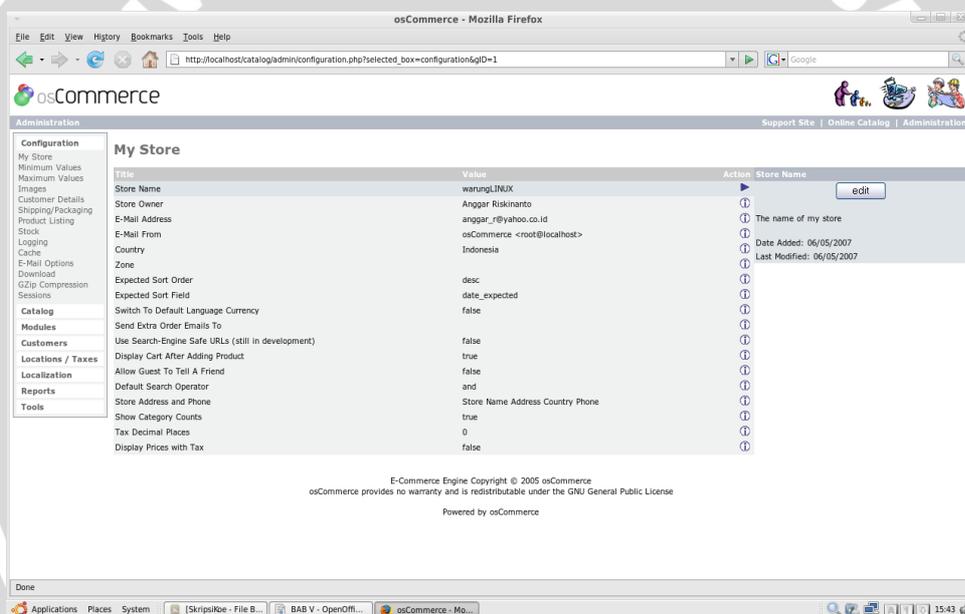


Gambar 5.2. Halaman Administration.
Sumber: [Implementasi]

Halaman administrasi pada WarungLINUX terbagi menjadi 8 bagian utama, yaitu:

1. Configuration.

Bagian Configuration berfungsi untuk konfigurasi umum pada toko *online* yang sedang digunakan. Di dalamnya terdapat sub bagian, yaitu My Store, Logging, dan Cache. Dalam My Store terdapat berbagai konfigurasi yang umum untuk toko *online*. Di bagian ini administrator dapat mengganti nama toko, pemilik toko, alamat *e-mail* pemilik toko. Pada tugas akhir, nama toko diubah menjadi WarungLINUX yang sesuai dengan perencanaan pada bab sebelumnya. Kemudian mengganti nama pemilik dan alamat *e-mail* menjadi Anggar Riskianto dan `anggar_r@yahoo.co.id`. Yang terakhir adalah mengganti negara pada masukan Country menjadi Indonesia.



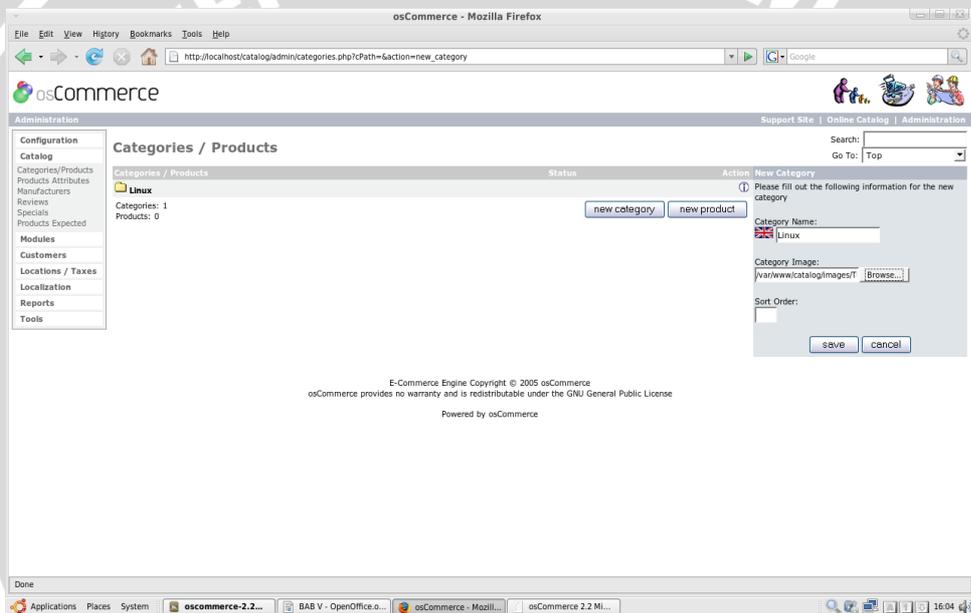
Gambar 5.3. Halaman My Store.

Sumber: [Implementasi]

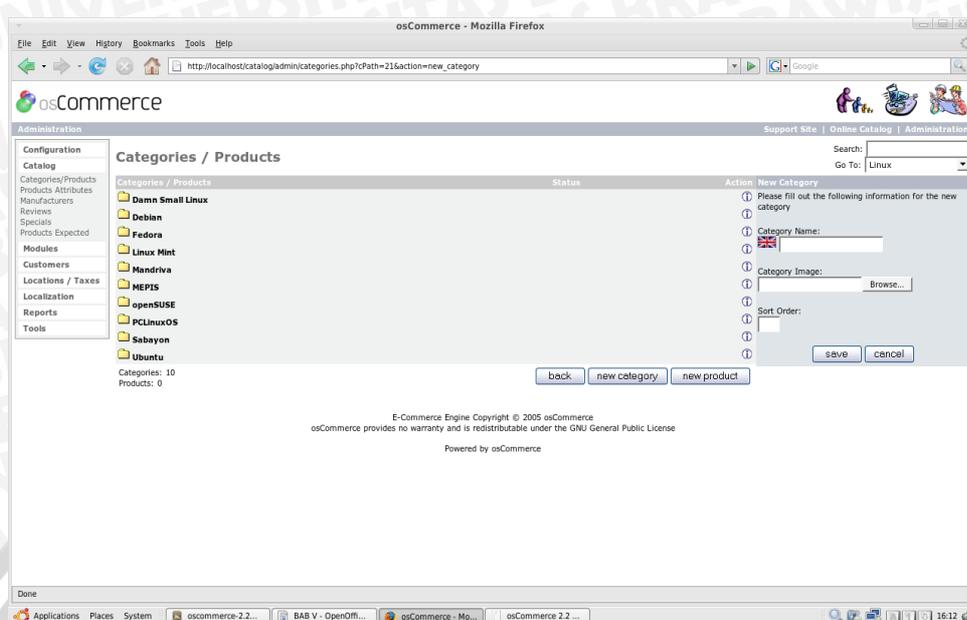
2. Catalog.

Bagian Catalog berfungsi untuk memasukkan produk yang akan dijual pada toko *online*. Di dalamnya terdapat beberapa sub bagian, yaitu Categories/Products, Products Attributes, Manufacturers, Reviews, Specials, dan Products Expected. Pada warungLINUX, sub bagian yang sering digunakan adalah pada Categories/Products. Di sub bagian ini administrator memasukkan kategori-kategori yang ada pada WarungLINUX. Pada WarungLINUX, ditambahkan sebuah kategori baru yaitu Linux. Hal ini dilakukan dengan cara menekan tombol

New Category yang ada pada halaman Categories/Products. Kemudian mengisi Linux pada masukan Category Name dan memasukkan alamat tempat logo pada masukan Category Image. Setelah selesai, kemudian menekan tombol Save di halaman tersebut. Setelah kategori Linux dibuat, maka perlu dibuat sub-sub kategori yang baru. Sub-sub kategori ini didasarkan pada 10 distro yang ada pada bab perancangan. Pembuatan sub kategori dapat dilakukan dengan cara meng-klik tombol New Category. Kemudian mengisi masukan yang sesuai pada kolom New Category yang telah terbuka. Setelah selesai, maka tombol Save diklik.



Gambar 5.4. Halaman Kategori.
Sumber: [Implementasi]

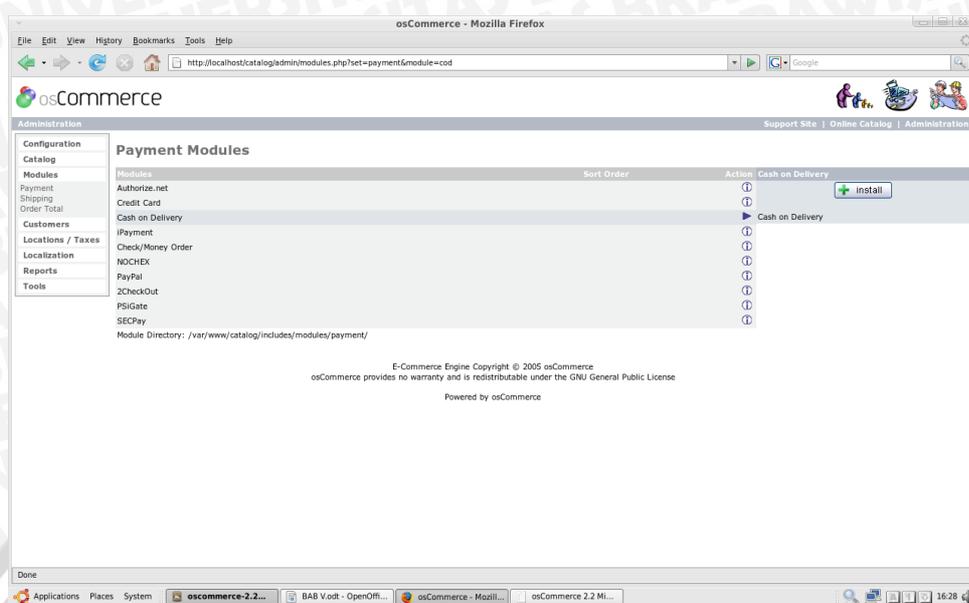


Gambar 5.5. Halaman Produk.
Sumber: [Implementasi]

Dalam sub-sub kategori tersebut perlu dimasukkan produk-produk baru. Hal ini dilakukan dengan cara meng-klik *link* sub kategori yang telah dibuat, yang akan membuka halaman baru. Di halaman ini, administrator menekan tombol *New Product* untuk membuat produk-produk baru. Dalam halaman *New Product*, administrator mengisi masukan yang sesuai dengan produk yang akan dijual. Setelah selesai mengisi, cukup menekan tombol *Preview*. Setelah tombol *Preview* ditekan akan membuka halaman baru yang akan menampilkan contoh halaman yang akan ditampilkan pada WarungLINUX nantinya. Jika halaman tersebut telah sesuai, maka tekan tombol *Insert* untuk benar-benar memasukkan produk ke dalam WarungLINUX.

3. Modules.

Pada bagian *Modules*, terdapat 3 sub bagian, yaitu *Payment*, *Shipping*, dan *Order Total*. Pada bagian *Payment*, perlu di non aktifkan semua bagian, kecuali *Cash On Delivery* dan *Check/Money Order*. Hal ini perlu dilakukan karena WarungLINUX tidak menerima pembayaran selain kedua cara tersebut. Dengan menekan tombol *Remove* pada setiap bagian, maka pembayaran yang tidak diinginkan menjadi non aktif.

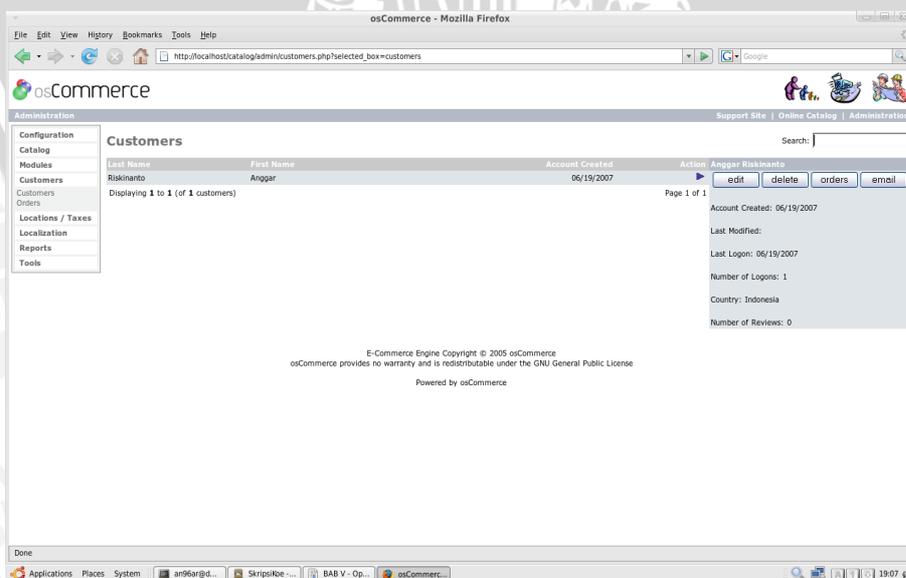


Gambar 5.6. Halaman Payment Modules.

Sumber: [Implementasi]

4. Customers.

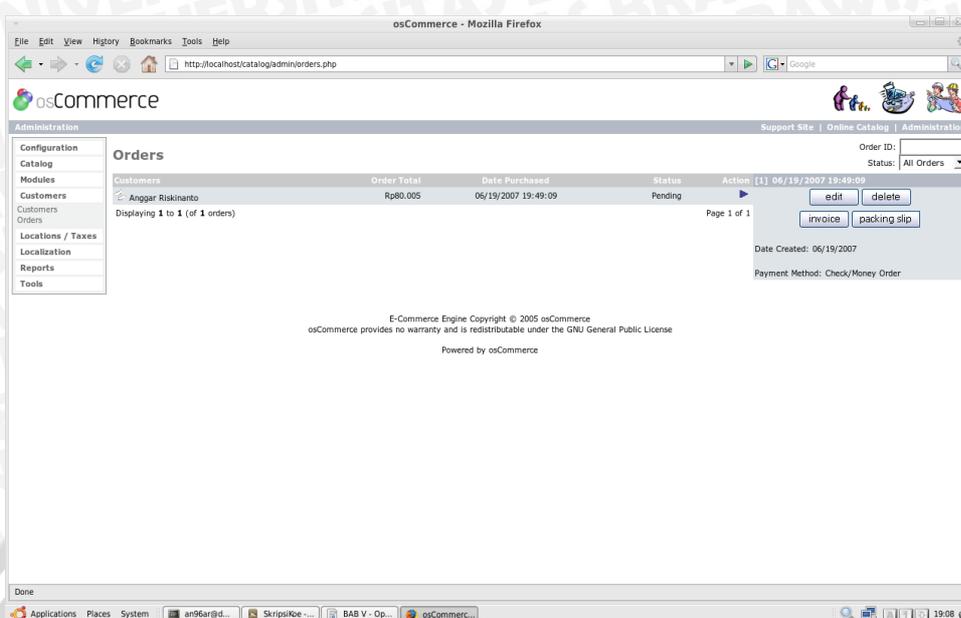
Bagian ini mengatur mengenai pelanggan dan pesannya. Terdapat dua sub bagian pada bagian ini, yaitu Customers dan Orders. Customers menampilkan daftar pelanggan yang telah mendaftarkan diri mereka pada WarungLINUX. Sedangkan Orders berisi daftar pesanan yang telah dilakukan oleh para pelanggan.



Gambar 5.7. Halaman Customers.

Sumber: [Implementasi]

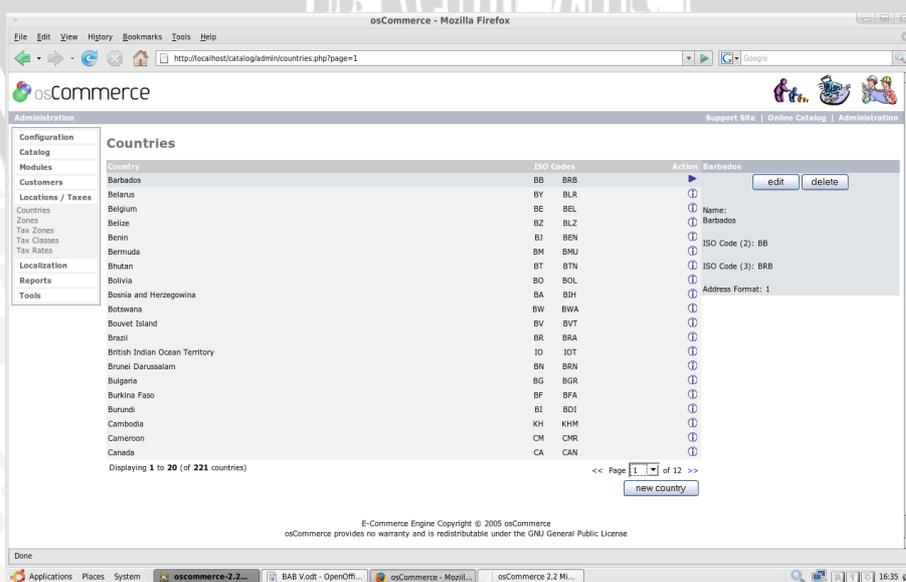




Gambar 5.8. Halaman Orders.
Sumber: [Implementasi]

5. Locations/Taxes.

Bagian ini mengatur mengenai pajak dan zona dimana WarungLINUX berada. Karena WarungLINUX hanya menangani transaksi belanja di dalam negeri, maka zona selain Indonesia perlu dihapus. Hal ini dilakukan pada sub bagian Countries. Pada negara yang akan dihapus hanya perlu menekan tombol Delete. Hal ini dilakukan berulang-ulang hingga hanya tersisa negara Indonesia.



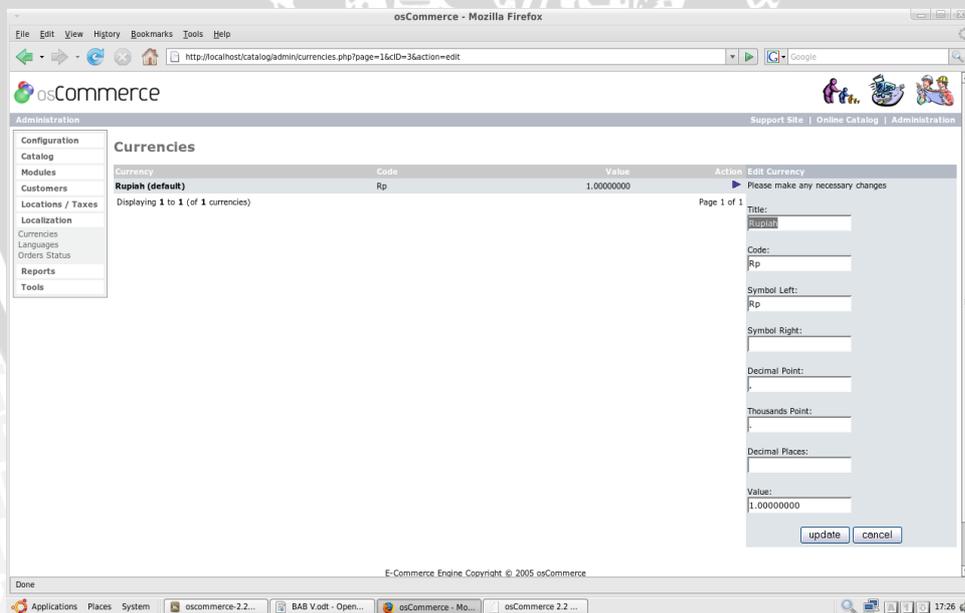
Gambar 5.9. Halaman Countries.
Sumber: [Implementasi]



Selain zona, hal-hal yang berhubungan dengan pajak juga perlu dihapus. Karena WarungLINUX tidak mengenakan pajak bagi para pembeli. Hal ini cukup dengan menekan tombol `Delete` pada bagian yang perlu dihapus, seperti pada bagian `Countries`.

6. Localizations.

Bagian ini mengatur mengenai mata uang, bahasa, dan status pemesanan. Dua bagian yang perlu diubah agar sesuai dengan tujuan WarungLINUX adalah `Currencies` dan `Languages`. Bagian `Currencies` hanya perlu menggunakan Rupiah. Hal ini dilakukan dengan cara menghapus mata uang yang telah ada dan membuat mata uang baru, yaitu Rupiah. Dengan menekan tombol `Delete` untuk menghapus mata uang yang telah ada. Kemudian menekan tombol `New Currencies` untuk membuat mata uang baru. Pada kolom `New Currencies` administrator memasukkan masukan yang diperlukan untuk membuat mata uang baru. Setelah sesuai, maka menekan tombol `Insert` untuk membuat mata uang baru, yaitu Rupiah.

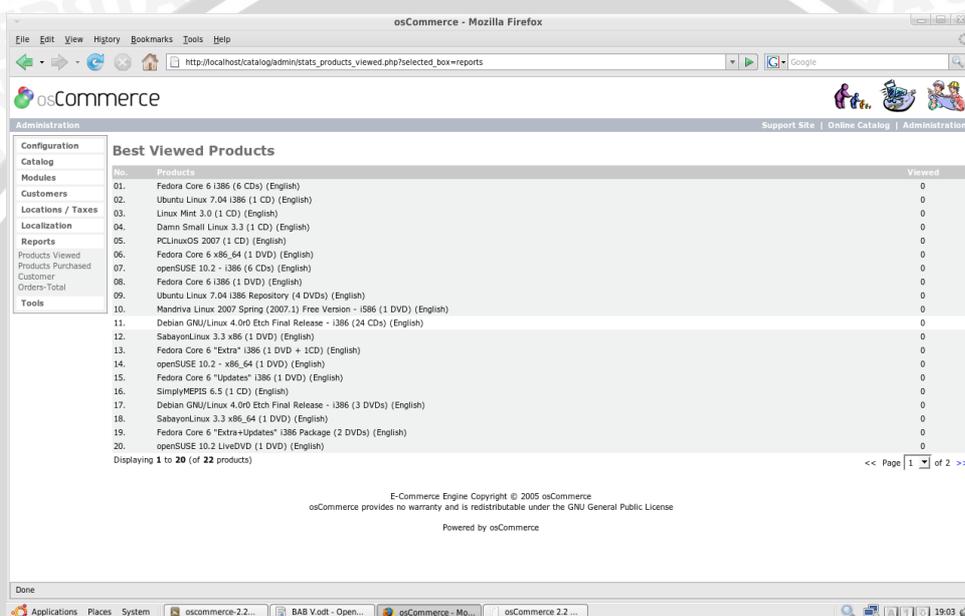


Gambar 5.10. Halaman `New Currencies`.
Sumber: [Implementasi]

7. Reports.

Bagian `Reports` berisi mengenai laporan dari perubahan yang terjadi di WarungLINUX. `Reports` mempunyai beberapa sub bagian, yaitu `Products`

Viewed, Products Purchased, dan Customers Orders-Totals. Sub bagian Products Viewed berisi laporan mengenai produk yang telah dilihat oleh pengunjung WarungLINUX, sedangkan Products Purchased berisi laporan daftar produk yang telah dibeli oleh pelanggan WarungLINUX. Customers Orders-Totals adalah laporan mengenai pembelian untuk membeli produk WarungLINUX.

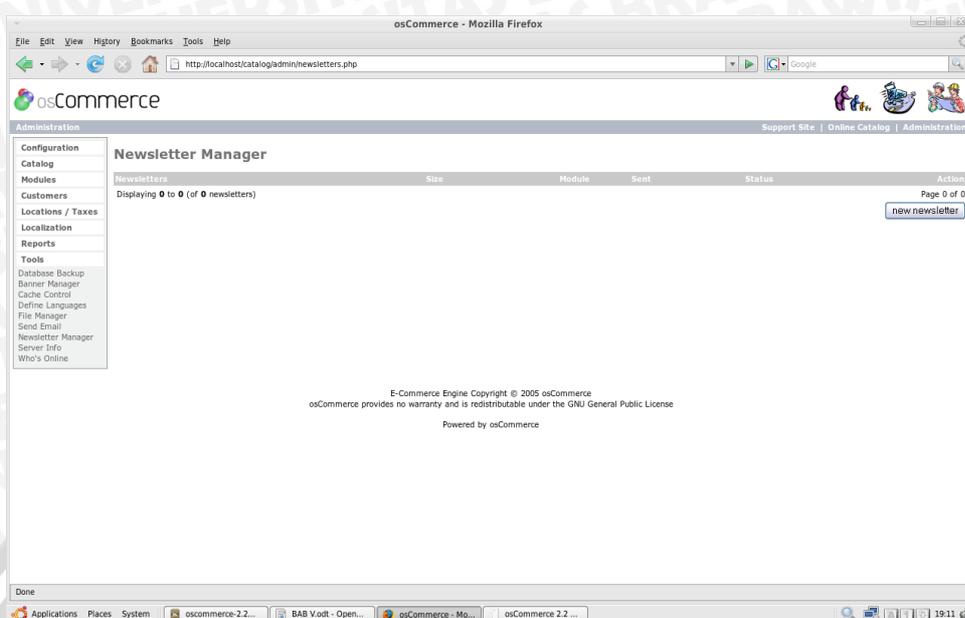


Gambar 5.11. Halaman Reports.

Sumber: [Implementasi]

8. Tools.

Bagian Tools berisi sub-sub bagian tambahan yang berguna untuk membantu administrator dalam mengatur WarungLINUX. Sub-sub bagian terpenting bagi administrator di bagian ini adalah Backup Manager, Banner Manager, Newsletter Manager, dan Who's Online. Backup Manager berfungsi untuk mem-backup data yang ada pada WarungLINUX. Hasil dari backup disimpan ke dalam file dengan format GZIP, ZIP, atau berbentuk SQL biasa. Banner Manager adalah suatu tool untuk mengatur iklan-iklan yang nantinya akan ditampilkan dalam WarungLINUX. Sedangkan Who's Online berisi daftar para pelanggan yang sedang mengunjungi WarungLINUX.



Gambar 5.12. Halaman Tools.

Sumber: [Implementasi]

Untuk membatasi agar halaman admin tidak bisa diakses oleh sembarang orang, maka diperlukan suatu tindakan pengamanan. Pengamanan dapat dilakukan dengan membatasi akses pada beberapa halaman dan *file* dengan cara:

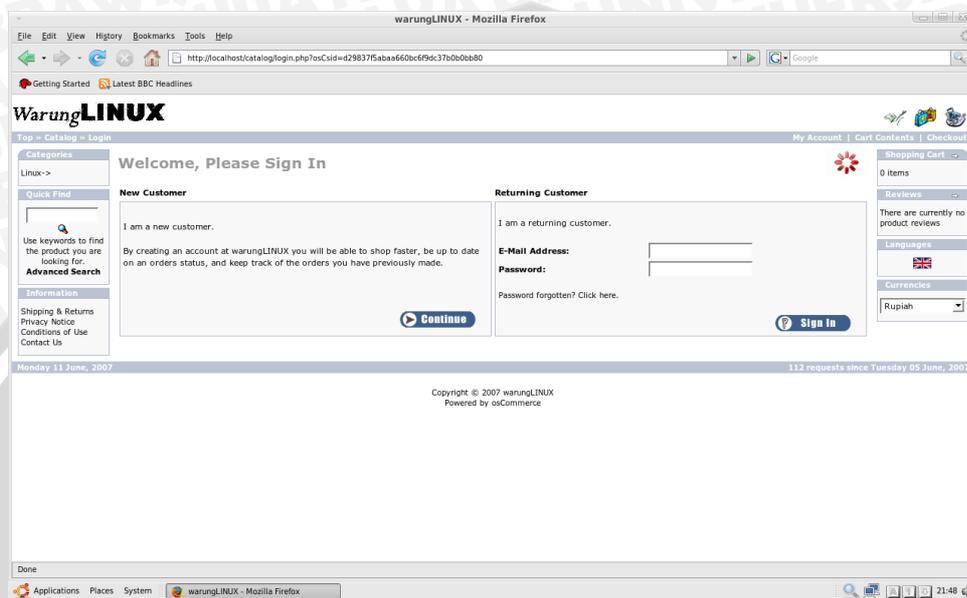
1. Merubah nama *folder* catalog atau menghapus *folder* catalog.
2. Mengembalikan hak akses pada `/catalog/includes/configure.php` menjadi 644 (jika masih mendapatkan pesan kesalahan pada bagian atas, set menjadi 444 yang merupakan *read only*).
3. Men-set hak akses pada direktori `/catalog/images` menjadi 777.
4. Mengembalikan hak akses pada `/catalog/admin/includes/configure.php` menjadi 644.
5. Membuat direktori `/catalog/admin/backups` dan men-set hak akses menjadi 777.
6. Men-set hak akses direktori `/catalog/admin/images/graphs` menjadi 777.

Selain cara diatas juga perlu memberikan `.htaccess` pada direktori `catalog/admin` sehingga dilindungi oleh *password*.

5.2.3 Implementasi Registrasi

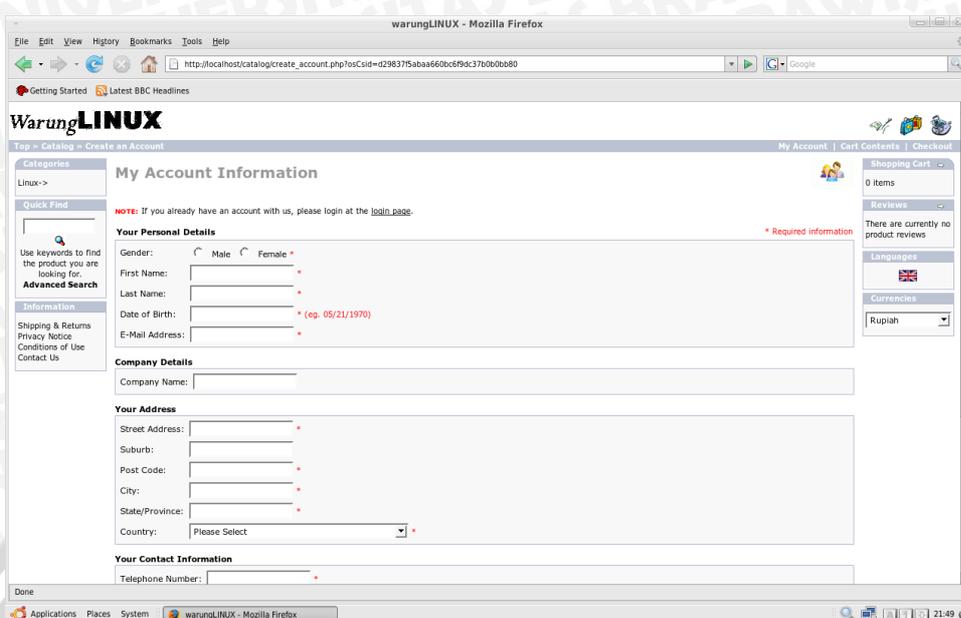
Untuk melakukan transaksi, pengunjung harus melakukan proses registrasi dahulu. Proses ini dilakukan dengan mengakses *link* "create an account" pada

halaman awal atau melalui menu My Account. Pada halaman Login, pengunjung perlu meng-klik *link* Continue untuk melakukan proses registrasi. Setelah meng-klik Continue, pengunjung dibawa menuju halaman My Account Information.



Gambar 5.13. Halaman Registrasi dan Login.
Sumber: [Implementasi]

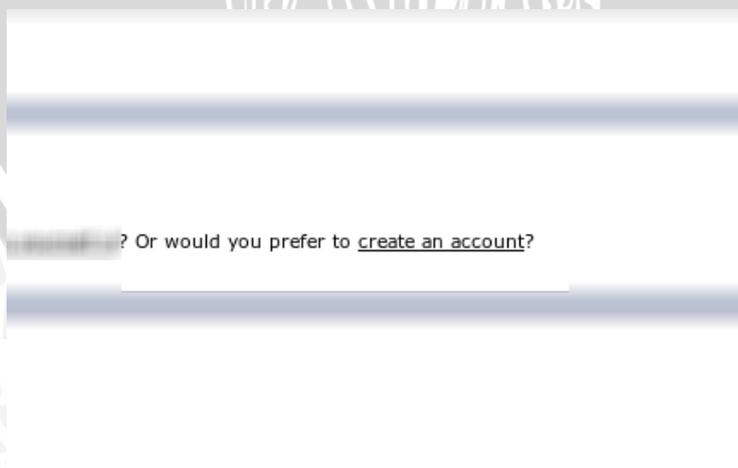
Pada halaman ini, pengunjung mengisi masukan-masukan yang diperlukan untuk proses Registrasi. Dua masukan terpenting adalah alamat *e-mail* dan *password*. Karena dua masukan ini nantinya akan digunakan untuk proses *login*. Setelah selesai mengisi semua masukan, pengunjung meng-klik *link* Continue.



Gambar 5.14. Halaman Registrasi.
Sumber: [Implementasi]

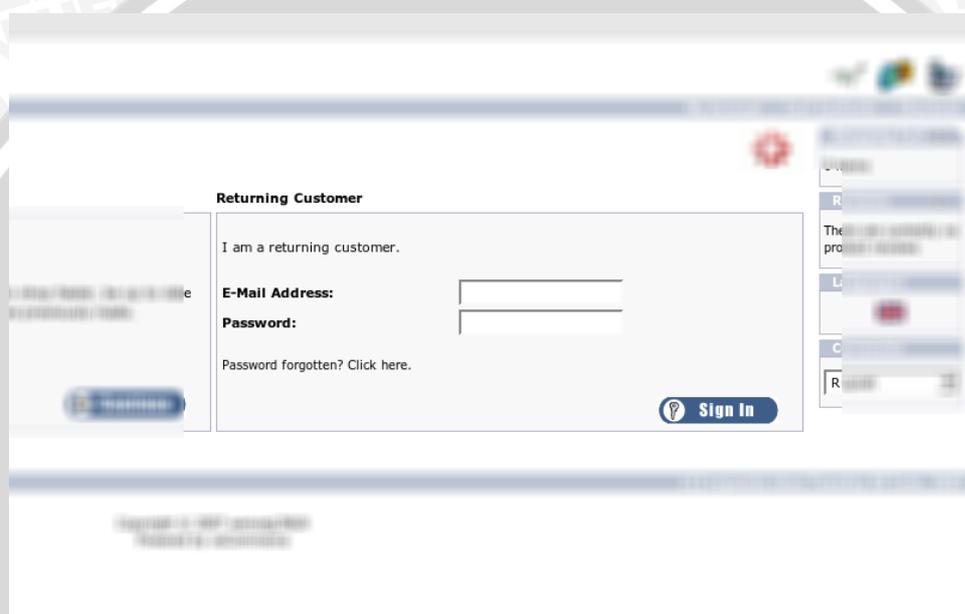
5.2.4 Implementasi Login

Seperti yang telah dijelaskan pada bab perancangan, pengunjung yang ingin melakukan transaksi diharuskan melakukan proses *login* terlebih dahulu. Apabila pengunjung tidak melakukan proses *login*, maka ia hanya dapat melihat-lihat barang yang ditawarkan. Proses *login* dilakukan dengan mengakses *link* “log yourself in” yang terdapat pada halaman utama atau melalui menu My Account yang juga terdapat pada halaman utama.



Gambar 5.15. Proses Login pada Halaman Depan.
Sumber: [Implementasi]

Pada halaman *login*, pengunjung harus mengisi dua masukan untuk alamat *e-mail* dan *password* yang sesuai pada kolom *Returning Customer*. Kedua masukan ini harus diisi dengan data yang telah dimasukkan ketika melakukan registrasi. Setelah mengisi selesai mengisi alamat *e-mail* dan *password* yang sesuai, maka pengunjung meng-klik *link Sign In*. Setelah proses tersebut, maka pengunjung telah berubah statusnya menjadi pelanggan dan dibawa menuju halaman depan atau halaman sebelum ia melakukan proses *login*.



Gambar 5.16. Halaman *Login*.
Sumber: [Implementasi]

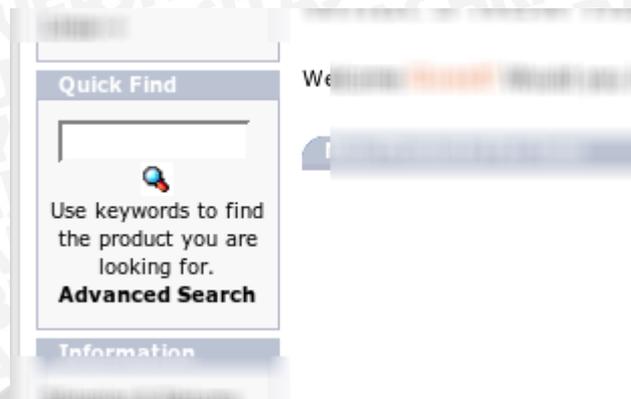
5.2.5 Implementasi Pencarian Barang

Untuk mempermudah dalam mencari barang, WarungLinux telah menyediakan fasilitas pencarian barang. Pencarian barang dapat dilakukan dengan memasukkan kata kunci yang diinginkan pada menu *Quick Find* yang ada pada kolom paling kanan pada WarungLinux. Selain melalui masukan yang ada pada menu *Quick Find*, pelanggan juga dapat mencari barang dengan lebih detail melalui *Advanced Search* yang bisa dicapai dengan meng-klik *link Advanced Search*.

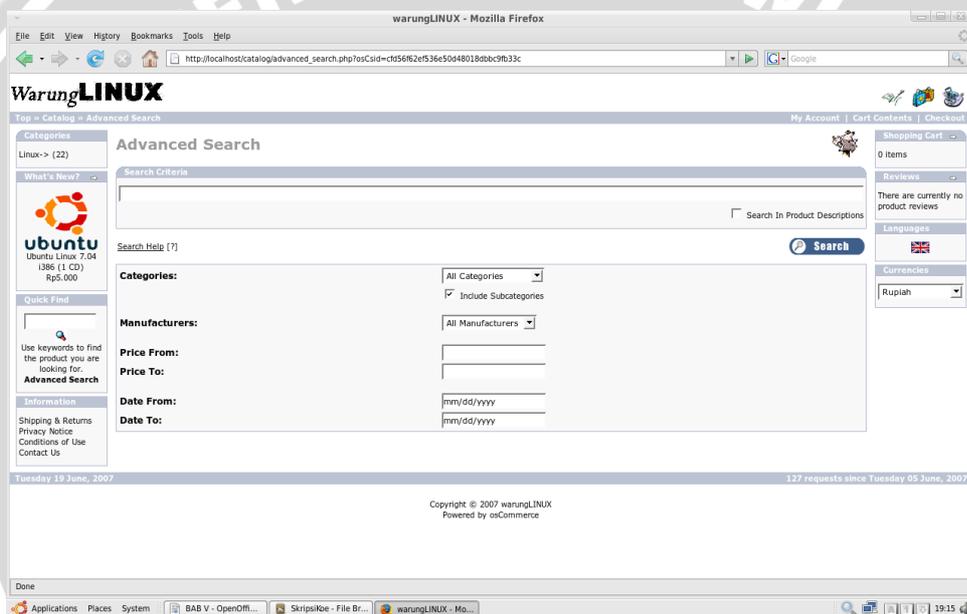
Pada halaman *Advanced Search*, pelanggan dapat mencari barang dengan lebih detail dengan memasukkan data-data masukan tambahan seperti:

1. Categories.
Masukan ini berfungsi untuk mencari barang sesuai dengan kata kunci yang terdapat pada kategori tertentu. Kategori yang ada merupakan kategori yang sebelumnya telah dibuat oleh administrator.
2. Manufacturers.
Masukan ini berfungsi untuk mencari barang sesuai dengan kata kunci yang terdapat pada manufaktur atau pabrikan tertentu.
3. Price From.
Masukan ini berfungsi untuk mencari barang sesuai kata kunci yang mempunyai harga yang dimulai dari harga yang telah dimasukkan pada masukan.
4. Price To.
Masukan ini berfungsi untuk mencari barang sesuai kata kunci yang mempunyai harga hingga mencapai harga yang telah dimasukkan pada masukan.
5. Date From.
Masukan ini berfungsi untuk mencari barang yang telah dimasukkan ke dalam WarungLINUX oleh administrator mulai dari tanggal yang telah dimasukkan pada masukan.
6. Date To.
Masukan ini berfungsi untuk mencari barang yang telah dimasukkan ke dalam WarungLINUX oleh administrator hingga mencapai tanggal yang telah dimasukkan pada masukan.

Kata kunci yang dimasukkan dapat digabungkan dengan kata kunci lainnya untuk mendapatkan hasil yang lebih memuaskan. Penggunaan AND dan OR sangat dianjurkan untuk mendapatkan hasil yang diinginkan. Kata kunci yang sama persis dapat dicari menggunakan tanda baca dua kutip untuk mencari rangkaian kata yang sama. Seperti penggunaan “toko komputer” akan mencari toko komputer daripada mencari toko dan komputer. Sedangkan penggunaan tanda baca kurung akan lebih mengontrol pencarian kata kunci. Penggunaan kata kunci toko AND (komputer OR linux) akan mencari kata kunci dengan kata-kata toko komputer atau toko linux.



Gambar 5.17. Pencarian Barang melalui Quick Find.
Sumber: [Implementasi]



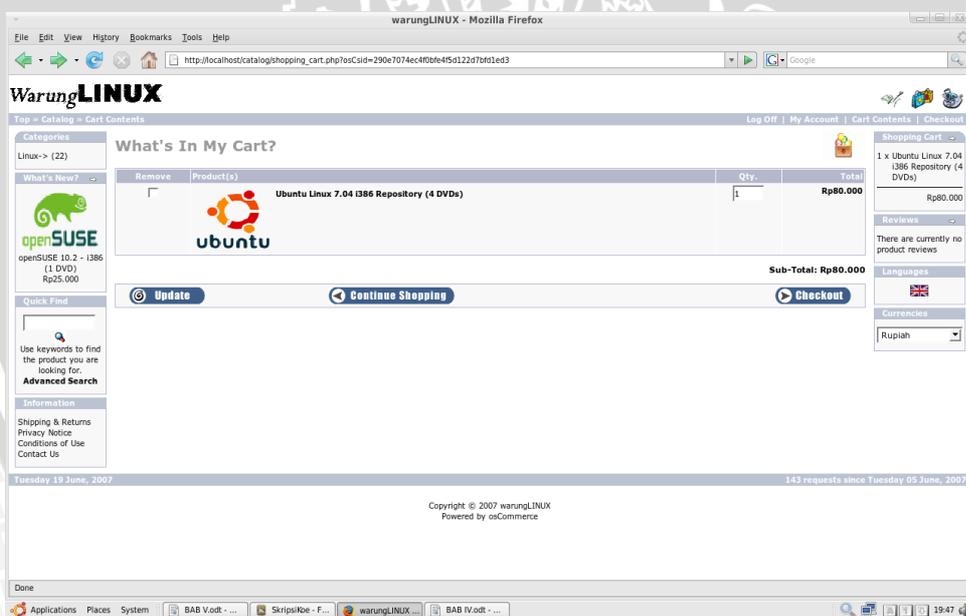
Gambar 5.18. Halaman Advanced Search.
Sumber: [Implementasi]

5.2.6 Implementasi Pembelian Barang

Pelanggan dapat melakukan pembelian barang dengan terlebih dahulu memilih barang yang terdapat dalam kategori-kategori yang tersedia dalam WarungLINUX. Dengan meng-klik *link* Add to Cart yang ada pada halaman suatu produk, maka barang yang telah dipilih akan dimasukkan ke dalam Cart. Isi dari Cart dapat dilihat melalui menu Cart Contents. Setelah selesai memilih barang, pelanggan dapat meng-klik menu Checkout untuk melakukan tahapan pembayaran.



Gambar 5.19. Halaman Produk.
Sumber: [Implementasi]

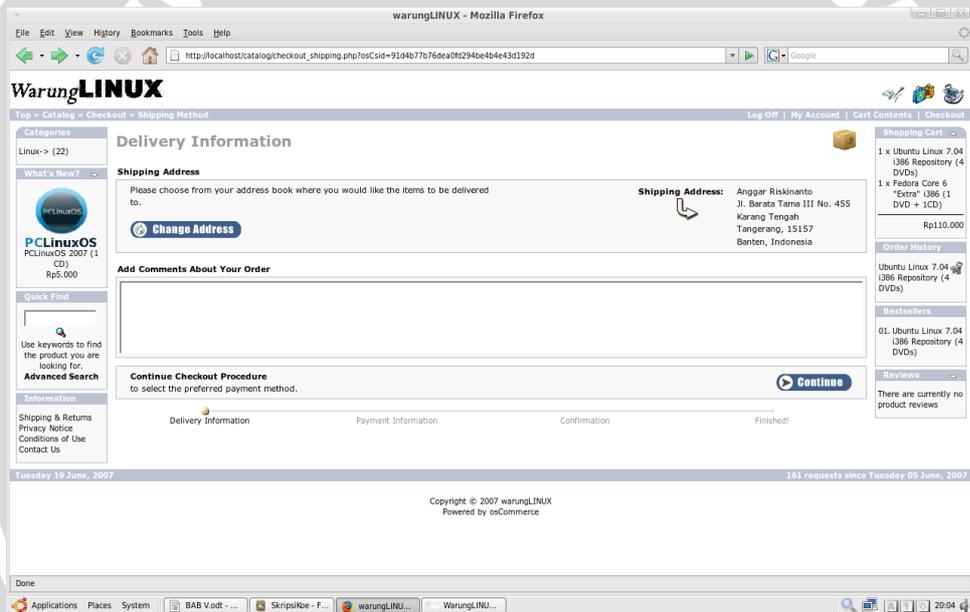


Gambar 5.20. Halaman Cart Content.
Sumber: [Implementasi]

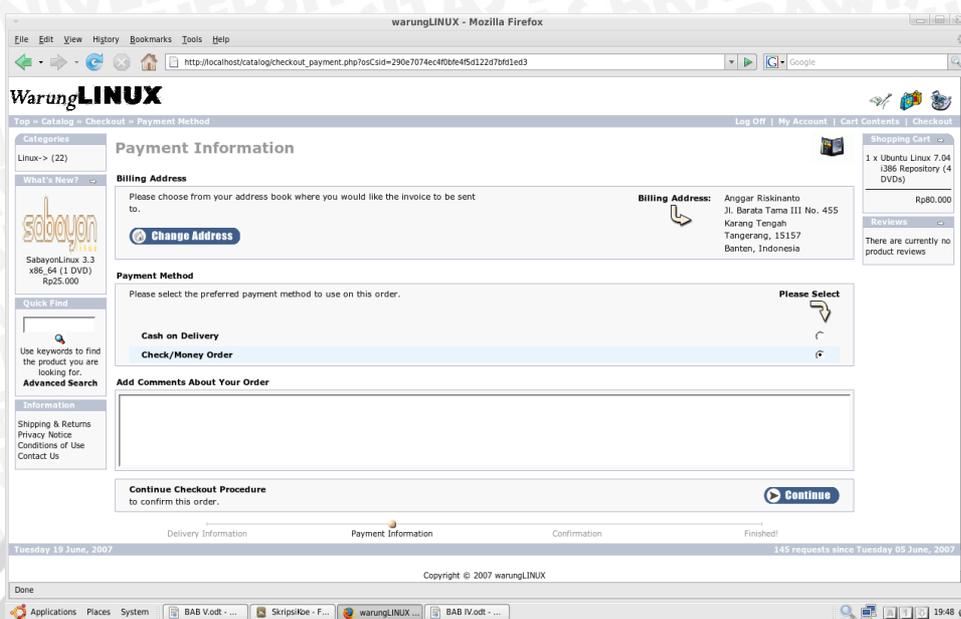
Tahapan pembayaran dimulai dengan memperlihatkan halaman Delivery Information. Pada halaman ini, pelanggan akan diberikan informasi mengenai alamat pengiriman paket. Pelanggan dapat mengganti alamat pengiriman sesuai dengan yang diinginkannya melalui *link* Change Address. Setelah alamat pengiriman sesuai dengan keinginan pelanggan, maka pelanggan meng-klik *link* Continue untuk meneruskan ke



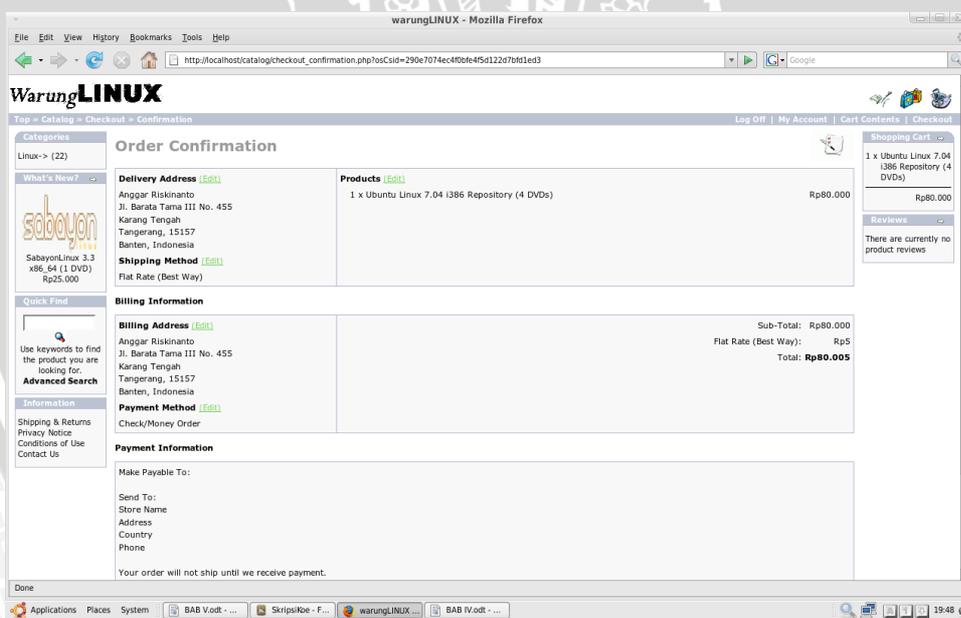
proses pembayaran yang terdapat pada halaman *Payment Information*. Di halaman ini pelanggan kembali dapat mengganti alamat penagihan bila tidak sesuai dengan keinginan pelanggan. Selain itu pelanggan diharuskan memilih salah satu opsi pembayaran yang ditawarkan yaitu *Cash on Delivery* atau *Check/Money Order*. Setelah selesai pada halaman ini, pelanggan dapat kembali meng-klik *link Continue* untuk melanjutkan ke tahap konfirmasi. Pada halaman *Confirmation Order*, pelanggan dapat meninjau kembali tahapan-tahapan pembayaran yang telah ia lakukan. Setelah sesuai dengan keinginan pelanggan, maka tahapan pembayaran telah selesai dilaksanakan. Hal ini ditandai dengan halaman yang memberitahukan bahwa pesanan telah diproses oleh WarungLINUX.



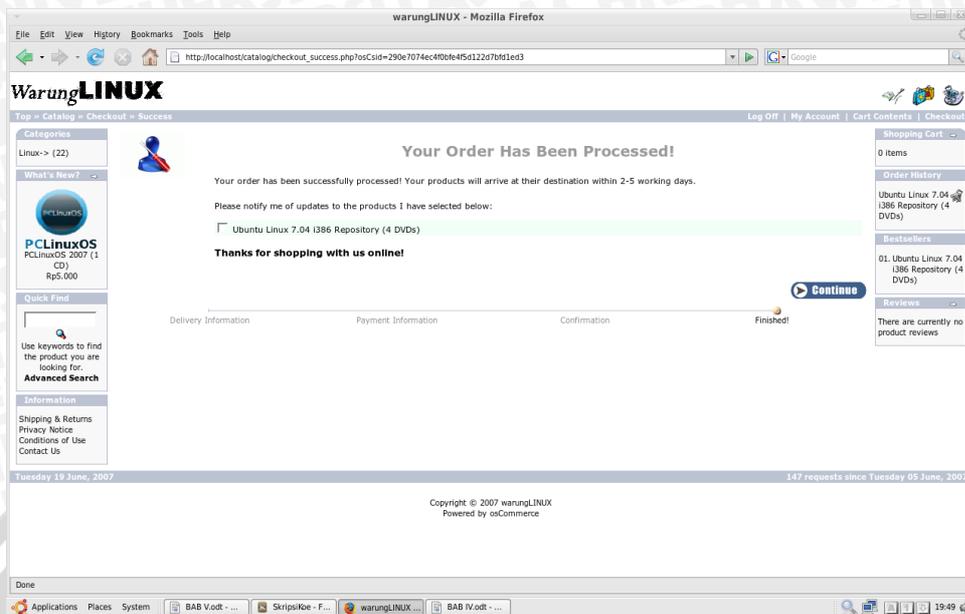
Gambar 5.21. Halaman Delivery Information.
Sumber: [Implementasi]



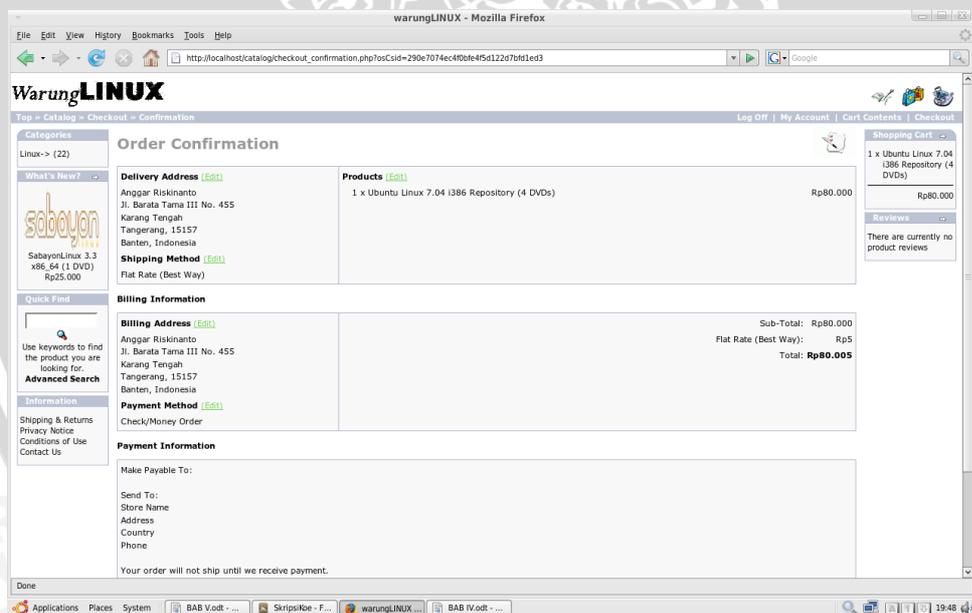
Gambar 5.22. Halaman Payment Information.
Sumber: [Implementasi]



Gambar 5.23. Halaman Order Information.
Sumber: [Implementasi]



Gambar 5.24. Halaman Order Confirmation.
Sumber: [Implementasi]



Gambar 5.25. Halaman Pemberitahuan Pemrosesan Pesanan.
Sumber: [Implementasi]

5.3 Implementasi Gnokii

SMS Gateway yang digunakan dalam pembuatan tugas akhir adalah Gnokii 0.6.14. Gnokii menggunakan *file* gnokiiirc untuk konfigurasi aplikasi. *File* tersebut ditempatkan pada *folder* home milik *user*. Beberapa *setting* yang diperlukan agar aplikasi dapat mengenali perangkat keras yang digunakan adalah:

```
port = /dev/ttyUSB0
model = 6510
connection = dku5
```

Konfigurasi diatas menjelaskan mengenai *port*, model, dan koneksi yang digunakan oleh Gnokii. Port yang terhubung dengan kabel data adalah USB0, dengan model ponsel yaitu Nokia 6510 atau semua ponsel yang mempunyai konfigurasi mirip dengan Nokia 6510. Baris *connection* menjelaskan mengenai kabel koneksi yang digunakan yaitu kabel DKU-5.

5.3.1 Implementasi Perangkat Keras

Agar Gnokii bisa digunakan untuk menerima SMS dari dunia luar, maka diperlukan perangkat keras yang kompatibel dengan Gnokii itu sendiri. Salah satu ponsel yang didukung oleh aplikasi Gnokii sebagai modem adalah Nokia 3100. Nokia 3100 menggunakan antarmuka Pop Port untuk berkomunikasi dengan PC. Pop Port sendiri merupakan antarmuka universal yang sering digunakan oleh banyak ponsel Nokia. Pop Port terdiri dari 1 pin metal pada salah satu sisi akhirnya dan beberapa *tab* plastik yang mempunyai 13 kontak.

Untuk menghubungkan Pop Port dengan *port* pada PC digunakan sebuah kabel data. Kabel data yang digunakan merupakan kabel data *clone* DKU-5. Kabel ini merupakan kabel yang mengemulasikan kabel data serial yang biasa dipakai oleh produk ponsel Nokia. Dalam kabel data terdapat sebuah *chipset* yang berfungsi sebagai *emulator*. Chipset yang digunakan adalah Prolific. Prolific didukung oleh Linux pada kernelnya, dengan driver p12303. Kabel data ini mempunyai koneksi ke *port* USB pada PC.

```
an96ar@darkbox:~$ gnokii --identify
GNOKII Version 0.6.14
.....
Received model RH-19
IMEI       : 351517004327514
Manufacturer : Nokia
Found model "RH-19"
Model      : 3100
Product name : RH-19
Revision   : V 04.01
Serial device: closing device
an96ar@darkbox:~$
```

Gambar 5.26. Hasil Perintah `gnokii -identify`.
Sumber: [Implementasi]

5.3.2 Implementasi Basis Data

Untuk memudahkan dalam mengolah SMS yang masuk dan keluar, Gnokii menggunakan sebuah basis data. Basis data ini diatur oleh sebuah aplikasi smsd. Basis data yang bernama sms ini mempunyai dua buah tabel yang menangani SMS yang masuk dan keluar, yaitu inbox dan outbox. Basis data yang didukung adalah MySQL dan PostgreSQL. Dalam tugas akhir, basis data yang digunakan untuk menangani SMS menggunakan MySQL. Dibawah perintah-perintah yang digunakan untuk membuat basis data sms:

```
# mysql
mysql> CREATE DATABASE smsd;
mysql> USE smsd;
mysql> CREATE TABLE inbox (
    id int(10) unsigned NOT NULL auto_increment,
    number varchar(20) NOT NULL default '',
    smsdate datetime NOT NULL default '0000-00-00 00:00:00',
    insertdate timestamp(14) NOT NULL,
    text varchar(160) default NULL,
    phone tinyint(4),
    processed tinyint(4) NOT NULL default '0',
    PRIMARY KEY (id)
)TYPE=MyISAM;
mysql> CREATE TABLE outbox (
    id int(10) unsigned NOT NULL auto_increment,
    number varchar(20) NOT NULL default '',
    processed_date timestamp(14) NOT NULL,
    insertdate timestamp(14) NOT NULL,
    text varchar(160) default NULL,
    phone tinyint(4),
    processed tinyint(4) NOT NULL default '0',
    error tinyint(4) NOT NULL default '-1',
    dreport tinyint(4) NOT NULL default '0',
    not_before time NOT NULL default '00:00:00',
    not_after time NOT NULL default '23:59:59',
    PRIMARY KEY (id)
) TYPE=MyISAM;
mysql>\q;
```

Selain cara diatas, cara lain adalah dengan mengetikkan perintah:

```
# mysql -u root -d sms < /usr/share/doc/gnokii-smsd-
mysql/sms.tables.mysql.sql
```

Dimana:

- root adalah *user* yang telah terdaftar dalam MySQL, *user* lain dapat digunakan bila *user* tersebut mempunyai hak akses yang sama dengan root.
- sms adalah nama basis data yang akan diisi dengan tabel yang dibutuhkan oleh smsd.
- /usr/share/doc/gnokii-smsd-mysql/sms.tables.mysql.sql adalah direktori dimana *file* basis data ditempatkan.
- sms.tables.mysql.sql adalah *file* skrip untuk membuat tabel.

Berikut merupakan isi dari tabel *inbox* dari basis data *sms*. Isi dari tabel *inbox* merupakan SMS yang diambil dari *inbox* pada ponsel.

id	number	smsdate	insertdate	text	phone	processed
1	+628990357113	2007-06-26 13:43:26	2007-06-29 210	Testing 123	NULL	0

1 row in set (0.00 sec)

Gambar 5.27. Isi Tabel *inbox* pada Basis Data *sms*.
Sumber: [Implementasi]

5.4 Implementasi Aplikasi Antarmuka

Aplikasi antarmuka bertujuan untuk mencacah isi dari tabel *inbox* pada basis data *sms* agar nantinya dimasukkan ke dalam tabel yang berhubungan dengan transaksi barang dalam basis data *warunglinux*. SMS yang berada pada tabel *inbox* berupa deretan kata yang mempunyai makna khusus. Deretan kata ini nantinya dicacah untuk dimasukkan ke dalam variabel. Variabel-variabel yang ada nantinya menjadi sebuah masukan untuk *query* basis data terhadap transaksi barang yang terjadi pada *warungLINUX*.

5.4.1 Koneksi Basis Data

Basis data yang diakses oleh aplikasi antarmuka akan terdiri dari dua basis data. Keduanya adalah *sms* yang merupakan milik aplikasi *Gnokii* dan *warunglinux* yang merupakan milik aplikasi *WarungLINUX*. Berikut adalah fungsi untuk koneksi terhadap kedua basis data.

```
$db1=DBI->connect('dbi:mysql:sms','root','rahasia');
$db2 =DBI->connect('dbi:mysql:warunglinux','root','rahasia');
```

Penjelasan lebih lanjut mengenai perintah-perintah di atas adalah:

- `$db1` dan `$db2` adalah variabel-variabel dimana perintah koneksi disimpan.
- `DBI->connect` adalah sebuah *driver* Perl untuk menghubungkan dengan sebuah basis data.
- `dbi:mysql:sms` adalah sebuah *driver* DBI MySQL yang mengkoneksikan dengan basis data `sms`.
- `dbi:mysql:warunglinux` adalah sebuah *driver* DBI MySQL yang mengkoneksikan dengan basis data `warunglinux`.
- `root` merupakan *user* yang melakukan koneksi ke basis data.
- `rahasia` merupakan password milik `root`.

5.4.2 Pencacahan SMS

Data masukan yang berupa teks SMS masih berupa data mentah. Data tersebut tersimpan pada tabel `inbox` pada basisdata `sms`. Agar data masukan tersebut dapat digunakan sebagai masukan untuk *query* ke dalam tabel milik `warunglinux` diperlukan proses pencocokan pola terlebih dahulu.

Proses pencocokan pola inilah yang dinamakan *regular expression* (regex). Regex akan mencacah masukan berupa teks SMS sehingga dapat dimasukkan ke dalam basis data `warunglinux`. Pola pencacahan SMS tersebut berdasarkan pada pola yang sudah dijelaskan pada Bab Perancangan. Fungsi di bawah adalah pola regex yang diterapkan untuk mencapai tujuan tersebut.

```
my @sms_tercacah=split(/ /,$sms_masukan);

foreach $kata(@sms_tercacah){
    $kata =~ s/#//;
    $kata =~ s/\*//;
    $kata =~ s/\=//;
}
```

Gambar 5.28. Pola Regex untuk Pencacahan SMS.
Sumber: [Implementasi]

Dalam Gambar 5.28 diperlihatkan pola regex untuk mencacah SMS. SMS yang diterima disimpan dalam *field* text dalam tabel `inbox` oleh `smsd`. *Field* tersebut

kemudian dimasukkan ke dalam *string* bernama `sms_masukan`. *String* `sms_masukan` kemudian dipisahkan berdasarkan spasi menggunakan pola `split(/ /, $sms_masukan)` dan dimasukkan ke dalam *array* lokal bernama `sms_tercacah`. Di dalam elemen masih terdapat karakter `#`, `*`, dan `=`. Untuk itu dalam setiap elemen `sms_tercacah` perlu dihilangkan karakter `#`, `*`, dan `=`. Hal ini dilakukan menggunakan pola regex:

```
$kata =~ s/#//;  
$kata =~ s/\*//;  
$kata =~ s/=//;
```

Setelah setiap elemen *array* `sms_tercacah` sesuai dengan tujuan pencacahan, maka elemen-elemen tersebut dapat dimasukkan ke dalam *query* transaksi pada basis data `warunglinux`.

5.4.3 Pemrosesan Kata Kunci pada Media SMS

Kata kunci (*password*) yang terdapat pada SMS akan dienkripsi menjadi sebuah bentuk baru. Bentuk baru ini nantinya akan dicocokkan dengan *password* yang terdapat dalam basis data `warunglinux`.

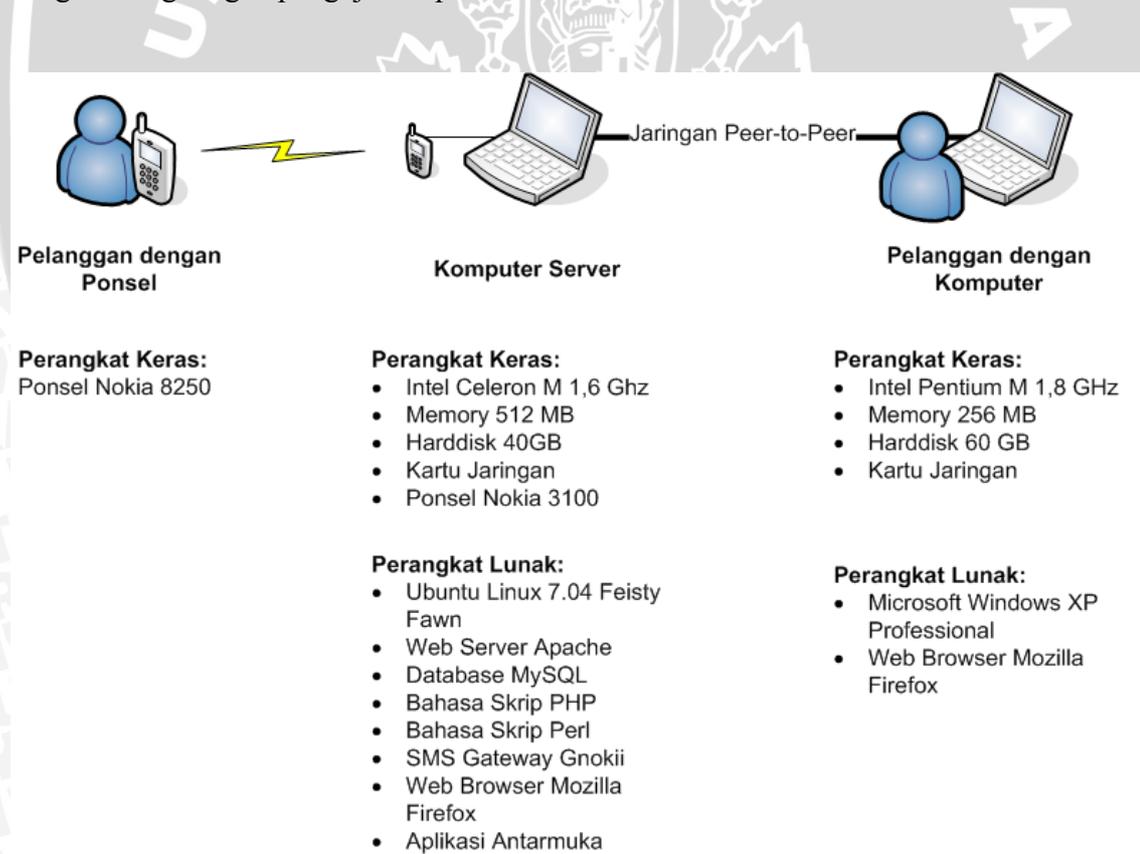
Proses enkripsi pada *password* yang terdapat pada SMS menggunakan metode MD5. Metode ini digabungkan dengan penggunaan *salt*. *Salt* adalah suatu kalimat acak yang nantinya akan dimasukkan 2 karakter pertama saja ke dalam *password* yang telah dienkripsi.

Password hasil penggabungan dari *password* terenkripsi dengan *salt* dienkripsikan lagi menggunakan metode MD5. *Password* terakhir ini yang nantinya akan dicocokkan dengan *password* pelanggan yang tersimpan dalam basis data `warunglinux`.

BAB VI PENGUJIAN DAN ANALISIS

Dalam bab ini akan dijelaskan mengenai pengujian dan analisis yang dilakukan terhadap tugas akhir *Pengembangan Media Transaksi Menggunakan SMS (Short Message Service) Untuk Belanja Online*. Pengujian dilakukan terhadap aplikasi toko *online*, aplikasi *SMS Gateway*, dan aplikasi antarmuka. Lingkungan pengujian dilakukan pada dua komputer yang terhubung melalui jaringan *peer-to-peer* menggunakan media kabel. Salah satu komputer berfungsi sebagai *server* sedangkan yang lain berfungsi sebagai *client*. Selain penggunaan komputer, pengujian juga dilakukan dengan penggunaan ponsel sebagai *client*.

Sistem operasi yang digunakan dalam pengujian adalah menggunakan Ubuntu Linux 7.04 Feisty Fawn pada *server* dan Microsoft Windows XP Professional pada *client*. Diagram lingkungan pengujian diperlihatkan dalam Gambar 6.1 di bawah.



Gambar 6.1. Blok Diagram Lingkungan Pengujian Sistem.
Sumber: [Pengujian]

Pengujian pada *server* dilakukan bertujuan untuk mengetahui:

- Aplikasi WarungLINUX dapat melakukan proses administrasi.
- Aplikasi WarungLINUX dapat melakukan proses administrasi barang.
- Aplikasi WarungLINUX dapat menangani transaksi melalui media Internet.
- Aplikasi SMS Gateway dapat berkomunikasi dengan ponsel yang terpasang pada *server* dan berfungsi sebagai *modem data*.
- Aplikasi SMS Gateway dapat berkomunikasi dengan basis data MySQL untuk penyimpanan SMS.
- Aplikasi SMS dapat mengirim pesan kepada pengirim memberikan konfirmasi bahwa transaksi telah diproses.
- Aplikasi antarmuka dapat mengubah data yang berasal dari basis data milik SMS Gateway menjadi data milik basis data toko *online*.

Pengujian pada *client* dilakukan untuk mengetahui bahwa:

- *Client* dapat melakukan transaksi pada toko *online* melalui media Internet.
- *Client* dapat melakukan administrasi pada toko *online* melalui media Internet.

6.1 Pengujian Administrasi WarungLINUX Melalui Media Internet

Pengujian administrasi dilakukan pada komputer *server*. Pada komputer tersebut, dilakukan proses administrasi pada aplikasi WarungLINUX.

Tujuan dari pengujian ini adalah untuk memeriksa apakah aplikasi WarungLINUX dapat berjalan dengan baik dan dapat menangani administrasi melalui media Internet.

6.1.1 Pengujian Akses Administrasi

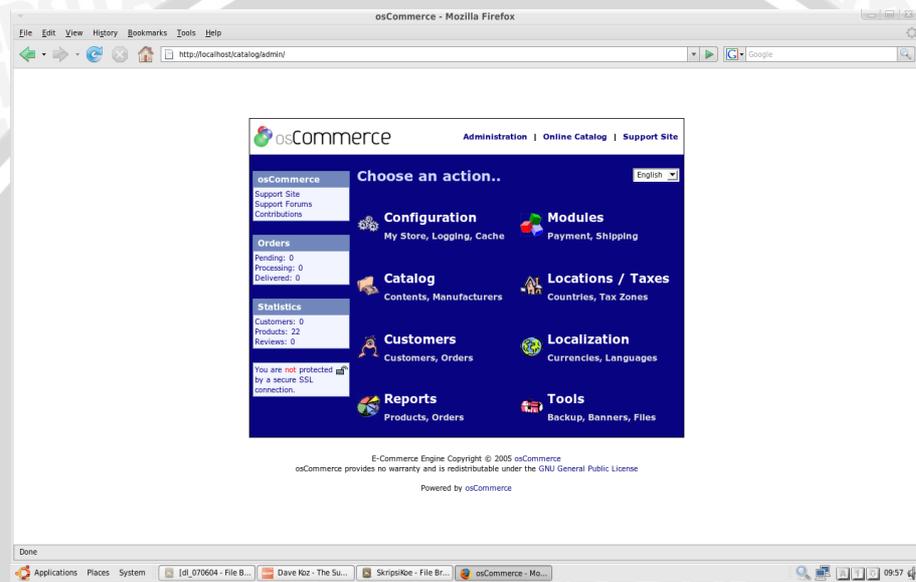
Pengujian dilakukan apakah administrator dapat mengakses halaman administrasi yang terdapat pada aplikasi WarungLINUX. Kemudian pengujian juga dilakukan apakah administrator dapat mengakses halaman-halaman dari administrasi aplikasi WarungLINUX.

A. Tujuan Pengujian

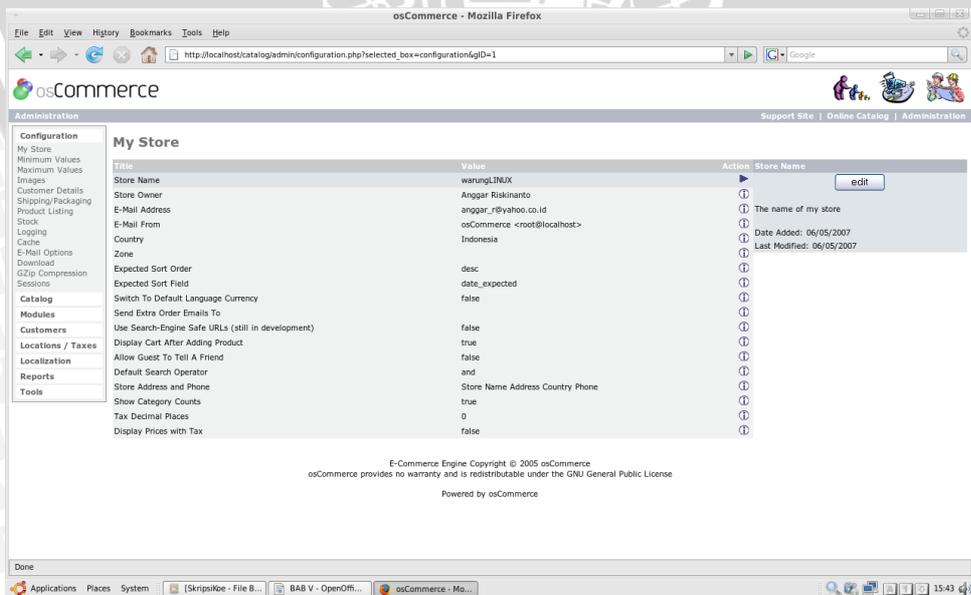
Tujuan dari pengujian adalah untuk mengetahui apakah administrator dapat mengakses halaman-halaman administrasi yang terdapat pada aplikasi WarungLINUX.

B. Prosedur Pengujian

Administrator membuka halaman administrasi menggunakan *web browser*. Pada *web browser*, administrator mengakses alamat `http://localhost/catalog/admin`. Setelah halaman administrasi terbuka, administrator membuka beberapa bagian dari halaman administrasi, seperti Configuration, Catalog, Modules, dan Customers melalui *link-link* yang tersedia pada halaman administrasi.

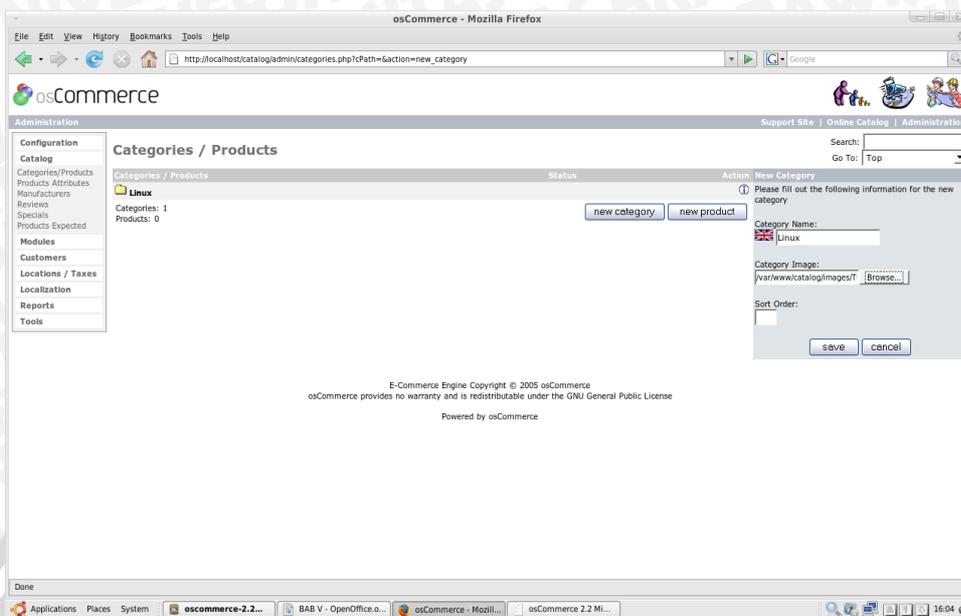


Gambar 6.2. Halaman Administrasi Aplikasi WarungLINUX.
Sumber: [Pengujian]

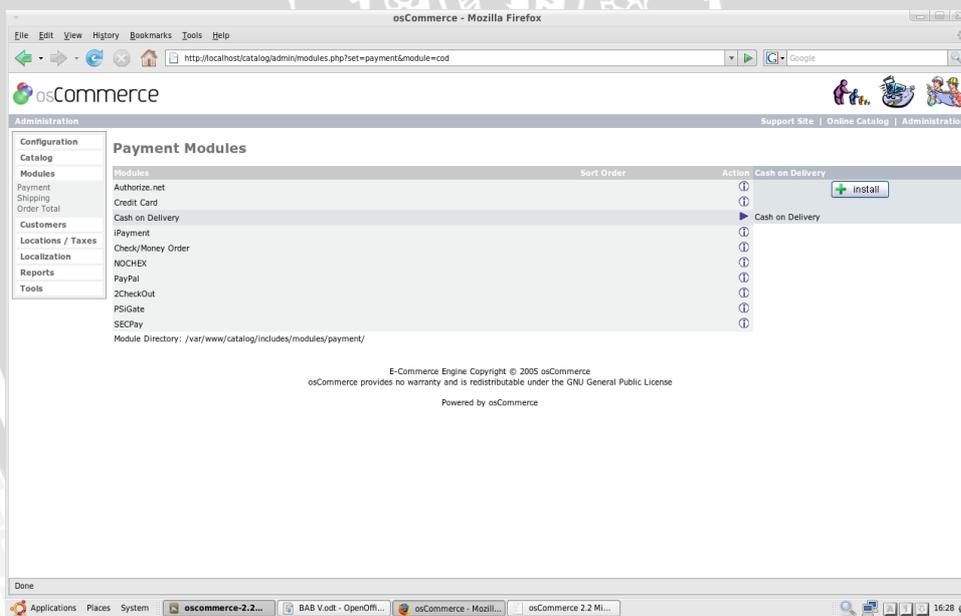


Gambar 6.3. Halaman Configuration pada Aplikasi WarungLINUX.
Sumber: [Pengujian]



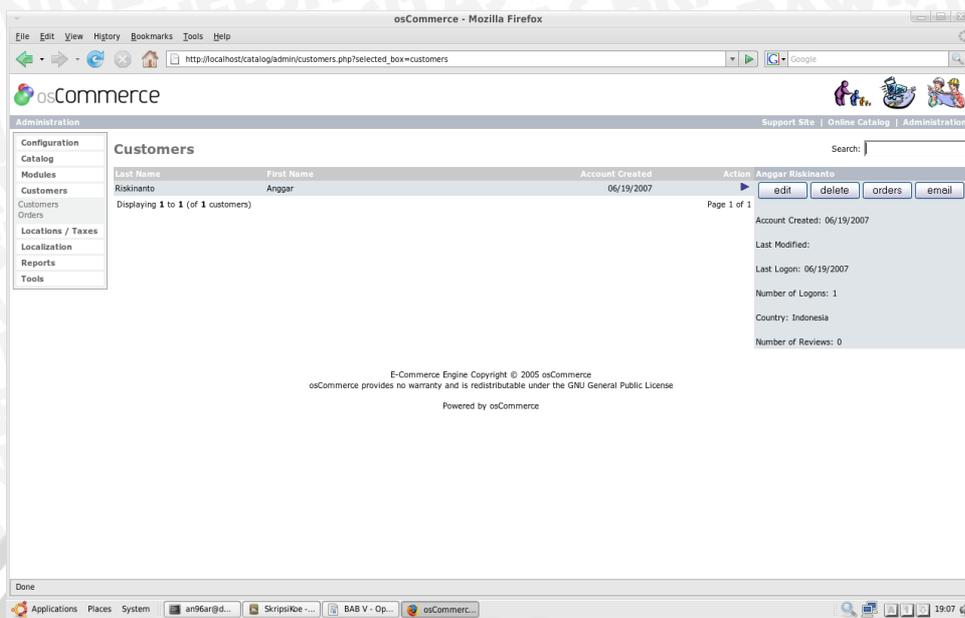


Gambar 6.4. Halaman Catalog pada Aplikasi WarungLINUX.
Sumber: [Pengujian]



Gambar 6.5. Halaman Modules pada Aplikasi WarungLINUX.
Sumber: [Pengujian]





Gambar 6.6. Halaman Customers pada Aplikasi WarungLINUX.
Sumber: [Pengujian]

C. Hasil Pengujian

Hasil dari pengujian terlihat dalam Gambar 6.2 sampai 6.6. Dalam gambar-gambar tersebut diperlihatkan halaman-halaman administrasi yang dapat dibuka oleh seorang administrator. Pada hasil pengujian diperlihatkan sub bagian administrasi yang telah dibuka, yaitu Configuration, Catalog, Modules, dan Customers.

D. Analisis Pengujian

Hasil pengujian akses administrasi diperlihatkan dalam Gambar 6.2 hingga Gambar 6.6. Dalam gambar tersebut diperlihatkan bagian-bagian yang dapat diakses oleh administrator dalam sebuah proses administrasi pada aplikasi WarungLINUX.

E. Kesimpulan Pengujian

Dapat disimpulkan bahwa halaman administrasi dapat diakses oleh administrator melalui alamat <http://localhost/catalog/admin>.

6.1.2 Pengujian Penambahan Kategori/Barang

Pengujian Penambahan Kategori/Barang dilakukan pada aplikasi WarungLINUX. Pengujian dilakukan oleh administrator pada halaman Catalog. Pengujian dilakukan dengan menambah atau membuat kategori baru yaitu Damn Small Linux. Kemudian di

dalam kategori tersebut dibuat produk baru yang bernama Damn Small Linux 3.3.

Pengujian dilakukan dengan membandingkan data pada aplikasi WarungLINUX dengan data yang terdapat pada basis data `warunglinux`.

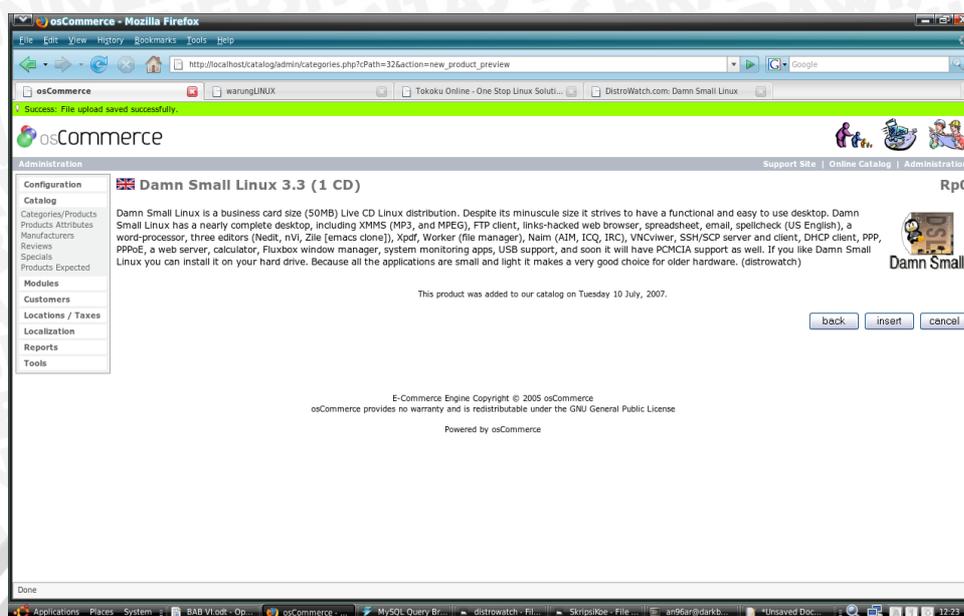
A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat digunakan untuk menambahkan kategori atau barang ke dalam basis data `warunglinux`.

B. Prosedur Pengujian

Pada komputer *server*, halaman administrasi dibuka melalui *web browser* oleh administrator dan diakses melalui alamat `http://localhost/catalog/admin`. Administrator memasukkan kategori dan barang baru melalui bagian `Catalog`. Administrator kemudian menambahkan kategori baru, yaitu `Damn Small Linux`. Setelah kategori `Damn Small Linux` dibuat, kemudian administrator membuat barang baru yaitu `Damn Small Linux 3.3`. Seperti diperlihatkan dalam Gambar 6.7. Kemudian administrator meng-klik *link* `Preview`. Setelah selesai, maka *link* `Insert` diklik untuk benar-benar memasukkan barang ke dalam katalog aplikasi WarungLINUX.

Sebelum perubahan dilakukan, administrator menjalankan program `mysql` melalui program `Terminal` yang dijalankan melalui `Applications | Accessories | Terminal`. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program `mysql` yang menjalankan perintah `select` pada tabel `categories`, `categories_description`, dan `products` dalam basis data `warunglinux`.



Gambar 6.7. Halaman *Preview* Barang.
Sumber: [Pengujian]

```
mysql> select * from categories;
Empty set (0.02 sec)

mysql> select * from categories_description;
Empty set (0.03 sec)

mysql> select * from categories;
+-----+-----+-----+-----+-----+-----+
| categories_id | categories_image | parent_id | sort_order | date_added | last_modified |
+-----+-----+-----+-----+-----+-----+
| 32 | damnsmall.png | 0 | 0 | 2007-07-10 12:18:17 | NULL |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from categories_description;
+-----+-----+-----+
| categories_id | language_id | categories name |
+-----+-----+-----+
| 32 | 1 | Damn Small Linux |
+-----+-----+-----+
1 row in set (0.01 sec)
```

Gambar 6.8. Tabel *categories* dan *categories_description* Sebelum dan Sesudah Diisi Data.
Sumber: [Pengujian]

```
mysql> select * from products;
Empty set (0.02 sec)

mysql> select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;
+-----+-----+-----+-----+-----+-----+
| products_id | products_quantity | products_image | products_price | products_date_added | products_weight |
+-----+-----+-----+-----+-----+-----+
| 50 | 100 | damnsmall.png | 0.0000 | 2007-07-10 12:23:27 | 0.05 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

Gambar 6.9. Tabel *products* Sebelum dan Sesudah Diisi Data.
Sumber: [Pengujian]



C. Hasil Pengujian

Dalam Gambar 6.8 terlihat dalam tabel `categories` bahwa kategori baru belum dimasukkan. Hal ini terlihat dengan menggunakan perintah `select * from categories` yang menghasilkan keluaran `Empty Set (0.02 sec)`. Hal yang sama juga terjadi pada tabel `categories_description`. Setelah data dimasukkan seperti terlihat dalam Gambar 6.7, maka terdapat perubahan pada tabel `categories` dan `categories_description`. Pada tabel `categories` terdapat field baru bernilai 32 pada kolom `products_id`. Pada tabel tersebut juga terdapat data gambar kategori yaitu `damnsmall.png` dan kapan data tersebut ditambahkan ke dalam katalog yaitu `2007-07-10 12:18:17`. Hal yang sama juga terjadi pada tabel `categories_description`, dimana terdapat data baru pada `categories_id` yang bernilai 32 dan `language_id` yang bernilai 1 serta `categories_name` yang bernilai `Damn Small Linux`.

Dalam Gambar 6.5 terlihat ketika data barang belum dimasukkan dalam aplikasi WarungLINUX terlihat pada keluaran yang menghasilkan `Empty set (0.02 sec)`. Kemudian setelah barang baru dimasukkan ke dalam aplikasi WarungLINUX, maka perintah `select products_id, products_quantity, products_image, products_price, products_date_added, products_weight from products;` menghasilkan keluaran seperti yang terlihat dalam Gambar 6.9. Terdapat nilai baru pada kolom `products_id` yaitu 50, kolom `products_quantity` yang bernilai 100 dan `products_image` yaitu `damnsmall.png`. Pada kolom `products_date_added` juga bernilai `2007-07-10 12:23:27` dan kolom `products_weight` yang bernilai 0.05.

D. Analisis Pengujian

Hasil pengujian penambahan kategori atau barang diperlihatkan dalam Gambar 6.8 dan Gambar 6.9. Dalam gambar tersebut diperlihatkan tabel `categories` dan `categories_description` yang belum berisi data. Dalam gambar tersebut juga terlihat perubahan yang terjadi setelah aplikasi WarungLINUX diisi oleh data. Hal yang sama juga terlihat pada tabel `products` dimana terlihat perbedaan isi tabel sebelum dan sesudah diisi data pada aplikasi WarungLINUX.

E. Kesimpulan Pengujian

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa proses penambahan kategori dan barang dapat dilakukan melalui aplikasi WarungLINUX. Data yang ditambahkan juga tersimpan dalam basis data `warunglinux`.

6.1.3 Pengujian Perubahan Kategori/Barang

Pengujian perubahan kategori atau barang dilakukan dengan mengubah data barang yang telah ada pada aplikasi WarungLINUX. Data tersebut diubah dan dilihat perubahannya pada basis data `warunglinux`.

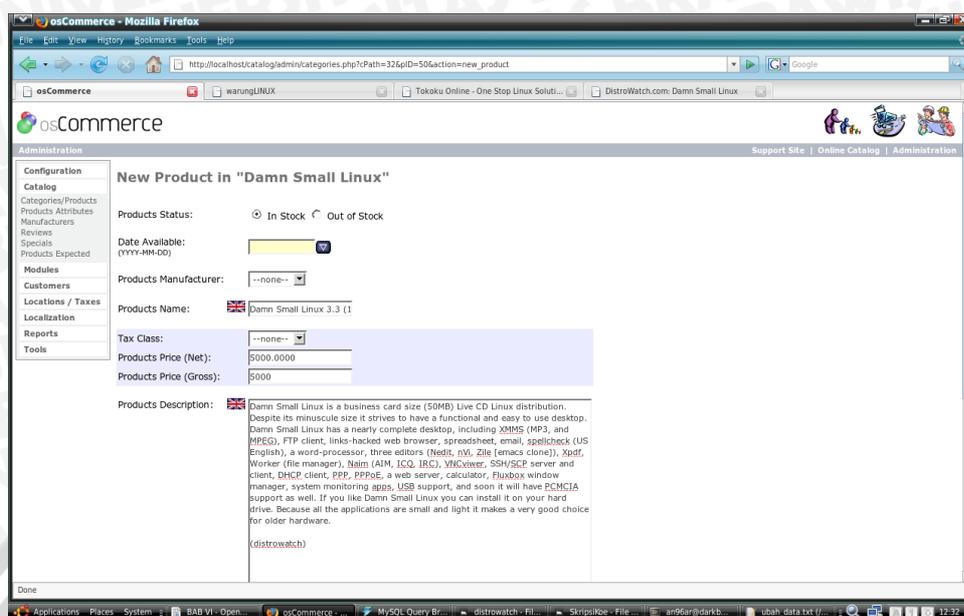
A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat digunakan untuk mengubah data yang ada melalui halaman administrasi. Kemudian hasil dari perubahan data tersebut disimpan dalam basis data `warunglinux`.

B. Prosedur Pengujian

Pada komputer *server*, administrator membuka halaman administrasi menggunakan *web browser* dan mengakses alamat `http://localhost/catalog/admin`. Kemudian mengubah barang yang seperti diperlihatkan dalam Gambar 6.10. Barang yang diubah adalah `Damn Small Linux 3.3`. Perubahan dilakukan dengan mengubah harga dari 0 menjadi 5000. Setelah selesai, kemudian meng-klik *link* `Preview`. Setelah yakin, maka administrator meng-klik *link* `Insert`.

Sebelum perubahan dilakukan, administrator menjalankan program `mysql` melalui program `Terminal` yang dijalankan melalui `Applications | Accessories | Terminal`. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program `mysql` yang menjalankan perintah `select` pada tabel `products` dalam basis data `warunglinux`.



Gambar 6.10. Halaman Barang yang Diubah.
Sumber: [Pengujian]

```
mysql> select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;
+-----+-----+-----+-----+-----+-----+
| products_id | products_quantity | products_image | products_price | products_date added | products weight |
+-----+-----+-----+-----+-----+-----+
|          50 |             100 | damnsmall.png |         0.0000 | 2007-07-10 12:23:27 |             0.05 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)

mysql> select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;
+-----+-----+-----+-----+-----+-----+
| products_id | products_quantity | products_image | products_price | products_date added | products weight |
+-----+-----+-----+-----+-----+-----+
|          50 |             100 | damnsmall.png |        5000.0000 | 2007-07-10 12:23:27 |             0.05 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.11. Tabel products Sebelum dan Sesudah Diubah Datanya.
Sumber: [Pengujian]

C. Hasil Pengujian

Hasil pengujian pengubahan kategori atau barang diperlihatkan dalam Gambar 6.11. Dalam gambar diperlihatkan tabel products yang telah berisi data dan yang telah diubah isi datanya. Pada pengujian, isi kolom products_price berubah nilainya dari 0.0000 menjadi 5000.0000.

D. Analisis Pengujian

Pada hasil pengujian diperlihatkan perubahan nilai harga barang yang semula bernilai 0 menjadi 5000. Perubahan dilakukan melalui aplikasi WarungLINUX. Kemudian perubahan yang terjadi dilihat pada tabel products. Pada tabel products perubahan yang terjadi ada pada kolom products_price.

E. Kesimpulan Pengujian

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi WarungLINUX dapat memproses perubahan data melalui halaman administrasi. Perubahan tersebut tersimpan dalam basis data `warunglinux`.

6.1.4 Pengujian Penghapusan Kategori/Barang

Pengujian penghapusan kategori atau barang dilakukan dengan menghapus data barang yang telah ada pada aplikasi WarungLINUX. Data tersebut diubah dan dilihat perubahannya pada basis data `warunglinux`.

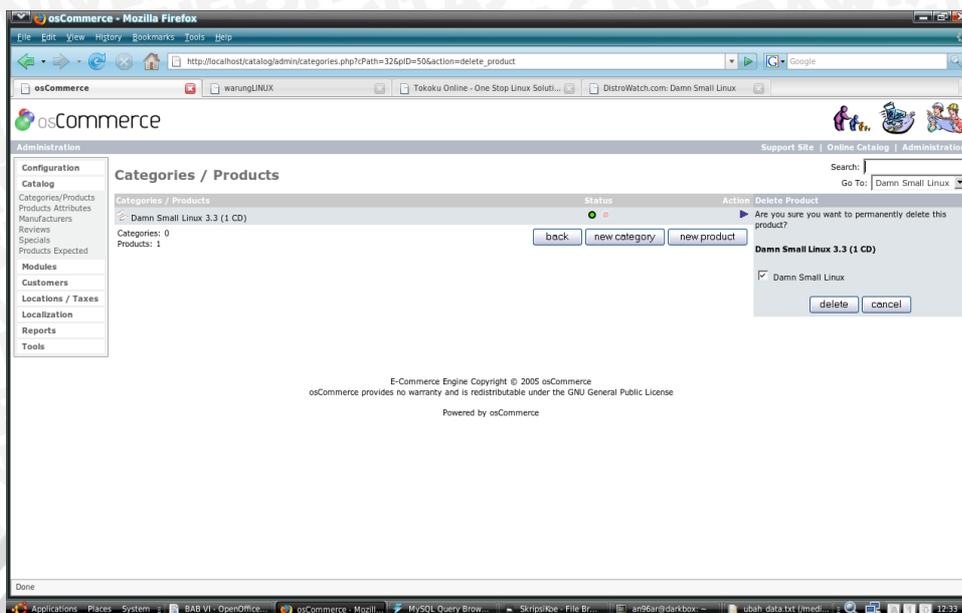
A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat digunakan untuk menghapus data yang ada melalui halaman WarungLINUX. Kemudian hasil dari penghapusan data tersebut disimpan ke dalam basis data `warunglinux`.

B. Prosedur Pengujian

Pada komputer *server*, administrator membuka halaman administrasi menggunakan *web browser* dan mengakses alamat `http://localhost/catalog/admin`. Kemudian menghapus kategori dan barang yang seperti diperlihatkan dalam Gambar 6.12. Setelah selesai, kemudian meng-klik *link* Delete.

Sebelum perubahan dilakukan, administrator menjalankan program `mysql` melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program `mysql` yang menjalankan perintah `select` pada tabel `products` dalam basis data `warunglinux`.



Gambar 6.12. Halaman Barang yang Akan Dihapus.
Sumber: [Pengujian]

```
mysql> select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;
+-----+-----+-----+-----+-----+-----+
| products_id | products_quantity | products_image | products_price | products_date added | products_weight |
+-----+-----+-----+-----+-----+-----+
|          50 |                100 | damnsmall.png |      5000.0000 | 2007-07-10 12:23:27 |             0.05 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;
Empty set (0.00 sec)
```

Gambar 6.13. Tabel products Sebelum dan Sesudah Dihapus Datanya.
Sumber: [Pengujian]

C. Hasil Pengujian

Hasil pengujian penghapusan kategori atau barang diperlihatkan dalam Gambar 6.13. Dalam gambar tersebut diperlihatkan tabel `products` yang telah berisi data dan yang telah dihapus isi datanya. Ketika data belum dihapus tabel `products` masih berisi data barang. Setelah barang dihapus melalui aplikasi WarungLINUX, maka tabel `products` tidak mempunyai data dalam kolom-kolomnya. Hal ini terlihat dari keluaran yang berupa `Empty set (0.00 sec)` yang berasal dari perintah `select products_id,products_quantity,products_image,products_price,products_date_added,products_weight from products;`



D. Analisis Pengujian

Dari hasil pengujian diperlihatkan ketika tabel masih berisi data barang. Kemudian barang tersebut dihapus melalui aplikasi WarungLINUX yang diperlihatkan dalam Gambar

E. Kesimpulan Pengujian

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi WarungLINUX dapat menghapus data melalui halaman administrasi. Data yang dihapus dapat tersimpan pada basis data `warunglinux`.

6.2 Pengujian Transaksi WarungLINUX Melalui Media Internet

Pengujian dilakukan pada komputer *client*. Pada komputer tersebut, dilakukan transaksi pada aplikasi WarungLINUX. Transaksi yang telah dilakukan kemudian dikirim ke komputer *server*.

Tujuan dari pengujian ini adalah untuk memeriksa apakah aplikasi WarungLINUX telah berjalan dengan baik dan dapat menangani transaksi yang dilakukan *client* melalui media Internet.

6.2.1 Pengujian Registrasi Pelanggan

Pengujian registrasi pelanggan dilakukan dengan menggunakan aplikasi WarungLINUX. Pada aplikasi WarungLINUX, pengunjung mendaftarkan dirinya pada bagian `My Account`, tepatnya pada kolom `New Customer`.

A. Tujuan Pengujian

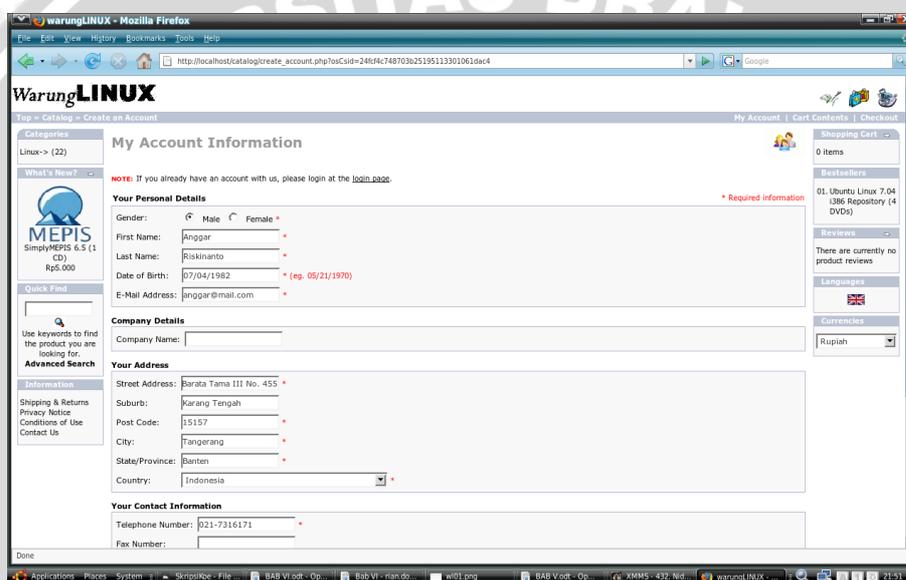
Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat menangani proses registrasi pelanggan. Pengujian juga bertujuan apakah proses registrasi menyimpan datanya pada basis data `warunglinux`.

B. Prosedur Pengujian

Pelanggan membuka *web browser* dan mengakses alamat `http://192.168.0.1/catalog/`. Kemudian meng-klik *link* `My Account` atau *link* `create an account` pada halaman utama. Kemudian meng-klik *link* `Continue` pada kolom `New Customer`.

Pada halaman My Account Information, pelanggan mengisi data yang diperlukan seperti yang diperlihatkan dalam Gambar 6.14. Setelah selesai mengisi, pelanggan dapat meng-klik *link* Continue.

Sebelum perubahan dilakukan, administrator menjalankan program mysql melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program mysql yang menjalankan perintah select pada tabel customers dalam basis data warunglinux.



Gambar 6.14. Halaman My Account Information.
Sumber: [Pengujian]

```
mysql> select * from customers;
Empty set (0.00 sec)

mysql> select customers_id,customers_gender,customers_firstname,customers_lastname,customers_email_address from
customers;
+-----+-----+-----+-----+-----+
| customers_id | customers_gender | customers_firstname | customers_lastname | customers_email_address |
+-----+-----+-----+-----+-----+
| 3 | m | Anggar | Riskinanto | anggar@mail.com |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.15. Tabel customers Sebelum dan Sesudah Diisi Data.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.15 diperlihatkan tabel customers ketika belum ada pelanggan yang meregistrasikan dirinya. Dalam gambar itu juga diperlihatkan tabel customers setelah pelanggan telah meregistrasikan dirinya. Sebelum pelanggan meregistrasikan

dirinya, tabel `customers` tidak mempunyai data apapun. Setelah pelanggan mendaftarkan dirinya, tabel `customers` berisi data yang telah diisikan oleh pelanggan. Pada kolom `customers_id` bernilai 3, kolom `customers_gender` bernilai m, kolom `customers_firstname` bernilai Anggar, kolom `customers_lastname` bernilai Riskianto, dan kolom `customers_email_address` bernilai `anggar@mail.com`.

D. Analisis Pengujian

Dari hasil pengujian diperlihatkan perubahan pada tabel `customers` ketika pelanggan mendaftarkan dirinya. Sebelum pelanggan mendaftarkan dirinya, tabel `customers` tidak berisi data apapun. Setelah pelanggan mendaftarkan dirinya, tabel `customers` berisi data yang telah dimasukkan oleh pelanggan melalui aplikasi WarungLINUX.

E. Kesimpulan Pengujian

Dari pengujian dapat disimpulkan bahwa aplikasi WarungLINUX dapat menangani proses registrasi yang dilakukan oleh pelanggan. Data proses registrasi tersebut dapat disimpan dalam basis data `warunglinux`.

6.2.2 Pengujian *Login* Pelanggan

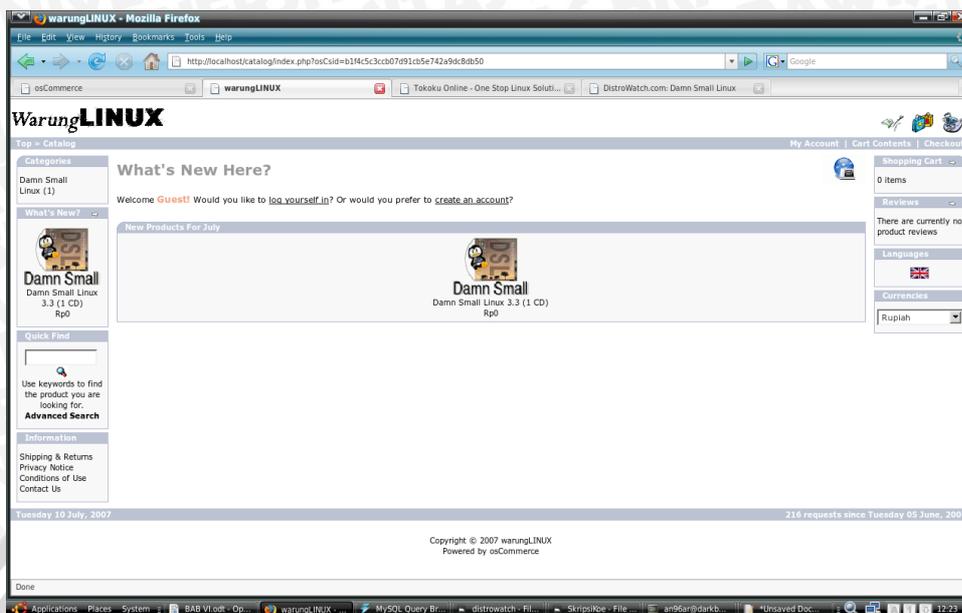
Pengujian *login* pelanggan dilakukan pada aplikasi WarungLINUX. Proses pengujian dilakukan dengan melakukan *login* pada aplikasi WarungLINUX melalui halaman Sign In.

A. Tujuan Pengujian

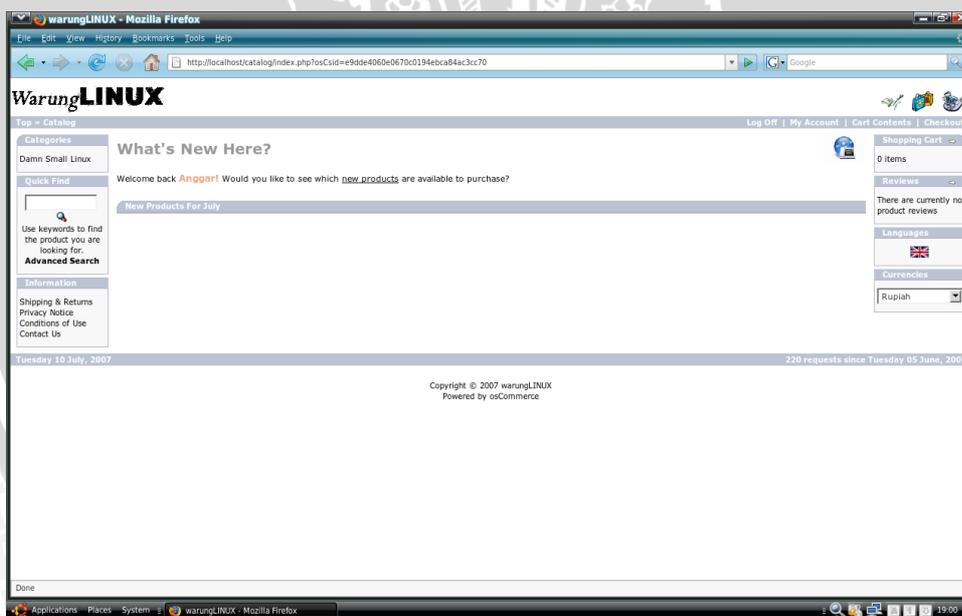
Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat menangani proses *login* yang dilakukan oleh pelanggan.

B. Prosedur Pengujian

Pelanggan mengakses alamat `http://192.168.0.1/catalog/` melalui *web browser*. Kemudian pelanggan meng-klik My Account untuk membuka halaman Sign In. Pada halaman tersebut, pelanggan memasukkan alamat *e-mail* `anggar@mail.com` dan *password* apaaja. Kemudian setelah pelanggan *login*, pelanggan *logout* dengan meng-klik *link* Log Off.



Gambar 6.16. Halaman Utama Sebelum Pelanggan Login.
Sumber: [Pengujian]



Gambar 6.17. Halaman Utama Setelah Pelanggan Login.
Sumber: [Pengujian]

C. Hasil Pengujian

Halaman ketika pelanggan belum melakukan *login* diperlihatkan dalam Gambar 6.16. Ketika pelanggan telah melakukan proses *login* diperlihatkan dalam Gambar 6.17. Dalam gambar tersebut diperlihatkan bahwa pelanggan dapat *login* setelah memasukkan *e-mail* dan *password* terlebih dahulu.



D. Analisis Pengujian

Dari hasil pengujian diperlihatkan perubahan pada halaman utama ketika pelanggan belum melakukan proses *login* dan setelah pelanggan melakukan proses *login*. Proses *login* dilakukan dengan memasukkan alamat *e-mail* dan *password* pada halaman Sign In.

E. Kesimpulan Pengujian

Dapat disimpulkan bahwa aplikasi WarungLINUX dapat menangani proses *login* oleh pelanggan dengan baik.

6.2.3 Pengujian Pemilihan Barang Oleh Pelanggan

Pengujian pemilihan barang merupakan pengujian dimana pelanggan memilih barang dan memasukkan barang tersebut dalam *Cart*. Pemilihan barang dilakukan pada aplikasi WarungLINUX.

A. Tujuan Pengujian

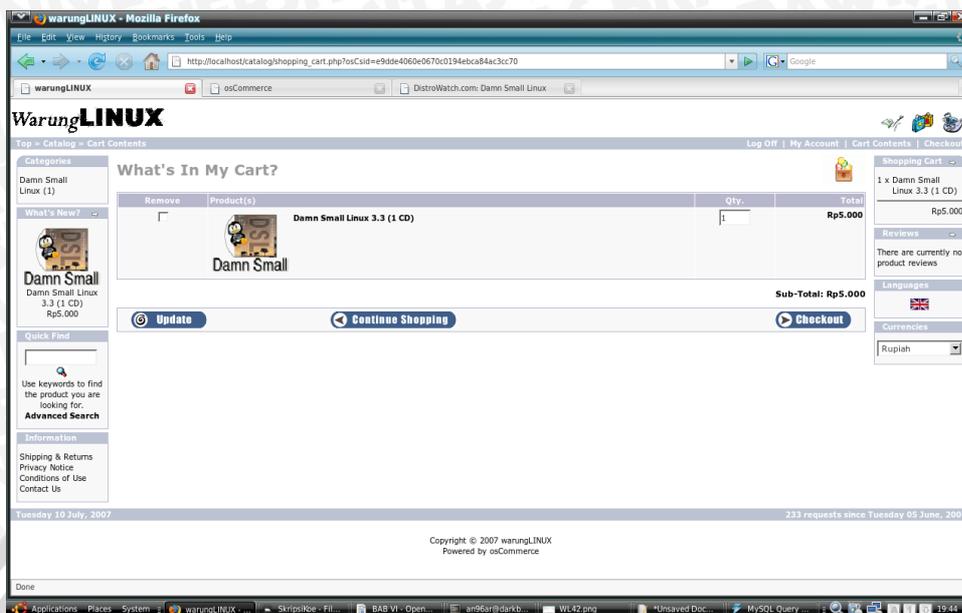
Pengujian dilakukan untuk mengetahui apakah WarungLINUX dapat menangani ketika pelanggan memilih barang yang akan dibeli. Pengujian juga bertujuan untuk mengetahui apakah barang yang sudah dipilih tersimpan dalam basis data *warunglinux*.

B. Prosedur Pengujian

Pelanggan membuka *web browser* dan mengakses alamat <http://1962.168.0.1/catalog> untuk membuka aplikasi WarungLINUX. Setelah halaman WarungLINUX terbuka, pelanggan melakukan proses *login*. Setelah *login*, pelanggan memilih barang yang ada pada katalog. Setelah melakukan proses memilih barang, pelanggan melihat isi *Cart* melalui *link* *Cart Contents*. Isi dari *Cart Contents* diperlihatkan dalam Gambar 6.18.

Setelah *Cart Contents* terisi oleh barang yang telah dipilih, pelanggan *logout* dengan meng-klik *link* *Log Off*.

Sebelum perubahan dilakukan, administrator menjalankan program *mysql* melalui program *Terminal* yang dijalankan melalui *Applications | Accessories | Terminal*. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program *mysql* yang menjalankan perintah *select* pada tabel *customers_basket*.



Gambar 6.18. Halaman Cart Contents.
Sumber: [Pengujian]

```
mysql> select customers_id,customers_id,customers_id,customers_id from
customers_id;
Empty set (0.00 sec)

mysql> select customers_id,customers_id,customers_id,customers_id from
customers_id;
+-----+-----+-----+-----+
| customers_id | customers_id | customers_id | customers_id |
+-----+-----+-----+-----+
| 5 | 3 | 1 | 20070710 |
+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

Gambar 6.19. Tabel customers_id Sebelum dan Sesudah Diisi Data.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.18 diperlihatkan isi dari Cart yang ada pada Cart Contents. Isi tersebut disimpan dalam tabel customers_id. Dalam gambar 6.19 juga diperlihatkan tabel customers_id ketika belum diisi data dan ketika telah diisi oleh data. Ketika barang belum dimasukkan ke dalam Cart, tabel customers_id tidak berisi data apapun. Ketika Cart telah terisi oleh barang, tabel customers_id terisi data barang. Pada kolom customers_id bernilai 5, kolom customers_id bernilai 3, kolom customers_id bernilai 1, dan kolom customers_id bernilai 20070710.

D. Analisis Pengujian

Pada hasil pengujian terlihat perbedaan ketika Cart belum diisi maupun telah diisi. Perubahan tersebut terlihat pada tabel `customers_basket`. Ketika Cart belum terisi, tabel `customers_basket` tidak berisi data apapun. Setelah Cart terisi, tabel `customers_basket` berisi data barang yang telah dimasukkan ke dalam Cart.

E. Kesimpulan Pengujian

Dapat disimpulkan bahwa aplikasi WarungLINUX dapat menangani ketika pelanggan memilih barang yang akan dipilih dan disimpan ke dalam basis data `warunglinux`.

6.2.4 Pengujian Pembelian Barang Oleh Pelanggan

Pengujian pembelian barang merupakan pengujian dimana pelanggan melakukan proses pembelian barang yang terdapat dalam Cart. Pembelian barang dilakukan pada aplikasi WarungLINUX.

A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat menangani ketika pelanggan akan membeli barang yang telah dipilih. Pengujian juga bertujuan apakah proses pembelian tersimpan dalam basis data `warunglinux`.

B. Prosedur Pengujian

Pelanggan membuka aplikasi WarungLINUX dengan mengakses alamat `http://192.168.0.1/catalog` melalui *web browser*. Setelah halaman WarungLINUX terbuka, pelanggan login dengan memasukkan alamat *e-mail* dan *password*. Setelah *login*, pelanggan mengakses halaman pembelian dengan meng-klik *link* Checkout.

Setelah melakukan rangkaian proses pembelian, pelanggan melakukan logout dengan meng-klik *link* Log Off.

Sebelum perubahan dilakukan, administrator menjalankan program `mysql` melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program `mysql` yang menjalankan perintah `select` pada tabel `orders_products` dan `orders` dalam basis data `warunglinux`.

```
mysql> select * from orders_products;
Empty set (0.00 sec)

mysql> select * from orders;
Empty set (0.00 sec)

mysql> select orders_products_id,orders_id,products_id,products_name,products_quantity from orders_products;
+-----+-----+-----+-----+-----+
| orders_products_id | orders_id | products_id | products_name | products_quantity |
+-----+-----+-----+-----+-----+
| 3 | 3 | 51 | Damn Small Linux 3.3 (1 CD) | 1 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select orders_id,customer_id,delivery_name,payment_method,date_purchased,orders_status from orders;
+-----+-----+-----+-----+-----+-----+
| orders_id | customer_id | delivery_name | payment_method | date_purchased | orders_status |
+-----+-----+-----+-----+-----+-----+
| 3 | 3 | Anggar Riskinanto | Check/Money Order | 2007-07-10 19:59:59 | 1 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.20. Tabel `orders_products` dan `orders` Sebelum dan Sesudah Diisi Data.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.20 diperlihatkan tabel-tabel yang berhubungan dengan pembelian barang. Dalam gambar tersebut juga diperlihatkan tabel-tabel ketika belum diisi data dan setelah diisi oleh data. Ketika proses pembelian belum terjadi, tabel `orders_products` dan `orders` belum terisi oleh data apapun. Ketika proses pembelian sudah terjadi, tabel `orders_products` telah berisi data, yaitu kolom `orders_products_id` bernilai 3, kolom `orders_id` bernilai 3, kolom `products_id` bernilai 51, kolom `products_name` bernilai Damn Small Linux 3.3 (1 CD), dan kolom `products_quantity` bernilai 1. Sedangkan tabel `orders` telah berisi data, yaitu kolom `orders_id` bernilai 3, kolom `customer_id` bernilai 3, kolom `delivery_name` bernilai Anggar Riskinanto, kolom `payment_method` bernilai Check/Money Order, kolom `date_purchased` bernilai 2007-07-10 19:59:59, dan kolom `orders_status` bernilai 1.

D. Analisis Pengujian

Dari hasil pengujian bisa terlihat perubahan yang terjadi ketika pelanggan belum membeli barang yang telah ia pilih. Pada tabel `orders_products` dan `orders` belum terisi produk apapun. Ketika pelanggan telah membeli barang yang telah ia pilih, tabel `orders_products` dan `orders` telah berisi data.

E. Kesimpulan Pengujian

Dari pengujian dapat disimpulkan bahwa aplikasi WarungLINUX dapat menangani ketika pelanggan membeli barang yang telah dipilih dan disimpan ke dalam basis data.

6.2.5 Pengujian Batasan Nilai Transaksi WarungLINUX

Pengujian dilakukan untuk memeriksa batasan nilai transaksi yang dapat diproses oleh aplikasi WarungLINUX. Pengujian dilakukan dengan memesan barang yang melebihi stok pada aplikasi WarungLINUX.

Tujuan dari pengujian ini adalah untuk memastikan bahwa aplikasi WarungLINUX dapat memproses transaksi oleh pelanggan dengan baik.

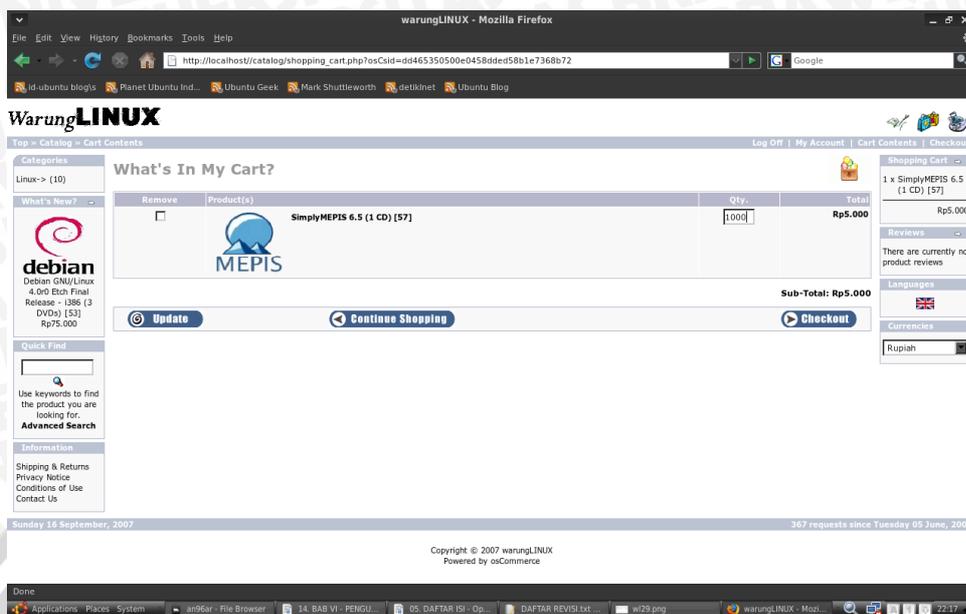
A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi WarungLINUX dapat menangani transaksi oleh pelanggan yang jumlah barangnya melebihi stok yang tersedia.

B. Prosedur Pengujian

Pelanggan membuka *web browser* dan mengakses alamat <http://1962.168.0.1/catalog> untuk membuka aplikasi WarungLINUX. Setelah halaman WarungLINUX terbuka, pelanggan melakukan proses *login*. Setelah *login*, pelanggan memilih barang yang ada pada katalog. Ketika melakukan pemilihan barang, pelanggan memasukkan jumlah barang dalam kolom *Qty* sebanyak 1000 seperti yang diperlihatkan dalam Gambar 6.21. Setelah melakukan proses memilih barang, pelanggan melihat isi *Cart* melalui *link* *Cart Contents*.

Setelah *Cart Contents* terisi oleh barang yang telah dipilih, pelanggan *logout* dengan meng-klik *link* *Log Off*.



Gambar 6.21. Halaman Pemesanan Barang dengan Jumlah Barang Sebanyak 1000 Buah.
Sumber : [Pengujian]



Gambar 6.22. Peringatan Ketika Jumlah Barang yang Dipesan Melebihi Stok.
Sumber : [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.21 diperlihatkan ketika pelanggan memasukkan jumlah barang sebanyak 1000 buah. Ketika pelanggan meng-klik *link* `Update`, terdapat peringatan yang diperlihatkan dalam Gambar 6.22. Peringatan tersebut menginformasikan bahwa barang yang akan dipesan melebihi jumlah stok yang tersedia pada aplikasi WarungLINUX. Namun pemesanan tetap dapat dilakukan dan akan diproses oleh pihak penjual.



D. Analisis Pengujian

Dari hasil pengujian dapat terlihat adanya pesan peringatan ketika pelanggan memesan barang melebihi jumlah stok. Namun pesanan yang dilakukan masih dapat diproses oleh aplikasi WarungLINUX.

E. Kesimpulan Pengujian

Dari pengujian dapat disimpulkan bahwa pesanan dengan jumlah barang melebihi stok masih dapat diproses oleh aplikasi WarungLINUX.

6.3 Pengujian Perangkat Keras SMS Gateway

Pengujian dilakukan dengan memeriksa apakah perangkat keras yang berupa ponsel seri Nokia 3100 dapat terhubung dengan komputer *server* menggunakan kabel data DKU-5. Selain itu juga memeriksa apakah ponsel dapat berfungsi sebagai *modem* data bagi komputer *server*.

Tujuan dari pengujian ini adalah untuk memastikan bahwa perangkat keras yang digunakan dapat berfungsi dengan baik.

6.3.1 Pengujian Koneksi Kabel Data

Pengujian kabel data adalah pengujian terhadap kabel data yang digunakan dalam tugas akhir. Kabel data yang digunakan adalah kabel data *clone* DKU-5 dengan *chipset* Prolific.

A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah kabel data yang digunakan dapat berfungsi dengan baik.

B. Prosedur Pengujian

Pengujian dilakukan pertama kali dengan menghubungkan kabel data DKU-5 pada *port* USB pada komputer *server*. Kemudian menjalankan perintah `dmesg | grep usb` melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Untuk lebih memperjelas pendeteksian kabel data oleh komputer *server*, juga dijalankan perintah `lsusb`.

```

an96ar@darkbox:~$ dmesg | grep usb
[17182041.820000] usb 1-2: new full speed USB device using ohci_hcd
and address 4
[17182042.024000] usb 1-2: configuration #1 chosen from 1 choice
[17182042.384000] usbcore: registered new driver usbserial
[17182042.384000] drivers/usb/serial/usb-serial.c: USB Serial support
registered for generic
[17182042.384000] usbcore: registered new driver usbserial_generic
[17182042.384000] drivers/usb/serial/usb-serial.c: USB Serial Driver
core
[17182042.400000] drivers/usb/serial/usb-serial.c: USB Serial support
registered for pl2303
[17182042.400000] usb 1-2: pl2303 converter now attached to ttyUSB0
[17182042.400000] usbcore: registered new driver pl2303
[17182042.400000] drivers/usb/serial/pl2303.c: Prolific PL2303 USB to
serial adaptor driver

```

Gambar 6.23. Hasil Perintah `dmesg | grep usb`.
Sumber: [Pengujian]

```

an96ar@darkbox:~$ lsusb
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 004: ID 067b:2303 Prolific Technology, Inc. PL2303
Serial Port
Bus 001 Device 003: ID 09da:000a A4 Tech Co., Ltd
Bus 001 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000

```

Gambar 6.24. Hasil Perintah `lsusb`.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.23 dan Gambar 6.24 diperlihatkan bahwa perangkat kabel data DKU-5 dapat dikenali dengan baik oleh komputer *server*. Kabel data tersebut terhubung dengan *port* `tty/USB0`. Hal ini terlihat pada baris `[17182042.400000] usb 1-2: pl2303 converter now attached to ttyUSB0`. Dalam gambar tersebut juga diperlihatkan *driver* Prolific yang digunakan oleh kernel yaitu PL2303. Hal tersebut terlihat pada baris `[17182042.400000] usbcore: registered new driver pl2303`.

D. Analisis Pengujian

Dari hasil pengujian kabel data DKU-5 menggunakan *chipset* Prolific dan terhubung pada *port* `tty/USB0`. Berdasarkan *chipset* yang terdeteksi, maka *driver* `pl2303` digunakan oleh sistem operasi.

E. Kesimpulan Pengujian

Dari pengujian yang telah dilakukan dapat disimpulkan bahwa kabel data DKU-5 dapat dikenali dengan baik oleh komputer *server*.

6.3.2 Pengujian Koneksi Ponsel Menggunakan Kabel Data

Pengujian ponsel adalah pengujian terhadap ponsel yang digunakan dalam tugas akhir yang berfungsi sebagai *modem* data. Ponsel yang digunakan dalam pengujian adalah Nokia 3100. Ponsel tersebut dihubungkan ke komputer *server* menggunakan kabel data DKU-5.

A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui apakah ponsel Nokia 3100 dapat berfungsi sebagai *modem* data sehingga dapat menerima dan mengirim SMS melalui SMS Gateway.

B. Prosedur Pengujian

Ponsel Nokia 3100 dihubungkan dengan kabel data DKU-5. Setelah terhubung, kabel data DKU-5 dihubungkan pada *port* USB milik komputer *server*. Setelah semua perangkat keras terhubung, dijalankan perintah `gnokii --identify` untuk melihat apakah ponsel Nokia 3100 dikenali oleh komputer *server*.

```
an96ar@darkbox:~$ gnokii --identify
GNOKII Version 0.6.14
LOG: debug mask is 0x1
phone instance config:
model: 6510
port_device: /dev/ttyUSB0
connection_type: 2
init_length: 0
serial_baudrate: 19200
.
.
Received model RH-19
Identifying...
.
.
Received model RH-19
IMEI       : 351517004327514
Manufacturer : Nokia
Found model "RH-19"
Model      : 3100
Product name : RH-19
Revision   : V 04.01
Serial device: closing device
```

Gambar 6.25. Hasil perintah `gnokii --identify`.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.25 diperlihatkan bahwa ponsel Nokia 3100 telah dikenali dengan baik oleh komputer *server* sebagai RH-19 yang kemudian dianggap sebagai model 3100. Dalam gambar tersebut juga terlihat bahwa ponsel Nokia 3100 terhubung dengan *port* USB yang ditunjukkan dengan baris `port_device: /dev/ttyUSB0`. Dalam gambar tersebut juga diperlihatkan nomor IMEI dari ponsel yang bersangkutan, yaitu 351517004327514.

D. Analisis Pengujian

Dari hasil pengujian terlihat bahwa ponsel terdeteksi sebagai RH-19. Ponsel ini terhubung melalui *port* USB. Dari hasil pengujian juga dapat diketahui nomor IMEI dari ponsel tersebut, yaitu 351517004327514.

E. Kesimpulan Pengujian

Dari pengujian yang dilakukan dapat disimpulkan bahwa ponsel Nokia 3100 dapat digunakan sebagai *modem* data dan berfungsi dengan baik.

6.4 Pengujian Pengiriman dan Penerimaan SMS Melalui SMS Gateway

Pengujian dilakukan dengan cara mengirim SMS dari ponsel menuju komputer *server*. Setelah itu pengujian juga dilakukan dengan cara mengirim SMS dari komputer *server* kepada ponsel.

A. Tujuan Pengujian

Tujuan dari pengujian ini untuk memastikan bahwa SMS dapat diterima dan dikirim oleh komputer *server* melalui SMS Gateway. Selain itu juga untuk mengetahui waktu yang dibutuhkan oleh SMS dari pengirim menuju penerima.

B. Prosedur Pengujian

Ponsel Nokia 8250 digunakan sebagai ponsel pengirim yang akan mengirimkan SMS ke komputer *server*. Ponsel tersebut mengirim SMS yang berisi pesan Test...Test. Kemudian dicek pada komputer *server*, waktu yang dibutuhkan untuk SMS tersebut diterima oleh SMS Gateway. Waktu yang dibutuhkan kemudian dicatat. Pesan kembali dikirim dari ponsel pengirim sebanyak 9 kali. Dan setiap waktu

penerimaan oleh komputer *server* dicatat. Dari keseluruhan hasil waktu penerimaan tersebut dihitung nilai rata-ratanya.

Kemudian komputer *server* mengirim SMS yang berisi pesan `Test...Test` kepada ponsel penerima yaitu Nokia 8250. Hal ini dilakukan sebanyak 10 kali. Waktu yang dibutuhkan ketika komputer *server* mengirim SMS hingga diterima oleh ponsel penerima dicatat. Kemudian dihitung nilai rata-ratanya.

Tabel 6.1. Hasil Pengujian Pengiriman SMS Dari Ponsel Pengirim ke Komputer *Server*.

Pengujian Ke-	Waktu yang Dibutuhkan (milidetik)
1	1334
2	1583
3	1444
4	1375
5	927
6	1327
7	1352
8	1613
9	959
10	1510

Sumber: [Pengujian]

Tabel 6.2. Hasil Pengujian Pengiriman SMS Dari Komputer *Server* ke Ponsel Penerima.

Pengujian Ke-	Waktu yang Dibutuhkan (milidetik)
1	1609
2	0993
3	0977
4	13810
5	1257
6	1148
7	1365
8	1578
9	1788
10	1058

Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Tabel 6.1 dapat dilihat bahwa waktu rata-rata yang diperlukan untuk sebuah ponsel pengirim mengirim SMS sebesar 1342,4 milidetik atau sekitar 13 detik. Dan sebuah ponsel menerima SMS dari komputer *server* sebesar 2558,3 milidetik atau sekitar 25 detik.

D. Analisis Pengujian

Dari hasil pengujian terlihat bahwa rata-rata waktu yang dibutuhkan ketika ponsel mengirim SMS ke komputer *server* sebesar sekitar 13 detik. Sedangkan ketika ponsel menerima SMS dari komputer *server* sebesar sekitar 25 detik.

E. Kesimpulan Pengujian

Dari pengujian yang telah dilakukan dapat disimpulkan bahwa komputer *server* dapat digunakan untuk menerima dan mengirim SMS melalui SMS *Gateway*.

6.5 Pengujian Basis Data SMS *Gateway*

Pengujian basis data SMS *Gateway* dilakukan dengan mengirimkan SMS menggunakan ponsel pengirim yaitu Nokia 8250. Pada komputer *server* digunakan program `smsd` untuk mengatur SMS yang masuk dan keluar untuk dimasukkan ke dalam basis data `sms`. Dalam basis data `sms` terdapat dua tabel, yaitu `inbox` dan `outbox`.

A. Tujuan Pengujian

Tujuan dari pengujian ini untuk memeriksa bahwa tabel `inbox` dapat digunakan sebagai penyimpan data bagi SMS yang masuk ke dalam komputer *server*. Begitu juga dengan tabel `outbox`, bertujuan untuk memeriksa apakah SMS yang akan keluar dari komputer *server* dapat disimpan dalam tabel `outbox`.

B. Prosedur Pengujian

Pada komputer *server* dijalankan program `smsd`. Program `smsd` ini akan membaca SMS yang masuk memasukkan data ke dalam tabel `inbox`. Data yang dimasukkan ke dalam `inbox` kemudian diperiksa dengan menggunakan *query* basis data. Selain itu juga dihitung waktu yang diperlukan oleh `smsd` untuk menerima SMS dari ponsel pengirim.

Pengujian pengiriman SMS dilakukan dengan cara dimasukkan data berupa SMS ke dalam tabel `outbox`. Kemudian dijalankan program `smsd` melalui Terminal. Nantinya program `smsd` akan mengirim SMS menggunakan data yang ada dalam tabel `outbox` ke ponsel pengirim. Kemudian diperiksa tabel `outbox` menggunakan *query* basis data. Selain itu juga dihitung waktu yang diperlukan oleh program `smsd` untuk mengirimkan SMS.

```
mysql> select * from inbox where id=1;
Empty set (0.00 sec)

mysql> select * from inbox where id=1;
+----+-----+-----+-----+-----+-----+-----+
| id | number | smsdate | insertdate | text | phone | processed |
+----+-----+-----+-----+-----+-----+-----+
| 1 | +628133350084 | 2007-06-25 12:42:08 | 2007-06-29 21:08:31 | Test...Test | NULL | 0 |
+----+-----+-----+-----+-----+-----+-----+
1 row in set (0.07 sec)
```

Gambar 6.26. Tabel `inbox` Sebelum dan Sesudah Menerima SMS.
Sumber: [Pengujian]

```
mysql> select * from outbox where id=1;
+----+-----+-----+-----+-----+-----+-----+
| id | number | smsdate | insertdate | text | phone | processed |
+----+-----+-----+-----+-----+-----+-----+
| 1 | +628133350084 | 2007-06-25 12:42:08 | 2007-06-29 21:08:31 | Test...Test | NULL | 0 |
+----+-----+-----+-----+-----+-----+-----+
1 row in set (0.06 sec)

mysql> select * from outbox where id=1;
Empty set (0.00 sec)
```

Gambar 6.27. Tabel `outbox` Sebelum dan Sesudah Mengirim SMS.
Sumber: [Pengujian]

Tabel 6.3. Hasil Pengujian Waktu Penerimaan dan Pengiriman SMS.

Nomor	Jenis Pengujian	Waktu yang Dibutuhkan (milidetik)
1	Proses Penerimaan SMS	1737
2	Proses Pengiriman SMS	1638

Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.26 dan Gambar 6.27 diperlihatkan SMS yang diterima dan dikirim melalui tabel `inbox` dan `outbox`. Dalam tabel `inbox` diperlihatkan ketika SMS belum diterima oleh komputer *server* dan SMS sudah diterima menggunakan program `smsd`. Setelah SMS diterima, pada kolom-kolom pada tabel `inbox` telah berisi data-data, yaitu kolom `id` bernilai 1, kolom `number` bernilai +628133350084, kolom `insert_date` bernilai 2007-06-29 21:08:31, kolom `text` bernilai Test...Test, kolom `phone` bernilai NULL, dan kolom `processed` bernilai 0.

Sedangkan pada tabel `outbox` diperlihatkan ketika program `smsd` belum menerima masukan untuk mengirim SMS dan ketika program `smsd` telah menerima masukan SMS untuk dikirim. Setelah menerima SMS masukan, kolom-kolom pada tabel `outbox` telah berisi data-data, yaitu kolom `id` bernilai 1, kolom `number` bernilai +6281333550084, kolom `insert_date` bernilai 2007-06-29 21:08:31, kolom `text` bernilai `Test...Test`, kolom `phone` bernilai `NULL`, dan kolom `processed` bernilai 0.

Dalam tabel 6.3 diperlihatkan waktu yang dibutuhkan oleh program `smsd` untuk menerima SMS dan mengirim SMS. SMS yang diterima membutuhkan waktu sebanyak 1737 milidetik atau sekitar 17 detik dari ponsel pengirim hingga diterima oleh program `smsd`. Sedangkan waktu yang dibutuhkan oleh program `smsd` untuk mengirim SMS hingga diterima oleh ponsel penerima adalah 1638 milidetik atau sekitar 16 detik.

D. Analisis Pengujian

Dari hasil pengujian diperlihatkan perubahan pada tabel `inbox` dan `outbox` dari basis data `sms`. Dari hasil pengujian juga didapatkan waktu bagi program `smsd` untuk menerima SMS sebesar sekitar 17 detik. Sedangkan waktu yang dibutuhkan bagi program `smsd` untuk mengirim SMS sebesar sekitar 16 detik.

E. Kesimpulan Pengujian

Dari pengujian diatas dapat disimpulkan bahwa basis data `sms` dapat berfungsi dengan baik untuk menerima dan mengirim SMS dengan menggunakan program `smsd`.

6.6 Pengujian Transaksi WarungLINUX Melalui Media SMS

Pengujian dilakukan dengan memeriksa hasil transaksi jika dilakukan dengan melalui media SMS. Hasil yang ada kemudian dibandingkan dengan transaksi dengan melalui media Internet.

A. Tujuan Pengujian

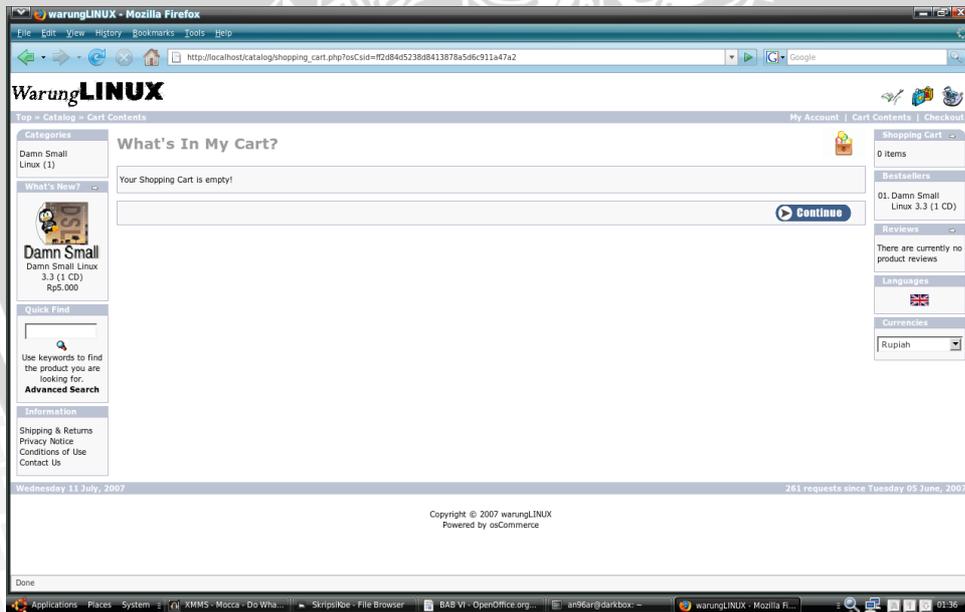
Tujuan dari pengujian ini adalah untuk melihat jika terdapat perbedaan bila melakukan transaksi melalui media SMS dibandingkan dengan melalui media Internet.

B. Prosedur Pengujian

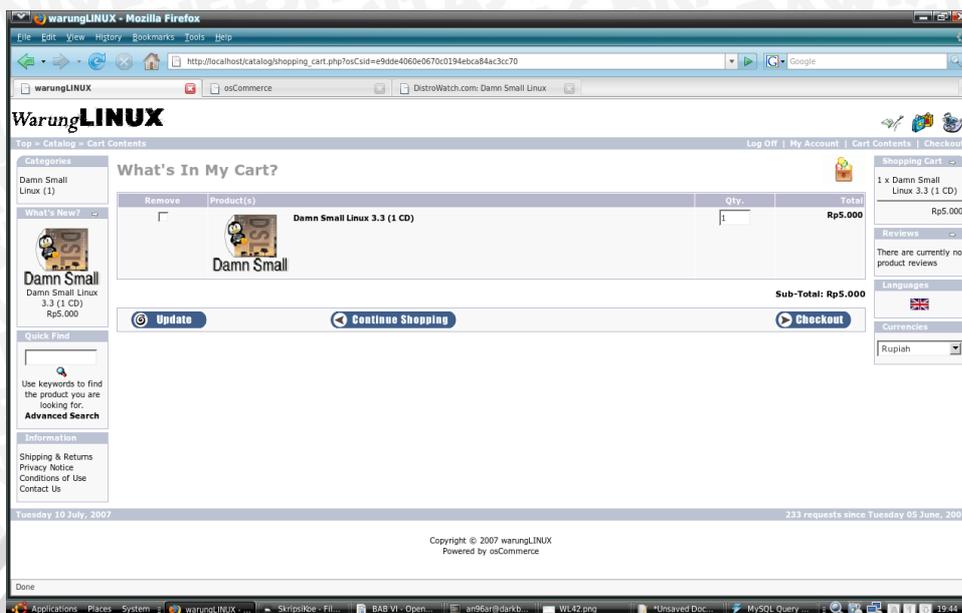
Pada ponsel pengirim dikirim SMS sesuai dengan format yang telah ditentukan pada Bab Perancangan. SMS itu nantinya akan diterima oleh komputer *server* yang ditangani oleh program *smsd*. Program *smsd* akan menyimpan SMS yang diterima ke dalam basis data *sms*. Program antarmuka akan membaca SMS yang tersimpan dalam basis data *sms* dan mengolahnya agar dapat dimasukkan ke dalam basis data *warunglinux*.

Sebelum perubahan dilakukan, administrator menjalankan program *mysql* melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Setiap perubahan yang dilakukan dalam aplikasi WarungLinux dilihat pada program *mysql* yang menjalankan perintah *select* pada tabel *inbox* pada basis data *sms* dan tabel *customers_basket* dalam basis data *warunglinux*.

Setelah proses tersebut selesai, dijalankan *web browser* untuk melihat transaksi yang telah dilakukan.



Gambar 6.28. Halaman Cart Contents Sebelum SMS Diterima.
Sumber: [Pengujian]



Gambar 6.29. Halaman Cart Contents Setelah SMS Diterima.
Sumber: [Pengujian]

```
mysql> select * from inbox where id=1;
Empty set (0.00 sec)

mysql> select id,number,smsdate,text,processed from inbox where id=1;
+----+-----+-----+-----+-----+
| id | number | smsdate | text | processed |
+----+-----+-----+-----+-----+
| 1 | +6281333550084 | 2007-06-25 12:42:08 | #anggar@mail.com #apaaja *51 =1 | 0 |
+----+-----+-----+-----+-----+
1 row in set (0.07 sec)
```

Gambar 6.30. Tabel inbox Sebelum dan Sesudah Menerima SMS.
Sumber: [Pengujian]

```
mysql> select * from customers_basket;
Empty set (0.00 sec)

mysql> select customers_basket_id, customers_id, products_id, customers_basket_quantity, customers_basket_date_added
from customers_basket;
+-----+-----+-----+-----+-----+
| customers_basket_id | customers_id | products_id | customers_basket_quantity | customers_basket_date_added |
+-----+-----+-----+-----+-----+
| 6 | 3 | 51 | 1 | 20070711 |
+-----+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

Gambar 6.31. Tabel customers_basket Sebelum dan Sesudah SMS Diterima.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.28 diperlihatkan ketika Cart Contents masih kosong. Setelah SMS diterima dan diolah oleh aplikasi antarmuka, maka Cart Contents telah terisi oleh barang yaitu Damn Small Linux 3.3 (1 CD) yang dipesan oleh pelanggan seperti yang terlihat dalam gambar 6.29.



Dalam Gambar 6.30 terlihat tabel `inbox` ketika belum dan telah menerima SMS yang berisi transaksi barang. Ketika telah menerima SMS, tabel `inbox` berisi data-data, yaitu kolom `id` bernilai 1, kolom `number` bernilai +6281333550084, kolom `text` bernilai #anggar@mail.com #apaaja *51 =1, dan kolom `processed` bernilai 0. Dalam Gambar 6.31 juga terlihat tabel `customers_basket` yang masih kosong. Setelah SMS dari pelanggan diterima dan aplikasi antarmuka mengolahnya, maka terlihat bahwa tabel `customers_basket` telah terisi. Data-data tersebut adalah kolom `customers_basket_id` bernilai 1, kolom `customers_id` bernilai 3, kolom `products_id` bernilai 51, kolom `customers_basket_quantity` bernilai 1, dan kolom `customers_basket_date_address` bernilai 20070711.

D. Analisis Pengujian

Dari hasil pengujian yang didapatkan, terlihat bahwa ketika SMS belum diterima, tabel `inbox` tidak berisi data apapun. Ketika SMS telah diterima, maka tabel `inbox` berisi data SMS. Ketika tabel `inbox` terisi oleh data, maka tabel `customers_basket` juga terisi data yang berasal dari tabel `inbox` milik basis data `sms`. Data ini diolah oleh program antarmuka dan dipindahkan ke tabel `customers_basket` milik basis data `warunglinux`.

E. Kesimpulan Pengujian

Dari pengujian diatas dapat disimpulkan bahwa transaksi menggunakan SMS dapat dilakukan dan tidak ada bedanya dengan melakukan transaksi melalui media Internet.

6.7 Pengujian Konfirmasi Hasil Transaksi WarungLINUX Melalui Media SMS

Pengujian dilakukan dengan memeriksa hasil dari konfirmasi yang dikirim oleh SMS Gateway ketika transaksi berhasil dilakukan pada ponsel pengirim. Hasil pengujian berupa waktu yang diperlukan ketika SMS Gateway mengirim pesan hingga diterima oleh ponsel pengirim.

A. Tujuan Pengujian

Pengujian dilakukan bertujuan untuk mengetahui apakah konfirmasi transaksi melalui media SMS dapat berjalan dengan baik.

B. Prosedur Pengujian

Pengujian dilakukan dengan cara mengirimkan SMS yang berisi transaksi pemesanan barang sesuai dengan format yang telah ditentukan pada Bab Perancangan. SMS itu nantinya akan diolah oleh aplikasi antarmuka. Jika transaksi berhasil diproses oleh aplikasi antarmuka, maka akan dikirim SMS konfirmasi hasil transaksi oleh aplikasi antarmuka. SMS konfirmasi ini dimasukkan ke dalam tabel `outbox` pada basis data `sms`.

Kemudian program `smsd` akan mengirim SMS sesuai dengan data yang ada pada tabel `outbox`.

Sebelum perubahan dilakukan, administrator menjalankan program `mysql` melalui program Terminal yang dijalankan melalui Applications | Accessories | Terminal. Setiap perubahan yang dilakukan dalam aplikasi WarungLINUX dilihat pada program `mysql` yang menjalankan perintah `select` pada tabel `outbox` pada basis data `sms`.

```
mysql> select id,number,smsdate,text,processed from outbox where id=1;
+-----+-----+-----+-----+-----+
| id | number      | smsdate      | text                                     | processed |
+-----+-----+-----+-----+-----+
| 1  | +628133350084 | 2007-06-25 12:42:08 | Pesanan Anda telah kami proses. Terima Kasih | 0        |
+-----+-----+-----+-----+-----+
1 row in set (0.07 sec)

mysql> select * from outbox where id=1;
Empty set (0.00 sec)
```

Gambar 6.32. Tabel `outbox` Sebelum dan Sesudah Mengirim SMS Konfirmasi.
Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.32 terlihat ketika tabel `outbox` telah berisi data berupa SMS konfirmasi yang dimasukkan oleh program antarmuka. Data tersebut berupa kolom `id` bernilai 1, kolom `number` bernilai +628133350084, kolom `smsdate` bernilai 2007-06-25 12:42:08, kolom `text` bernilai Pesanan Anda telah kami proses. Terima Kasih, dan kolom `processed` bernilai 0. Kemudian juga terlihat tabel `outbox` yang telah kosong karena SMS telah dikirim oleh program `smsd` sesuai dengan nomor yang dituju.

D. Analisis Pengujian

Dari hasil pengujian diketahui bahwa sebelum SMS dikirim, data masih ada pada tabel `outbox`. Ketika SMS telah dikirim, data pada tabel `outbox` sudah tidak ada.

E. Kesimpulan Pengujian

Berdasarkan pengujian diatas dapat disimpulkan bahwa SMS konfirmasi berhasil dikirimkan oleh aplikasi antarmuka setelah transaksi berhasil diproses.

6.8 Pengujian Kata Kunci pada Media SMS

Pengujian dilakukan dengan memasukkan kata kunci (*password*) yang terdapat pada media SMS yang sesuai dengan *password* milik pelanggan pada aplikasi WarungLINUX kemudian melihat perbedaan yang terjadi.

A. Tujuan Pengujian

Tujuan pengujian untuk mengetahui apakah terdapat perbedaan pada *password* pada media SMS dengan *password* pada aplikasi WarungLINUX.

B. Prosedur Pengujian

Pengujian dilakukan dengan membandingkan *password* pada media SMS yang sesuai dengan *password* pada aplikasi WarungLINUX. Proses perbandingan dilakukan dengan cara menjalankan program yang telah dibuat sebelumnya oleh penulis. Program ini akan mengeluarkan hasil enkripsi berdasarkan masukan yang telah ditentukan. Hasil keluaran yang terlihat kemudian dibandingkan dengan isi dari field `password_customers` yang terdapat pada tabel `customers` pada basis data `warunglinux`.

```
an96ar@darkbox:~$ perl enkripsi.pl
189b2f87f4ae1abc6bd1be1ee0314e21
```

Gambar 6.33. Program Enkripsi Password Pada Media SMS.

Sumber: [Pengujian]

```
mysql> select customers_firstname, customers_password from customers;
+-----+-----+
| customers_firstname | customers_password |
+-----+-----+
| Anggar             | 189b2f87f4ae1abc6bd1be1ee0314e21:04 |
+-----+-----+
1 row in set (0.04 sec)
```

Gambar 6.34. Isi Tabel `customers` pada Basis Data `warunglinux`.

Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Gambar 6.33 diperlihatkan hasil keluaran dari *password* yang telah dienkripsi. *Password* yang dienkripsi adalah `apaaaja` dan *salt* yang digunakan adalah `04`. Hasil yang didapatkan dari proses enkripsi adalah `189b2f87f4ae1abc6bd1be1ee0314e21`. Hasil keluaran tersebut sesuai dengan *password* yang terdapat pada aplikasi WarungLINUX yang diperlihatkan dalam Gambar 6.34. Dalam Gambar 6.34, diperlihatkan *field* `customers_password` yang berisi *password* yang telah dienkripsi beserta *salt*.

D. Analisis Pengujian

Dari hasil pengujian diketahui bahwa *password* yang telah dienkripsi menggunakan program adalah sama dengan *password* yang terdapat pada aplikasi WarungLINUX.

E. Kesimpulan Pengujian

Dari pengujian yang telah dilakukan dapat disimpulkan bahwa tidak ada perbedaan antara *password* pada media SMS dengan *password* pada aplikasi WarungLINUX.

6.9 Pengujian Sistem Keseluruhan

Pengujian dilakukan dengan cara memasukkan pesanan melalui media SMS kemudian melihat hasil transaksi pada aplikasi WarungLINUX.

A. Tujuan Pengujian

Pengujian bertujuan untuk mengetahui bahwa transaksi melalui media SMS dapat dilakukan dan hasilnya dapat dilihat pada aplikasi WarungLINUX.

B. Prosedur Pengujian

Pengujian dilakukan dengan mengirimkan SMS transaksi yang sesuai dengan format yang telah ditentukan pada Bab Perancangan menggunakan ponsel Nokia 8250. Kemudian menjalankan program `smsd` yang akan memasukkan SMS ke dalam basis data `sms`. Setelah itu program antarmuka dijalankan untuk mengolah SMS yang terdapat pada basis data `sms`. Setelah SMS berhasil diolah, aplikasi antarmuka akan memasukkan SMS konfirmasi ke dalam tabel `outbox` pada basis data `sms`. Program `smsd` kembali dijalankan

untuk mengirimkan SMS yang ada pada tabel `outbox`. Waktu yang diperlukan dari mulai mengirimkan SMS transaksi hingga diterimanya SMS konfirmasi dihitung.

Proses transaksi diulang sebanyak 4 kali. Kemudian dihitung waktu rata-rata terhadap waktu yang diperlukan ketika SMS pertama kali dikirimkan hingga SMS konfirmasi diterima.

Tabel 6.4. Tabel Hasil Pengujian Waktu Transaksi.

Pengujian Ke-	Waktu yang Diperlukan (milidetik)
1	2589
2	2364
3	2745
4	2139
5	2484

Sumber: [Pengujian]

C. Hasil Pengujian

Dalam Tabel 6.4 diperlihatkan waktu-waktu yang diperlukan transaksi menggunakan SMS hingga selesai. Berdasarkan Tabel 6.4 diperoleh waktu rata-rata sebesar 2464,2 milidetik atau sekitar 24 detik.

D. Analisis Pengujian

Berdasarkan hasil pengujian, didapatkan waktu rata-rata yang diperlukan dari transaksi dimulai hingga selesai sebesar sekitar 24 detik.

E. Kesimpulan Pengujian

Dari pengujian diatas dapat disimpulkan bahwa transaksi menggunakan media SMS dapat dilakukan.

6.10 Analisis Pengembangan Sistem Transaksi

Analisis pengembangan sistem secara keseluruhan dilakukan setelah aplikasi selesai diimplementasikan. Analisis pengembangan sistem meliputi kelebihan dan kekurangan dari sistem yang telah dikembangkan.

6.10.1 Kelebihan Pengembangan Sistem Transaksi

Kelebihan dari pengembangan sistem yang telah dibuat berdasarkan hasil pengujian adalah:

1. Memiliki jangkauan yang luas dalam menerima transaksi (sama dengan luasnya jaringan *provider* yang digunakan).
2. Bentuk masukan transaksi yang ringkas, sehingga dapat menampung variasi barang dengan cukup banyak.
3. Masukan transaksi menggunakan *username* dan *password* sehingga orang yang tidak mempunyai akses tidak dapat melakukan transaksi.
4. Pelanggan menerima konfirmasi ketika transaksi berhasil dilakukan.

6.10.2 Kekurangan Pengembangan Sistem Transaksi

Kekurangan pengembangan sistem yang didapatkan dari hasil pengujian berupa:

1. Hanya dapat menggunakan satu *provider* sebagai penerima SMS untuk SMS Gateway.
2. Sistem sangat tergantung pada kualitas jaringan *provider* yang digunakan.
3. SMS Gateway Gnokii yang digunakan hanya dapat mendukung beberapa tipe ponsel saja.
4. SMS transaksi terbatas pada jumlah karakter SMS.
5. Jika transaksi tidak berhasil dilakukan, aplikasi antarmuka tidak mengirim pesan kesalahan kepada pelanggan.
6. Diperlukan biaya tambahan untuk mengirimkan SMS konfirmasi kepada pelanggan.

BAB VII

PENUTUP

Dalam bab ini akan dijelaskan mengenai kesimpulan yang didapatkan dalam pembuatan tugas akhir. Selain itu juga terdapat saran yang berguna untuk perbaikan bagi tugas akhir.

7.1 Kesimpulan

Berdasarkan pada pembuatan tugas akhir yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Aplikasi WarungLINUX merupakan aplikasi toko *online* yang berjalan secara *client-server*. Aplikasi WarungLINUX disimpan pada komputer *server* dan dapat dijalankan pada komputer *client* menggunakan *web browser*.
2. Administrasi WarungLINUX hanya dapat dilakukan oleh seseorang yang memiliki hak akses administrator. Administrasi pada WarungLINUX meliputi antara lain data konfigurasi, data kategori/barang, data modul, dan data pelanggan. Proses administrasi kategori/barang meliputi tambah, ubah, dan atau hapus data.
3. Pengembangan transaksi menggunakan media SMS dapat diterapkan pada aplikasi WarungLINUX dengan menggunakan ponsel sebagai *modem data*.
4. Penggunaan SMS dapat digunakan sebagai masukan transaksi untuk aplikasi WarungLINUX. Sementara aplikasi antarmuka berguna untuk mengubah masukan yang berupa SMS menjadi masukan untuk basis data yang digunakan oleh aplikasi WarungLINUX.
5. Waktu yang dibutuhkan untuk sebuah transaksi dikirim melalui SMS dan menerima konfirmasi transaksi adalah sebesar 2464,2 milidetik atau sekitar 24 detik.

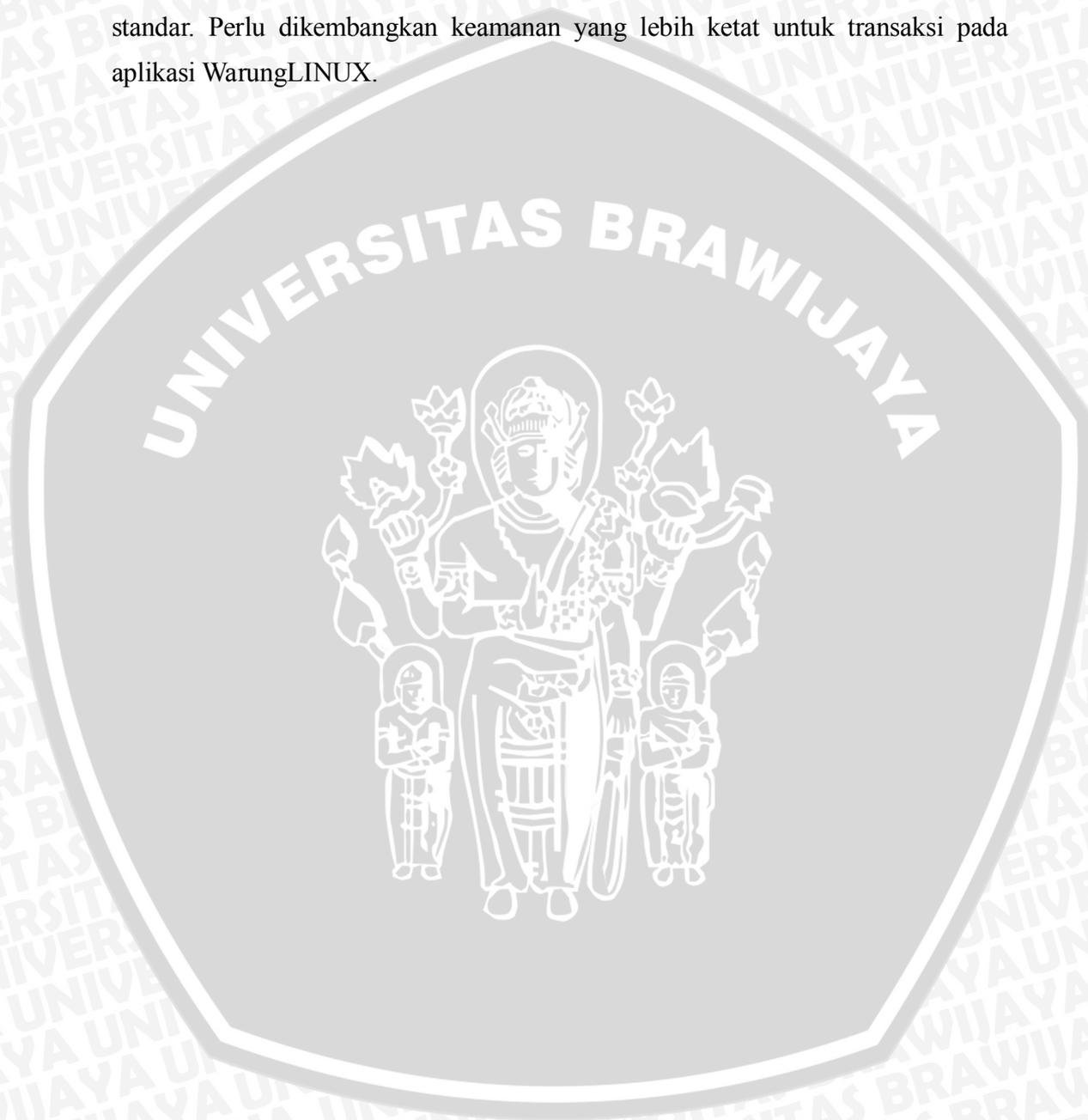
7.2 Saran

Saran-saran yang bisa diterapkan untuk pengembangan tugas akhir diantaranya adalah :

1. Penggunaan ponsel sebagai *modem data* sangat terbatas dalam menangani jumlah SMS yang masuk sekaligus. Untuk itu perlu digunakan sebuah *modem GSM* untuk menangani SMS yang masuk.



2. Penggunaan satu kartu GSM sangat terbatas dalam menangani SMS yang masuk. Untuk itu perlu dikembangkan aplikasi yang dapat menangani lebih dari satu nomor GSM.
3. Keamanan yang ada pada aplikasi WarungLINUX menggunakan konfigurasi standar. Perlu dikembangkan keamanan yang lebih ketat untuk transaksi pada aplikasi WarungLINUX.



DAFTAR PUSTAKA

- [ALT-92] Alter, Steven: "Information Systems: A Management Perspective", The Benjamin/Cummings Publishing Company, Inc, 1992.
- [ANO-05] Anonimus: "What is a SMS Gateway?",
<http://www.logixmobile.com/faq/show.asp?catid=1&faqid=9>, 2005.
- [ANO-06] Anonimus: "osCommerce: About osCommerce",
<http://www.oscommerce.com/about/about>, 2006.
- [ANO-07] Anonimus: "Cable USB Nokia DKU5 ca-42 + 3 CD 6101 3220 6103",
http://www.deremate.com.ar/categorias/listados/celulares-telefonias/440/44021_0_1.html, 2007.
- [ANI-06] Anonimus: "Short message service",
http://en.wikipedia.org/wiki/Short_message_service, 2006
- [ANI-07] Anonimus: "Data Flow Diagram",
http://en.wikipedia.org/wiki/Data_flow_diagram, 2007.
- [ANN-06] Anonimus: "OsCommerce – Wikipedia, the free encyclopedia",
<http://en.wikipedia.org/wiki/OsCommerce>, 2006.
- [ANN-07] Anonimus: "Analisis Dengan Diagram Aliran Data (DFD)",
http://ilkom.unsri.ac.id/dosen/hartini/materi/VIII_DFD.pdf, 2007.
- [ANM-06] Anonimus: "tep_database-pr2.2-CVS",
http://www.oscommerce.com/solutions/downloads/tep_database-pr2.2-CVS.pdf, 2006.
- [ANM-07] Anonimus: "DistribusiDvdReposUbuntu – Wiki Ubuntu Indonesia",
<http://wiki.ubuntu-id.org/DistribusiDvdReposUbuntu>, 2007.
- [ANU-07] Anonimus: "Panduan Tidak Resmi Bagi Pemula Ubuntu 6.10 (Edgy Eft)",
<http://ubuntuguide.org/>, 2007.
- [ANS-07] Anonimus: "Oscommerce, ZenCart, X-Cart, CRE Loaded, OSC Max Comparison of Shopping Carts",
http://www.itwebexperts.com/oscommerce_zencart_xcart_comparison.php, 2007.
- [AON-07] Anonimus: "Shopping Carts Overview", <http://info.template-help.com/print/pre->

- sale_questions/CMS_and_sitebuilders_questions/article7.html, 2007.
- [AOI-07] Anonimus: “RootSudo – Community Ubuntu Documentation”,
<https://wiki.ubuntu.com/RootSudo>, 2007
- [AOM-07] Anonimus: “Ihwal Ubuntu | Komunitas Ubuntu Indonesia”, <http://www.ubuntu-id.org/ubuntu/ihwal>, 2007.
- [AOU-07] Anonimus: “Free Software/Tools and Libraries for Sending and Receiving SMS Messages with a Computer/PC”,
<http://www.developershome.com/sms/freeLibForSMS.asp>, 2007.
- [BLA-01] Blank-Edelman, David N.: “Perl for System Administration”, O’Reilly & Associates, Inc, 2001.
- [BRO-01] Brown, Martin C: “Perl: The Complete Reference Second Edition”, MacGraw-Hill/Osborne, 2001.
- [COZ-00] Cozens, Simon: “Beginning Perl”, Wrox Press Ltd., 2000.
- [FAB-05] Fabella, Ryan: “Memonitor Koneksi Server dengan SMS Gateway”, InfoLINUX, 06/2005, 2005, hal. 58 – 60.
- [FAT-02] Fathansyah: “Basis Data”, Penerbit CV. Informatika, 2002.
- [FIN-04] Finf, Andreas., Rodrigues, Bruno., Tolj, Stipe., Syvänen, Aarno., Malysh, Alexander., Wirzenius, Lars., Marjola, Kalle.: “Kannel 1.4.0 User’s Guide: Open Source WAP and SMS gateway”, Kannel Group, 2004.
- [HAL-01] Hall, James: “Accounting Information Systems. 3rd Edition”, South Western College Publishing, 2001.
- [HAR-04] Haryanto, Steven: “Regex: Kumpulan Resep Pemrograman”, Penerbit Dian Rakyat, 2004.
- [IRE-02] Indrajit, Richardus Eko., Prastowo, Bambang N., Syukri, Muhammad.: “Buku Pintar Linux: Aplikasi Web Database dengan PHP & MySQL”, Penerbit PT. Elex Media Komputindo, 2002.
- [KAD-02] Kadir, Abdul: “Dasar Pemrograman PERL”, Penerbit Andi, 2002.
- [KAD-03] Kadir, Abdul: “Pengenalan Sistem Informasi”, Penerbit Andi, 2003.
- [KER-04] Kerner, Sean Michael: “Getting More Than You Paid For: Working with osCommerce's Open Source Storefront”, <http://www.ecommerce->

guide.com/article.php/3428461, 2004.

[KOE-04] Koegler, Scott: “Getting More Than You Paid For: osCommerce's Open Source Storefront”,

http://ecommerce.internet.com/how/build/article/0,,10362_3399041,00.html, 2004.

[NOP-07] Noprianto: “Distro Untuk Anda”, InfoLINUX, PT InfoLINUX Media Utama, 2007.

[NUG-97] Nugraha K, Dany: “Sisi Keamanan: Cara Membuat Penjahat Bekerja Lebih Keras”, <http://www.infokomputer.com/arsip/internet/1197/cakra/cakrawa1.shtml>, 1997

[WAL-05] Wall, Larry: “Perl Programmer Reference Guide”, Manual Perl, 2005.

[YUN-06] Yuniarto: “Membangun Aplikasi SMS Gateway di Linux”, Penerbit Dian Rakyat, 2006.



LAMPIRAN A: Instalasi dan Konfigurasi Apache

Instalasi :

Untuk instalasi Apache, masukkan perintah-perintah berikut pada *prompt* terminal:

```
$ sudo apt-get install apache2  
$ sudo apt-get install libapache2-mod-auth-mysql
```

Konfigurasi :

Agar *user* yang sedang aktif merupakan administrator halaman-halaman PHP, maka perlu dilakukan konfigurasi. Untuk melakukannya, ubah *file* konfigurasi Apache :

```
$ gksudo "gedit /etc/apache2/apache2.conf"
```

Cari kedua *string* yang dimulai oleh “User” dan “Group”, dan ganti nama-nama tersebut dengan *username* yang sedang aktif dan *groupname* yang sedang digunakan. Kemudian Apache perlu di-*restart*.

Opsi-opsi konfigurasi yang berhubungan secara spesifik dengan website user (diakses melalui `localhost/~username`) terdapat dalam `/etc/apache2/mods-enabled/userdir.conf`.



LAMPIRAN B: Instalasi dan Konfigurasi MySQL

Instalasi :

Untuk instalasi MySQL, jalankan perintah berikut dari sebuah *prompt* terminal:

```
$ sudo apt-get install mysql-server mysql-client
```

Ketika instalasi selesai, *server* MySQL seharusnya mulai secara otomatis. Jalankan perintah berikut dari *prompt* terminal untuk mengecek apakah *server* MySQL telah berjalan:

```
$ sudo netstat -tap | grep mysql
```

Ketika perintah dijalankan, seharusnya terlihat baris berikut atau sesuatu yang serupa:

```
tcp        0      0 localhost.localdomain:mysql    :::* LISTEN -
```

Jika *server* tidak berjalan dengan benar, ketik perintah berikut untuk memulainya:

```
$ sudo /etc/init.d/mysql restart
```

Konfigurasi :

Secara *default*, *password* administrator tidak diberlakukan. Setelah MySQL ter-*install*, hal pertama yang perlu dilakukan adalah meng-konfigurasi *password* administrator MySQL. Untuk melakukan hal ini, jalankan perintah berikut:

```
$ sudo mysqladmin -u root password rahasia
```

```
$ sudo mysqladmin -p -u root -h localhost password rahasia
```

LAMPIRAN C: Instalasi PHP

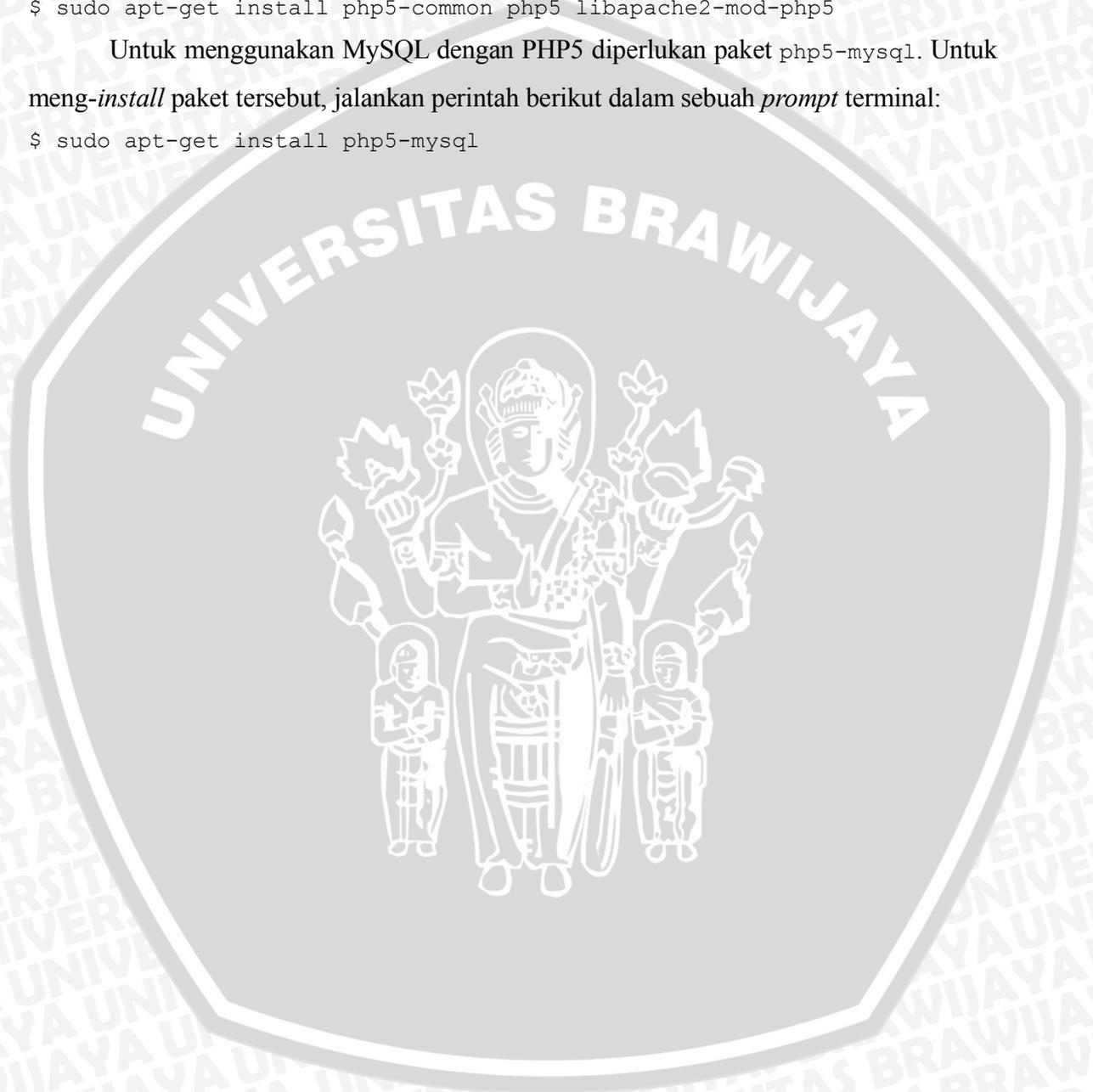
Instalasi :

Untuk meng-*install* PHP5, masukkan perintah berikut dalam sebuah *prompt* terminal:

```
$ sudo apt-get install php5-common php5 libapache2-mod-php5
```

Untuk menggunakan MySQL dengan PHP5 diperlukan paket `php5-mysql`. Untuk meng-*install* paket tersebut, jalankan perintah berikut dalam sebuah *prompt* terminal:

```
$ sudo apt-get install php5-mysql
```



LAMPIRAN D: Instalasi osCommerce

Instalasi :

Upload direktori katalog ke dalam *server*. Semua *file* PHP harus di-*upload* dalam mode ASCII, bukan mode binari.

Struktur dari osCommerce adalah:

catalog
catalog/admin
extras

Upload folder *catalog* yang menyertakan folder *admin* ke dalam dokumen *root* (*public_html*, *www*, *htdocs*).

Buat sebuah basis data dan *user*, dan tugaskan *user* ke dalam basis data tersebut. Tulis nama basis data, *username*, dan *password* basis data tersebut untuk prosedur instalasi basis data.

Buka *web browser* dan arahkan ke <http://yourserver/catalog/install/> dan klik pada tombol *Install* untuk memulai prosedur instalasi osCommerce. Ikuti prosedur instalasi yang ada.

