

**APLIKASI *REMOTE ACCESS CONSOLE* LINUX BERBASIS
SMS GATEWAY MENGGUNAKAN GAMMU DAN GAMBAS
(STUDY KASUS *PROXY SERVER GLOBAL DEVELOPMENT
LEARNING NETWORK UNIVERSITY OF RIAU*)**

TUGAS AKHIR

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

Oleh :

TAUFIQ RIDWAN
10551001510



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2010**

**APLIKASI *REMOTE ACCESS CONSOLE* LINUX BERBASIS
SMS GATEWAY MENGGUNAKAN GAMMU DAN GAMBAS
(STUDY KASUS *PROXY SERVER GLOBAL DEVELOPMENT
LEARNING NETWORK UNIVERSITY OF RIAU*)**

TAUFIQ RIDWAN
10551001510

Tanggal Sidang : 6 Juli 2010
Periode wisuda : Oktober 2010

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Global Development Learning Network (GDLN) Universitas Riau memiliki sebuah *proxy server* untuk menyediakan kebutuhan-kebutuhan *client* dalam menggunakan layanan internet. *Server* tersebut harus selalu dalam keadaan aktif untuk melayani kebutuhan *client*. Tetapi admin tidak bisa setiap saat memonitoring *server* tersebut, hal ini akan menjadi kendala jika *server* terjadi masalah karena *services* atau layanan yang disediakan pada *server* tidak berjalan dengan baik (*error*) sehingga perlu dilakukannya *restart* terhadap *services* tersebut atau melakukan *restart* terhadap *server* tersebut.

Pada tugas akhir ini dibuat sebuah aplikasi *remote console* linux berbasis SMS (*Short Message Service*) *gateway*. Alasan digunakannya media SMS karena layanan SMS pasti ada disetiap ponsel, berbeda dengan MMS dan WAP yang hanya ada dibeberapa merek dan tipe ponsel tertentu, serta menggunakan SMS tidak memakan banyak biaya. Aplikasi ini dibangun dengan menggunakan gambas sebagai bahasa pemrograman, gammu sebagai *driver* SMS *gateway*, dan *Mysql* sebagai *database*. Sementara pada SMS *gateway* menggunakan operator GSM (*Global system for mobile Communications*).

Hasil akhir penelitian ini adalah SMS perintah *shell* yang dikirimkan oleh admin dapat dieksekusi oleh sistem linux dan SMS balasan yang diterima admin adalah *output* hasil eksekusi yang sesuai dengan hasil yang dikeluarkan oleh sistem linux. Dari hasil pengujian dapat ditarik kesimpulan bahwa aplikasi *remote console* linux berbasis SMS *gateway* dapat diimplementasikan pada *proxy server* dan mampu menjadi alternatif admin untuk melakukan *remote* terhadap *server*.

Kata Kunci : *Client, Proxy Server, Remote Console, Server, Services, SMS Gateway*.

**REMOTE ACCESS CONSOLE LINUX APPLICATION WITH
SMS GATEWAY BASIS USING GAMMU AND GAMBAS
(CASE STUDY OF PROXY SERVER GLOBAL DEVELOPMENT
LEARNING NETWORK UNIVERSITY OF RIAU)**

TAUFIQ RIDWAN
10551001510

Date of Final Exam : 06th July 2010
Graduate Ceremony Period : October 2010

Informatics Engineering Departement
Faculty of Science and Technology
Islamic State University of Sultan Syarif Kasim Riau

ABSTRACT

Global Development Learning Network (GDLN) University of Riau has a proxy server to provide client needs in using Internet Services. The server should always be switched on to serve the client needs. But the admin can not monitor that server any time, this will become an obstacle if server problems occur because the services or services provided on the server does not run well (error) so the restarting these services or server are needed.

In this thesis, a remote console Linux applications with SMS (Short Message Service) gateway basis is made. The reason of media use SMS because of SMS service must exist in every cell phone, in contrast to MMS, and WAP which exist only in some particular cell phone brand and type, and using SMS is not costly. This application is built using Gambas as a programming language, Gammu as a SMS gateway driver, and MySQL as database. While the SMS gateway using GSM (Global System for Mobile Communications) operator.

The final results of this study is an shell SMS command sent by the admin can be executed by the Linux system and received a reply SMS admin is the output execution results are in accordance with the results of issued by the linux system. From the test results can be concluded that the a remote console Linux applications with SMS (Short Message Service) gateway basis can be implemented on a proxy server and can be an alternative to the admin to perform remote server.

Keywords : Client, Proxy Server, Remote Console, Server, Services, SMS Gateway.

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-3
1.3 Batasan Masalah.....	I-4
1.4 Tujuan Penelitian	I-4
1.5 Sistematika Penulisan	I-4
BAB II LANDASAN TEORI.....	II-1
2.1 Sistem Operasi Linux.....	II-1
2.1.1 Sejarah Linux	II-1
2.1.2 Distribusi Linux	II-2
2.1.3 Direktori <i>Filesystem</i> di Linux	II-4
2.2 <i>Shell</i>	II-6
2.3 Konsep Dasar SMS (<i>Short Message Service</i>).....	II-7
2.3.1 Alasan Menggunakan SMS.....	II-8
2.1.2 Mekanisme Kerja SMS	II-9
2.1.2 Implementasi Teknologi SMS	II-12

2.4	SMS Gateway	II-13
2.5	GSM (<i>Global For Mobile Communications</i>).....	II-14
2.6	Gammu.....	II-16
2.7	Gambas.....	II-18
2.7.1	Sejarah Nama Gambas	II-19
2.7.2	Kelebihan dan Kekurangan Gambas.....	II-20
2.8	Teknik Pengujian Aplikasi.....	II-22
2.8.1	Pengujian <i>Whitebox</i>	II-23
2.8.2	Pengujian <i>Blackbox</i>	II-23
2.8.3	<i>User Acceptance Test</i> (UAT).....	II-25
BAB III	METODOLOGI PENELITIAN.....	III-1
3.1	Pengumpulan data	III-1
3.2	Analisa.....	III-2
3.3	Perancangan	III-3
3.4	Implementasi	III-3
3.5	Pengujian.....	III-4
3.6	Kesimpulan dan Saran	III-5
BAB IV	ANALISA DAN PERANCANGAN	IV-1
4.1	Deskripsi Umum Sistem	IV-1
4.2	Analisa Sistem.....	IV-2
4.2.1	Analisa Sistem Yang Sedang Berjalan.....	IV-3
4.2.2	Analisa Sistem Yang Akan Dibangun	IV-5
4.2.2.1	Analisa Pada Sistem <i>Server</i>	IV-5
4.2.2.2	Analisa Fungsional Sistem.....	IV-6
4.2.2.3	Analisa Penyelesaian.....	IV-11
4.3	Perancangan Sistem	IV-13
4.3.1	Perancangan SMS Gateway	IV-13
4.3.2	Perancangan Algoritma RemoteServ	IV-14
4.3.3	Perancangan Layanan dan Format SMS	IV-16
4.3.4	Perancangan <i>Database</i>	IV-17
4.3.5	Perancangan Arsitektur Fisisk.....	IV-22

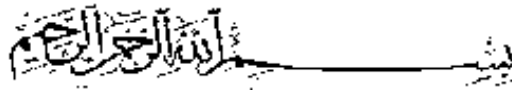
BAB V	IMPLEMENTASI DAN PENGUJIAN	V-1
5.1	Implementasi	V-1
5.1.1	Lingkungan Implementasi.....	V-3
5.1.2	Batasan Implementasi	V-4
5.1.3	Hasil Implementasi.....	V-5
5.2	Pengujian.....	V-13
5.2.1	Identifikasi Rencana Pengujian.....	V-13
5.2.2	Lingkungan Pengujian	V-14
5.2.3	Pengujian Layanan Pada Aplikasi.....	V-14
5.3	Kesimpulan Pengujian	V-27
BAB VI	PENUTUP	VI-1
6.1	Kesimpulan	VI-1
6.2	Saran.....	VI-2

DAFTAR PUSTAKA

LAMPIRAN

RIWAYAT HIDUP

KATA PENGANTAR



Puji syukur kehadiran Allah SWT berkat rahmat dan hidayah-Nya penulis bisa menyelesaikan Tugas Akhir yang berjudul **Aplikasi Remote Access Console Linux Berbasis SMS Gateway Menggunakan Gammu dan Gambas (Study Kasus Proxy Server Global Development Learning Network University of Riau)**. Tujuan penulisan Tugas Akhir ini adalah untuk memenuhi salah satu syarat kelulusan tingkat sarjana di Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.

Dalam menyelesaikan laporan Tugas Akhir ini penulis banyak dibimbing dan dibantu oleh berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. H. M. Nazir selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj. Yenita Morena, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Bapak Jasril, S.Si, M.Sc selaku ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi sekaligus sebagai penguji I.
4. Bapak Novriyanto, ST, M.Sc selaku Pembimbing I sekaligus penasehat akademis saya.
5. Bapak Harris Simaremare selaku pembimbing II.
6. Bapak Iwan Iskandar, ST selaku penguji II.
7. Seluruh dosen Jurusan Teknik Informatika UIN Suska Riau yang telah memberikan ilmu dan pengetahuan yang bermanfaat kepada Saya selama mengikuti perkuliahan di Jurusan Teknik Informatika.

8. Ayahanda dan Ibunda tercinta atas segala doa, nasehat, motivasi, dan kasih sayang yang tidak ternilai harganya serta pengorbanan moril dan materil untuk kebahagiaan dan kesuksesan Saya.
9. Heru, Amin, bapak Budiman, yang telah membantu saya dalam menyelesaikan tugas akhir ini.
10. Sahabat seperjuangan di Teknik Informatika Angkatan 2005. Terus berjuang untuk meraih kesuksesan.
11. Semua teman-teman di Fakultas Sains dan Teknologi.
12. Dan semua pihak yang telah membantu dalam penyelesaian tugas akhir ini yang tidak dapat saya sebutkan satu-persatu.

Demikian yang dapat Saya sampaikan, Saya menyadari dalam penulisan Tugas Akhir ini masih banyak terdapat kekurangan, oleh sebab itu kritik dan saran Saya harapkan dari para pembaca demi menyempurnakan kekurangan-kekurangan yang ada sehingga didapat hasil yang lebih memuaskan. Dan akhirnya Saya berharap semoga laporan ini bermanfaat bagi kita semua.

Pekanbaru, 6 Juli 2010

Taufiq Ridwan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* dilengkapi dengan sistem operasi khusus, yang disebut dengan sistem operasi jaringan (*network operating system*). Pada umumnya dalam sebuah sistem operasi terdapat beberapa layanan (*services*) yang menggunakan arsitektur *client/server*. Contoh dari layanan itu diantaranya, DHCP (*Dynamic Host Control Protocol*) *Server*, FTP (*File Transfer Protocol*) *Server*, *Proxy Server*, Samba (*File Sharing*) *Server*, DNS (*Domain Name Service*) *server*, dan HTTP (*Hypertext Transfer Protocol*) *Server*, dan lain-lainnya. Setiap layanan tersebut akan merespons permintaan data atau informasi dari sisi *client*, respons ini sesuai dengan jenis permintaan yang dilakukan *client*. Misalnya suatu atau beberapa *client* akan meminta IP (*Internet Protocol*) ketika dihubungkan kesuatu jaringan, maka DHCP *Server* akan merespons dengan membagikan IP kepada *client* tersebut.

Pada *Global Development Learning Network University of Riau* (GDLN UR) terdapat suatu jaringan komputer (*Local Area Network*). Yang termasuk didalamnya yaitu warnet GDLN, *office* GDLN, pasca sarjana, *wifi* untuk mahasiswa pasca sarjana, dan *video conference*. Jaringan tersebut diatur oleh sebuah komputer *server* yang menyediakan beberapa layanan (*services*).

Secara lengkap layanan yang disediakan pada *server* tersebut diantaranya, *Proxy Server*, *DHCP Server*, dan *Samba Server*. Ketiga jenis layanan ini dijalankan dalam sebuah komputer dengan menggunakan sistem operasi linux (*ubuntu Server*). Ketiga layanan tersebut memiliki tugas yang berbeda. *Proxy server* bertugas mengatur lalu lintas jaringan. Sementara itu *DHCP Server* digunakan untuk memberikan IP (*internet Protocol*) secara otomatis (*dynamic*) tanpa perlu memberikan IP satu persatu disetiap komputer, hal ini akan memudahkan admin untuk pengelolaan IP. Sementara itu *samba server* berfungsi sebagai pusat *file sharing* yang nantinya bisa diambil kapan saja oleh *client*. Selain itu *server* ini juga bertugas sebagai *gateway* untuk menghubungkan jaringan lokal dengan jaringan luar (*internet*).

Berdasarkan fungsinya, *server* yang terdapat pada GDLN tersebut harus terus menerus dalam keadaan aktif (*online*), dan admin bertugas selalu memonitoring *server* tersebut. Namun admin tidak bisa setiap saat berada di tempat *server* berada untuk mengontrol jaringan, hal ini akan menjadi kendala apabila *server* terjadi masalah, misalnya *server* tidak memberikan IP ketika *client* melakukan *request* (*DHCP*), hal ini tentunya akan menghambat aktifitas *client* untuk menggunakan internet, dari masalah tersebut admin berkewajiban memperbaiki masalah tersebut, sementara admin harus mempunyai akses setiap saat ke *server*. Solusi yang memungkinkan untuk memperbaiki masalah tersebut adalah melakukan *remote* ke *server* melalui jaringan internet, akan tetapi *server* yang berada di GDLN tidak memiliki IP *public* yang bisa diakses langsung dari luar (*internet*), dan jika solusi IP *public* diterapkan tentunya akan memakan biaya

yang cukup besar. Selain media IP *public* ada sarana lain yang lebih murah dan memungkinkan untuk dijadikan media antara admin dan *server* yaitu jaringan GSM (*Global System Mobile*) melalui SMS (*short message service*).

Dalam tugas akhir ini akan dirancang suatu aplikasi *remote access* melalui media SMS, yang nantinya *server* akan dihubungkan dengan sebuah ponsel yang berfungsi sebagai *SMS gateway*. *SMS gateway* ini dibangun sebagai media penghubung antara *server* dan admin. Dilihat dari sisi kebutuhan admin yang cenderung harus berkomunikasi setiap saat terhadap *server* maka dipilihlah media SMS sebagai komunikasi antara *server* dan admin, mengingat SMS dapat dilakukan kapan saja dan dimana saja, sehingga berkomunikasi dengan *server* semudah mengirimkan pesan singkat (SMS) kepada seseorang.

Dalam pengimplementasian aplikasi *remote console*, akan digunakan *tool* *gammu* dan bahasa pemrograman *gambas*. Alasan menggunakan *gammu* karena *support* dengan banyak tipe ponsel, serta mudah untuk diimplementasikan. Sementara alasan menggunakan *gambas* karena sudah memiliki tampilan *interface* GUI (*Grafik User Interface*) yang familiar dan *simple*. Selain itu juga alasan menggunakan aplikasi ini karena keduanya bersifat *open source* dibawah GPL (*General Public License*).

1.2 Rumusan Masalah

Permasalahan yang telah dirumuskan berdasarkan latar belakang diatas adalah bagaimana merancang dan mengimplementasikan *remote access console* linux melalui *SMS gateway* menggunakan *Gammu* dan *Gambas*.

1.3 Batasan Masalah

Pembuatan aplikasi ini memiliki beberapa batasan masalah, diantaranya :

1. Aplikasi ini digunakan hanya pada lingkungan atau keluarga linux ubuntu/debian.
2. Keamanan dari sistem hanya berupa *filtering* pada nomor ponsel yang sudah didaftarkan saja.
3. Pada SMS *gateway* hanya menggunakan operator GSM.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah menganalisa dan membangun suatu aplikasi *remote console* berbasis SMS *gateway* untuk mengelola dan mengontrol *server* di GDLN UR.

1.5 Sistematika Penulisan

BAB I : PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir ini, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, serta sistematika penulisan laporan tugas akhir.

BAB II : LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan spesifikasi pembahasan penelitian yang akan diangkat, yang terdiri dari pembahasan tentang *remote access console* linux, gammu dan gambas.

BAB III : METODOLOGI PENELITIAN

Bagian ini menjelaskan tentang metode pembuatan sistem, pengumpulan data dan objek, tempat serta waktu penelitian.

BAB IV : ANALISA DAN PERANCANGAN

Berisikan tentang analisa aplikasi *remote acces* melalui *console*. Pada tahap perancangan berisikan tentang perancangan sistem dan aplikasi, identifikasi kebutuhan, penggambaran alur kerja sistem.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Pada bagian implementasi dan pengujian berisi pembahasan mengenai pembuatan dan implementasi *remote access console* linux berbasis *SMS gateway* beserta *preview*, serta pengujian aplikasi.

BAB VI: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari pembahasan tentang *remote access console* berbasis *SMS gateway* dan beberapa hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Sistem Operasi Linux

Sistem operasi linux adalah sistem operasi yang bersifat *open source*. Dari waktu ke waktu sistem operasi ini terus berkembang, mulai dari industri, edukasi, hingga bisnis telah menggunakan linux sebagai sistem operasinya. Disamping linux digunakan pada tataran *server*, linux juga digunakan pada keperluan *desktop*. Sehingga setiap orang telah mampu menggunakan linux seperti halnya sistem operasi lain yang bersifat X windows.

2.1.1 Sejarah Linux

Linux pada awalnya dibuat oleh seorang mahasiswa Finlandia yang bernama Linus Torvalds. Dulunya Linux merupakan proyek hobi yang diinspirasi dari Minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum.

Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991. Kemudian pada tanggal 5 Oktober 1991, Linus mengumumkan versi resmi Linux, yaitu versi 0.02 yang hanya dapat menjalankan *shell bash* (GNU Bourne Again Shell) dan gcc (GNU C Compiler).

Saat ini Linux adalah sistem UNIX yang sangat lengkap, bisa digunakan untuk jaringan, pengembangan software dan bahkan untuk pekerjaan sehari-hari. Linux sekarang merupakan alternatif sistem operasi yang jauh lebih murah jika

dibandingkan dengan sistem operasi komersial (misalnya Windows 9.x/NT/2000/ME). Linux mempunyai perkembangan yang sangat cepat. Hal ini dapat dimungkinkan karena Linux dikembangkan oleh beragam kelompok orang. Keragaman ini termasuk tingkat pengetahuan, pengalaman serta geografis. Agar kelompok ini dapat berkomunikasi dengan cepat dan efisien, internet menjadi pilihan yang sangat tepat.

Karena kernel Linux dikembangkan dengan usaha yang independent, banyak aplikasi yang tersedia, sebagai contoh, *C Compiler* menggunakan gcc dari *Free Software Foundation GNU's Project*. *Compiler* ini banyak digunakan pada lingkungan Hewlett-Packard dan Sun. Sekarang ini, banyak aplikasi Linux yang dapat digunakan untuk keperluan kantor seperti untuk *spreadsheet*, *word processor*, *database* dan program grafis yang memiliki fungsi dan tampilan seperti Microsoft Office, yaitu Star Office. Selain itu, juga sudah tersedia versi Corel untuk Linux dan aplikasi seperti Matlab yang pada Linux dikenal sebagai Scilab.

2.1.2 Distribusi Linux

Terdapat banyak edaran atau distribusi Linux (lebih dikenali sebagai Distro), yang dibuat oleh individu, grup, dan lembaga lain. Masing masingnya mungkin disertakan dengan program sistem dan program aplikasi tambahan, di samping menyertakan suatu program yang memasang keseluruhan sistem dalam komputer baru.

Linux bisa didapatkan dalam berbagai distribusi (sering disebut Distro). Distro adalah bundel dari kernel Linux, beserta sistem dasar linux, program instalasi, *tools basic*, dan program-program lain yang bermanfaat sesuai dengan tujuan pembuatan distro. Ada banyak sekali distro Linux, diantaranya :

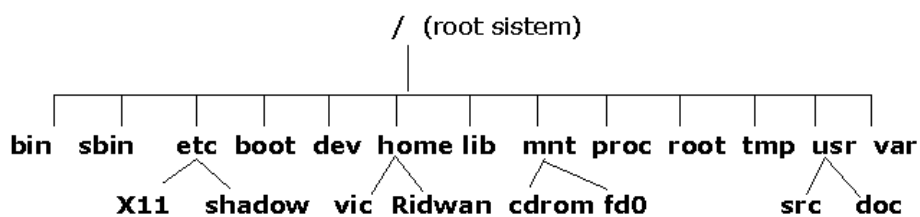
1. RedHat, distribusi yang paling populer, minimal di Indonesia. RedHat merupakan distribusi pertama yang instalasi dan pengoperasiannya mudah.
2. Debian, distribusi yang mengutamakan kestabilan dan kehandalan, meskipun mengorbankan aspek kemudahan dan kemutakhiran program. Debian menggunakan .deb dalam paket instalasi programnya.
3. Slackware, merupakan distribusi yang pernah merajai di dunia Linux. Hampir semua dokumentasi Linux disusun berdasarkan Slackware. Dua hal penting dari Slackware adalah bahwa semua isinya (kernel, *library* ataupun aplikasinya) adalah yang sudah teruji. Sehingga mungkin agak tua tapi yang pasti stabil. Yang kedua karena dia menganjurkan untuk menginstall dari source sehingga setiap program yang kita install teroptimasi dengan sistem kita. Ini alasannya dia tidak mau untuk menggunakan binary RPM dan sampai Slackware 4.0, ia tetap menggunakan libc5 bukan glibc2 seperti yang lain.
4. SuSE, distribusi yang sangat terkenal dengan YaST (*Yet another Setup Tools*) untuk mengkonfigurasi sistem. SuSE merupakan distribusi pertama dimana instalasinya dapat menggunakan bahasa Indonesia.

5. Mandrake, merupakan varian distro RedHat yang dioptimasi untuk pentium. Kalau komputer kita menggunakan pentium ke atas, umumnya Linux bisa jalan lebih cepat dengan Mandrake.
6. WinLinux, distro yang dirancang untuk diinstall di atas partisi DOS (WIndows). Jadi untuk menjalankannya bisa di-klik dari Windows. WinLinux dibuat seakan-akan merupakan suatu program aplikasi under Windows.

Dan masih banyak distro-distro lainnya yang telah tersedia maupun yang akan muncul.

2.1.3 Direktori *filesystem* linux

Direktori sitem linux menggunakan susunan hirarki (akar), sehingga setiap *user* mempunyai kekuasaan bertingkat-tingkat, kekuasaan tertinggi adalah *user* root, lagi-lagi untuk keamanan :



Gambar 2.1 Direktori Sistem Linux

1. /

Merupakan induk direktori sistem (root) yang menyimpan *file* sistem

2. /bin & /sbin (*binary, super binary*)

menyimpan program-program penting untuk sistem yang dapat dieksekusi (*executable*)

3. /boot

menyimpan semua *file-file* yang dibutuhkan saat booting linux, yaitu *mvlinux* (kernel *image*), *initrd* (initial runtime daemon), dll

4. /dev (*device*)

Menyimpan *file-file device* komputer seperti harddisk, cdrom, floppy, flashdisk, dll

5. /etc

Menyimpan *file-file* yang digunakan untuk konfigurasi sistem linux,

6. /home

Tempat menyimpan direktori dan dokumen tiap *users* yang ada

7. /lib (*library*)

Berisi kumpulan *library* yang dibutuhkan oleh sistem Linux

8. /mnt (*mount*)

Berisi direktori *mount point* dari *device*, seperti partisi HD, cdrom, floppy, tetapi ada distro yang menempatkan removable *device* seperti flashdisk, floppy, cdrom di /media

9. /proc (*process*)

Tempat menyimpan *file* informasi proses-proses yang sedang berjalan di sistem

10. /root

Direktori *home*-nya *user* root untuk menyimpan dokumen root

11. */tmp (temporary)*

Tempat menampung *file-file* sementara, misalnya pada saat instalasi

12. */usr (user)*

Berisi kumpulan program, *library*, dokumentasi yang dapat digunakan untuk semua *user*

13. */var (variable)*

Data yang dinamis di Linux, misal *www*, *mail*, *log*, *syslog*, *demog*

14. */opt (optional)*

Paket program tambahan, contoh: *netbean*, *Adobe Reader*, *Firefox*

2.2 *Shell*

Shell adalah penterjemah pada sistem linux. Perangkat lunak inilah yang menjadi jembatan antara pemakai dan sistem linux. *User* cukup memberikan perintah dan *shell* yang akan menanganinya. Perintahnya dapat berupa:

1. Perintah *built-in*, perintah ini merupakan bagian internal dari *shell*.
2. Perintah eksternal, perintah ini bukan bagian internal dari *shell* (dapat berupa utilitas atau program aplikasi)

Ada beberapa jenis *shell* yang beredar saat ini. Tiga diantaranya yang sangat menonjol adalah :

1. *Bourne Shell*
2. *C Shell*, dan
3. *Korn Shell*

Tabel 2.1 Tiga perintah *Shell* Yang Terkenal

Perintah <i>Shell</i>	Nama Program	Pembuat
<i>Bourne Shell</i>	Sh	Stephen R. Bourne
<i>C Shell</i>	Csh	Bill Joy
<i>Korn Shell</i>	Ksh	David G. Korn

Bourne shell adalah jenis *shell* yang tertua. Nama *shell* ini didasarkan atas nama pembuatnya, Stephen R. Bourne, dari laboratorium Bell, AT&T. *Shell* ini diperkenalkan pertama kali pada akhir 70-an dan dipakai sebagai *shell* utama di UNIX yang berkiblat AT&T.

C shell dibuat oleh Bill Joy. *Shell* ini menjadi standar pada system UNIX versi Berkeley. Format perintahnya menyerupai bahasa C.

Korn shell adalah *shell* yang dibuat oleh David G. Korn di laboratorium Bell, AT&T pada tahun 1986. *Shell* ini kompatibel dengan *Bourne shell*, artinya perintah-perintah yang didukung *Bourne shell* dapat dijalankan pada *Korn shell*.

2.3 Konsep Dasar SMS (*Short message service*)

Short message service atau akrab dikenal dengan SMS, telah menjadi suatu trend, bahkan sudah menjadi kebutuhan tersendiri. Dimana-mana bisa dapat dijumpai orang menuliskan pesan singkat yang juga dikenal dengan sebutan untuk SMS tadi.

SMS sendiri adalah sebuah teknologi yang memungkinkan untuk menerima dan mengirim pesan antar ponsel bergerak (*mobile*). Teknologi ini pertama kali

diperkenalkan pada tahun 1992 di Eropa oleh ETSI (*European Telecommunications Standard Institute*), dan pada awalnya menjadi standar untuk telepon wireless yang berbasis GSM (*Global system for mobile Communications*). Namun, teknologi CDMA dan TDMA pun memasukkan SMS ini sebagai fitur standar mereka.

Seperti namanya, SMS yang berarti layanan pesan pendek, maka besar data yang dapat ditampung oleh SMS ini sangatlah terbatas. Untuk satu SMS yang dikirim hanya dapat menampung 1120 *bites*. Bila dirubah dalam bentuk karakter, maka untuk satu SMS hanya dapat berisi paling banyak 160 karakter latin, dan 70 untuk karakter non-latin, seperti karakter China atau Jepang.

2.3.1 Alasan menggunakan SMS

Bisa dilihat SMS begitu banyak digunakan orang. SMS begitu populer bahkan menjadi gaya hidup. Ada beberapa alasan mengapa SMS menjadi begitu populer disegala kalangan, yaitu:

1. SMS adalah layanan yang pasti ada pada setiap ponsel.
2. SMS dapat dibaca dan dikirimkan, kapanpun, dimanapun kita berada.
3. SMS dikirimkan kenomor tujuan, walaupun nomor tujuan sedang tidak aktif, tapi SMS tetap akan dikirimkan dan akan diterima setelah nomor tersebut aktif kembali. Hal ini dikarenakan SMS memiliki masa tunggu, jadi jika masa tunggu belum habis, SMS akan tetap dikirim kenomor walaupun terlambat.

4. Ramah lingkungan, tidak menimbulkan suara yang berarti. Tidak seperti halnya panggilan langsung yang tentu akan mengganggu orang disekitarnya.
5. Layanan SMS pasti ada di setiap ponsel. Berbeda dengan MMS dan WAP yang hanya ada di beberapa merek dan tipe ponsel tertentu
6. SMS dianggap lebih murah dan praktis, dibandingkan dengan panggilan langsung.
7. SMS yang masuk tidak dapat ditolak. Hingga saat ini belum ada operator atau aplikasi khusus untuk menolak SMS yang masuk. Berbeda dengan panggilan langsung yang bisa ditolak bila kita malas untuk menerimanya.
8. SMS sudah menjadi tren tersendiri dan telah menjadi gaya hidup. Hal ini bisa dilihat dari televisi atau iklan-iklan yang beredar di setiap harinya.

2.3.2 Mekanisme kerja SMS

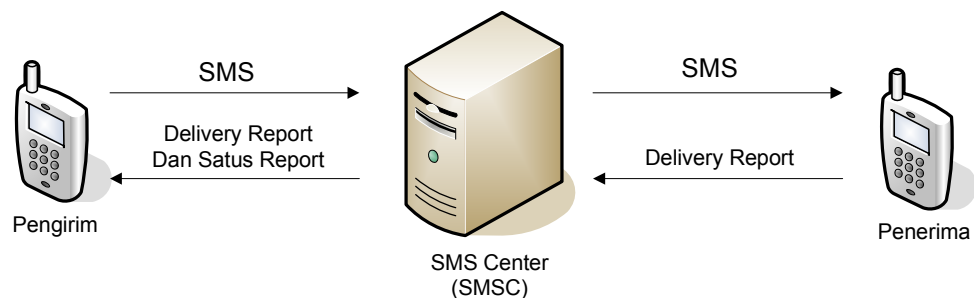
Ketika SMS dikirimkan dari suatu nomor ponsel ke nomor ponsel yang lainnya, maka SMS tersebut tidaklah langsung dikirimkan ke nomor tujuan tersebut. Tetapi SMS tersebut akan masuk terlebih dahulu ke SMS *center* (SMSC) operator yang kita gunakan. SMS *center* (SMSC) dapat juga diartikan dengan sebuah *server* yang bertanggung jawab sepenuhnya pada proses pengiriman sms dalam suatu operator. SMS yang dikirimkan dari suatu ponsel akan masuk ke SMSC ini, kemudian SMSC ini yang akan meneruskan sms tersebut ke nomor tujuan. Apabila ponsel tersebut dalam keadaan mati atau *offline*, SMSC ini akan menyimpan SMS tersebut untuk sementara waktu, hingga nomor tujuan tersebut

hidup kembali. Lama waktu penyimpanan sangat tergantung pada lama waktu yang telah ditetapkan oleh operator yang kita gunakan. Dan nomor yang telah menerima SMS tersebut akan mengirimkan laporan ke SMSC bahwa SMS telah diterima. Kemudian laporan tersebut akan diteruskan kembali ke nomor pengirim SMS.

Secara garis besar, mekanisme kerja pengiriman SMS dapat dibagi menjadi tiga macam, yaitu :

1. Pengiriman SMS dalam satu operator atau sering disebut dengan *intra-operator*.

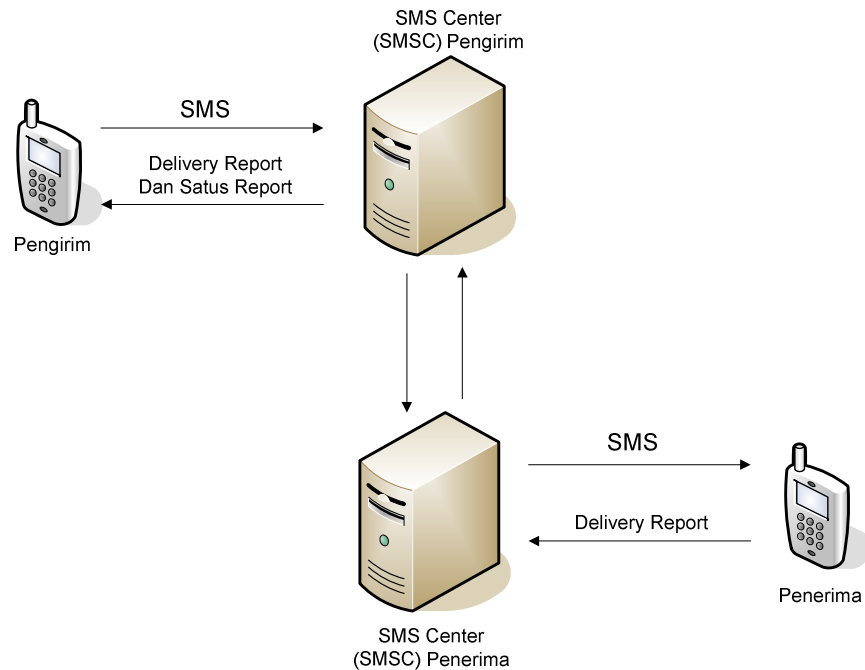
Mekanisme pengiriman *intra-operator* dapat dilihat dari gambar berikut.



Gambar 2.2 Mekanisme *intra-operator*

Dari gambar diatas, SMS yang dikirimkan oleh nomor pengirim akan dimasukkan terlebih dahulu ke SMSC. Dan dari SMSC akan meneruskan SMS tersebut ke nomor tujuan secara langsung. Nomor penerima kemudian akan mengirimkan *delivery report* ke SMSC. Kemudian SMSC akan meneruskan *delivery report* ke nomor pengirim SMS beserta *status report*.

2. Pengiriman SMS antar operator atau lebih dikenal dengan *inter-operator*



Gambar 2.3 Mekanisme *inter-operator*

Pada gambar diatas, selain masuk ke SMSC operator pengirim, SMS yang dikirimkan akan diteruskan ke SMSC operator penerima, kemudian baru diteruskan ke nomor tujuan. *Delivery report* pun akan melalui jalur tersebut, agar dapat sampai ke nomor pengirim SMS. Dalam mekanisme ini, terlihat ada sebuah komunikasi tidak langsung antar dua operator yang berbeda. Komunikasi tidak langsung ini akan terjadi setelah antar kedua operator terjadi sebuah kesepakatan kerja sama antar operator tersebut. Tidak adanya kesepakatan tersebut, dapat menyebabkan SMS yang dikirimkan ke nomor tujuan dengan operator yang berbeda, tidak akan sampai kepada nomor tujuan tersebut.

3. Pengiriman SMS dari operator suatu Negara kenegara lain atau lebih dikenal SMS internasional. Mekanisme ini terjadi tidak jauh berbeda dengan pengiriman SMS *inter-operator*. Perbedaannya terletak pada SMSC pengirim berada negara dengan SMSC penerima. Dan penambahan kode Negara nomor tujuan.

2.3.3 Implementasi Teknologi SMS

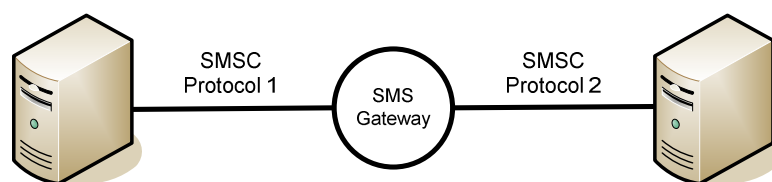
Kemudahan dan kepopuleran teknologi SMS banyak pihak yang memanfaatkan teknologi ini. Berikut beberapa implementasi teknologi SMS dalam kehidupan sehari-hari adalah :

1. Penyampaian undangan, kartu ucapan maupun hal-hal yang bersifat perseorangan atau individu
2. *Chatting*
3. Penyampain pesan atau informasi dari operator kepelanggan operator tersebut. Dapat berupa tagihan, berita, maupun layanan-layanan lainnya.
4. *Downloads ringtone, wallpaper*, maupun data-data *binary* lainnya
5. Promosi atau iklan suatu produk baru
6. Transaksi perdagangan
7. Kendali jarak jauh, atau *remote system*
8. *Votting* maupun *polling*
9. Pengganti surat

2.4 SMS Gateway

Gateway dalam kamus Inggris-Indonesia dapat berarti pintu gerbang. Namun dalam dunia komputer *gateway*, dapat diartikan juga sebagai jembatan penghubung antar satu sistem dengan sistem lain yang berbeda. Sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian SMS *gateway* dapat diartikan sebagai suatu penghubung untuk lalu lintas data-data SMS, baik yang dikirim maupun yang diterima.

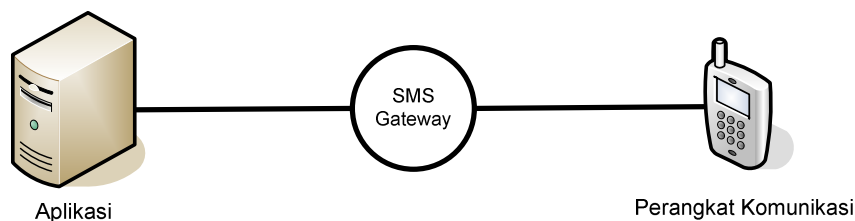
Pada awalnya SMS *gateway* dibutuhkan untuk menjembatani antar SMSC. Hal ini dikarenakan SMSC yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri, dan protokol-protokol itu sendiri bersifat pribadi. Sebagai contoh, Nokia memiliki protokol SMSC yang disebut CIMD, sedangkan CMG memiliki protokol yang disebut dengan EMI. SMS *gateway* ini kemudian ditempatkan di antara kedua SMSC berbeda tersebut, yang berfungsi sebagai *relay* bagi keduanya, yang kemudian akan menterjemahkan data dari protokol SMSC satu keprotokol SMSC lainnya yang dituju.



Gambar 2.4 Ilustrasi SMS Gateway

Namun seiring perkembangan teknologi komputer, baik dari sisi *hardware* maupun *software*, dan teknologi komunikasi, SMS *gateway* tidak lagi

dimaksudkan sebagaimana ilustrasi di atas. Dewasa ini, masyarakat lebih mengartikan *SMS gateway* sebagai jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel) dengan perangkat komputer mereka, yang menjadikan aktifitas menjadi lebih mudah dan menyenangkan. Pengertian ini kemudian lebih mengarah pada sebuah program yang mengomunikasikan antara sistem operasi komputer, dengan perangkat komunikasi yang terpasang untuk mengirim dan menerima SMS. Salah satu komunikasi yang terjadi, dapat dilakukan dengan mengirimkan perintah AT pada perangkat komunikasi tersebut, kemudian hasil operasinya dikirimkan kembali kekomputer. Dibutuhkan suatu *interface* baik dalam bentuk aplikasi maupun halaman web untuk membaca SMS yang masuk, atau mengirim SMS tersebut.



Gambar 2.5 Aplikasi *SMS Gateway*

2.5 GSM (*Global System For Mobile Communications*)

GSM (*Global System for Mobile Communication*) adalah sebuah teknologi komunikasi selular yang bersifat digital. Teknologi GSM banyak diterapkan pada *mobile communication* khususnya ponsel atau *handphone*. Teknologi ini memanfaatkan gelombang mikro dan pengiriman sinyal yang dibagi berdasarkan waktu, sehingga sinyal informasi yang dikirim akan sampai pada tujuan. GSM

dijadikan standar global untuk komunikasi selular sekaligus sebagai teknologi selular yang paling banyak digunakan orang di seluruh dunia.

GSM seperti dijelaskan oleh wikipedia, muncul pada pertengahan 1991 dan akhirnya dijadikan standar telekomunikasi selular untuk seluruh Eropa oleh ETSI (*European Telecommunication Standard Institute*). Pengoperasian GSM secara komersil baru dapat dimulai pada awal kuartal terakhir 1992 karena GSM merupakan teknologi yang kompleks dan butuh pengkajian yang mendalam untuk bisa dijadikan standar. Standar tipe *approval* untuk ponsel disepakati dengan mempertimbangkan dan memasukkan puluhan item pengujian dalam memproduksi GSM. Pada awal pengoperasiannya, GSM telah mengantisipasi perkembangan jumlah penggunanya yang sangat pesat dan arah pelayanan per area yang tinggi, sehingga arah perkembangan teknologi GSM adalah DCS (*Digital Cellular System*) pada alokasi frekuensi 1800 Mhz. Dengan frekuensi tersebut, akan dicapai kapasitas pelanggan yang semakin besar per satuan sel. Selain itu, dengan luas sel yang semakin kecil akan dapat menurunkan kekuatan daya pancar *handphone*, sehingga bahaya radiasi yang timbul terhadap organ kepala akan dapat di kurangi. Pemakaian GSM kemudian meluas ke Asia dan Amerika, termasuk Indonesia. Indonesia awalnya menggunakan sistem telepon selular analog yang bernama AMPS (*Advances Mobile Phone System*) dan NMT (*Nordic Mobile Telephone*). Namun dengan hadir dan dijadikannya standar sistem komunikasi selular membuat sistem analog perlahan menghilang, tidak hanya di Indonesia, tapi juga di Eropa. Pengguna GSM pun semakin lama

semakin bertambah. Akhirnya GSM tumbuh dan berkembang sebagai sistem telekomunikasi seluler yang paling banyak digunakan di seluruh dunia.

GSM, sebagai sistem telekomunikasi seluler digital memiliki keunggulan yang jauh lebih banyak dibanding sistem analog, di antaranya:

1. Kapasitas sistem lebih besar, karena menggunakan teknologi digital dimana penggunaan sebuah kanal tidak hanya diperuntukkan bagi satu pengguna saja. Sehingga saat pengguna tidak mengirimkan informasi, kanal dapat digunakan oleh pengguna lain.
2. Sifatnya yang sebagai standar internasional memungkinkan *international roaming*
3. Dengan teknologi digital, tidak hanya mengantarkan suara, tapi memungkinkan *service* lain seperti teks, gambar, dan video.
4. Keamanan sistem yang lebih baik
5. Kualitas suara lebih jernih dan peka.
6. *Mobile* (dapat dibawa kemana-mana)

Bagaimanapun, keunggulan GSM yang beragam pantas saja membuatnya menjadi sistem telekomunikasi seluler terbesar penggunaannya di seluruh dunia.

2.6 Gammu

Gammu (GNU *All Mobile Management Utilities*) merupakan aplikasi atau *tool SMS Gateway* yang bersifat *opensource*, gammu juga dapat dijadikan sebagai modul yang bisa digabungkan dengan bahasa pemrograman, seperti gambah,

python, ataupun *bash scripting*. Aplikasi *SMS Gateway* ini mampu menyalin SMS ponsel kedalam *database* dan mampu mengirim SMS yang ada didalam *database*, sehingga data yang terdapat dalam *database* dapat diolah sesuai kebutuhan. Selain itu gammu juga mamiliki fitur lain, seperti melihat info ponsel, pengelolaan *phonebook*, calender, *todo*, *todo*, MMS, dan lain sebagainya.

Ada banyak merk ponsel yang dapat didukung oleh gammu, diantara merk yang paling banyak digunakan adalah :

1. Alcatel
2. Ben-Q
3. Nokia
4. Samsung
5. Motorola
6. Toshiba
7. Vodafone
8. Siemens
9. Sharp
10. Philips
11. LG
12. Huawei
13. HTC
14. Apple
15. Vodafone
16. Dan banyak merk lainnya.

Dengan lisensi GNU/GPL, menjamin kebebasan menggunakan *tool* ini tanpa perlu takut masalah legalitas dan biaya mahal yang harus dikeluarkan. Agar dapat menggunakan atau memanfaatkan SMS *gateway*.

Kelebihan dari *tool* gammu ini adalah :

1. Mampu dijalankan di sistem operasi linux ataupun windows.
2. Banyak tipe / merk ponsel yang kompatibel oleh gammu.
3. Gammu menggunakan *database* MySQL.
4. Baik kabel data USB maupun SERIAL, semuanya kompatibel di gammu.
5. Dalam implementasi, konfigurasi gammu lebih mudah dibandingkan dengan *tool* SMS *Gateway* yang lain.

2.7 Gambas

Gambas merupakan salah satu bahasa pemrograman yang berorientasi pada grafis atau visual, yang berjalan pada sistem operasi linux. Secara umum bahasa pemrograman ini sangat mirip dengan visual basic milik Microsoft, namun pembuat gambas bernama Benoit Minisini menegaskan bahwa gambas tidak sama dengan visual basic dan tidak akan pernah sama.

Kemiripan yang terdapat pada gambas dengan visual basic karena gambas memang dibuat sebagai interpreter bahasa basic, yang juga merupakan cikal bakal visual basic. Meski demikian, kemiripan yang dimiliki oleh gambas membuat para programmer yang terbiasa bekerja dengan visual basic menjadi lebih mudah untuk migrasi dari Windows yang mahal, ke sistem operasi linux yang bersifat *open*

source dan *free*. Gamas adalah sebuah jawaban dari sekian banyak pertanyaan para programmer yang telah terbiasa bekerja dengan visual basic di Windows untuk beralih ke linux, tanpa harus lagi belajar bahasa pemrograman baru lagi di linux.

Selain gamas masih ada beberapa pemrograman lagi yang dikembangkan di linux untuk para programmer yang terbiasa bekerja di Windows. Di antaranya adalah :

1. Kylix, pemrograman Delphi untuk linux
2. Free Pascal, pemrograman Pascal untuk linux.

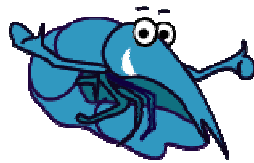
2.7.1 Sejarah Nama Gamas

Sudah menjadi sebuah budaya pada proyek yang dikerjakan dalam lingkungan GNU/Linux, bahwa hampir sebagian besar penamaan proyek menggunakan nama-nama dan logo binatang untuk proyek tersebut. Beberapa contoh diantaranya:

1. Python, merupakan nama ular
2. Camel, sebutan untuk unta
3. Lynx, nama binatang sejenis harimau
4. Bluebird, nama sejenis burung
5. Firefox, nama sejenis rubah, dan lain-lain

Untuk gamas sendiri, Minisini menginginkan nama program tersebut merupakan rekursif (perulangan) akronim, atau singkatan dari beberapa kata

mengandung *Basic*. akhirnya dipilihlah nama *Gambas*, yang mana pada kata ini mengandung kata *basic*. Arti dari *Gambas* itu sendiri adalah '*Gambas Almost Means Basic*'.



Gambar 2.6 Logo Gambas

Untuk logo gambas sendiri minisini mengambil gambar seekor udang dengan warna ungu, sebagaimana yang terlihat pada gambar diatas.

2.7.2 Kelebihan dan Kekurangan Gambas

Apabila berbicara masalah kelebihan suatu produk, hampir semua produk yang dibuat oleh manusia memiliki kelebihan dan kekurangan. Kelebihan dan kekurangan itu dapat ditemukan, bila telah dibandingkan dengan produk lain. Untuk gambas, maka yang menjadi perbandingan tentu saja adalah visual basic milik Microsoft. Karena visual basic adalah produk yang dekat dengan gambas.

Berikut adalah kelebihan yang dimiliki oleh gambas, dibandingkan dengan visual basic (Yunianto, 2006):

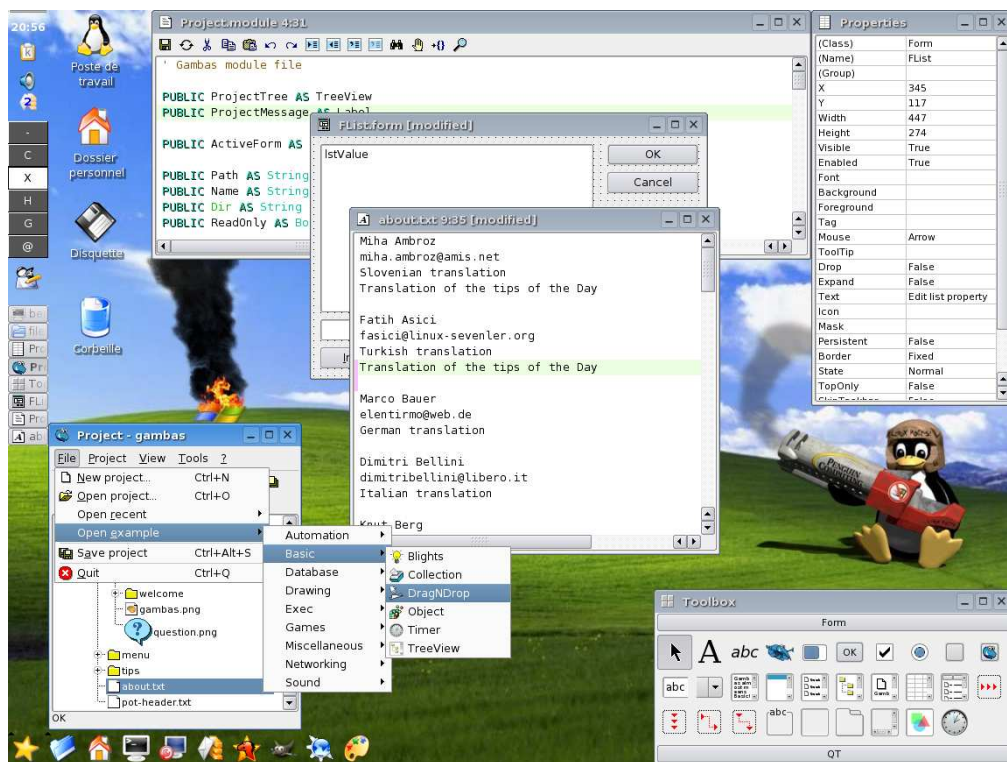
1. Sebagai program yang berjalan dibawah platform GNU/Linux, gambas memiliki sifat yang sama dengan linux tersebut, yaitu *open source* dan *free*. Hal ini tentu saja sangat menguntungkan bagi *developer* karena tidak perlu lagi memikirkan masalah lisensi dan biaya yang harus dikeluarkan, hanya untuk menggunakan visual basic.

2. Memiliki tampilan GUI yang familiar dan *simple*, sehingga sangat mudah dipelajari.
3. Berdasarkan informasi dari pembuatnya, lingkungan kerja gambah dibuat dengan gambah itu sendiri.
4. Gambah dapat dijadikan *debugger* yang baik.
5. Komponen program gambah diletakkan dalam satu direktori, sehingga ingin menghapus gambah dari system bisa dilakukan hanya dengan menghapus direktori tempat gambah berada.
6. Gambah sangat mudah diterjemahkan dalam bahasa apapun.
7. Gambah mampu mendukung *database* seperti PostgreSQL dan MySQL.
8. Saat membuat proyek baru dalam gambah, secara otomatis gambah akan membentuk direktori tersendiri untuk proyek tersebut. Hal ini tidak ditemukan pada visual basic, kecuali pada visual basic. Net.
9. Gambah mampu membaca *source code* dan meng-*import*-kan proyek dari visual basic. Sementara visual basic tidak dapat meng-*import*-kan proyek gambah ke visual basic.

Akan tetapi gambah juga memiliki kekurangan, diantara kekurangannya adalah (Yunianto, 2006) :

1. Minimnya dokumentasi ataupun forum-forum yang membahas pemrograman ini dibandingkan dengan visual basic. Tentu saja ini menjadi kendala bagi *developer* yang beralih ke gambah. Tentunya masalah ini akan terpecahkan dengan suatu ketekunan bermain linux tentunya.

2. Karena berjalan di linux, seperti biasanya, instalasi terkadang tidak semulus dipetunjuknya (*readme*), berbeda dengan visual basic yang hanya dengan “*next*” saja, proses instalasi terasa begitu mudah. Akan tetapi beberapa distribusi telah menyertakan Gamas sebagai paket yang terintegrasi didalamnya. Misalnya distribusi linux SUSE.



Gambar 2.7 lingkungan kerja Gamas di desktop KDE Linux

2.8 Teknik Pengujian Aplikasi

Pengujian dilakukan untuk melihat apakah aplikasi yang dibuat sesuai dengan kebutuhan atau tidak. Pengujian memiliki mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Ada banyak metode dalam

melakukan pengujian aplikasi, diantaranya adalah *white-box*, *black-box* dan *User Acceptance Test*.

2.8.1. Pengujian *Whitebox*

Pengujian *white box* atau disebut juga dengan *glass box* adalah pengujian yang didasarkan pada pengecekan terhadap detil perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Penentuan kasus uji disesuaikan dengan struktur sistem, pengetahuan mengenai program digunakan untuk mengidentifikasi kasus uji tambahan.

Penggunaan metode pengujian *white box* dilakukan untuk :

1. Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali
2. Menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*
3. Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
4. Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

2.8.2. Pengujian *Blackbox*

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari aplikasi. Karna itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat

fungsi suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *white-box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white-box*.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

Tidak seperti metode *whitebox* yang dilaksanakan diawal proses, ujicoba *blackbox* diaplikasikan di beberapa tahapan berikutnya. Karena ujicoba *blackbox* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Ujicoba didesain untuk dapat menjawab pertanyaan pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji ?
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba *blackbox*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari ujikasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan
2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, daripada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

2.8.3. *User Acceptance Test (UAT)*

User acceptance adalah sebuah proses untuk mendapatkan konfirmasi dari seorang *Subject Matter Expert* (ahli di bidangnya), terutama pemilik atau klien yang mengerti tentang objek yang sedang dalam phase pengetesan. Atau *user acceptance* dapat juga didefinisikan sebagai keinginan *user* atau grup *user* dalam memanfaatkan aplikasi atau teknologi informasi yang didesain untuk membantu pekerjaan mereka. *User Acceptance Test (UAT)* diposisikan sebagai tahap terakhir dari sebuah projek dan dilaksanakan sebelum klien menerima dan mengaplikasikan sistem baru tersebut. Hal yang penting untuk diperhatikan dalam UAT ini adalah, pembuat (tim developer) dan *tester* harus sudah benar-benar yakin kalau aplikasi tersebut sudah *bugfree* atau sudah tidak ada kemungkinan *error* dalam aplikasi yang dibuat.

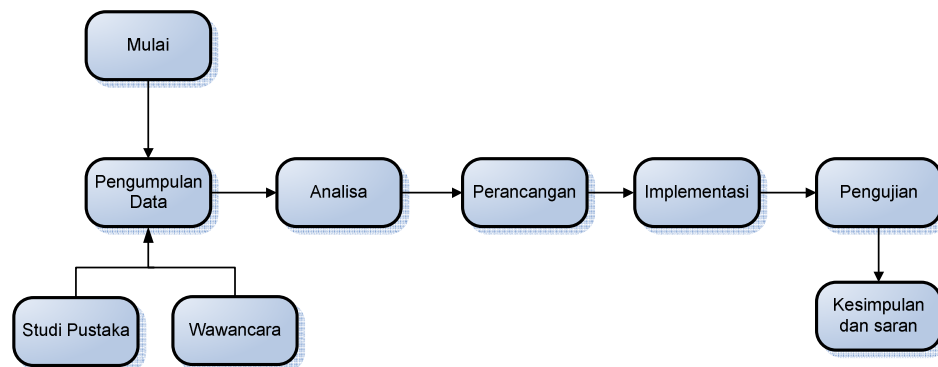
Kurangnya *user acceptance* akan sangat berpengaruh terhadap kesuksesan sebuah aplikasi yang dibuat. Karena itu, *user acceptance* harus dipandang sebagai

faktor sentral yang akan menentukan sukses atau tidaknya suatu projek aplikasi atau sistem informasi.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dipaparkan tentang langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian. Tahapan Penelitian yang akan dilaksanakan pada pembuatan *aplikasi remote console linux berbasis sms gateway* dapat dilihat pada gambar dibawah ini :



Gambar 3.1 Tahapan Penelitian

3.1 Pengumpulan Data

Pengumpulan data merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan data :

1. Studi Pustaka

Berfungsi sebagai pendukung dari penelitian yang akan dilaksanakan. Teori-teori mengenai jaringan, *server*, linux, dan *SMS gateway* yang digunakan bersumber dari buku, jurnal, forum-forum di internet dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan.

2. Wawancara

Wawancara berfungsi untuk mengumpulkan informasi yang akan berguna untuk tahap analisa dan tahap-tahap selanjutnya. Wawancara dilakukan terhadap admin *server* jaringan yang menggunakan linux dan para pengguna linux yang biasa menggunakan dengan basis text (*text mode*), dalam hal ini adalah admin GDLN UR.

3.2 Analisa

Disini akan dilakukan tahap mengenali seluruh permasalahan yang akan muncul, dan menganalisa data yang didapat dari proses pengumpulan data. Dalam tahap analisa ini dilakukan beberapa aktifitas :

1. Analisa yang sedang berjalan
2. Analisa sistem yang akan dibangun.

3. Analisa sistem pada Server
4. Menganalisa fungsional sistem dalam bentuk *Flowchart* dan DFD.
5. Analisa Penyelesaian.

Dalam tahap analisa ini akan dilakukan juga pemeriksaan kembali setiap fase yang dilalui, apabila masih terdapat kekurangan dan kesalahan, maka kembali ke fase awal (*waterfall*).

3.3 Perancangan

Setelah analisa dilakukan maka masuk kepada tahap perancangan, pada perancangan ini bagaimana mengimplementasikan masalah yang ada kedalam sebuah program. Pada tahap perancangan ini terdiri dari :

1. Perancangan SMS *gateway*.
2. Perancangan Algoritma.
3. Perancangan Layanan dan format SMS
4. Perancangan *database*.
5. Perancangan arsitektur dari sisi Admin, SMS *gateway*, dan *server*.

3.4 Implementasi

Setelah dilakukan perancangan, maka masuk ketahap implementasi. Implementasi ini merupakan tahap *coding* atau pengkodean program, lingkungan dalam melakukan

implementasi terbagi dua, yaitu lingkungan perangkat lunak dan perangkat keras. Untuk lingkungan implementasi perangkat lunak adalah :

1. Implementasi *database* dengan menggunakan MySQL
2. Implementasi pemrograman dengan bahasa pemrograman gambah. Gambah digunakan untuk membuat aplikasi.
3. Implementasi SMS *gateway* dengan menggunakan *tool* gammu. Gammu digunakan sebagai *driver* untuk ponsel pada sistem operasi linux.

Sementara Lingkungan implementasi perangkat keras memiliki spesifikasi sebagai berikut :

1. *Processor* : Intel Core 2 Duo 1.83 GHz
2. RAM : 1 G
3. *Harddisk* : 120 G
4. Ponsel sebagai SMS *Gateway* pada *server*.
5. Kabel Konektor USB penghubung antara *server* ke ponsel.
6. Dua Simcard (kartu) GSM. Yaitu digunakan pada ponsel SMS *gateway* dan digunakan pada admin dalam melakukan *remote*.

3.5 Pengujian

Pengujian merupakan tahapan dimana aplikasi akan dijalankan, ini bertujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan atau

tidak. Pada tahapan ini akan dilakukan pengujian terhadap perangkat lunak menggunakan metode pengujian *black-box* dan *User Acceptance Test*.

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari aplikasi. Pengujian ini memungkinkan sistem analis untuk memperoleh kondisi input yang akan mengerjakan seluruh keperluan fungsional sistem. Sementara *User Acceptance Test* adalah dengan melakukan pengujian terhadap admin *server* GDLN dengan mengisi atau menjawab kuesioner yang berkaitan dengan penggunaan aplikasi yang telah dibuat.

3.5 Kesimpulan dan Saran

Kesimpulan dan saran merupakan tahapan akhir dari sebuah penelitian. Kesimpulan dapat bernilai positif maupun negatif, hal ini sesuai dengan hasil yang diperoleh pada pengujian sistem. Sedangkan saran adalah harapan untuk masa yang akan datang bagi perkembangan sistem selanjutnya.

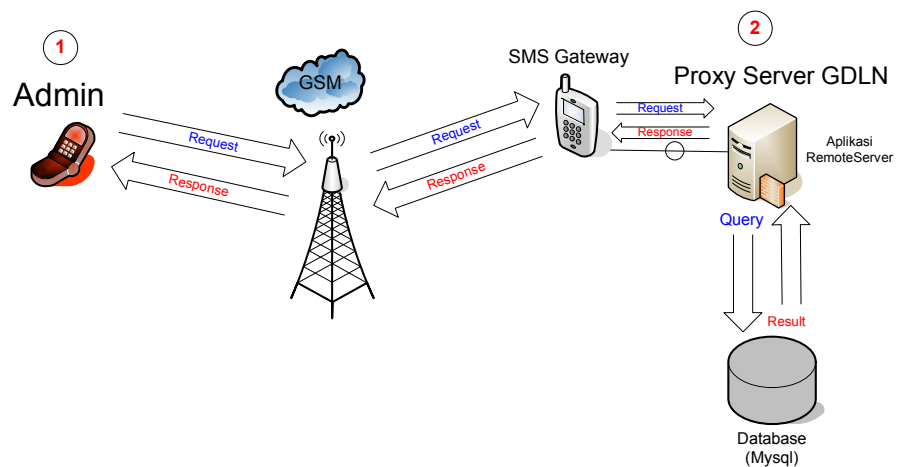
BAB IV

ANALISA DAN PERANCANGAN

Dalam membuat suatu aplikasi *remote* berbasis *console*, terlebih dahulu dilakukan analisa. Analisa dilakukan untuk memahami persoalan atau identifikasi masalah sebelum melakukan tahap perancangan. Sementara pada perancangan, yang dibuat harus memiliki kesesuaian dengan analisa sistem yang sebelumnya telah dilakukan.

4.1 Deskripsi Umum Sistem

Ditujukan untuk memberi gambaran cara kerja sistem yang akan dibangun secara umum. Sistem yang berjalan pada *server* berbasis *SMS Gateway*. Sementara admin hanya menggunakan layanan SMS pada ponsel untuk melakukan *remote* ke *server*. Deskripsi umum sistem dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Deskripsi Umum Sistem

Penjelasan mengenai deskripsi umum sistem adalah sebagai berikut :

1. Admin GDLN melakukan *remote* berupa perintah *shell* melalui layanan SMS dengan memanfaatkan jaringan GSM dan pastinya nomor ponsel admin tersebut telah terdaftar pada *database* yang boleh melakukan *remote* terhadap *server*.
2. *Proxy Server* GDLN dihubungkan dengan sebuah ponsel sebagai SMS *gateway* untuk menerima SMS yang dikirimkan oleh admin GDLN. Aktifitas yang terdapat pada *Proxy Server* ini adalah :
 - a. Melakukan input SMS yang diterima kedalam *database*, inputan tersebut berupa perintah yang dikirimkan oleh admin.
 - b. Mengeksekusi perintah yang diambil dari *database* SMS.
 - c. Mengirimkan *result* dari perintah yang telah di eksekusi kepada admin melalui SMS *gateway*. *Result* berupa *output* yang dihasilkan dari eksekusi perintah *shell* oleh sistem linux.

4.2 Analisa Sistem

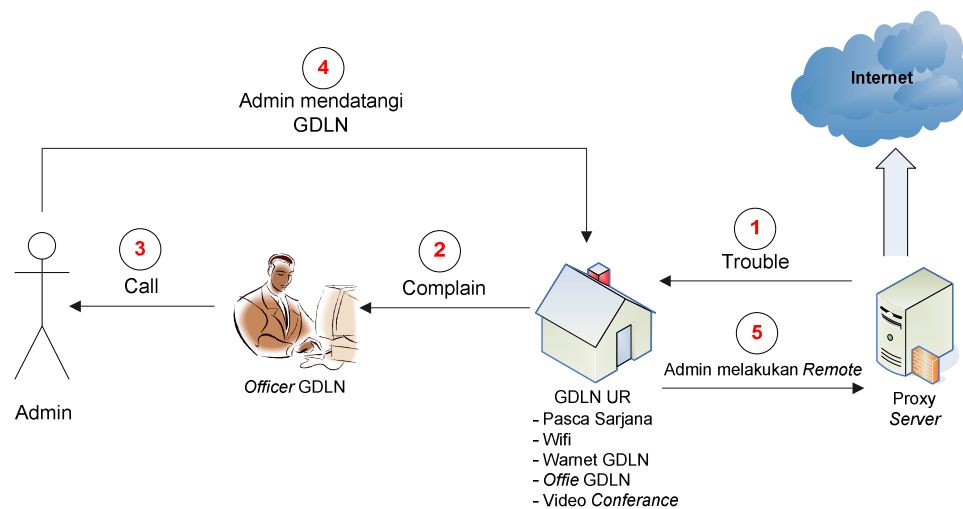
Analisa sistem dibagi dalam dua bagian pembahasan yaitu analisa pada sistem yang berjalan dan analisa pada sistem yang akan dibangun. Pada analisa sistem yang akan dibangun akan dibahas juga analisa pada sistem *server* termasuk SMS *Gateway*, dan analisa fungsional sistem yang terdiri dari analisa berupa DFD (*Data flow diagram*) dan *flowchart*.

4.2.1 Analisa Sistem Yang Sedang Berjalan

Analisa pada sistem yang sedang berjalan yaitu :

1. Seluruh aktifitas di jaringan GDLN terjadi setiap hari (termasuk hari minggu atau diluar jam kerja). Yang menggunakan layanan-layanan (*services*) pada *server* adalah warnet GDLN, wifi, pasca sarjana, *office* GDLN, dan *video conference*. Dan jaringan dikontrol oleh seorang admin.
2. Seluruh aktifitas jaringan dan layanan-layanan terpusat pada sebuah komputer *server* yang disebut dengan *proxy Server*.
3. Admin sering mengalami kesulitan dalam melakukan *remote access* jika *proxy server* terjadi masalah, misalnya beberapa *services* pada *server* yang tidak berjalan dengan baik dan perlu melakukan *restart services*, atau bahkan melakukan restart pada *proxy server* tersebut.

Untuk lebih jelasnya, alur kerja sistem yang sedang berjalan saat ini akan diilustrasikan melalui gambar 4.2 berikut.



Gambar 4.2 Alur Kerja Sistem yang Sedang Berjalan

Penjelasan gambar :

1. *Proxy server* GDLN terjadi suatu masalah, sehingga seluruh aktifitas pengguna (*client*) dalam satu jaringan di GDLN tidak berjalan dengan lancar, dan harus diperbaiki, sementara admin tidak berada pada lokasi *server* (GDLN).
2. Pengguna jaringan melaporkan (*complain*) masalah tersebut kepada *officer* GDLN, atau pengelola dari GDLN.
3. *Officer* tersebut menghubungi admin untuk memberitahukan bahwa jaringan di GDLN telah terjadi masalah, perlu dilakukan pengecekan dan perbaikan terhadap *server* tersebut. Yang bisa melakukan pengecekan dan perbaikan terhadap *server* hanya admin.
4. Karena tidak memiliki penghubung atau *remote access* secara langsung antara admin dan *server*, maka admin mendatangi GDLN langsung atau lokasi *server* untuk melakukan pengecekan terhadap masalah yang terjadi dan perbaikan terhadap masalah tersebut.
5. Setelah melakukan perjalanan memakan waktu admin sampai di GDLN barulah pengecekan dan perbaikan terhadap *server* dapat dilakukan.

Pada proses diatas tidak memiliki keefektifan dalam melakukan *remote* terhadap *server*, karena dalam menempuh perjalanan saat mendatangi GDLN membutuhkan waktu cukup lama untuk admin dalam melakukan *troubleshooting* terhadap *server*.

4.2.2 Analisa Sistem Yang Akan Dibangun

Sistem yang akan dibangun adalah bagaimana admin dapat berkomunikasi langsung atau melakukan *remote* langsung kepada *server* dengan perantara SMS *gateway* memanfaatkan jaringan GSM.

Pembahasan analisa pada sistem yang akan dibangun hanya pada sisi *server*, pada sisi administrator yang menggunakan sistem ini tidak akan dibahas karena pengguna sistem (administrator) hanya membutuhkan sebuah ponsel yang mampu mengirimkan pesan singkat (SMS) untuk melakukan *remote*. Analisa sistem pada *server* meliputi analisa kebutuhan data *server*, analisa fungsional sistem, dan analisa penyelesaian.

4.2.2.1 Analisa Sistem Pada *Server*.

Pada analisa kebutuhan data *server* mencakup juga analisa koneksi SMS *gateway* terhadap *server*, yaitu :

1. Kebutuhan data sebagai konfigurasi koneksi antara SMS *gateway* dengan komputer *server*. Data yang dibutuhkan berupa tipe koneksi, tipe ponsel dan *port* yang akan digunakan.
2. Kebutuhan data sebagai konfigurasi koneksi SMS *gateway* dengan *database*. Data yang dibutuhkan untuk koneksi SMS *gateway* dengan *database* yaitu berupa nama *database*, *username*, dan password dari *database*.
3. Kebutuhan data untuk eksekusi perintah SMS. Yaitu SMS yang diterima oleh SMS *gateway* akan disimpan kedalam *database*. Data yang tersimpan

dalam *database* tersebutlah yang akan diolah menjadi suatu perintah. Data yang akan dieksekusi membutuhkan informasi yaitu berupa informasi pengirim SMS dan isi dan format SMS.

4. Kebutuhan data untuk memberikan balasan kepada admin berupa hasil eksekusi SMS yang telah dilakukan oleh sistem linux. Data balasan SMS membutuhkan data nomer ponsel pengirim, dan isi dari hasil eksekusi SMS tersebut.
5. Kebutuhan data untuk membuat layanan tambahan selain melakukan *remote* terhadap *server*, layanan tersebut diantaranya pengelolaan data nomor admin dan SMS masuk dan SMS keluar. Untuk penambahan layanan tersebut membutuhkan nomer pengirim, format SMS dan isi dari SMS tersebut.

4.2.2.2 Analisa Fungsional Sistem.

Analisa fungsional akan dilakukan dengan menggambarkan bagaimana masukan diproses oleh aplikasi menjadi keluaran yang diharapkan oleh pengguna sistem. Adapun cakupan analisa fungsional disini membahas diagram alir dari *Context Diagram* (diagram konteks) sampai *Data Flow Diagram* (DFD), dan *flowchart* sistem. Context diagram dapat dilihat pada gambar 4.3. Untuk DFD dapat dilihat dari gambar 4.4, 4.5, dan 4.6. sedangkan *flowchart* sistem dapat dilihat pada gambar 4.7.

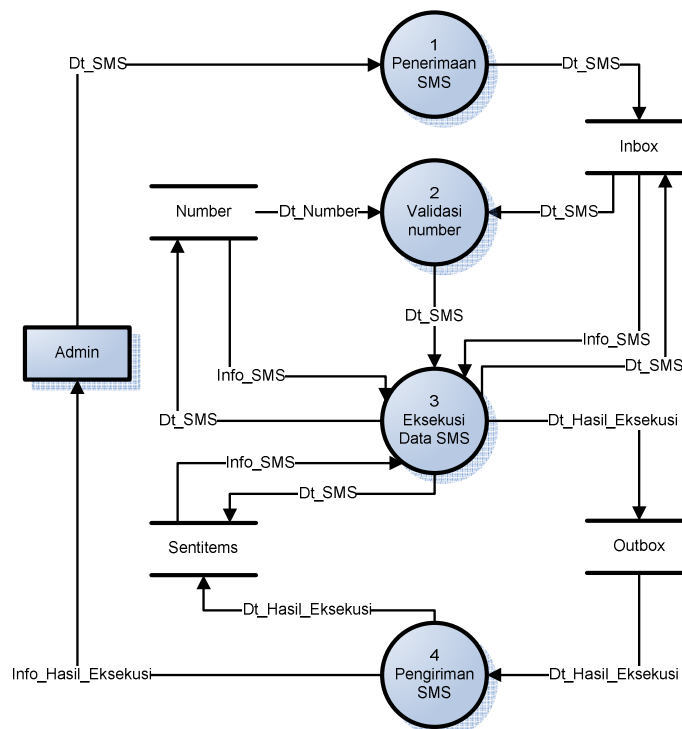
4.2.2.2.1 Data Flow Diagram (DFD)

Context diagram atau DFD level nol (0) digunakan untuk menggambarkan proses kerja sistem di *server* secara umum.



Gambar 4.3 Context Diagram Aplikasi Remote (RemoteServ)

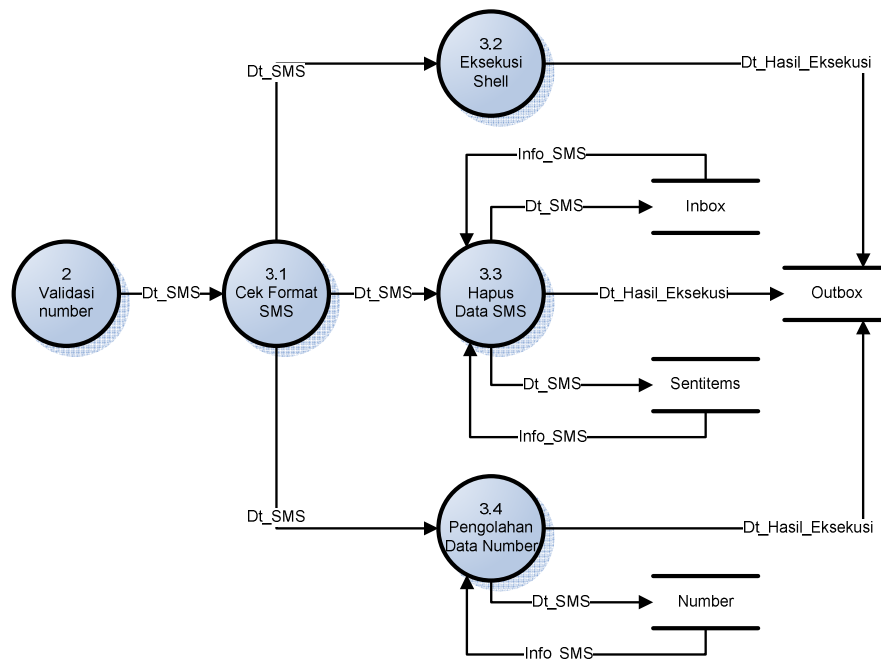
DFD digunakan untuk suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut tersimpan. DFD Level 1 dapat dilihat dari Gambar 4.4.



Gambar 4.4 DFD Level 1 RemoteServ

Tabel 4.1 DFD Level 1 RemoteServ

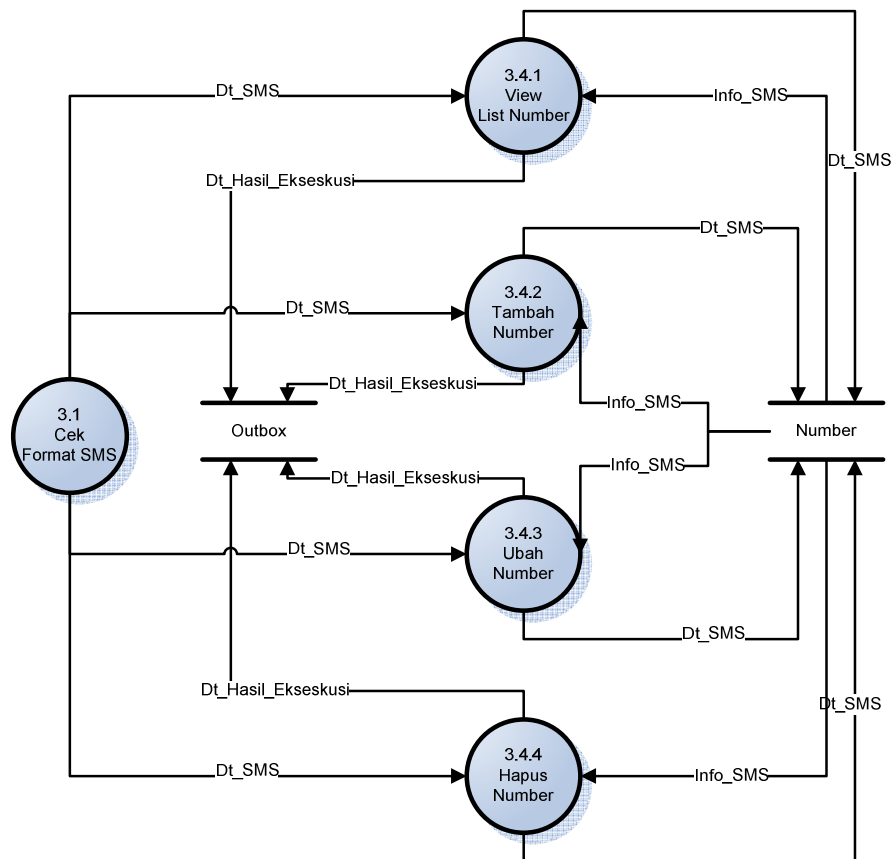
No	Nama Proses	Masukan	Keluaran	Keterangan
1.	Penerimaan SMS	Dt_SMS	Dt_SMS	SMS yang dikirim oleh admin akan diterima oleh SMS Gateway (gammu) dan menyimpannya dalam tabel inbox
2.	Validasi Number	Dt_SMS	Dt_SMS Dt_Number	SMS yang masuk akan diperiksa nomor pengirimnya, jika terdaftar maka SMS akan di eksekusi.
3	Eksekusi Data SMS	Dt_SMS	Dt_Hasil_Eksekusi	Data SMS akan di eksekusi berdasarkan Format SMS yang telah ditetapkan dan Pesan hasil Eksekusi akan dikirimkan ketabel outbox.
4	Pengiriman SMS	Dt_Hasil_Eksekusi	Info_hasil_Eksekusi	Hasil Eksekusi akan dikirimkan kepada Admin, dan SMS yang terkirim akan disimpan kedalam tabel senitem.



Gambar 4.5 DFD Level 2 Eksekusi Data SMS

Tabel 4.2 Proses DFD Level 2 Eksekusi Data SMS

No	Nama Proses	Masukan	Keluaran	Keterangan
3.1	Cek Format SMS	Dt_SMS	Dt_SMS	SMS akan diperiksa formatnya
3.2	Eksekusi <i>Shell</i>	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa perintah <i>Shell</i> .
3.3	Hapus Data SMS	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa penghapusan seluruh data di tabel inbox dan sentitem
3.4	Pengelolaan Data Number	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa pengelolaan data number.



Gambar 4.6 DFD Level 3 Pengelolaan Data Number

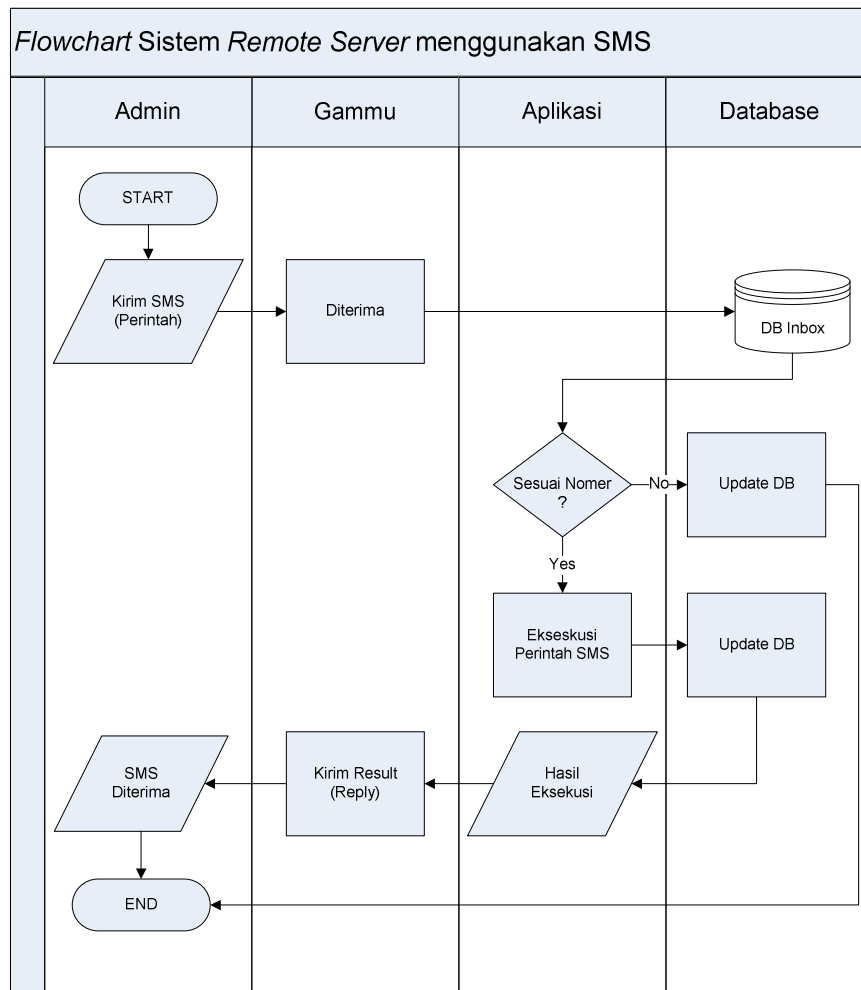
Tabel 4.3 DFD Level 2 Eksekusi Data SMS

No	Nama Proses	Masukan	Keluaran	Keterangan
3.4.1	View List Number	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa Menampilkan seluruh list number yang terdapat pada tabel number
3.4.2	Tambah Number	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa Penambahan data number.
3.4.3	Ubah Number	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa Perubahan pada data number
3.4.4	Hapus Number	Dt_SMS	Dt_Hasil_Eksekusi	SMS yang di eksekusi berupa Menghapus data Number

4.2.2.2.2 Flowchart Sistem RemoteServ

Proses-proses yang terjadi pada sistem RemoteServ secara rinci digambarkan menggunakan *flowchart*. Pada flowchart terdiri empat bagian, yaitu Admin, gammu, aplikasi RemoteServ, dan *Server*.

Admin merupakan orang yang akan melakukan *remote* terhadap *server*, yaitu admin melakukan *remote* berupa perintah yang akan dieksekusi dalam bentuk SMS, sementara gammu berfungsi sebagai *driver* penerima SMS yang dikirimkan oleh admin, sekaligus mengirimkan hasil eksekusi kepada admin. Dan aplikasi yang akan dibuat adalah aplikasi yang akan melakukan eksekusi perintah SMS, dan database adalah tempat penyimpanan data SMS yang dikirimkan admin, ataupun data yang dikirimkan sistem kepada admin.



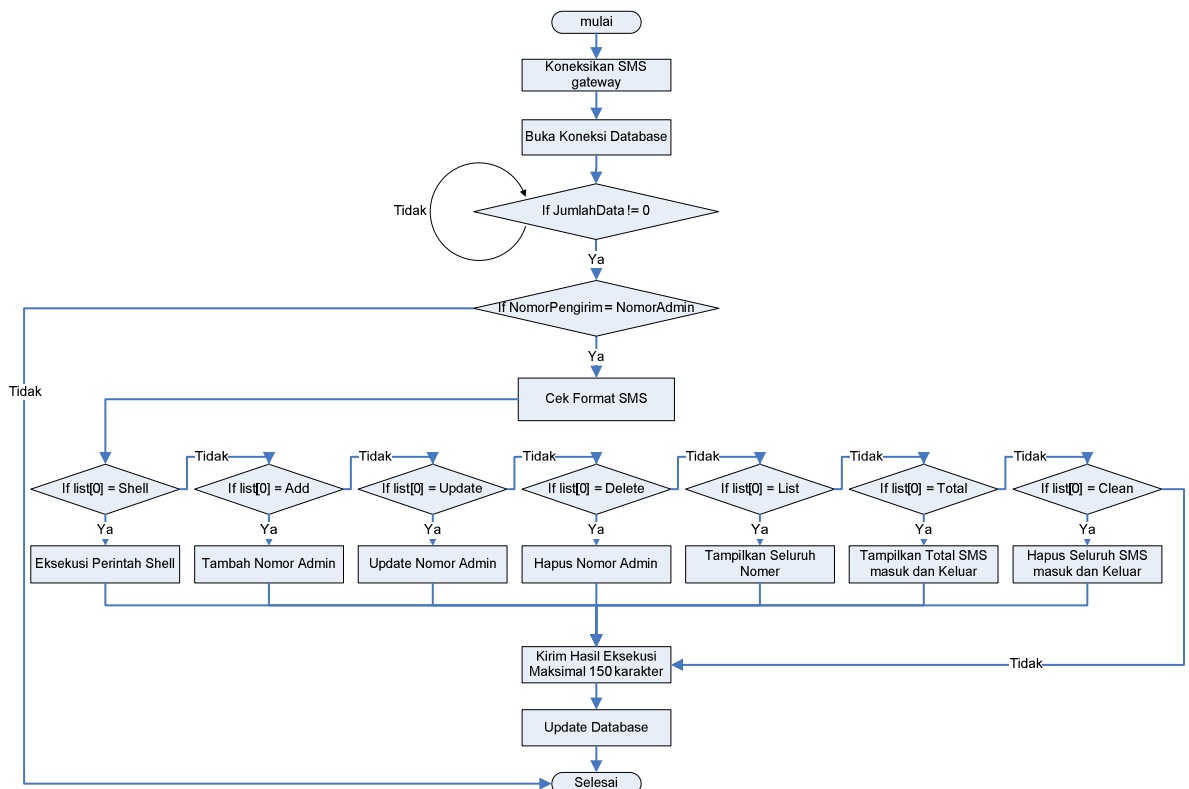
Gambar 4.7 *Flowchart Sistem*

4.2.3 Analisa Penyelesaian

Pada tahapan ini dilakukan analisa penyelesaian terhadap kasus permasalahan dengan menggunakan *flowchart* proses eksekusi data SMS. Berdasarkan *Data Flow Diagram* yang telah dibahas, terdapat tujuh format SMS yang dapat dieksekusi oleh aplikasi remoteserv, yaitu:

1. Format SMS untuk eksekusi perintah *shell*.

2. Format SMS untuk menambah nomor yang melakukan *remote* terhadap *server*.
3. Format SMS untuk mengubah nomor yang sudah terdaftar menjadi nomor baru.
4. Format SMS untuk menghapus nomor ponsel yang terdapat pada tabel *number*.
5. Format SMS untuk melihat seluruh nomor yang terdaftar dalam *database*.
6. Format SMS untuk melihat total seluruh SMS yang masuk dan SMS yang terkirim.
7. Format untuk menghapus SMS yang masuk dan SMS yang terkirim.



Gambar 4.8 Flowchart Aplikasi RemoteServ

4.3 Perancangan Sistem

Perancangan pada sistem ini berupa perancangan SMS gateway, perancangan database, Perancangan layanan dan format SMS, dan perancangan arsitektur fisik dari sistem remote acces server, serta komponen-komponen apa saja yang dibutuhkan pada admin, SMS gateway, dan server.

4.3.1 Perancangan SMS Gateway

SMS gateway digunakan sebagai jembatan penghubung antara admin dengan server. Aplikasi SMS gateway yang digunakan adalah gammu. Gammu akan diinstallkan pada komputer server dan dilakukan beberapa konfigurasi terkait dengan koneksi ponsel dan koneksi database. Alat (device) yang akan digunakan sebagai SMS gateway adalah :

1. Ponsel sebagai SMS gateway, yaitu Siemens M55
2. Kabel interface USB Penghubung antara ponsel SMS gateway dengan server.

Pabel 4.4 dan 4.5 berikut akan dijelaskan konfigurasi dan parameter yang harus diinputkan pada file konfigurasi gammu.

Tabel 4.4 konfigurasi koneksi ponsel SMS gateway dengan server

Konfigurasi	Isi parameter	Keterangan
Port	/dev/ttyUSB0	Koneksi port yang digunakan, yang digunakan adalah port USB
Model	AT	Model tipe ponsel yang akan digunakan.
Connection	at19200	Baud rate kecepatan transfer data, kecepatan yang digunakan adalah 19200

Tabel 4.5 konfigurasi koneksi SMS Gateway dengan *database*

Konfigurasi	Isi parameter	Keterangan
User	root	Username yang digunakan ketika akan membuka koneksi. User yang digunakan adalah root.
Password	xxxxx	Password <i>database</i> yang sesuai dengan <i>username</i> root
Pc	localhost	Komputer <i>database server</i> yang digunakan adalah localhost
Database	sms	Nama <i>database</i> yang digunakan

4.3.2 Perancangan Algoritma RemoteServ

Algoritma yang akan dibahas adalah algoritma mulai dari proses koneksi SMS gateway, eksekusi SMS berdasarkan formatnya, hingga pengiriman hasil eksekusi (*output*).

```

PROCEDURE Eksekusi (Input NumberPengirim, TextDecoded;)
Algoritma :
BukaKoneksiSMGW();
BukaKoneksiDB();
While true
Data = Select Id, TextDecoded, NumberPengirim from inbox where
Processed='false';
#Ambil karakter dari karakter ke-6
Perintah = (TextDecoded, -6)
#Cek Perintah yang belum dieksekusi
If JumlahData != 0 Then
    If NumberPengirim = NumberAdmin then #cek nomor pengirim
        Try List = Split(TextDecoded, ' ') #Baca SMS berdasarkan spasi
#Format SMS Untuk melakukan Perintah Remote Shell
        If List[0] = Shell Perintah then
            SHELL Perintah > /tmp/dump
            {Eksekusi Perintah shell dan Simpan hasil

```

```

        eksekusi    pada file /tmp/dump}
        SHELL CharReplacer.jar #konversi karakter single
        backquote
        Open /tmp/dump
        Read /tmp/dump
        Output = Right(150)
        #ambil 150 karakter terakhir isi file /tmp/dump
        Kirim Output ke NomorPengirim
    Else
#Format SMS untuk Penambahan Nomor yang boleh melakukan remote
        List[0] = Add Number
        Tambah NumberAdmin kedalam database
    Else
#Format SMS untuk Mengubah nomor yang sudah terdaftar
        List[0] = Update Number
        Ubah NumberAdmin pada database
    Else
#Format SMS untuk menghapus nomor yang terdaftar
        List[0] = Delete Number
        Hapus NumberAdmin pada database
    Else
#Format SMS untuk melihat list nomor yang terdaftar
        List[0] = List
        Lihat Seluruh NumberAdmin yang terdapat dalam database
    Else
#Format SMS untuk menampilkan total SMS masuk dan SMS terkirim
        List[0] = Total
        Lihat Total SMS masuk dan SMS terkirim
    Else
#Format SMS untuk menghapus seluruh SMS masuk dan SMS terkirim
        List[0] = Clean
        Hapus seluruh SMS masuk dan SMS terkirim
    Endif
    Endif
Endif
Update inbox
End While

```

4.3.3 Perancangan Layanan dan Format SMS.

Berdasarkan algoritma RemoteServ terdapat tujuh perintah yang dapat dieksekusi oleh aplikasi RemoteServ. Sebagai tambahan adalah format untuk bantuan (*help*). Pada tabel 4.6 berikut ini akan dijelaskan format-format layanan yang digunakan untuk menggunakan layanan dari Aplikasi RemoteServ.

Tabel 4.6 Format SMS Layanan RemoteServ

Layanan	Format
Melihat Bantuan	Help Contoh SMS : Help
Melakukan Perintah <i>Remote</i>	Shell <spasi>[perintah] Contoh SMS : Shell shutdown -r 0
Tambah Nomor kedalam <i>database</i>	Add <Spasi>[Nomor]<Spasi>[Nama] Contoh SMS : Add +6285278270116 Taufiq
Ubah Nomor yang terdapat pada <i>database</i>	Update <spasi>[Nomor Lama] <spasi>[Nomor baru] <spasi>[Nama baru] Contoh SMS : Update +6285278270116 +6285278910112 Ridwan
Hapus List Nomor dari <i>database</i>	Delete <spasi>[Nomor] Contoh SMS : Delete +6285278910112
Cek total SMS masuk dan SMS keluar	Total Contoh SMS : Total sms
Hapus Seluruh SMS	Clean Contoh SMS : Clean sms
Melihat List Nomor yang terdaftar	List Contoh SMS : List

Beberapa hal yang harus diperhatikan adalah untuk melakukan perintah *remote*, perintah tersebut bersifat *case sensitive*, artinya perintah tersebut sangat

rentan terhadap kesalah (misalnya pemakaian huruf besar atau kecil), karena perintah tersebut akan dibaca oleh sistem linux. Pada layanan **Add, update, delete**, nomor yang diinputkan harus diawali dengan kode negara (untuk indonesia +62).

4.3.4 Perancangan Database

Database yang digunakan pada sistem ini adalah *database* bawaan yang telah ada dalam paket instalasi gammu. Tabel-tabel yang diperlukan telah mencukupi untuk memenuhi kebutuhan dari sistem. Hanya saja ada penambahan sebuah tabel khusus untuk menyimpan data number. Tabel ini difungsikan sebagai tabel nomor yang boleh melakukan *remote*.

Database sistem ini memiliki nama *database* sms. *Database* bawaan gammu ini memiliki beberapa tabel, diantaranya yaitu tabel inbox, tabel outbox, dan tabel sentitems. Tanel-tabel tersebut tidak memiliki relasi atau hubungan, karena hanya sebagai tempat penyimpanan SMS masuk dan SMS keluar saja. Tabel inbox hanya untuk penyimpanan SMS masuk, tabel outbox hanya untuk tempat sementara untuk SMS dikirimkan, setelah SMS terkirim maka SMS tersebut akan disimpan pada tabel sentitems, dan SMS yang ada pada tabel outbox akan terhapus secara otomatis. Pada tabel tambahan yaitu tabel number yang berfungsi sebagai tempat menyimpan nomor yang boleh melakukan *remote*. Nomor yang ada pada tabel ini akan menjadi validasi ketika suatu SMS akan dieksekusi, jika nomor terdaftar pada tabel number maka SMS tersebut akan dieksekusi, jika tidak maka SMS tidak akan dieksekusi.

Nama Tabel : Inbox

Primary Key : ID

Deskripsi : Menyimpan SMS Masuk

Tabel 4.7 *Inbox*

No	Nama Field	Tipe Data	Null	Keterangan
1	UpdatedInDB	Date	NOT NULL	Waktu update terakhir
2	ReceivingDateTime	Date	NOT NULL	Waktu SMS diterima
3	Text	Text	NOT NULL	Decode Text SMS
4	SenderNumber	varchar(20)	NOT NULL	Nomor Ponsel Pengirim
6	UDH	Text	NOT NULL	Encoded Text UDH (User Data Header)
7	SMSCNumber	varchar(20)	NOT NULL	Nomor SMS <i>center</i>
8	Class	Integer	NOT NULL	Kelas SMS
9	Coding	Enum(Default_No_Compression,Unicode_No_Compression,8bit,Default_Compression,Unicode_Compression)	NOT NULL	Coding teks SMS
10	TextDecoded	varchar(160)	NOT NULL	Isi text SMS, atau decode teks (dalam bentuk Abjad).
11	ID	Integer	NOT NULL	Nomor ID SMS
12	RecipientID	Text	NOT NULL	Daemon Gammu yang telah ditambahkan
13	Processed	Enum(false,true)	NOT NULL	Penanda SMS yang sudah diproses atau belum

Nama Tabel : Outbox

Primary Key : ID

Deskripsi : Tabel untuk mengirimkan SMS

Tabel 4.8 *Outbox*

No	Nama Field	Tipe Data	Null	Keterangan
1	UpdatedInDB	Date	NOT NULL	Waktu update terakhir
2	InsertIntoDB	Date	NOT NULL	Waktu SMS dimasukkan ke dalam database
3	SendingDateTime	Date		Nilai tertentu, memaksa pengiriman setelah beberapa waktu yang telah ditentukan
4	Text	Text	NULL	SMS dalam bentuk hexa
5	DestinationNumber	varchar(20)	NOT NULL	Nomor Ponsel Tujuan / penerima
6	Coding	Enum(Default_No_Compression,Unicode_No_Compression,8bit,Default_Compression,Unicode_Compression)	NOT NULL	Coding teks SMS
7	UDH	Text	NULL	User Data Header (UDH) dalam bentuk hexa.
8	Class	Integer	NOT NULL	Kelas SMS
9	TextDecoded	Varchar(160)	NOT NULL	Isi text SMS, atau decode teks (dalam bentuk Abjad).
10	ID	Integer	NOT NULL	Nomor ID SMS

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>	Keterangan
11	MultiPart	Enum(false,true)	NOT NULL	Info apakah SMS lebih dari 1 SMS (multi SMS)
12	RelativeValidity	Integer	NOT NULL	Validasi SMS.
13	SenderID	Varchar(255)	NULL	Pengurutan Pengiriman SMS
14	SendingTimeOut	Date	NULL	Waktu yang ditargetkan SMSD saat SMS dikirimkan.
15	DeliveryReport	Enum(default,yes,no)	NOT NULL	Laporan Pengiriman SMS
16	CreatorID	Text	NOT NULL	Identifikasi Pengirim

Nama Tabel : Sentitems

Primary Key : ID

Deskripsi : Tempat menyimpan SMS keluar / terkirim

Tabel 4.9 *Sentitems*

No	Nama <i>Field</i>	Tipe Data	<i>Null</i>	Keterangan
1	UpdatedInDB	Date	NOT NULL	Waktu update terakhir
2	InsertIntoDB	Date	NOT NULL	Waktu SMS dimasukkan ke dalam database
3	SendingDateTime	Date	NOT NULL	Waktu pesan telah terkirim
4	DeliveryDateTime	Date	NULL	Waktu laporan diterima
5	Text	Text	NOT NULL	SMS dalam bentuk hexa
6	DestinationNumber	Varchar(20)	NOT NULL	Nomor Ponsel Tujuan

No	Nama <i>Field</i>	Type Data	Null	Keterangan
7	Coding	Enum(Default_No_Compression,Unicode_No_Compression,8bit,Default_Compression,Unicode_Compression)	NOT NULL	Coding teks SMS
8	UDH	Text	NOT NULL	User Data Header (UDH) dalam bentuk hexa.
9	SMSCNumber	Varchar(20)	NOT NULL	Nomor SMS <i>center</i>
10	Class	Integer	NOT NULL	Kelas SMS
11	TextDecoded	Varchar(160)	NOT NULL	Isi text SMS
12	ID	Integer	NOT NULL	Nomor ID SMS
13	SenderID	Varchar(255)	NOT NULL	Pengurutan Pengiriman SMS
14	SequencePosition	Integer	NOT NULL	Nomor SMS dalam urutan SMS yang dikirm
15	Status	Enum(SendingOK,SendingOKNoReport,SendingError,DeliveryOK,DeliveryFailed,DeliveryPending,DeliveryUnknown,Error)	NOT NULL	Status pengiriman pesan
16	StatusError	Integer	NOT NULL	Status pengiriman dari laporan pengiriman pesan
17	TPMR	Integer	NOT NULL	Referensi pesan dari spesifikasi GSM
18	RelativeValidity	Integer	NOT NULL	Validasi SMS.
19	CreatorID	Text	NOT NULL	Identifikasi Pengirim

Nama Tabel : Number

Primary Key : number

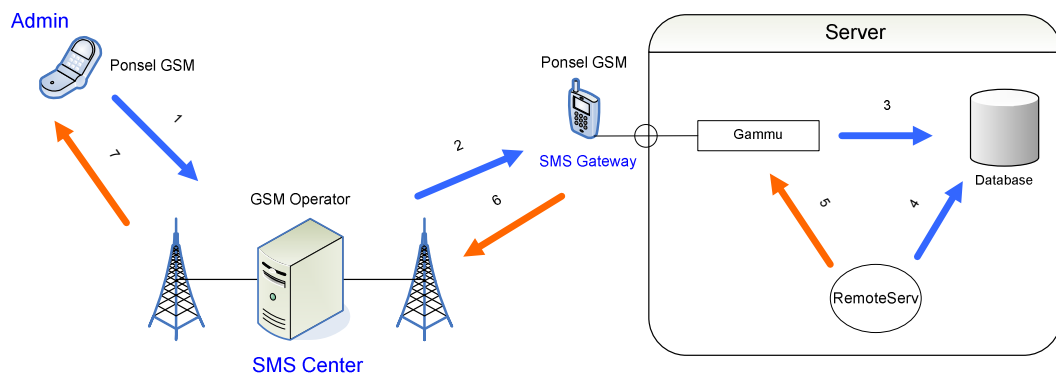
Deskripsi : Tempat menyimpan number admin

Tabel 4.10 *Number*

No	Nama Field	Tipe Data	Null	Keterangan
1	ID	Int (10)	NOT NULL	Nomor id dari <i>number</i>
2	Number	Varchar(20)	NOT NULL	Daftar Nomor ponsel yang boleh melakukan <i>remote</i>
3	Name	Varchar(30)	NOT NULL	Identifikasi nama pemilik nomor

4.3.5 Perancangan Arsitektur Fisik

Secara umum arsitektur fisik serta komponen sistem dapat dilihat pada gambar 4.6.



Gambar 4.8. Arsitektur Sistem RemoteServ

Dari gambar diatas maka dibutuhkan beberapa arsitektur fisik yang akan digunakan pada sistem, yaitu :

1. Komputer *Server*.
2. Ponsel Sebagai *SMS gateway*

3. Kabel data penghubung antara ponsel SMS *gateway* dan komputer *server*
4. Ponsel untuk admin
5. *Simcard* (kartu) GSM untuk ponsel SMS *gateway*
6. *Simcard* GSM atau CDMA untuk ponsel admin.

1. Kebutuhan Pada Sisi *Server*.

Pada sisi *server*, arsitektur yang dirancang memiliki beberapa komponen didalamnya, yaitu :

1.1. *Gammu*.

Gammu adalah *driver* atau protokol yang menjadi penterjemah SMS yang masuk kedalam sistem komputer. *Gammu* menguasai setiap fungsi-fungsi pada ponsel, sehingga perangkat komputer dapat membaca tipe data dari ponsel tersebut. Sementara koneksi dari ponsel menuju perangkat komputer *server* menggunakan Kabel *interface* USB.

1.2. *Database* (Mysql).

Setelah SMS diterima oleh *gammu*, maka SMS tersebut diletakkan atau diarahkan menuju *database*, dalam hal ini *database* yang digunakan adalah Mysql. Data yang ada dalam *database* tersebut dijadikan sebagai data yang nantinya akan diolah menjadi perintah *shell* atau sesuai dengan kebutuhan.

1.3. *RemoteServ*.

RemoteServ ini adalah aplikasi yang mengolah data yang terdapat pada *database* menjadi perintah-perintah yang akan di eksekusi oleh *server*. Aplikasi ini juga yang mem-*forward* hasil eksekusi ke *gammu*, dan

seterusnya akan dikirimkan ke admin melalui perangkat SMS *gateway*. Aplikasi ini selalu dalam keadaan aktif dan melakukan pengecekan secara *looping* terhadap *database*.

2. Kebutuhan Pada Sisi SMS Gateway.

Arsitektur SMS *gateway* berupa ponsel GSM yang terhubung langsung ke komputer *server* dengan menggunakan kabel konektor. Ponsel ini membutuhkan sebuah *simcard* (kartu) operator GSM sebagai penyedia layanan penghubung atau komunikasi antara admin dan ponsel SMS *gateway*. Komponen yang dibutuhkan pada ponsel SMS *gateway* ini hanya aplikasi SMS yang mampu mengirim dan menerima SMS.

3. Kebutuhan Pada Sisi Admin.

Yang mengimplementasikan aplikasi RemoteServ pada *server* adalah admin, admin inilah pengguna sekaligus administrator dari sistem RemoteServ. Ponsel admin juga membutuhkan sebuah *simcard* (kartu) GSM untuk dapat berkomunikasi dengan *server* melalui SMS *gateway*. Komponen atau aplikasi yang digunakan pada sisi admin sama halnya dengan komponen pada sisi SMS *gateway* yaitu aplikasi SMS yang mampu mengirim dan menerima SMS dengan memanfaatkan layanan operator GSM.

Namun pada sisi admin dapat juga digunakan *simcard* (kartu) CDMA dengan menggunakan layanan operator CDMA. Hal ini tidak akan mengubah

sistem tapi hanya mengubah proses atau lalu lintas pengiriman data SMS yang dilakukan antar lintas operator.

BAB V

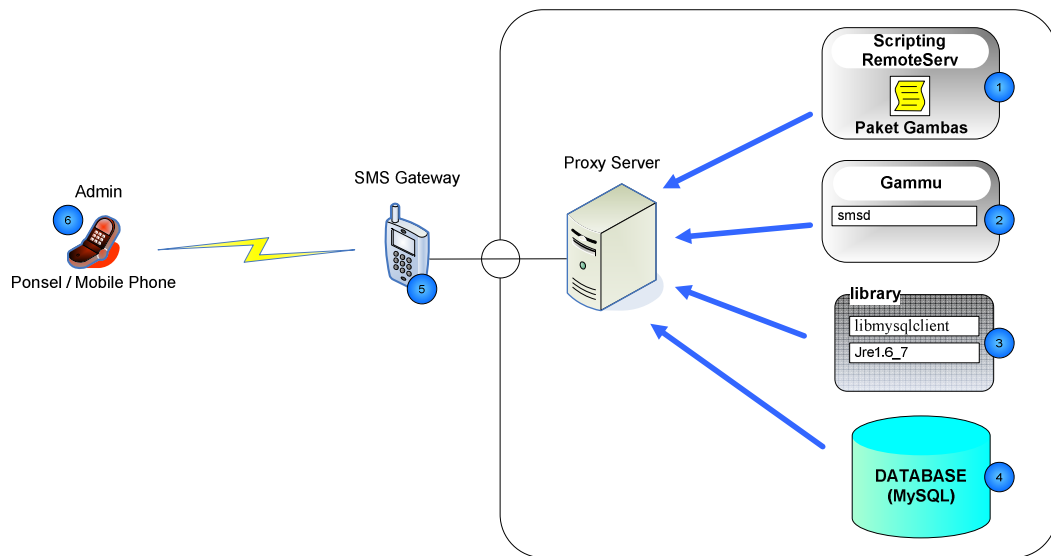
IMPLEMENTASI DAN PENGUJIAN

Implementasi dan pengujian merupakan tahap yang dilakukan setelah tahap analisa dan perancangan selesai.

5.1 Implementasi

Implementasi sistem meliputi tahapan konfigurasi SMS *gateway*, penulisan program dan uji aplikasi. Pada tahapan ini, aplikasi yang telah selesai akan dioperasikan dan dilakukan pengujian untuk melihat sejauh mana aplikasi yang dibuat dapat menghasilkan tujuan yang ingin dicapai.

Implementasi sistem ini berupa konfigurasi SMS *gateway* dan dibuatnya suatu program atau aplikasi *remote server*. Aplikasi dan SMS *gateway* ini nantinya akan ditempatkan pada sisi *server*, pembuatan aplikasi ini menggunakan bahasa pemrograman *gambas*, sementara *driver* ponsel SMS *gateway* yang digunakan untuk pendeteksi ponsel pada sistem linux adalah *gammu*, selain itu digunakan *database* sebagai penyimpanan data SMS dan nomor ponsel admin, dalam hal ini menggunakan MySQL. Sementara pada sisi admin yang menggunakan sistem tidak membutuhkan aplikasi tambahan ataupun program khusus, yang digunakan hanya fitur SMS yang tersedia pada setiap ponsel. Deskripsi dari implementasi sistem dapat dilihat dari Gambar 5.1.



Gambar 5.1 Deskripsi Implementasi Sistem

Objek-objek yang berperan pada implementasi sistem meliputi :

1. *Scripting* atau pengkodean aplikasi RemoteServ menggunakan bahasa pemrograman gambah.
2. Gammu sebagai *driver* atau pendeteksi ponsel pada sistem operasi linux. Pada gammu diinstall juga paket smsd (SMS *daemon*), sebagai daemon pada gammu.
3. Menggunakan aplikasi penunjang CharReplacer.jar untuk mengkonversikan karakter *single bacquote* ke *single quote*.
4. *Library* sebagai *runtime* program. *Library* yang di-include-kan pada server yaitu libmysqlclient dan jre1.6. Masing-masing berfungsi, libmysqlclient sebagai *library* untuk mengkoneksikan gammu ke *database* MySQL, dan jre1.6_7 sebagai *runtime* java untuk *script converter single backquote* (`) menjadi *single quote* (').

5. *Database* (MySQL) sebagai tempat penyimpanan data yang diterima oleh gammu, selanjutnya data tersebut akan dieksekusi sesuai dengan format datanya.
6. Ponsel SMS *gateway* yang menerima SMS yang dikirimkan oleh admin. Pada ponsel ini hanya menggunakan fitur SMS yang telah disediakan pada setiap ponsel.
7. Ponsel Admin untuk mengirim SMS. Admin hanya menggunakan fitur SMS yang telah ada pada ponsel.

5.1.1. Lingkungan Implementasi

Lingkungan implementasi terdiri dari lingkungan perangkat keras atau fisik fisik dan lingkungan perangkat lunak atau aplikasi penunjang yang digunakan untuk membuat aplikasi RemoteServ.

Pada Lingkungan perangkat keras dalam implementasi memiliki spesifikasi sebagai berikut :

Perangkat Keras

1. *Processor* : Intel Core 2 Duo 1.83 GHz
2. RAM : 1 G
3. *Harddisk* : 120 G
4. Ponsel sebagai SMS *Gateway* pada *server*.
5. Kabel *interface* USB penghubung antara *server* ke ponsel SMS *gateway*.
6. Simcard (kartu) GSM.

Sementara implementasi lingkungan perangkat lunak, aplikasi yang digunakan memiliki spesifikasi sebagai berikut :

1. Sistem operasi : Linux Ubuntu 9.10
2. Gambas Version : 5.13
3. MySQL Version : 5.1
4. Gammu version : 1.24.0
5. Jre Version : 1.6

5.1.2. Batasan Implementasi

Batasan implementasi dari tugas akhir ini adalah :

1. Menggunakan bahasa pemrograman gambas sebagai pengkodean.
2. *Driver* untuk pendeteksi ponsel SMS *gateway* pada *server* menggunakan *gammu*.
3. Ponsel yang menjadi SMS *gateway* adalah ponsel Siemens M55 GSM.
4. Menggunakan aplikasi penunjang yaitu *CharReplacer.jar* untuk mengkonversikan karakter *single backquote* (`) menjadi *single quote* (').
5. Pada implementasi ini, ponsel SMS *gateway* yang digunakan adalah ponsel tipe GSM.
6. Hasil eksekusi sistem (*output*) dikirimkan kembali ke admin maksimal sebanyak 150 karakter.
7. Perintah yang dieksekusi hanya perintah yang tidak mengandung dialog ataupun perintah *editing* file.

5.1.3. Hasil Implementasi

Hasil Implementasi yang telah dilakukan adalah berupa pengecekan koneksi ponsel SMS *gateway* dengan *server* dan implementasi format layanan pada aplikasi RemoteServ. Pengecekan koneksi ponsel SMS *gateway* terhubung dengan *server* dapat dilihat pada gambar 5.2. Sementara untuk implementasi format layanan pada aplikasi, SMS akan dikirimkan ke nomor ponsel SMS *gateway*, dan hasil eksekusi akan dikirim kembali ke nomor pengirim yaitu admin. Gambar hasil implementasi Dapat dilihat pada gambar 5.3 sampai 5.16.

```
root@vic-ubuntu:~# gammu --identify
Manufacturer      : Siemens
Model             : unknown (M55)
Firmware          : 07
IMEI              : 352030000377763
SIM IMSI          : 510107849523826
root@vic-ubuntu:~#
```

Gambar 5.2. Hasil Pengecekan Ponsel SMS *gateway*

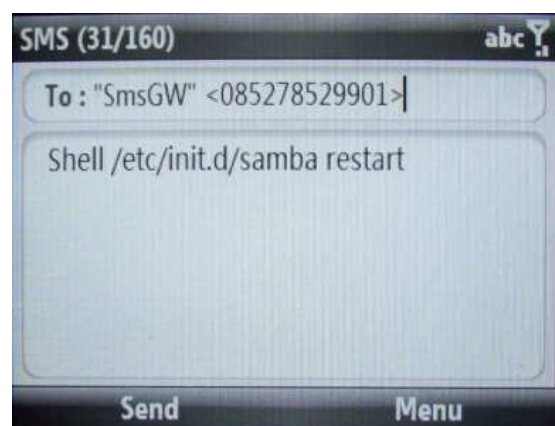


Gambar 5.3 SMS Pengiriman *Help* (Bantuan)

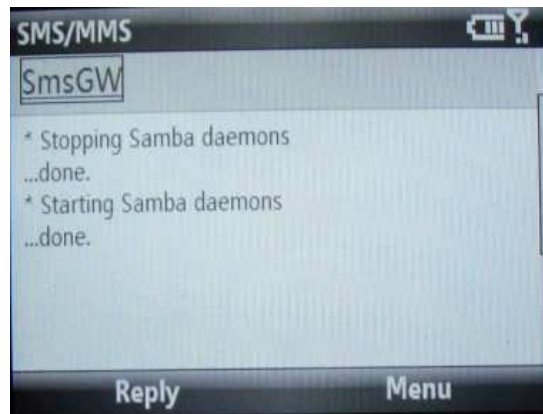


Gambar 5.4 SMS Hasil Balasan SMS *Help*

Pada gambar 5.2 SMS yang dikirimkan oleh admin adalah perintah melihat bantuan (*Help*) atau melihat layanan yang disediakan oleh aplikasi RemoteServ. Pada gambar 5.2 ditampilkan format SMS yang dikirimkan yaitu perintah *Help* dan pada gambar 5.3 merupakan hasil atau balasan yang dikirimkan oleh aplikasi RemoteServ kepada admin.



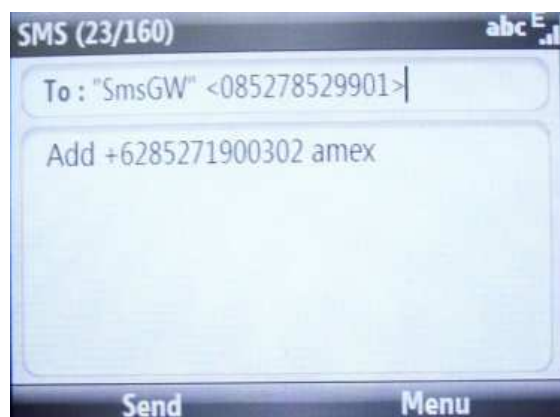
Gambar 5.5 SMS pengiriman Perintah Shell (*Remote Shell*)



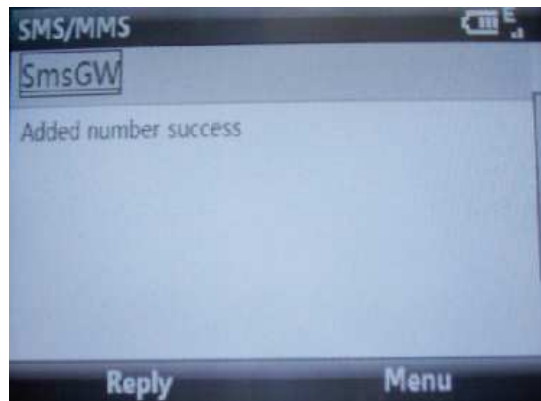
Gambar 5.6 SMS Balasan Perintah Shell (*Remote Shell*)

Pada gambar 5.4 admin melakukan *remote* dengan mengirimkan perintah *shell* untuk melakukan *restart* pada *samba server*. Perintah tersebut akan di eksekusi oleh sistem linux. Pada gambar 5.5 adalah hasil atau *output* dari perintah yang telah di eksekusi oleh sistem linux.

Untuk layanan penambahan nomor yang boleh melakukan *remote* terhadap *server* dapat dilihat pada gambar 5.6 dan 5.7.

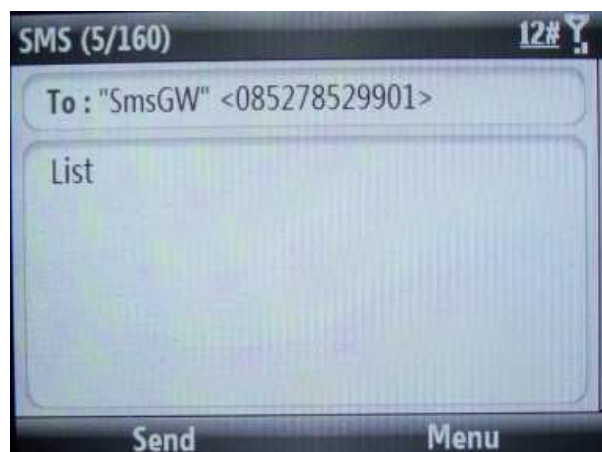


Gambar 5.7 SMS Penambahan Nomor



Gambar 5.8 SMS Balasan Penambahan Nomor Berhasil

Gambar 5.6 merupakan format untuk menambahkan nomor yang boleh melakukan remote terhadap *server*. Jika penambahan berhasil maka aplikasi RemoteServ akan mengkonfirmasi bahwa penambahan berhasil seperti terlihat pada gambar 5.7. Untuk melihat seluruh nomor yang terdaftar dalam *database* dapat dengan melakukan SMS dengan perintah 'List'. Seperti pada gambar 5.8 berikut.



Gambar 5.9 SMS Pengiriman *List* Nomor

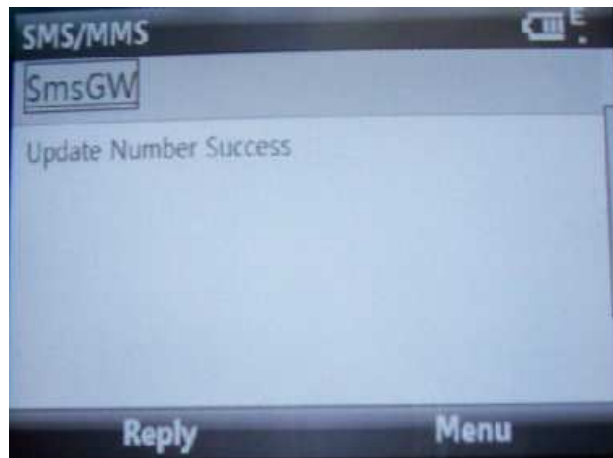


Gambar 5.10 SMS Balasan *List* Nomor

Hasil eksekusi dari perintah 'List' dapat dilihat pada gambar 5.9. pada gambar ditampilkan seluruh nomor yang terdaftar beserta nama identitas dari nomor. Jika penambahan nomor untuk melakukan *remote* terdapat kesalahan maka nomor yang sudah terdaftar tersebut dapat diubah menjadi nomor baru seperti yang diinginkan. Perintah perubahan dapat dilihat pada gambar 5.10.

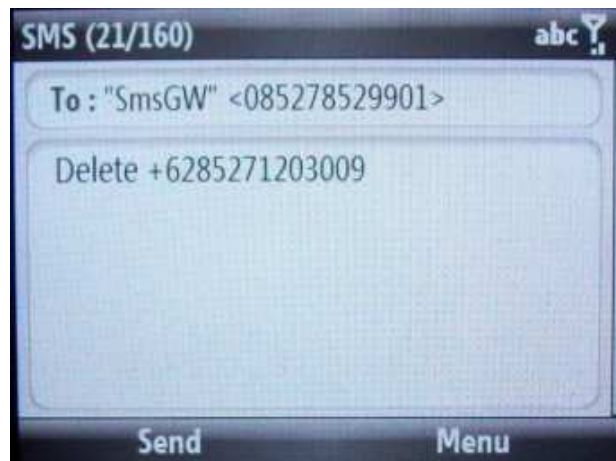


Gambar 5.11 SMS Pengiriman *Update* Nomor

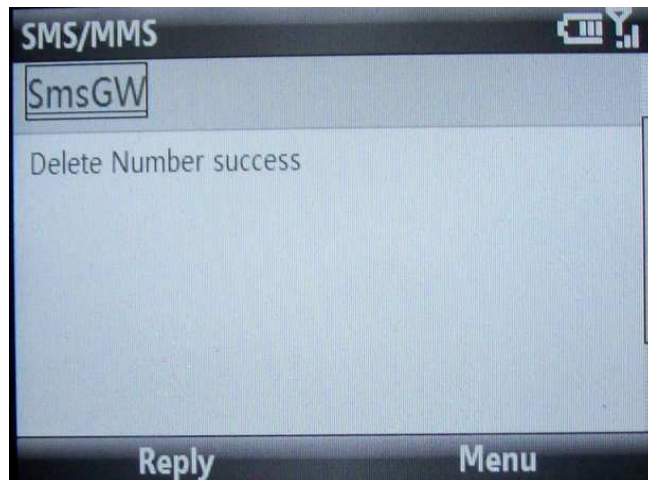


Gambar 5.12 SMS Balasan *Update* Nomor

Jika berhasil maka aplikasi akan memberikan balasan berupa pemberitahuan bahwa nomor berhasil diubah, seperti terlihat pada gambar 5.11. Untuk penghapusan nomor format SMS dapat dilihat pada gambar 5.12.



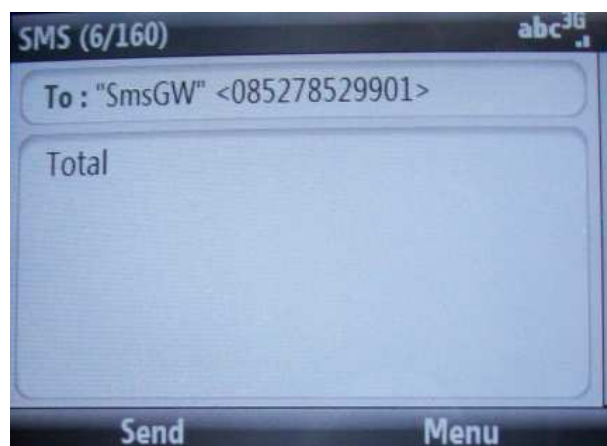
Gambar 5.13 SMS Penghapusan Nomor



Gambar 5.14 SMS Balasan Penghapusan Nomor

Gambar 5.12 merupakan perintah penghapusan nomor pada nomor yang sudah didaftarkan sebelumnya oleh admin. Sementara gambar 5.13 merupakan konfirmasi bahwa nomor berhasil dihapus.

Dari keseluruhan SMS masuk dan SMS yang terkirim dapat dilihat totalnya dengan format SMS yang terdapat pada gambar 5.14.



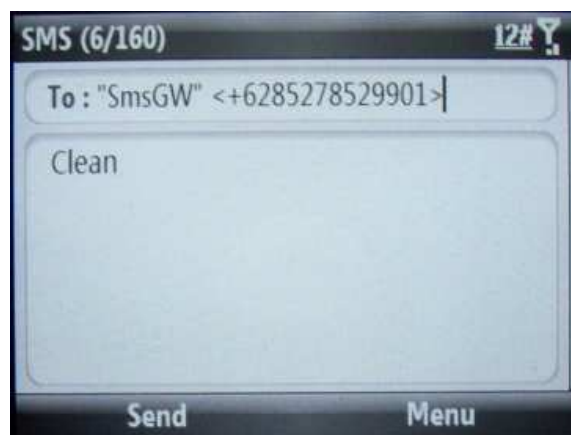
Gambar 5.15 SMS Pengiriman Total SMS masuk dan Keluar



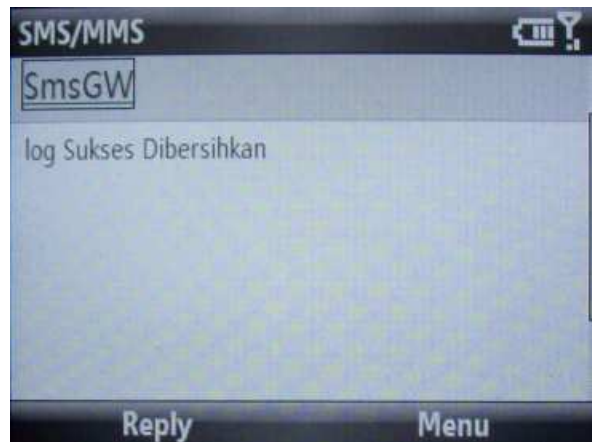
Gambar 5.16 SMS Balasan Total SMS masuk dan Keluar.

Gambar 5.14 adalah format pengiriman SMS dilakukan untuk melihat total SMS yang masuk dan SMS yang terkirim. Sementara balasan atau informasi SMS masuk dan keluar dapat dilihat pada gambar 5.15. pada gambar dapat dilihat total SMS yang masuk adalah 10 dan total SMS yang keluar adalah 11.

Dari keseluruhan SMS yang masuk dan SMS terkirim dapat dihapus dengan perintah 'Clean' seperti pada gambar 5.16.



Gambar 5.17 SMS Pengiriman Penghapusan SMS masuk dan Keluar



Gambar 5.18 SMS Balasan Penghapusan SMS masuk dan Keluar

Pada gambar 5.16 SMS dikirimkan untuk menghapus SMS, hasil dari berhasilnya penghapusan SMS dapat dilihat pada gambar 5.17.

5.2 Pengujian

Pengujian dilakukan untuk mengetahui hasil dari aplikasi Remote Server (RemoteServ) ketika di operasikan. Sebelum aplikasi digunakan sepenuhnya oleh admin jaringan GDLN UR maka perlu dilakukan pengujian yang bertujuan untuk membandingkan hasil analisis dengan keinginan yang akan dicapai.

5.2.1 Identifikasi Pengujian

Identifikasi dan rencana pengujian aplikasi Pencarian data pemilih ganda dapat dilihat pada Tabel 5.1 berikut ini.

Tabel 5.1 Identifikasi Rencana Pengujian

Modul Pengujian	Butir uji	Tingkat pengujian	Jenis pengujian	Jadwal
Perintah shell dan Layanan-layanan yang terdapat pada aplikasi	Normal	Pengujian aplikasi	<i>Black box, User acceptance Test (UAT)</i>	17 Mei 2010

5.2.2 Lingkungan Pengujian

Lingkungan pengujian yang digunakan adalah sebagai berikut:

Perangkat Keras

1. *Processor* : Intel Pentium Dual Core 1.80 GHz
2. *Memory* : 1024 MB
3. *Harddisk* : 120 GB

Perangkat Lunak

1. Sistem Operasi : Linux Ubuntu Server 9.10
2. Bahasa Pemograman : Gambas
3. *Database* : MySQL

5.2.3 Pengujian Layanan Pada Aplikasi

Pengujian disesuaikan dengan banyaknya layanan yang disediakan pada aplikasi. Tapi khusus pada layanan *remote shell* (perintah *shell*), akan dilakukan pengujian dengan sepuluh *sample* perintah yang biasa dilakukan admin. Pengujian ini dilakukan dengan dua teknologi yang berbeda pada sisi admin yaitu menggunakan teknologi GSM dan CDMA. Sebelum melakukan pengujian

layanan pada aplikasi, akan dilakukan pengujian koneksi ponsel SMS *gateway* dengan *server*, pengujian SMS *gateway* dapat dilihat pada tabel 5.2.

Tahapan dan deskripsi pengujian *black box* layanan pada aplikasi dengan menggunakan ponsel GSM dijelaskan dari tabel 5.3 sampai tabel 5.9. Dan pengujian *black box* layanan aplikasi dengan ponsel CDMA dijelaskan dari tabel 5.10 sampai table 5.16.

Tabel 5.2 Pengujian *Black Box* Koneksi SMS *gateway* Terhadap *Server*

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian koneksi ponsel SMS gateway terhadap server	Inputkan perintah pendeteksi ponsel ke terminal	Gammu --identify	Manufacturer: Siemens Model: unknown (M55) Firmware: 07 IMEI: 3520300003777763 SIM IMSI: 5101078495223826	Manufacturer: Siemens Model: unknown (M55) Firmware: 07 IMEI: 3520300003777763 SIM IMSI: 5101078495223826	Berhasil mendeteksi ponsel SMS gateway

Tabel 5.3 Pengujian *Black Box* Perintah *Remote Shell* (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Perintah shell untuk melihat user yang aktif.	Inputkan melalui SMS perintah shell.	Shell w	0.08, 0.02 USER TTY FROM LOGIN@	0.08, 0.02 USER TTY FROM LOGIN@	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
				IDLE JCPU PCPU WHAT root pts/1 192.168.0.1 5 14:48 0.00s 0.20s - bash	IDLE JCPU PCPU WHAT root pts/1 192.168.0.1 5 14:48 0.00s 0.20s - bash	
2.	Pengujian Perintah shell untuk melakukan restart services samba.	Inputkan melalui SMS perintah shell.	Shell /etc/init.d/samba restart	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	Berhasil di eksekusi system linux
3.	Pengujian Remote shell/ Perintah shell.	Inputkan melalui SMS perintah shell.	Shell cat /etc/sudoers	s, so you might need to move # it further down) # %sudo ALL=NOPASSWD: ALL # Members of the admin group may gain root privileges %admin ALL=(ALL) ALL	s, so you might need to move # it further down) # %sudo ALL=NOPASSWD: ALL # Members of the admin group may gain root privileges %admin ALL=(ALL) ALL	Berhasil di eksekusi system linux
4.	Pengujian Perintah shell untuk melakukan cek koneksi ke situs	Inputkan melalui SMS perintah shell.	Shell ping -c 2 google.com	google.com ping statistics --- 2 packets transmitted, 2 received,	google.com ping statistics --- 2 packets transmitted, 2 received,	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
	google.com.			0% packet loss, time 1615ms rtt min/avg/max/mdev = 855.223/878.499/901.776/23.295 ms	0% packet loss, time 1615ms rtt min/avg/max/mdev = 855.223/878.499/901.776/23.295 ms	
5.	Pengujian Perintah shell untuk log user yang mencoba login.	Inputkan melalui SMS perintah shell.	Shell cat /var/log/aut h.log	m_unix(cron:session): session opened for user root by (uid=0) May 21 16:40:01 ubuntu CRON[3844]: pam_unix(cron:session) : session closed for user root	m_unix(cron:session): session opened for user root by (uid=0) May 21 16:40:01 ubuntu CRON[3844]: pam_unix(cron:session) : session closed for user root	Berhasil di eksekusi system linux
6.	Pengujian Perintah shell untuk melihat DNS yang terkoneksi.	Inputkan melalui SMS perintah shell.	Shell cat /etc/resolv.conf	nameserver 203.130.193.74 nameserver 202.134.0.155 #nameserver 208.67.220.220	nameserver 203.130.193.74 nameserver 202.134.0.155 #nameserver 208.67.220.220	Berhasil di eksekusi system linux
7.	Pengujian Perintah shell untuk melakukan cek koneksi dengan melakukan	Inputkan melalui SMS perintah shell.	Shell traceroute google.com	5 209.85.254.166 (209.85.254.166) 85.103 ms 85.911 ms 86.613	5 209.85.254.166 (209.85.254.166) 85.103 ms 85.911 ms 86.613	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
	tracert ke situs google.			ms 6 sin01s01-in- f104.1e100. net (216.239.61. 104) 80.311 ms 81.035 ms 81.898 ms	ms 6 sin01s01-in- f104.1e100. net (216.239.61. 104) 80.311 ms 81.035 ms 81.898 ms	
8.	Pengujian Perintah shell melihat log acces dari squid.	Inputkan melalui SMS perintah shell.	Shell tail /var/log/squid/access.log	9.251 te 1274439672 .896 937 192.168.0.2 TCP_MISS/ 200 1200 GET http://update s.installshiel d.com/Versi on.asp? - DIRECT/64 .14.29.252 text/html	9.251 te 1274439672 .896 937 192.168.0.2 TCP_MISS/ 200 1200 GET http://update s.installshiel d.com/Versi on.asp? - DIRECT/64 .14.29.252 text/html	Berhasil di eksekusi system linux
9.	Pengujian Perintah shell untuk melakukan restart services networking.	Inputkan melalui SMS perintah shell.	Shell /etc/init.d/networking restart	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	Berhasil di eksekusi system linux
10.	Pengujian Perintah shell untuk melakukan restart komputer.	Inputkan melalui SMS perintah shell.	Shell shutdown -r 0	Perintah shutdown sukses di eksekusi	Komputer melakukan restart	Berhasil di eksekusi system linux

Tabel 5.4 Pengujian *Black Box* Penambahan Nomor Admin (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6281276908862 pakabu	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor
2.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6285278579439 maya	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor
3.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6285767627808 nedo	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor

Tabel 5.5 Pengujian *Black Box* Meliaht *List* Nomor Yang Terdaftar (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian List Nomor yang sudah terdaftar	Inputkan melalui SMS perintah list nomor	List	Menampilkan Seluruh List Number yang terdaftar	+6285278270116 vic +6281276908862 pakabu +6285278579439 maya +6285767627808 nedo	Berhasil Menampilkan list nmer

Tabel 5.6 Pengujian *Black Box* Mengubah Nomor Yang Terdaftar (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6285278579439 +6285278787878 nurul	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor
2.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6281276908862 +6281365222737 iwan	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor
3.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6285767627808 +6285767676767 desnedi	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor

Tabel 5.7 Pengujian *Black Box* Menghapus Nomor Yang Terdaftar (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +6285278787878	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor
2.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +6281365222737	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
3.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +628576767 6767	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor

Tabel 5.8 Pengujian *Black Box* Melihat Total SMS Masuk dan Keluar (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Melihat informasi total SMS yang masuk dan keluar	Inputkan melalui SMS perintah total	Total	Menampilkan total SMS masuk dan SMS keluar	Total SMS masuk 10 dan SMS keluar 11	Berhasil Menampilkan total SMS masuk dan keluar.

Tabel 5.9 Pengujian *Black Box* Menghapus Seluruh SMS (GSM)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Melihat informasi total SMS yang masuk dan keluar	Inputkan melalui SMS perintah total	Clean	Informasi berupa : Log berhasil dibersihkan	Log berhasil dibersihkan	Berhasil Menghapus seluruh SMS masuk dan SMS keluar

Tabel 5.10 sampai tabel 5.16 akan menjelaskan hasil pengujian dengan menggunakan ponsel CDMA.

Tabel 5.10 Pengujian *Black Box* Perintah *Remote Shell* (ponsel CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Perintah shell untuk melihat user yang aktif.	Inputkan melalui SMS perintah shell.	Shell w	0.03, 0.01 USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT root pts/0 192.168.0.1 5 09:34 0.00s 0.05s 0.00s w	0.03, 0.01 USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT root pts/0 192.168.0.1 5 09:34 0.00s 0.05s 0.00s w	Berhasil di eksekusi system linux
2.	Pengujian Perintah shell untuk melakukan restart services samba.	Inputkan melalui SMS perintah shell.	Shell /etc/init.d/samba restart	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	Berhasil di eksekusi system linux
3.	Pengujian Remote shell/ Perintah shell.	Inputkan melalui SMS perintah shell.	Shell cat /etc/sudoers	s, so you might need to move # it further down) # %sudo ALL=NOP ASSWD: ALL # Members of the admin group may gain root privileges %admin ALL=(ALL) ALL	s, so you might need to move # it further down) # %sudo ALL=NOP ASSWD: ALL # Members of the admin group may gain root privileges %admin ALL=(ALL) ALL	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
4.	Pengujian Perintah shell untuk melakukan cek koneksi ke situs google.com.	Inputkan melalui SMS perintah shell.	Shell ping -c 10 google.com	le.com ping statistics --- 10 packets transmitted, 10 received, 0% packet loss, time 9016ms rtt min/avg/max/mdev = 370.392/138.4847/2117.785/578.206 ms, pipe 3	le.com ping statistics --- 10 packets transmitted, 10 received, 0% packet loss, time 9016ms rtt min/avg/max/mdev = 370.392/138.4847/2117.785/578.206 ms, pipe 3	Berhasil di eksekusi system linux
5.	Pengujian Perintah shell untuk log user yang mencoba login.	Inputkan melalui SMS perintah shell.	Shell cat /var/log/auth.log	:session): session opened for user root by (uid=0) Jun 20 10:10:01 ubuntu CRON[2774]: pam_unix(cron:session) : session opened for user root by (uid=0)	:session): session opened for user root by (uid=0) Jun 20 10:10:01 ubuntu CRON[2774]: pam_unix(cron:session) : session opened for user root by (uid=0)	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
6.	Pengujian Perintah shell untuk melihat DNS yang terkoneksi.	Inputkan melalui SMS perintah shell.	Shell <code>cat /etc/resolv.conf</code>	#nameserver 203.130.193.74 #nameserver 202.134.0.155 #nameserver 208.67.220.220 nameserver 192.168.6.254	#nameserver 203.130.193.74 #nameserver 202.134.0.155 #nameserver 208.67.220.220 nameserver 192.168.6.254	Berhasil di eksekusi system linux
7.	Pengujian Perintah shell untuk melakukan cek koneksi dengan melakukan tracert ke situs google.	Inputkan melalui SMS perintah shell.	Shell <code>tracert google.com</code>	5 209.85.254.166 (209.85.254.166) 85.103 ms 85.911 ms 86.613 ms 6 sin01s01-in-f104.1e100.net (216.239.61.104) 80.311 ms 81.035 ms 81.898 ms	5 209.85.254.166 (209.85.254.166) 85.103 ms 85.911 ms 86.613 ms 6 sin01s01-in-f104.1e100.net (216.239.61.104) 80.311 ms 81.035 ms 81.898 ms	Berhasil di eksekusi system linux
8.	Pengujian Perintah shell melihat log acces dari squid.	Inputkan melalui SMS perintah shell.	Shell <code>tail /var/log/squid/access.log</code>	8.184 1689 192.168.0.3 TCP_MISS/ 200 6505 GET http://www.facebook.com/ajax/stream/profile.php? - DIRECT/12	8.184 1689 192.168.0.3 TCP_MISS/ 200 6505 GET http://www.facebook.com/ajax/stream/profile.php? - DIRECT/12	Berhasil di eksekusi system linux

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
				5.160.16.96 application/x-javascript	5.160.16.96 application/x-javascript	
9.	Pengujian Perintah shell untuk melakukan restart services networking.	Inputkan melalui SMS perintah shell.	Shell /etc/init.d/networking restart	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	* Stopping Samba daemons ...done. * Starting Samba daemons ...done.	Berhasil di eksekusi system linux
10.	Pengujian Perintah shell untuk melakukan restart komputer.	Inputkan melalui SMS perintah shell.	Shell shutdown -r 0	Perintah shutdown sukses di eksekusi	Komputer melakukan restart	Berhasil di eksekusi system linux

Tabel 5.11 Pengujian *Black Box* Penambahan Nomor Admin (CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6285278270116 vic	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor
2.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6281371340576 noname	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
3.	Pengujian penambahan nomor yang boleh melakukan remote	Inputkan melalui SMS penambahan Number	Add +6285278579439 maya	Informasi berupa : Added Number success	Added Number success	Berhasil Menambahkan nomor

Tabel 5.12 Pengujian *Black Box* Melihat *List* Nomor Yang Terdaftar (CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian List Nomor yang sudah terdaftar	Inputkan melalui SMS perintah list nomor	List	Menampilkan Seluruh List Number yang terdaftar	+627614812198 flexi +6285278270116 vic +6281371340576 noname +6285278579439 maya	Berhasil Menampilkan list nomor

Tabel 5.13 Pengujian *Black Box* Mengubah Nomor Yang Terdaftar (CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6285278270116 +6285278270611 ridwan	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor
2.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6281371340576 +6281371341111 Fulan	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
2.	Pengujian melakukan perubahan nomor yang sudah terdaftar	Inputkan melalui SMS perubahan Nomor	Update +6285278579439 +6285278787878 nurul	Informasi berupa : Update Number success	Update Number success	Berhasil mengubah nomor

Tabel 5.14 Pengujian *Black Box* Menghapus Nomor Yang Terdaftar (CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +6285278270611	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor
2.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +6281371341111	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor
3.	Pengujian Menghapus nomor yang sudah terdaftar	Inputkan melalui SMS penghapusan Nomor	Delete +6285278787878	Informasi berupa : Delete Number success	Delete Number success	Berhasil menghapus nomor

Tabel 5.15 Pengujian *Black Box* Melihat Total SMS Masuk dan Keluar (CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
----	-----------	--------------------	----------	--------------------------	--------------------	------------

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Melihat informasi total SMS yang masuk dan keluar	Inputkan melalui SMS perintah total	Total	Menampilkan total SMS masuk dan SMS keluar	Total SMS masuk 10 dan SMS keluar 11	Berhasil Menampilkan total SMS masuk dan keluar.

Tabel 5.16 Pengujian *Black Box* Menghapus Seluruh SMS (ponsel CDMA)

No	Deskripsi	Prosedur pengujian	Perintah	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1.	Pengujian Melihat informasi total SMS yang masuk dan keluar	Inputkan melalui SMS perintah total	Clean	Informasi beupa : Log berhasil dibersihkan	Log berhasil dibersihkan	Berhasil Menghapus seluruh SMS masuk dan SMS keluar

Pengujian dengan *User Acceptance Test* (UAT) dilakukan dengan cara admin GDLN melakukan pengujian aplikasi secara langsung serta menjawab pertanyaan-pertanyaan yang terdapat pada form kuisisioner atau angket. Kuisisioner ini dapat dilihat lebih jelas pada lampiran A.

5.2.4 Kesimpulan Pengujian

Dari Hasil pengujian dengan menggunakan *black box* dan UAT, sistem yang sudah dilakukan maka diperoleh kesimpulan sebagai berikut :

1. Dari layanan dan format SMS yang disediakan semuanya berfungsi.

2. Dari sepuluh perintah *remote shell* yang dilakukan terhadap *server* semuanya berhasil dieksekusi oleh sistem linux. Dan menghasilkan *output* seperti yang diharapkan.
3. Pada sisi admin ponsel yang dapat digunakan untuk melakukan remote tidak hanya ponsel GSM tetapi juga CDMA.
4. Berdasarkan Kuisisioner yang telah dilakukan dapat disimpulkan bahwa aplikasi ini mudah digunakan dan sangat membantu admin dalam mengelola *proxy server*.

BAB VI

PENUTUP

Penutup merupakan bagian akhir dari laporan tugas akhir yang berisi kesimpulan dan saran yang dapat dilihat pada uraian berikut.

6.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab-bab sebelumnya dalam Tugas Akhir ini dapat disimpulkan :

1. Aplikasi *Remote console* linux berbasis *SMS gateway* dapat dibangun dan diimplementasikan pada *proxy server* GDLN UR. Aplikasi ini dapat membantu dalam mengelola dan mengontrol *server* di GDLN UR. Tetapi memiliki keterbatasan tampilan, tergantung pada layar ponsel yang digunakan.
2. Aplikasi *remote console* berbasis *SMS gateway* ini memanfaatkan bahasa pemrograman gsmmodem dan Gammu sebagai *driver* ponsel *SMS gateway*. Kedua program ini dapat diimplementasikan dengan baik dan keduanya dapat digabungkan sehingga dapat membangun aplikasi baru yaitu RemoteServ yang dapat digunakan oleh admin dalam melakukan remote terhadap *server*.

6.2 Saran

Ada beberapa hal yang dapat disebutkan sebagai saran untuk lebih menyempurnakan Tugas Akhir ini dimasa yang akan datang antara lain :

1. Aplikasi *remote console* hendaknya dikembangkan juga pada distro atau distribusi linux yang lain, seperti fedora yang cukup banyak diimplementasikan pada tataran *server*.
2. Keamanan *remote console* menggunakan SMS lebih ditingkatkan lagi, tidak hanya dengan *filtering* nomor ponsel yang terdaftar saja tetapi bisa dengan metode pengamanan yang lain.

DAFTAR PUSTAKA

- Ali Bahrami, “*Object Oriented Systems Development*”, McGraw-Hill, Singapore, 1999.
- Eko Indrajit, Richardus, Nurcahyo Prastowo, Rudianto, Dudy. *Administrasi Sistem Linux RedHat*. PT Elex Media Komputindo, Jakarta, 2002.
- Pressman Roger S. “*Rekayasa Perangkat Lunak Pendekatan Praktisi*” Terjemahan LN Harnaningrum, Edisi 2, Andi, Yogyakarta, 2002.
- Yunianto. *Membangun Aplikasi SMS Gateway di Linux*. Dian Rakyat, Jakarta, 2006.
- _____, *Blackbox Testing*, [online] Available
http://www.ayuliana_st.staff.gunadarma.ac.id/downloads/files/12482/Pertemuan+06+-++_BlackBox+Testing_.pdf, diakses pada Mei 2010.
- _____, *CDMA*, [online] Available
<http://id.wikipedia.org/wiki/CDMA>, diakses pada Januari 2010.
- _____, *GAMBAS, Pemrograman "Visual Basic" Ala Linux*. [Online] Available
<http://eras.web.id/artikel-85-gambas-pemrograman-visual-basic-ala-linux2.html>. diakses pada April 18, 2009.
- _____, *Gammu-SMSD-TABLES*, [online] Available
<http://www.wammu.org/phones/>, diakses pada Mei 2010.
- _____, *Global System For Mobile Communications*, [online] Available
http://id.wikipedia.org/wiki/Global_System_for_Mobile_Communications, diakses pada agustus 2009.
- _____, *Pemrograman-Gambas*. [Online] Available
<http://fivembois.wordpress.com/2008/07/25/pemrograman-gambas/>. diakses pada Juli 2009.
- _____, *Pengertian GSM*, [online] Available
<http://www.ceptelefoncunuz.net/pengertian-gsm/>, diakses pada Februari, 2009.

_____, *10 Alasan Kenapa Linux Ubuntu Lebih Baik Dari Windows.*

[Online] Available <http://www.amin888.e-mandiri.com>, diakses pada agustus 2009.