

**RANCANG BANGUN SMS GATEWAY BERBASIS WEB
DENGAN SISTEM OPERASI LINUX UBUNTU**

TUGAS AKHIR

Oleh:

IQBAL IRWANSYAH

0405230205



**PROGRAM EKSTENSI DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

2008

**RANCANG BANGUN SMS GATEWAY BERBASIS WEB
DENGAN SISTEM OPERASI LINUX UBUNTU**

TUGAS AKHIR

Oleh:

IQBAL IRWANSYAH

0405230205



**TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**PROGRAM EKSTENSI DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

2008

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul:

RANCANG BANGUN SMS GATEWAY BERBASIS WEB DENGAN SISTEM OPERASI LINUX UBUNTU

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Pendidikan Sarjana Ekstensi Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, May 2008

Iqbal Irwansyah
NIM. 0405230205

PENGESAHAN

Tugas Akhir dengan judul :

RANCANG BANGUN SMS GATEWAY BERBASIS WEB DENGAN SISTEM OPERASI LINUX UBUNTU

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Pendidikan Sarjana Ekstensi Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Tugas Akhir ini telah diujikan pada sidang ujian tugas akhir pada tanggal 11 Juni 2008 dan dinyatakan memenuhi syarat/sah sebagai tugas akhir pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 11 Juni 2008

Dosen Pembimbing,

Ir. Arifin Djauhari MT

NIP. 130 891 107

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

Ir. Arifin Djauhari MT

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga tugas akhir ini dapat selesai dengan baik.

RANCANG BANGUN SMS GATEWAY BERBASIS WEB DENGAN SISTEM OPERASI LINUX UBUNTU

ABSTRAK

SMS Gateway adalah suatu sistem antar muka antara web/internet dengan sistem seluler dengan menggunakan salah satu fitur yang paling banyak digunakan yaitu SMS. Dengan *SMS Gateway* ini, pengguna dapat mengirimkan atau menerima SMS dengan antar muka web. Pengguna dapat mengirim SMS ke banyak tujuan (*SMS Broadcasting*) atau menerima SMS dari pengguna dan dapat menjawabnya secara otomatis (*Auto Replying SMS*). Dengan antar muka web, maka penggunaannya akan menjadi mudah, siapapun dapat menggunakannya tanpa kesulitan.

SMS Gateway Server adalah sebuah perangkat yang akan menjadi *server* antar muka yang menghubungkan web dengan operator seluler. Aplikasi ini bersifat lintas operator, jadi pengguna (*user*) dapat menggunakan operator seluler apapun. *SMS Gateway Server* dapat berupa sebuah komputer pribadi biasa (PC) atau dapat pula sebuah komputer khusus kelas *server*. *SMS Gateway Server* akan dibuat dengan sistem operasi Ubuntu Linux yang akan dilengkapi dengan Apache Web Server untuk antar muka web. *SMS Gateway Server* juga akan dilengkapi dengan GSM modem untuk menghubungkannya dengan operator seluler. Untuk koneksi dengan internet dapat dilakukan dengan menyewa ISP (*Internet Service Provider*) yang dilengkapi dengan sebuah IP *dedicated*. Dengan IP *dedicated*, maka server dapat diakses oleh *web user* dan mengakses jaringan internet.

Pada Tugas Akhir ini penulis akan membuat perangkat lunak pengaksesan SMS melalui web dan fungsi *SMS gateway* untuk keperluan berbagai layanan SMS seperti pengiriman *single* SMS, pengiriman SMS *broadcast*, dan fitur balas SMS otomatis (*auto-reply*) dengan bahasa pemrograman PHP.

Kata Kunci: SMS, Web, Linux, Ubuntu

Iqbal Irwansyah
NPM 0405230205
Departemen Teknik Elektro

Dosen Pembimbing
Ir. Arifin Djauhari MT

BUILD-UP WEB BASED SMS GATEWAY WITH UBUNTU LINUX OPERATING SYSTEM

ABSTRACT

SMS Gateway is a system interface between web/internet with cellular system using one of most used future which is SMS. With this SMS Gateway, user able to sent or receive SMS with web interface. User able to sent SMS to many recipient (Broadcasting SMS) or receive SMS from user and answer it automatically (Auto Replying SMS). With web interface, making simple for user, so anyone able to use it without any difficulties.

SMS Gateway Server is a server that provide interface and connecting web with the cellular operator. This application work with any operator, so user able to use their own operator to receive/sent SMS. SMS Gateway Server can be just an ordinary personal computer (PC) or can be a specially made for server class. This SMS Gateway Server will be made with Ubuntu Linux as the operating system and equipped with Apache Web Server for web interface. SMS Gateway Server will also equipped with GSM modem to connect it with cellular operator. For internet connection can be implmented by providing an ISP (Internet Service Provider) with a dedicated IP. With a dedicated IP, so the server able to access by web user and accessing the internet.

This final project will concentrate the writing to create a software for accessing SMS via web and SMS gateway function for every SMS service such as sending single SMS, sending broadcast SMS and SMS auto reply feature by using PHP programming language.

Kata Kunci: SMS, Web, Linux, Ubuntu

DAFTAR ISI

Pernyataan Keaslian Tugas Akhir	i
Pengesahan	ii
Ucapan terima kasih	iii
Abstrak	iv
Daftar Isi	vi
Daftar Gambar	ix
Daftar Singkatan	xii
 BAB 1 PENDAHULUAN	 1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENELITIAN	2
1.3 BATASAN MASALAH	2
1.4 SISTEMATIKA PENULISAN	2
 BAB 2 KONSEP DASAR SMS, WEB SERVICE DAN LINUX	 3
2.1 KONSEP DAN PENGENALAN TELEKOMUNIKASI SELULER	3
2.1.1 Konsep Dasar Seluler	4
2.1.2 Arsitektur GSM	4
2.1.3 Mobile Station	5
2.1.4 Base Tranceiver Station	6
2.1.5 Base Station Controller	6
2.1.6 Mobile Switching Center Dan Visitor Location Register	6
2.1.7 Home Location Register	6
2.2 KONSEP DAN PENGENALAN SHORT MESSAGE SERVICE	7
2.2.1 Deskripsi Layanan	7
2.2.2 Arsitektur Jaringan GSM Dengan SMS Center	7
2.2.3 Short Message Entities	8
2.2.4 Service Center	9
2.2.5 Mode Pengalamatan (<i>Addressing Modes</i>)	9
2.2.6 Protocol Layer	10
2.2.7 Transport Protocol Data Unit	12
2.3 LINUX DAN <i>OPEN SOURCE</i>	13

2.3.1 Ubuntu Linux	15
2.3.2 Alasan Memilih Ubuntu Linux	15
2.4 <i>WEB SERVER</i>	16
2.4.1 Prinsip Kerja <i>Web Server</i>	17
2.4.2 Prinsip Kerja Apache	18
2.4.3 TCP/IP	18
2.4.4 Apache Dan TCP/IP	20
2.4.5 Koneksi Dari Sisi Klien	21
2.4.5 Koneksi Dari Sisi <i>Server</i>	23
BAB 3 PERANCANGAN SMS GATEWAY	24
3.1 PERANCANGAN SISTEM	24
3.2 INSTALASI UBUNTU LINUX	25
3.2.1 Kebutuhan Sistem	25
3.2.2 Proses Instalasi Ubuntu	26
3.2.3 Proses Update Ubuntu	32
3.3 INSTALASI APACHE WEB SERVER, PHP, MYSQL SERVER DAN KANNEL	34
3.4 PERANGKAT LUNAK SISTEM	36
3.4.1 Konsep Kerja PHP	36
3.4.2 Bahasa Pemrograman PHP	37
3.4.2.1 <i>Variabel pada PHP</i>	39
3.4.2.2 <i>Mengakses database MySQL pada PHP</i>	42
3.4.2.3 <i>Flowchart pemrograman</i>	45
BAB 4 UJICoba SISTEM DAN ANALISA	46
4.1 PENGUJIAN APACHE <i>WEB SERVER</i>	46
4.2 PENGUJIAN LOGIN <i>SMS GATEWAY</i>	47
4.3 PENGUJIAN MENGISI <i>PHONEBOOK SMS GATEWAY</i>	49
4.4 PENGUJIAN MENGIRIM <i>SINGLE SMS</i>	53
4.5 PENGUJIAN MENGIRIM <i>BROADCAST SMS</i>	54
4.6 PENGUJIAN <i>AUTO-REPLY SMS</i>	56
4.6.1 Pembuatan Skema Auto Reply	56
4.6.2 Pengujian Auto-Reply SMS Berdasarkan Skema Yang Telah Dibuat ...	60
BAB 5 KESIMPULAN	62
DAFTAR ACUAN	63



DAFTAR GAMBAR

1. Gambar 2.1 Konsep Seluler	4
2. Gambar 2.2 Arsitektur jaringan GSM	5
3. Gambar 2.3 Contoh kartu SIM	5
4. Gambar 2.4 Arsitektur jaringan GSM yang dilengkapi dengan layanan SMS	8
5. Gambar 2.5 Setting SMS pada <i>handset</i> perangkat bergerak	8
6. Gambar 2.6 Format Sub-addressing	9
7. Gambar 2.7 Sub-addressing pada SMSC yang sama	10
8. Gambar 2.8 SMS <i>Protocol Stack</i>	11
9. Gambar 2.9 Transfer pesan diantara dua SME pada operator yang sama	12
10. Gambar 2.10 Transfer pesan diantara dua operator yang berbeda (lintas operator) ...	12
11. Gambar 2.11 Format TPDU	12
12. Gambar 3.1 Perancangan sistem kerja SMS Gateway	24
13. Gambar 3.2 Tampilan awal proses instalasi Ubuntu Linux	26
14. Gambar 3.3 Tampilan standar Ubuntu Linux	27
15. Gambar 3.4 Pemilihan bahasa yang ingin digunakan	27
16. Gambar 3.5 Pemilihan lokasi geografis	28
17. Gambar 3.6 Pemilihan jenis <i>keyboard</i> yang digunakan	29
18. Gambar 3.7 Pemilihan lokasi instalasi pada <i>hardisk</i>	29
19. Gambar 3.8 Mengisi data pribadi untuk administrator	30
20. Gambar 3.9 Verifikasi akhir instalasi Ubuntu	31
21. Gambar 3.10 Proses instalasi telah selesai	31
22. Gambar 3.11 Tampilan awal <i>Update Manager</i>	32
23. Gambar 3.12 Proses download <i>update</i> Ubuntu	33
24. Gambar 3.13 Aplikasi yang memiliki <i>update</i> dari Ubuntu	33
25. Gambar 3.14 Synaptic Package Manager	34
26. Gambar 3.15 Memilih paket Apache <i>Web Server</i>	35
27. Gambar 3.16 Konfirmasi instalasi Apache <i>Web Server</i>	35
28. Gambar 3.17 Proses <i>download</i> paket aplikasi	36
29. Gambar 3.18 Skema HTML	37
30. Gambar 3.19 Skema PHP	37
31. Gambar 3.20 Hasil kode HTML pada <i>browser</i>	38

32. Gambar 3.21 Hasil kode PHP pada <i>browser</i>	39
33. Gambar 3.22 Hasil skrip latihan-variabel.php tampak pada <i>browser</i>	40
34. Gambar 3.23 Hasil skrip latihan-input.php tampak pada <i>browser</i>	41
35. Gambar 3.24 Mengisi input pada latihan-input.php	42
36. Gambar 3.25 Hasil skrip salam.php tampak pada <i>browser</i>	42
37. Gambar 3.26 <i>Flowchart</i> pemrograman perangkat lunak	45
38. Gambar 4.1 Uji coba instalasi Apache	46
39. Gambar 4.2 Uji coba instalasi PHP	47
40. Gambar 4.3 Halaman login <i>SMS Gateway</i>	47
41. Gambar 4.4 Halaman administrator <i>SMS Gateway</i>	48
42. Gambar 4.5 Halaman <i>user SMS Gateway</i>	49
43. Gambar 4.6 Halaman <i>phonebook</i>	50
44. Gambar 4.7 Halaman membuat <i>group</i> pada <i>phonebook</i>	50
45. Gambar 4.8 Halaman notifikasi bahwa <i>group</i> baru sudah tercipta	51
46. Gambar 4.9 Halaman <i>phonebook</i> menunjukkan <i>group</i> yang baru tercipta	51
47. Gambar 4.10 Halaman menambah nomor pada <i>group phonebook</i>	52
48. Gambar 4.11 Halaman notifikasi bahwa nomor sudah dimasukan pada <i>group</i>	52
49. Gambar 4.12 Halaman <i>phonebook</i> bahwa nomor sudah dimasukkan pada <i>group</i>	53
50. Gambar 4.13 Mengirim SMS dengan <i>SMS Gateway</i>	53
51. Gambar 4.14 Hasil kiriman SMS pada kota keluar (<i>Outgoing SMS</i>) <i>SMS gateway</i> ..	54
52. Gambar 4.15 Hasil kiriman SMS pada penerima SMS	54
53. Gambar 4.16 Halaman <i>SMS broadcast</i>	55
54. Gambar 4.17 Hasil kiriman <i>SMS broadcast</i> pada <i>SMS Gateway</i>	55
55. Gambar 4.18 Hasil kiriman <i>SMS broadcast</i> pada penerima SMS	56
56. Gambar 4.19 Tampilan halaman pada <i>SMS auto reply</i>	57
57. Gambar 4.20 Menambahkan nama skema <i>SMS auto reply</i>	57
58. Gambar 4.21 Notifikasi nama skema <i>SMS auto reply</i> telah berhasil dibuat	57
59. Gambar 4.22 Nama skema <i>SMS auto reply</i> telah berhasil dibuat	58
60. Gambar 4.23 Halaman skenario <i>SMS auto reply</i>	58
61. Gambar 4.24 Halaman menambahkan skenario <i>SMS auto reply</i>	59
62. Gambar 4.25 Notifikasi skenario <i>SMS auto reply</i> telah berhasil dibuat	59
63. Gambar 4.26 Skenario <i>SMS auto reply</i> berhasil dibuat	60
64. Gambar 4.27 SMS yang dikirim untuk menguji <i>SMS auto reply</i>	60
65. Gambar 4.28 Kotak masuk (Inbox) yang diterima <i>SMS Gateway</i>	61

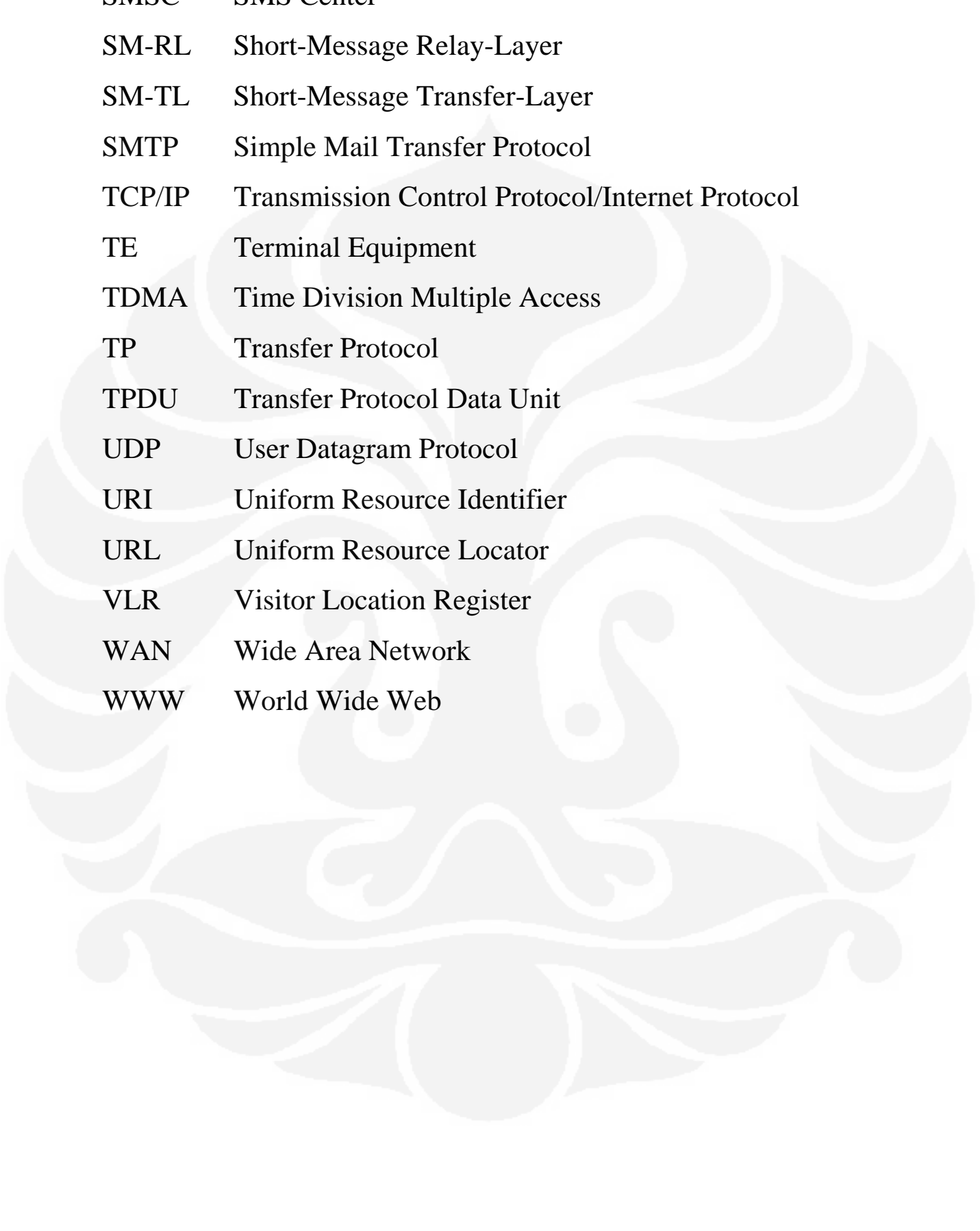
66. Gambar 4.29 Kotak masuk (Inbox) yang diterima oleh pengirim	61
---	----



DAFTAR SINGKATAN

2G	Second Generation
3G	Third Generations
3GPP	3rd Generation Partnership Project
BIOS	Basic Input Output System
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CEPT	Conférence Européenne des Postes et Télécommunications (European Conference on Post and Telecommunication)
CERN	Conseil Européenne pour la Recherche Nucleaire
CSPDN	Circuit Switched Public Data Network
DCS	Digital Cellular System
DNS	Domain Name Service
ETSI	European Telecommunications Standard Institute
FOSS	Free and Open Source Software
FSF	Free Software Foundation
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMEI	International Mobile Equipment Identity
IMSI	International Subscriber Identity
ISDN	Integrated Services Digital Network

ISP	Internet Service Provider
LAN	Local Area Network
ME	Mobile Equipment
MAP	Mobile Application Part
MS	Mobile Station
MSC	Mobile Switching Center
MSISDN	Mobile Station ISDN Number
NCSA	National Center for Supercomputing Applications
NNTP	Network News Transfer Protocol
NSS	Network Subsystem
OSS	Operation Subsystem
PC	Personal Computer
PDA	Personal Digital Assistance
PHP	PHP Hypertext Processor
PIM	Personal Information Manager
PLB	Perangkat Lunak Bebas
PSPDN	Packet Switched Public Data Network
PSTN	Public Switched Telephone Network
RDBMS	Relational Database Management System
SS7	Signaling System Number 7
SC	Service Center
SM-AL	Short-Message Application-Layer
SME	Short Message Entities
SM-LL	Short-Message Link-Layer
SMS	Short Message Service



SMSC	SMS Center
SM-RL	Short-Message Relay-Layer
SM-TL	Short-Message Transfer-Layer
SMTP	Simple Mail Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TE	Terminal Equipment
TDMA	Time Division Multiple Access
TP	Transfer Protocol
TPDU	Transfer Protocol Data Unit
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLR	Visitor Location Register
WAN	Wide Area Network
WWW	World Wide Web

BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG

Perkembangan layanan telepon seluler (*wireless communication*) semakin hari semakin meningkat. Layanan yang dimiliki oleh telepon seluler juga semakin hari semakin bervariasi. Mulai dari layanan *text messaging* konvensional, layanan *browsing* internet kecepatan tinggi sampai layanan *video call*.

Semakin hari teknologi seluler semakin berkembang. Teknologi seluler dapat dimanfaatkan untuk berbagai keperluan sehari-hari. Salah satu teknologi yang paling mudah digunakan adalah *text messaging* atau yang biasa disebut SMS (*Short Message Service*). Kini, SMS dapat dimanfaatkan untuk berbagai aplikasi seperti aplikasi *Mobile Banking*, *Pooling* interaktif, *Premium Service*, dan lain-lain.

SMS sangat mudah penggunaannya, sehingga siapa saja dapat memanfaatkannya. Mulai dari anak-anak, orang dewasa sampai manula. Karena mudahnya penggunaan ini maka sangat tepat jika pemanfaatan teknologi ini untuk berbagai keperluan.

Dengan adanya berbagai layanan yang dapat diberikan melalui SMS, maka kini banyak muncul penyedia layanan konten yang berbasis SMS. Berbagai layanan ini dapat dimanfaatkan oleh pengguna dengan mendaftarkan (registrasi) nomor seluler yang dimilikinya ke penyedia layanan konten melalui SMS. Biasanya dengan mengetikkan suatu kata kunci untuk registrasi kemudian kata kunci registrasi tersebut dikirimkan ke nomor tertentu maka pengguna dapat langsung menikmati layanan SMS yang diregistrasikannya. Meski terkadang layanan SMS tersebut memakai sistem pentarifan yang berbeda dengan tarif SMS biasa.

Disisi lain, penggunaan internet semakin hari semakin berkembang. Semakin hari semakin banyak orang yang terhubung dengan internet. Semakin banyak juga orang yang bergantung dengan internet. Teknologi internet memungkinkan orang untuk berkomunikasi ke berbagai penjuru dunia dengan mudah. Disamping itu, teknologi internet juga dimanfaatkan untuk menyediakan informasi untuk pengguna internet. Kita dapat mencari berbagai informasi yang kita inginkan di internet. Berbagai layanan internet dapat kita jumpai saat ini. Mulai dari Email, VoIP, Blog, jaringan pertemanan, dan lain sebagainya. Kita dapat memanfaatkan berbagai teknologi internet untuk keperluan kita. Dengan adanya teknologi seluler, internet dapat diakses kapan saja, dimana saja dan oleh siapa saja.

Disamping teknologi SMS dan internet, teknologi lain yang kini sangat berkembang adalah Linux. Linux adalah system operasi yang sangat maju dan berkembang setiap hari. Linux adalah system operasi yang terbuka yang biasa disebut dengan FOSS (*Free and Open Source Software*). Setiap orang dapat menggunakan linux tanpa harus membayar biaya tertentu untuk penggunaannya. Ini yang menjadikan perbedaan dengan system operasi lain dimana system operasi lain membutuhkan biaya tertentu (atau yang biasa disebut dengan biaya aktivasi) untuk menggunakan system operasinya secara legal. Dengan system operasi yang *Open Source*, biaya penggunaan system operasi kini tak menjadi masalah yang berarti. Sistem *Open Source* juga sangat mudah untuk didapat dengan mudah melalui *download* dan dapat di distribusikan secara bebas. Dengan tidak adanya biaya ini tentu sangat baik untuk menurunkan biaya penyediaan layanan ketika digunakan untuk kepentingan komersial.

Berdasarkan latar belakang tersebut, maka penulis akan membuat sebuah aplikasi perangkat lunak untuk mengakses SMS melalui web untuk mengirim, menerima SMS, mengirim SMS *broadcast* dan mengirim *auto reply* SMS dengan menggunakan bahasa pemrograman PHP pada PC dengan sistem operasi Linux Ubuntu.

1.2. TUJUAN PENELITIAN

Tujuan penelitian ini adalah sebagai berikut:

Merancang bangun SMS gateway berbasis web dengan sistem operasi Linux Ubuntu.

1.3. BATASAN MASALAH

Mengingat konteks *SMS Gateway* berbasis web ini sangat kompleks, maka penulis membatasi masalah hanya pada pembuatan perangkat lunak pengaksesan SMS melalui web dan fungsi SMS gateway untuk keperluan berbagai layanan SMS seperti pengiriman *single* SMS, pengiriman SMS *broadcast*, dan fitur balas SMS otomatis (*auto-reply*) dengan bahasa pemrograman PHP.

1.4. SISTEMATIKA PENULISAN

Sistematika penulisan tugas akhir ini adalah sebagai berikut: Bab pertama menjelaskan tentang latar belakang, tujuan penelitian, pembatasan masalah, dan sistematika penulisan. Bab kedua membahas landasan teori yang memuat tentang prinsip dasar SMS, prinsip dasar web service dan prinsip-prinsip dasar sistem operasi Linux Ubuntu. Bab ketiga berisi perancangan sistem *SMS Gateway* berbasis web. Bab keempat berisi pengujian dan analisa mengenai aplikasi *SMS Gateway* berbasis web dengan sistem operasi Linux. Bab kelima berisi kesimpulan yang didapat dari penulisan Tugas Akhir ini.

BAB 2

KONSEP DASAR SMS, WEB SERVICE DAN LINUX

2.1.KONSEP DAN PENGENALAN TELEKOMUNIKASI SELULER

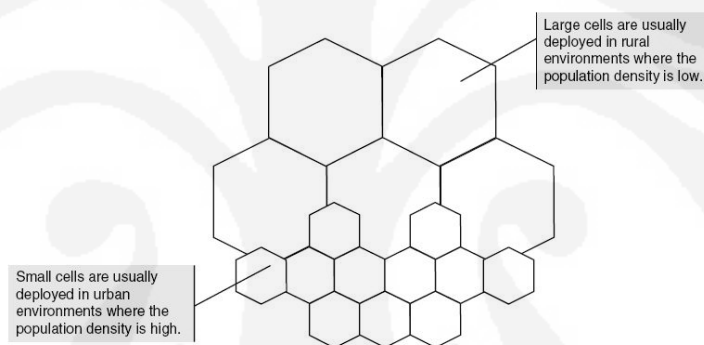
Sebelum dikenalnya sistem GSM (*Global System for Mobile Communication*), jaringan seluler yang dipakai oleh berbagai negara biasanya berbeda satu dengan yang lainnya sehingga satu dengan yang lainnya tidak cocok untuk saling berkomunikasi karena perbedaan sistem. Ketidakcocokan ini menyebabkan tidak praktisnya penggunaan teknologi seluler bagi pengguna yang berpindah antar sistem yang berbeda. Untuk menyiasati ketidakcocokan ini, pada tahun 1982, *Conférence Européenne des Postes et Télécommunications* (CEPT) menciptakan komite *Groupe Spécial Mobile* yang dikemudian hari berubah menjadi *Global System for Mobile Communication* (GSM). Tugas utama dari komite tersebut adalah untuk menstandarkan sistem komunikasi seluler benua Eropa yang ketika itu menggunakan frekuensi radio 900 MHz. Pada tahun 1989, *European Telecommunications Standard Institute* (ETSI) mengambil alih tanggung jawab untuk mengembangkan evolusi teknologi dan spesifikasi GSM. Pada tahun 2000, tanggung jawab ini ditransfer ke *Third Generation Partnership Project* (3GPP)[1]. Inisiatif ini sangat berhasil sehingga jaringan seluler memenuhi standar spesifikasi GSM yang saat ini telah berkembang di seluruh dunia. Variasi standar spesifikasi GSM ini juga distandarkan pada frekuensi radio 1800 MHz dan 1900 MHz yang dikenal dengan DCS 1800 dan DCS 1900. Pada bulan Maret 2004, asosiasi GSM[2] melaporkan total jumlah pelanggan GSM sebanyak 1046,8 juta orang yang tersebar di 207 negara[3].

Jaringan GSM dikarakteristikan oleh komunikasi suara digital dan mendukung layanan data kecepatan rendah. Antar muka GSM didasarkan pada *Time Division Multiple Access* (TDMA). Dengan TDMA, sebuah *band* radio digunakan bersama oleh beberapa pengguna dengan mengalokasikan satu atau lebih *timeslot* pada radio *carrier* yang diberikan kepada setiap pengguna. Dengan GSM, transfer data dapat dihasilkan diatas koneksi *circuit-switched*. Untuk komunikasi data seperti ini, bit rate yang didapat mencapai 14,4 Kbps pada koneksi slot tunggal. Koneksi *slot* tunggal ini biasa disebut dengan *Circuit Switched Data* (CSD). *Bit rate* yang lebih tinggi yang mencapai 57,6 Kbps dapat dicapai dengan mengalokasikan lebih dari satu buah *slot* pada koneksi data. Koneksi *multi slot* ini disebut dengan *High Speed CSD* (HSCSD).

Salah satu yang paling populer pada layanan GSM adalah *Short Message Service* (SMS). Layanan ini memungkinkan pelanggan SMS untuk bertukar pesan teks singkat. Pembahasan lebih dalam pada layanan ini akan dibahas lebih lanjut.

2.1.1. Konsep Dasar Selular

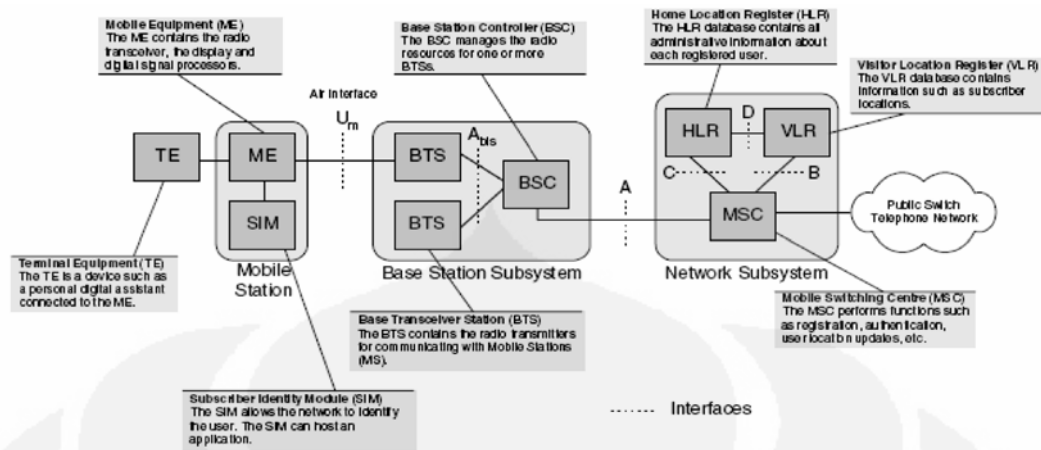
Band radio tersedia untuk komunikasi tanpa kabel pada jaringan bergerak mengingat radio frekuensi adalah sumber daya yang terbatas. Untuk dapat menggunakan sumber daya tersebut secara efisien, jaringan GSM dibuat berdasarkan konsep seluler. Dengan konsep ini, sumber daya radio yang sama (dikarakteristikan oleh *band* frekuensi dan *timeslot*) dapat dimanfaatkan secara simultan oleh pengguna tanpa *interface* jika mereka dibatasi oleh jarak yang minimum. Jarak minimum diantara dua pengguna tergantung pada jalan yang ditempuh oleh gelombang propagasi radio pada lingkungan dimana kedua pengguna tersebut berada. Pada jaringan GSM, semakin kecil *cell*, maka akan semakin besar faktor frekuensi *reuse*, seperti yang ditunjukkan oleh Gambar 2.1.



Gambar 2.1 Konsep Seluler [4]

2.1.2. Arsitektur GSM

Jaringan GSM disusun atas tiga buah subsystem yaitu: *Base station Subsystem* (BSS), *Network Subsystem* (NSS), dan *Operation Subsystem* (OSS). OSS mengimplementasikan fungsi sebagai administrator jaringan. Untuk lebih jelas dalam menggambarkan elemen dari GSM arsitektur, biasanya OSS tidak dimasukkan dalam penggambaran jaringan seperti ditunjukkan dalam Gambar 2.2. Elemen dari BSS dan NSS akan dijelaskan lebih lanjut pada bagian selanjutnya.



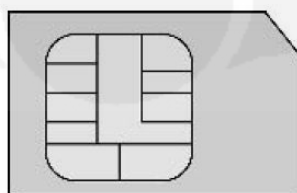
Gambar 2.2 Arsitektur jaringan GSM [5]

2.1.3. Mobile Station

Mobile Station (MS) adalah alat yang mengirim dan menerima sinyal radio dalam *cell* yang ada pada sebuah *site*. Sebuah *mobile station* dapat berupa *handset* telepon bergerak biasa sampai sekompleks seperti perangkat *Personal Digital Assistance* (PDA). *Handset* perangkat bergerak memiliki kemampuan seperti komunikasi suara, fitur pesan singkat, dan manajemen penyimpanan daftar telepon. Tambahan dari kemampuan dasar ini, sebuah perangkat PDA biasanya dikemas bersama dengan kemampuan berselancar mikro (*microbrowser*) di internet dan kemampuan untuk mengatur kontak, kalender dan penjadwalan pribadi melalui *Personal Information Manager* (PIM). Ketika pengguna bergerak, kontrol jaringan untuk koneksi MS berpindah-pindah dari sel satu ke sel lainnya untuk mendukung mobilitas MS. Proses ini disebut dengan *handover*.

Mobile Station (MS) terdiri dari *Mobile Equipment* (ME) dan *Subscriber Identity Module* (SIM). *International Mobile Equipment Identity* (IMEI) yang unik disimpan dalam ME untuk mengidentifikasi setiap perangkat yang terhubung ke jaringan bergerak.

Kartu SIM biasanya disediakan oleh penyelenggara jaringan (operator) dimana pengguna yang terdaftar dapat menggunakan jaringan tersebut dalam bentuk kartu cerdas (*Smart Card*). Sebuah *microchip* biasanya dikeluarkan dari kartu cerdas tersebut dan dimasukkan langsung ke dalam *slot* khusus di dalam perangkat bergerak (*handset*). Contoh kartu SIM ditunjukkan pada Gambar 2.3.



Gambar 2.3 Contoh kartu SIM [6]

MS saat ini dapat dihubungkan dengan perangkat eksternal seperti PDA atau *Personal Computer* (PC). Perangkat eksternal tersebut dinamakan dengan *Terminal Equipment* (TE) pada arsitektur jaringan GSM pada Gambar 2.2.

Sebuah *Short Message Service* (SMS) biasanya disimpan dalam MS. Kebanyakan *handset* memiliki kemampuan menyimpan di dalam SIM. Produk kelas tinggi terkadang menggabungkan kemampuan penyimpanan pada perangkat bergerak seperti *flash memory* pada *handset*.

2.1.4. Base Tranceiver Station

Base Tanceiver Station (BTS) berperan dalam menjembatani sebagai antarmuka komunikasi udara diantara semua MS yang aktif yang berada dibawah cakupan areanya (*cell site*). Ini termasuk modulasi/demodulasi sinyal, *signal equalizing* dan *error coding*. Beberapa BTS terhubung dengan satu buah *Base Station Controller* (BSC).

2.1.5. Base Station Controller

Base Station Controller (BSC) berfungsi untuk mengatur koneksi dari BTS yang berada dibawah kontrolnya. Fungsinya juga memungkinkan untuk mengoperasikan seperti *handover*, konfigurasi *cell site*, mengatur sumber daya radio, dan pengaturan level power radio frekuensi dari BTS. Fungsi lain BSC adalah sebagai penghubung koneksi terkonsentrasi jaringan ke arah MSC.

2.1.6. Mobile Switching Center Dan Visitor Location Register

Mobile Switching Center (MSC) berfungsi sebagai pengatur switching komunikasi dari sistem dan bertanggung jawab untuk *call set-up*, *release* dan *routing*. Ini juga menyediakan fungsi sebagai perangkat pencatat biaya dan antar muka dengan jaringan lainnya.

Visitor Location Register (VLR) berisi informasi dinamis tentang pengguna yang terhubung dengan jaringan dan lokasi dari pengguna tersebut. VLR biasanya terintegrasi dengan MSC.

Melalui MSC, jaringan bergerak berkomunikasi dengan jaringan lain seperti *Public Switched Telephone Network* (PSTN), *Integrated Services Digital Network* (ISDN), *Circuit Switched Public Data Network* (CSPDN), and *Packet Switched Public Data Network* (PSPDN).

2.1.7. Home Location Register

Home Location Register (HLR) adalah sebuah element yang berisi detail data yang menjadi pelanggan pada penyelenggara jaringan. HLR biasanya mampu untuk mengatur informasi ratusan ribu bahkan jutaan pelanggan.

Pada jaringan GSM, *signaling* didasarkan pada protokol *Signaling System Number 7* (SS7). Kegunaan SS7 bersama dengan protokol *Mobile Application Part* (MAP) untuk *signaling* protokol yang spesifik. Pada waktu tertentu MAP digunakan untuk pertukaran lokasi dan pertukaran informasi pelanggan antara HLR dan element jaringan lainnya seperti MSC. Untuk setiap pelanggan, HLR menjaga pemetaan antara *International Subscriber Identity* (IMSI) dan *Mobile Station ISDN Number* (MSISDN).

Untuk alasan keamanan, IMSI terkadang ditransmisikan melalui interface udara dan hanya dapat di diketahui oleh jaringan GSM dimana pelanggan terdaftar. IMSI dibentuk berdasarkan rekomendasi format ITU-E.212. Tidak seperti IMSI, MSISDN mengidentifikasikan pelanggan diluar jaringan GSM. MSISDN dikonstruksikan menurut standar format ITU-E.164.

2.2.KONSEP DAN PENGENALAN SHORT MESSAGE SERVICE

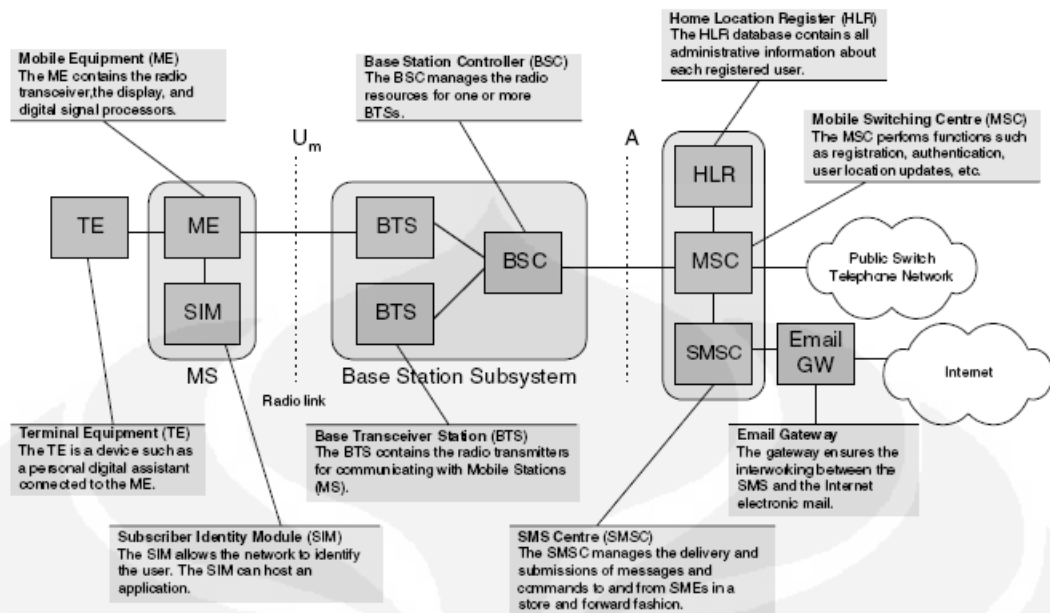
Short Message Service (SMS) adalah layanan dasar yang memungkinkan pengguna untuk bertukar pesan teks singkat ke pelanggan lainnya. Pesan teks pertama dipercaya ditransferkan pada tahun 1992 melalui kanal signaling pada jaringan GSM Eropa. Sejak keberhasilan percobaan pertama, penggunaan SMS menunjukkan pertumbuhan yang sangat pesat. Mobile Data Assosiation²⁾ melaporkan bahwa jumlah total pesan teks orang ke orang yang dikirim dari empat buah jaringan GSM di Inggris pada tahun 2003 berjumlah 20,5 milyar.

2.2.1. Deskripsi Layanan

Dikembangkan sebagai bagian dari GSM Phase 1 spesifikasi teknis ETSI, SMS memungkinkan MS dan perangkat yang terhubung dengan jaringan bergerak lainnya untuk bertukar pesan teks singkat. Tugas mula dari pengembangan standar SMS berawal dari ETSI dan kini diteruskan oleh aktifitas 3GPP. Sejak pengenalan awal dari jaringan GSM, SMS ditempatkan dalam teknologi jaringan lainnya seperti GPRS dan CDMA.

2.2.2. Arsitektur Jaringan GSM Dengan SMS Center

Realisasi dari SMS dapat diimplementasikan bersama dengan beberapa elemen lainnya pada arsitektur jaringan. Gambar 2.4 menggambarkan arsitektur dari jaringan GSM yang memiliki layanan SMS.



Gambar 2.4 Arsitektur GSM yang dilengkapi dengan layanan SMS [7]

Dua tambahan elemen jaringan pada arsitektur diatas adalah *SMS Center* (SMSC) dan *Email Gateway*. Sebagai tambahan, sebuah elemen yang disebut dengan entitas pesan singkat, biasanya dalam bentuk aplikasi perangkat lunak pada perangkat bergerak, dibutuhkan untuk menangani pesan-pesan tersebut (seperti mengirim, menerima, menyimpan, menghapus dan lain sebagainya). *Short Message Entities* (SME) tidak ditunjukkan pada gambar tersebut.

2.2.3. Short Message Entities

Elemen yang dapat mengirim dan menerima SMS biasa disebut dengan *Short Message Entities* (SME). Sebuah SME dapat berupa aplikasi perangkat lunak pada *handset* tetapi dapat juga dijumpai pada perangkat faximile, telex, dan lain lain. *Handset* harus dikonfigurasi secara benar agar dapat bekerja secara baik pada jaringan bergerak. Sebuah *handset* biasanya dibuat pra-konfigurasi oleh pabrik pembuat handset tersebut, tetapi dapat dibuat konfigurasi manual. Konfigurasi SMS ditunjukkan oleh Gambar 2.5.



Gambar 2.5 Setting SMS pada *handset* perangkat bergerak [8]

2.2.4. Service Center

Service Center (SC) atau *SMS Center* (SMSC) memainkan peranan penting dalam arsitektur SMS. Fungsi utama dari SMSC menyiarkan pesan singkat diantara SME dan menyimpan dan meneruskan dari pesan singkat (menyimpan SMS tersebut jika penerima SMS sedang tidak terhubung dengan jaringan).

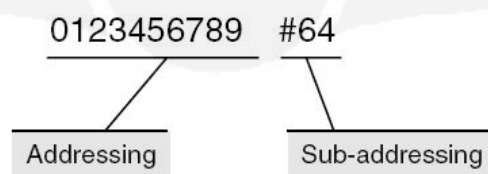
Penyelenggara jaringan bergerak biasanya memiliki kerjasama komersial saling menguntungkan untuk memungkinkan saling pertukaran pesan diantara penyelenggara jaringan. Ini artinya pesan yang dikirim dari sebuah SME yang terhubung pada jaringan A dapat dikirimkan ke SME lain yang terhubung pada jaringan B. Kemampuan untuk bertukar pesan ini dapat dilakukan meski mereka tidak terdaftar pada jaringan yang sama dan kadang kala pelanggan berada pada lokasi yang berbeda negara adalah salah satu fitur kunci yang membuat SMS begitu sukses.

Saat ini SMSC komersial biasanya dapat memproses lebih dari 1000 SMS perdetik untuk satu SMSC. Penyelenggara jaringan yang besar sering kali mempunyai beberapa SMSC untuk memenuhi kebutuhan pelanggannya.

2.2.5. Mode Pengalamatan (*Addressing modes*)

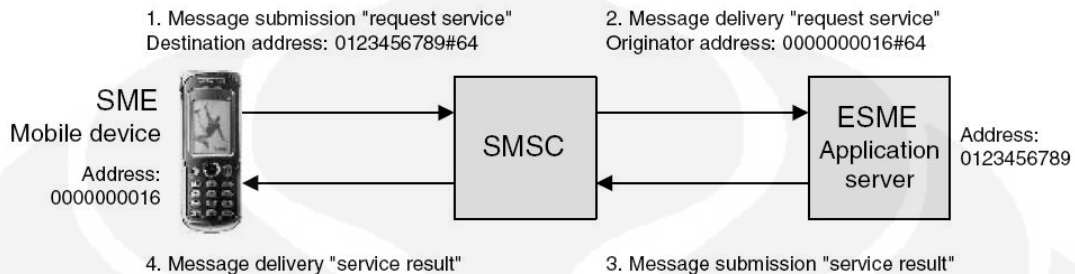
Dengan SMS, beberapa mode mungkin dapat dibuat untuk pengalamatan penerima pesan. Mode pengalamatan yang paling umum terdiri dari penggunaan format MSISDN [ITU-E.164] (misalnya +62811955955).

Sebuah pengalamatan opsional lainnya pada fitur SMS terdiri dari penyampaian informasi *sub-address* sebagai bagian dari pesan. Informasi *sub-address* ditambahkan pada bagian akhir dari pesan pada penerima oleh SME asal sebelum pesan dikirim. Ketika skema ini dipilih, SMSC akan mengekstrak informasi alamat dari alamat penerima yang spesifik untuk pesan yang akan dikirimkan dan akan menambahkan alamat asal dari pesan tersebut untuk dikirimkan. Fitur opsi ini dapat dimanfaatkan untuk menjaga identifikasi sesi untuk pertukaran pesan atau untuk mengidentifikasi kode layanan tertentu dimana pesan tersebut terhubung. Informasi *sub-addressing* dipisahkan dari informasi pengalamatan normal oleh penanda batas "#" (karakter ini adalah bagian dari pengalamatan *sub-addressing*) seperti ditunjukkan oleh Gambar 2.6.



Gambar 2.6 Format Sub-addressing [9]

Informasi *sub-addressing* adalah kombinasi dari digit karakter ”#” dan ”*”. Sebagai contoh, sebuah *handset* bergerak dapat saja meminta *update* dari informasi cuaca dengan mengirimkan pesan ke *server* aplikasi. Dengan mengidentifikasi layanan yang diminta maka dapat dispesifikasikan sebagai bagian dari informasi *sub-addressing* seperti ditunjukkan pada Gambar 2.7.



Gambar 2.7 Contoh Sub-addressing [10]

2.2.6. Protocol Layer

Protocol stack untuk SMS disusun atas 4 buah layer, yaitu: Layer Aplikasi, Layer Transfer, Layer Relay dan Layer Link.

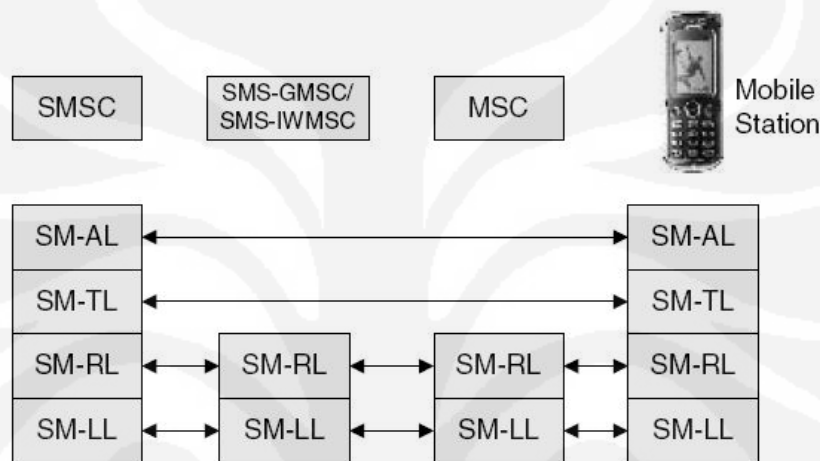
Layer aplikasi diterapkan pada SME dalam bentuk aplikasi perangkat lunak yang dapat mengirim, menerima dan mengsumsikan isi dari pesan (misalnya: penyunting pesan/editor, permainan dan lain-lain). Layer aplikasi juga dikenal sebagai SM-AL atau *Short-Message Application-Layer*.

Layer transfer, pesan dianggap sebagai sebuah urutan dari *oktet* yang mengandung informasi seperti panjang pesan, asal pengirim pesan atau penerima pesan, tanggal penerimaan, dan lain-lain. Layer transfer juga dikenal sebagai SM-TL atau *Short-Message Transfer-Layer*.

Layer relay memungkinkan pesan dapat ditransportasikan diantara operator jaringan yang berbeda. Pada layer relay, SMSC menangani dua buah fungsi sebagai tambahan dari kemampuan *switching* biasanya. Fungsi pertama disebut dengan *SMS gateway MSC* (SMS-GMSC) terdiri dari menerima pesan dari SMSC dan menanyakan HLR untuk memperoleh informasi *routing* dan selanjutnya mengirimkan pesan ke jaringan penerima. Fungsi kedua disebut dengan *SMS InterWorking MSC* (SMS-IW MSC) terdiri dari menerima pesan dari jaringan bergerak dan mengirimkannya kepada SMSC yang melayani. Layer relay dikenal dengan SM-RL atau *Short-Message Relay-Layer*.

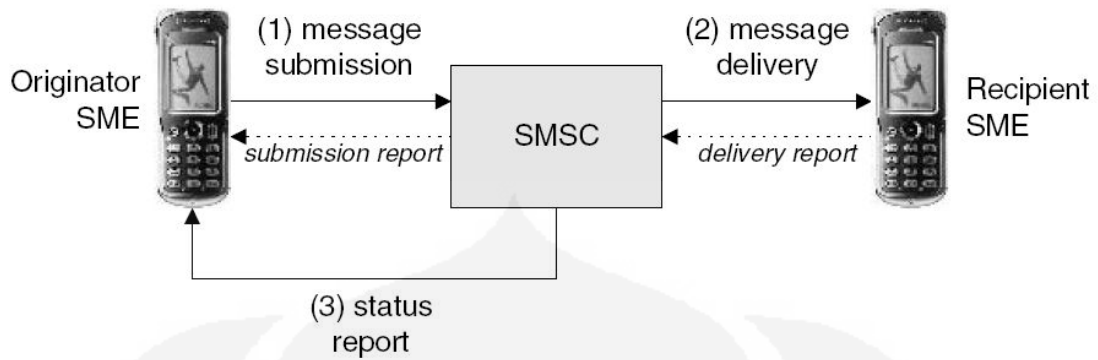
Layer link memungkinkan transmisi dari pesan pada tingkat fisik. Untuk tujuan ini, pesan di lindungi untuk mengatasi *error channel* pada tingkat rendah. Link layer ini biasa dikenal dengan SM-LL atau *Short-Message Link-Layer*.

Stack dari *transport protocol layer* ditunjukkan pada Gambar 2.8. Untuk tujuan transport, sebuah aplikasi memetakan isi pesan dan mengasosiasikan instruksi pengiriman pada sebuah *Transfer Protocol Data Unit* (TPDU) pada *layer transfer*. Sebuah TPDU disusun dari berbagai parameter yang mengindikasikan tipe pesan, menspesifikasikan apakah ada permintaan laporan status, mengandung bagian dari pesan teks dan lain-lain. Setiap parameter di prefikskan dengan singkatan TP untuk *Transfer Protocol* seperti TP-Message-Type-Indicator (disingkat TP-MTI), TP-Status-Report-Indicator (disingkat TP-SRI), TPUser-Data (disingkat TP-UD) dan lain-lain.

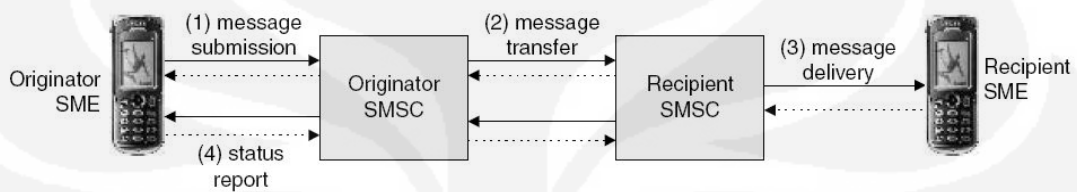


Gambar 2.8 SMS Protocol Stack [11]

Pada *layer transfer*, pertukaran pesan dari SME asal pengirim kepada SME penerima terdiri dari tiga langkah. Ketiga langkah tersebut ditunjukkan oleh Gambar 2.9 untuk pelanggan pada operator yang sama dan Gambar 2.10 untuk pelanggan dengan operator berbeda. Setelah pembuatan SME asal, pesan dikirimkan ke SMSC (langkah 1). SMSC akan memverifikasi dengan elemen jaringan lain bahwa pengirim pesan diperbolehkan untuk mengirim pesan (misalnya: pulsa yang cukup untuk mengirim pesan, pelanggan yang berada pada jaringan yang sama, dan lain-lain). SMSC akan mengirimkan pesan ke penerima (langkah 2). Jika penerima tidak ada untuk pengiriman pesan, maka SMSC akan menyimpan pesan tersebut sementara sampai SME penerima ada atau sampai validitas pesan sudah kadaluarsa. Ketika pesan sampai ke tujuan atau ketika pesan dihapus dari jaringan karena tidak sampai, sebuah laporan pesan akan dikirimkan ke SME pengirim (langkah 3), hanya jika pengirim pesan meminta laporan pengiriman ketika mengirim pesan.



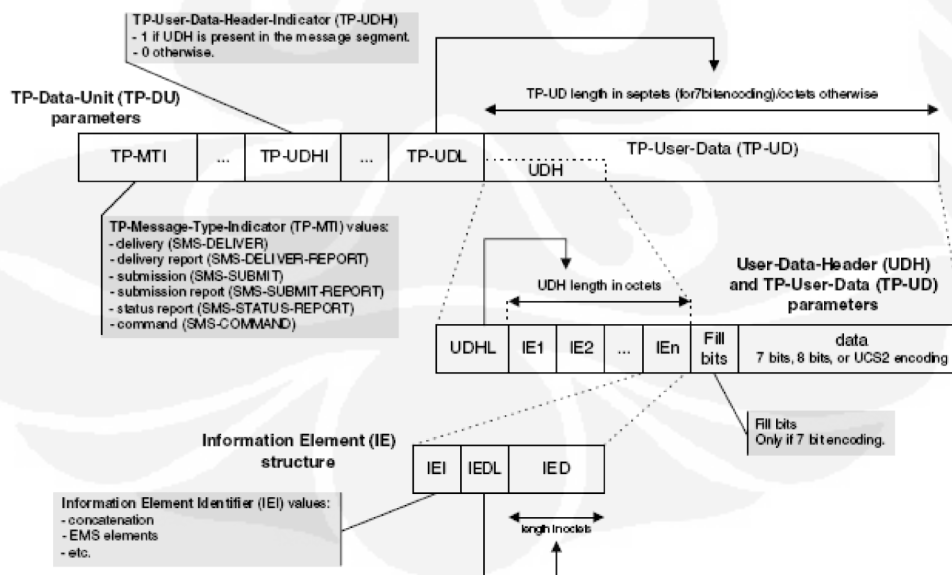
Gambar 2.9 Transfer pesan diantara dua SME pada operator yang sama [12]



Gambar 2.10 Transfer pesan diantara dua operator yang berbeda (lintas operator) [13]

2.2.7. Transport Protocol Data Unit

Transport Protocol Data Unit (TPDU) ditugaskan untuk setiap transaksi yang dapat terjadi antara sebuah SME dengan SMSC pada *layer transfer*. Tergantung pada setiap tipe, sebuah TPDU disusun dari nomor yang bervariasi dari parameter yang disusun menurut *layout* TPDU yang telah didefinisikan sebelumnya. Representasi dari TPDU ditunjukkan dalam Gambar 2.11.



Gambar 2.11 Format TPDU [14]

Parameter tingkat tinggi di TPDU (yang namanya disebutkan dimulai dari TP) menginformasikan pada tipe transaksi TP-Message-Type-Indicator, kehadiran dari

elemen biner pada pesan seperti misalnya penggabungan instruksi TP-User-Data-Header-Indicator dan lain-lain.

2.3.LINUX DAN OPEN SOURCE

Personal Computer dan system operasinya sudah mencapai jalan yang jauh sejak akhir 1970-an, ketika komputer pertama keluar dipasaran. Pada waktu itu, hanya dapat menekan-nekan saklar didepan mesin yang kemudian dapat menjalankan program tersebut dan hanya program tersebut sampai nantinya secara manual memuat program lainnya, pada waktu dimana program pertama tersebut dikeluarkan dari sistem. Pada saat ini program *personal computer* menyediakan sistem grafis yang handal dan kaya akan antar muka yang ramah pengguna yang membuat sangat mudah untuk memilih dan menjalankan berbagai variasi dari perangkat lunak secara bersamaan.

Meskipun kekuatan dan kemampuan dari *personal computer* saat ini sangat jauh kemampuannya dari pertama kali dibuat, ide dari penulisan perangkat lunak dan membagi perangkat lunak secara bebas dengan yang lainnya tidak pernah pergi. Meskipun itu tidak pernah dipublikasikan karena tidak ada yang mendapatkan uang dari pembuatannya, perangkat lunak bebas (dan bahkan beserta dengan kode sumbernya) terus untuk dapat tersedia dari klub komputer, sistem buletin dan jaringan komputer seperti internet pada saat ini. Pergerakan perangkat lunak bebas akhirnya mekar menjadi tiga bagian yang penting:

- Pembuatan proyek GNU[15] oleh Richard Stallman pada tahun 1983, sebuah proyek yang ditujukan untuk mengembangkan perangkat lunak dimana kode sumbernya akan selalu untuk tersedia secara bebas.
- Pengumuman *Free Software Foundation* (FSF)[16], pada awalnya untuk pengumpulan dana untuk proyek GNU.
- Pengenalan proyek sistem operasi bebas pada tahun 1991 yang dikemudian hari dikenal dengan Linux, oleh pelajar perangkat lunak komputer yang bernama Linus Torvalds.

Kemunculan Linux, sistem operasi PLB (Perangkat Lunak Bebas), adalah penanda peristiwa pada dunia komputer modern. Hari ini, Linux adalah sistem operasi yang paling populer yang sering digunakan pada lingkungan server, dan lebih cepat peningkatan jumlah pengguna yang sangat signifikan sebagai sistem operasi pada *personal computer* pada penggunaan perumahan. Alasan yang paling umum menggunakan Linux [17] menurut adalah:

- **Sangat Kuat, desain yang modern:** Linux didesain agar dapat menjalankan banyak program pada saat bersamaan dan menyediakan layanan yang dapat digunakan pada komputer.
- **Tersedianya kode sumber yang bebas:** Terlepas dari sistem operasi yang digunakan saat ini, terkadang kita mendapati masalah dengan aplikasi atau bekerjanya berbagai hal. Bagaimanapun, jika menggunakan sebuah sistem operasi berbayar seperti Microsoft atau Apple, pembaruan perangkat lunak hanya didapat dari Microsoft atau Apple. Linux adalah terbuka dan bebas, jadi apabila anda tidak suka dengan cara kerja Red Hat's Linux, sejauh mana biayanya, atau tipe dari *customer support* yang tersedia, anda dapat selalu berpindah ke Novell SUSE Linux, Mandriva Linux, atau mungkin Ubuntu Linux.
- **Ribuan aplikasi yang bebas dan handal:** Butuh aplikasi pemrosesan kata? Download dan instal aplikasi OpenOffice, AbiWord, Kwrite, atau lusinan lainnya. Butuh aplikasi database? *Download* dan instal MySQL, PostgreSQL, atau banyak yang lainnya. Butuh aplikasi pembuat grafis dan memanipulasi foto digital? Tidak ada yang lebih baik dari GIMP (*GNU Image Manipulation Program*). Jika apapun lainnya, masalah yang dihadapi oleh Linux hanya terlalu banyak pilihan aplikasi dan tidak satupun yang mengeluarkan uang.
- **Mendukung standarisasi:** Linux dan aplikasi Linux didesain untuk mendukung standarisasi, karena standard adalah bahasa komersial intelektual bebas. Aplikasi Linux mendukung aplikasi modern dan format data untuk audio, multimedia dan format dokument, data spreadsheet, dan banyak lagi. Karena Linux terbuka dan bebas, maka tidak ada kepemilikan pribadi atas aplikasi Linux dan format aplikasi.
- **Biaya kepemilikan yang sangat rendah:** Jika anda ingin menggunakan Linux untuk aplikasi bisnis, bebas untuk mendapatkannya dan banyak praktisi Linux yang siap membantu untuk mengaplikasikan apapun yang anda butuhkan. Tidak dikenakan biaya lisensi.
- **Stabil, Powerful, dan bebas virus:** Linux sangat matang, system multi-user yang dapat diandalkan, stabil, sistem keamanan yang built-in dan bebas virus.

Biasanya menggunakan Linux membutuhkan pengetahuan khusus, tetapi itu pada dasarnya bukan merupakan masalah lagi. Distribusi Linux seperti Ubuntu membuat Linux mudah – atau lebih tepatnya mereka membuat Linux tidak terlihat sulit.

2.3.1. Ubuntu Linux

Ubuntu Linux[18] adalah sebuah distribusi Linux yang dibuat pada tahun 2004 dan memfokuskan pada kebutuhan pengguna. Ubuntu Linux adalah produk dari proyek Ubuntu yang disponsori oleh Canonical, Ltd[19], sebuah perusahaan yang didirikan oleh Mark Shuttleworth, seorang pengusaha Afrika Selatan yang sukses, pengembang Debian Linux yang sudah lama dan pendukung *open source* umum. Ubuntu berpondasikan sistem distribusi Debian Linux yang menggunakan GNOME sebagai antarmuka grafis pengguna (GUI).

2.3.2. Alasan Memilih Ubuntu Linux

Beberapa alasan menggunakan Ubuntu Linux [20] adalah:

- **Teratur, perangkat lunak dengan rilis terkini:** Kernel Linux dan ribuan paket perangkat lunak yang meliputi pengguna Linux dan lingkungan administratif secara konstan di update.
- **Komitmen terhadap kualitas:** Kualitas distribusi Linux terdiri dari dua hal, yaitu: seberapa baikkah kualitasnya dan komitmen pada solusi terhadap permasalahan yang muncul. Dalam kedua hal tersebut, Ubuntu bersinar. Setiap rilis setelah melalui pengujian internal yang luas oleh tim Ubuntu dan pengujian publik yang luas melalui kandidat rilis. Setelah muncul rilis, update untuk rilis tersebut dihantarkan dengan waktu minimum 18 bulan (sesuai dengan kebutuhan tentunya).
- **Dukungan komunitas dan komersial:** Banyak dukungan untuk distribusi Linux apapun datang dari komunitas penggunanya, dan sulit untuk mengalahkan keinginan besar dan komitmen dari komunitas Ubuntu.
- **Kemudahan mengambil dan mendapatkan *update aplikasi*:** Sangat penting untuk dapat menjaga agar sistem yang terinstal untuk selalu *up-to-date*. Ubuntu menyediakan alat yang hebat untuk mengingatkan pengguna ketika ada *update* dan membuatnya mudah untuk mendapatkan dan menginstalnya.
- **Fokus pada kegunaannya:** Ubuntu menyatakan dirinya sebagai "*Linux for human being*" dan menyediakan grafis yang disesuaikan, dekorasi, dan skema warna yang didesain untuk menyediakan lingkungan desktop yang atraktif dan berguna untuk orang sesungguhnya untuk penggunaan pribadi dan untuk

bekerja. Seperti distribusi Linux lainnya, Ubuntu dapat disesuaikan dengan keinginan pengguna.

- **Fokus pada internasionalisasi:** Untuk sebagian orang, dapat menjadi suatu kejutan bahwa ada sebagian orang yang tidak dapat berbahasa Inggris dan bahasa yang mereka gunakan tidak menggunakan alfabet dan karakter berbeda dari umumnya. Ubuntu memfokuskan secara ekstrim pada dukungan usaha terjemahan dan menyediakan distribusi Linux yang dapat digunakan oleh setiap orang pada planet ini dengan bahasa asli mereka, dengan karakter set asli mereka.
- **Komunitas yang aktif dan terlibat dalam pengembangan:** Sulit untuk mencari komunitas pengguna Linux yang sangat aktif, dinamis dan terlibat aktif pada pengembangan Linux selain komunitas disekitar Ubuntu. Hal ini dapat dilihat dari banyaknya tempat untuk bertanya, banyaknya kemungkinan untuk mendapatkan jawaban, dan banyaknya pengguna berpengalaman yang ramah dan mau untuk berbagi.

2.4.WEB SERVER

Keuntungan yang paling signifikan dari teknologi komputer adalah apa yang disebut dengan aplikasi pamungkas – sesuatu yang signifikan, unik, *powerful*, memaksa orang untuk ikut mengalir bersama dengan teknologi tersebut. *Web server* saat ini adalah aplikasi yang menjadi tulang punggung (bagian utama) dari teknologi internet di era teknologi informasi yang saat ini sudah sangat dikenal oleh sebagian besar orang. Dengan *web server*, berbagai aplikasi dapat dijalankan mulai dari situs website informasi, blog, perbankan, email, dan lain sebagainya.

Web server pertama kali dibuat oleh Tim Berners-Lee di CERN (European Centre for Nuclear Research) di Jenewa, Swiss pada tahun 1989. *World Wide Web* kemudian digagas oleh peneliti Tim Berners-Lee dan Robert Cailliau untuk dijadikan standarisasi yang bertujuan untuk saling berbagi informasi hasil penelitian oleh CERN. Dikemudian hari pemerintah Amerika juga membuat aplikasi yang sama melalui NCSA (National Center for Supercomputing Applications) [21]. Karena aplikasi tersebut dikembangkan oleh pemerintah dengan menggunakan dana pajak dari pembayar pajak di Amerika, maka kode sumbernya dapat di download secara cuma-cuma oleh semua orang. Kode tersebut ditulis dalam bahasa C dan dapat di download di website NSCA[22]. Dari kode-kode inilah akhirnya sebuah web

server yang kini paling banyak digunakan di dunia saat ini mulai dikembangkan. *Web server* tersebut adalah Apache[23] yang untuk pertama kali di rilis pada tahun 1995.

Apache saat ini ditulis dan dikembangkan oleh suka relawan programmer dimana kode sumber dan aplikasinya dapat di download secara bebas tanpa harus membayar lisensi apapun atau yang biasa disebut dengan PLB. Perkembangan Apache kini sudah sangat jauh signifikan, sehingga banyak pengelola situs dan penyedia jasa *hosting* mempercayakan Apache sebagai *web server*-nya. Sehingga Apache sudah tidak diragukan lagi kehandalannya.

2.4.1. Prinsip Kerja Web Server

Ketika menghubungi URL (*Uniform Resource Locator*) dari sebuah situs yang dituju misalnya **http://www.yahoo.com/**. Dengan mengetikkan alamat tersebut pada perambah internet, maka komputer akan mengirim pesan melalui internet kepada mesin di alamat tersebut. Mesin yang dituju diharapkan dapat melayani permintaan, kemudian koneksi internet bekerja dan siap untuk menerima dan menjawab permintaan tersebut.

URL terdiri atas tiga bagian, misalnya URL **http://www.yahoo.com/** tersebut terdiri dari:

`<method>://<host>/<absolute path URL (apURL)>`

Jadi pada contoh ini, `<method>` adalah **http**, artinya perambah tersebut harus menggunakan HTTP (*Hypertext Transfer Protocol*); `<host>` adalah **www.yahoo.com**; dan `<apURL>` adalah `"/`, artinya direktori teratas dari situs tersebut. Menggunakan HTTP/1.1 perambah akan mengirimkan permintaan berikut melalui internet:

`GET / HTTP/1.1`

`Host: www.yahoo.com`

Permintaan akan disampaikan melalui port 80 (port HTTP pada umumnya) pada host **www.yahoo.com**. Pesan terdiri dari tiga bagian, yaitu: metode (sebuah metode HTTP, bukan metode URL), dalam kasus ini adalah **GET**, tetapi dapat sama dengan **PUT**, **POST**, **DELETE**, atau **CONNECT**; URI-nya adalah `"/`; dan versi dari protokol yang kita gunakan. Kemudian diserahkan kepada web server pada host tersebut untuk melakukan sesuatu terhadap permintaan dari pesan ini [24].

Dapat dikatakan disini bahwa semua urusan dari web server adalah menerjemahkan sebuah URL menjadi sebuah nama file, dan kemudian mengirim balik file tersebut melalui internet, atau kepada sebuah nama program, dan kemudian menjalankan program tersebut dan mengirimkan kembali hasil keluarannya. Itulah fungsi utamanya, selebihnya hanya riasan saja.

2.4.2. Prinsip Kerja Apache

Apache adalah program yang berjalan dibawah sistem operasi yang cocok untuk *multitasking*. Sebagai contohnya Windows 95/98/NT/XP/Vista atau yang biasa disebut dengan Win32, Unix dan Linux, dimana aplikasi dalam bentuk binary disebut dengan ***apache.exe*** (untuk versi apache 2 disebut dengan ***apache2.exe***) di Win32 dan ***httpd*** di Unix atau Linux yang biasanya berjalan dibelakang. Setiap salinan ***httpd/apache*** yang sudah dimulai memiliki perhatian untuk diarahkan ke sebuah website, dimana untuk lebih praktis, sebuah direktori. Pada umumnya dalam instalasi ***httpd/apache*** terdiri dari empat sub-direktori:

- **Conf:** Terdiri dari konfigurasi file-file, dimana file ***httpd.conf*** adalah yang paling penting. File ini adalah referensi dari file konfigurasi dari apache web server.
- **Htdocs:** Terdiri dari skrip HTML untuk diberikan kepada site klien. Direktori ini dan direktori-direktori dibawahnya, ruang web, dapat diakses oleh siapapun melalui internet dan oleh karena itu resiko keamanan perlu untuk dipertimbangkan.
- **Logs:** Terdiri dari log data, keduanya untuk akses dan error.
- **Cgi-bin:** Terdiri dari skrip CGI. Ini adalah program atau skrip *shell* yang ditulis oleh dan untuk *webmaster* yang dapat dieksekusi oleh Apache atas nama klien. Karena yang paling penting adalah untuk alasan keamanan, maka direktori ini tidak pada ruang web.

Pada kondisi *idle*, Apache tidak melakukan apapun kecuali mendengarkan alamat IP dan port TCP atau port yang dispesifikasikan pada file konfigurasi. Ketika permintaan datang pada port yang tepat, Apache menerima permintaan HTTP dan menganalisa *header*-nya. Kemudian mengaplikasikan aturan yang telah dibuat pada file konfigurasi dan mengambil aksi yang sepatutnya.

2.4.3. TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekumpulan protokol yang memungkinkan komputer untuk saling berkomunikasi satu dengan yang lainnya melalui jaringan. Dua protokol diantara yang paling penting, tetapi masih banyak yang lainnya.

TCP/IP hanya berguna jika ada jaringan. Setiap komputer pada jaringan yang ingin menggunakan TCP/IP memiliki alamat IP, misalnya 192.168.123.1 (format sesuai dengan standarisasi IPv4)

Alamat IP terdiri dari empat bagian, dipisahkan oleh titik. Setiap bagian melambangkan *byte*, jadi seluruh alamat memiliki panjang empat byte. Karena itu maksimum nilai yang didapat adalah diantara 0-255.

Meski tidak dibutuhkan oleh protokol, menurut konvensi terdapat baris pemisah disuatu tempat didalam angka tersebut: kekiri menunjukkan nomor jaringan dan kekanan menunjukkan nomor *host*. Dua buah mesin pada jaringan fisik yang sama – biasanya pada sebuah LAN (*Local Area Network*) – pada umumnya memiliki nomor jaringan yang sama dan berkomunikasi secara langsung menggunakan TCP/IP.

Batas pemisah ditentukan oleh angka pertama dari empat angka paling awal: jika nilai angka pertama adalah:

- 0-127 (*byte* pertama adalah 0xxxxxxx dalam biner), garis pemisah adalah setelah angka pertama, dan angka ini dikategorikan jaringan Class A. Terdapat beberapa jaringan Class A –yang dapat digunakan sebanyak 125- tetapi setiap jaringan mendukung 16.777.214 *host*.
- 128-191 (*byte* pertama adalah 10xxxxxx dalam biner), garis pemisah adalah setelah angka kedua, dan ini dikategorikan jaringan Class B -16.382- dan setiap jaringan mendukung 65.534 *host*.
- 192-223 (*byte* pertama adalah 110xxxxx dalam biner), garis pemisah adalah setelah angka ketiga, dan ini dikategorikan jaringan Class C -2.097.150- tetapi setiap jaringan mendukung 254 *host*.

Nilai sisanya adalah dimana angka pertamanya diantara 224-255, tidak ada hubungannya disini. Nomor jaringan –bagian disebelah kiri- yang semuanya 0 atau 1 dalam biner sudah dipesan untuk keperluan khusus dan karena itu tidak ada hubungannya disini. Alamat ini adalah sebagai berikut:

- 0.x.x.x
- 127.x.x.x
- 128.x.x.x
- 191.255.x.x
- 192.0.0.x
- 223.255.255.x

Sering kali untuk *membypass* peraturan jaringan Class A, B dan C menggunakan *subnet masks*. Ini memungkinkan untuk lebih jauh membagi lagi jaringan dengan menggunakan kelebihan bit untuk nomor jaringan dan lebih sedikit untuk nomor *host*.

TCP/IP memungkinkan dua buah komputer dengan alamat IP X dan Y untuk dapat saling berkomunikasi. Jika X dan Y berada pada jaringan yang sama, dan dikonfigurasi dengan benar sehingga mereka berada pada nomor jaringan yang sama dan berbeda nomor *hostnya*, sehingga komunikasi TCP/IP dapat terjadi diantara X dan Y dan dapat saling mengirimkan paket satu dengan yang lainnya melalui jaringan lokal, secara fisik tanpa kesulitan.

Jika nomor jaringan tidak sama, TCP/IP mengirim paket ke sebuah *router*, sebuah mesin yang khusus dapat mencari dimana mesin lainnya dan mengirimkan paket ke mesin tujuan. Komunikasi ini dapat melalui internet atau dapat terjadi pada WAN (*Wide Area Network*).

Terdapat dua cara komputer berkomunikasi menggunakan TCP/IP:

- UDP (*User Datagram Protocol*): Sebuah cara untuk mengirim paket tunggal dari satu mesin ke mesin yang lain. Ini tidak dijamin dapat sampai ke tujuan, dan tidak ada *acknowledgement* dari penerima.
- TCP (*Transmission Control Protocol*): Sebuah cara untuk berkomunikasi diantara dua komputer. Dapat diandalkan untuk mengirimkan paket pesan berbagai ukuran. TCP lebih baik dari UDP.

2.4.4. Apache Dan TCP/IP

Berbagai jenis *server* dengan protokolnya yang biasa ditemui saat ini adalah:

- *Network News Transfer Protocol* (NNTP): news
- *Simple Mail Transfer Protocol* (SMTP): mail
- *Domain Name Service* (DNS)
- HTTP: *World Wide Web*

Server dapat menentukan bagaimana menangani berbagai permintaan berbeda karena empat *byte* alamat IP akan mengarahkan permintaan pada antarmukanya dan akan diikuti dengan nomor *port*. Layanan yang berbeda akan terhubung dengan *port* yang berbeda:

- NNTP: nomor *port* 119
- SMTP: nomor *port* 25
- DNS: nomor *port* 53
- HTTP: nomor *port* 80

Apache *web server* secara pada umumnya akan melayani permintaan dengan protokol HTTP, oleh karena itu pada umumnya akan melayani permintaan yang datang pada port 80.

2.4.5. Koneksi Dari Sisi Klien

Ketika klien ingin mengakses sebuah URL, misalnya **http://www.apache.org** maka perambah (*web browser*) akan mengamati permintaan klien ini. Permintaan ini diawali dengan **http:** perambah akan menyimpulkan untuk menggunakan protokol HTTP. Tanda **"/"** mengatakan bahwa URL adalah absolut, bukan sebuah URL relatif. Bagian selanjutnya adalah nama server, **www.apache.org**. Klien kemudian akan mengontak nama server melalui DNS server, dimana DNS server akan mengubah nama ini menjadi sebuah alamat IP. Pada saat tulisan ini dibuat, alamat IP yang diberikan oleh DNS server adalah 140.211.11.130.

Salah satu cara untuk mengecek validitas dari nama host adalah dengan cara sebagai berikut (tanda **"\$"** menandakan *command line* untuk sistem operasi Linux sedangkan tanda **">"** menandakan *command line* pada Windows):

Pada sistem operasi Linux:

```
$ ping -c 5 www.apache.org
```

Pada sistem operasi Windows:

```
> ping -c 5 www.apache.org
```

Jika host terhubung ke internet, responnya akan menjadi :

```
PING www.apache.org (140.211.11.130): 56 data bytes
64 bytes from taz.apache.org (140.211.11.130): icmp_seq=0 ttl=247 time=1380 ms
64 bytes from taz.apache.org (140.211.11.130): icmp_seq=1 ttl=247 time=1930 ms
64 bytes from taz.apache.org (140.211.11.130): icmp_seq=2 ttl=247 time=1380 ms
64 bytes from taz.apache.org (140.211.11.130): icmp_seq=3 ttl=247 time=1230 ms
64 bytes from taz.apache.org (140.211.11.130): icmp_seq=4 ttl=247 time=1360 ms
--- www.apache.org ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss round-trip min/avg/
max = 1230/1456/1930 ms
```

Alamat web **http://www.apache.org** tidak memasukkan alamat *port*, karena alamat *port* yang digunakan adalah alamat *port default*, sehingga tidak perlu memasukkan nomor *port* pada perambah. Jika klien menginginkan alamat *port* yang lain, maka alamat *port* harus disebutkan pada perambah setelah sebuah titik dua – misalnya **http://www.apache.org:8000/**. URL akan selalu memasukkan tanda **"/"** pada akhir alamat setelah nomor *port* meski terkadang klien tidak menambahkannya, perambah akan menambahkannya secara otomatis. Jika klien menginginkan sebuah dokumen tertentu pada *server*, misalnya **/some/where/foo.html** pada *port* 8000, maka URL-nya akan menjadi **http://www.apache.org:8000/some/where/foo.html**. Selanjutnya permintaan

tersebut akan diterima oleh *server* dan akan melakukan sesuai dengan konfigurasi yang diberikan oleh administrator untuk merespon permintaan tersebut.

Klien akan membuat sebuah koneksi TCP pada nomor *port* 8000 pada alamat IP 140.211.11.130, dan mengirimkan pesan berikut pada koneksi (jika menggunakan HTTP/1.0):

```
GET /some/where/foo.html HTTP/1.0<CR><LF><CR><LF>
```

Carriage Return (CR) dan *Line Feed* (LF) ini sangat penting karena keduanya memisahkan *header* HTML dan *body*. Jika permintaannya sebuah metode **POST**, kemudian akan diikuti dengan data. Server kemudian akan merespon kembali dan menutup koneksi. Untuk melihat bagaimana prosesnya, dengan memasukkan perintah berikut pada *command line*:

```
$ telnet www.apache.org 80
> telnet www.apache.org 80
```

Telnet pada umumnya meminta nama *host* diikuti dengan nomor *port*. Setelah terhubung, ketik perintah berikut:

```
$ GET
```

Setelah memberikan perintah tersebut, maka akan memberikan output sebagai berikut:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more contributor
    license agreements. See the NOTICE file distributed with this work for additional
    information regarding copyright ownership. The ASF licenses this file to you under the
    Apache License, Version 2.0 (the "License"); you may not use this file except in
    compliance with the License. You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed
    to in writing, software distributed under the License is distributed on an "AS IS"
    BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
    the License for the specific language governing permissions and limitations under the
    License.
-->
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!--
      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      This file is generated from XML source: DO NOT EDIT!
      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    -->
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <link rel="stylesheet" href="./style/compressed.css" type="text/css"
media="screen, projection"/>
    <link rel="stylesheet" href="./style/style.css" type="text/css" media="screen,
projection"/>
    <!--[if IE]><link rel="stylesheet" href="./style/ie.css" type="text/css"
media="screen, projection"/><![endif]-->
```

```
<link rel="alternate" title="announce@apache.org Archives" type="application/atom+xml"
href="http://mail-archives.apache.org/mod_mbox/www-announce/?format=atom" />
  <meta name="author" content="The Apache Software Foundation" /><meta name="email"
content="apache.AT.apache.DOT.org" />
  <title>Welcome! - The Apache Software Foundation</title>
</head>
<body>
  .
  .
  === dihapus ===
  .
  .
  .
</body>
</html>
```

Berkas output pada bagian `<body>` tersebut sengaja di hapus beberapa bagian agar tidak terlalu panjang. Tujuannya adalah hanya untuk menunjukkan bagaimana prinsip kerja perambah internet bekerja pada sisi klien.

2.4.6. Koneksi Dari Sisi Server

Pada sisi *server* – dalam hal ini Apache *web server*. Apache akan mengambil URL dari internet yang diminta, dan kemudian akan memberikan sebuah nama *file*, kemudian mengirimkan *file* tersebut kembali ke klien melalui internet. Pada intinya dari sisi *server* adalah bagaimana *server* dapat melayani permintaan klien dengan cara memberikan *file*, data atau skrip kembali kepada klien sebagai respon terhadap permintaan dari klien.

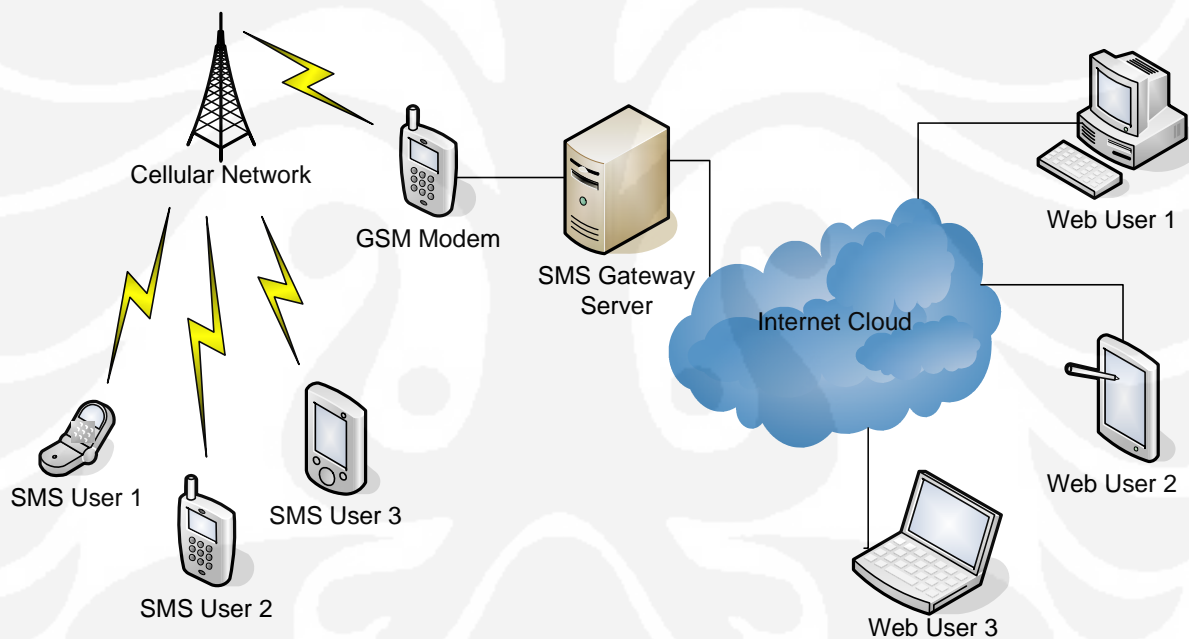
BAB 3

PERANCANGAN SMS GATEWAY

3.1. PERANCANGAN SISTEM

SMS Gateway adalah suatu sistem antar muka antara web/internet dengan sistem seluler dengan menggunakan salah satu fitur yang paling banyak digunakan yaitu SMS. Dengan *SMS Gateway* ini, pengguna dapat mengirimkan atau menerima SMS dengan antar muka web. Pengguna dapat mengirim SMS ke banyak tujuan (*SMS Broadcasting*) atau menerima SMS dari pengguna dan dapat menjawabnya secara otomatis (*Auto Replying SMS*). Dengan antar muka web, maka penggunaannya akan menjadi mudah, siapapun dapat menggunakannya tanpa kesulitan.

Gambaran perancangan sistem kerjanya dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Perancangan sistem kerja *SMS Gateway*.

SMS Gateway Server adalah sebuah perangkat yang akan menjadi *server* antar muka yang menghubungkan web dengan operator seluler. Aplikasi ini bersifat lintas operator, jadi pengguna (*user*) dapat menggunakan operator seluler apapun. *SMS Gateway Server* dapat berupa sebuah komputer pribadi biasa (PC) atau dapat pula sebuah komputer khusus sekelas *server*. *SMS Gateway Server* akan dibuat dengan sistem operasi Ubuntu Linux yang akan dilengkapi dengan Apache Web Server untuk antar muka web. *SMS Gateway Server* juga akan dilengkapi dengan GSM modem untuk menghubungkannya dengan operator seluler. Untuk koneksi dengan internet dapat dilakukan dengan menyewa ISP (*Internet Service*

Provider) yang dilengkapi dengan sebuah IP *dedicated*. Dengan IP *dedicated*, maka server dapat diakses oleh *web user* dan mengakses jaringan internet.

GSM Modem adalah sebuah perangkat yang akan mengirimkan dan menerima SMS dari *SMS Gateway Server*. *GSM Modem* akan mengirimkan atau menerima pesan SMS melalui *Air Interface* dari operator seluler. Pesan SMS yang dikirimkan akan melalui operator seluler untuk sampai ke penerima demikian juga sebaliknya, pesan SMS dari pengguna SMS (*SMS User*) dapat sampai ke pengguna web (*Web User*) dengan antar muka dari operator seluler yang akan diterima oleh *GSM Modem* dan akan dibaca oleh *SMS Gateway Server* dan kemudian akan dimasukkan dalam database.

3.2. INSTALASI UBUNTU LINUX

3.2.1. Kebutuhan Sistem

Pada saat tulisan ini ditulis, Ubuntu Linux[25] telah sampai pada rilis Ubuntu 7.10 yang dinamakan rilis Gutsy Gibbon. Ubuntu Linux secara periodik berkomitmen setiap enam bulan sekali mengeluarkan rilis terbaru sejak Oktober 2004.

Sistem arsitektur prosesor yang saat ini didukung oleh Ubuntu Linux adalah:

- I386 (Intel 386) atau prosesor dari Intel, AMD, Cyrix, dan lainnya
- G3 - atau lebih tinggi – prosesor pada PowerPC (PPC) sistem Apple Machintosh
- 64-bit AMD atau EM64T – prosesor yang mendukung sistem komputasi 64 bit, termasuk Athlon64, Opteron, EM64T Xeon, dan seri Intel DualCore/Core2/Core2duo)
- Sistem Sun UltraSPARC

Perangkat keras yang dibutuhkan untuk instalasi sistem operasi Ubuntu Linux 7.10 (Gutsy Gibbon):

- RAM minimum 384 MB
- Prosessor dengan kecepatan minimum 700 MHz
- CD-ROM drive
- Kartu Ethernet
- Antar muka grafis VGA
- *Hardisk* minimum 3 GB

Untuk mendapatkan rilis terbaru dari Ubuntu Linux, dapat dilihat melalui website Ubuntu. Setiap rilis terbaru dari Ubuntu akan diumumkan melalui website tersebut. Ada dua cara untuk mendapatkan kopi dari rilis terbaru Ubuntu. Cara pertama adalah dengan

mendownload dari *server* Ubuntu langsung[26] – atau melalui *server mirror*-nya – dan cara yang kedua adalah dengan mendapatkan CD cuma-cuma yang ditawarkan Ubuntu[27]. Meskipun dapat saja membeli CD Ubuntu melalui websitenya.

3.2.2. Proses Instalasi Ubuntu

Untuk menginstal Ubuntu Linux, dibutuhkan CD yang sesuai dengan prosessor yang digunakan untuk di instal Ubuntu Linux. Terdapat dua buah jenis CD instalasi Ubuntu Linux, yaitu CD untuk prosessor 32 bit dan CD untuk prosessor 64 bit. Untuk mendapatkan salinan CD tersebut sudah dijelaskan pada bagian sebelumnya.

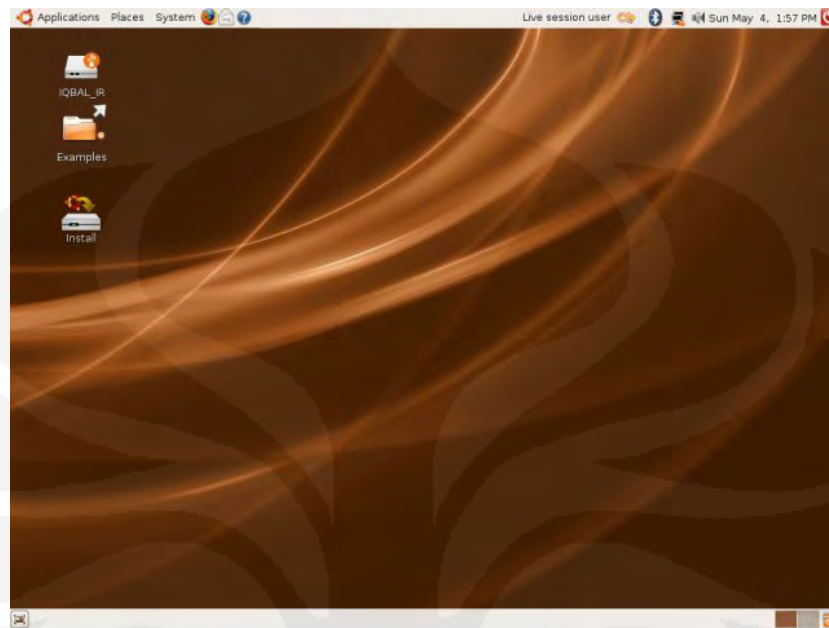
Jika sudah mendapatkan CD yang dibutuhkan, jalankan CD instalasi Ubuntu Linux pada komputer yang akan di instal Ubuntu. Pada saat pertama kali *booting*, akan muncul tampilan pada Gambar 3.2.



Gambar 3.2 Tampilan awal proses instalasi Ubuntu Linux.

Untuk memulai instalasi dan jika ingin melihat bagaimana fitur yang ada di Ubuntu, silahkan pilih "*Start Ubuntu in safe graphic mode*". Memilih pilihan ini pengguna dapat mencoba Ubuntu Linux dan melihat fitur-fitur yang ada didalamnya tanpa harus menginstal dan tanpa menyentuh sedikitpun sistem operasi yang sudah ada didalamnya. Jadi pilihan ini sangat aman bagi pengguna yang masih ingin mencoba, tanpa perlu menginstal.

Berikutnya setelah memilih pilihan *"Start Ubuntu in safe graphic mode"*, maka proses instalasi akan berlanjut dengan munculnya tampilan standar Ubuntu Linux seperti ditunjukkan pada Gambar 3.3.



Gambar 3.3 Tampilan standar Ubuntu Linux.

Proses selanjutnya untuk instalasi Ubuntu Linux adalah proses instalasi. Untuk melakukannya tinggal mengklik tombol *"Install"* pada dekstop. Setelah itu akan muncul jendela berikut seperti Gambar 3.4.



Gambar 3.4 Pemilihan bahasa yang ingin digunakan.

Pada tahap ini, proses instalasi berjalan dengan memilih bahasa yang ingin digunakan untuk Ubuntu yang akan di instal. Dalam hal ini bahasa yang ingin digunakan adalah bahasa

Inggris. Setelah memilih bahasa, proses berikutnya adalah menentukan lokasi spesifik keberadaan komputer yang ingin di instal Ubuntu seperti ditunjukkan pada Gambar 3.5.



Gambar 3.5 Pemilihan lokasi geografis.

Pada tahap ini, proses berjalan untuk menentukan lokasi tempat komputer yang akan di instal Ubuntu. Dalam hal ini, lokasi yang dipilih adalah Jakarta. Penentuan lokasi ini juga bertujuan untuk menentukan lokasi pembagian waktu yang akan digunakan pada Ubuntu. Secara otomatis waktu yang tertera pada jendela tersebut akan membaca waktu yang telah di set pada BIOS komputer. Proses selanjutnya adalah menentukan jenis *keyboard* yang digunakan seperti pada Gambar 3.6.



Gambar 3.6 Pemilihan jenis *keyboard* yang digunakan.

Pada proses ini akan diminta jenis *keyboard* yang ingin digunakan. Dalam hal ini akan dipilih jenis *keyboard* "US English". Proses selanjutnya adalah menentukan lokasi instalasi Ubuntu pada *hardisk* seperti pada Gambar 3.7.



Gambar 3.7 Pemilihan lokasi instalasi pada *hardisk*.

Pada proses ini akan diminta lokasi instalasi pada *hardisk*. Jika ingin menggunakan seluruh *hardisk* untuk di instal Ubuntu, silahkan pilih pilihan "*Guided – use entire disk*" diikuti dengan memilih *hardisk* yang dipilih. Jika ingin di instal pada lokasi kosong pada *hardisk*, silahkan pilih "*Guided – use the largest continues free space*". Jika ingin memilih sendiri dan menentukan sendiri besarnya *hardisk* yang ingin digunakan untuk instalasi Ubuntu, maka silahkan pilih "*Manual*".

Akhirnya sampai pada tahap akhir instalasi. Ini adalah tahap terakhir dari proses instalasi Ubuntu. Pada tahap ini akan diminta data pribadi pengguna sebagai administrator dari komputer yang akan di instal Ubuntu seperti pada Gambar 3.8.



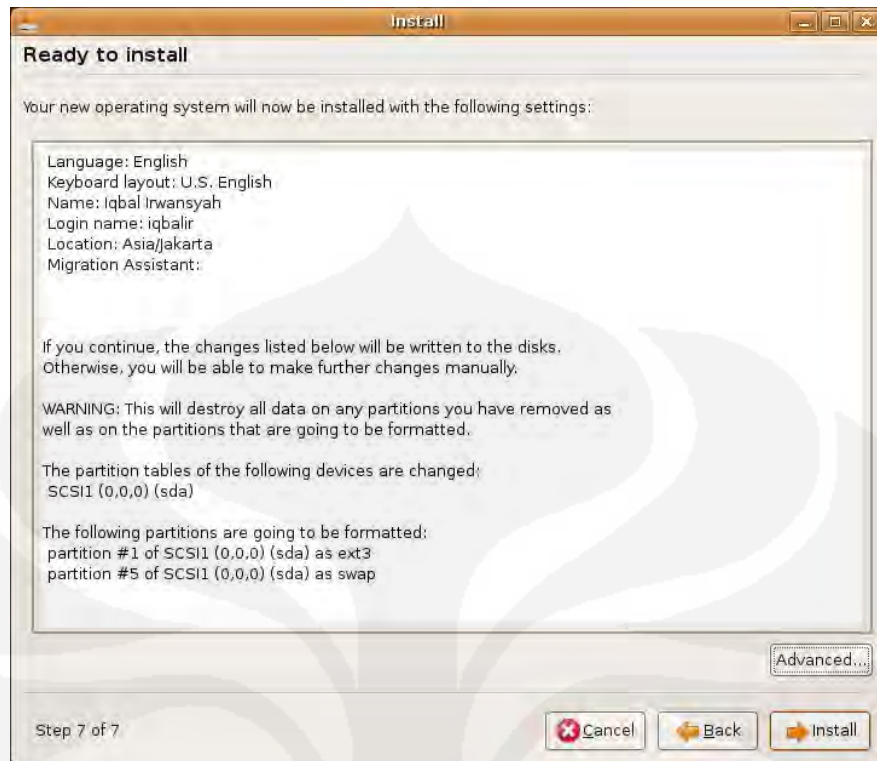
The screenshot shows the 'Install' window for Ubuntu. The title bar says 'Install'. The main heading is 'Who are you?'. Below it, there are four sections with input fields:

- 'What is your name?' with a text box containing 'Iqbal Irwansyah'.
- 'What name do you want to use to log in?' with a text box containing 'iqbalir'. Below this is a note: 'If more than one person will use this computer, you can set up multiple accounts after installation.'
- 'Choose a password to keep your account safe.' with two password boxes, both containing '*****'. Below this is a note: 'Enter the same password twice, so that it can be checked for typing errors.'
- 'What is the name of this computer?' with a text box containing 'unknown-localhost'. Below this is a note: 'This name will be used if you make the computer visible to others on a network.'

At the bottom left, it says 'Step 6 of 7'. At the bottom right, there are three buttons: 'Cancel' (with a red X icon), 'Back' (with a left arrow icon), and 'Forward' (with a right arrow icon).

Gambar 3.8 Mengisi data pribadi untuk administrator.

Setelah pengisian data pribadi, maka proses instalasi Ubuntu akan melakukan verifikasi data yang telah dimasukkan sebelumnya seperti pada Gambar 3.9. Jika semua data yang telah dimasukkan sudah sesuai, maka proses instalasi dapat berlanjut dengan instalasi Ubuntu ke dalam *hardisk*.



Gambar 3.9 Verifikasi akhir instalasi Ubuntu.

Setelah verifikasi akhir, maka proses instalasi Ubuntu tinggal tinggal selangkah lagi yaitu dengan menekan tombol "Install", Ubuntu akan langsung memulai proses instalasi ke dalam *hardisk*. Proses instalasi ke dalam hardisk dapat mencapai 30 menit bahkan dapat juga sampai 1 jam.

Setelah proses instalasi selesai, akan muncul jendela konfirmasi bahwa proses instalasi telah selesai seperti ditunjukkan pada Gambar 3.10. Dengan selesainya proses ini, maka proses instalasi telah selesai. Untuk masuk dan melihat hasil instalasi tinggal menekan tombol "Restart now".



Gambar 3.10 Proses instalasi telah selesai.

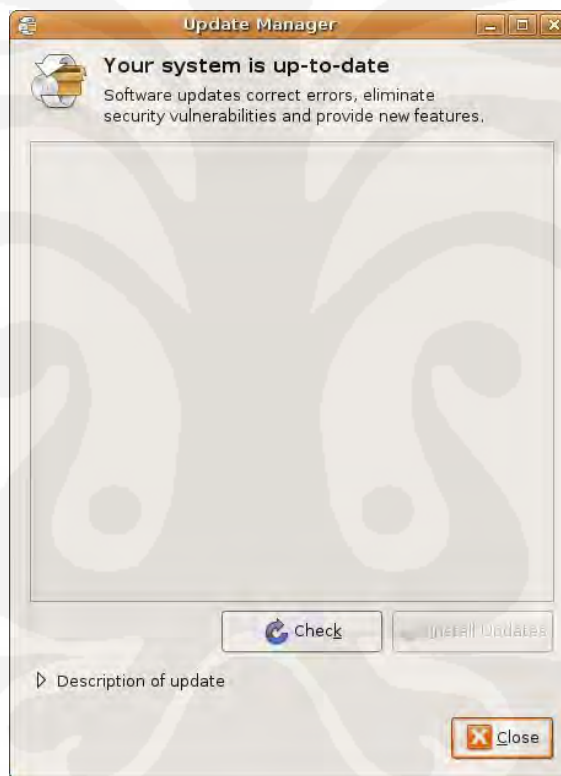
Proses restart akan berjalan dan kemudian keluarkan CD instalasi Ubuntu. Setelah itu jalankan Ubuntu Linux.

3.2.3. Proses Update Ubuntu

Setelah Ubuntu berhasil terinstal pada komputer, maka langkah selanjutnya adalah untuk meng-*update* sistem operasi tersebut. Tujuan update Ubuntu adalah:

- Agar sistem tersebut menggunakan aplikasi perangkat lunak terkini
- Memperbaiki berbagai *bugs* yang muncul
- *Update* berbagai *patch* aplikasi
- *Security Update*, *update* yang berkenaan dengan keamanan sistem Ubuntu

Untuk meng-*update* Ubuntu Linux, tinggal masuk ke menu Ubuntu yaitu System → Administration → Update Manager. Tampilan awal dari *Update Manager* seperti ditunjukkan pada Gambar 3.11.



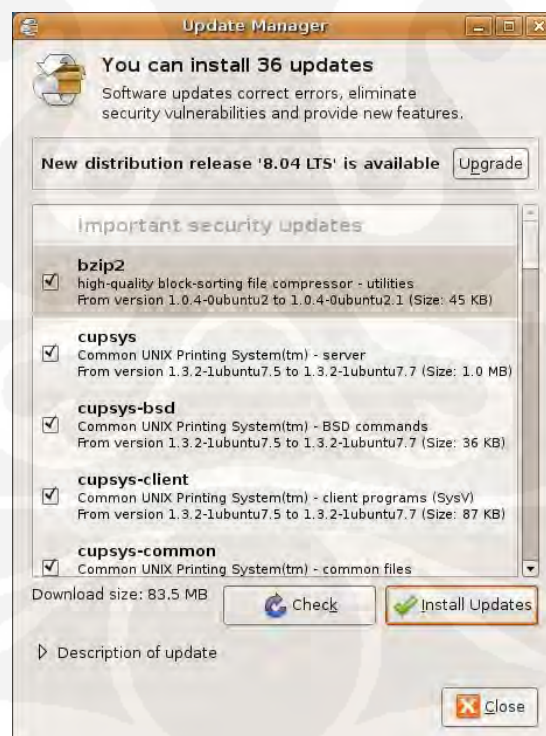
Gambar 3.11 Tampilan awal *Update Manager*.

Untuk melakukan update, langkah yang dibutuhkan adalah tinggal mengklik "*Check*". Untuk melakukan update, komputer harus terhubung dengan internet. Komputer kemudian akan melakukan update aplikasi melalui internet. Proses mendownload informasi terbaru aplikasi dapat dilihat pada Gambar 3.12.



Gambar 3.12 Proses download *update* Ubuntu.

Jika terdapat update pada aplikasi Ubuntu, maka keluaran dari proses update tersebut akan menjadi seperti Gambar 3.13 berikut.



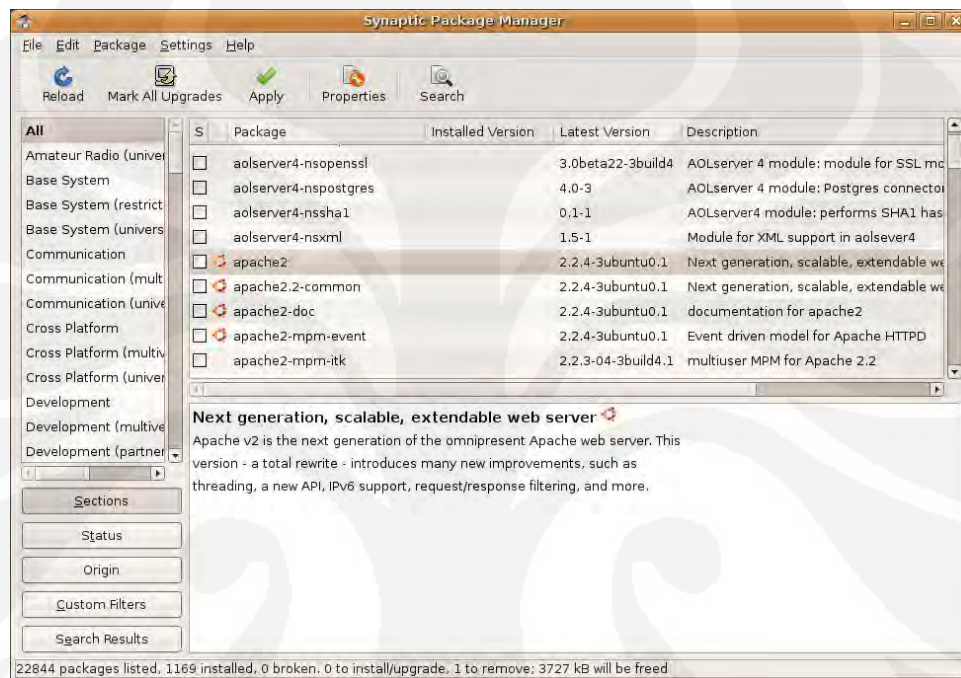
Gambar 3.13 Aplikasi yang memiliki *update* dari Ubuntu.

Setelah mengetahui berbagai update yang didapat, maka langkah selanjutnya adalah tinggal klik "*Install Updates*". Maka komputer akan mendownload aplikasi-aplikasi terbaru

dari Ubuntu melalui internet. Waktu proses *mendonwload* bergantung dari koneksi internet yang digunakan.

3.3. INSTALASI APACHE WEB SERVER, PHP, MYSQL SERVER DAN KANNEL

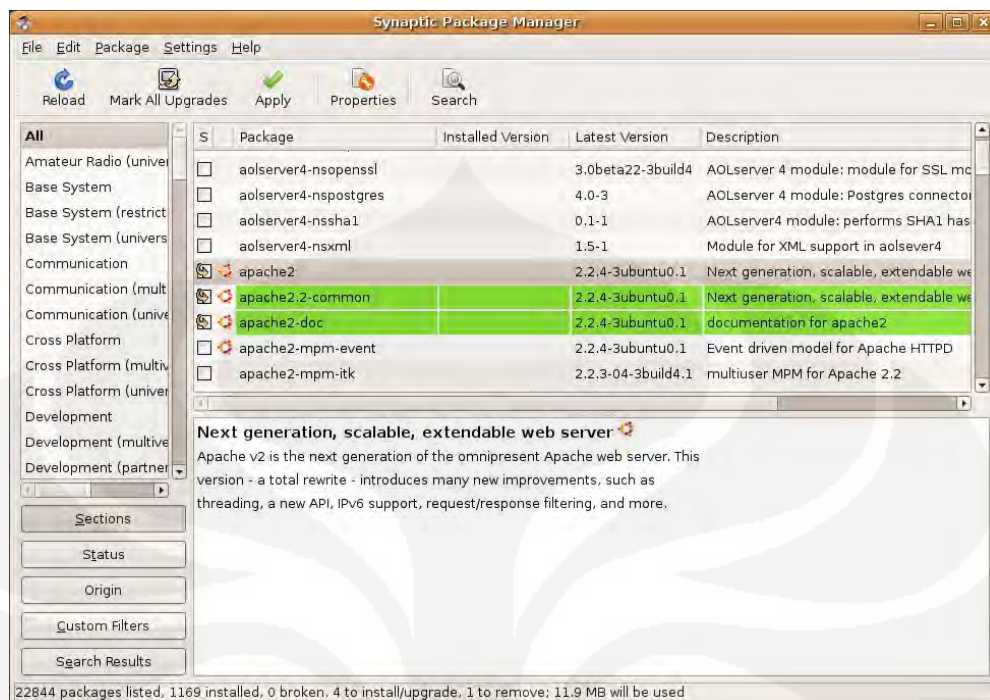
Setelah selesai menginstal Ubuntu Linux dan melakukan update, langkah selanjutnya adalah menginstal Apache Web Server. Untuk menginstalnya, masuk ke menu Ubuntu yaitu System → Administration → Synaptic Package Manager. Tampilan awal dari *Synaptic Package Manager* ditampilkan pada Gambar 3.14.



Gambar 3.14 Synaptic Package Manager.

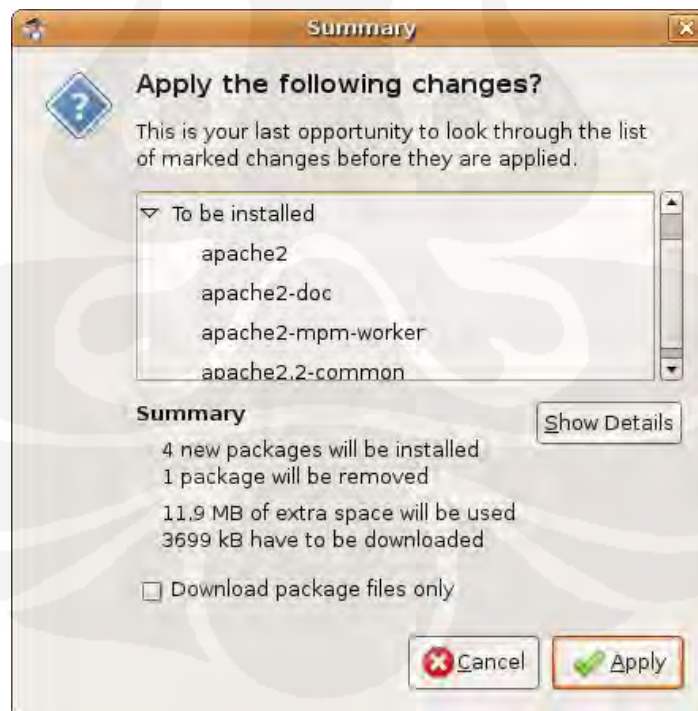
Untuk menginstal Apache Web Server, tinggal mengklik modul pada paket-paket yang dipilih. Paket-paket yang dipilih untuk instalasi Apache adalah `apache2`, `apache2.2-common`, `apache2-mpm-prefork`, `apache2-utils`, `apachetop`, dan pilihan *optional* yaitu `apache2-doc` untuk dokumentasi Apache. Untuk menginstal MySQL[28] Server untuk database, paket-paket yang dipilih adalah `mysql-server`, `mysql-client-5.0`, `mysql-common`, `mysql-query-browser`, `mysql-query-browser-common`, `mysql-admin`, `mysql-admin-common`, dan `libapache2-mod-auth-mysql`. Langkah selanjutnya adalah menginstal PHP[29] untuk bahasa pemrograman web. Paket-paket yang dibutuhkan adalah `php5-common`, `php5`, `libapache2-mod-php5`, dan `php5-mysql`. Kesemuanya wajib di instal agar *web server* berfungsi sebagaimana mestinya. Satu lagi tambahan paket yang dibutuhkan adalah Kannel, untuk *interface driver* antara PC dan *GSM Modem*. Oleh karena itu, perlu ditambahkan lagi satu paket `kannel` untuk di pilih pada *Synaptic Package Manager*.

Hasilnya dapat dilihat pada Gambar 3.15.



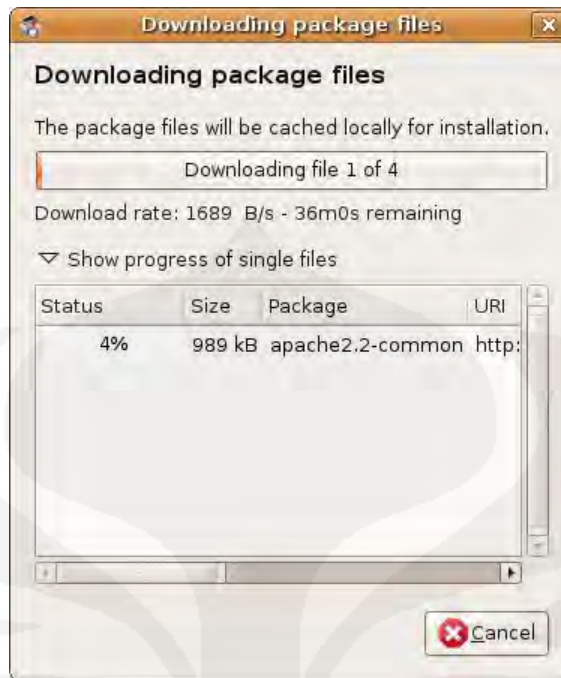
Gambar 3.15 Memilih paket Apache Web Server.

Setelah memilih paket tersebut, kini tinggal mengeksekusi instalasi tersebut. Untuk itu, tinggal mengklik tombol "Apply" pada *Synaptic Package Manager*. Setelah menekan tombol "Apply", maka akan muncul jendela berikut sebagai konfirmasi seperti Gambar 3.16.



Gambar 3.16 Konfirmasi instalasi Apache Web Server.

Untuk konfirmasi instalasi, klik kembali tombol "Apply", maka proses instalasi akan berlanjut dengan proses *download* aplikasi seperti pada Gambar 3.17.



Gambar 3.17 Proses *download* paket aplikasi.

Setelah semua paket *terdownload*, maka *Package Manager* akan meminta *Restart* komputer agar dapat digunakan.

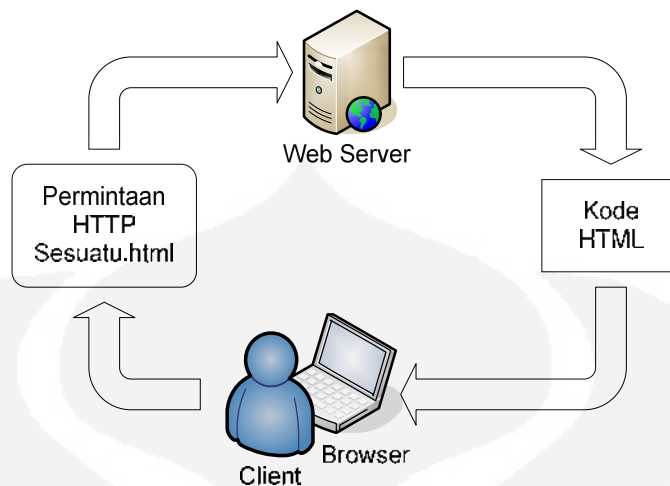
3.4. PERANGKAT LUNAK SISTEM

3.4.1. Konsep Kerja PHP

Model kerja HTML diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*.

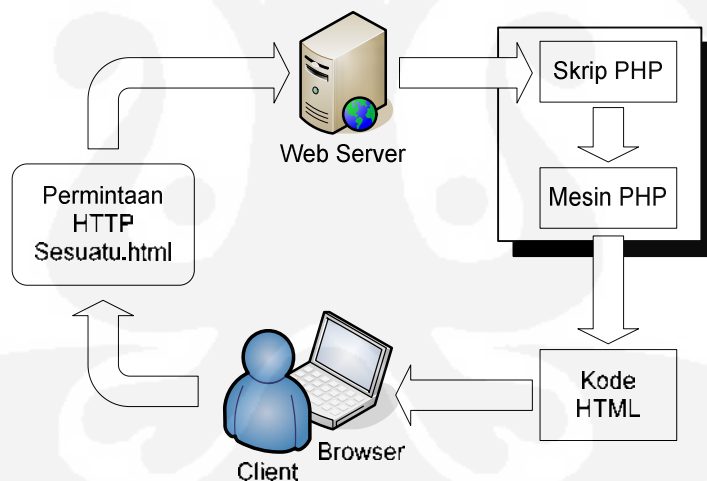
Selanjutnya, *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkannya ke layar pemakai.

Skema HTML digambarkan oleh Gambar 3.18.



Gambar 3.18 Skema HTML.

Untuk kode PHP, pada prinsipnya hampir sama dengan HTML. Hanya saja, ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang akan memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya *web server* akan memberikan hasilnya pada klien. Skemanya seperti ditunjukkan pada Gambar 3.19.



Gambar 3.19 Skema PHP.

3.4.2. Bahasa Pemrograman PHP

PHP adalah *PHP Hypertext Processor*, merupakan bahasa berbentuk skrip yang ditempatkan pada *server* dan diproses oleh *server*. Hasil dari proses tersebut yang akan dikirimkan kepada klien dimana klien menggunakan perambah (*internet browser*).

Skrip PHP berkedudukan sebagai *tag* dalam bahasa HTML. Skrip paling sederhana dari HTML adalah:

```
<html>
```

```

<head>
<title>Tugas Akhir</title>
</head>
<body>
Selamat mengerjakan Tugas Akhir.
</body>
</html>

```

Kode diatas disimpan dengan nama file `tugas-akhir.html` dan disimpan didalam folder `/var/www/`. Dengan mengakses URL dengan alamat `http://localhost/tugas-akhir.html` yang dijalankan dalam *browser*, maka hasilnya akan seperti berikut Gambar 3.20 berikut.



Gambar 3.20 Hasil kode HTML tampak pada *browser*.

Kode PHP dapat dimasukkan ke dalam kode HTML. Misalnya seperti dalam kode berikut:

```

<html>
<head>
<title>Tugas Akhir</title>
</head>
<body>
Selamat mengerjakan Tugas Akhir.<br>
<?php
    printf("Tgl. Sekarang: %s ", Date("d F Y"));
?>
</body>
</html>

```

Kode tersebut disimpan dengan nama `tugas-akhir.php`. Perhatikan baris-baris berikut:

```
<?php
    printf("Tgl. Sekarang: %s ", Date("d F Y"));
?>
```

Kode inilah yang disebut dengan kode PHP. Kode PHP diawali dengan `<?php` dan diakhiri dengan `?>`. Pasangan kedua kode inilah yang berfungsi sebagai *tag* kode PHP. Berdasarkan *tag* inilah, *server* akan memahami kode PHP dan kemudian memprosesnya. Hasilnya akan dikirimkan ke *browser*.

Jika kode tersebut dijalankan dalam *browser*, maka hasilnya akan tampak seperti Gambar 3.21 berikut.



Gambar 3.21 Hasil kode PHP tampak pada *browser*.

3.4.2.1. Variabel pada PHP

Semua bahasa pemrograman mengenal variabel, yang berfungsi untuk menyimpan suatu nilai dan nilai yang ada didalamnya dapat diubah sewaktu-waktu. Skrip berikut untuk memberikan contoh penggunaan variabel:

```
<HTML>
<HEAD>
<TITLE>Latihan Variabel</TITLE>
</HEAD>
<BODY>
<?php
    $nama = "Iqbal Irwansyah";
    printf("Halo, %s", $nama);
?>
</BODY>
</HTML>
```

File tersebut disimpan dengan nama **latihan-variabel.php** dan disimpan didalam direktori `/var/www/`.

Pada kode tersebut, nama adalah variabel. Pada program PHP, variabel selalu ditulis dengan diawali dengan tanda \$ seperti baris pada contoh tersebut:

```
$nama = "Iqbal Irwansyah";
```

Merupakan pernyataan yang digunakan untuk memberikan string "Iqbal Irwansyah" ke variabel nama. Selanjutnya,

```
printf("Halo, %s", $nama);
```

Merupakan pernyataan untuk menampilkan tulisan 'Halo', diikuti dengan nilai yang terletak disebelah kanan koma, yaitu isi variabel nama. Tanda %s merupakan tanda untuk string (artinya, data yang akan mensubstitusikan tanda tersebut adalah data string).

Pada saat skrip tersebut dipanggil melalui *browser*, maka akan mendapatkan hasil seperti Gambar 3.22 berikut:



Gambar 3.22 Hasil skrip latihan-variabel.php tampak pada *browser*

Dengan menggunakan PHP, dapat juga mengirimkan suatu nilai yang ditangani melalui formulir (dimasukkan oleh pemakai melalui *keyboard*) dan kemudian nilai ini dikirim ke skrip yang lain. Seperti pada skrip berikut:

```
<HTML>
<HEAD>
<TITLE>Latihan Pemasukan Data</TITLE>
</HEAD>
<FORM ACTION=salam.php METHOD=get>
  Silakan Masukkan nama Anda:<BR>
  <INPUT TYPE=TEXT NAME=nama_pemakai><BR>
  <INPUT TYPE=SUBMIT Value="Kirim">
</FORM>
</BODY>
</HTML>
```


File tersebut disimpan dengan nama **latihan-input.php** dan disimpan dalam folder `/var/www/`. Pada kode tersebut tidak terdapat kode PHP sama sekali, kode tersebut murni kode HTML. Namun tidak menjadi masalah untuk disimpan dalam file dengan ekstensi `.php`.

Pada saat skrip tersebut dipanggil melalui *browser*, maka akan mendapatkan hasil seperti Gambar 3.23 berikut:



Gambar 3.23 Hasil skrip latihan-input.php tampak pada *browser*.

Pada saat pengguna memasukkan nama, maka nama tersebut akan disimpan pada variabel `nama_pemakai`. Pada kode tersebut, `nama_pemakai` tidak diawali dengan `$`, sebab variabel tersebut tidak berada dalam kode PHP, tetapi merupakan variabel yang berada dalam kode HTML.

Selanjutnya ketika pengguna mengklik tombol bernama "*Kirim*", maka file tersebut akan menjalankan file **salam.php**. Berikut adalah file **salam.php**:

```
<HTML>
<HEAD>
<TITLE>Latihan Menampilkan Variabel</TITLE>
</HEAD>
<BODY>
  <?php
    $nama_pemakai=$_GET['nama_pemakai'];
    print("Selamat belajar PHP, <B>$nama_pemakai</B>");
  ?>
</BODY>
</HTML>
```

Pernyataan yang digunakan untuk menampilkan nilai bukanlah `printf` melainkan `print`. `Printf` biasa digunakan kalau ada format (seperti `%s`) yang disertakan dalam string. Adapun `print` biasa digunakan kalau tidak ada format.

Setelah selesai membuat file **salam.php**, kini saatnya untuk menguji skrip input tersebut dengan membuka file **latihan-input.php** melalui *browser* dan menginput sebuah nama pada formulir tersebut seperti pada Gambar 3.24.



Gambar 3.24 Mengisi input pada skrip latihan-input.php.

Ketika pengguna menginput sebuah nama pada file **latihan-input.php** dan kemudian mengklik "Kirim" maka akan mengirimkan sebuah variabel dengan string `nama_pemakai` melalui metode GET pada sebuah file bernama **salam.php**. Maka ketika selesai mengklik file **latihan-input.php** maka selanjutnya akan menampilkan file **salam.php**. Tampilan hasil dari **salam.php** akan menjadi seperti Gambar 3.25 berikut:



Gambar 3.25 Hasil skrip salam.php tampak pada *browser*.

3.4.2.2. Mengakses database MySQL dengan PHP

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu, ia bersifat *free* (tanpa bayaran untuk menggunakannya) dan dapat berjalan pada berbagai *platform*.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Fungsi MySQL pada PHP diawali dengan `mysql_` yang digunakan untuk mengakses database MySQL.

- Fungsi `mysql_connect`

Fungsi `mysql_connect` digunakan untuk membuat hubungan ke database MySQL yang terdapat pada suatu *host*. Bentuk pemanggilannya:

```
mysql_connect (host, user_name, password)
```

Host menyatakan nama *host* tempat MySQL berada, `user_name` menyatakan nama pemakai yang digunakan untuk melakukan hubungan dan `password` adalah *password* milik pemakai.

Fungsi ini menghasilkan nilai balik berupa nilai bertipe integer yang menyatakan hubungan dan digunakan sebagai pengenalan pada berbagai fungsi yang berawalan `mysql_`. Bila terjadi suatu kesalahan, nilai balik akan berupa FALSE. Seperti pada skrip berikut:

```
$id_mysql = mysql_connect ("localhost", "php",  
"penguin");
```

Skrip diatas merupakan perintah untuk membentuk hubungan ke database MySQL yang terdapat pada "localhost", dengan pemakai berupa "php" dan password berupa "penguin". Pengenal hubungan akan disimpan pada variabel `id_mysql`.

- Fungsi `mysql_close`

Fungsi `mysql_close` digunakan untuk menutup hubungan ke database MySQL. Bentuk pemanggilannya:

```
mysql_close (pengenal_hubungan)
```

Dalam hal ini `pengenal_hubungan` menyatakan pengenalan yang diperoleh dari pemanggilan fungsi `mysql_connect`. Ketika skrip PHP berakhir, hubungan ke database MySQL yang dibentuk melalui `mysql_connect` juga dengan sendirinya akan ditutup.

Skrip php berikut digunakan untuk menguji koneksi ke database MySQL.

```
<HTML>  
<HEAD>  
<TITLE>Tes MySQL</TITLE>  
</HEAD>  
<BODY>  
<?php  
    $user_name = "php";  
    $password = "penguin";
```

```

$id_mysql = mysql_connect("localhost",
                           $user_name,
                           $password);

if (! $id_mysql)
    die("Database MySQL tak dapat dibuka");
mysql_close($id_mysql);
print("Sukses");
?>
</BODY>
</HTML>

```

- Fungsi `mysql_select_db`

Fungsi `mysql_select_db` digunakan untuk memilih database MySQL. Bentuk pemanggilannya:

```
mysql_select_db (database, pengenal_hubungan)
```

Dalam hal ini `pengenal_hubungan` menyatakan pengenal yang diperoleh dari pemanggilan fungsi `mysql_connect` dan argumen database adalah nama database yang dipilih.

Skrip berikut menggunakan `mysql_select_db`:

```
mysql_select_db ("buku_alamat", $id_mysql);
```

- Fungsi `mysql_query`

Fungsi `mysql_query` digunakan untuk mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel. Bentuk pemanggilannya:

```
mysql_query (permintaan, pengenal_hubungan)
```

Dalam hal ini `pengenal_hubungan` menyatakan pengenal yang diperoleh dari pemanggilan fungsi `mysql_connect` dan argumen permintaan berupa pernyataan SQL untuk mengakses tabel. Tabel-tabel yang dapat diakses adalah tabel milik database yang disebutkan melalui `mysql_select_db`.

Fungsi ini menghasilkan suatu nilai yang digunakan sebagai pengenal hasil bagi sejumlah fungsi lainnya yang berawalan dengan `mysql_`. Bisa terjadi suatu kesalahan, hasilnya berupa `FALSE`.

- Fungsi `mysql_db_query`

Fungsi `mysql_db_query` digunakan untuk menjalankan suatu permintaan terhadap suatu database. Bentuk pemanggilannya:

```
mysql_db_query (database, permintaan, pengenalan_hubungan)
```

Dalam hal ini, `pengenal_hubungan` menyatakan pengenalan yang diperoleh dari pemanggilan fungsi `mysql_connect` dan argumen database adalah nama database yang dipilih. Adapun permintaan berupa pernyataan SQL. Perlu diketahui argumen ketiga bisa tidak ditulis. Bila tidak ditulis, maka dengan sendirinya akan dihubungkan ke hubungan MySQL yang sudah terbentuk atau akan melakukan hubungan seperti kalau memanggil `mysql_connect`.

Fungsi ini akan memberikan nilai balik berupa suatu angka bulat yang menyatakan pengenalan hasil, yang diperlukan pada berbagai fungsi berawalan `mysql_` lainnya. Hasil FALSE menyatakan bahwa terdapat suatu kesalahan.

```
mysql_select_db ("buku_alamat", $id_mysql);
```

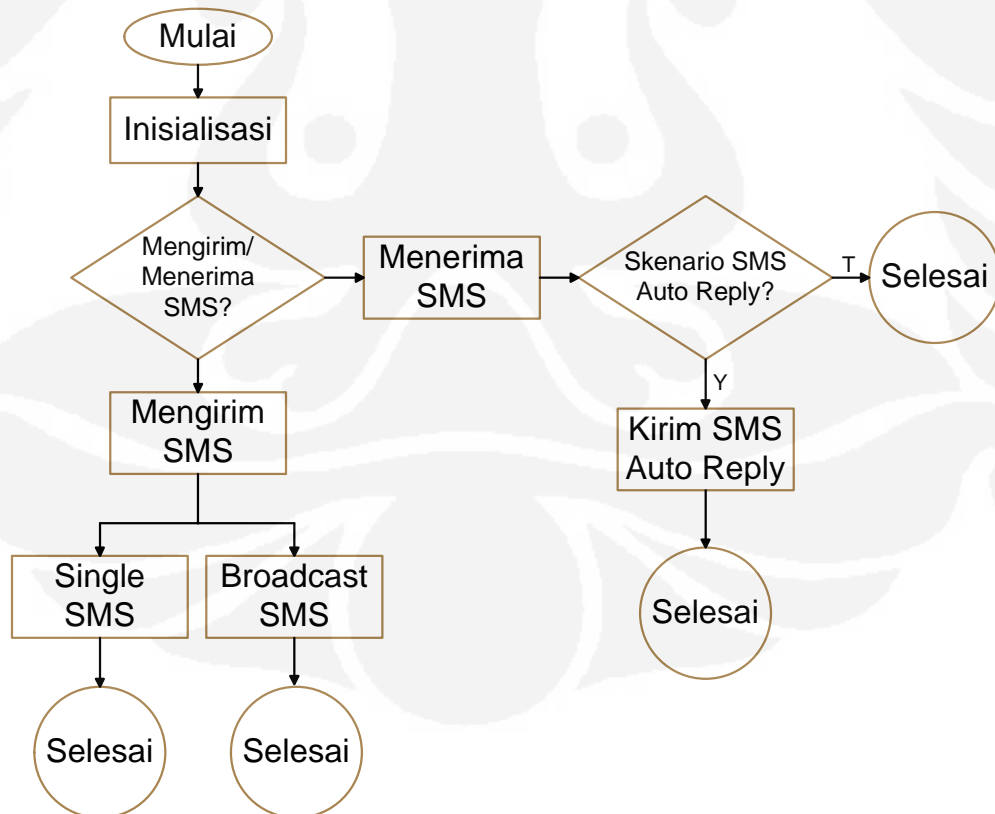
```
mysql_query ("SELECT * FROM kota", $id_mysql);
```

Pernyataan tersebut dapat diganti dengan:

```
mysql_query_db ("buku_alamat", "SELECT * FROM kota",  
$id_mysql);
```

3.4.2.3. Flowchart pemrograman

Flowchart pemrograman perangkat lunak keseluruhan dapat dilihat pada Gambar 3.26.



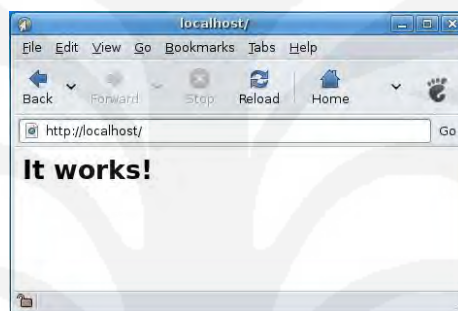
Gambar 3.26 Flowchart pemrograman perangkat lunak.

BAB 4

UJICOBA SISTEM DAN ANALISA

4.1. PENGUJIAN APACHE WEB SERVER

Setelah selesai menginstal semua aplikasi yang dibutuhkan, maka selanjutnya adalah menguji hasil instalasi. Untuk menguji Apache Web Server, caranya adalah dengan membuka perambah (*browser*) kemudian mengetikkan alamat **http://localhost/** pada alamat URL yang dituju. Jika berhasil, maka akan menghasilkan keluaran seperti Gambar 4.1 berikut:



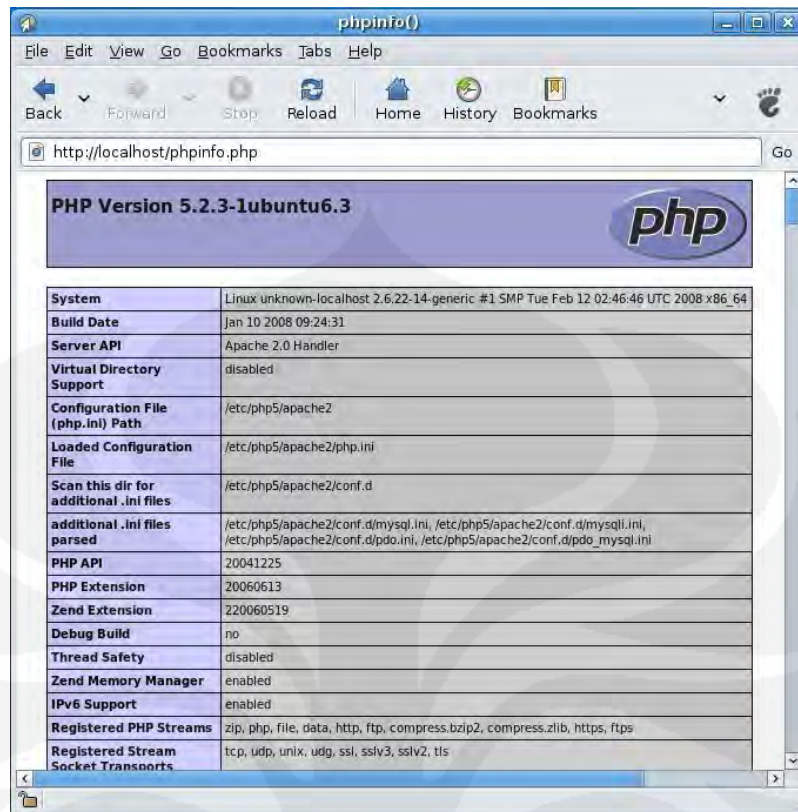
Gambar 4.1 Uji coba instalasi Apache

Untuk menguji PHP, caranya adalah dengan membuat sebuah file dengan nama misalnya **phpinfo.php** yang berisi baris berikut:

```
<php
Phpinfo();
?>
```

File tersebut dapat diletakkan pada folder `/var/www/`. Kemudian cara mengujinya yaitu dengan membuka perambah dan memasukkan alamat berikut **http://localhost/phpinfo.php**. Hasilnya adalah seperti pada Gambar 4.2.

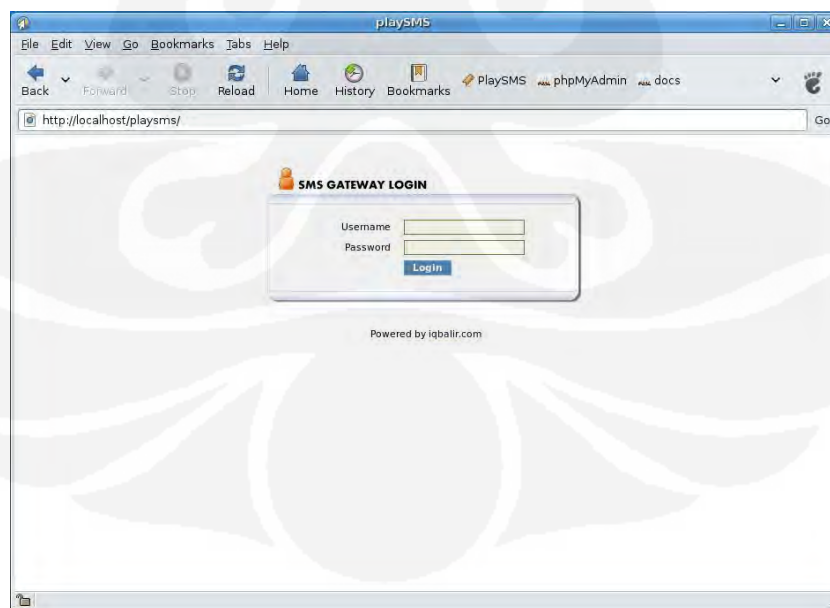
Jika Apache Web Server dan PHP telah menunjukkan hasil seperti pada Gambar 4.1 dan Gambar 4.2 menunjukkan bahwa instalasi Apache Web Server dan PHP telah berhasil.



Gambar 4.2 Uji coba instalasi PHP

4.2. PENGUJIAN LOGIN SMS GATEWAY

Untuk menguji *SMS Gateway*, caranya adalah dengan membuka perambah (*browser*) kemudian mengetikkan alamat **http://localhost/playsms/** pada *browser*. Jika berhasil, maka akan menghasilkan keluaran berisi halaman login seperti Gambar 4.3 berikut:

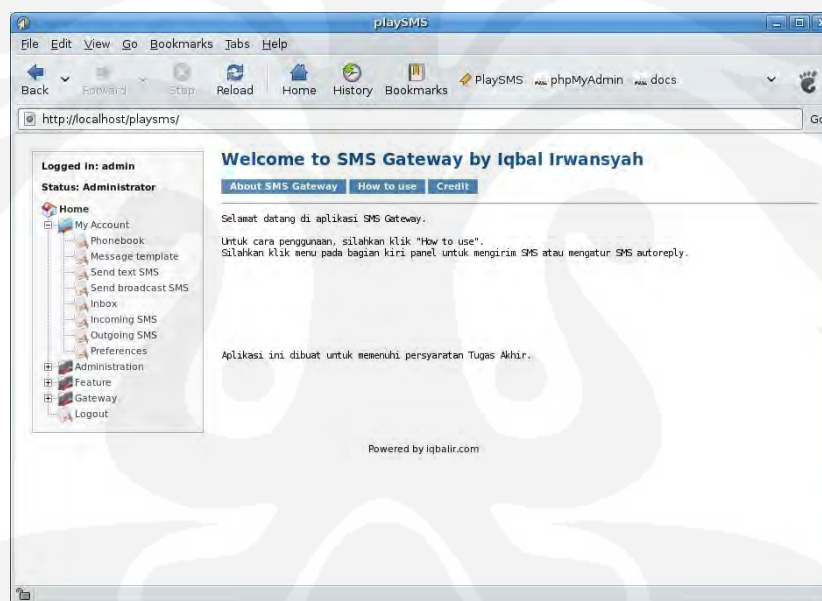


Gambar 4.3 Halaman login SMS Gateway

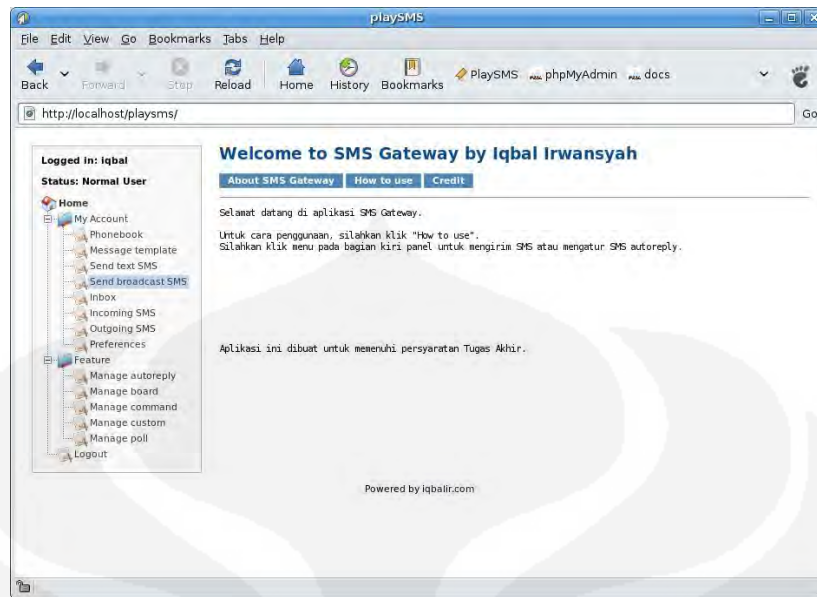
Pada Gambar 4.3 ditunjukkan halaman login *SMS Gateway*. Dengan mengetikkan *username* dan *password*, maka administrator atau pengguna dapat masuk ke dalam aplikasi *SMS Gateway*.

Setelah mengetikkan *username* dan *password* kemudian mengklik tombol "Login" maka pengguna dapat menggunakan aplikasi *SMS Gateway* ini. Setelah sukses login, maka pengguna akan masuk dalam aplikasi *SMS Gateway*. Aplikasi ini memberikan dua buah *privileges* (kewenangan istimewa) yaitu *administrator* dan *user* (pengguna). Kewenangan *administrator* dapat melakukan konfigurasi *SMS Gateway*, mengubah konfigurasi modem, menambah/menghapus *user* dan memiliki semua kewenangan yang dimiliki oleh *user*. Kewenangan *user* yaitu hanya dapat mengirim SMS, *broadcast SMS*, memiliki *phonebook* sendiri, dapat menerima SMS, dapat mengeset aplikasi *auto-reply SMS*.

Gambar 4.4 menunjukkan halaman *administrator* dan Gambar 4.5 menunjukkan halaman *user* biasa.



Gambar 4.4 Halaman administrator *SMS Gateway*

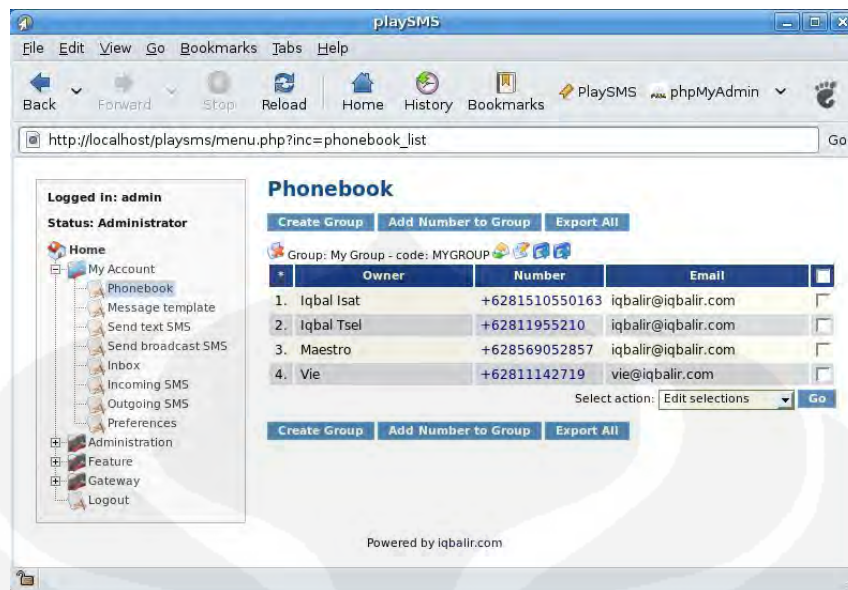


Gambar 4.5 Halaman *user SMS Gateway*

4.3. PENGUJIAN MENGISI *PHONEBOOK SMS GATEWAY*

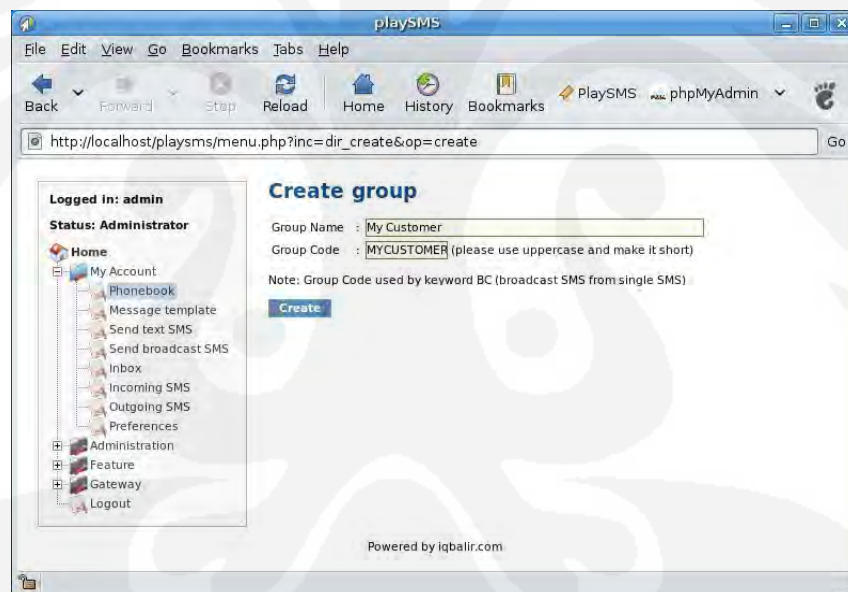
Setelah sukses melakukan login dengan memasukkan *username* dan *password*, maka saatnya untuk mencoba mengirimkan SMS dengan *SMS Gateway*. Untuk mengirimkan SMS, bisa melalui dua cara, yaitu mengirimkan SMS melalui daftar *phonebook* atau dapat mengisi langsung nomor tujuan SMS. Tentu saja *phonebook* dapat diisi sebelumnya jika ingin mengirimkan SMS melalui *phonebook*. Melalui *phonebook* juga dapat dibuat beberapa *group* untuk memudahkan mengirim *SMS broadcast* (akan diuji kemudian).

Untuk mengisi *phonebook* SMS adalah dengan masuk ke menu "Phonebook". Pada menu tersebut, pengguna dapat mengisi daftar nomor telepon untuk tujuan mengirim SMS. Pada menu tersebut juga dapat membuat *group* (pengelompokan) nomor tujuan SMS agar dapat memudahkan ketika nanti ingin mengirim *SMS broadcast*. Pada Gambar 4.6 ditampilkan halaman daftar buku telepon (*phonebook*).



Gambar 4.6 Halaman *phonebook*

Untuk membuat *group* (pengelompokkan), tinggal mengklik menu "*Create Group*". Setelah mengklik menu tersebut, akan didapat halaman seperti pada Gambar 4.7 berikut.

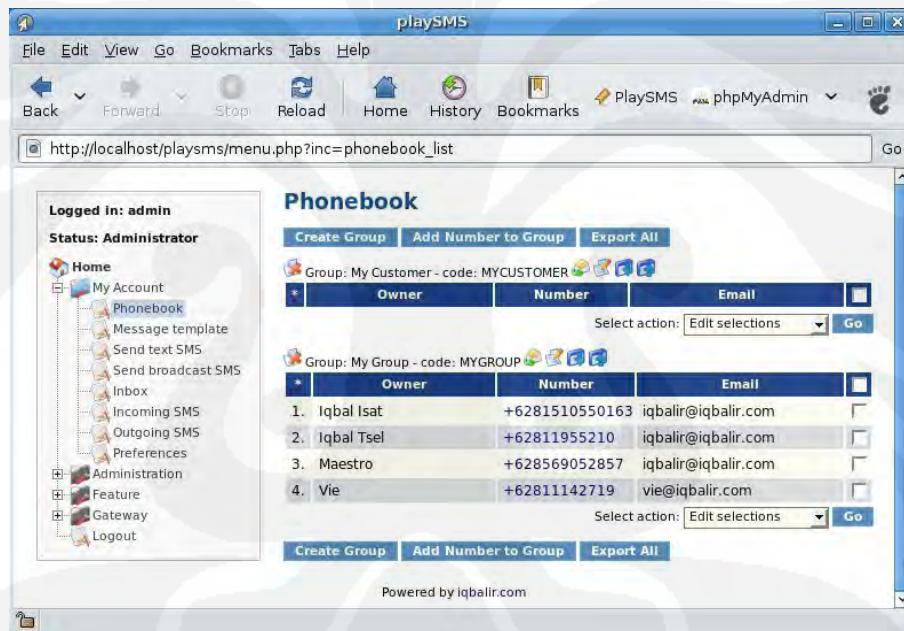


Gambar 4.7 Halaman membuat *group* baru pada *phonebook*

Setelah mengisi *Group Name* dan *Group Code*, kemudian diikuti dengan mengklik tombol "*Create*" maka *group* akan tercipta. Pada pengujian ini *Group Name* diisi dengan "*My Customer*" dan *Group Code* diisi dengan "*MYCUSTOMER*". *Group Code* akan digunakan ketika nanti akan mengirimkan SMS *broadcast*. Hasilnya adalah seperti pada Gambar 4.8 dan Gambar 4.9.



Gambar 4.8 Halaman notifikasi bahwa *group* baru sudah tercipta



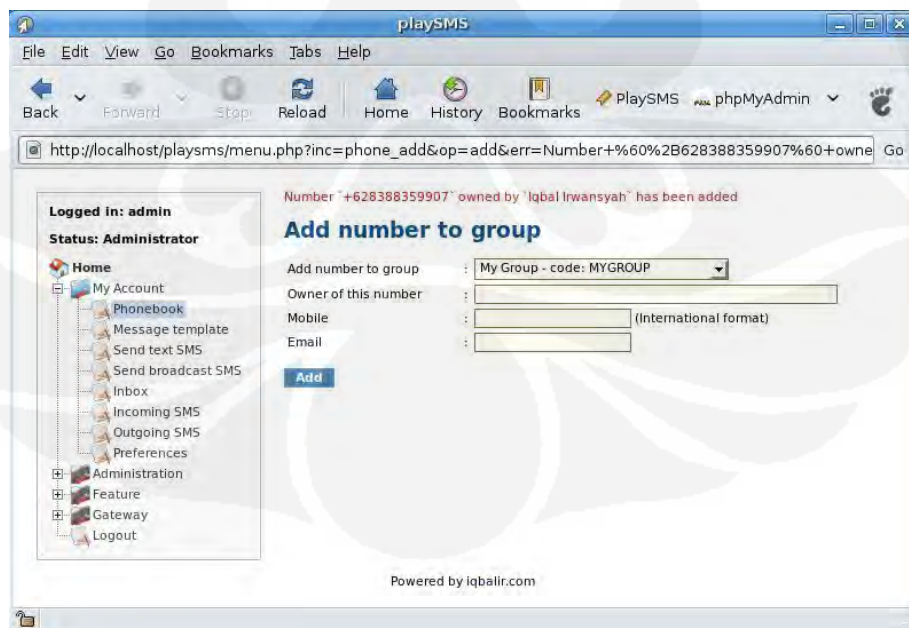
Gambar 4.9 Halaman *phonebook* menunjukkan *group* yang baru tercipta

Gambar 4.8 adalah notifikasi bahwa *group* sudah berhasil tercipta dan Gambar 4.9 adalah hasil tersebut dapat dilihat pada menu *phonebook*. Ini ditunjukkan dengan *group* "My Customer" pada Gambar 4.9. Karena *group* tersebut baru saja dibuat, maka isi dari *group* tersebut masih kosong. Untuk mengisinya dapat mengklik menu "Add Number to Group" seperti ditunjukkan pada Gambar 4.10.

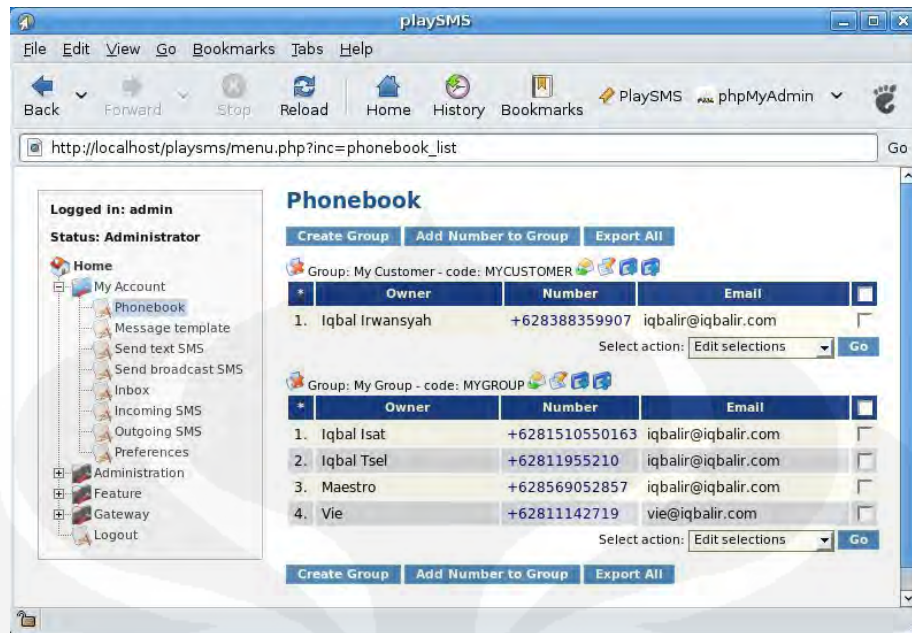


Gambar 4.10 Halaman menambah nomor pada *group phonebook*

Pada pengujian berikut, *group* yang dipilih adalah "MYCUSTOMER", ini adalah *group* yang sebelumnya dibuat. "Owner of this number" adalah nama pemilik nomor telepon, dalam hal ini dimasukan "Iqbal Irwansyah", *Mobile* adalah nomor telepon dari nama tersebut, dalam hal ini dimasukan nilai "+628388359907", *Email* adalah sebuah pilihan yang boleh diisi dan boleh juga tidak diisi. Kemudian untuk memasukkan data tersebut pada *phonebook* tinggal mengklik tombol "Add", maka akan muncul notifikasi bahwa nomor sudah dimasukan pada *phonebook* seperti pada Gambar 4.11. Hasilnya dapat dilihat pada menu *phonebook* bahwa *user* dengan nama "Iqbal Irwansyah" telah tercipta seperti pada Gambar 4.12.



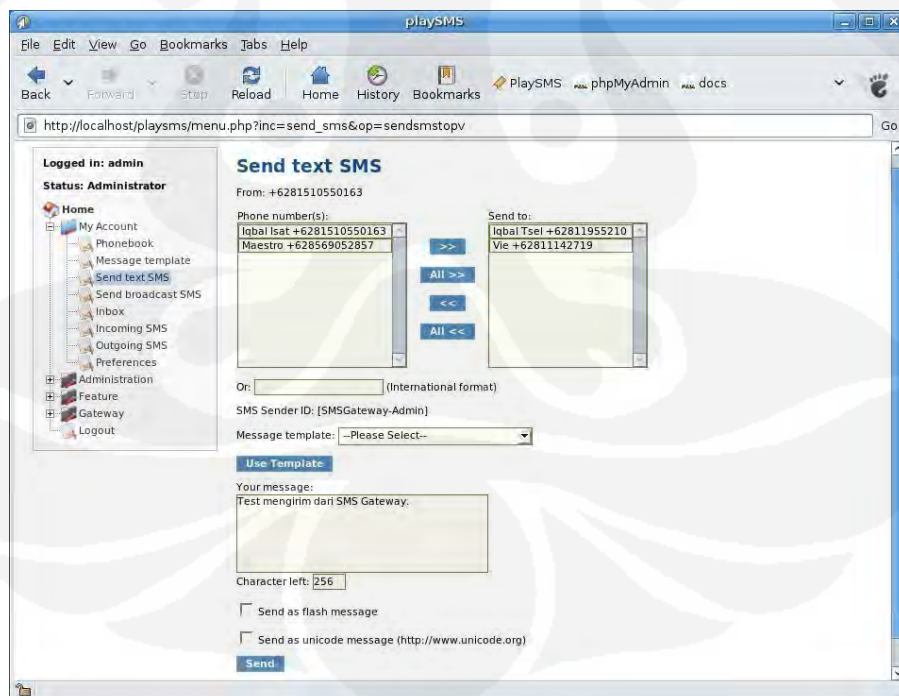
Gambar 4.11 Halaman notifikasi bahwa nomor sudah dimasukan pada *group*



Gambar 4.12 Halaman *phonebook* bahwa nomor sudah dimasukan pada *group*

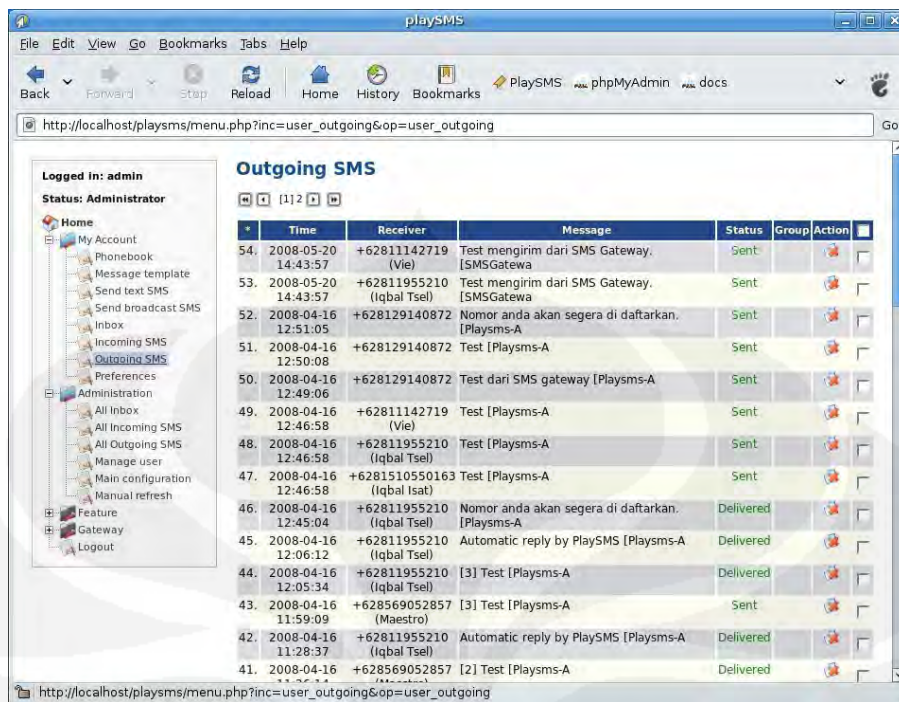
4.4. PENGUJIAN MENGIRIM *SINGLE SMS*

Untuk mengirimkan SMS, caranya adalah dengan masuk ke menu "*Send text SMS*". Pada halaman tersebut akan diminta memasukkan nomor tujuan, dapat memasukkan nomor tujuan yang sudah ada pada *phonebook* maupun dapat memasukkan nomornya secara manual. Gambar 4.13 berikut ini adalah halaman mengirim SMS pada *SMS gateway*.



Gambar 4.13 Mengirim SMS dengan *SMS Gateway*

Setelah terkirim, status hasil kirimannya dapat dilihat pada menu "*Outgoing SMS*". Hasil dari status kiriman SMS tersebut dapat dilihat seperti pada Gambar 4.14.



Gambar 4.14 Hasil kiriman SMS pada kotak keluar (*Outgoing SMS*) *SMS Gateway*

Hasil kiriman pada penerima SMS yang diterima pada *handphone* ditunjukkan pada Gambar 4.15 berikut.



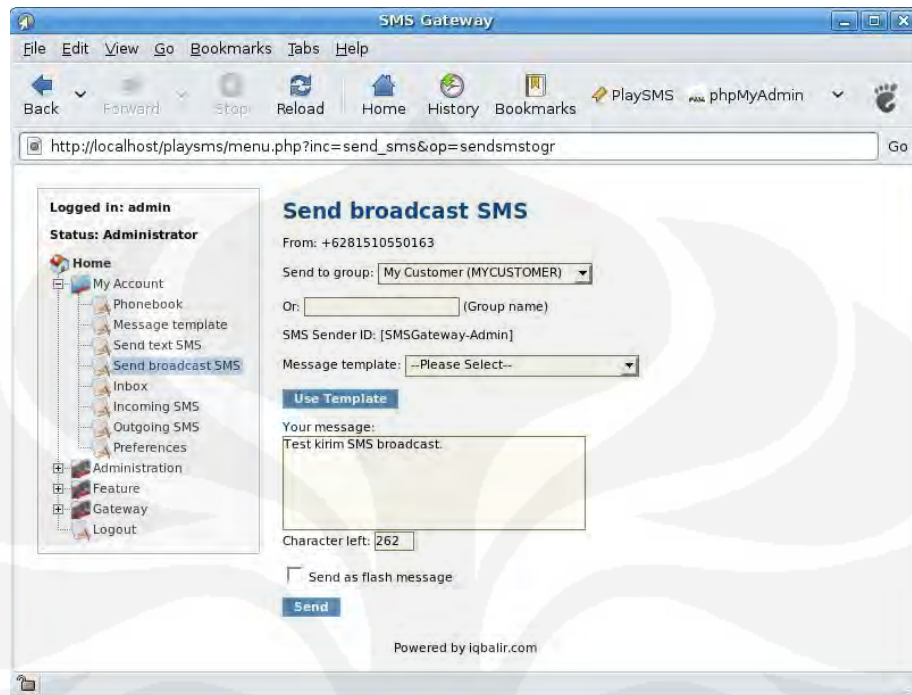
Gambar 4.15 Hasil kiriman SMS pada penerima SMS

4.5. PENGUJIAN MENKIRIM *BROADCAST SMS*

Untuk menguji SMS *broadcast*, caranya hampir sama dengan mengirimkan SMS biasa, hanya saja daftar tujuan yang dituju adalah berupa *group*. Jadi semua anggota-anggota *group* tersebut akan menerima SMS yang dikirimkan. Untuk mengujinya, masuk ke menu "*Send broadcast SMS*". Kemudian pilih *group* yang akan dikirimkan SMS *broadcast*, setelah itu mengisi pesan yang akan dikirimkan.

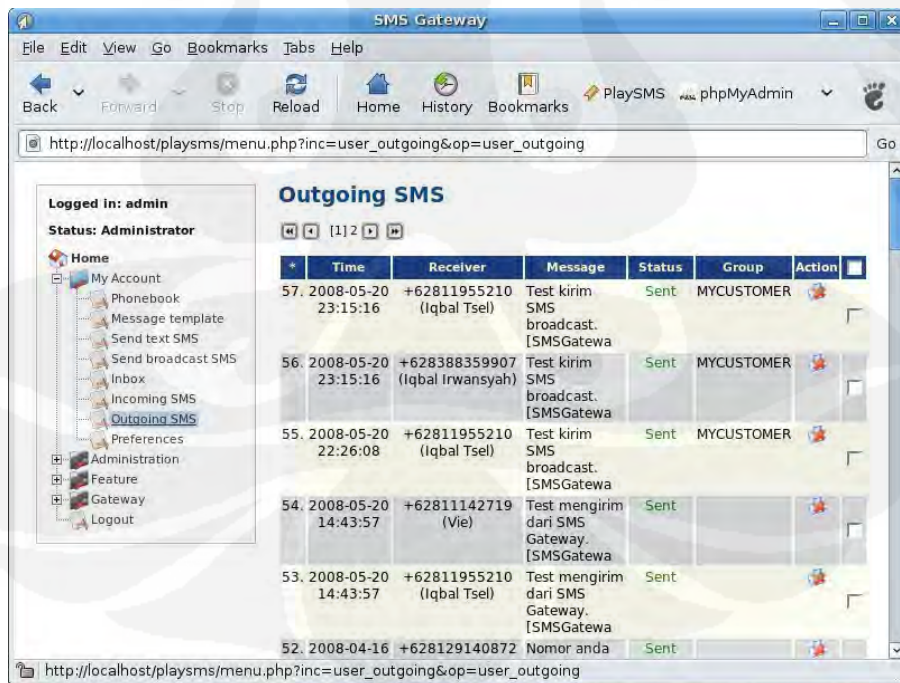
Pengujian kali ini dengan mengirimkan SMS pada *group* "MYCUSTOMER" karena itu pada "Sent to group" dipilih "MYCUSTOMER". Dilanjutkan dengan mengisi pesan yang akan dikirimkan. Setelah itu dilanjutkan dengan mengklik tombol "*Send*" seperti ditunjukkan

pada Gambar 4.16. Semua anggota yang masuk ke dalam *group* "MYCUSTOMER" akan mendapatkan SMS *broadcast* tersebut.



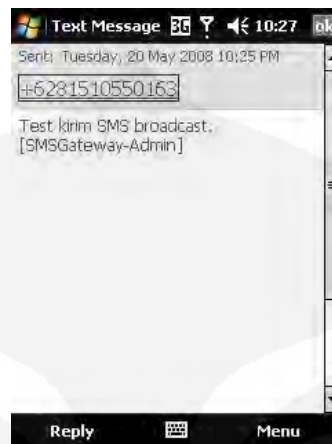
Gambar 4.16 Halaman SMS *broadcast*

Setelah terkirim, status hasil kirimannya dapat dilihat pada menu "Outgoing SMS". Hasil dari status kiriman SMS tersebut dapat dilihat seperti pada Gambar 4.17.



Gambar 4.17 Hasil kiriman SMS *broadcast* pada SMS Gateway

Hasil kiriman pada penerima SMS *broadcast* yang diterima pada *handphone* ditunjukkan pada Gambar 4.18 berikut.



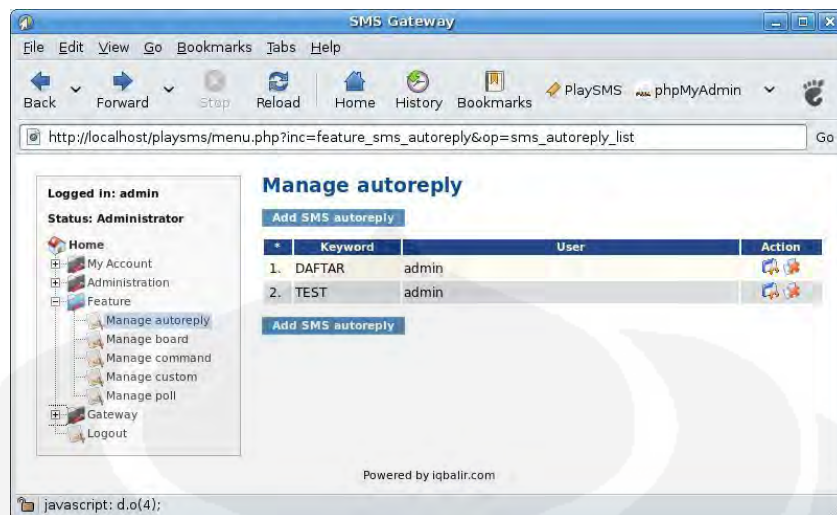
Gambar 4.18 Hasil kiriman SMS broadcast pada penerima SMS.

4.6. PENGUJIAN AUTO-REPLY SMS

Untuk menguji SMS *Auto-Reply*, caranya dilakukan dengan membuat skema *auto-reply* ketika pengguna mengirimkan sebuah kata tertentu kepada nomor *SMS Gateway*. Jadi ketika pengguna mengirimkan sebuah perintah tertentu melalui SMS, maka aplikasi *SMS Gateway* ini akan menjawab SMS tersebut dengan SMS yang skemanya telah diset sebelumnya. Jadi untuk dapat mengaplikasikan fungsi ini, maka perlu di set skemanya terlebih dahulu.

4.6.1. Pembuatan Skema Auto Reply

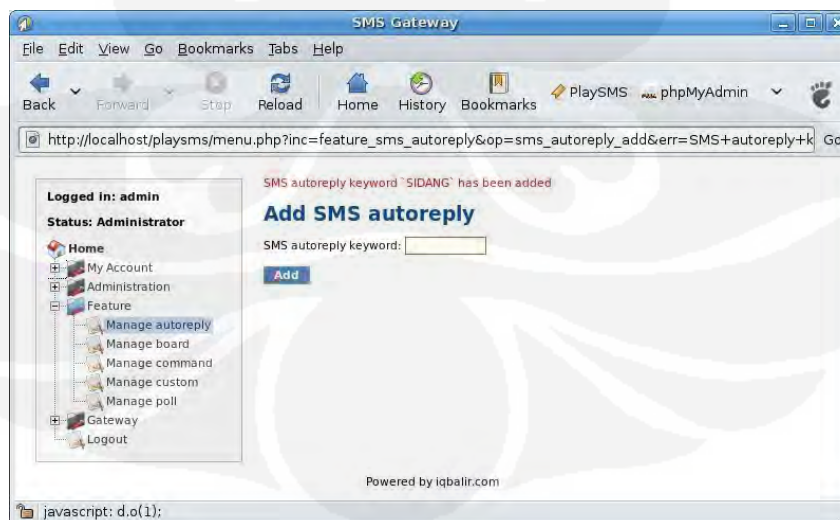
Untuk membuat skema SMS *auto reply*, tinggal mengklik menu "Manage autoreply". Pada menu tersebut, dapat melakukan konfigurasi SMS *auto reply*. Tampilan menu tersebut seperti digambarkan pada Gambar 4.19. Tahap awal dari pembuatan nama skema SMS *auto reply* adalah dengan mengklik tombol "Add SMS autoreply". Kemudian akan muncul halaman seperti pada Gambar 4.20. Dalam pengujian ini, dimasukkan *keyword* "SIDANG" pada nama skema. Setelah itu masukkan nama skema untuk SMS *auto reply* dan kemudian menekan tombol "Add". Ketika tombol tersebut telah di klik, maka akan muncul notifikasi seperti pada Gambar 4.21.



Gambar 4.19 Tampilan halaman SMS *auto reply*



Gambar 4.20 Menambahkan nama skema SMS *auto reply*



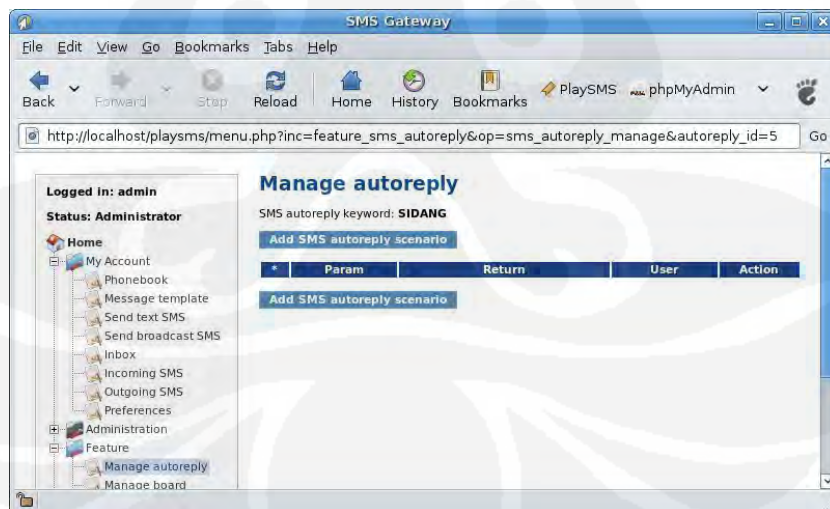
Gambar 4.21 Notifikasi nama skema SMS *auto reply* telah berhasil dibuat

Setelah nama skema telah berhasil dibuat, maka kembali ke menu *"Manage autoreply"*. Gambar 4.22 menunjukkan skema yang baru dibuat tersebut. Kemudian setelah berhasil membuat nama skema *"SIDANG"* pada SMS autoreply, kini saatnya untuk menambahkan skema *auto reply* pada skema tersebut. Caranya dengan mengklik *"Edit"* pada menu *"Action"* dibagian kanan menu tersebut.



Gambar 4.22 Nama skema SMS *auto reply* telah berhasil dibuat

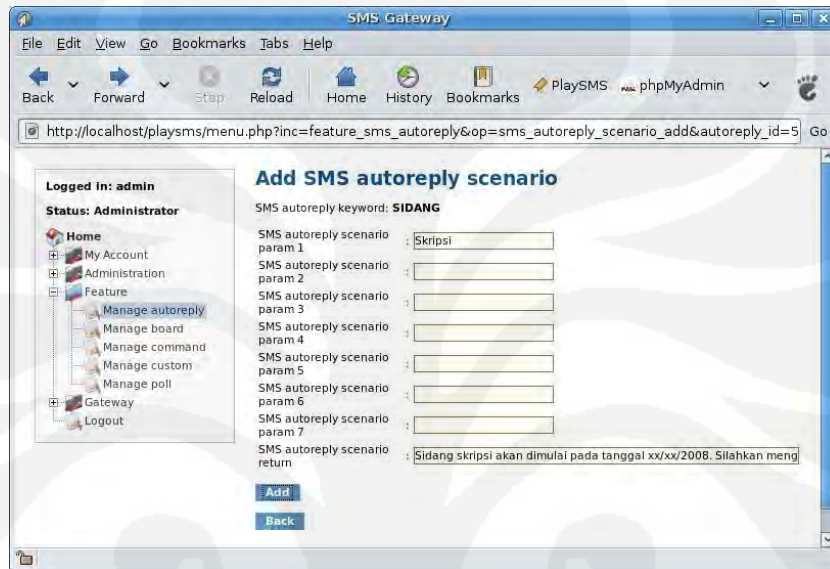
Setelah menu *"Edit"* diklik, maka akan muncul halaman seperti pada Gambar 4.23 dibawah ini. Setelah masuk ke halaman tersebut, kemudian klik *"Add SMS autoreply scenario"*. Fungsinya untuk membuat skenario auto reply, disini lah perintah SMS tertentu dari pengguna dapat didefinisikan.



Gambar 4.23 Halaman skenario SMS *auto reply*

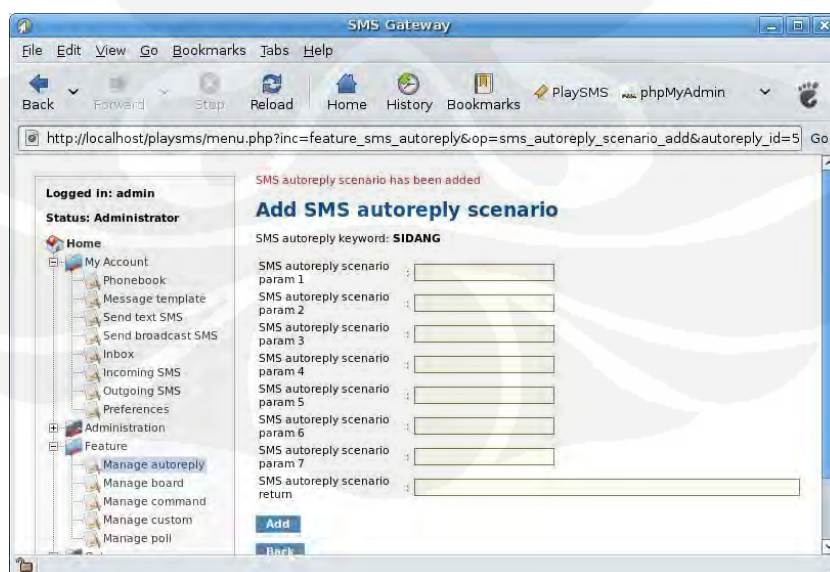
Setelah mengklik *"Add SMS autoreply scenario"*, maka akan muncul halaman seperti pada Gambar 4.24. Pada skenario *"SIDANG"* ini, diisikan skenario *"Skripsi"* pada parameter *"SMS autoreply scenario param 1"* dan pada parameter *"SMS autoreply scenario return"* diisikan *"Sidang skripsi akan dimulai pada tanggal xx/xx/2008"*. Silahkan menghubungi

administrasi jurusan untuk mendaftarkan sidang. Terima kasih.” Setelah semua parameter tersebut diisikan, maka selanjutnya adalah mengklik tombol”Add”. Maksud dari isi data-data ini adalah jika pengguna mengirimkan SMS dengan kata ”SIDANG Skripsi” ke nomor *SMS Gateway*, maka sistem *SMS Gateway* akan secara otomatis menjawab dengan mengirimkan kembali SMS kepada nomor pengguna tersebut yang isinya seperti yang telah di set sebelumnya yaitu ”Sidang skripsi akan dimulai pada tanggal xx/xx/2008. Silahkan menghubungi administrasi jurusan untuk mendaftarkan sidang. Terima kasih.”

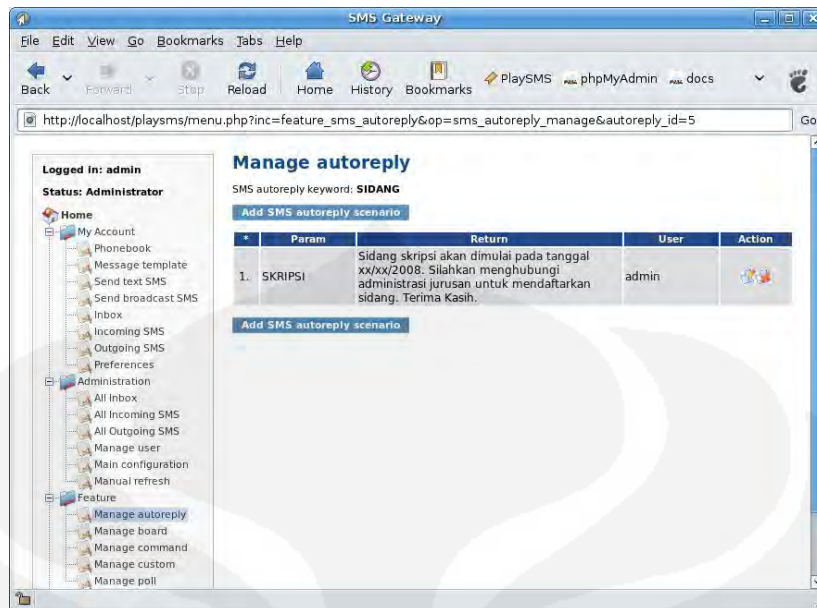


Gambar 4.24 Halaman menambahkan skenario SMS *auto reply*

Setelah tombol ”Add” berhasil di klik, maka akan muncul notifikasi bahwa skenario telah berhasil dibuat seperti pada Gambar 4.25. Gambar 4.26 menunjukkan skenario SMS *auto reply* yang telah berhasil dibuat.



Gambar 4.25 Notifikasi skenario SMS *auto reply* telah berhasil dibuat



Gambar 4.26 Skenario SMS *auto reply* telah berhasil dibuat

Setelah berhasil membuat skenario SMS *auto reply*. Maka selanjutnya adalah menguji skenario tersebut. Pengujian tersebut akan dibahas pada sub bab berikut.

4.6.2. Pengujian Auto-Reply SMS Berdasarkan Skenario Yang Telah Dibuat

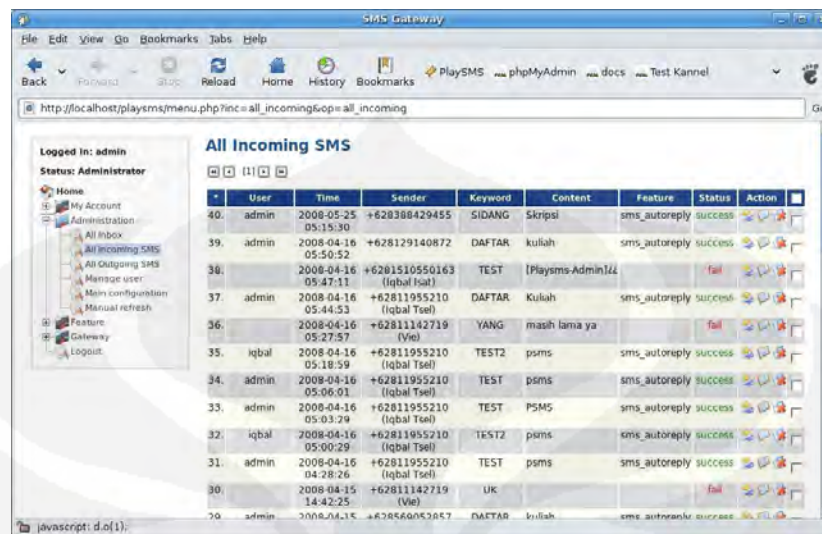
Setelah membuat skema dan skenario SMS *auto reply*. Kini saatnya menguji sistem tersebut dengan cara mengirimkan SMS yang berisi kata "SIDANG Skripsi" dan dikirimkan ke nomor *SMS Gateway*. Gambar 2.27 menunjukkan sebuah SMS yang dikirim untuk menguji sistem SMS *auto reply* dan dikirimkan kepada nomor *SMS Gateway* (dalam hal ini menggunakan nomor 0838831780).



Gambar 4.27 SMS yang dikirim untuk menguji SMS *auto reply*

Setelah SMS pengujian tersebut dikirim, maka sistem *SMS Gateway* akan menerima SMS pada kotak masuknya (*Inbox*) seperti ditunjukkan pada Gambar 4.28. Pada gambar tersebut, SMS yang baru dikirimkan sudah masuk yang ditunjukkan pada nomor urutan 40 pada kotak masuk SMS. Karena SMS yang dikirimkan sesuai dengan skema dan skenario SMS *auto reply*, maka pada kolom "Feature" ditandai dengan kata "sms_autoreply" sebagai

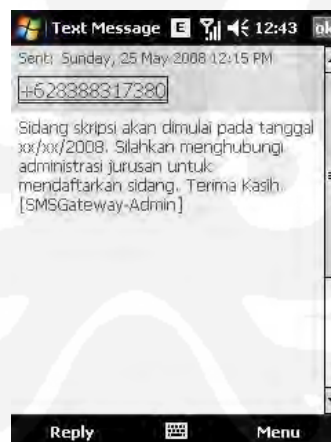
penanda fitur yang berlaku pada SMS tersebut. Kolom "Status" menunjukkan keberhasilan pengiriman SMS auto reply tersebut.



Gambar 4.28 Kotak masuk (Inbox) yang diterima SMS Gateway

Setelah sistem SMS *auto reply* berjalan, maka SMS Gateway akan mengirimkan SMS kembali kepada pengirim berdasarkan skema dan skenario yang tadi telah dibuat.

Gambar 4.29 menunjukkan SMS balasan dari sistem SMS *auto reply* yang telah diuji tadi.



Gambar 4.29 Kotak masuk (Inbox) yang diterima oleh pengirim

BAB 5

KESIMPULAN

Berdasarkan hasil pembahasan tentang Rancang bangun SMS gateway berbasis web dengan sistem operasi Linux Ubuntu dapat disimpulkan :

1. *SMS Gateway* dapat dibangun dengan sistem operasi Linux Ubuntu.
2. Dengan *SMS Gateway* pengguna dapat mengirim SMS melalui antar muka web dan dapat mengirimkan SMS *broadcast* ke banyak tujuan.
3. *SMS Gateway* dapat menangani aplikasi SMS *auto reply* dari pengguna dan mengirimkan SMS balasan secara otomatis.

DAFTAR ACUAN

- [1] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 3.
- [2] Website resmi organisasi GSM: <http://www.gsmworld.com>
- [3] Website resmi MDA-MobileData: <http://www.mda-mobiledata.org>
- [4] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 4.
- [5] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 5.
- [6] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 6.
- [7] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 52.
- [8] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 53.
- [9] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 55.
- [10] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 56.
- [11] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 58.
- [12] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 59.
- [13] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 59.
- [14] Gwenaël Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005), hal 62.
- [15] Website resmi proyek GNU: <http://gnu.org>
- [16] Website resmi Free Software Foundation: <http://fsf.org>
- [17] William Von Hagen, *Ubuntu Linux Bible* (Indiana Polis, USA: Wiley Publishing, Inc., 2007), hal 4.
- [18] Website resmi Ubuntu: <http://ubuntu.com>
- [19] Website resmi Canonical: <http://canonical.com>
- [20] William Von Hagen, *Ubuntu Linux Bible* (Indiana Polis, USA: Wiley Publishing, Inc., 2007), hal 10.
- [21] Ben & Peter Laurie, *Apache The Definitive Guide* (California, USA: O'Reilly & Associates, Inc., 1999) hal xi.
- [22] Website resmi NSCA: <http://www.ncsa.uiuc.edu>

- [23] Website resmi Apache Web Server: <http://apache.org>
- [24] Ben & Peter Laurie, *Apache The Definitive Guide* (California, USA: O'Reilly & Associates, Inc., 1999) hal 1.
- [25] Website resmi Ubuntu Linux: <http://www.ubuntu.com/>
- [26] Download packet Ubuntu Linux: <http://www.ubuntu.com/getubuntu/download>
- [27] Pengiriman paket CD ubuntu gratis: <http://shipit.ubuntu.com/>
- [28] Website resmi MySQL: <http://mysql.com>
- [29] Website resmi PHP: <http://php.net>

DAFTAR PUSTAKA

- Gutmans, Andi., Bakken, Stig Sæther., Rethans, Derick., *PHP 5 Power Programming* (Indiana Polis, USA: Prentice Hall-Pearson Education, Inc., 2005)
- Kadir, Abdul., *Dasar Pemrograman Web Dinamis Menggunakan PHP* (Yogyakarta: ANDI, 2003)
- Laurie, Ben., Laurie, Peter., *Apache The Definitive Guide* (California, USA: O'Reilly & Associates, Inc., 1999)
- Le Bodic, Gwenaël., *Mobile Messaging Technologies and Services SMS, EMS, and MMS* (Chichester, Inggris: John Wiley & Sons Ltd, 2005)
- Von Hagen, William., *Ubuntu Linux Bible* (Indiana Polis, USA: Wiley Publishing, Inc., 2007)