SKRIPSI

PEMBUATAN APLIKASI LAYANAN PENGIRIMAN EMAIL TO SMS DAN SMS TO EMAIL BERBASIS SMS GATEWAY

Disusun Oleh :

DHIKAREZA MULAWARMAN

NIM: 05.12.555

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

LEMBAR PERSETUJUAN

PEMBUATAN APLIKASI LAYANAN PENGIRIMAN EMAIL TO SMS DAN SMS TO EMAIL BERBASIS SMS GATEWAY

SKRIPSI

Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik

Disusun oleh :

DHIKAREZA MULAWARMAN

NIM: 05.12.555

Diperiksa dan Disetujui,

Dosen Pembimbing I	Dosen Pembimbing II
Ir. F. Yudi Limpraptono, MT	**Dr. Eng. Aryuanto Soetedjo, ST.MT**
NIP.Y.1039500274	**NIP.Y.1030800417**

Mengetahui,

Ketua Jurusan T.Elektro S-1

Ir. F. Yudi Limpraptono, MT

NIP.Y.1039500274

JURUSAN TEKNIK ELEKTRO S-1

KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2010

ABSTRAKSI

PEMBUATAN APLIKASI LAYANAN PENGIRIMAN EMAIL TO SMS DAN SMS TO EMAIL BERBASIS SMS GATEWAY

Dhikareza Mulawarman, 05.12.555
Email : reza_cow@yahoo.co.id / reza.cow@gmail.com
Jurusan Teknik Elektro S-1
Konsentrasi Teknik Komputer & Informatika
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang

Dosen Pembimbing I: Ir. F. Yudi Limpraptono, MT
Dosen Pembimbing II: Dr. Eng. Aryuanto Soetedjo, ST. MT

Pada era informasi saat ini bermacam sarana telekomunikasi berkembang dengan sangat pesat dan mudah didapatkan baik itu dari telepon kabel, telepon seluler hingga satelit berkembang dengan sangat pesat sehingga dengan mudah bisa dinikmati. Perkembangan teknologi informasi dan komunikasi yang sangat berpengaruh langsung terhadap kehidupan umat manusia adalah internet dan telepon seluler. Salah satu teknologi internet yang paling sering digunakan yaitu pengiriman surat elektronik (e-mail) yang memungkinkan seseorang mengirim data dari tempat yang jauh dan tidak memakan waktu yang sangat lama serta biaya yang relative murah.

Sedangkan salah satu fasilitas dari telepon seluler yang banyak dipakai saat ini adalah SMS. Hal tersebut dimungkinkan karena berbagai keunggulan dan manfaat yang dimilikinya, diantaranya adalah keunggulan dari segi biaya dan kemudahan penggunaannya di dalam kehidupan sehari-hari.

Dengan berkembangnya teknologi informasi dan komunikasi yang sangat pesat seperti saat ini, orang bisa mengirim dan menerima E-mail yang orang tersebut miliki lewat telepon seluler dengan mudah, semudah mengirim dan menerima SMS. Untuk itu perlu sebuah sistem berbasis SMS yang bisa melakukan transaksi dengan database, yaitu SMS Gateway. SMS Gateway adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistribusikan pesan-pesan yang di-generate lewat sistem informasi melalui media SMS yang di-handle oleh jaringan seluler.

Kata Kunci : *Internet, E-mail, Telepon Seluler, SMS, SMS Gateway*

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiratMu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul "**Pembuatan Aplikasi Layanan Pengiriman Email to SMS dan SMS to Email Berbasis SMS Gateway**" dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1 dan Dosen Pembimbing I.
4. Bapak Yusuf Ismail Nahkoda ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Bapak Dr. Eng. Aryuanto Soetedjo, ST. MT selaku Dosen Pembimbing II.
6. Bapak M. Ibrahim Ashari, ST.MT selaku Dosen Penguji I.
7. Ibu Sandy Nataly M, S.Kom selaku Dosen Penguji II.

8. Ayah dan Ibu serta saudara-saudara kami yang selalu memberikan do'a restu, dorongan dan semangat.
9. Teman-teman dan semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan bangsa dan negara.

Malang, Februari 2010

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada era informasi yang kita alami saat ini bermacam sarana telekomunikasi berkembang dengan sangat pesat dan dengan mudah kita dapatkan baik itu dari telepon kabel, telepon seluler hingga satelit berkembang dengan sangat pesatnya sehingga kita dengan mudah bisa menikmatinya. Perkembangan teknologi telekomunikasi ditandai dengan kehadiran sejumlah piranti komunikasi mutakhir dimana setiap orang dapat mengolah, memproduksi serta mengirimkan atau menerima segala bentuk pesan komunikasi dimana saja dan kapan saja seolah-olah tanpa mengenal batasan ruang dan waktu.

Perkembangan teknologi informasi dan komunikasi yang sangat berpengaruh langsung terhadap kehidupan umat manusia adalah internet dan telepon seluler. Internet memberikan kemudahan dalam mengakses informasi yang sangat berharga dan sangat murah yang tidak bergantung pada lokasi dimanapun menghubungkan jarak yang begitu jauh untuk berkomunikasi. Perkembangan teknologi selalu mempunyai peranan yang sangat tinggi dan ikut memberikan arah perkembangan kepada dunia multimedia, pendidikan, hiburan dan lain-lain.

Teknologi informasi adalah bagian dari media yang digunakan untuk menyampaikan pesan pada banyak orang. Kemajuan teknologi yang sangat pesat ini mengubah gaya hidup manusia menjadi serba mudah dan praktis. Teknologi telepon seluler dapat memudahkan seseorang berkomunikasi dengan orang lain dimanapun dia berada. Salah satu fasilitas dari telepon seluler yang banyak

dipakai saat ini adalah SMS. Hal tersebut dimungkinkan karena berbagai keunggulan dan manfaat yang dimilikinya, diantaranya adalah keunggulan dari segi biaya dan kemudahan penggunaannya di dalam kehidupan sehari-hari.

Perkembangan teknologi telekomunikasi memungkinkan orang mengirim dan menerima informasi dari segala penjuru dunia. Melalui jaringan internet segala informasi dapat diperoleh dengan mudah. Salah satu teknologi yang paling sering digunakan yaitu pengiriman surat elektronik (*E-mail*) yang memungkinkan seseorang mengirim data dari tempat yang jauh dan tidak memakan waktu yang sangat lama serta biaya yang relative murah, tidak seperti pengiriman surat dari pos yang dapat memakan waktu berhari-hari dan biaya pengiriman yang mahal.

Dengan berkembangnya teknologi informasi dan komunikasi yang sangat pesat seperti saat ini, kita bisa mengirim dan menerima *E-mail* yang kita miliki lewat telepon seluler dengan mudah, semudah mengirim dan menerima SMS. Untuk itu kita perlu sebuah sistem berbasis SMS yang bisa melakukan transaksi dengan *database*, yaitu SMS *Gateway*. SMS *Gateway* adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistribusikan pesan-pesan yang di-*generate* lewat sistem informasi melalui media SMS yang di-*handle* oleh jaringan seluler.

Berdasarkan uraian diatas penyusun mengambil skripsi dengan judul **"Pembuatan Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* berbasis SMS *Gateway*"**

1.2. Identifikasi Masalah

Berdasarkan uraian di atas maka penulis mengidentifikasikan permasalahan diatas yaitu :

1. Bagaimana membangun aplikasi layanan pengiriman dan penerimaaan *e-mail* melalui SMS.
2. Bagaimana membangun aplikasi yang dapat mempermudah pengguna dalam melakukan komunikasi data dimanapun.

1.3. Maksud dan Tujuan

Berdasarkan uraian diatas, maka penulis bermaksud membuat aplikasi layanan pengiriman *E-mail* to SMS dan SMS to *E-mail* berbasis SMS *Gateway.* Adapun tujuan dari dibangunnya aplikasi ini adalah dapat memudahkan penerimaan dan pengiriman *E-mail.*

1.4. Batasan Masalah

Agar pembahasan yang dilakukan lebih terarah, maka perlu ditentukan batas permasalahan. Adapun batasan masalahnya adalah:

1. Aplikasi yang dibangun dapat mengirim dan menerima *E-mail* pada telepon seluler dalam bentuk SMS.
2. Webmail yang digunakan adalah GMAIL.
3. Orang yang dapat menggunakan aplikasi ini diasumsikan sebagai orang yang sudah terdaftar dengan cara mengetik format SMS tertentu dan dikirimkan ke nomor HP SMS Gateway.

4. Orang yang dapat memanfaatkan layanan aplikasi ini adalah orang yang memiliki nomor HP dan *account E-mail* di www.gmail.com
5. Tidak membahas masalah biaya yang dikeluarkan dalam mengirim dan menerima *E-mail* dengan telepon seluler.
6. Tidak dapat meng-*attachment* gambar ke dalam proses SMS *to E-mail.*
7. Proses SMS *to E-mail* tidak bisa lebih dari 160 karakter.
8. *Body E-mail* yang dapat ditampilkan ke HP pelanggan dalam proses *E-mail to* SMS hanya *body E-mail* yang ber-*text* biasa, berjumlah 160 karakter, dan tidak ber-*attachment* gambar.
9. Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah
   - Borland Delphi 7
   - Microsoft Access

1.5. Metodelogi Penelitian

Metodelogi yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

1.5.1. Tahap Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur.

   Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

b. Observasi.

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

c. *Interview.*

Teknik pengumpulan data dengan mengadakan tanya jawab secara langsung yang ada kaitannya dengan topik yang diambil.

1.5.2. Tahap Pembuatan Perangkat Lunak.

Teknik analisis data dalam pembuatan perangkat lunak menggunakan paradigma perangkat lunak secara *waterfall*, yang meliputi beberapa proses diantaranya:

a. *System / Information Engineering*

Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya kedalam pembentukan perangkat lunak.

b. *Analisis*

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

c. *Design*

Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti oleh user.

d. *Coding*

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang keadalam bahasa pemrograman tertentu.

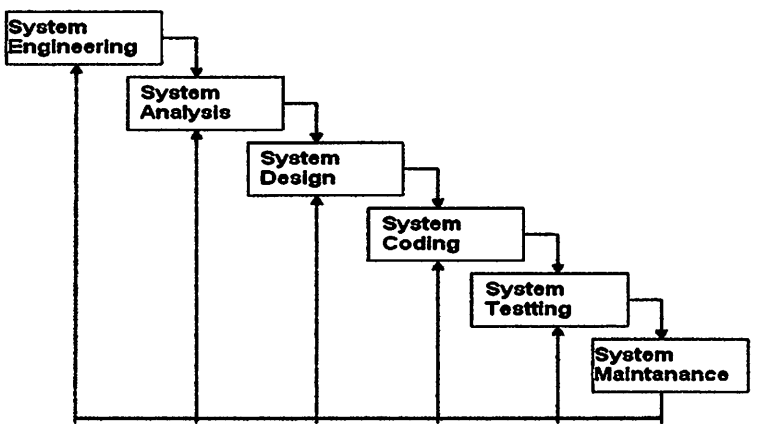
e. *Pengujian*

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.

f. *Maintenance*

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan–perubahan atau penambahan sesuai dengan permintaan pelanggan.

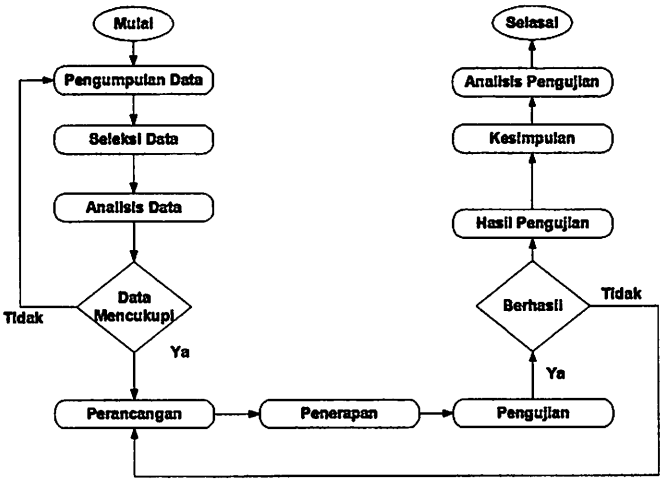
Berikut diagram *waterfall* yang menggambarkan teknik analisis data dalam pembuatan perangkat lunak pada Gambar 1.5.2:



Gambar 1.5.2 *Diagram WaterFall*

1.5.3. Diagram Alir Penelitian

Diagram alir penenlitian berfungsi untuk memberikan panduan dalam melaksanakan proses penelitian, maka penulis membuat suatu diagram alir penelitian yang akan digambarkan pada gambar 1.5.3. berikut ini:



Gambar 1.5.3 Diagram Alir Penelitian Aplikasi layanan pengiriman *E-Mail to SMS* dan *SMS to E-Mail*

1.6. Sistematika Penulisan

Pada penulisan skripsi ini penulis menggunakan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, metodologi dan sistematika penulisan dari aplikasi yang dibuat.

BAB II. LANDASAN TEORI

Berisi tentang landasan teori yang mendukung dalam perencanaan dan pembuatan aplikasi.

BAB III. PERANCANGAN DAN DESAIN SISTEM

Dalam bab ini membahas tentang perancangan sistem dan program mencakup desain sistem. Desain modul dan desain *software*.

BAB IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang implementasi *software* terhadap sistem yang telah dirancang dan berikut pengujian serta analisa dari perancangan sistem yang telah dibuat. Pengujian aplikasi dilakukan untuk mengetahui permasalahan-permasalahan yang mungkin muncul. Analisa dilakukan untuk menyempurnakan aplikasi yang dibuat jika terdapat kekurangan atau kesalahan pada jalannya program maupun dari hasil yang diperoleh.

BAB V. KESIMPULAN DAN SARAN

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran serta petunjuk pengembangan dan penyempurnaan hasil skripsi ini.

BAB II

LANDASAN TEORI

2.1 Komunikasi Data

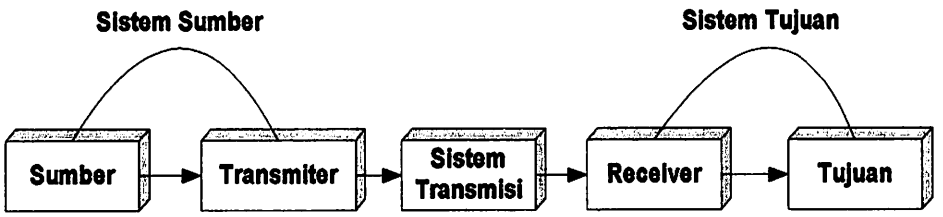
Komunikasi data adalah proses pengiriman informasi diantara dua titik menggunakan kode biner melewati saluran transmisi dan peralatan switching dapat terjadi antara komputer dengan komputer, komputer dengan terminal atau komputer dengan peralatan. Komunikasi data merupakan gabungan dari teknik telekomunikasi dengan teknik pengolahan data.

Adapun tujuan dari komunikasi data adalah sebagai berikut:

1. Memungkinkan pengiriman data dalam jumlah besar, efisien, tanpa kesalahan dan ekonomis dari suatu tempat ketempat yang lain.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (*remote computer use).*
3. Memungkinkan penggunaan komputer secara terpusat maupun secara tersebar sehingga mendukung manajemen dalam hal kontrol, baik desentralisasi ataupun sentralisasi.
4. Mempermudah kemungkinan pengelolaan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
5. Mengurangi waktu untuk pengelolaan data.
6. Mendapatkan data langsung dari sumbernya.
7. Mempercepat penyebarluasan informasi.

2.1.1 Model Komunikasi Data

Komunikasi data berkaitan dengan pertukaran data diantara dua perangkat yang terhubuang secara langsung, memungkinkan adanya pertukaran data antar kedua pihak. Gambar 2.1 menggambarkan proses komunikasi data.



Gambar 2.1 Komunikasi Data

Pada gambar 2.1 terdapat elemen-elemen dalam kunci model tersebut:

1. *Source* (sumber) : Alat ini membangkitkan data sehingga dapat ditransmisikan, contoh telepon, *Personal Computer (PC).*
2. *Transmitter* (pengirim) : Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya. Sebuah *transmitter* cukup memindah dan menandai informasi dengan cara yang sama seperti sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.
3. Sistem transmisi : Berupa jalur transmisi tunggal *(single transmission)* atau jaringan komplek (*complex network*) yang menghubungkan antara sumber dengan tujuan (*destination).*
4. Tujuan (*destination)* : Menangkap data yang dihasilkan oleh *receiver.*

2.1.2 Jaringan Komunikasi Data

Jaringan komunikasi dapat diartikan sebagai suatu sistem yang terbentuk dari interkoneksi fasilitas-fasilitas yang dirancang untuk membawa trafik dari beragam sumber telekomunikasi.

Suatu jaringan terdiri dari *link* dan *node*. Istilah *node* digunakan untuk merepresentasikan sentral, *junction* atau keduanya. Istilah *link* digunakan untuk merepresentasikan kabel, peralatan transmisi, dan sebagainya. Sedangkan trafik adalah informasi yang terdapat di dalam jaringan, yang mengalir melalui *link* dan *node*.

Suatu jaringan komunikasi merupakan sumber daya yang dapat dipakai secara bersamaan (*shared*) oleh sejumlah *end user* untuk berkomunikasi dengan *user* lain yang lokasinya berjauhan. Tidak semua *user* menggunakan jaringan pada waktu yang bersamaan, oleh karena itu merupakan suatu hal yang logis apabila sumber daya jaringan yang sangat penting ini dipakai bersama-sama. Penggunaan sumber daya secara bersamaan ini melahirkan konsep sentral. Berikut beberapa tipe jaringan komunikasi:

a. LAN (Local Area Network)

LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkan bisa mencapai 5 - 10 km. Suatu LAN bisanya bekerja pada kecepatan mulai 10 Mbps - 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya yang dapat digunakan itu misalnya suatu *mainframe, file server,* printer, dan sebagainya.

b. MAN (Metropolitan Area Network)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN - LAN yang lokasinya berjauhan. Jangkauan MAN mencapai 10 km - beberapa ratus km. Suatu MAN biasanya bekerja pada kecepatan 1,5 - 150 Mbps.

c. WAN (Wide Area Network)

WAN dirancang untuk menghubungkan computer - komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari suatu kota ke kota yang lain didalam suatu negara. Cakupan WAN bisa meliputi 100 km - 1.000 km, dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps - 2,4 Gbps. Dalam WAN, biaya peralatan untuk transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan *public.*

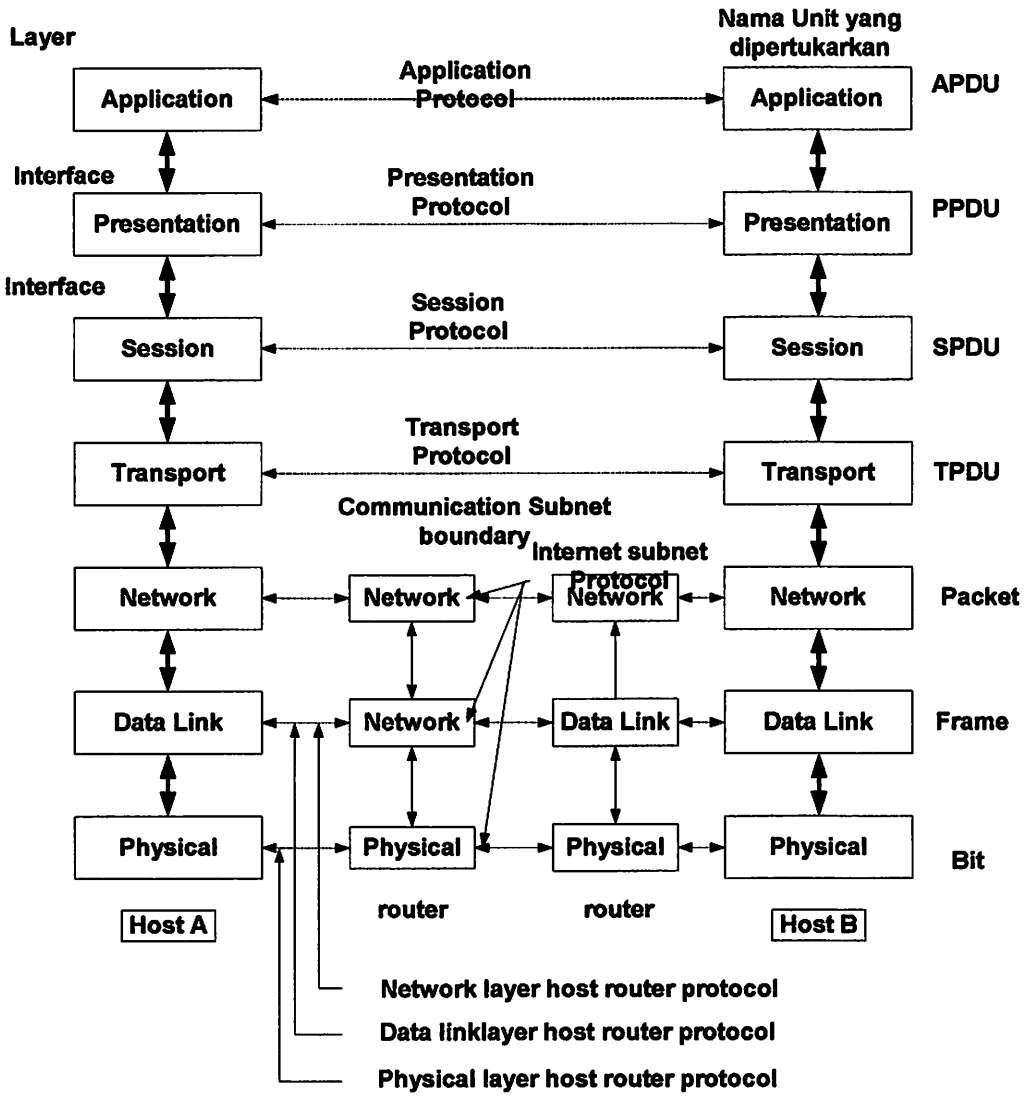
d. GAN (Global Area Network)

GAN merupakan suatau jaringan yang menghubungkan negara - negara diseluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps - 100 Gbps dan cakupannya mencakupi ribuan kilometer.

2.1.3 Arsitektur Protokol

Open system interconnection (OSI), istilah ini mengacu pada standar untuk pertukaran informasi diantara sistem-sistem yang "terbuka" yaitu sistem yang menerapkan standar OSI. Fakta bahwa suatu sistem merupakan sistem yang terbuka tidak berpengaruh terhadap implementasi, teknologi, atau interkoneksi sistem, melainkan berhubungan dengan kesesuaian suatu sistem terhadap standar yang diterapkan. Berdasarkan hal ini, ISO (*organization for Standarization)* telah menspesifikasikan suatu model referensi OSI yang membagi-bagi fungsi komunikasi ke dalam tujuh lapis *(layer).* Model OSI menyediakan secara konseptual kerangka kerja untuk komunikasi. Sebenarnya komunikasi dapat terjadi karena menggunakan protokol komunikasi. Dalam konteks jaringan (komunikasi data), sebuah protokol adalah suatu aturan formal dan kesepakatan yang menentukan bagaimana komputer bertukar informasi melalui sebuah media

jaringan. Sebuah protokol mengimplementsikan salah satu atau lebih *layer-layer* OSI. Setiap layer ditugaskan untuk melaksankan suatu subset dari fungsi komunikasi tertentu yang pada gilirannya diterapkan di dalam suatu *data terminal equipment (DTE)* yang berkomunikasi dengan DTE yang lain. Setiap layer mengandalkan layer yang ada dibawahnya untuk melaksanakan fungsi-fungsi yang lebih primitive serta sekaligus menyediakan layanan untuk mendukung layer yang berada diatasnya. Layer-layer ini dirancang agar setiap perubahan yang terjadi di dalam suatu layer tidak mempengaruhi layer yang lainnya. Model Referensi OSI diperlihatkan pada gambar 2.2



Gambar 2.2 Model Referensi OSI

a. ***Physical Layer***

Lapisan ini bertanggung jawab untuk mengaktifkan dan mengatur Physical interface jaringan komputer. Pada lapisan ini, hubungan antar interface - interface dari perangkat keras diatur seperti hubungan antara DTE dan DCE. Interface yang didefinisikan pada lapisan ini antara lain 10BaseT, 100BaseTX, V35, X.21 dan *High Speed Serial Interface (HSSI).*

b. ***Data Link Layer***

Lapisan ini mengatur topologi jaringan, *error notification* dan *flow control.* Tugas utama *data link layer* adalah sebagai fasilitas transmisi raw data dan mentransformasikan data tersebut ke saluran yang bebas dari kesalahan transmisi.

c. ***Network Layer***

Network Layer berfungsi untuk mengendalikan operasi subnet dengan meneruskan paket - paket dari saru *node* ke *node* yang lain dalam jaringan. Masalah desain yang penting adalah bagaimana cara menentukan route pengiriman paket dari sumber ke tujuannya.

d. ***Transport Layer***

Fungsi utama *Transport layer* adalah menerima data dari *session layer,* memecah data menjadi bagian - bagian yang lebih kecil bila perlu, meneruskan data ke *network layer,* dan manjamin bahwa semua potongan data tersebut bisa tiba disisi lainnya dengan benar. Selain itu, semua hal tersebut harus dilaksanakan secara efisien, dan bertujuan dapat melindungi *layer-layer* bagian atas dari perubahan teknologi *hardware* yang tidak dapat dihindari.

e. ***Session Layer***

Session Layer mengijinkan para pengguna untuk menetapkan *session* dengan pengguna lainnya. Layer ini membuka, mengatur dan menutup suatu *session* antara aplikasi-aplikasi.

f. ***Presentation Layer***

Presentation Layer melakukan fungsi-fungsi tertentu yang diminta untuk menjamin penemuan sebuah penyelesaian umum bagi masalah tertentu. Selain memberikan sarana - sarana pelayanan untuk konversi, format dan enkripsi data,

Presentation Layer juga beberja dengan file berformat ASCII, EBCDIC, JPEG, MPEG, TIFF, PICT, MIDI, dan Quick Time.

g. ***Aplication Layer***

Lapisan ini memberikan saran pelayanan langsung ke *user*, yang berupa aplikasi - aplikasi dan mengadakan komunikasi dari program ke program. Jika kita mencari suatu *file* dari *file server* untuk digunakan sebagai aplikasi pengolah kata, maka proses ini bekerja melaui *layer* ini. Demikian pula ketika melakukan pengiriman *email, browse* ke internet, *chatting,* membuka *telnet session,* atau menjalankan FTP, maka semua proses tersebut dilaksanakan di *layer* ini.

2.1.4 Protokol

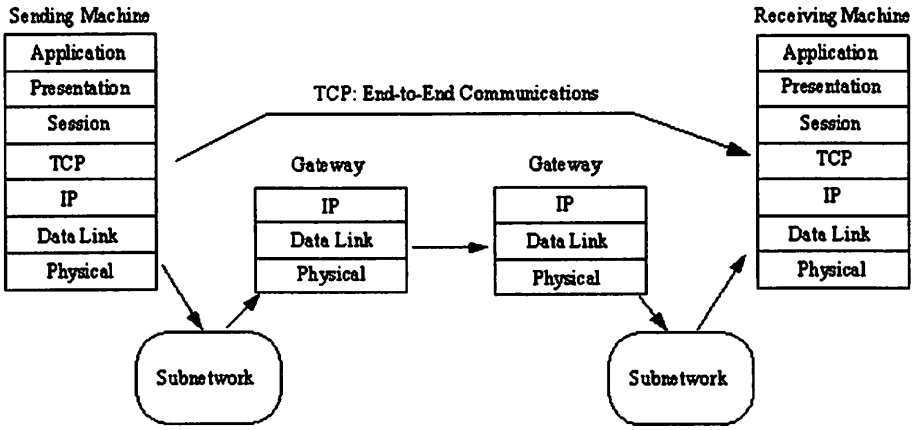
Agar komputer - komputer dalam suatu jaringan dapat berkomunikasi, maka dibutuhkan suatu protokol atau suatu aturan standar komunikasi baik antar komputer maupun antar jaringan komputer, seperti IPX / SPX yang biasa digunakan oleh NOVELL Netware, NETBIOS / NETBEUI biasa digunakan oleh jaringan Microsoft LAN Manager ataupun Microsoft Windows Networking, AppleTalk yang biasa digunakan oleh jaringan komputer Apple Macintosh dan sebagainya. Ada satu protokol yang dikembangkan oleh *DARPA (Defense Advanced Research Projects Agency),* dalam pengembangan dari ARPANET dan juga digunakan oleh jaringan komputer berbasis sistem operasi UNIX yaitu protokol TCP / IP (*Transmision Control Protocol/Internet Protocol).* Protokol TCP / IP ini menjadi standar protokol yang digunakan pada jaringan internet, kerena TCP / IP dikembangkan untuk dapat diterapkan di hampir segala jenis *platform* komputer, biasa dikenal dengan konsep *open system.*

a. ***IP (internet protocol)***

Gambaran yang umum tentang TCP / IP adalah gabungan dari dua protokol komunikasi yang dipakai untuk komunikasi data, TCP kepanjangan dari *transmissioncontrol-protocol* dan IP kepanjangan dari *Internet Protocol*. Kedua protokol tersebut dipakai untuk menyatakan sekelompok protokol yang memiliki kaitan dengan protokol TCP dan IP seperti *User Datagram Protocol (UDP), File Transfer Protocol (FTP), Terminal Emulation Protocol* (TELNET), dan lain-lain. *Internet Protocol (IP)* adalah mekanisme transmisi yang digunakan oleh TCP / IP yang sifatnya *unreliable* dan *connectionless*. Banyak yang mengistilahkan dengan *best effort delivery*, artinya bahwa IP menyediakan *no error checking* atau *tracking*. Jika diperlukan reliabilitas maka IP mesti dipasangkan dengan protokol yang reliable misalya TCP.

b. ***TCP (Transmission Control Protocol)***

TCP merupakan protokol CO *virtual-circuit* yang setara dengan layer transport. Pengiriman data dilakukan byte demi byte secara asinkron dan stream (mengalir), sehingga pesan disampaikan dalam segmen - segmen (atau paket) berukuran tidak tetap, tidak *fixed-size*. Dalam pesan tersebut terdapat identifikasi unik pelayanan host berupa nomor *(socket)*. Penerima juga akan memberikan nomor socket yang unik untuk dikembalikan kepada penerima. Diperoleh (srchost, srcport, dsthost, dstport) yang unik untuk sebuah komunikasi pada layer TCP. Komunikasi TCP / IP diperlihatkan pada gambar 2.3



Gambar 2.3 TCP/IP pada Layer OSI

Berbeda dengan Internet Protokol (IP), TCP mempunyai prinsip kerja seperti "*virtual circuit*" pada jaringan telepon. TCP lebih mementingkan tata cara dan keandalan dalam pengiriman data antara dua komputer dalam jaringan. TCP tidak perduli dengan apa - apa yang dikerjakan oleh IP, yang penting adalah hubungan komunikasi antara dua komputer berjalan dengan baik. Dalam hal ini, TCP mengatur bagaimana cara membuka hubungan komunikasi, jenis aplikasi apa yang akan dilakukan dalam komunikasi tersebut (misalnya mengirim e-mail, transfer file dan sebagainya.). Di samping itu, juga mendeteksi dan mengoreksi jika ada kesalahan data. TCP mengatur seluruh proses koneksi antara satu komputer dengan komputer yang lain dalam sebuah jaringan komputer.

Berikut ini merupakan beberapa aplikasi protokol TCP / IP:

c. SMTP (*Simple Mail Transfer Protocol*)

SMTP merupakan kepanjangan dari *Simple Mail Transfer Protocol*. SMTP adalah suatu protokol aplikasi yang merupakan sistem pengiriman *message* / pesan atau *e-mail*. SMTP dapat mendukung 3 jenis pengiriman pesan:

1. Pengiriman pesan saja kepada satu atau lebih pengguna
2. Pengiriman pesan yang termasuk dalamya teks, suara, video, atau grafik.
3. Pengiriman pesan ke pengguna - pengguna yang diluar jaringan / internet.

Untuk melakukan operasinya SMTP memanfaatkan layanan protokol TCP (lapisan 4) dengan menggunakan alamat port = 25.

SMTP (*Simple Mail Transfer Protocol*) didefinisikan dan digunakan dalam internet untuk mengirimkan *electronic mail (e-mail)*. Alamat pengirim atau penerima menggunakan standard internet termasuk nama dan domain user. Domain akan diganti dengan informasi yang lain sebagai keterhubungan langsung yang telah dibentuk atau terdapat mesin pembalas dalam path. SMTP menggunakan *Domain Name Server* untuk semua alamat. Sekalipun *e-mail* digambarkan akan dapat sampai ketempat tujuan seketika itu juga, kenyataannya tidak sesederhana itu. *E-mail* internet tergantung juaga pada teknologi *"store and forward"*, yang berarti pesan - pesan akan ditahan dan disimpan dalam satu komputer atau lebih sepanjang perjalanan surat tersebut dan diteruskan lagi pada selang waktu yang tetap atau pada saat lalu lintas jaringan memungkinkan. Perlu diketahui bahwa suatu jalur lurus jarang terjadi diantara dua buah komputer mana saja di internet. Meskipun pesan dikirim dengan kecepatan cahaya dari satu titik ketitik yang lain, mungkin surat tersebut harus melewati lusinan komputer, jaringan, router, geteway serta perpotongan rute internet lainnya dan ada kemungkinan surat itu ditahan sementara pada suatu penghentian di sepanjang perjalanannya. Meskipun demikian *e-mail* ini memiliki keandalan yang cukup tinggi dan terhitung sangat cepat, sebab banyak pesan yang dapat mencapai

tempat tujuan dalam hitungan menit, dan hanya sedikit diantaranya yang memakan waktu lebih dari beberapa jam.

d. POP *(Post Office Protocol)*

POP *(Post Office Protocol)* dirancang untuk mengirimkan pesan mail dari server ke mesin klien. User yang mempunyai mail menyimpan dalam mail box pada server. User dalam hal ini adalah POP client mengakses pesan yang disimpan menggunakan POP. Menjawab atau membuat mail baru digerakan oleh client yang menyimpannya melalui SMTP. Mail dikirim dari client yang tidak memerlukan host yang lain untuk mengirimkan *mail* atau mengirim *mail* langsung ke penerima jarak jauh.

2.2 GSM, SMS DAN SMS GATEWAY

2.2.1 GSM (Global System for Mobile)

2.2.1.1 Sejarah GSM

Di Indonesia, liberalisasi bisnis seluler dimulai sejak tahun 1995, saat pemerintah mulai membuka kesempatan kepada swasta untuk berbisnis telepon seluler dengan cara kompetisi penuh. Bisa diperhatikan, bagaimana ketika teknologi *GSM* (*Global System for Mobile*) datang dan menggantikan teknologi seluler generasi pertama yang sudah masuk sebelumnya ke Indonesia seperti *NMT (Nordic Mobile Telephone)* dan AMPS *(Advance Mobile Phone System).*

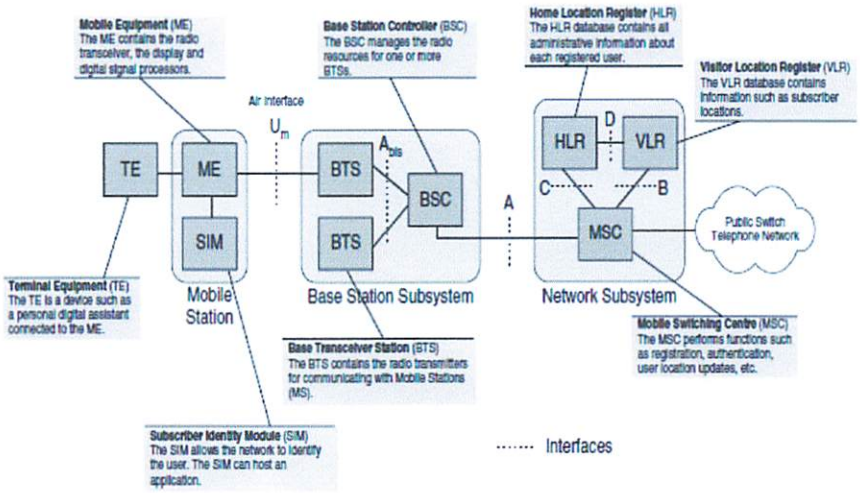
Sekarang, dalam kurun waktu lebih dari satu dekade, teknologi GSM telah menguasai pasar dengan jumlah pelanggan lebih dari jumlah pelanggan telepon tetap. Di tahun 1980-an, teknologi *Global System for Mobile Communication (GSM)* datang ke Indonesia, maka para operator pemakai

teknologi AMPS (*Advanced Mobile Phone System*) menghilang. Lalu, muncul Satelindo sebagai pemenang, yang kemudian disusul oleh Telkomsel.

Dan pada akhirnya teknologi GSM lebih unggul dan berkembang dengan pesat, ini dikarenakan kapasitas jaringan lebih tinggi, serta efisiensi di spektrum frekuensi dari pada teknologi NMT dan AMPS.

2.2.1.2 Arsitektur Jaringan GSM

Jaringan GSM terdiri atas tiga subsistem: *Base Station Subsystem (BSS), Network Subsystem (NSS)* dan *Operation Subsystem (OSS).* OSS tidak dijelaskan lebih lanjut, unsure - unsur BSS dan NSS akan diuraikan lebih lanjut. Gambar 2.4 menunjukkan unsur - unsur yang utama pada arsitektur GSM.



Gambar 2.4 Arsitektur GSM

a. *Mobile Station (MS)*

Mobile Station (MS) adalah perangkat yang mengirim dan menerima signal radio. *MS* dapat berupa mobile handset atau *Personal Digital Assistant (PDA).* MS terdiri dari *Mobile Equipment (ME)* dan *Subscriber Identity Module (SIM).*

ME berisi transceiver radio, display dan Digital Signal Processor. SIM digunakan agar network dapat mengenali user.

b. *Base Transceiver Station (BTS)*

Base transceiver Station (BTS) berfungsi sebagai interface komunikasi semua MS yang aktif dan berada dalam coverage area BTS tersebut. Di dalamnya termasuk modulasi signal, demodulasi, equalize signal dan error coding. Beberapa BTS terhubung pada satu *Base Station Controller (BSC)*. Satu BTS biasanya mampu menghandle 20 - 40 komunikasi serentak.

c. *Base Station Controller (BSC)*

BSC berfungsi mengatur koneksi BTS - BTS yang berada dalam kendalinya. Fungsi tersebut memungkinkan operasi seperti *handover, cell site configuration, management of radio resources* dan menyetel power level dari frekuensi radio BTS. Pada jaringan GSM, BSC mengatur lebih dari 70 BTS.

d. *Mobile Switching Centre (MSC)* dan *Visitor Location Register* (VLR)

Mobile Switching Centre (MSC) melakukan fungsi registrasi, authentikasi, update lokasi user, *billing service* dan sebagai *interface* dengan jaringan lain. Selain itu MSC juga bertanggung jawab untuk *call set-up, release* dan *routing*. *Visitor Location Register (VLR)* berisi informasi dinamis tentang user yang terkoneksi dengan mobile network termasuk lokasi user tersebut. VLR biasanya terintegrasi dengan MSC.

Melalui MSC, mobile network terhubung dengan jaringan lain seperti PSTN (*Public Switched Telephone Network*), *ISDN (Integrated Service Digital Network), CSPDN (Circuit Switched Public Data Network)* dan *PSPDN (Packet Switched Public Data Network).*

e. ***Home Location Register (HLR)***

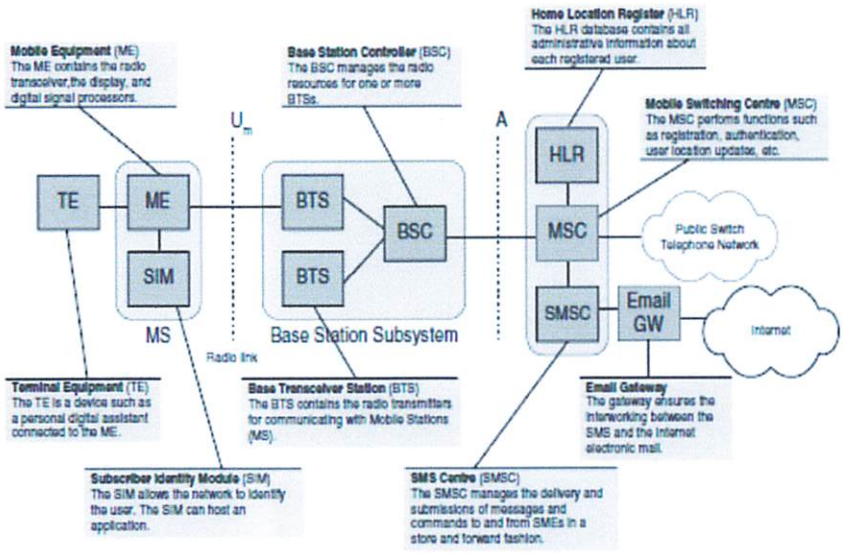
Home Location Register (HLR) adalah elemen jaringan yang berisi detil dari setiap subscriber. Sebuah HLR biasanya mampu mengatur ratusan bahkan ribuan subscriber. Pada jaringan GSM, signaling berbasis pada protokol *Signaling System Number 7* (SS7). Penggunaan SS7 dilengkapi dengan penggunaan protokol *Mobile Application Part (MAP)*. MAP digunakan untuk pertukaran informasi lokasi dan subscriber antara HLR dan elemen jaringan lainnya seperti MSC.

Untuk setiap subscriber, HLR mengatur pemetaan antara *International Mobile Subscriber Identity (IMSI)* dan *Mobile Station ISDN Number (MSISDN)*. Untuk alasan keamanan, IMSI jarang ditransmisikan melalui perantara radio dan hanya dikenali pada jaringan GSM yang ditentukan. IMSI menggunakan format [ITU-E.212]. Tidak seperti IMSI, MSISDN mengidentifikasi subscriber di luar jaringan GSM, MSISDN menggunakan format [ITU-E.164].

2.2.2 SMS *(Short Message Service)*

SMS merupakan fitur GSM yang paling poluler hingga saat ini. Dimulai dengan diperkenalkannya sistem telepon *wireless* / seluler digital memberikan beberapa kelebihan, seperti kemampuan optimasi sistem yang ditunjukkan dengan kemampuan kompresi dan pengkodean data digital. *Handset* yang diperlukan untuk sistem ini juga menjadi sangat simpel, kecil, dan ringan, karena digunakannya *chip* digital untuk *SIM* (*subscriber identification module*). Teknologi *chip* digital juga memungkinkan penambahan fitur - fitur baru sebagai layanan tambahan, seperti *voice mail*, *call waiting*, dan *short message service (SMS)*.

SMS dimaksudkan untuk menjadi alat pertukaran informasi antara dua mobile subscriber. Elemen - elemen utama pada arsitektur SMS terdiri dari *Short Message Entity (SME)*, *SMS Service Centre (SMSC)* dan Email Gateway yang terkoneksi dengan elemen - elemen pada GSM sebagai channel penghantar. Gambar 2.5 memperlihatkan arsitektur SMS pada jaringan GSM.



Gambar 2.5 SMS pada jaringan GSM

a. *Short Message Entity* (SME)

Short Message Entity (SME) adalah elemen yang dapat mengirim atau menerima pesan singkat. SME dapat berupa software aplikasi pada mobile handset, dapat juga berupa perangkat facsimile, perangkat telex, *remote internet server*, dll.

Sebuah SME dapat berupa server yang terkoneksi dengan SMS center secara langsung atau melalui gateway. Dikenal juga *External SME (ESME)* yang merepresentasikan sebuah WAP proxy / server, *Email Gateway* atau *Voice Mail server.*

b. ***SMS Service Centre (SMSC)***

SMS Service Centre (SMSC) memegang peran kunci dalam arsitektur SMS. Fungsi utama SMSC adalah menyampaikan pesan singkat antara SME dengan MS, juga menyimpan dan meneruskan pesan singkat (menyimpan pesan jika penerima SME tidak tersedia). SMSC dapat terintegrasi sebagai bagian dari mobile network (cth: terintegrasi dengan MSC) atau sebagai entitas network independen.

c. ***Email Gateway***

Email Gateway memungkinkan sebuah email beroperasi menjadi SMS dengan interkoneksi SMSC pada internet. Dengan email gateway, pesan dapat dikirim dari sebuah SME menuju sebuah host internet dan sebaliknya. Peran email gateway adalah mengubah format pesan (dari SMS ke email dan sebaliknya) dan merelay pesan antara SMS dan domain internet.

2.2.3 *Basic Features SMS*

SMS mempunyai beberapa *basic* fitur, diantaranya:

a. ***Message Submission and Delivery***

Terdiri dari *message sending* dan *message delivery*. Pada *message sending*, pesan dikirm dari MS ke SMSC, dialamatkan ke SME lain sebagai mobile user lain atau host internet. *Originator* (asal) SME menentukan *validity period* dari pesan tersebut, pesan yang sudah tidak valid lagi akan dihapus oleh SMSC sepanjang pengiriman pesan. Fitur ini dikenal sebagai *Short Message - Mobile Originated (SM-MO).*

Pada message delivery, pesan disampaikan oleh SMSC ke MS. Dikenal sebagai *Short Message Mobile Terminated (SM-MT).* SM-MO dan SM-MT dapat

dikirim atau diterima saat voice call atau koneksi data sedang berlangsung. Pada GSM pesan dikirim pada channel SDCCH / SACCH, pada GPRS pesan dikirim pada channel PDTCH.

b. *Status Report*

SME asal (*originator*) meminta status report pada pengiriman pesan singkat ke SME penerima *(recipient)*. Status report memberikan indikasi pada user asal apakah pesan terkirim dengan sukses atau tidak kepada SME penerima.

c. *Reply Path*

Replay Path dapat diatur oleh SME asal (atau *SMSC serving*) untuk mengindikasikan bahwa SMSC serving dan mampu untuk menghandle secara langsung *reply* dari SME penerima.

d. *Addressing Mode*

Addressing mode menggunakan MSISDN pada format [ITU-E.164]. Email *address* ditentukan oleh IETF pada format [RFC-2822] atau *operator specific numbering.*

e. *Validity Period*

Pesan *originator* dimungkinkan untuk menentukan *validity period* sebuah pesan. *Validity period* ini menentukan batas waktu sebuah pesan harus dikirim sebelum akhirnya dihilangkan oleh jaringan.

2.2.4 *Protocol Layer*

SMS protokol layer terdiri dari 4 layer : *application layer, transfer layer, relay layer* dan *link layer.*

a. ***Application Layer***

Implementasi pada SME dalam bentuk software aplikasi yang mengirim, menerima dan menginterpretasikan isi pesan (seperti : editor pesan, *games*, dll). *Application layer* disebut juga SM-AL (*Short Message Application Layer*).

b. ***Transfer Layer***

Pesan dianggap sebagai serangkaian bilangan oktet yang mengandung informasi seperti panjang pesan, pengirim atau penerima pesan, tanggal penerimaan pesan. Transfer layer disebut juga *SM-TL* (*Short Message Transfer Layer).*

c. ***Relay Layer***

Relay layer mengizinkan pengiriman pesan antar elemen network yang berbeda. Sebuah elemen network menyimpan pesan sementara jika elemen berikutnya dimana pesan akan di *forward* tidak tersedia. Pada relay layer MSC menghandle 2 fungsi *switching* : SMS-GMSC dan SMS-IWMSC.

d. ***SMS-GMSC (SMS-Gateway Mobile Switching Center)***

Merupakan MSC yang dapat menerima SMS dari SMSC, menanyakan info routing ke HLR, dan mengirim pesan ke MSC dari MS penerima.

e. ***SMS-IWMSC (SMS-Interworking Mobile Switching Center)***

Merupakan MSC yang dapat menerima SMS dari jaringan mobile dan mengirimkannya ke SMS yang tepat. SMS-GMSC / SMS-IMSC biasanya diintegrasikan dengan SMSC. Relay layer disebut juga *SM-RL (Short Message Relay Layer).*

f. ***Link Layer***

Link layer mengizinkan pengiriman pesan pada level physical. Untuk tujuan ini, pesan diprotek untuk mengatasi kesalahan low level channel. *Link layer* disebut juga *SM-LL* (*Short Message Link Layer*).

2.2.5 *SMS Protocol Data Unit*

Menggunakan protocol connectionless, terdiri dari 6 PDU:

1. SMS-DELIVER : Mengirimkan pesan dan info terkait dari SMSC ke MS
2. SMS-DELIVER-REPORT : Mengirimkan pesan berisi penyebab kegagalan dari MS ke SMSC
3. SMS-SUBMIT : Mengirimkan pesan dan info terkait dari MS ke SMSC
4. SMS-SUBMIT-REPORT : Mengirimkan pesan berisi penyebab kegagalan dari SMSC ke MS
5. SMS-STATUS-REPORT : Mengirim laporan status
6. SMS-COMMAND : Mengirim perintah yang akan dieksekusi untuk SMS-SUBMIT yang dikirim dari MS ke SMSC.

2.3 SMS Gateway

Satu permasalahan dalam SMS messaging adalah SMSC yang dikembangkan oleh perusahaan berbeda menggunakan protokol komunikasi mereka sendiri. Sebagai contoh, Nokia mempunyai suatu protokol SMSC yaitu CIMD sedangkan SMSC penjual yang lain, CMG, mempunyai suatu protokol SMSC yaitu EMI. Kita tidak bisa menghubungkan dua SMSC jika mereka tidak mendukung suatu SMSC protokol umum. Untuk mengatasi masalah ini, SMS Gateway ditempatkan diantara kedua SMSC. Pada gambar dibawah SMS Gateway bertindak sebagai relay antara kedua SMSC. Tugasnya menerjemahkan protokol SMSC yang satu

dengan protokol SMSC yang lain. Ini dapat digunakan untuk saling behubungan antar SMSC untuk tujuan seperti pertukaran antar operator SMS. Gambar 2.6 memperlihatkan hubungan antar SMS center satu dengan yang lain.

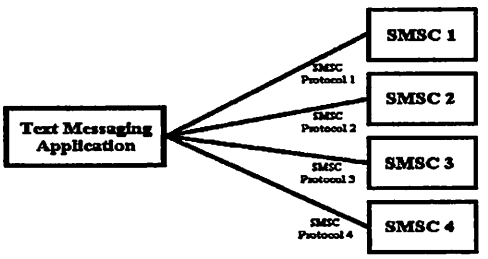
Gambar 2.6 SMS Gateway menghubungkan SMS Center 1 dan SMS Center 2

2.4 Arsitektur Aplikasi SMS

2.4.1 Arsitektur Aplikasi SMS Secara Umum

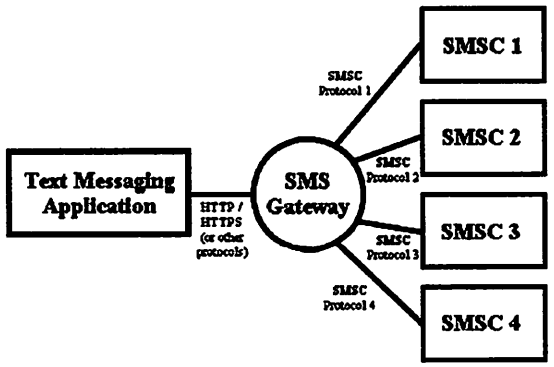
Secara umum terdapat tiga gambaran arsitektur aplikasi SMS, yaitu:

1. Aplikasi pesan SMS yang terhubung ke SMSC tanpa melalui SMS Gateway. Gambar 2.7 memperlihatkan arsitektur SMS yang terhubung ke SMSC tanpa melalui SMS Gateway.



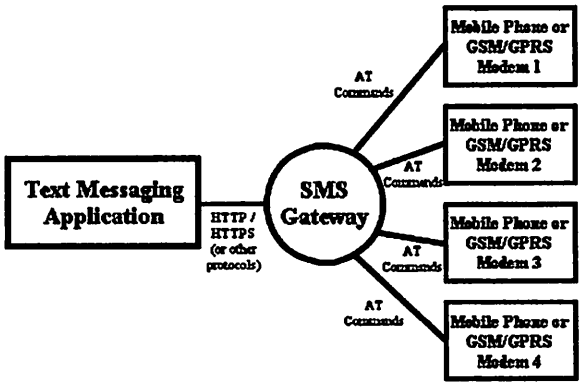
Gambar 2.7 SMSC tanpa melalui SMS Gateway

2. Aplikasi pesan SMS yang terhubung ke SMS Center melalui SMS Gateway. Gambar 2.8 memperlihatkan arsitektur SMS yang terhubung ke SMSC melalui SMS Gateway.



Gambar 2.8 SMS Center melalui SMS Gateway

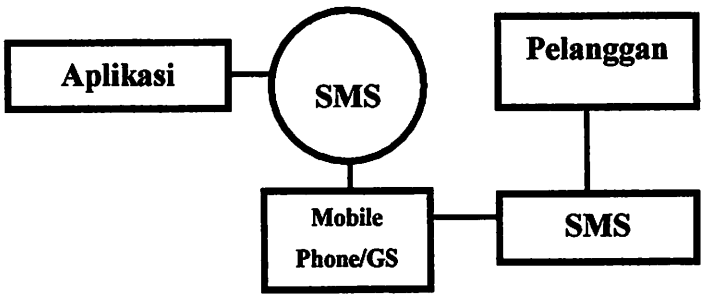
3. Aplikasi pesan SMS terhubung ke mobile phone atau GSM modem melalui SMS Gateway. Gambar 2.9 memperlihatkan arsitektur Aplikasi pesan SMS yang terhubung ke mobile phone atau GSM modem melalui SMS Gateway.



Gambar 2.9 Mobile Phone atau GSM Modem melalui SMS Gateway

2.4.2 Arsitektur Aplikasi SMS *Auto Respon Info*

Gambar 2.10 memperlihatkan arsitektur SMS Auto *Respons Info*.



Gambar 2.10 Arsitektur Aplikasi *SMS Auto Respon Info*

Berikut penjelasan bagaimana perjalanan data dari pelanggan hingga pelanggan (yang meminta informasi) menerima SMS balasan (informasi yang diminta):

1. Pelanggan mengirim SMS ke nomor SMS Gateway untuk meminta informasi.
2. Sebelum diterima SMS Gateway, SMS diterima SMS Center kemudian diteruskan ke GSM modem di SMS Gateway.
3. Aplikasi SMS mendapat informasi SMS masuk, kemudian aplikasi SMS mengecek format SMS untuk mengetahui informasi apa yang diminta oleh pelanggan.
4. Aplikasi SMS mengirimkan SMS balasan yang melalui GSM modem.
5. Sebelum diterima pelanggan, SMS diterima SMS Center kemudian diteruskan ke pelanggan.
6. SMS balasan diterima pelanggan.

2.5 Macam-macam Aplikasi SMS

Ada banyak macam aplikasi SMS yang sedang berkembang sekarang. Aplikasi dimana pesan SMS dapat digunakan hampir tak terbatas. Beberapa contoh umum aplikasi SMS diantaranya sebagai berikut:

2.5.1 *Person to person text messaging*

Person to person text messaging sangat banyak digunakan oleh aplikasi SMS dan memang pada awalnya teknologi SMS dirancang untuknya. Dalam beberapa macam aplikasi pesan teks, seorang pengguna mobile phone mengetik sebuah pesan menggunakan keypad, selanjutnya ia memasukkan no tujuan / penerima dan menekan option selanjutnya menekan "OK" atau "Send", untuk mengirim. Ketika pesan diterima oleh penerima, akan diberitahu menggunakan suara atau getaran. Pengguna dapat membaca pesan teks suatu waktu dan membalasnya jika perlu.

Sebuah aplikasi chatting merupakan contoh lain dari aplikasi *Person to person text messaging* yang mengijinkan sekelompok / grup orang untuk bertukar SMS secara interaktif. Dalam sebuah aplikasi chatting, semua teks SMS dikirim dan diterima, kemudian ditampilkan di layar dan diurutkan berdasarkan waktu.

2.5.2 *Provision of Information*

Sebuah aplikasi popular dari teknologi SMS selain *Person to Person Messaging* adalah Provision of Information (Ketetapan Informasi) yakni menyampaikan ketetapan informasi ke *mobile phone* pelanggannya. Banyak penyedia jasa SMS (*Content Provider*) menggunakannya untuk mengirim informasi seperti berita, ramalan cuaca, data keuangan kepada pelanggan.

Kebanyakan informasi ini tidak cuma-cuma. Tagihan SMS dibebankan kepada pengguna informasi adalah cara yang digunakan oleh penyedia jasa.

2.5.3 *Downloading*

Pesan SMS dapat membawa data biner dan juga SMS dapat digunakan sebagai media untuk mendownload secara wireless. Seperti ringtone, wallpaper, gambar dan logo operator dapat di encode (disandikan) kedalam satu atau lebih pesan, tergantung ukuran objeknya. Seperti layanan informasi, layanan download ini juga biasanya tidak gratis dan tagihan SMS akan dibebankan kepada pelanggan yang menggunakan layanan ini. Objek download di encode kedalam satu atau lebih pesan SMS.

2.5.4 *Alert and Notifications* (Peringatan dan Pemberitahuan)

SMS merupakan teknologi yang sangat cocok untuk menyampaikan peringatan dan memberitahukan kejadian yang penting / sangat penting. Ada dua pertimbangan, yaitu:

1. Mobile phone biasanya dibawa oleh pemiliknya kapan saja. Kapan saja sebuah SMS diterima, mobile phone akan memberitahukan dengan menggunakan suara atau getaran. Dan pemiliknya dapat membaca apa isi dari pesan tersebut.
2. Teknologi SMS mengijinkan "*push* (desakan)" informasi. Berbeda dengan model "*pull* (tarikan)" dimana sebuah alat akan menanya server terlebih dahulu untuk memeriksa apakah ada informasi baru atau tidak. Model "tarikan" kurang cocok untuk aplikasi peringatan dan pemberitahuan. Karena membuang bandwidth dan meningkatkan aktifitas server.

2.5.5 *E-Commerce* dan Transaksi Kartu Kredit

Kapan saja suatu *e-commerce* dan transaksi kartu kredit dibuat, server akan mengirim pesan teks kepada pelanggan. Dengan itu pelanggan dapat mengetahui apakah transaksi telah dibuat.

2.5.6 Peringatan Bursa Saham

Dalam suatu aplikasi peringatan bursa saham, program secara konstan memonitor dan meneliti bursa saham. Jika suatu kondisi tertentu telah terpenuhi, maka program akan mengirimkan pesan menyangkut situasi tersebut. Sebagai contoh, kita dapat mengatur sistem peringatan jika harga saham suatu perusahaan lebih rendah dari nilai tertentu, program mengirimkan pesan peringatan kepada pengguna layanan ini.

2.5.7 *Remote System Monitoring*

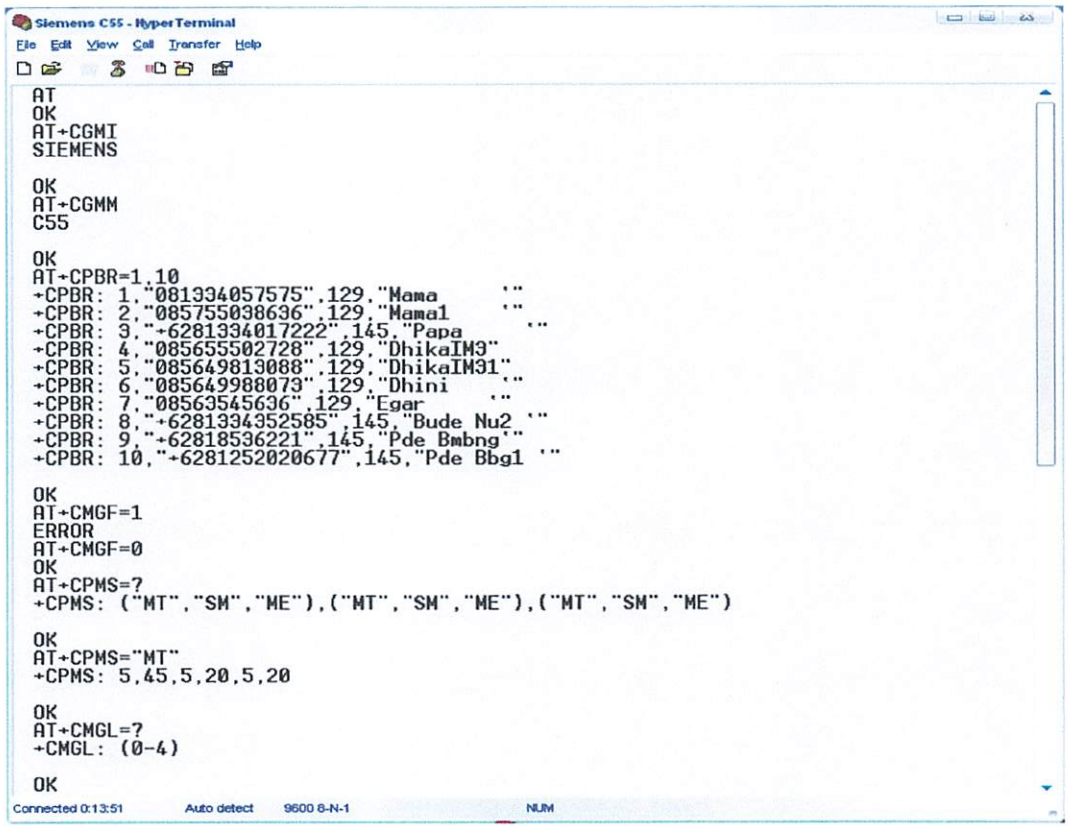
Dalam suatu aplikasi *remote system monitoring*, sebuah program (kadang - kadang menggunakan suatu sensor) apakah secara konstan memonitoring status dari suatu sistem. Jika dalam keadaan tertentu, program akan mengirim pesan kepada admin untuk memberitahukan situasi yang terjadi. Sebagai contoh program melakukan "*ping*" terhadap server, jika respon tidak diterima maka program akan mengirim pesan kepada admin.

2.6 AT *Command* (Perintah AT)

2.6.1 Memulai AT *Command*

Untuk mecoba AT Command sebelumnya penulis menyiapkan beberapa alat yang dibutuhkan yaitu sebuah handphone dan kabel datanya. Dalam melakukan percobaan ini digunakan handphone Siemens C55 dan kabel data DCA-510. Kemudian install driver kabel data. Setelah driver kabel data diinstall

buka Hyper Teminal untuk mencoba AT Command. Perintah AT Command diketik pada Hyper Terminal yang dimiliki oleh sistem operasi Microsoft Windows, disini penulis mencoba untuk mensimulasikan AT Command, sebelum melanjutkan ke pembuatan program. Perintah langsung diketikkan di terminal dan diakhiri dengan menekan tombol Enter. Gambar 2.11 menggambarkan hasil percobaan pengetikkan perintah AT Command di *Hyper Terminal* pada *Windows* XP.



Gambar 2.11 *Hyper Terminal* pada *Windows*

Berikut hasil percobaan:

1. ***Memulai Perintah AT***

   AT

   OK

2. ***Melihat Merk dan Model HP***

AT + CGMI

Siemens

OK

AT + CGMM

C55

OK

3. ***Melihat Buku Telpon***

AT + CPBR = 1, 10

+CPBR: 1, "081334057575", 129, "Mama"

+CPBR: 2, "085755038636", 129, "Mama1"

+CPBR: 3, "+6281334017222", 145, "Papa "

+CPBR: 4, "085655502728", 129, "DhikaIM3"

+CPBR: 5, "085649813088", 129, "DhikaIM31"

+CPBR: 6, "085649988073", 129, "Dhini"

+CPBR: 7, "08563545636", 129, "Egar"

+CPBR: 8, "+6281334352585", 145, "Bude Nu2"

+CPBR: 9, "+62818536221", 145, "Pade Bmbng"

+CPBR: 10, "+6281252020677", 145, "Pde Bbg1"

OK

4. ***Mengatur Jenis Pesan (PDU=0, Text=1)***

AT + CMGF = 1

ERROR

AT + CMGF = 0

OK

5. ***Melihat Settingan Memori***

AT + CPMS =?

+CPMS: ("MT", "SM", "ME"), ("MT", "SM", "ME")

OK

6. ***Mengatur Sumber Memori (Phone=ME, SIM=SM, ALL=MT)***

AT + CPMS = "MT"

+CPMS: 5, 45, 5, 20, 5, 20

OK

7. ***Melihat Option untuk Membaca List SMS***

AT + CMGL =?

+CMGL: (0 - 4)

OK

8. **Melihat *List SMS***

AT + CMGL= 1

+CMGL: 28,1,,156

07912658050000F0040C912658860542460000901131220180829C453C1C2

40ECB416E903B6C0789

C372103D4C4E1740F33CE85C7787D76137A84E5F83DA65771D6402CD

D769370B448783E66BF91934

CF83CE61373D0D8287D765D0F8DD8683C26C383AEC024DAB4990FA0

C12D7C374D07C9D7683DC20B8

FC7D9697E77390382C6781E6E830BC2CA6BB41E27A980E8287DD6536

287F0685C96190F82D2797E5EE7CD805

OK

9. **Membaca SMS (dengan menentukan *index*-nya, SMS akan tampil jika SMS dengan indeks tersebut berada dalam memori dan muncul "OK", jika tidak ada akan muncul "*ERROR*")**

AT + CMGR = 1

+CMGR: 0, ,0

OK

10. **Mengrim SMS**

AT+CMGW="+6285649813088">Test...kirim SMS

ERROR

2.6.2 Mengirim dan Membaca SMS Menggunakan AT Command

a. Mengirim SMS

Tabel 2.1 menjelaskan perintah-perintah AT Command yang digunakan untuk mengirim SMS beserta keterangan dari perintah-perintah AT Command tersebut.

Tabel 2.1 Mengirim SMS

Perintah	Keterangan
AT OK AT+CMGF=0 OK	Memulai AT Command. Menentukan format pesan (PDU=0, teks=1).

AT+CMGW="+6285649813088"> Test...kirim SMS	Menulis pesan diakhiri dengan menekan ctrl+Z

b. Membaca SMS

Tabel 2.2 menjelaskan perintah-perintah AT Command yang digunakan untuk membaca SMS beserta keterangan dari perintah-perintah AT Command tersebut.

Tabel 2.2 Membaca SMS

Perintah	**Keterangan**
AT OK AT+CMGF=0 OK AT+CPMS="MT" OK	Memulai AT Command. Menentukan format pesan (PDU=0, Text=1). Setting memori (Phone=ME, SIM=SM, ALL=MT).
AT+CMGL=1 +CMGL: 28,1,,156 07912658050000F0040C91265886054 2460000901131220180829C453C1C2 40ECB416E903B6C0789 C372103D4C4E1740F33CE85C7787 D76137A84E5F83DA65771D6402CD D769370B448783E66BF91934 CF83CE61373D0D8287D765D0F8D D8683C26C383AEC024DAB4990FA	Melihat list SMS dengan index = 1

0C12D7C374D07C9D7683DC20B8 FC7D9697E77390382C6781E6E830B C2CA6BB41E27A980E8287DD65362 87F0685C96190F82D2797E5 EE7CD805 OK	

2.6.3 Format PDU

Dalam pengiriman dan penerimaan pesan SMS terdapat dua mode, yaitu mode teks dan mode *Protocol Data Unit* (PDU). Mode teks adalah format pesan dalam bentuk teks asli yang dituliskan pada saat akan mengirimkan pesan. Sesungguhnya mode teks ini adalah hasil pengkodean yang direpresentasikan dalam format PDU. Sedangkan mode PDU adalah format pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter.

a. SMS PDU Pengirim

SMS PDU Pengirim adalah pesan yang dikirimkan oleh *handphone* pengirim kepada *handphone* penerima melalui SMSC. Dalam hal ini, pesan yang dikirimkan masih dalam mode teks, sedangkan dalam pengiriman ke SMSC, pesan harus dalam mode PDU. Untuk itu sebelum dikirim, *handphone* akan melakukan perubahan dari format teks menjadi format PDU. Proses ini sering disebut dengan *encodec*. Gambar 2.12 memperlihatkan format SMS PDU pengirim.

SCA	PDUType	MR	DA	PID	DCS	VP	UDL	UD

Gambar 2.12 Format SMS PDU Pengirim

1. ***Service Center Address* (SCA)**

SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama, yaitu *len*, *type of number*, dan *service center number*. Dalam pengiriman SMS, *service center number* tidak dicantumkan. Berikut Tabel 2.3 yang menjelaskan 3 komponen utama dari *Service Center Address* (SCA).

Tabel 2.3 *Service Center Address*

Oktet	Keterangan
Len	Panjang informasi SMSC dalam oktet
Type of number	Format nomor dari SMSC 81h = format lokal 91h = format internasional
Service center number	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil, maka pada karakter terakhir ditambahkan 0Fh

2. ***PDU Type***

Nilai *default* dari *PDU Type* untuk SMS pengirim adalah 11h (atau 00010001). Berikut Tabel 2.4 yang menjelaskan tentang nilai *default* dari *PDU Type* untuk SMS pengirim.

Tabel 2.4 *PDU Type* SMS Pengirim

Bit No	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRR	VPF	VPF	RD	MTI	MTI
Nilai	0	0	0	1	0	0	0	1

Keterangan:

RP : ***Reply Path.*** Parameter yang menunjukkan bahwa alur jawaban ada.

UDHI : ***User Data Header Indicator.*** Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul atau tema.

SRR : ***Status Report Request.*** Bit ini bernilai 1 jika laporan pengiriman data diminta.

VPF : ***Validity Period Format.*** Format dari batas waktu pengiriman jika pesan gagal diterima. Nilai:

00 = pesan tidak disimpan di SMSC

10 = format relatif (satu oktet)

01 = format *enhanced* (tujuh oktet)

11 = format absolut (tujuh oktet)

RD : ***Reject Duplicates.*** Parameter yang menandakan apakah *service center* akan menerima pengiriman pesan untuk pesan yang masih disimpan dalam *service center* tersebut.

MTI : ***Message Type Indicator.*** Bit ini bernilai 1 untuk menunjukkan bahwa PDU ini adalah suatu SMS Pengirim.

3. ***Message Reference* (MR)**. *Message Reference* adalah acuan dari pengaturan pesan SMS. Apabila pengaturan dilakukan sendiri oleh *handphone* tujuan, maka nilai untuk MR adalah 00.

4. ***Destination Address* (DA)**. *Destination Address* adalah alamat (nomor) tujuan, yang terdiri dari panjangnya nomor tujuan (*Len*), format dari nomor tujuan (*Type Number*), dan nomor tujuan (*Destination Number*). Berikut Tabel 2.5 yang menjelaskan tentang bagian-bagian *destination address* dari SMS PDU pengirim.

Tabel 2.5 *Destination Address*

Oktet	Keterangan
Len	Panjang informasi SMSC dalam oktet
Type of number	Format nomor dari SMSC 81h = format lokal 91h = format internasional
Service center number	Nomor tujuan. Jika panjangnya ganjil, maka pada karakter terakhir ditambahkan 0Fh

5. ***Protocol Identifier* (PID)**. *Protocol Identifier* adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari *handphone* pengirim, misalnya tipe *standard text*, *fax*, e-mail, dan lain-lain. Nilai *default* dari PID adalah 00 yang berarti menggunakan tipe *standard text*.

6. ***Data Coding Scheme* (DCS)**. *Data Coding Scheme* adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa *standard text* SMS, *flash* SMS, atau *blinking* SMS. Apabila menggunakan *standard text*, maka nilai dari DCS adalah 00.

7. ***Validity Period* (VP)**. *Validity Period* adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal diterima oleh *handphone* penerima. Berikut Tabel 2.6 yang menjelaskan tentang *validity period* atau lamanya waktu pesan SMS disimpan di SMSC.

Tabel 2.6 *Validity Period*

Waktu VP	Nilai VP
5 menit – 720 menit (12 jam)	Waktu VP / 5) - 1
12,5 Jam – 24 Jam	143 + ((Waktu VP – 12) * 2)
2 – 30 hari	166 + Waktu VP
Lebih dari 4 minggu	192 + Waktu VP

Hasil nilai VP merupakan nilai desimal sehingga diubah terlebih dahulu ke dalam format heksadesimal sebelum dimasukkan ke dalam SMS PDU Pengirim.

8. ***User Data Length* (UDL)**. *User Data Length* adalah panjangnya pesan SMS yang diterima dalam bentuk *standard text*.

9. ***User Data* (UD)**. *User Data* adalah isi pesan yang akan diterima dalam format heksadesimal. Berikut Tabel 2.7 yang menjelaskan tentang isi pesan yang akan diterima dalam format heksadesimal.

Tabel 2.7 *Default Alphabet* ETSI GSM 03.38

Desimal	0	1	2	3	4	5	6	7	8	9
0										
1	LF			CR						
2										
3			SP	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

b. SMS PDU Penerima

SMS PDU Penerima adalah pesan yang diterima dari handphone *pengirim* melalui SMSC pada *handphone* penerima. Karena pada SMSC pesan tersebut telah diubah dalam bentuk PDU, maka pada *handphone* penerima pesan tersebut akan diubah ke dalam mode teks. Proses ini sering disebut dengan *decodec*. Gambar 2.13 memperlihatkan format SMS PDU penerima.

SCA	PDUType	OA	PID	DCS	SCTS	UDL	UD

Gambar 2.13 Format SMS PDU Penerima

1. ***Service Center Address* (SCA).** SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama, yaitu *len*, *type of number*, dan *service center number*. Tabel 2.8 di bawah ini menjelaskan tentang 3 komponen utama dari SCA (*Service Center Address*).

Tabel 2.8 *Service Center Address*

Oktet	Keterangan
Len	Panjang informasi SMSC dalam oktet
Type of number	Format nomor dari SMSC 81h = format lokal 91h = format internasional
Service center number	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil, maka pada karakter terakhir ditambahkan 0Fh

2. ***PDU Type.***

Nilai *default* dari *PDU Type* untuk SMS Penerima adalah 04h (atau 00000100). Tabel 2.9 di bawah ini menjelaskan tentang nilai *default* dari *PDU Type* untuk SMS Penerima.

Tabel 2.9 *PDU Type* SMS Penerima

Bit No	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRI	<nn>	<nn>	MMS	MTI	MTI
Nilai	0	0	0	0	0	1	0	0

Keterangan:

RP : ***Reply Path***. Parameter yang menunjukkan bahwa alur jawaban ada.

UDHI : ***User Data Header Indicator***. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul atau tema.

SRI : ***Status Report Indication***. Bit ini bernilai 1 jika laporan pengiriman data akan dikembalikan ke SME.

MMS : ***More Messages to Send.*** Bit ini bernilai 0 jika ada pesan lebih yang akan dikirim.

MTI : ***Message Type Indicator***. Bit ini bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS Penerima.

3. ***Originator Address*** **(OA)**. *Originator Address* adalah alamat (nomor) dari pengirim yang terdiri dari panjangnya nomor pengirim (*Len*), format dari nomor pengirim (*Type of Number*), dan nomor pengirim (*Originator Number*). Tabel 2.10 di bawah ini menjelaskan tentang bagian-bagian dari *originator address*.

Tabel 2.10 *Originator Address*

Oktet	Keterangan
Len	Panjang nomor pengirim
Type of number	Format dari nomor pengirim 81h = format lokal 91h = format internasional
Originator number	Nomor pengirim. Jika panjangnya ganjil, maka pada karakter terakhir ditambahkan 0Fh.

4. ***Protocol Identifier*** **(PID)**. *Protocol Identifier* adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari *handphone* pengirim, misalnya tipe *standard text*, *fax*, e-mail, dan lain-lain. Nilai *default* dari PID adalah 00 yang berarti menggunakan tipe *standard text*.

5. ***Data Coding Scheme*** **(DCS)**. *Data Coding Scheme* adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa *standard text* SMS, *flash* SMS, atau *blinking* SMS. Apabila menggunakan *standard text*, maka nilai dari DCS adalah 00.

6. ***Service Center Time Stamp*** **(SCTS)**. *Service Center Time Stamp* adalah waktu dari penerimaan pesan oleh SMSC penerima. SCTS terdiri atas tahun, bulan, tanggal, jam, menit, detik, dan zona waktu.
7. ***User Data Length*** **(UDL)**. *User Data Length* adalah panjangnya pesan SMS yang diterima dalam bentuk *standard text*.
8. ***User Data*** **(UD)**. *User Data* adalah isi pesan yang akan diterima dalam format heksadesimal.

2.7. Delphi

Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi adalah kompiler/penterjemah bahasa Pascal (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan basic, C. Bahasa pemrograman di delphi disebut bahasa *procedural* artinya bahasa/sintaknya mengikuti urutan tertentu/prosedur. Delphi termasuk keluarga visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Delphi juga merupakan bahasa berorentasi objek. Saat anda pertama kali masuk ke Visual Basic maupun delphi , anda akan diperhadapkan pada sebuah form kosong yang akan dibuat secara otomatis. Form tersebut diberi nama Form1. Form ini merupakan tempat bekerja untuk membuat antarmuka pengguna (*user interface*).

2.7.1. Kegunaan Delphi

1. Untuk membuat aplikasi windows.
2. Untuk merancang aplikasi program berbasis grafis.
3. Untuk membuat program berbasis jaringan (*client / server*).

4. Untuk merancang program .net (berbasis internet).

2.7.2. Keunggulan Delphi

1. IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source code delphi yang merupakan turunan dari pascal, sehingga tidak perlu suatu penyesuaian lagi.
4. Bersifat multi purphase, artinya bahasa pemrograman delphi dapat digunakan untuk pengembangan aplikasi.

2.8. Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah., dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access. Para pengguna / *programmer* yang telah berpengalaman dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang berpengalaman dapat menggunakanya karena Microsoft Accsess merupakan program yang telah

disetting sedemikian rupa agar para penggunanya baik dari para programmer yang handal atau tidak. Access juga menawarkan teknik-teknik pemrograman berorientasi objek.

Keunggulan Microsoft Access:

1. Aplikasinya mudah diperoleh.
2. Manipulasi tabel & data sangat mudah dilakukan.
3. Relasi antar tabel dapat dibuat dengan mudah.
4. Perintah SQL dapat diberikan.
5. Dapat diintegrasikan dengan bahasa pemrograman, misalnya Delphi, VB, dll.
6. Tersedia fasilitas untuk sekuriti data.
7. Mampu menyimpan data dalam jumlah yang sangat besar (jauh lebih besar daripada Paradox).

2.9. Aplikasi Internet

2.9.1. Electronic Mail (E-mail)

Electronic Mail (E-mail) merupakan aplikasi yang paling banyak digunakan, dan termasuk salah satu dari aplikasi pertama di internet. Dengan *Electronic Mail (E-mail)* anda dapat mengirim dan menerima surat, pesan, dokumen secara elektronik dengan pemakai lain di internet yang memiliki alamat *Electronic Mail (E-mail).*

Electronic Mail (E-mail) merupakan sarana komunikasi yang cukup handal, perbandingan dengan surat biasa adalah waktu pengirimannya yang sangat cepat. *Electronic Mail (E-mail)* bukanlah pelayanan *"end to end"* karena mesin pengirim dan penerima tidak perlu berkomuniksi secara langsung. Proses

penyampaian *electronic mail (e-mail)* dapat dianalogikan dengan penyampaian surat oleh kantor pos dan giro. Proses ini desebut *"store and forward"*.

Sistem *electronic mail (e-mail)* bekerja bagaimana surat biasa, namun bedanya *electronic mail (e-mail)* dikirim dalam bentuk dokumen dan didistribusikan secara elektronik. Kelebihan *Electronic Mail (E-mail)* adalah kecepatan, jarak ribuan mil hanya akan ditempuh selama beberapa detik, sedang dengan jasa pos bisa memakan waktu berminggu - minggu. Setiap *Electronic Mail (e-mail)* memiliki pemilik (bisa perorangan, organisasi) dan alamatnya yang jelas, unik, dan berbeda satu sama lain sehingga bisa dipakai sebagai penunjuk identitas.

Format pengalamatan *electronic mail (e-mail)* mengikuti standar sebagai berikut: identitas_pemilik@nama_host, misalnya reza_cow@yahoo.co.id. (reza_cow) menunjukkan identitas pemilik alamat kotak surat (*mail box*), @ menunjukkan bahwa format pengalamatan ini khusus untuk email, (yahoo) nama host, (co) menunjukkan identitas *domain host* yahoo, yaitu organisasi, (id) identitas *top domain host* yaitu berada di Negara Indonesia. Secara umum format ini mirip dengan *Universal Resource Locator (URL)*.

Protokol yang menentukan distribusi *electronic mail (e-mail)* di internet disebut SMTP *(Simple Mail Transfer Protocol)* yang berbasis kode ASCII. Format *electronic mail (e-mail)* dalam kode ASCII di pergunakan khusus untuk dokumen *Electronic Mail (E-mail)* yang berupa teks. Sedangkan untuk transfer *Electronic Mail (E-mail)* dalam bentuk grafis digunakan format biner dan menggunakan protokol khusus disebut *Multi Purpose Internet Mail Extension (MIME)*.

Protokol yang dipakai untuk menerima *electronic mail (e-mail)* disebut *Post Office Protocol (POP).* Protokol ini mendefinisikan bagaimana sebuah *electronic mail (e-mail)* dalam bentuk asal (format SMTP) diterjemahkan dan diterima oleh sistem yang dimiliki user, dengan kata lain SMTP adalah gerbang untuk mengirim *electronic mail (e-mail)* dan sebaliknya POP adalah gerbang untuk menerima *electronic mail (e-mail).*

Electronic Mail (E-mail) mempunyai cara kerja yang sederhana yang memiliki kemiripan dan perbedaan dengan sistem surat dikantor pos. Mirip dengan kantor pos, *electronic mail (e-mail)* juga membutuhkan alamat pengirim dan penerima. Sebagai media transportasi, *electronic mail (e-mail)* menggunakan TCP / IP. Pada proses transfer *electronic mail (e-mail)* digunakan protokol SMTP. SMTP merupakan protokol yang bekerja diatas protokol TCP / IP. Berbeda dengan surat pos yang diantar langsung kerumah, *electronic mail (e-mail)* tidak akan langsung diantarkan ke *inbox* program *e-mail client.* Pengguna *e-mail* harus menghubungi *e-mail server* tempat *e-mail* tersimpan. Proses pengambilan ini biasanya menggunakan protokol POP3.

2.9.2. News Usenet

Digunakan sebagai sarana untuk berdiskusi antar pemakai jaringan internet. Aplikasi ini hampir serupa dengan suatu papan pengumuman, dimana setiap orang dapat mengirim, melihat, dan menaggapi berita atau suatu topik diskusi dengan fasilitas yang hampir sama dengan *e-mail.* Topik diskusi dipisahkan oleh group, dan pemakai yang berminat dapat melihat isi diskusi pada *newsgroup* tersebut.

2.9.3. World Wide Web

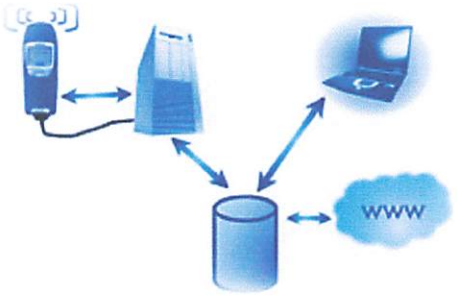
Salah satu layanan aplikasi di internet ini adalah *World Wide Web* (WWW), pelayanan yang cukup baru dikembangkan di internet dan menjadi layana aplikasi yang paling popular digunakan pemakai jaringan internet dan perkembangannya terus dilakukan sampai saat ini untuk menyempurnakan teknolog ini. WWW atau yang biasa disebut dengan web saja, bekerja menggunakan teknologi yang disebut *hypertext,* yang kemudian dikembangkan menjadi suatu protokol aplikasi yang disebut HTTP (*HyperText Transfer Protocol).* Dengan adanya fasilitas ini menjadikan web sebagai salah satu aplikasi yang paling luwes untuk menjelajahi internet. Dengan menggunakan WWW, peng-aksesan beragam sumber informasi di internet misalnya gopher, WAIS, FTP, mail dan sebagainya, dapat dilakukan melaui suatu cara yang menggabungkan beberapa jenis representasi dan metode pengaksesan informasi dan menyajikannya dalam beragam bentuk informasi seperti text, grafik, suara, animasi, video, dan sebagainya.

BAB III

PERANCANGAN DAN DESAIN SISTEM

3.1 Analisa Sistem

SMS *Gateway* adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistribusikan pesan-pesan yang di-*generate* lewat sistem informasi melalui media SMS yang di-*handle* oleh jaringan seluler. Gambar 3.1 di bawah ini menggambarkan tentang SMS Gateway.



Gambar 3.1 SMS *Gateway*

Salah satu mode komunikasi yang handal saat ini adalah pesan pendek / *short messaging system* (SMS). Implikasinya, salah satu model komunikasi data yang bisa dipakai adalah SMS. Artinya, SMS tersebut harus bisa melakukan transaksi dengan *database*. Untuk itu perlu dibangun sebuah sistem yang disebut sebagai SMS *Gateway*. Secara khusus, sistem ini akan memiliki fungsi-fungsi sebagai berikut:

Message Management dan *Delivery*

1. Pengaturan pesan yang meliputi manajemen prioritas pesan, manajemen pengiriman pesan, dan manajemen antrian.

2. Pesan yang dilalukan harus sedapat mungkin *fail safe*. Artinya, jika terdapat gangguan pada jaringan telekomunikasi, maka sistem secara otomatis akan mengirim ulang pesan tersebut.

Korelasi

Berfungsi untuk melakukan korelasi data untuk menghasilkan data baru hasil korelasi. Pada sistem yang terpasang saat ini, arsitektur lalu lintas data melalui SMS sudah terjalin cukup baik. Hanya saja, keterbatasan akses data dan tujuan informasi SMS yang belum terfokus menyebabkan banyaknya jawaban standar (*default replies*) masih banyak terjadi. SMS *Gateway* banyak digunakan dalam berbagai proses bisnis dan usaha.

Proses bisnis dan bidang layanan yang bisa ditangani oleh aplikasi SMS *Gateway*, yaitu sebagai berikut:

1. *Manajemen Inventori*
2. CRM (*Customer Relationship Management*), misalnya Rumah Makan, *Cafe*, *Executive Club*, Stasiun Radio, Stasiun TV, Lembaga Pendidikan
3. *Call Center* dan SMS Pengaduan, misalnya Polisi, PLN, PAM, Instansi Pemerintah. SMS *Gateway* memanfaatkan arsitektur teknologi komunikasi SMS untuk menerapkan aplikasi bernilai tambah dengan memanfaatkan komunikasi SMS untuk optimalisasi proses bisnis perusahaan dan peningkatan kualitas layanan dari institusi pelayanan publik.

Fungsi SMS *Gateway* diantaranya:

1. Memperbesar skala aplikasi teknologi informasi dengan menggunakan komunikasi SMS interaktif

2. Menyediakan aplikasi kolaborasi komunikasi SMS berbasis web untuk pengguna di institusi atau perusahaan
3. Menjangkau konsumen maupun pengguna jasa layanan institusi atau perusahaan secara mudah menggunakan komunikasi SMS interaktif

Fitur-fitur standar SMS *Gateway* yaitu komunikasi SMS interaktif dua arah, SMS *info on demand*, SMS *service settings*, SMS *Automatic Registration*, *polling* SMS, pengiriman SMS *Broadcast*, pengiriman SMS ke *Call Group*, pengiriman SMS terjadwal, personalisasi SMS, antarmuka aplikasi berbasis web, buku alamat dan *call group*, manajemen pengguna, sistem *security access*, serta sistem parameter.

Fitur-fitur advance SMS *Gateway* yaitu antarmuka dinamis untuk integrasi ke *database* perusahaan, SMS *Remote Control*, *E-mail to* SMS, SMS *to E-mail*, ekspansi modem GSM, dan koneksi langsung ke SMSC via SMPP.

3.2 Metode Perancangan

3.2.1 Perancangan Sistem Aplikasi

Sistem yang dibangun pada aplikasi, berbasis SMS *Gateway* dengan layanan pengiriman *E-mail to* SMS & SMS *to E-mail*. Cara kerja dari layanan pengiriman *E-mail to* SMS & SMS *to E-mail* yang dibangun sangat sederhana yaitu:

i. Orang / calon pelanggan yang akan menggunakan fasilitas ini harus melakukan registrasi terlebih dahulu dengan mengetik format SMS berikut:

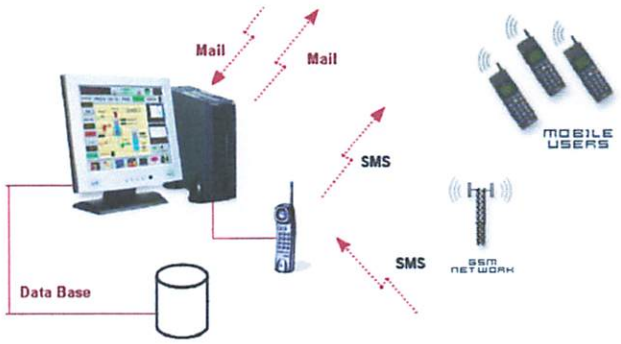
REG#NAMA_LENGKAP#JENIS_KELAMIN#ALAMAT_LENGKAP#GMAIL_ADDRESS#GMAIL_PASSWORD# dan dikirimkan ke nomor HP SMS *Gateway*.

ii. Setelah melakukan registrasi, barulah orang tersebut bisa mengirimkan SMS *to E-mail* dengan format SMS sebagai berikut: **EMAIL#TO_NAME#TO_EMAIL_ADDRESS#TO_SUBJECT#TO_BODY#** dan dikirimkan ke nomor HP SMS *Gateway*.

iii. Untuk format *E-mail to* SMS, format SMS-nya tidak ada, karena kalau orang sudah registrasi, *E-mail* orang itu akan di*download* secara otomatis oleh aplikasi dan langsung dikirimkan ke nomor HP orang tersebut.

iv. Sedangkan user / pengoperasi aplikasi ini, yaitu Admin / Operator, bertugas menginputkan data orang / calon pelanggan yang melakukan registrasi melalui SMS, mengirimkan *E-mail* ke **TO_NAME** & **TO_EMAIL_ADDRESS** yang dituju berdasarkan nomor HP yang mengirim format SMS *to E-mail*, dan mendownload *E-mail* pelanggan serta mengirimkannya ke nomor HP pelanggan.

Gambar 3.2 di bawah ini menggambarkan desain sistem aplikasi layanan pengiriman *E-mail to* SMS & SMS *to E-mail*.



Gambar 3.2 Desain Sistem Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

Penjelasan dari gambar di atas sebagai berikut:

- HP yang terhubung dengan PC adalah **HP Siemens C55**. HP ini sebagai sarana penghubung antara PC dan telepon seluler pengirim yang akan melakukan registrasi, pengiriman *E-mail to* SMS & SMS *to E-mail* secara otomatis dengan catatan PC yang digunakan harus tetap dalam keadaan menyala dan terhubung ke internet.
- Di dalam HP tersebut terdapat **SIM *Card* yang masih aktif** sebagai nomor tujuan dari layanan pengiriman *E-mail to* SMS & SMS *to E-mail* ini.
- Kabel data yang digunakan untuk menghubungkan HP dengan PC adalah **kabel data USB *type* DCA-510**. *Driver* yang digunakan agar PC dapat mengenali HP ada 2 macam yaitu *driver* **Siemens Data *Suite*** untuk menyetting HP Siemens sebagai modem dan **USB *driver* Siemens *type* DCA-510** untuk PC mengenali *type* kabel data dari HP Siemens.
- Sistem Operasi yang digunakan untuk menjalankan aplikasi ini adalah **Windows XP SP 2**, bahasa pemrograman (*development language*) yang digunakan untuk merancang aplikasi ini adalah **Borland Delphi 7**,

basis data yang digunakan untuk membangun aplikasi ini adalah **Microsoft Access**.

❖ Komponen Delphi:

- **ExpressForum v1.3**

  Component ini digunakan untuk tampilan *ProgressBar* Sinyal dan *Battery* HP.

- **Email Active-X dan Imap ActiveX**

  Component E-mail dan IMAP ini digunakan untuk mengirim *E-mail* dan men*download E-mail.*

- **Shockwave Flash (Version 1.0)**

  Component ini digunakan untuk menampilkan animasi *Flash* pada aplikasi SMS *Gateway E-mail to* SMS dan SMS *to E-mail.*

- **Alpha**

  Component Alpha ini digunakan untuk merubah *SKIN* pada aplikasi SMS *Gateway E-mail to* SMS dan SMS *to E-mail.*

- **Button Comps**

  Component ini digunakan untuk menampilkan tulisan pada kotak yang pinggirannya terdapat *border* berwarna (seperti untuk menampilkan tulisan ".:: Informasi Handphone ::., .:: Sinyal, Battery, dan Indikator ::., dan lain-lain").

- **JlAqua**

  Component Button yang bagus dan menarik dengan berbagai pilihan warna.

- **OUX**

  Component ini digunakan untuk mengkoneksikan aplikasi dengan *Handphone* yang dijadikan sebagai modem untuk SMS *Gateway.*

- **ADOConnection**

  Component ini digunakan untuk mengkoneksikan aplikasi dengan *database* Microsoft Access yang dijadikan sebagai *database* aplikasi SMS *Gateway E-mail to* SMS dan SMS *to E-mail.*

- **QuickReport**

  Component ini digunakan untuk menampilkan data sehingga bisa digunakan untuk kepentingan melakukan print data.

3.2.2. Perancangan Form Aplikasi

Perancangan form aplikasi hanya terdiri dari 1 bagian yaitu form pada aplikasi *server*, sedangkan untuk form pada aplikasi pelanggan tidak ada, karena pelanggan hanya melakukan registrasi melalui SMS dan mengirim SMS *to E-mail* dengan format SMS yang telah dijelaskan di atas.

➢ Form Aplikasi *Server*

Pada form aplikasi *server*, terdapat 2 form yaitu form *Login* dan form Utama.

❖ Form *Login* terdiri dari 3 Label yaitu *user role*, *user ID*, dan *password*, 1 *ComboBox* untuk memilih *user role*, 2 *TextBox* untuk memasukkan *user ID* dan *password*, serta 2 *Button* untuk masuk ke Form Utama dan membatalkan *Login.*

- Form Utama terdiri dari *MainMenu* (Menu Utama) dan beberapa *Button* yang berada pada bagian atas form. Setiap *Button* mengarah ke form yang berbeda dan juga mempunyai fungsi masing-masing diantaranya:
  - *Button Login / Logout* digunakan untuk masuk / keluar dari aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).
  - *Button User Account* digunakan untuk memasukkan, merubah, menghapus data *user* / pengoperasi aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).
  - *Button* Info SMS digunakan untuk membaca informasi SMS yang ada pada HP SMS *Gateway*.
  - *Button* Daftar *Phonebook* digunakan untuk membaca daftar buku telepon yang ada pada HP SMS *Gateway*.
  - *Button Phone Call* digunakan untuk melakukan panggilan keluar / telepon.
  - *Button* Kirim SMS digunakan untuk mengirim SMS secara manual dari aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).
  - *Button Broadcast* SMS digunakan untuk mengirim SMS secara manual ke banyak nomor yang telah teregistrasi dari aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).
  - *Button* Kirim *E-mail* digunakan untuk mengirim *E-mail* secara manual dari aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).

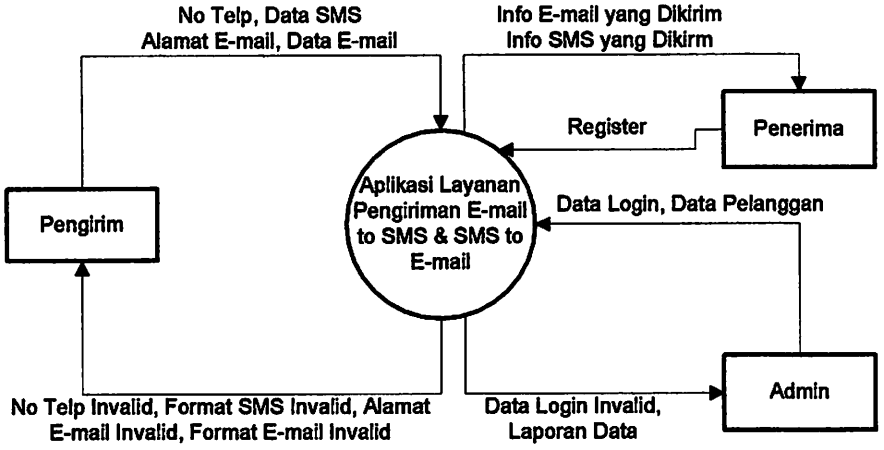
- *Button* Data Registrasi digunakan untuk memasukkan / merubah data pelanggan yang sudah melakukan registrasi.
- *Button View* Data *User* digunakan untuk melihat informasi mengenai data *user* / pengoperasi aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*), yaitu Admin / Operator.
- *Button View* Data *Login* digunakan untuk melihat data saat *login* / siapa saja yang pernah menggunakan aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).
- *Button View* Data Pelanggan digunakan untuk melihat data orang yang sudah melakukan registrasi.
- *Button View* Data *Phonebook* digunakan untuk melihat daftar buku telepon pada HP SMS *Gateway*.
- *Button View* Data *Inbox* SMS digunakan untuk melihat data SMS masuk yang ada pada HP SMS *Gateway*.
- *Button View* Data *Inbox E-mail* digunakan untuk melihat data *E-mail* yang masuk hasil *download E-mail* dari webmail gmail.com.
- *Button View* Data *Sent E-mail* / SMS digunakan untuk melihat data *E-mail* maupun SMS yang terkirim.
- *Button* Keluar digunakan untuk keluar dari aplikasi SMS *Gateway* (Layanan Pengiriman *E-mail to* SMS & SMS *to E-mail*).

3.2.3. Perancangan Aliran Data

Perancangan Aliran Data bertujuan untuk mengetahui proses informasi yang mengalir melalui perangkat lunak. Untuk menggambarkan perancangan secara umum digunakan alat bantu yaitu *Data Flow Diagram* (DFD).

3.2.3.1 DFD Level 0

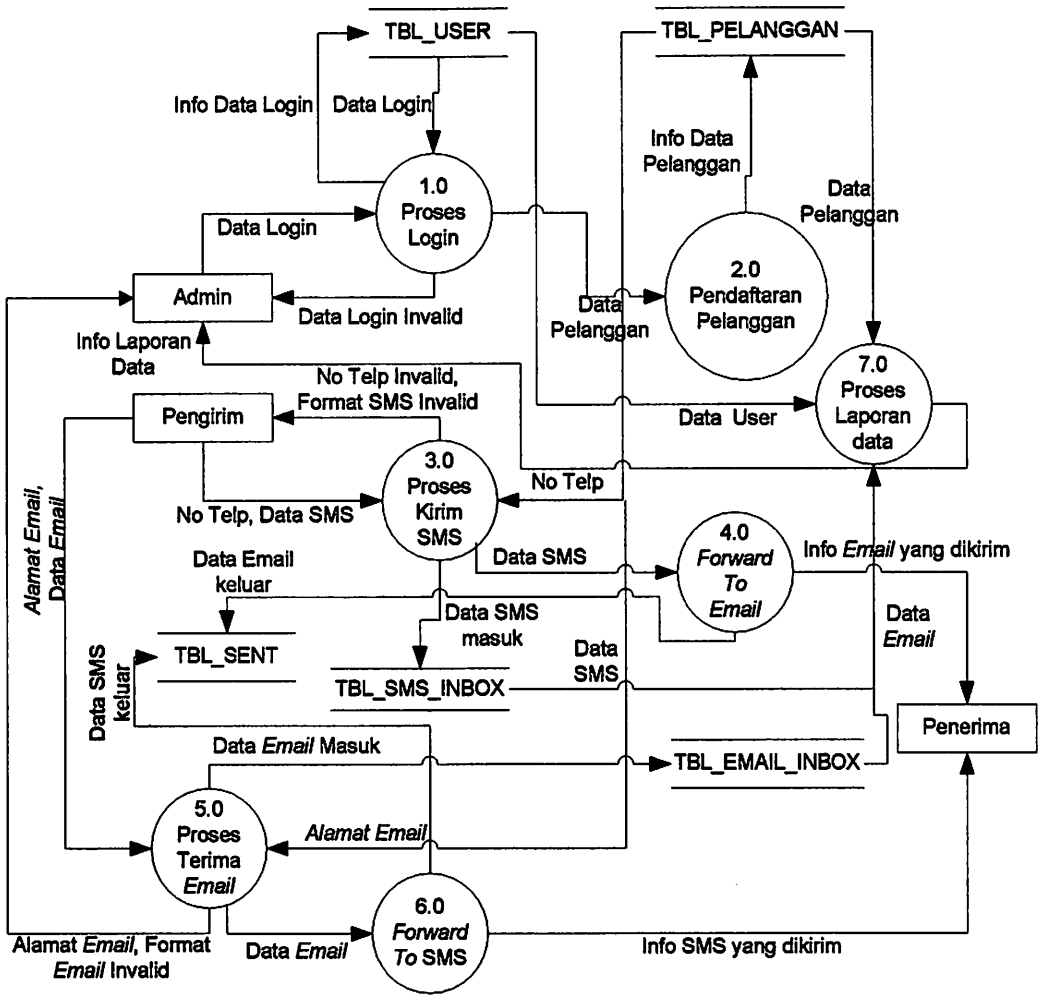
DFD Level 0 digunakan untuk menggambarkan hubungan antar entitas - entitas yang terdapat diluar sistem dan masukan serta keluaran dari sistem. Gambar 3.3 di bawah ini menggambarkan bagaimana proses yang terjadi pada aplikasi SMS *Gateway* untuk mengirimkan *E-mail* dari telepon seluler serta bagaimana user melakukan interaksi terhadap aplikasi SMS *Gateway*.



Gambar 3.3 DFD Level 0 Pengiriman *E-mail to* SMS dan SMS *to E- mail*

3.2.3.2. DFD Level 1

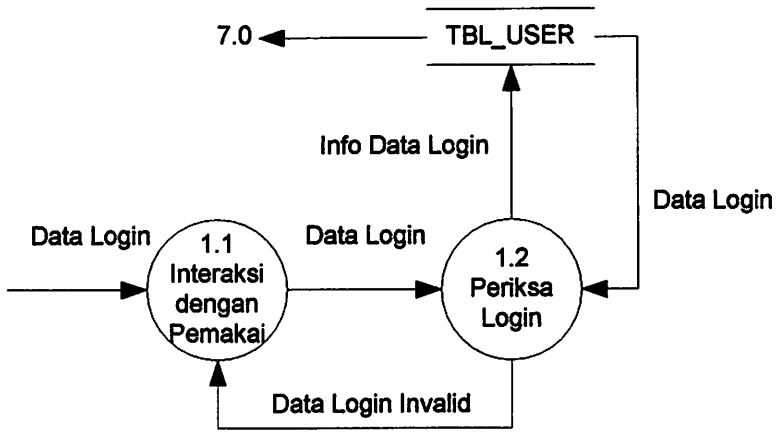
DFD Level 1 merupakan penjabaran dari proses DFD Level 0 yang memuat proses-proses yang ada dalam sistem secara garis besar dan keseluruhan. Gambar 3.4 di bawah ini menggambarkan DFD Level 1 dalam perancangan aliran data.



Gambar 3.4 DFD Level 1 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 1 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan keseluruhan proses Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail.*

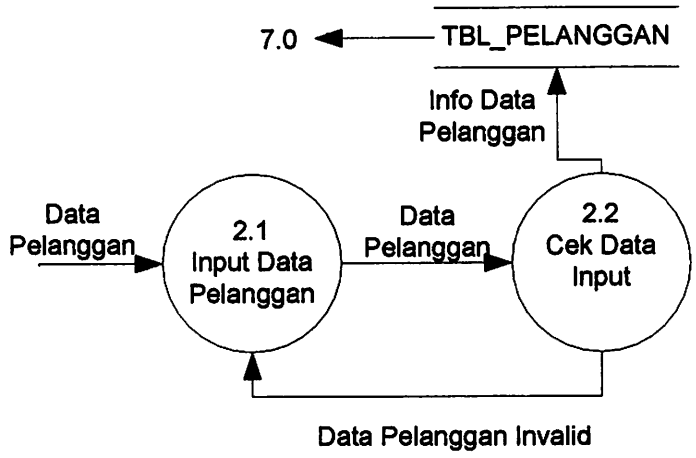
Gambar 3.5 di bawah ini merupakan gambar DFD Level 2 Proses 1 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses *Login* untuk Administrator / Operator.



Gambar 3.5 DFD Level 2 Proses 1 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 1 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses *login* untuk Administrator / Operator. Administrator / Operator akan dihadapkan pada form *login*, kemudian data *login* yang dimasukkan yaitu *user role*, *user id*, dan *password*, terlebih dahulu akan diperiksa ke TBL_USER. Apabila data *login* sesuai (*valid*), maka Administrator / Operator dapat menggunakan aplikasi yang kemudian akan dihadapkan pada form utama aplikasi, akan tetapi jika data *login* yang dimasukkan tidak *valid* maka aplikasi akan memberitahukan bahwa *login invalid*, maka Aministrator dan Operator tidak dapat menggunakan aplikasi.

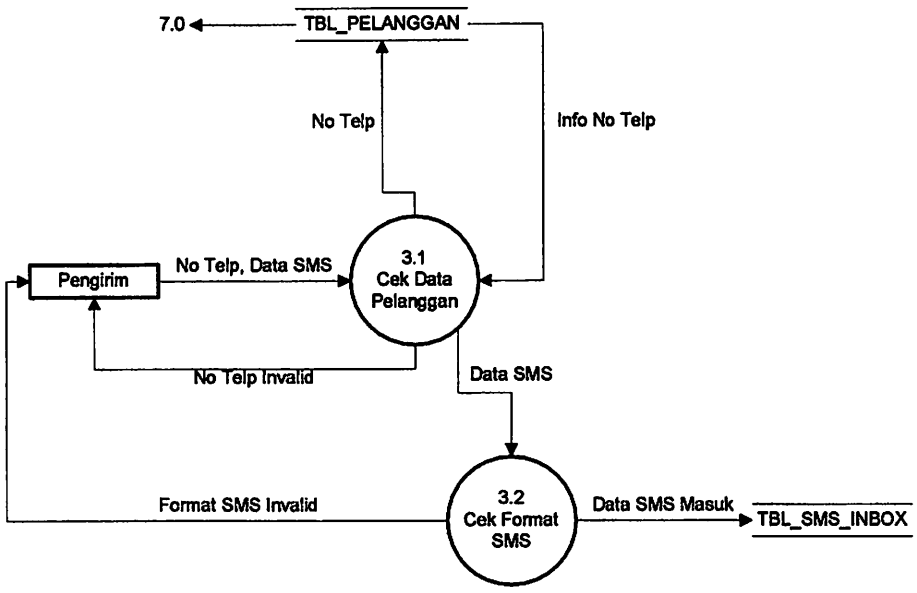
Gambar 3.6 di bawah ini merupakan gambar DFD Level 2 Proses 2 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses *input* data pelanggan yang dilakukan oleh Administrator / Operator.



Gambar 3.6 DFD Level 2 Proses 2 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 2 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses *input* data pelanggan, dimana data pelanggan akan dibutuhkan untuk memfilter pelanggan yang dapat menggunakan layanan aplikasi ini. Administrator / Operator melakukan *input* data pelanggan yang dapat menggunakan layanan aplikasi ini, dimana data yang dimasukkan akan diperiksa oleh aplikasi, apabila data *valid* maka data akan disimpan pada TBL_PELANGGAN, apabila data *invalid* maka Administrator / Operator mendapatkan *feedback* berupa peringatan data yang di*input invalid.*

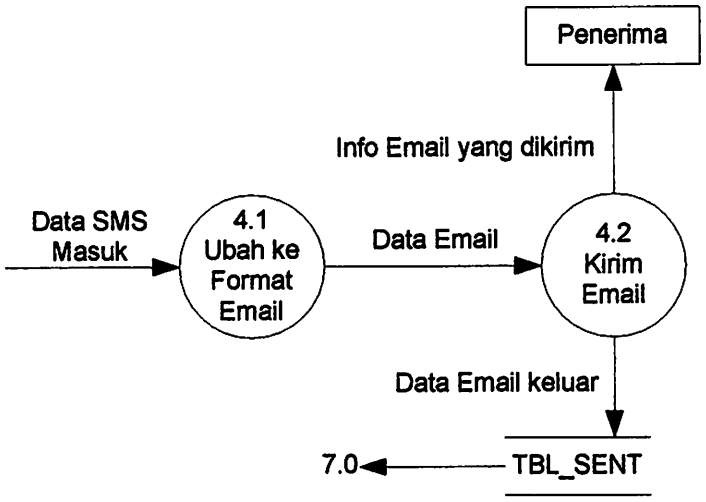
Gambar 3.7 di bawah ini merupakan gambar DFD Level 2 Proses 3 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses pengiriman SMS *to E-mail* secara otomatis.



Gambar 3.7 DFD Level 2 Proses 3 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 3 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses pengiriman SMS *to E-mail.* Tahap pertama yang dilakukan pada proses ini adalah membaca data SMS yang masuk, kemudian aplikasi akan melakukan *request* ke *database* untuk memperoleh keterangan apakah orang yang mengirim SMS dapat menggunakan layanan ini, jika orang tersebut tidak terdaftar maka aplikasi akan mengirimkan pesan peringatan ke pengirim bahwa pengirim tidak terdaftar sehingga tidak dapat menggunakan layanan aplikasi ini. Setelah data orang itu diperiksa langkah selanjutnya mengecek format SMS yang dikirimkan apakah sesuai atau tidak, apabila tidak maka aplikasi akan mengirimkan pesan peringatan bahwa SMS yang dikirimkan oleh orang itu salah format, jika format SMS benar maka akan dilanjutkan ke proses *Forward to E-mail.* Data SMS yang masuk akan disimpan di TBL_SMS_INBOX.

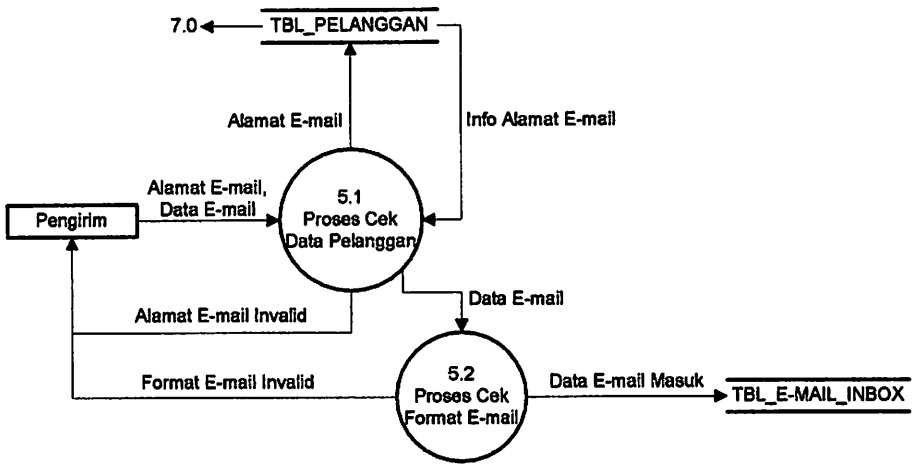
Gambar 3.8 di bawah ini merupakan gambar DFD Level 2 Proses 4 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses merubah data SMS masuk ke format *E-mail.*



Gambar 3.8 DFD Level 2 Proses 4 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 4 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses *Forward to Email*, dimana data SMS yang masuk diubah ke format *E-mail,* kemudian *E-mail* akan dikirimkan ke penerima. *E-mail* yang terkirim akan disimpan di TBL_SENT.

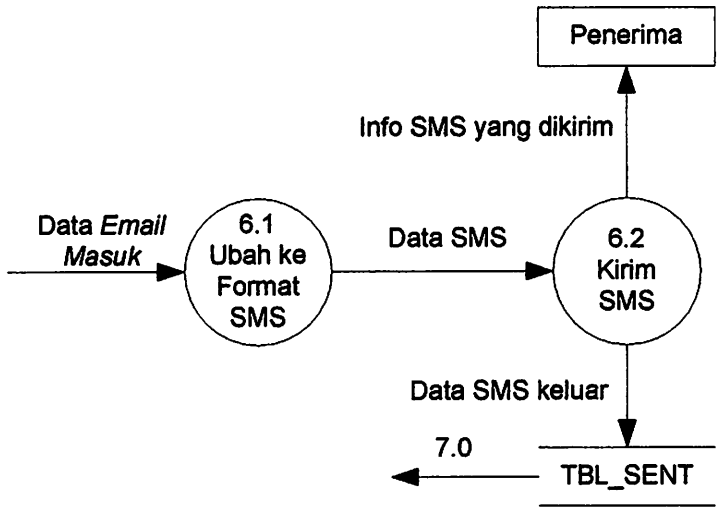
Gambar 3.9 di bawah ini merupakan gambar DFD Level 2 Proses 5 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses terima *E-mail* dalam bentuk SMS.



Gambar 3.9 DFD Level 2 Proses 5 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 5 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses terima *E-mail.* Tahap pertama yang dilakukan pada proses ini adalah membaca *E-mail* yang masuk kemudian aplikasi akan melakukan *request* ke *database* untuk memperoleh keterangan apakah orang yang mengirim *E-mail* dapat menggunakan layanan ini, jika orang itu tidak terdaftar maka aplikasi akan mengirimkan pesan peringatan ke pengirim bahwa pengirim tidak terdaftar sehingga tidak dapat menggunakan layanan aplikasi ini. Setelah data orang itu diperiksa langkah selanjutnya mengecek format *E-mail* yang dikirimkan apakah sesuai atau tidak, apabila tidak maka aplikasi akan mengirimkan pesan peringatan bahwa *E-mail* yang dikirimkan oleh orang itu salah format, jika format *E-mail* benar maka akan dilanjutkan ke proses *Forward to SMS. E-mail* yang masuk akan disimpan di TBL_EMAIL_INBOX.

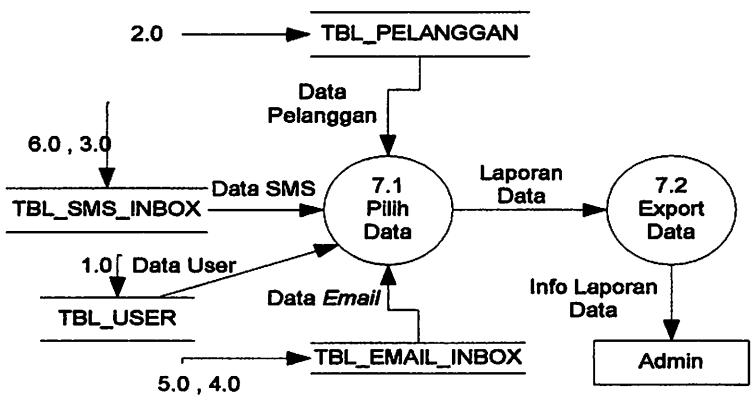
Gambar 3.10 di bawah ini merupakan gambar DFD Level 2 Proses 6 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses merubah data *E-mail* masuk ke format SMS.



Gambar 3.10 DFD Level 2 Proses 6 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 6 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses *Forward to SMS*, dimana data *E-mail* diubah ke format SMS, kemudian SMS akan dikirimkan ke penerima. SMS yang terkirim akan disimpan di TBL_SENT.

Gambar 3.11 di bawah ini merupakan gambar DFD Level 2 Proses 7 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang menggambarkan proses laporan data ke Admin.



Gambar 3.11 DFD Level 2 Proses 7 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

DFD Level 2 Proses 7 Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* menggambarkan proses laporan data ke Admin, dimana data-data yang dilaporkan ke Admin meliputi data *User*, data Pelanggan, data *SMS_Inbox*, dan data *E-mail_Inbox*.

3.2.4. Perancangan Struktur Basis Data (*Database*)

Database yang digunakan dalam aplikasi ini adalah Microsoft Access dengan pertimbangan selain *software database* bawaan dari Microsoft dan tidak perlu menginstal *software database* lainnya, Microsoft Access mudah digunakan untuk membangun sebuah aplikasi SMS *Gateway*.

Perancangan struktur *database* tidak terlepas dari perancangan masukan (*input*) dan keluaran (*output*), perancangan *input* dan *output* akan berpengaruh besar terhadap kelengkapan informasi yang akan dibuat. Berikut adalah gambaran dari rancangan tabel yang akan dibuat untuk memenuhi kebutuhan *database* aplikasi yang akan dibuat nantinya.

3.2.4.1. TBL_PELANGGAN

TBL_PELANGGAN berisi tentang data pelanggan yang sudah melakukan registrasi melalui SMS / data pelanggan yang di*input*kan secara manual oleh Admin / Operator. Tabel 3.1 di bawah ini memperlihatkan desain dari TBL_PELANGGAN yang digunakan untuk menyimpan data-data pelanggan yang sudah melakukan registrasi.

Tabel 3.1 TBL_PELANGGAN

Nama field	Tipe Data	Ukuran Data	Primary key
ID	AutoNumber	Long Integer	*
VCH_FULLNAME	Text	100	

VCH_GENDER	Text	1	
VCH_PHONE_NUMBER	Text	50	*
VCH_HOME_ADDRESS	Text	255	
VCH_EMAIL_ADDRESS	Text	50	*
VCH_EMAIL_PASSWORD	Text	255	
VCH_STATUS	Text	50	
DTM_DATE_REGISTER	Date/Time	General Date	

3.2.4.2. TBL_USER

TBL_USER berisi tentang data-data *user* / pengoperasi aplikasi yaitu Admin / Operator. Tabel 3.2 di bawah ini memperlihatkan desain dari TBL_USER yang digunakan untuk menyimpan data-data *user* / pengoperasi aplikasi.

Tabel 3.2 TBL_USER

Nama field	Tipe Data	Ukuran Data	Primary key
ID	AutoNumber	Long Integer	*
VCH_USER_ID	Text	25	
VCH_PASSWORD	Text	255	
VCH_ROLE	Text	25	
VCH_FULLNAME	Text	100	
VCH_GENDER	Text	1	
VCH_EMAIL_ADDRESS	Text	50	*
VCH_HOME_ADDRESS	Text	255	
VCH_PHONE_NUMBER	Text	50	*

DTM_DATE_REGISTER	Date / Time	General Date	

3.2.4.3. TBL_SMS_INBOX

TBL_SMS_INBOX berisi tentang data SMS masuk yang ada pada HP SMS *Gateway*. Tabel 3.3 di bawah ini memperlihatkan desain dari TBL_SMS_INBOX yang digunakan untuk menyimpan data-data SMS masuk dari HP SMS *Gateway*.

Tabel 3.3 TBL_SMS_INBOX

Nama field	Tipe Data	Ukuran Data	Primary key
ID	AutoNumber	Long Integer	*
VCH_PHONE_NUMBER	Text	50	*
VCH_DATE	Text	40	
VCH_ISI_SMS	Text	160	
VCH_JENIS	Text	10	
VCH_PROCESS	Text	5	

3.2.4.4. TBL_SENT

TBL_SENT berisi tentang data *E-mail* maupun SMS yang terkirim. Tabel 3.4 di bawah ini memperlihatkan desain dari TBL_SENT yang digunakan untuk menyimpan hasil dari mengirim *E-mail* maupun SMS secara otomatis.

Tabel 3.4 TBL_SENT

Nama field	Tipe Data	Ukuran Data	Primary key
ID	AutoNumber	Long Integer	*
VCH_EMAIL_ADDRESS	Text	50	*

DTM_EMAIL_DATE	Date / Time	General Date	
VCH_TO_NAME	Text	50	
VCH_TO_EMAIL_ADDRESS	Text	50	
VCH_SUBJECT	Text	255	
VCH_BODY	Text	255	
VCH_STATUS	Text	50	
VCH_PHONE_NUMBER	Text	50	*

3.2.4.5. TBL_EMAIL_INBOX

TBL_EMAIL_INBOX berisi tentang data *E-mail* yang masuk hasil *download E-mail* dari webmail gmail.com. Tabel 3.5 di bawah ini memperlihatkan desain dari TBL_EMAIL_INBOX yang digunakan untuk menyimpan data-data *E-mail* hasil *download E-mail* dari webmail GMAIL.

Tabel 3.5 TBL_EMAIL_INBOX

Nama Field	Tipe Data	Ukuran Data	Primary Key
ID	AutoNumber	Long Integer	*
NMB_IDX	Number	Long Integer	
VCH_EMAIL_ADDRESS	Text	50	*
DTM_EMAIL_DATE	Text	255	
VCH_FROM_NAME	Text	50	
VCH_FROM_EMAIL_ADDRESS	Text	50	
VCH_SUBJECT	Text	255	
VCH_BODY	Text	255	

VCH_STATUS	Text	50	
VCH_PROCESS	Text	5	

3.2.4.6. TBL_LOGIN

TBL_LOGIN berisi tentang data saat *login* / siapa saja yang pernah menggunakan aplikasi. Tabel 3.6 di bawah ini memperlihatkan desain dari TBL_LOGIN yang digunakan untuk menyimpan data siapa saja yang pernah menggunakan aplikasi.

Tabel 3.6 TBL_LOGIN

Nama Field	Tipe Data	Ukuran Data	Primary Key
ID	AutoNumber	Long Integer	*
DTM_LOGIN	Date / Time	General Date	
DTM_LOGOUT	Date / Time	General Date	
VCH_USER_ID	Text	25	*
VCH_ROLE	Text	25	
VCH_FULLNAME	Text	100	
VCH_STATUS	Text	25	

3.2.4.7. TBL_PHONEBOOK

TBL_PHONEBOOK berisi tentang daftar buku telepon pada HP SMS *Gateway*. Tabel 3.7 di bawah ini memperlihatkan desain dari TBL_PHONEBOOK yang digunakan untuk menyimpan / meng*copy* daftar buku telepon yang ada pada HP SMS *Gateway*.

Tabel 3.7 TBL_PHONEBOOK

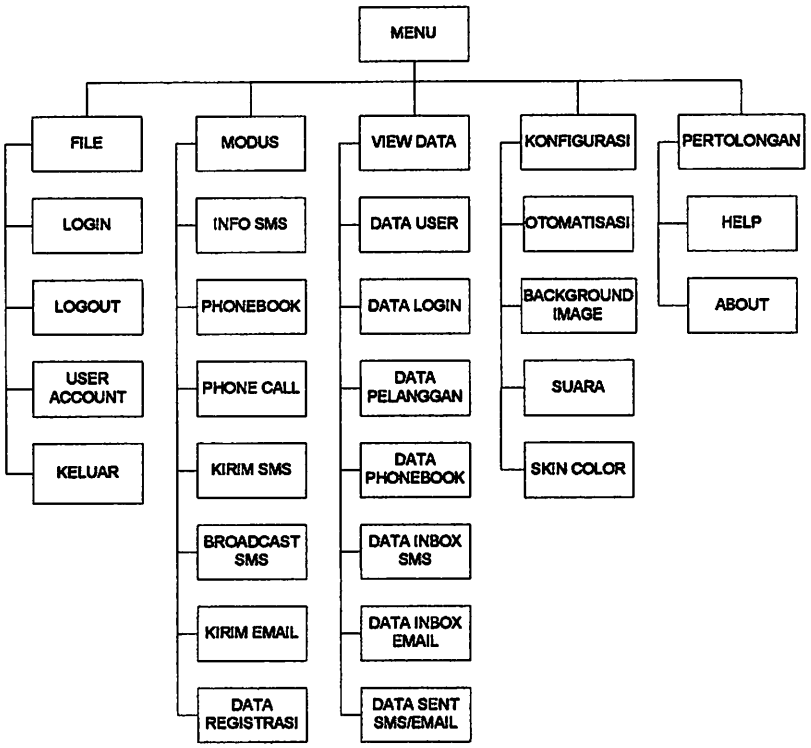
Nama Field	Tipe Data	Ukuran Data	Primary Key
ID	AutoNumber	Long Integer	*
VCH_FULLNAME	Text	100	
VCH_PHONE_NUMBER	Text	50	*
VCH_STATUS	Text	50	

3.2.5. Perancangan Antar Muka

Perancangan *interface* merupakan rancang bangun dari interaksi *user* dengan komputer. Interaksi ini dapat berupa proses peng*input*an data ke sistem, *update* data dan menjalankan aplikasi.

3.2.5.1. Struktur Menu

Struktur menu program memberikan gambaran mengenai modul-modul yang menyusun program dan menggambarkan hirarki diantara modul-modul tersebut. Berikut ini adalah Gambar 3.12 yang menggambarkan struktur menu Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail.*



Gambar 3.12 Struktur Menu Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

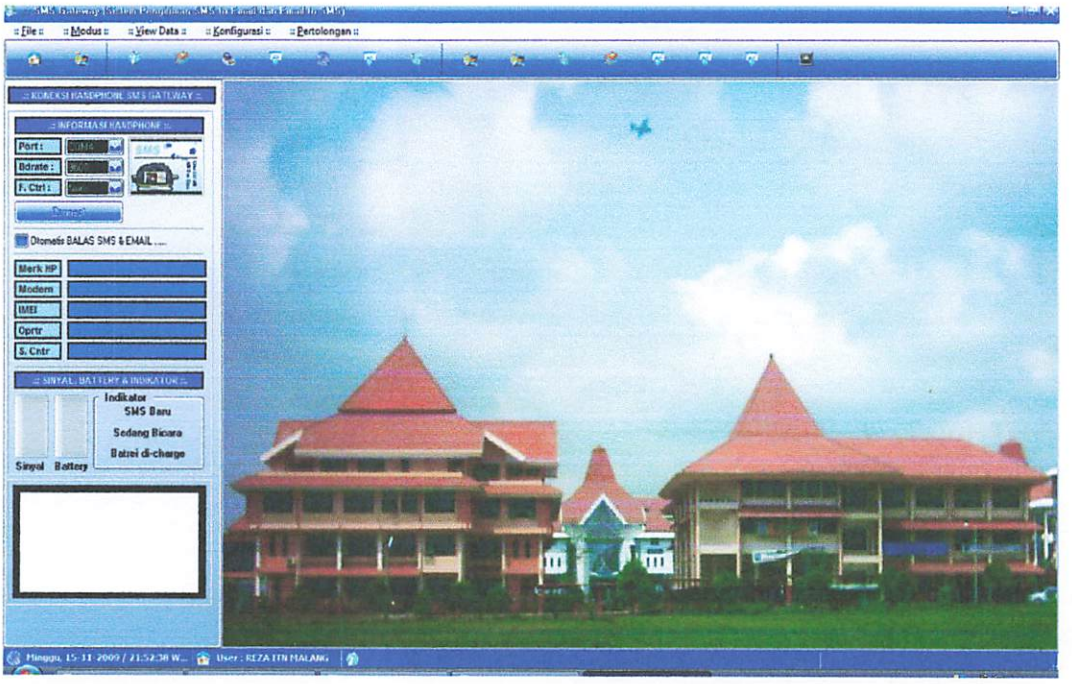
a. Antarmuka *Login*

Antarmuka *login* merupakan antarmuka pertama yang akan dihadapi oleh Admin / Operator dimana Admin / Operator melakukan proses *input* data yang berupa *user role*, *user id*, & *password* pada *textbox* diikuti dengan tombol OK. Gambar 3.13 menampilkan form antarmuka *login*:

Gambar 3.13 Form Antarmuka Login Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

b. Antarmuka Halaman Utama

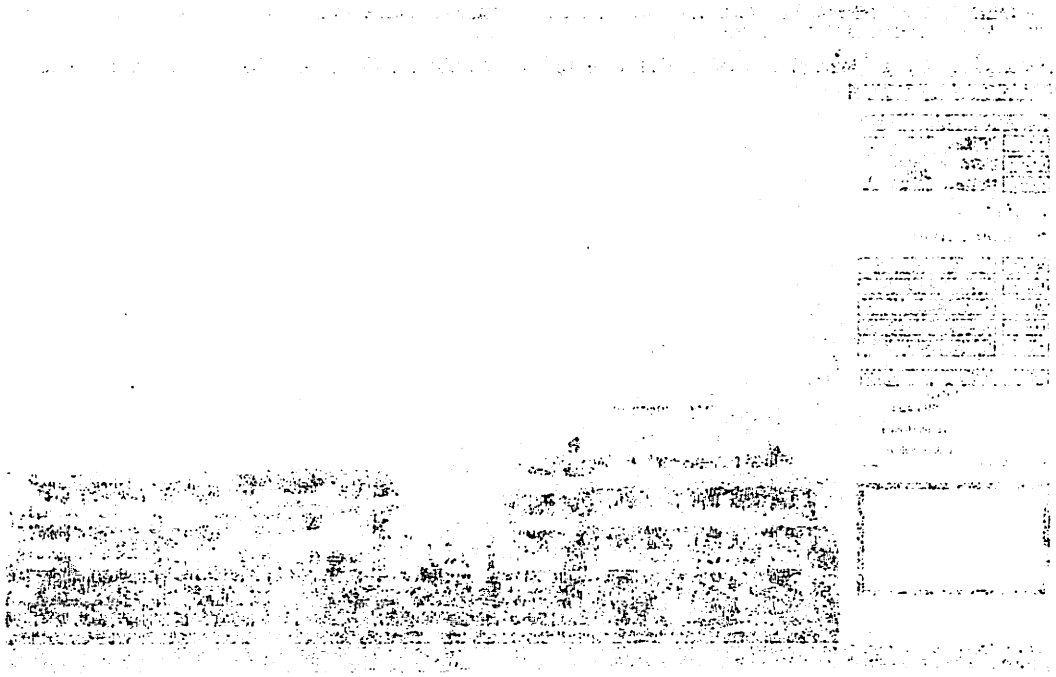
Antarmuka halaman utama merupakan antarmuka ke dua setelah Admin / Operator berhasil melakukan *Login*. Di dalamnya terdapat berbagai macam menu yang bisa digunakan oleh Admin / Operator untuk menjalankan aplikasi, serta informasi mengenai HP yang digunakan sebagai HP SMS *Gateway*, seperti informasi merk HP, modem, IMEI, operator, service center, sinyal, battery, dan indikator. Gambar 3.14 menampilkan form antarmuka halaman utama:

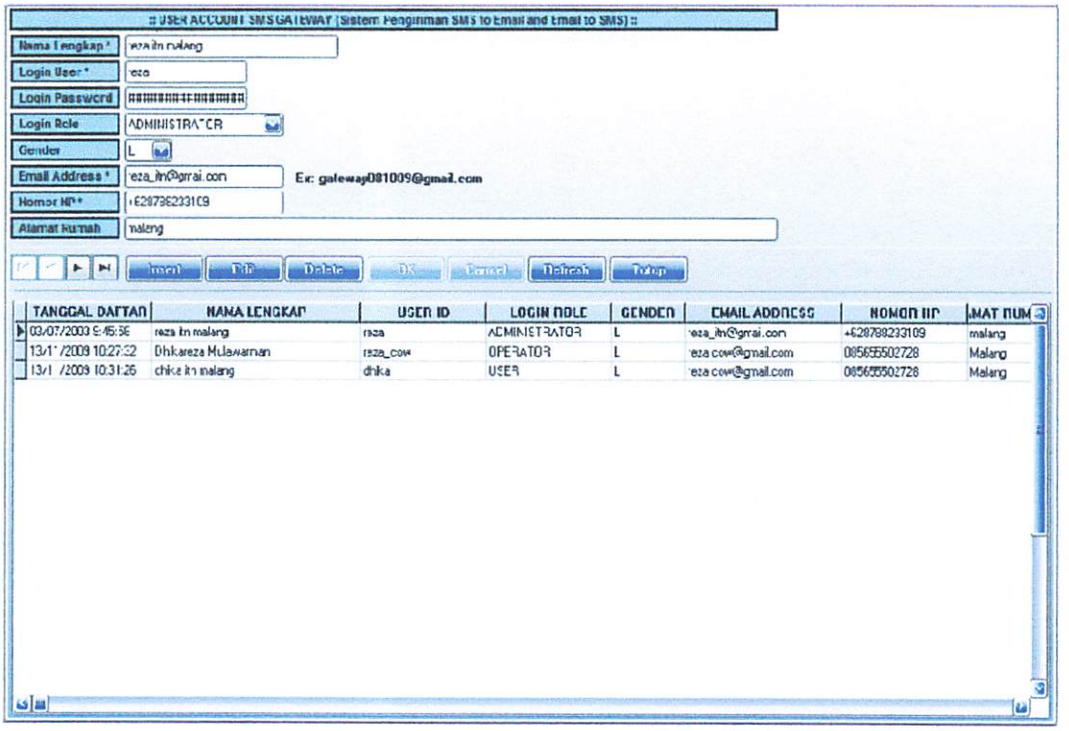


Gambar 3.14 Form Antarmuka Halaman Utama Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

c. Antarmuka User Account

Antarmuka User Account digunakan untuk memasukkan, merubah, dan menghapus data mengenai *user* / pengoperasi aplikasi. Gambar 3.15 menampilkan form antarmuka user account:

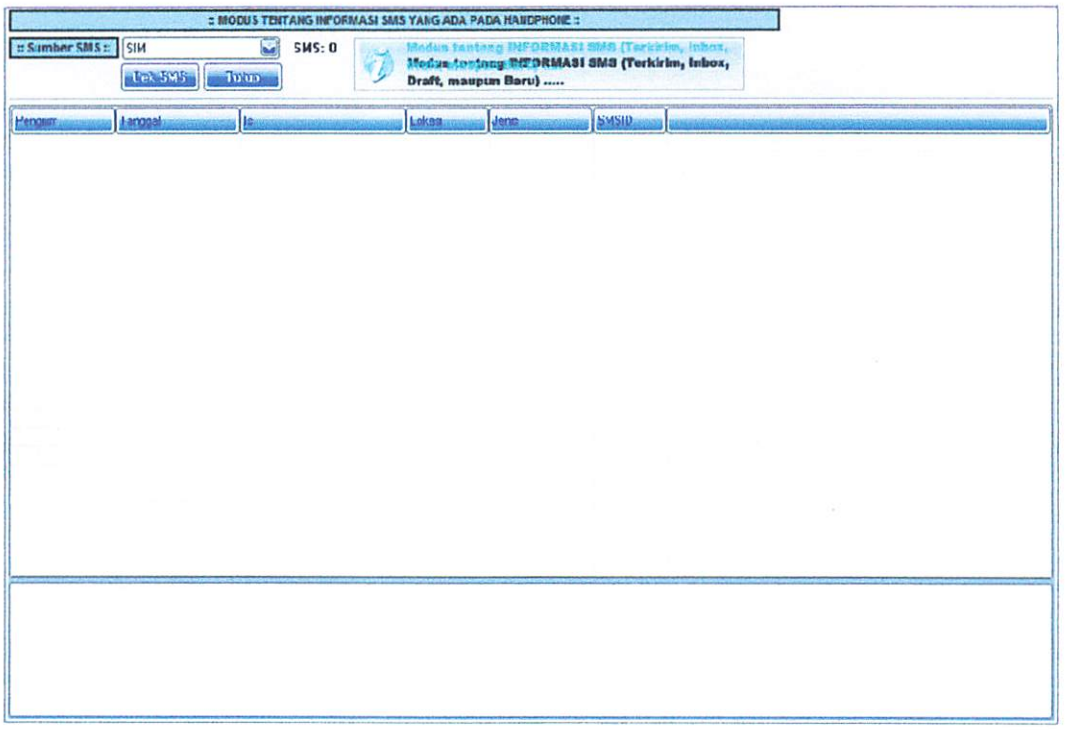




Gambar 3.15 Form Antarmuka User Account Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

d. Antarmuka Modus Informasi SMS

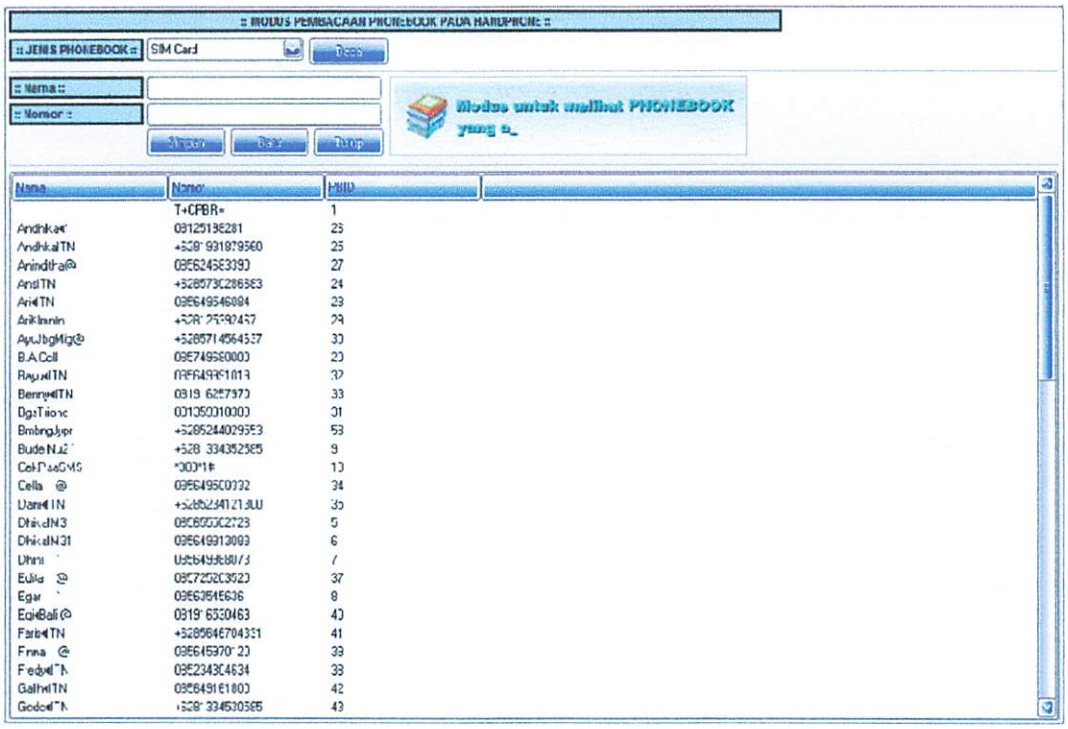
Antarmuka Modus Informasi SMS menampilkan data SMS yang ada pada *Inbox* HP SMS *Gateway*. Sumber SMS didapat dari memory kartu SIM / memory handphone / ke dua-duanya. Gambar 3.16 menampilkan form antarmuka modus informasi SMS:



Gambar 3.16 Form Antarmuka Modus Informasi SMS Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

e. Antarmuka Modus Pembacaan Phonebook

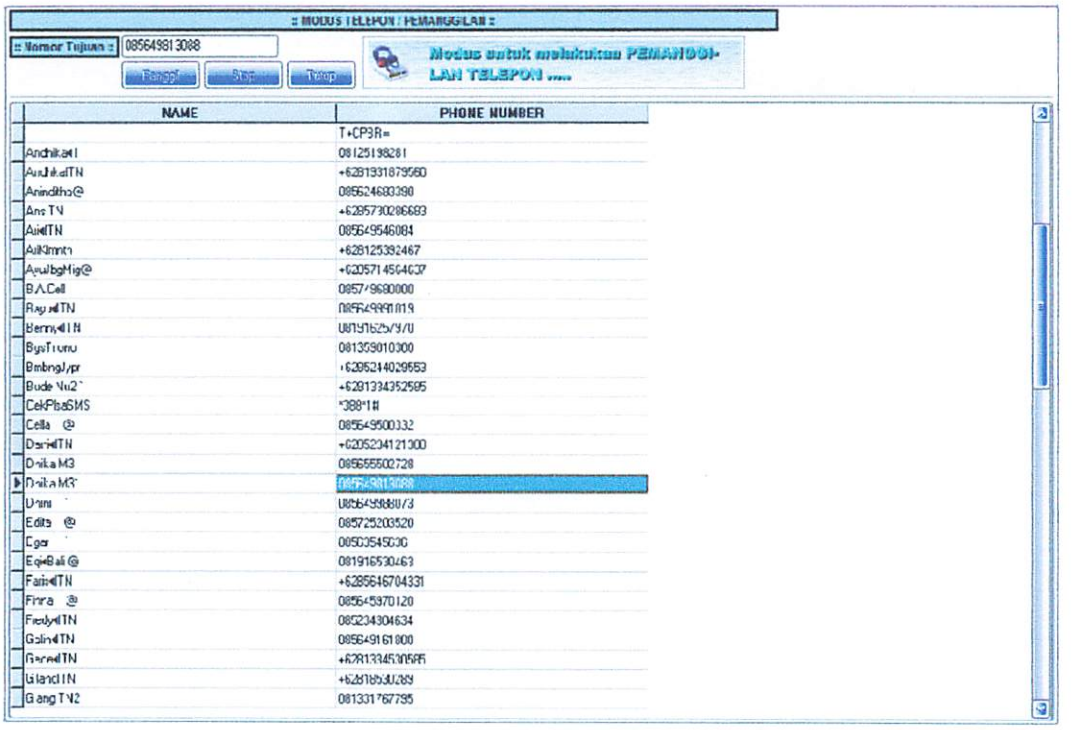
Antarmuka Modus Pembacaan Phonebook menampilkan daftar buku telepon dari HP SMS *Gateway*. Gambar 3.17 menampilkan form antarmuka modus pembacaan *phonebook*:



Gambar 3.17 Form Antarmuka Modus Pembacaan Phonebook Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

f. Antarmuka Modus Telepon / Pemanggilan

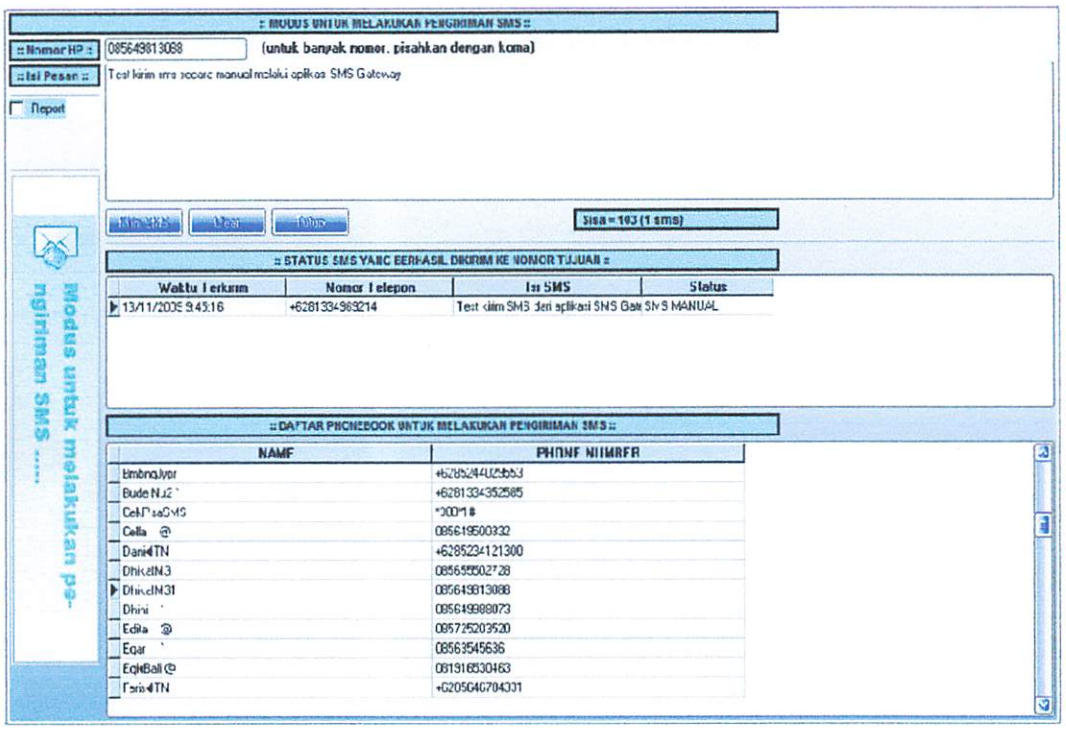
Antarmuka Modus Telepon / Pemanggilan adalah antarmuka yang tujuannya untuk melakukan panggilan / telepon keluar. Gambar 3.18 menampilkan form antarmuka modus telepon / pemanggilan:



Gambar 3.18 Form Antarmuka Modus Telepon / Pemanggilan Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

g. Antarmuka Modus Pengiriman SMS

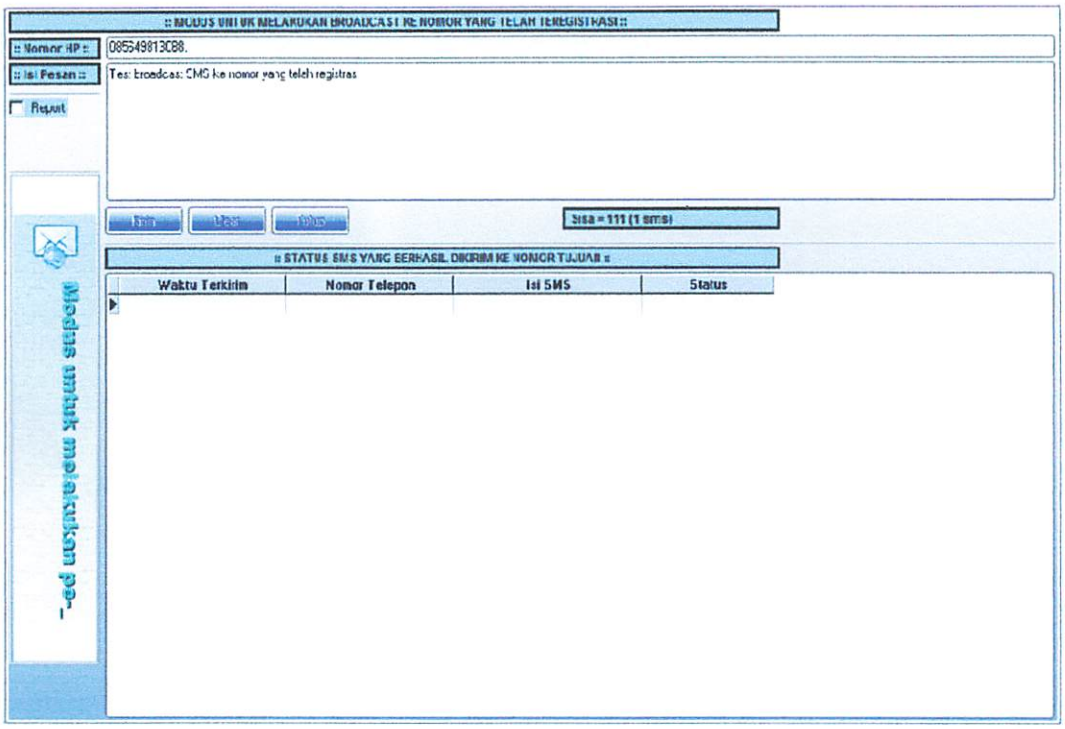
Antarmuka Modus Pengiriman SMS adalah antarmuka yang digunakan untuk mengirim SMS secara manual. Gambar 3.19 menampilkan form antarmuka modus pengiriman SMS:



Gambar 3.19 Form Antarmuka Modus Pengiriman SMS Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

h. Antarmuka Modus Broadcast Pengiriman SMS

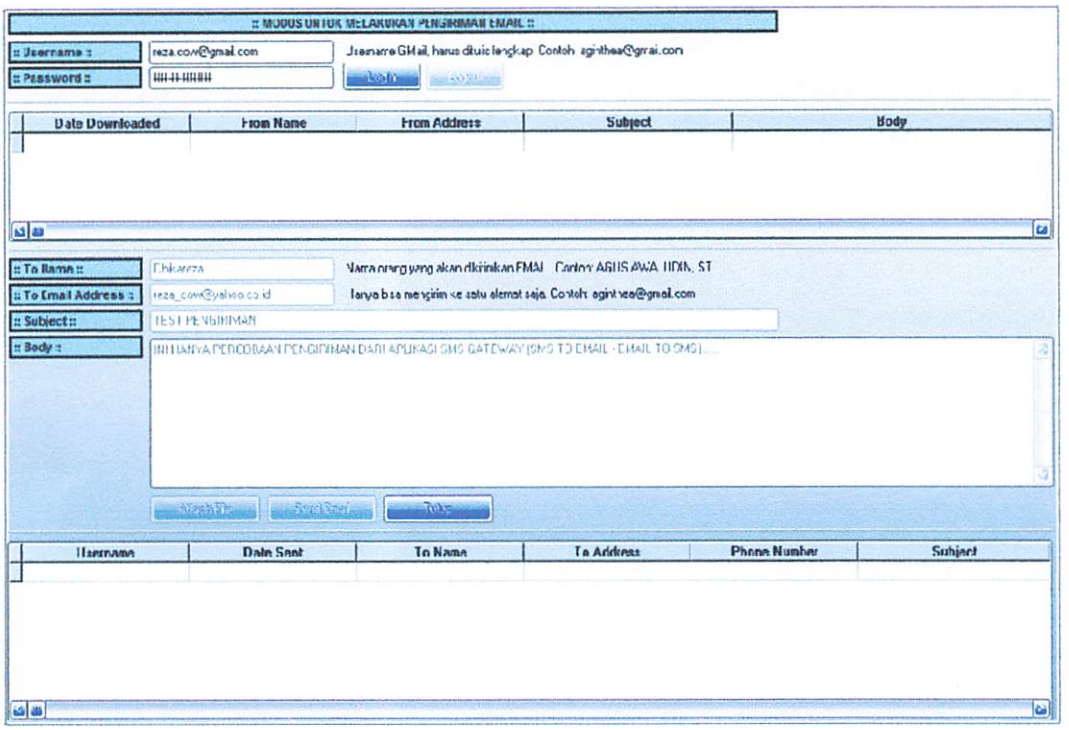
Antarmuka Modus Broadcast Pengiriman SMS adalah antarmuka yang tujuannya untuk mengirimkan SMS secara manual ke banyak nomor handphone yang telah teregistrasi. Gambar 3.20 menampilkan form antarmuka modus broadcast pengiriman SMS:



Gambar 3.20 Form Antarmuka Modus Broadcast Pengiriman SMS Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

i. Antarmuka Modus Pengiriman Email

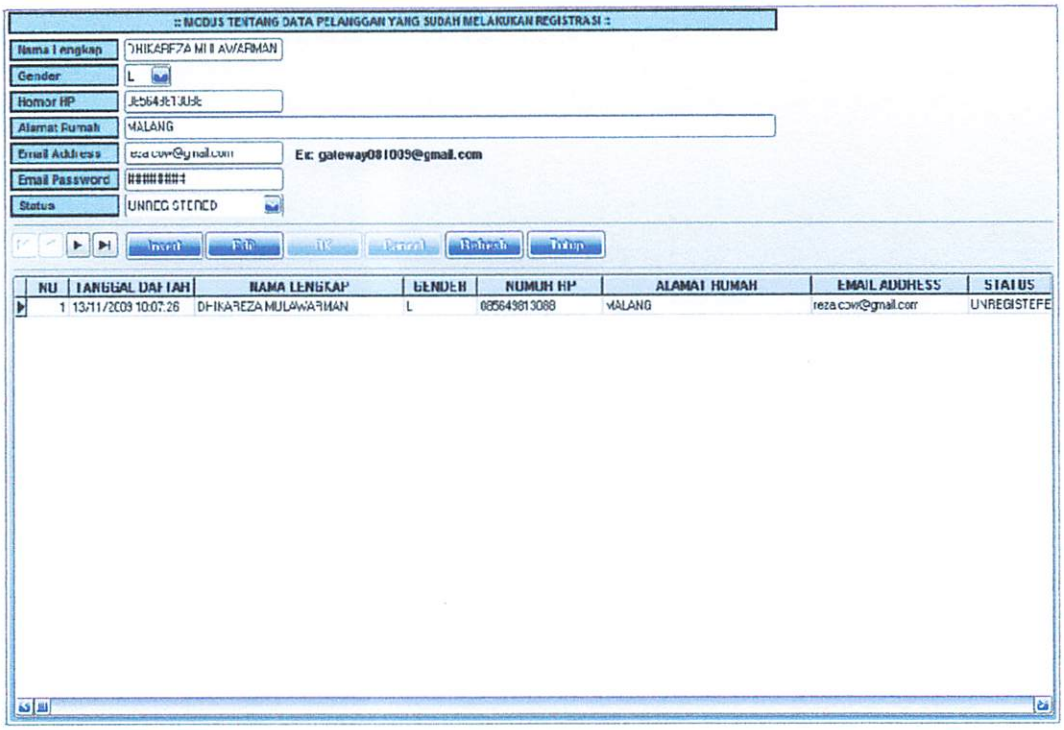
Antarmuka Modus Pengiriman Email adalah antarmuka yang digunakan untuk mengirim *E-mail* secara manual. Gambar 3.21 menampilkan form antarmuka modus pengiriman *E-mail*:



Gambar 3.21 Form Antarmuka Modus Pengiriman Email Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

j. Antarmuka Modus Data Pelanggan

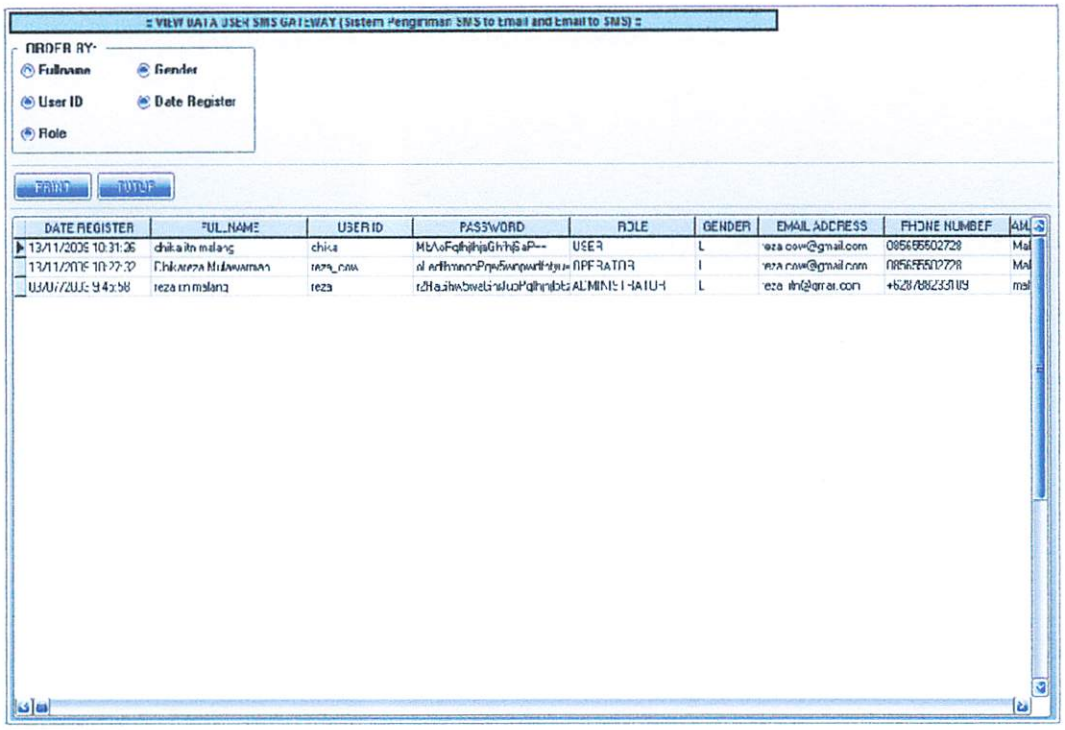
Antarmuka Modus Data Pelanggan menunjukkan informasi tentang data pelanggan yang sudah melakukan registrasi melalui SMS. Dalam antarmuka ini, Admin / Operator bisa memasukkan dan merubah data pelanggan secara manual. Gambar 3.22 menampilkan form antarmuka modus data pelanggan.



Gambar 3.22 Form Antarmuka Modus Data Pelanggan Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

k. Antarmuka View Data User

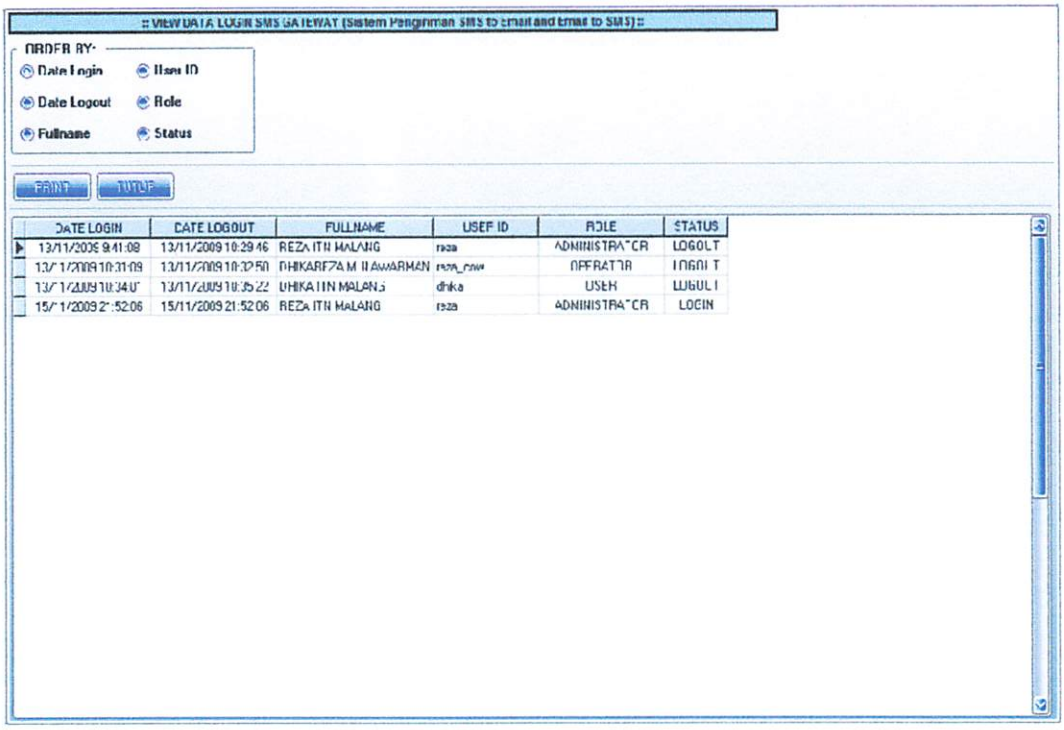
Antarmuka View Data User digunakan untuk melihat informasi mengenai data-data *user* / pengoperasi aplikasi yaitu Admin / Operator. Gambar 3.23 menampilkan form antarmuka *view* data *user*:



Gambar 3.23 Form Antarmuka View Data User Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

l. Antarmuka View Data Login

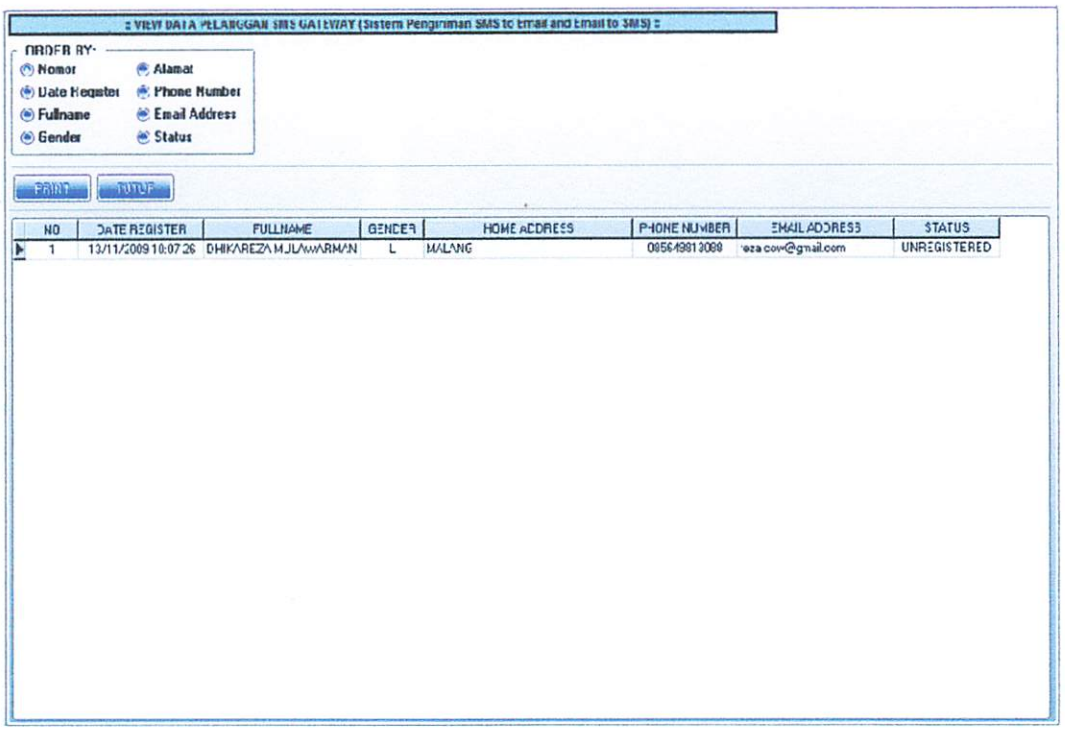
Antarmuka View Data Login digunakan untuk melihat data saat *login* / siapa saja yang pernah menggunakan aplikasi. Gambar 3.24 menampilkan form antarmuka *view* data *login*:



Gambar 3.24 Form Antarmuka View Data Login Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

m. Antarmuka View Data Pelanggan

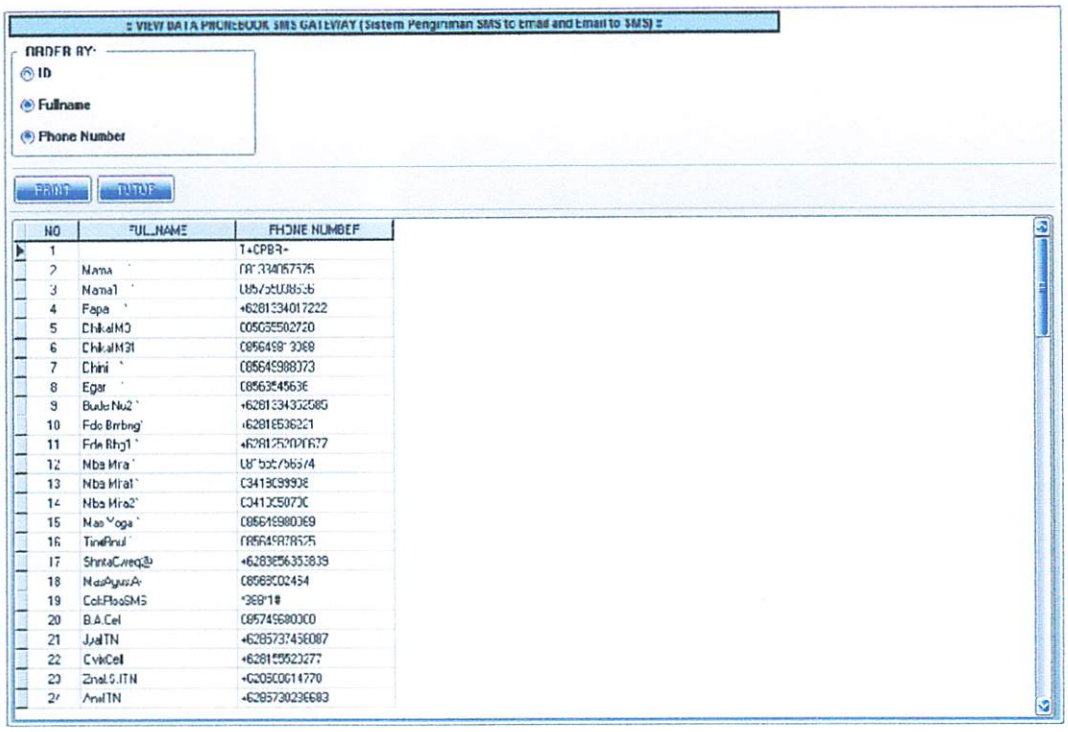
Antarmuka View Data Pelanggan digunakan untuk melihat data orang yang sudah melakukan registrasi. Gambar 3.25 menampilkan form antarmuka *view* data pelanggan:



Gambar 3.25 Form Antarmuka View Data Pelanggan Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

n. Antarmuka View Data Phonebook

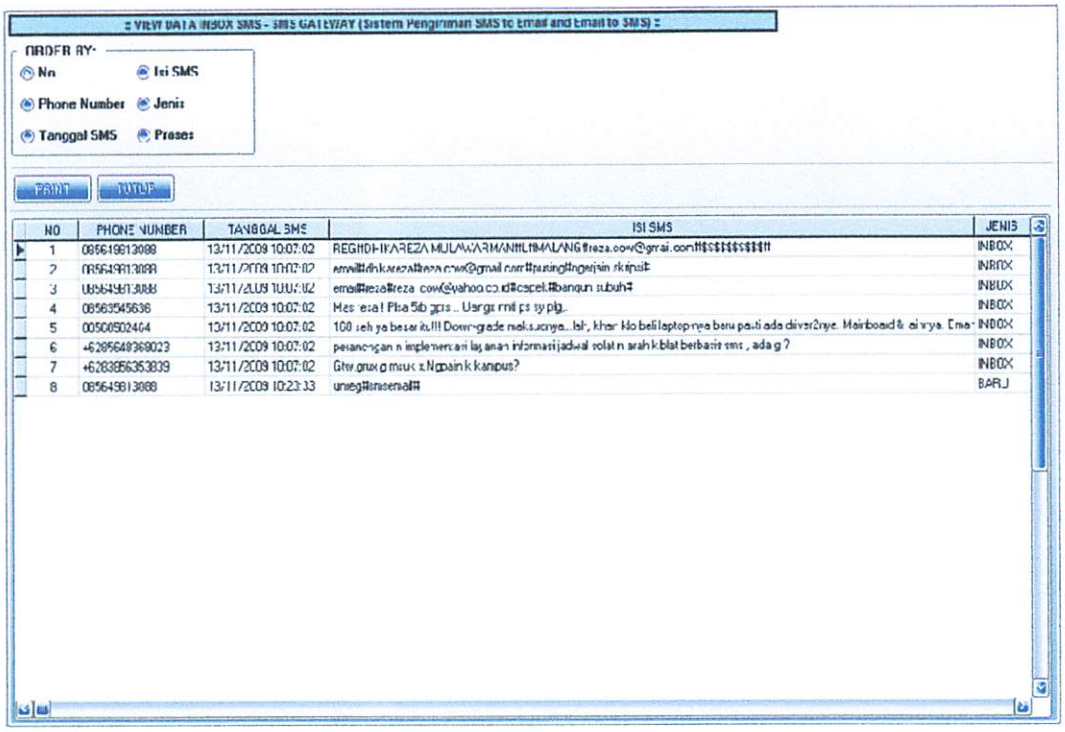
Antarmuka View Data Phonebook digunakan untuk melihat daftar buku telepon pada HP SMS *Gateway*. Gambar 3.26 menampilkan form antarmuka *view* data *phonebook*:



Gambar 3.26 Form Antarmuka View Data Phonebook Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

o. Antarmuka View Data Inbox SMS

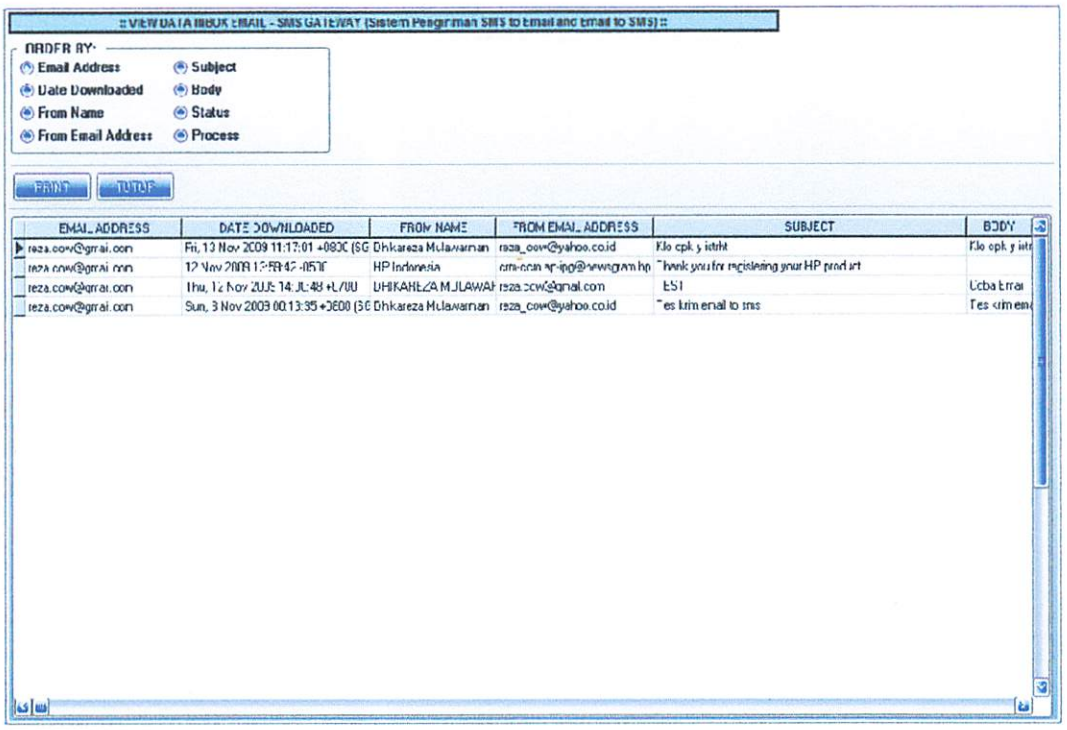
Antarmuka View Data Inbox SMS digunakan untuk melihat data SMS masuk yang ada pada HP SMS *Gateway*. Gambar 3.27 menampilkan form antarmuka *view* data *inbox* sms:



Gambar 3.27 Form Antarmuka View Data Inbox SMS Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

p. Antarmuka View Data Inbox Email

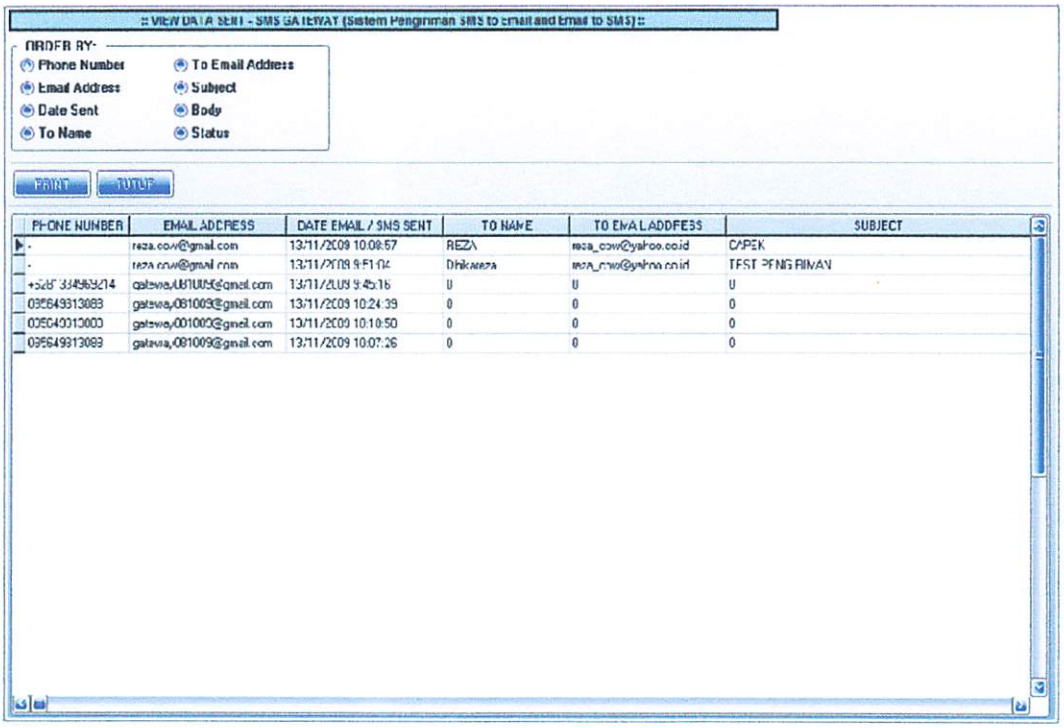
Antarmuka View Data Inbox Email digunakan untuk melihat data *E-mail* yang masuk hasil *download E-mail* dari webmail gmail.com. Gambar 3.28 menampilkan form antarmuka *view* data *inbox E-mail*:



Gambar 3.28 Form Antarmuka View Data Inbox Email Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

q. Antarmuka View Data Sent

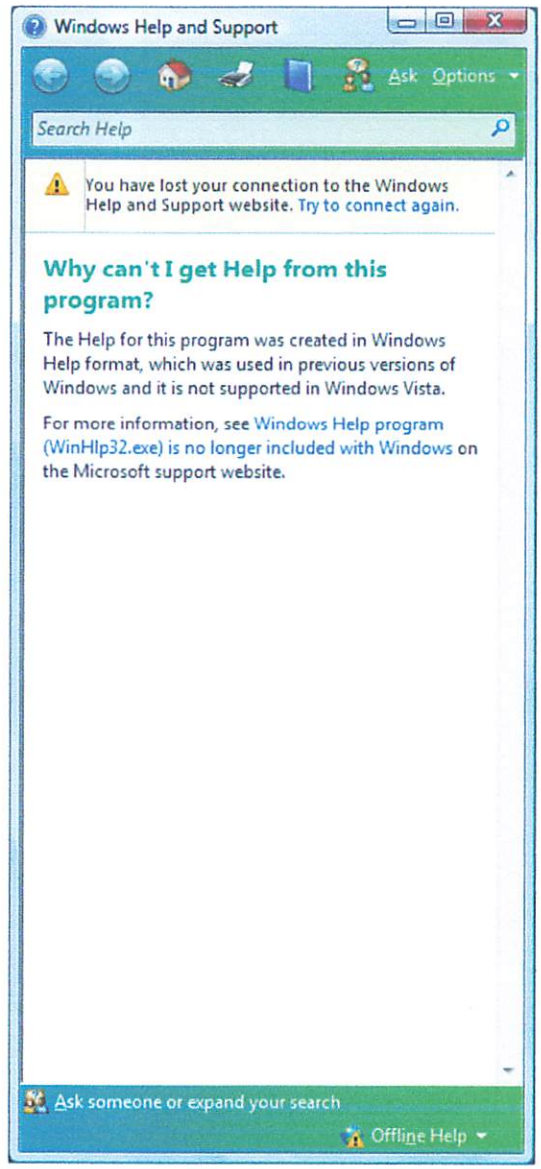
Antarmuka View Data Sent digunakan untuk melihat data *E-mail* maupun SMS yang terkirim. Gambar 3.29 menampilkan form antarmuka *view* data *sent*:



Gambar 3.29 Form Antarmuka View Data Sent Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

r. Antarmuka Jendela Help

Antarmuka Jendela Help memberikan pertolongan kepada *user* / pengoperasi aplikasi untuk mengoperasikan Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail.* Gambar 3.30 menampilkan antarmuka jendela *help*:



Gambar 3.30 Antarmuka Jendela Help Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

s. Antarmuka Form About

Antarmuka Form About memberikan informasi tentang nama aplikasi dan pembuat aplikasi. Gambar 3.31 menampilkan antarmuka form *about*:

Gambar 3.31 Antarmuka Form About Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail*

3.3. Analisis Kebutuhan Hardware & Software

Kebutuhan *hardware* dan *software* yang digunakan untuk membangun sistem tersebut antara lain :

3.3.1. Hardware

a) Processor Intel Pentium IV 2.4 MHz

b) CD Drive

c) Memory 256 MB

d) Video Card 64 MB

e) Minimum Hard Disk 100 MB

f) Monitor

3.3.2. Software

a) Microsoft Access 2007

b) Borland Delphi 7

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Pengujian Sistem

Pengujian ini bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini bila diterapkan dalam sistem. Pengujian ini juga untuk mencari letak *error* atau kesalahan program dalam aplikasi.

4.1.1. Metode Pengoperasian

Pada pengoperasian aplikasi ini diharapkan pengoperasi aplikasi memenuhi prosedur yang ditetapkan. Prosedur yang ditetapkan adalah sebagai berikut :

1. Mengoneksikan PC / laptop dengan HP dan internet sebelum menjalankan aplikasi.
2. *Login* sesuai dengan hak akses yang dimiliki pengoperasi aplikasi.
3. Menyetting *Port Number* & *Baudrate* HP pada aplikasi, kemudian meng-klik tombol *Connect* agar aplikasi terkoneksi dengan HP dan mencentang pada *CheckBox* "Otomatis Balas SMS & *E-mail*" agar aplikasi membalas otomatis SMS-SMS / *E-mail* - *E-mail* yang masuk ke dalam aplikasi.

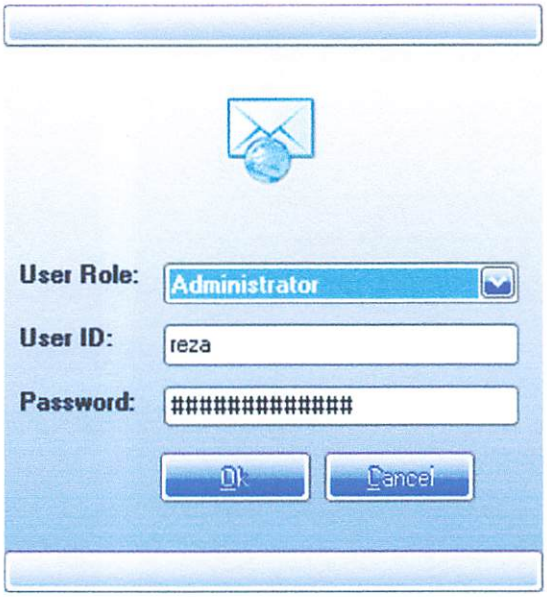
4.2. Pengujian Aplikasi *Server*

Pengujian pada aplikasi *server* dilakukan agar kemungkinan terjadi kesalahan pada aplikasi *server* dapat diidentifikasikan sejak awal. Pengujian dilakukan pada semua modul-modul yang terdapat pada aplikasi *server*.

4.2.1. Pengujian Form *Login*

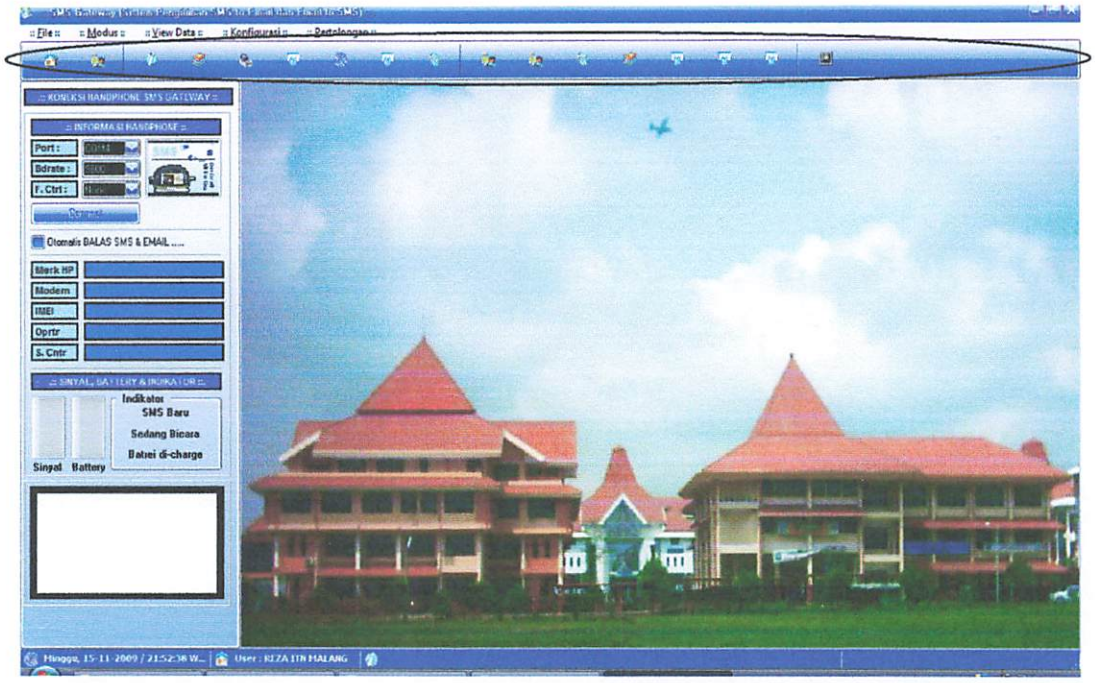
Pengujian disini tujuannya untuk memverifikasi data *login* dan menampilkan Form Halaman Utama sesuai dengan hak akses yang dimiliki

pengoperasi aplikasi. Hak akses pengoperasi aplikasi dibedakan menjadi 3 bagian, yaitu Administrator, Operator, dan *User*. Apabila pengoperasi aplikasi *login* sebagai Administrator / Operator, maka dia bisa melakukan apapun pada aplikasi tersebut, tetapi kalau *login* sebagai *User*, dia hanya bisa *view* data, *add phonebook*, & *phonecall* saja. Gambar 4.1 di bawah ini menggambarkan proses *Login* sebagai Administrator.



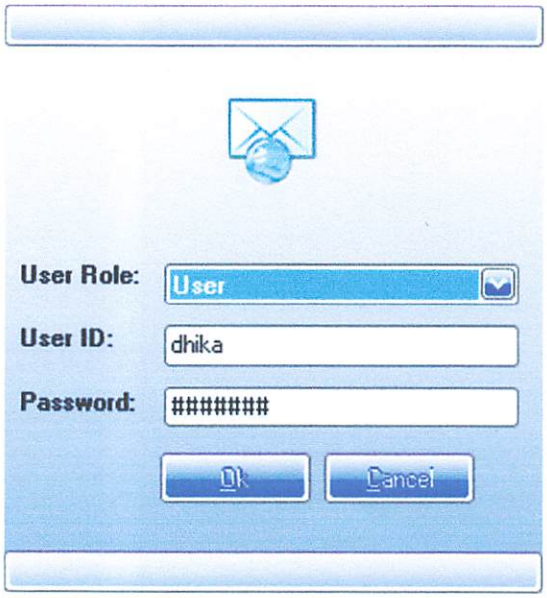
Gambar 4.1 Proses *Login* sebagai Administrator

Gambar 4.2 di bawah ini menampilkan Form Halaman Utama dari Administrator.



Gambar 4.2 Halaman Utama Administrator

Gambar 4.3 di bawah ini menggambarkan proses *Login* sebagai *User* biasa.



Gambar 4.3 Proses *Login* Sebagai *User*

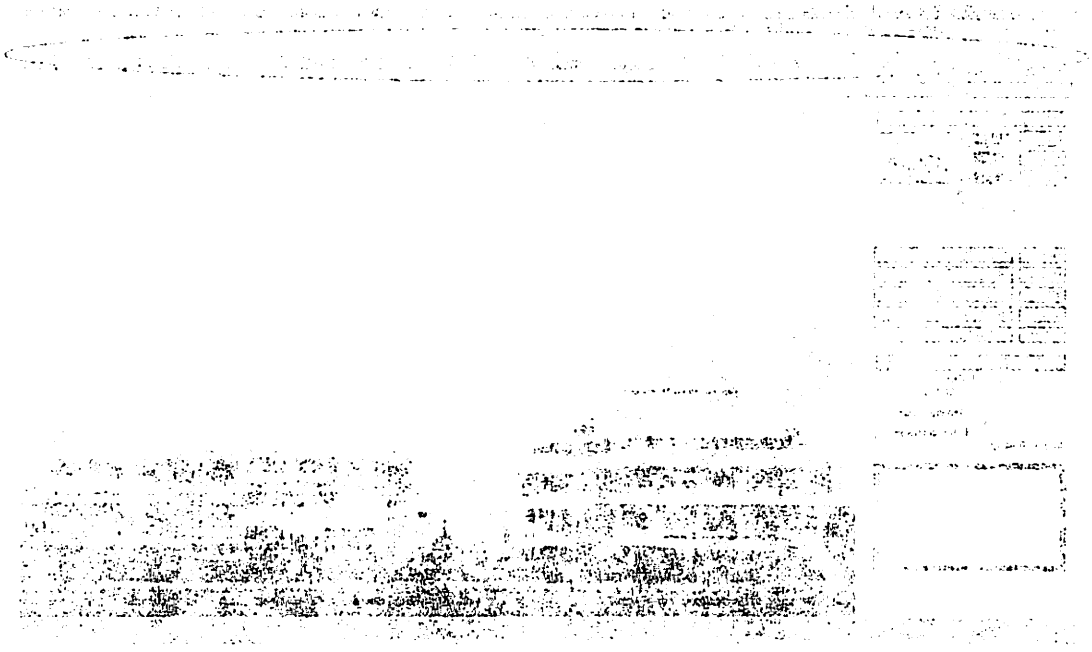
Gambar 4.4 di bawah ini menampilkan Form Halaman Utama dari *User* biasa.

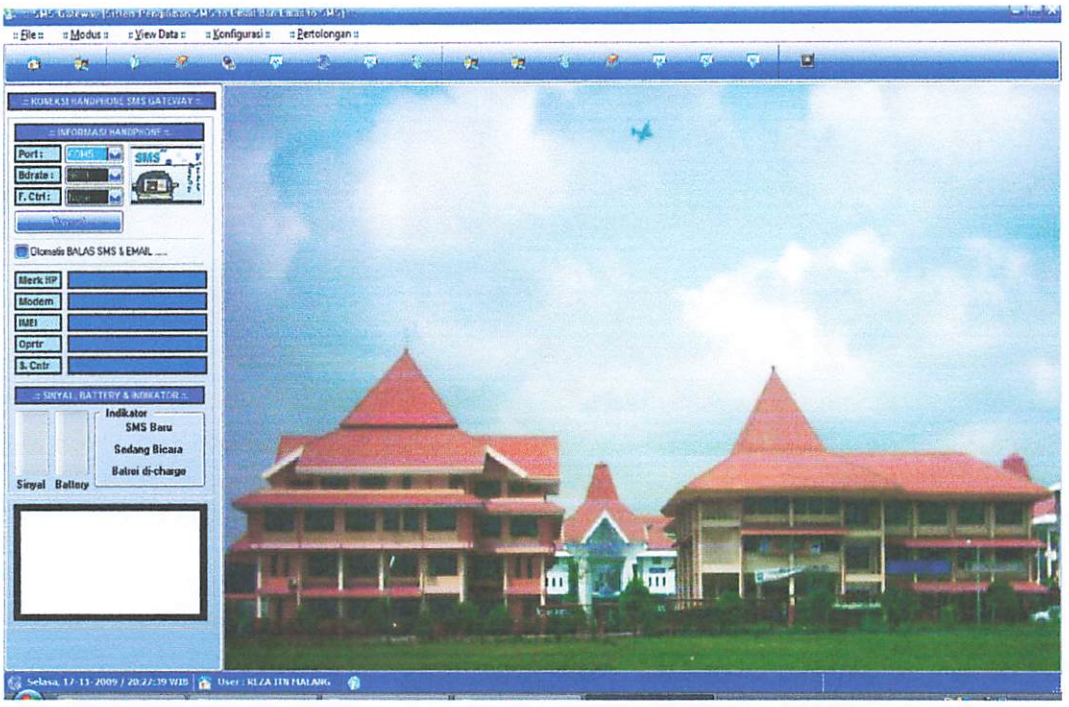
Gambar 4.4 Halaman Utama *User*

Untuk pengujian verifikasi data *login* dan menampilkan Form Halaman Utama sesuai dengan hak akses yang dimiliki pengoperasi aplikasi pada form *Login* telah sukses dan berjalan dengan baik.

4.2.2. Pengujian Form Halaman Utama

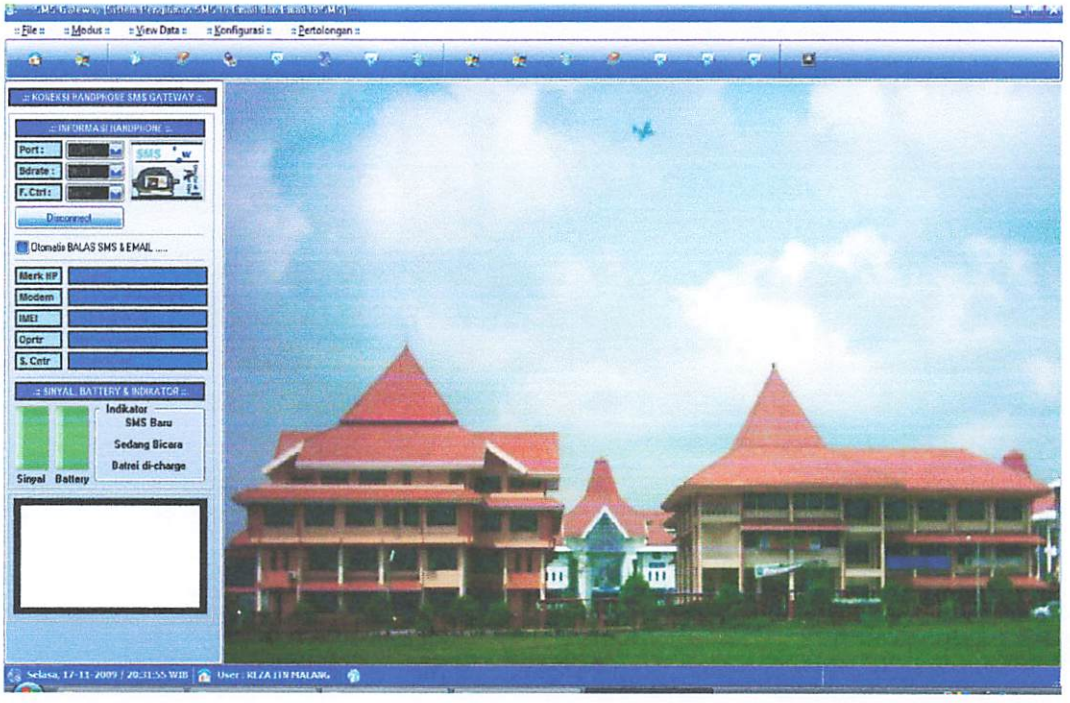
Untuk Form Halaman Utama, pengujian dilakukan dengan menguji semua menu yang ada didalam form dan mengoneksikan form dengan HP SMS *Gateway*. Agar semua menu-menu dapat diuji, maka pengoperasi aplikasi harus *Login* dengan hak akses sebagai Administrator / Operator. Gambar 4.5 di bawah ini menampilkan Form Halaman Utama Administrator / Operator sebelum terkoneksi dengan HP SMS *Gateway*.





Gambar 4.5 Form Halaman Utama Administrator / Operator Sebelum Terkoneksi dengan HP SMS *Gateway*

Gambar 4.6 di bawah ini menampilkan Form Halaman Utama Administrator / Operator sesudah terkoneksi dengan HP SMS *Gateway*.



Gambar 4.6 Form Halaman Utama Administrator / Operator Sesudah Terkoneksi dengan HP SMS *Gateway*

Agar dapat mengoneksikan HP SMS *Gateway* dengan aplikasi, pengoperasi aplikasi harus menyetting *Port Number & Baudrate* terlebih dahulu kemudian klik button *Connect*. Dalam pengujian kali ini, *Port Number* di*setting* pada COM 5 & *Baudrate* di*setting* pada nilai 9600. Setelah *Port Number*, *Baudrate* di*setting* dan mengklik *button Connect*, akan muncul informasi mengenai HP SMS *Gateway* yang meliputi:

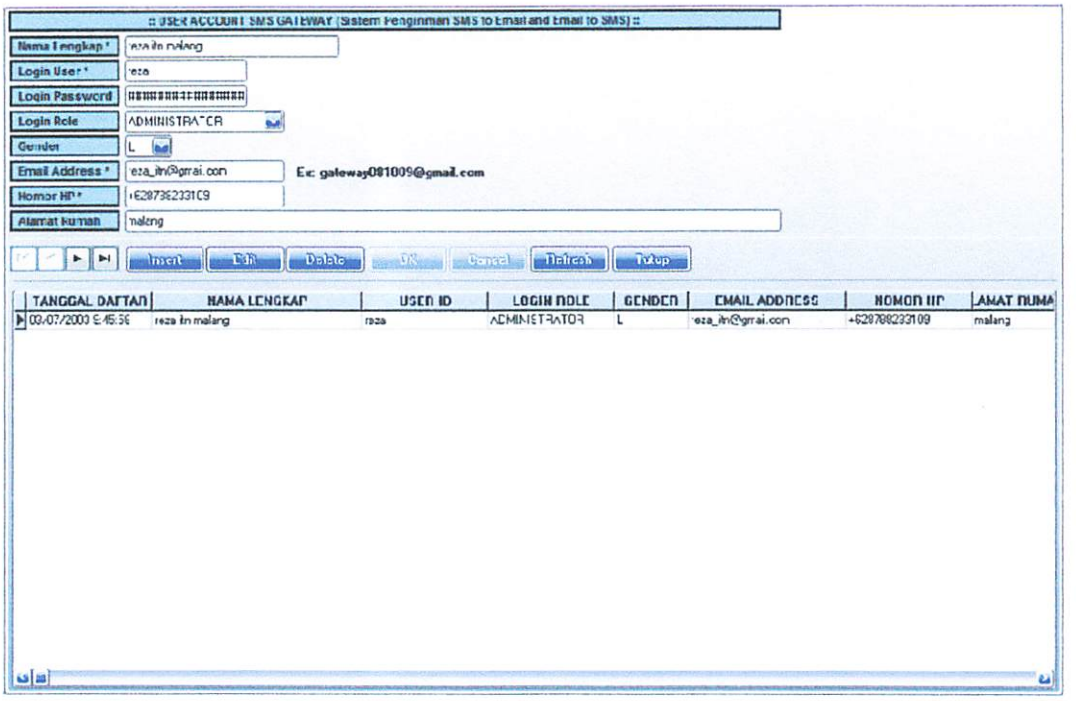
1. Merk HP : Siemens
2. Modem : Gipsy Soft Protocolstack
3. IMEI : 351792008134345
4. Oprtr : "IND SATELINDOCEL"
5. S. Cntr : +62855000000

Serta informasi mengenai level Sinyal, level *Battery*, dan Indikator. Level Sinyal & *Battery* diinformasikan dengan *component ProgressBar* berwarna hijau disebelah kiri Form Halaman Utama.

Pengujian mengoneksikan Form Halaman Utama dengan HP SMS *Gateway* telah sukses dan berjalan dengan baik.

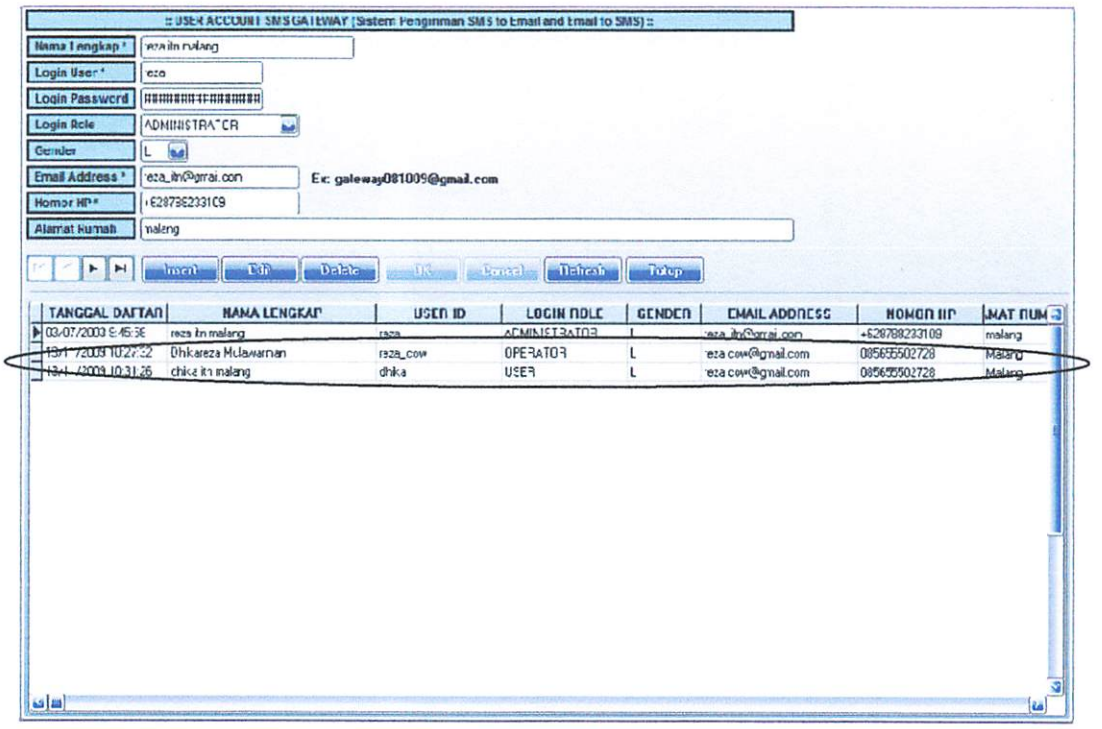
4.2.2.1. Pengujian Form Menu *User Account*

Pengujian Form Menu *User Account* dilakukan dengan melakukan *insert, edit, delete* data *user* / pengoperasi aplikasi. Gambar 4.7 di bawah ini menampilkan Form Menu *User Account* sebelum proses *insert* data *user* baru.



Gambar 4.7 Form Menu *User Account* Sebelum Proses *Insert* Data *User* Baru

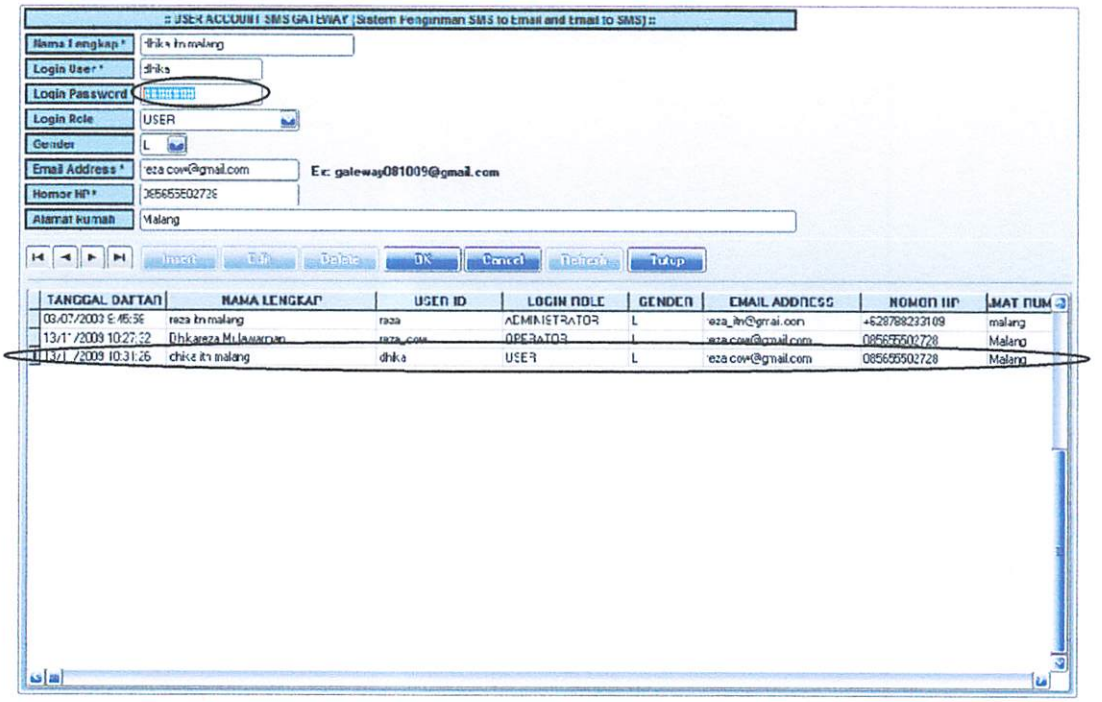
Gambar 4.8 di bawah ini menampilkan Form Menu *User Account* sesudah proses *insert* data *user* baru.



Gambar 4.8 Form Menu *User Account* Sesudah Proses *Insert* Data *User* Baru

Untuk melakukan *insert* data *user* / pengoperasi aplikasi, klik *button* Insert kemudian isi data-data *user* pada *edit text* yang telah tersedia. Setelah selesai mengisi semua *edit text*, klik *button* OK untuk menyimpan kedalam *database*. Sedangkan untuk batal, klik *button* Cancel. Pada gambar di atas data *user* yang baru di *insert* adalah data *user* dengan nama lengkap "dhika itn malang" dan "Dhikareza Mulawarman".

Gambar 4.9 di bawah ini menggambarkan proses *edit* data *user* / pengoperasi aplikasi.

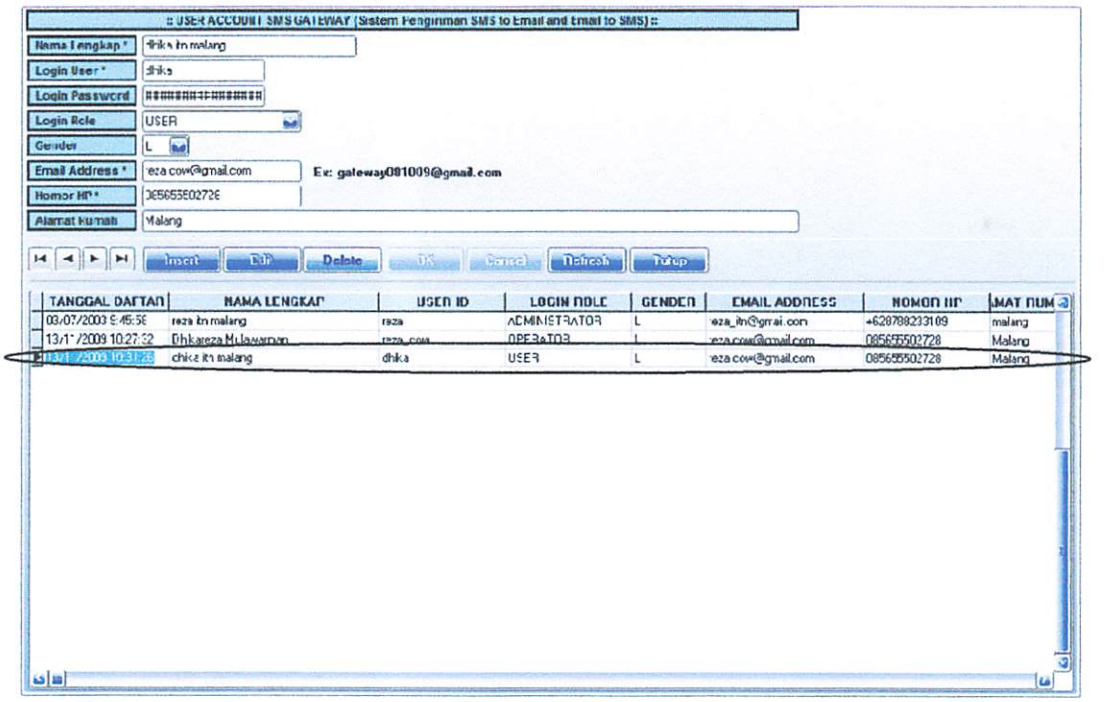


Gambar 4.9 Proses *Edit* Data *User* / Pengoperasi Aplikasi

Untuk melakukan *edit* data *user* / pengoperasi aplikasi, pertama-tama pilih data *user* yang ingin di *edit*, kemudian klik *button* Edit dan rubah pada data yang ingin di *edit*. Setelah selesai, klik button OK untuk menyimpan hasil rubahan kedalam *database*. Sedangkan untuk batal, klik *button*

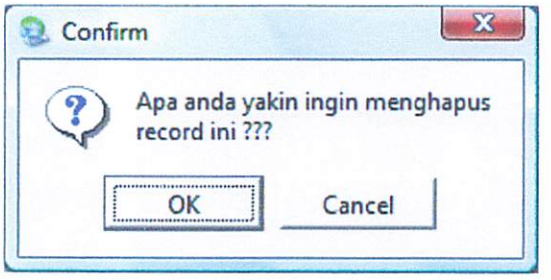
Cancel. Pada gambar di atas data *user* yang dipilih untuk di *edit* adalah data *user* dengan nama lengkap "dhika itn malang" dan data yang di *edit* adalah data *Login Password.*

Gambar 4.10 di bawah ini menampilkan Form Menu *User Account* sebelum proses *delete* data *user*.



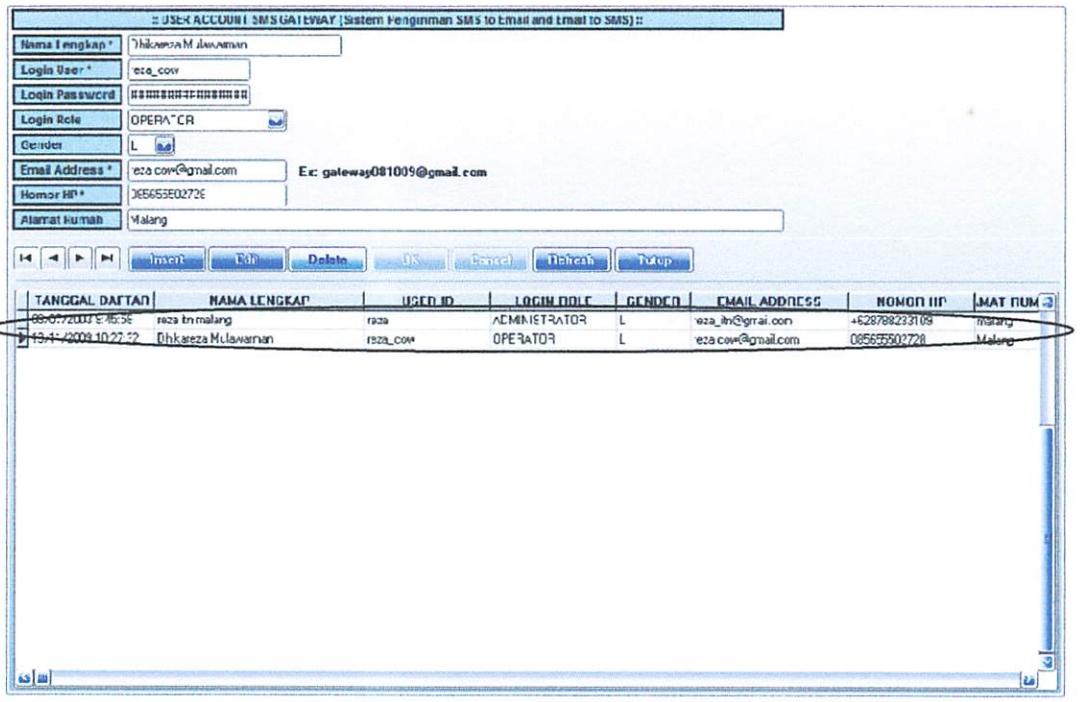
Gambar 4.10 Form Menu *User Account* Sebelum Proses *Delete* Data *User*

Untuk melakukan *delete* data *user* / pengoperasi aplikasi, pertama-tama pilih data *user* yang ingin di *delete*, kemudian klik *button* Delete. Sebelum data benar-benar dihapus, akan muncul pesan:



Apabila sudah yakin data benar-benar akan dihapus, klik *button* OK , sedangkan apabila ingin membatalkan proses *delete* data *user*, klik *button* Cancel . Pada gambar di atas data *user* yang dipilih untuk di *delete* adalah data *user* dengan nama lengkap "dhika itn malang".

Gambar 4.11 di bawah ini menampilkan Form Menu *User Account* sesudah proses *delete* data *user*.



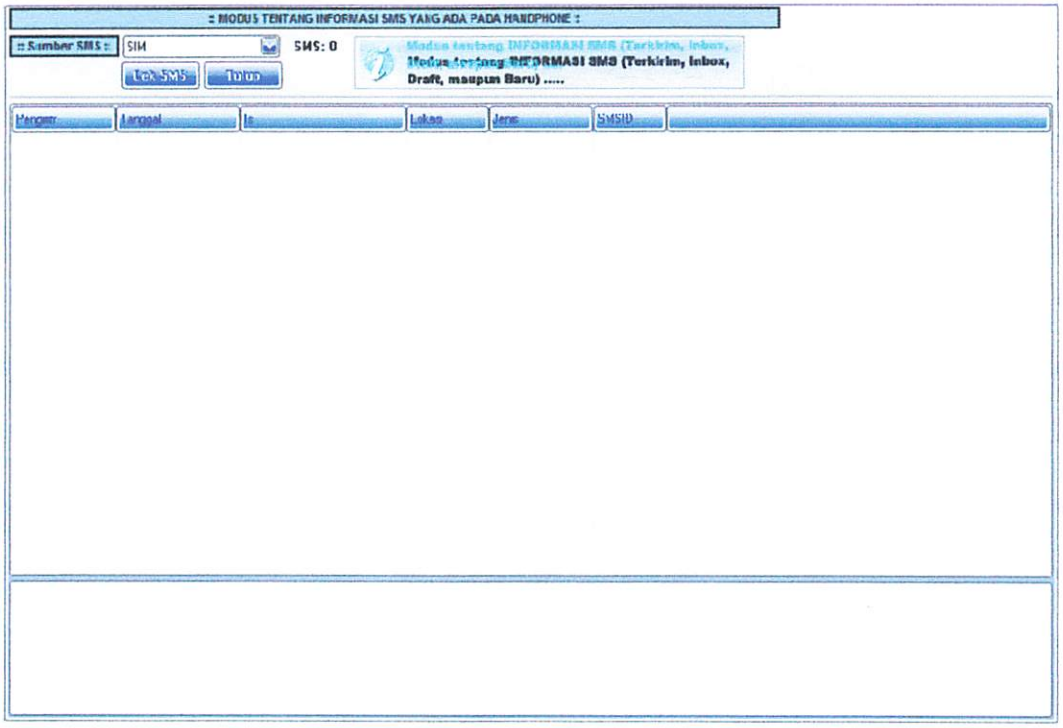
Gambar 4.11 Form Menu *User Account* Sesudah Proses *Delete* Data *User*

Pengujian *Insert, Edit, Delete* Data *User* / pengoperasi aplikasi pada Form Menu *User Account* telah sukses dan berjalan dengan baik.

4.2.2.2. Pengujian Form Menu Informasi SMS

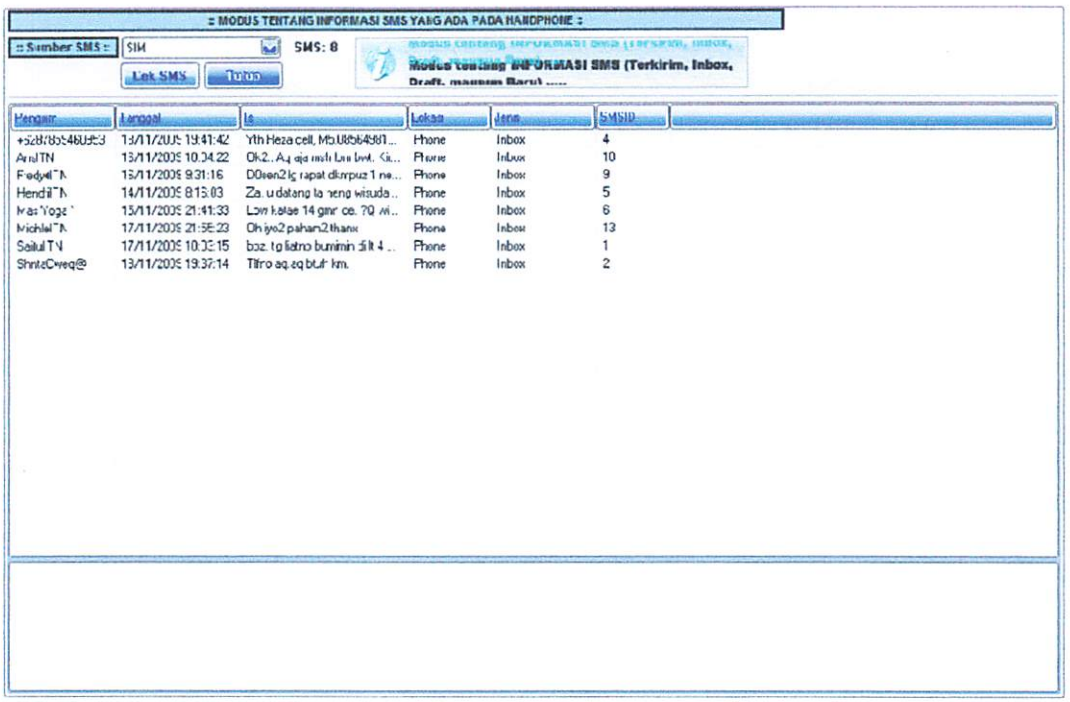
Untuk pengujian Form Menu Informasi SMS dilakukan dengan menentukan terlebih dahulu darimana sumber SMS yang didapat sebelum menampilkan data SMS *Inbox* HP SMS *Gateway*. Sumber SMS bisa didapat dari *memory* kartu SIM / *memory* HP / ke dua-duanya. Gambar 4.12 di bawah ini

memperlihatkan Form Menu Informasi SMS sebelum menampilkan data SMS *inbox* HP SMS *Gateway*.



Gambar 4.12 Form Menu Informasi SMS Sebelum Menampilkan Data SMS *Inbox* HP SMS *Gateway*

Gambar 4.13 di bawah ini memperlihatkan Form Menu Informasi SMS sesudah menampilkan data SMS *inbox* HP SMS *Gateway*.



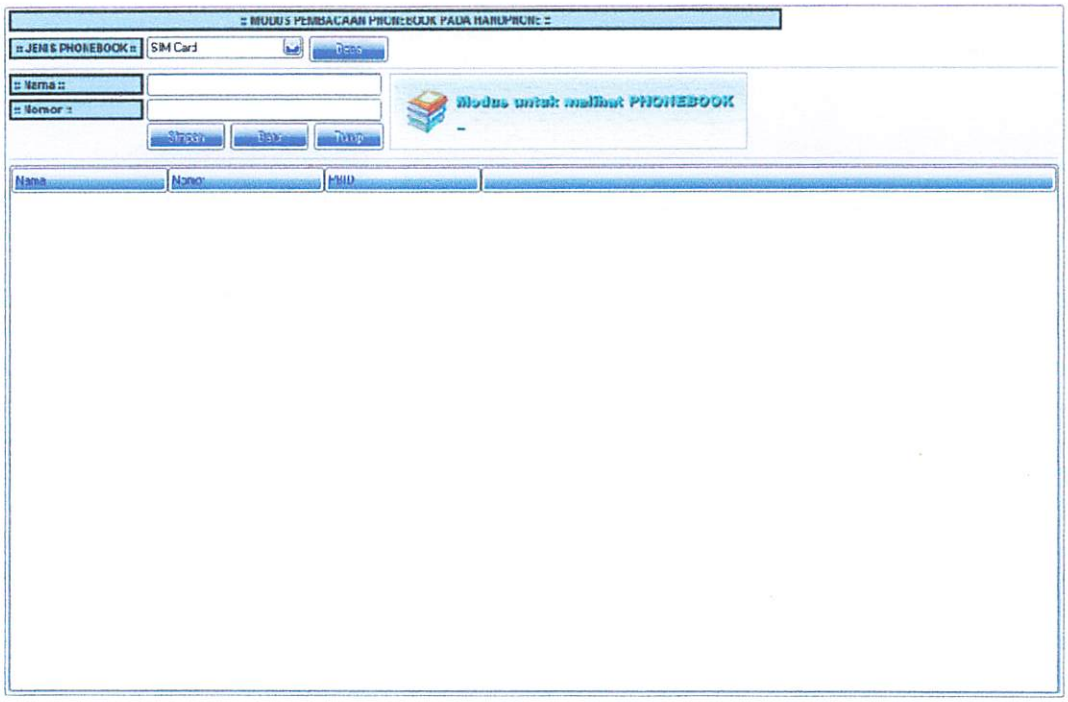
Gambar 4.13 Form Menu Informasi SMS Sesudah Menampilkan Data SMS *Inbox* HP SMS *Gateway*

Dalam pengujian kali ini, sumber data SMS *Inbox* HP SMS *Gateway* didapat dari *memory* kartu SIM dan sebanyak 8 SMS berhasil ditampilkan.

Pengujian menampilkan data SMS *Inbox* HP SMS *Gateway* pada Form Menu Informasi SMS telah sukses dan berjalan dengan baik.

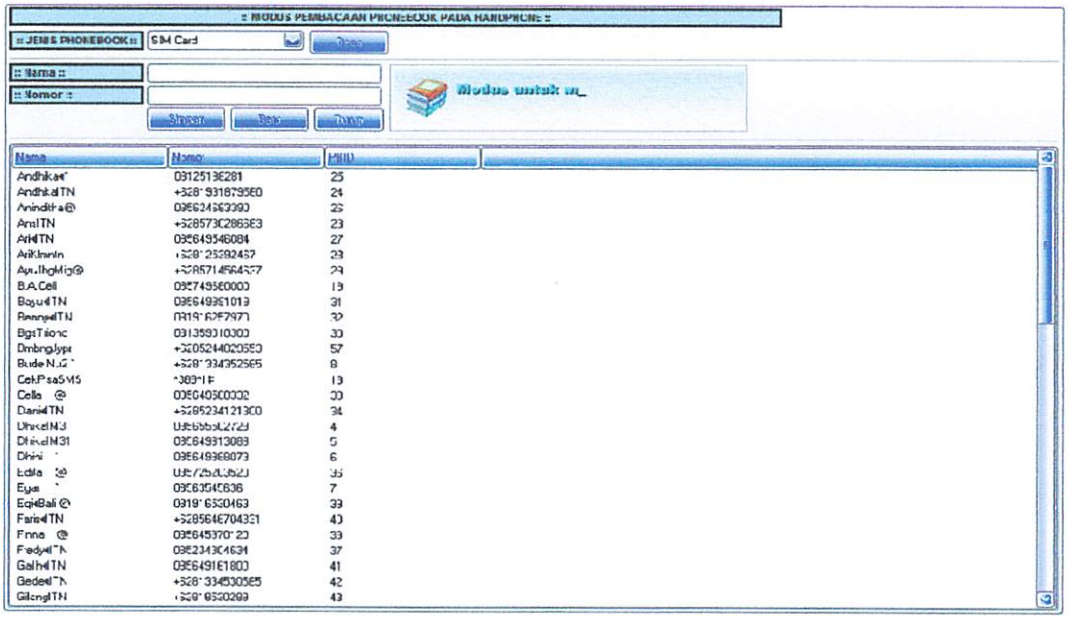
4.2.2.3. Pengujian Form Menu Pembacaan Buku Telepon (*Phonebook*)

Pengujian disini tujuannya untuk membaca daftar buku telepon dari HP SMS *Gateway* dengan menentukan jenis *phonebook* yang akan dibaca. Jenis *phonebook* bisa dari *memory* SIM Card / *memory* telepon. Gambar 4.14 di bawah ini menampilkan Form Menu Pembacaan *Phonebook* sebelum membaca daftar buku telepon dari HP SMS *Gateway*.



Gambar 4.14 Form Menu Pembacaan *Phonebook* Sebelum Membaca Daftar Buku Telepon dari HP SMS *Gateway*.

Gambar 4.15 di bawah ini menampilkan Form Menu Pembacaan *Phonebook* sesudah membaca daftar buku telepon dari HP SMS *Gateway*.



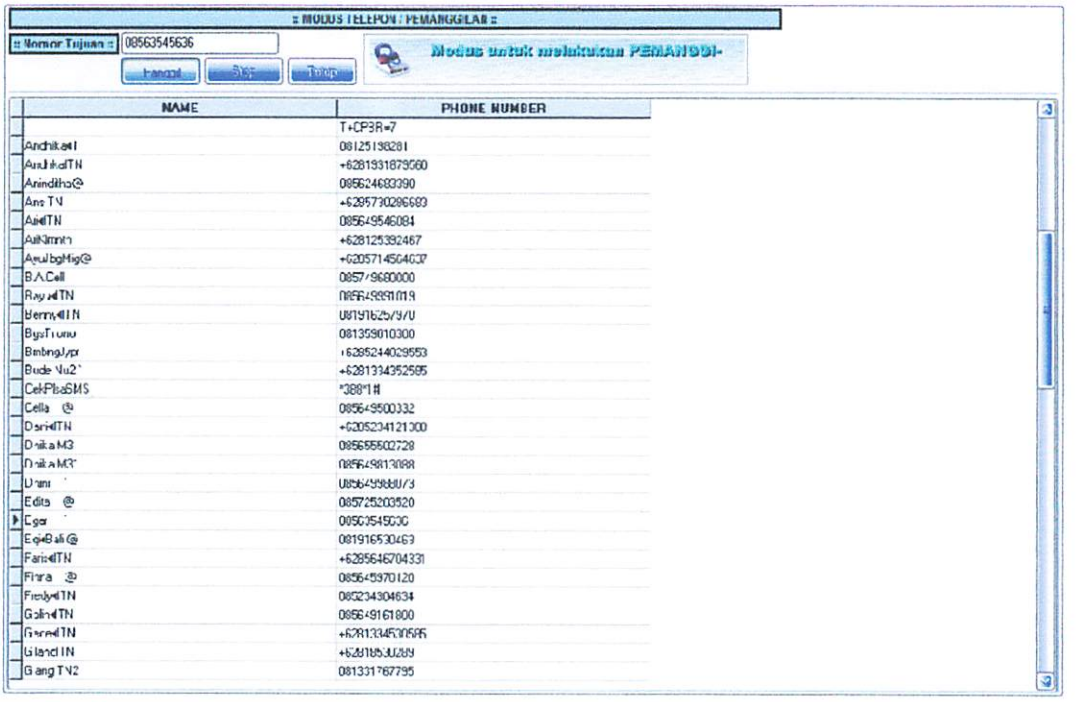
Gambar 4.15 Form Menu Pembacaan *Phonebook* Sesudah Membaca Daftar Buku Telepon dari HP SMS *Gateway*.

Dalam pengujian kali ini, jenis *phonebook* yang dibaca didapat dari *memory* SIM Card dan sebanyak 75 daftar buku telepon berhasil dibaca.

Pengujian membaca daftar buku telepon HP SMS *Gateway* pada Form Menu Pembacaan *Phonebook* telah sukses dan berjalan dengan baik.

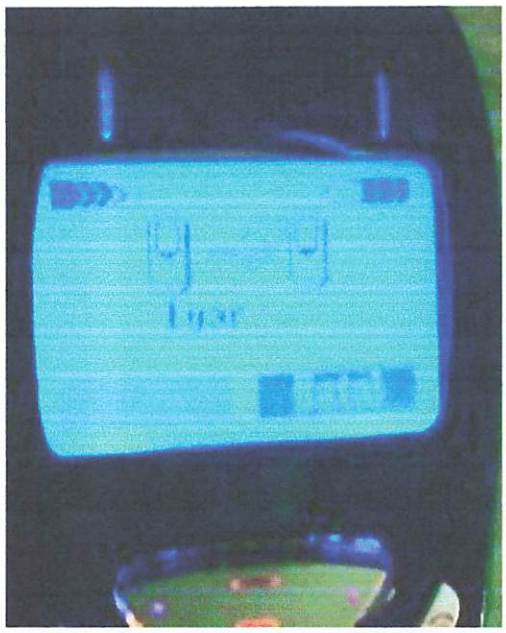
4.2.2.4. Pengujian Form Menu Telepon / Pemanggilan

Untuk Form Menu Telepon / Pemanggilan, pengujian dilakukan dengan melakukan panggilan keluar ke salah 1 nomor HP yang terdapat pada daftar *phonebook*. Gambar 4.16 di bawah ini menggambarkan Form Menu Telepon / Pemanggilan dalam proses telepon ke nomor HP tujuan.



Gambar 4.16 Form Menu Telepon / Pemanggilan dalam Proses Telepon ke Nomor HP Tujuan

Gambar 4.17 di bawah ini menampilkan HP SMS *Gateway* melakukan panggilan ke Nomor HP tujuan.

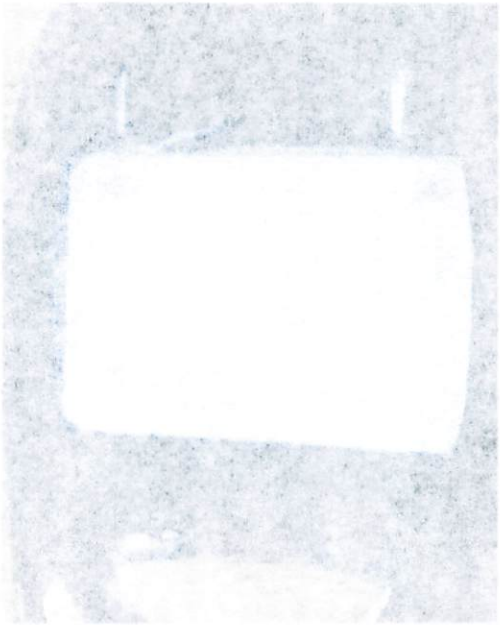


Gambar 4.17 HP SMS *Gateway* Melakukan Panggilan ke Nomor HP Tujuan

Gambar 4.18 di bawah ini menampilkan HP tujuan menerima panggilan dari HP SMS *Gateway*.

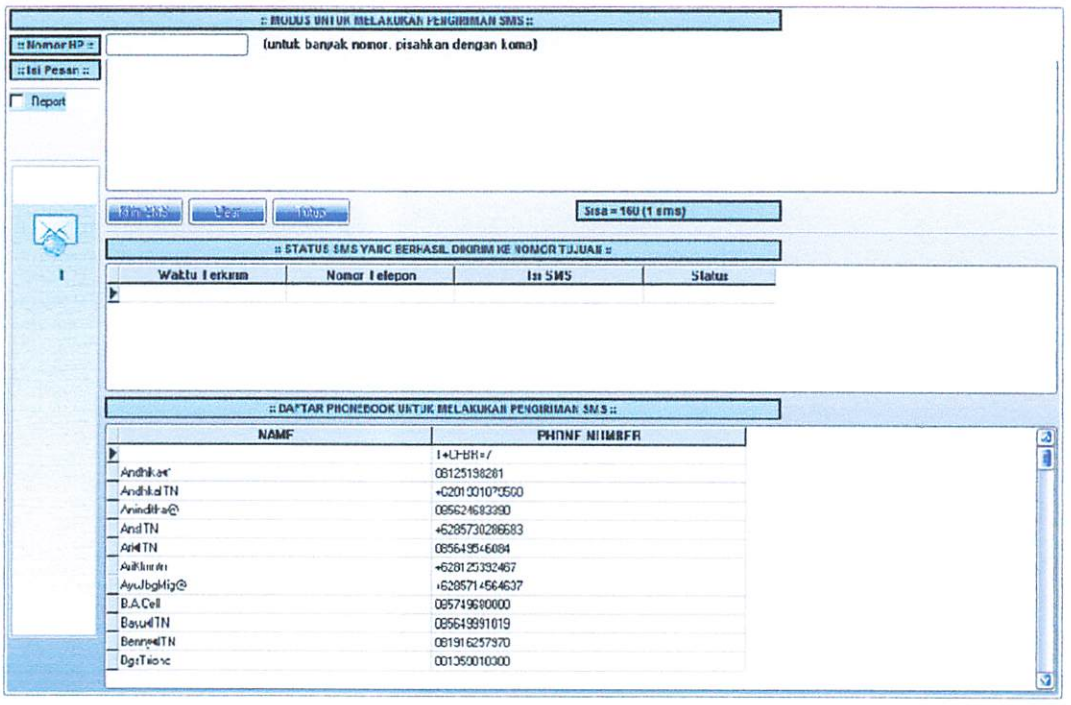
Gambar 4.18 HP Tujuan Menerima Panggilan dari HP SMS *Gateway*

Dalam pengujian yang telah dilakukan, nomor HP tujuan yang dipilih adalah 08563545636 dengan nama Egar. Pengujian panggilan keluar ke salah 1 nomor HP yang terdapat pada daftar *phonebook* melalui Form Menu Telepon / Pemanggilan telah sukses dan berjalan dengan baik.



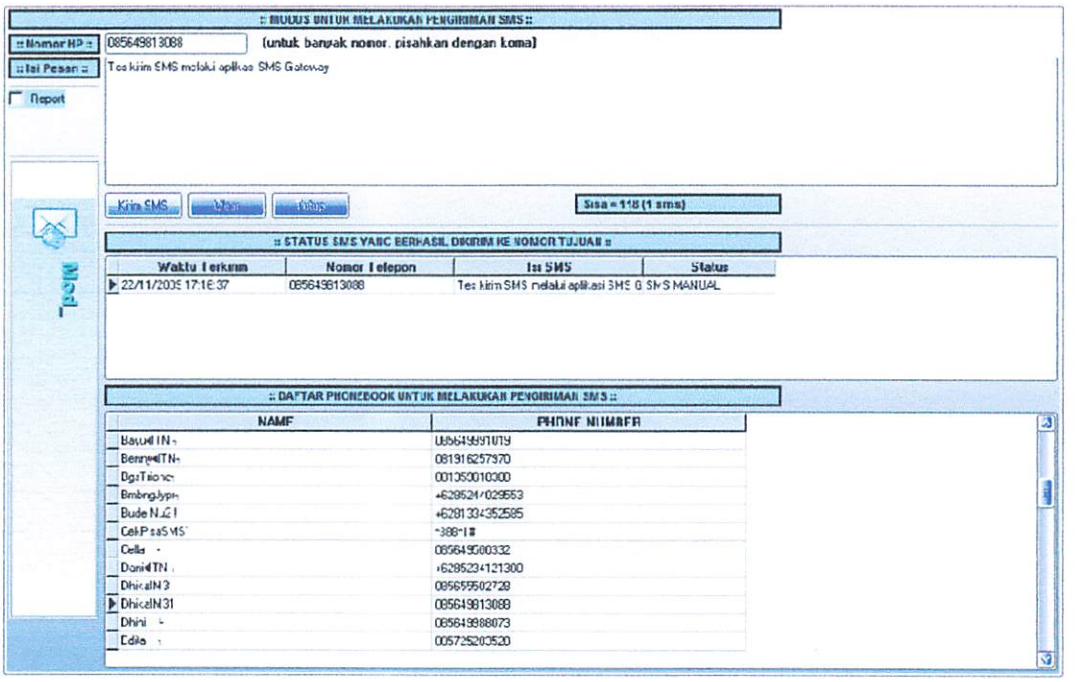
4.2.2.5. Pengujian Form Menu Pengiriman SMS

Pengujian Form Menu Pengiriman SMS dilakukan dengan mengirim SMS secara manual ke salah 1 nomor HP yang terdapat pada daftar buku telepon. Status SMS yang berhasil dikirim dapat dilihat secara langsung pada Form Menu Pengiriman SMS ini. Gambar 4.19 di bawah ini menggambarkan Form Menu Pengiriman SMS sebelum proses pengiriman SMS.



Gambar 4.19 Form Menu Pengiriman SMS Sebelum Proses Pengiriman SMS

Gambar 4.20 di bawah ini menggambarkan Form Menu Pengiriman SMS sesudah proses pengiriman SMS.



Gambar 4.20 Form Menu Pengiriman SMS Sesudah Proses Pengiriman SMS

Gambar 4.21 di bawah ini menampilkan HP tujuan menerima SMS dari aplikasi SMS *Gateway*.

Gambar 4.21 HP Tujuan Menerima SMS dari Aplikasi SMS *Gateway*

Agar dapat melakukan pengiriman SMS melalui aplikasi ini, pertama-tama pengoperasi aplikasi harus memilih salah 1 nomor HP yang terdapat pada daftar *phonebook*, kemudian menuliskan isi pesan pada *TextBox* yang telah tersedia. Apabila isi pesan tersebut dirasa sudah cukup, maka pengoperasi aplikasi

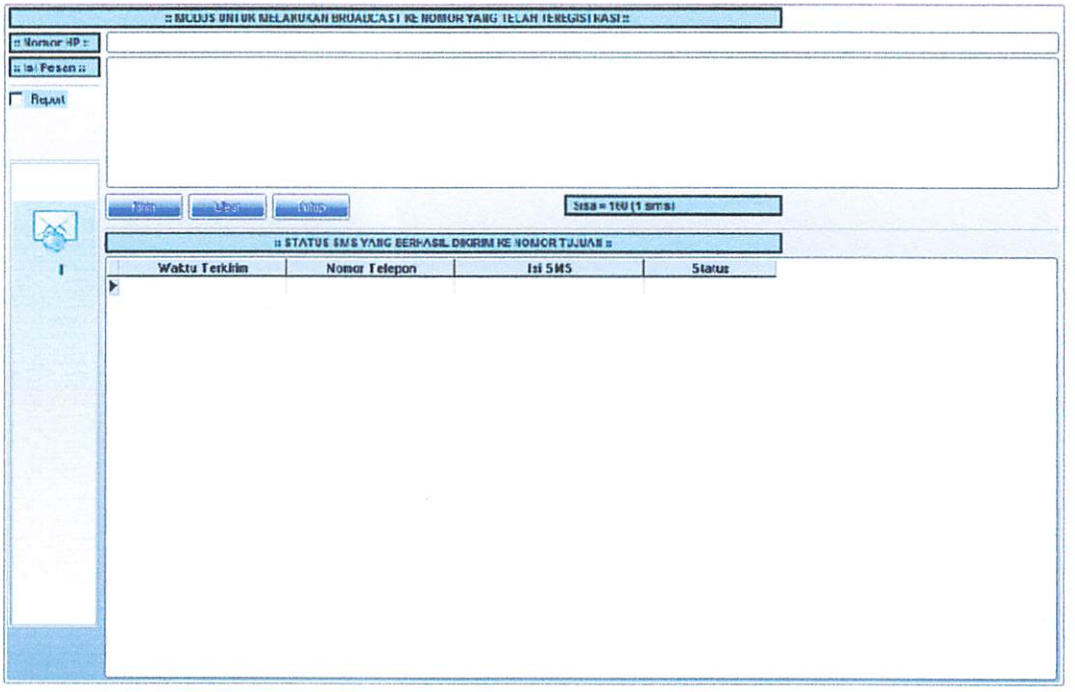
bisa mengirimkan SMS dengan meng-klik *button* Kirim SMS. Untuk menghapus / mengganti isi pesan yang akan dikirim, pengoperasi aplikasi tinggal meng-klik *button* Clear. Dan untuk menutup Form Menu Pengiriman SMS, pengoperasi aplikasi tinggal meng-klik *button* Tutup. Status SMS yang berhasil dikirim ke nomor tujuan dapat dilihat pada bagian bawah *TextBox* untuk menulis isi pesan.

Dalam pengujian kali ini, pengoperasi aplikasi mengirim SMS ke nomor HP "**085649813088**", isi pesannya "**Tes kirim SMS melalui aplikasi SMS Gateway**", dan statusnya berhasil dikirim pada tanggal **22/11/2009 17:16:37**.

Pengujian mengirim SMS secara manual ke salah 1 nomor HP melalui Form Menu Pengiriman SMS telah sukses dan berjalan dengan baik.

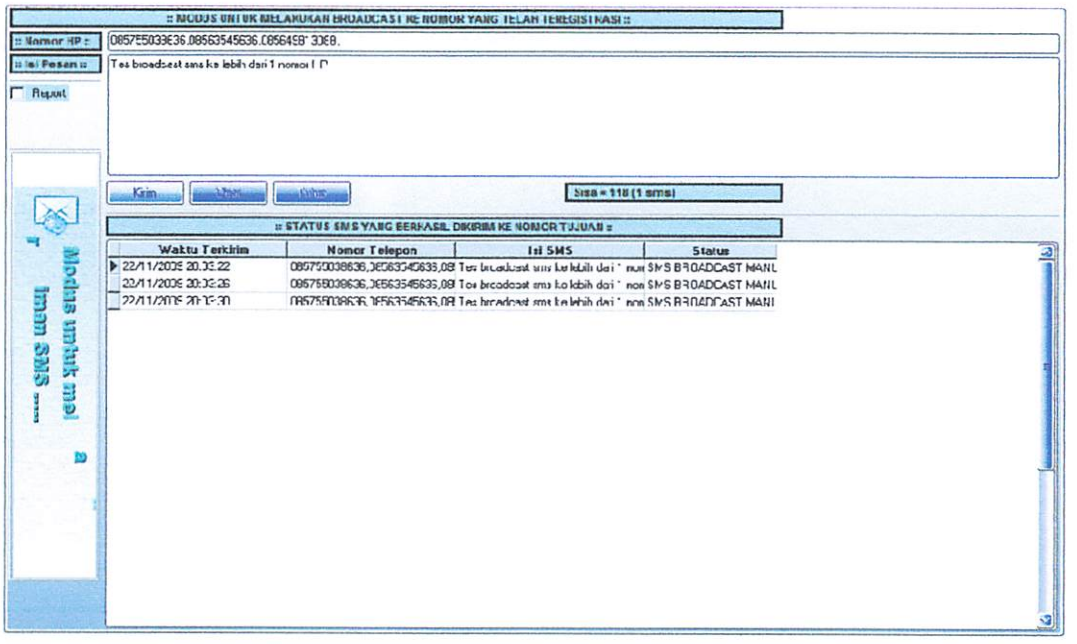
4.2.2.6. Pengujian Form Menu *Broadcast* Pengiriman SMS

Untuk pengujian Form Menu *Broadcast* Pengiriman SMS dilakukan dengan mengirim SMS secara manual ke lebih dari 1 nomor HP yang telah teregistrasi. Gambar 4.22 di bawah ini menggambarkan Form Menu *Broadcast* Pengiriman SMS sebelum proses pengiriman SMS ke lebih dari 1 nomor HP yang telah teregistrasi.



Gambar 4.22 Form Menu *Broadcast* Pengiriman SMS Sebelum Proses Pengiriman SMS ke Lebih dari 1 Nomor HP yang Telah Teregistrasi

Gambar 4.23 di bawah ini menggambarkan Form Menu *Broadcast* Pengiriman SMS sesudah proses pengiriman SMS ke lebih dari 1 nomor HP yang telah teregistrasi.



Gambar 4.23 Form Menu *Broadcast* Pengiriman SMS Sesudah Proses Pengiriman SMS ke Lebih dari 1 Nomor HP yang Telah Teregistrasi

Gambar 4.24 di bawah ini menampilkan HP tujuan 1 menerima *broadcast* SMS dari aplikasi SMS *Gateway*.

Gambar 4.24 HP Tujuan 1 Menerima *Broadcast* SMS dari Aplikasi SMS *Gateway*

Gambar 4.25 di bawah ini menampilkan HP tujuan ke 2 menerima *broadcast* SMS dari aplikasi SMS *Gateway*.

Gambar 4.25 HP Tujuan ke 2 Menerima *Broadcast* SMS dari Aplikasi SMS *Gateway*

Gambar 4.26 di bawah ini menampilkan HP tujuan ke 3 menerima *broadcast* SMS dari aplikasi SMS *Gateway*.

Gambar 4.26 HP Tujuan ke 3 Menerima *Broadcast* SMS dari Aplikasi SMS *Gateway*

Form Menu Pengiriman SMS dan Form Menu *Broadcast* Pengiriman SMS fungsinya sama yaitu mengirim SMS secara manual ke nomor HP tujuan, tetapi nomor HP tujuan dari Form Menu Pengiriman SMS didapat dari daftar *phonebook* yang ada di HP SMS *Gateway* dan hanya bisa mengirim SMS ke 1 nomor HP saja, sedangkan nomor HP tujuan dari Form Menu *Broadcast* Pengiriman SMS didapat dari nomor yang telah melakukan registrasi dan bisa mengirim SMS ke lebih dari 1 nomor HP.

Dalam pengujian kali ini, aplikasi mengirim SMS ke 3 nomor HP Tujuan yang telah teregistrasi melalui SMS dengan isi pesan "**Tes broadcast sms ke lebih dari 1 nomor HP**". Tabel 4.1 di bawah ini menjelaskan hasil pengujian Form Menu *Broadcast* Pengiriman SMS.

Tabel 4.1 Hasil Pengujian Form Menu Broadcast Pengiriman SMS

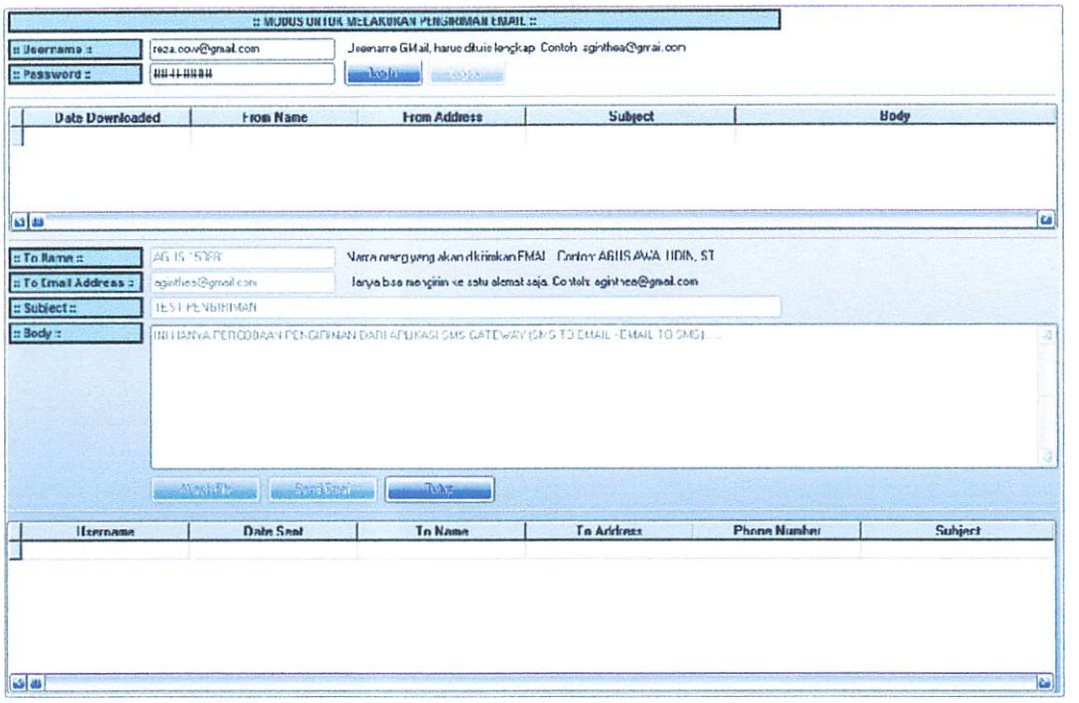
HP Tujuan	Nomor HP	Isi SMS	Waktu Terkirim	Perbedaan Waktu
1	085649813088	Tes broadcast sms ke	22/11/2009 20:03:22	4 detik

2	085755038636	lebih dari 1 nomor HP	22/11/2009 20:03:26	
3	08563545636		22/11/2009 20:03:30	

Pengujian mengirim SMS secara manual ke lebih dari 1 nomor HP yang telah teregistrasi melalui Form Menu *Broadcast* Pengiriman SMS telah sukses dan berjalan dengan baik.

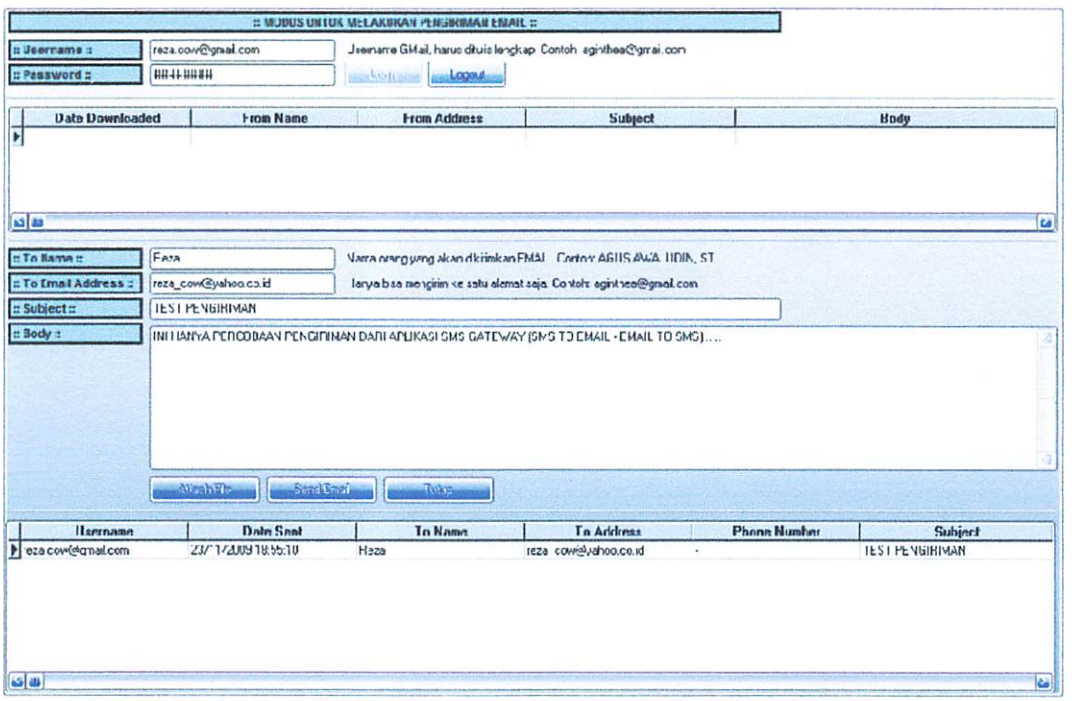
4.2.2.7. Pengujian Form Menu Pengiriman *E-mail*

Pengujian disini tujuannya untuk mengirim *E-mail* secara manual ke satu nama orang & satu alamat *E-mail* saja. Untuk dapat mengirim *E-mail* dari aplikasi ini, aplikasi harus terlebih dahulu terhubung ke internet & pengoperasi aplikasi harus mempunyai *account E-mail* di *webmail* GMAIL. Karena sebelum dapat mengirim *E-mail* dari aplikasi ini, pengoperasi aplikasi harus *Login* dengan menggunakan *Username & Password* GMAIL. Gambar 4.27 di bawah ini menggambarkan Form Menu Pengiriman *E-mail* sebelum proses pengiriman *E-mail*.



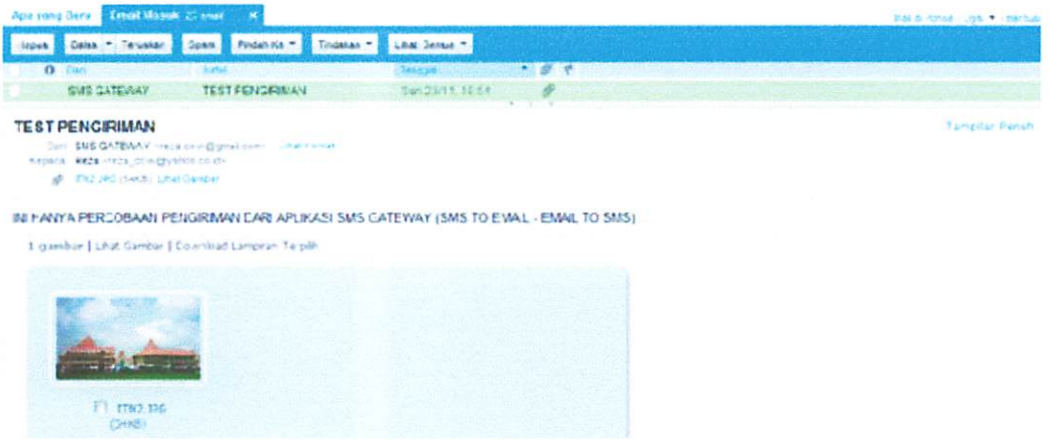
Gambar 4.27 Form Menu Pengiriman *E-mail* Sebelum Proses Pengiriman *E-mail*

Gambar 4.28 di bawah ini menggambarkan Form Menu Pengiriman *E-mail* sesudah proses pengiriman *E-mail*.



Gambar 4.28 Form Menu Pengiriman *E-mail* Sesudah Proses Pengiriman *E-mail*

Gambar 4.29 di bawah ini menampilkan hasil pengujian Form Menu Pengiriman E-mail pada *account E-mail* tujuan.



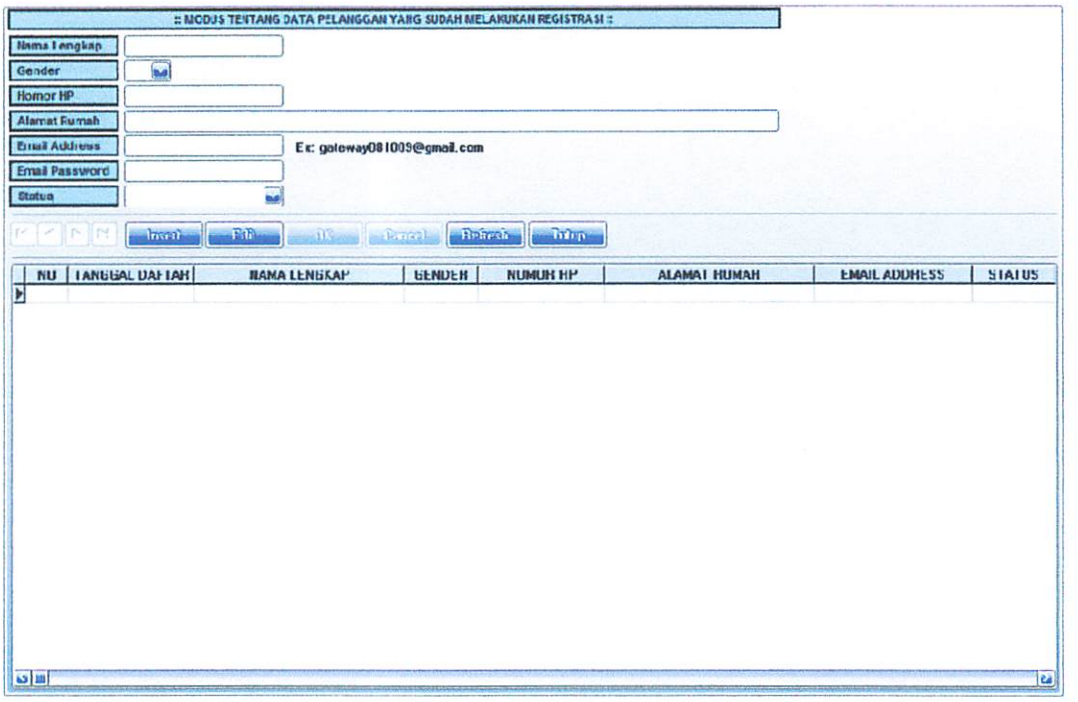
Gambar 4.29 Hasil Pengujian Form Menu Pengiriman E-mail pada *Account E-mail* Tujuan.

Dalam pengujian kali ini, pengoperasi aplikasi melakukan *Login* dengan menggunakan *Username & Password* GMAIL yang dimiliki. Kemudian pengoperasi aplikasi mengirim *E-mail* ke alamat *E-mail* "reza_cow@yahoo.co.id" dengan nama "Reza", *subject*nya "TEST PENGIRIMAN", *body E-mail*nya "INI HANYA PERCOBAAN PENGIRIMAN DARI APLIKASI SMS *GATEWAY* (SMS *TO E-MAIL* & *E-MAIL TO* SMS)...", dan *file* yang di *attach* adalah gambar ITN2.JPG. *E-mail* yang berhasil dikirim akan ditampilkan pada DBGrid yang terdapat pada bagian paling bawah dari Form Menu Pengiriman *E-mail*.

Pengujian mengirim *E-mail* secara manual ke 1 nama orang & 1 alamat *E-mail* melalui Form Menu Pengiriman *E-mail* telah sukses dan berjalan dengan baik.

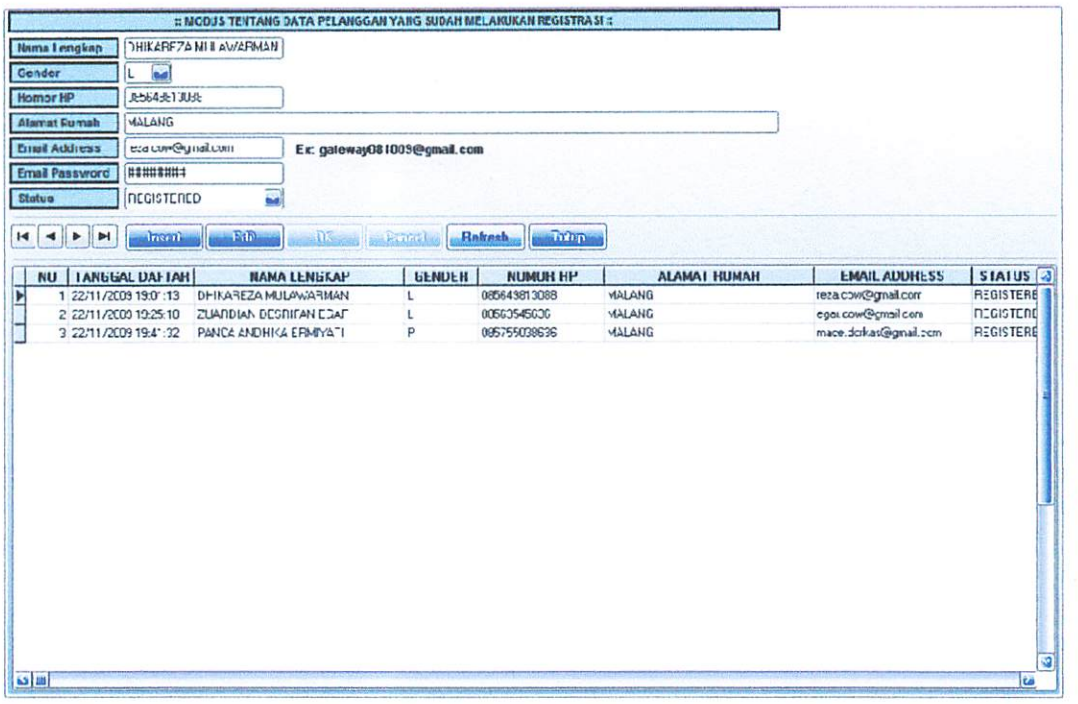
4.2.2.8. Pengujian Form Menu Data Pelanggan

Untuk Form Menu Data Pelanggan, pengujian dilakukan dengan meng-*insert* & meng-*edit* data pelanggan secara manual. Gambar 4.30 di bawah ini menampilkan Form Menu Data Pelanggan sebelum ada pelanggan yang registrasi.



Gambar 4.30 Form Menu Data Pelanggan Sebelum Ada Pelanggan yang Registrasi

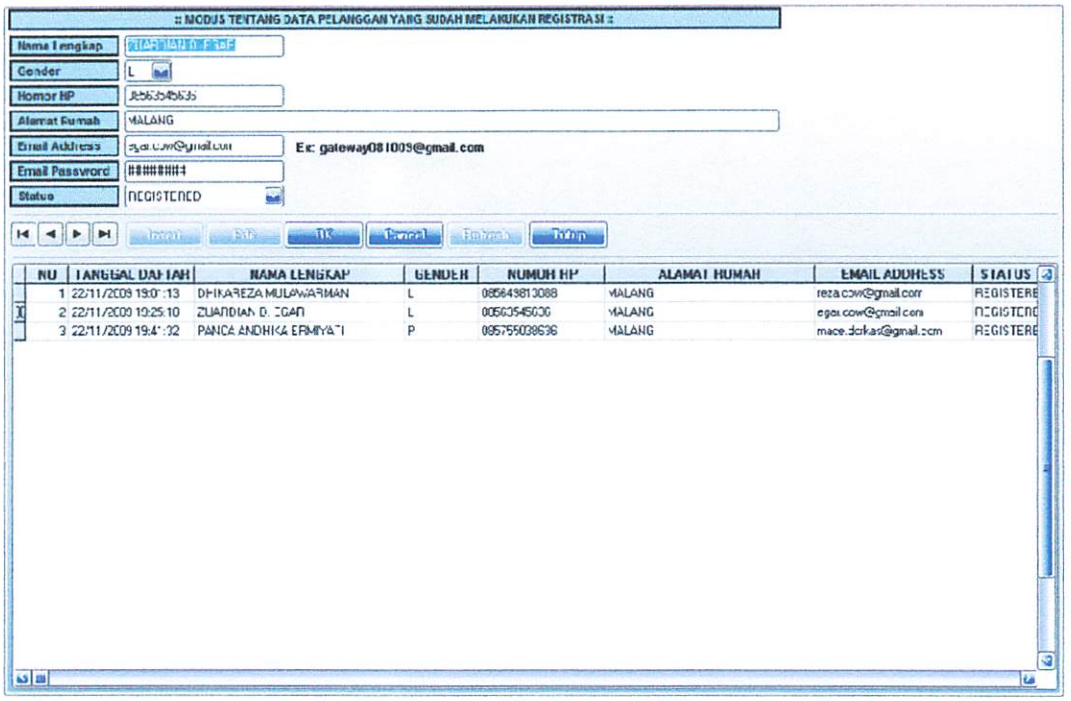
Gambar 4.31 di bawah ini menampilkan Form Menu Data Pelanggan sesudah ada pelanggan yang registrasi.



Gambar 4.31 Form Menu Data Pelanggan Sesudah Ada Pelanggan yang Registrasi

Untuk melakukan *insert* data pelanggan, klik *button* Insert kemudian isi data-data pelanggan pada *TextBox* yang telah tersedia. Setelah selesai mengisi semua *TextBox*, klik *button* OK untuk menyimpan kedalam *database*. Sedangkan untuk batal, klik *button* Cancel. Pada gambar di atas merupakan data pelanggan-pelanggan yang baru melakukan registrasi melalui SMS.

Gambar 4.32 di bawah ini menggambarkan Form Menu Data Pelanggan dalam proses *edit* data pelanggan.



Gambar 4.32 Form Menu Data Pelanggan dalam Proses *Edit* Data Pelanggan

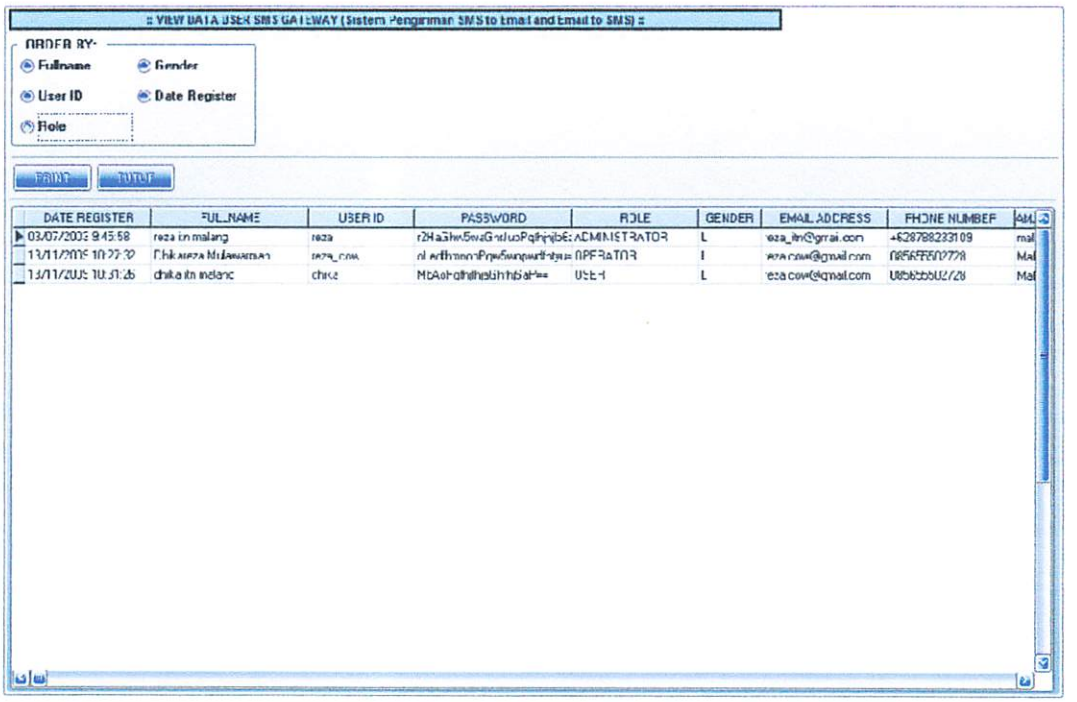
Untuk melakukan *edit* data pelanggan, pertama-tama pilih data pelanggan yang ingin di *edit*, kemudian klik *button* Edit dan rubah pada data yang ingin di *edit*. Setelah selesai, klik button OK untuk menyimpan hasil rubahan kedalam *database*. Sedangkan untuk batal, klik *button* Cancel. Pada gambar di atas data pelanggan yang dipilih untuk di *edit* adalah data pelanggan dengan nama lengkap "ZUARDIAN DESRIFAN EGAR" dan data yang di *edit* adalah data Nama Lengkap berubah menjadi "ZUARDIAN D. EGAR".

Pengujian *Insert & Edit* data pelanggan pada Form Menu Data Pelanggan telah sukses dan berjalan dengan baik.

4.2.2.9. Pengujian Form Menu *View* Data *User*

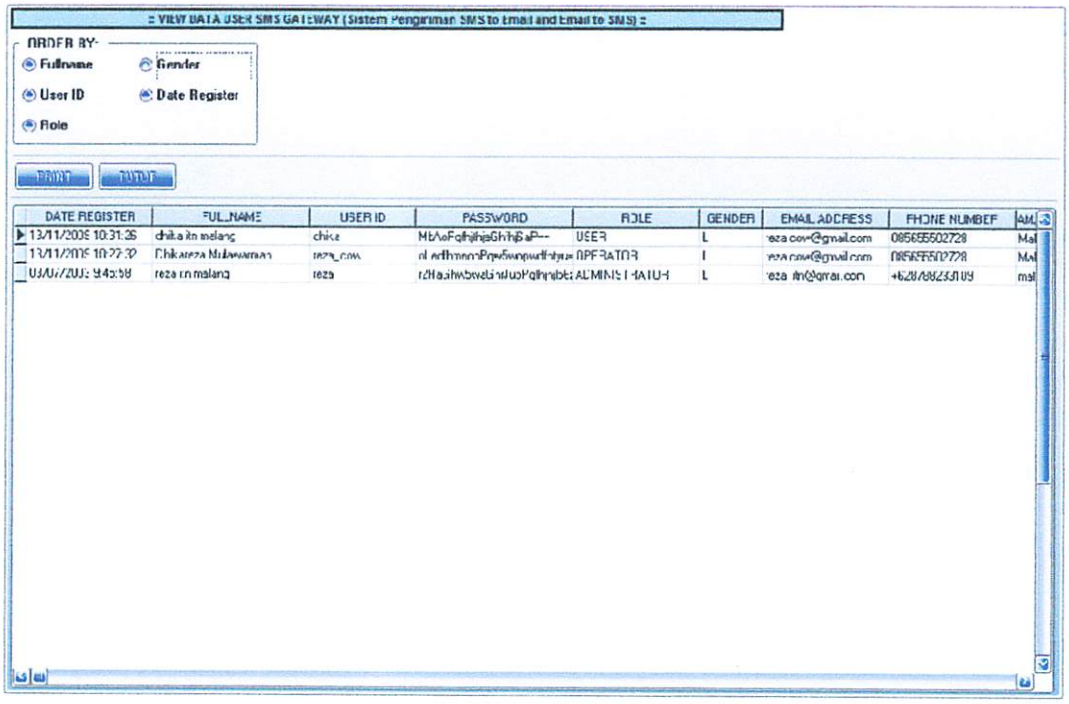
Pengujian Form Menu *View* Data *User* dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada *component RadioGroup* "*Order By:*".

Form Menu *View* Data *User* ini digunakan untuk melihat informasi mengenai data-data *user* / pengoperasi aplikasi sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By:*". Gambar 4.33 di bawah ini merupakan tampilan data-data *user* sesuai dengan *role* (Hak Akses).



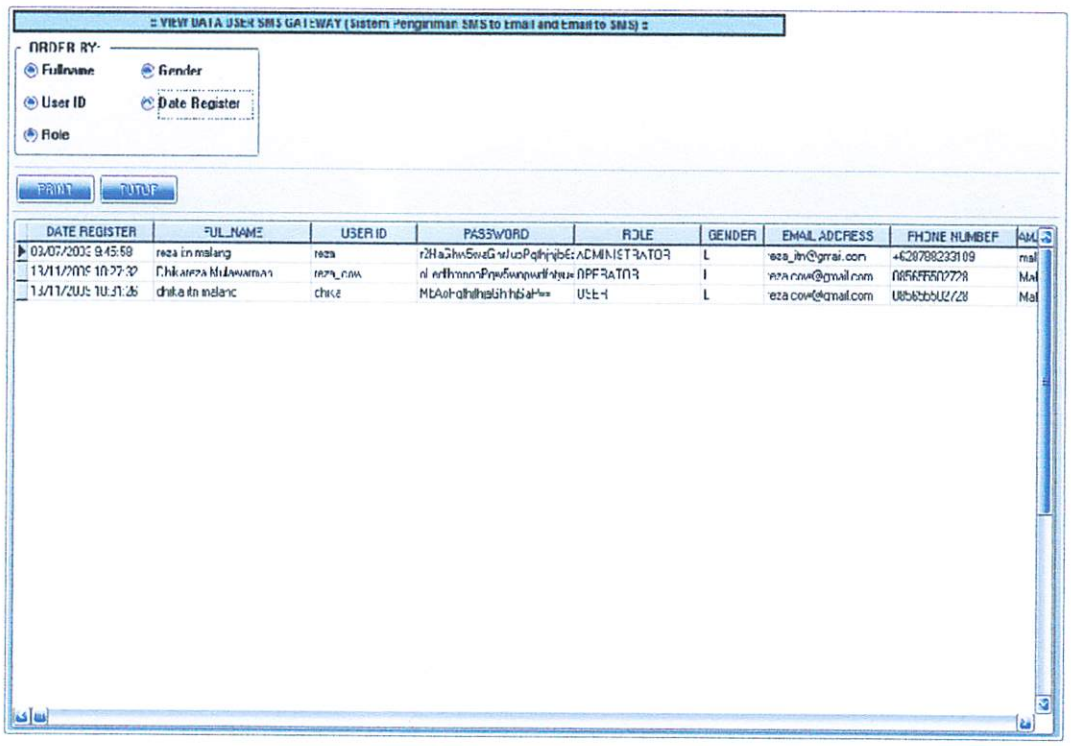
Gambar 4.33 Tampilan Data-Data *User* Sesuai dengan *Role* (Hak Akses)

Gambar 4.34 di bawah ini merupakan tampilan data-data *user* sesuai dengan *gender* (Jenis Kelamin).



Gambar 4.34 Tampilan Data-Data *User* Sesuai dengan *Gender* (Jenis Kelamin)

Gambar 4.35 di bawah ini merupakan tampilan data-data *user* sesuai dengan *date register* (Tanggal Daftar).

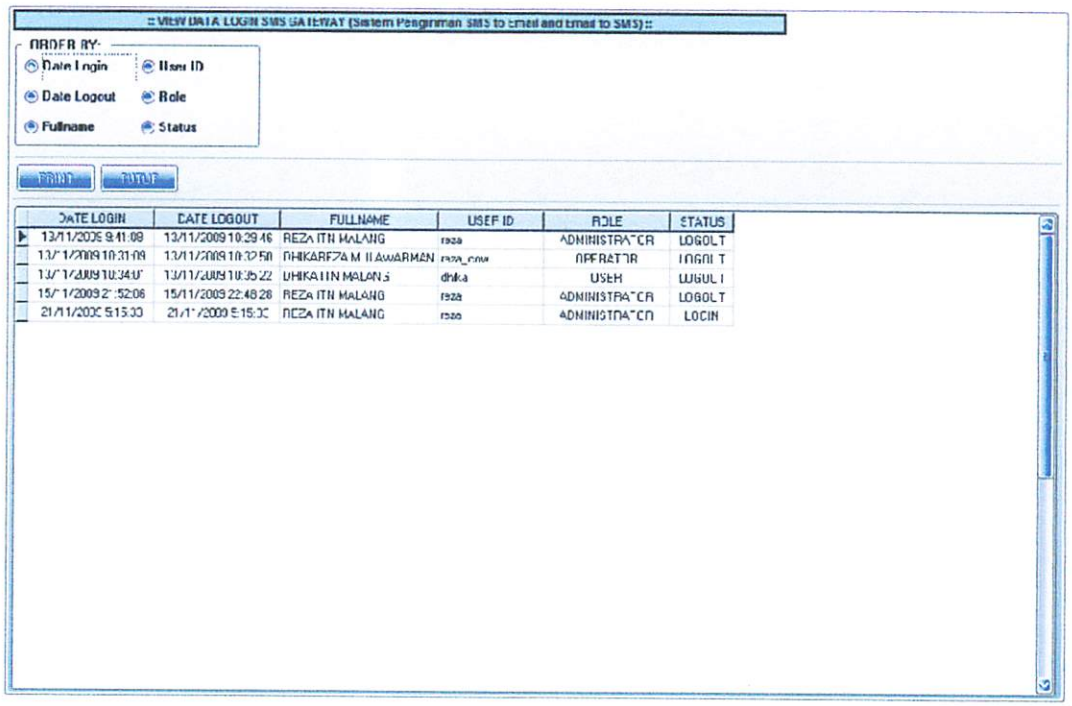


Gambar 4.35 Tampilan Data-Data *User* Sesuai dengan *Date Register* (Tanggal Daftar)

Pengujian melihat informasi mengenai data-data *user* / pengoperasi aplikasi sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" pada Form Menu *View* Data *User* telah sukses dan berjalan dengan baik.

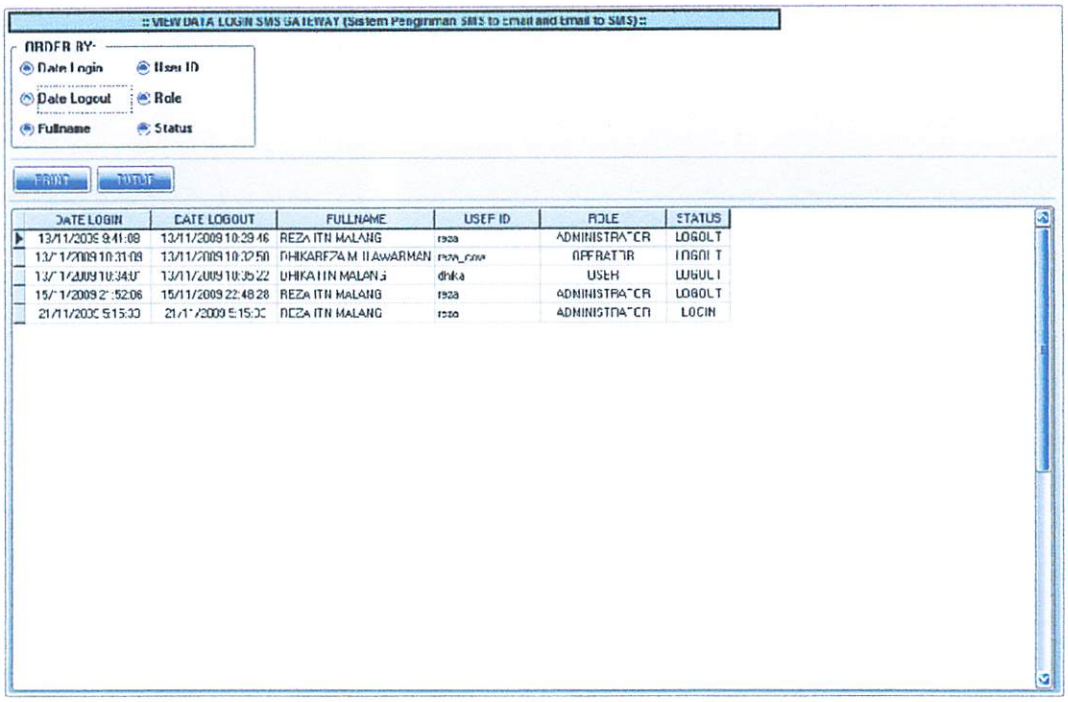
4.2.2.10. Pengujian Form Menu *View* Data *Login*

Untuk pengujian Form Menu *View* Data *Login* dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada *component RadioGroup* "*Order By*:". Form Menu *View* Data *Login* ini digunakan untuk melihat data saat *login* / siapa saja yang pernah menggunakan aplikasi. Gambar 4.36 di bawah ini merupakan tampilan data *login* sesuai dengan *date login* (Tanggal Masuk).



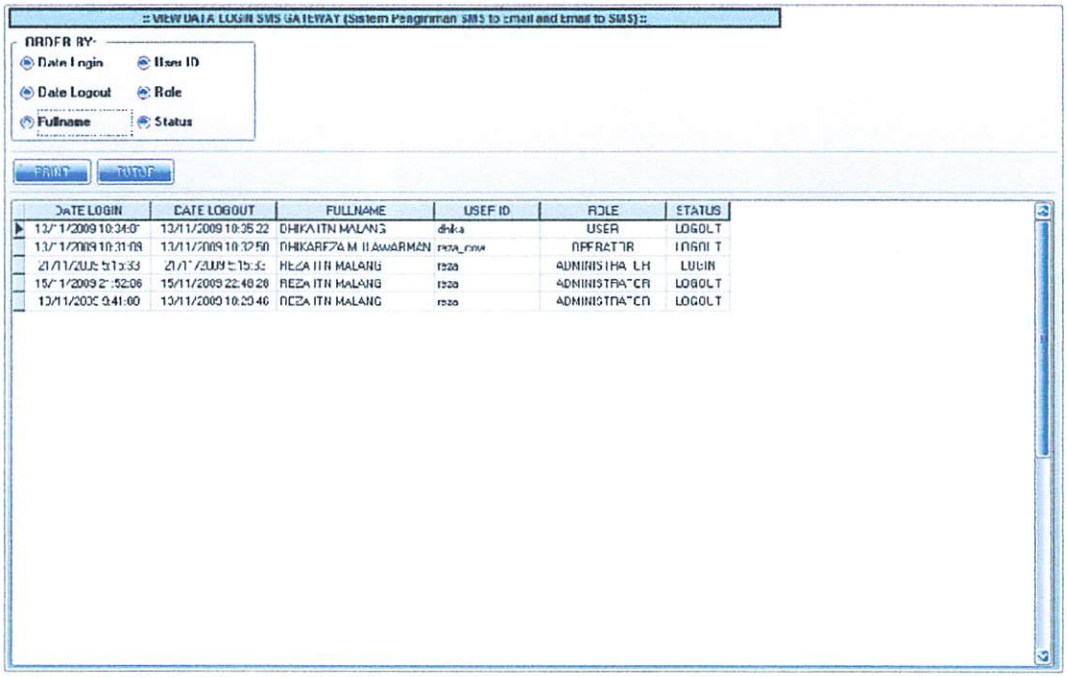
Gambar 4.36 Tampilan Data *Login* Sesuai dengan *Date Login* (Tanggal Masuk)

Gambar 4.37 di bawah ini merupakan tampilan data *login* sesuai dengan *date logout* (Tanggal Keluar).



Gambar 4.37 Tampilan Data *Login* Sesuai dengan *Date Logout* (Tanggal Keluar)

Gambar 4.38 di bawah ini merupakan tampilan data *login* sesuai dengan *fullname* (Nama Lengkap).

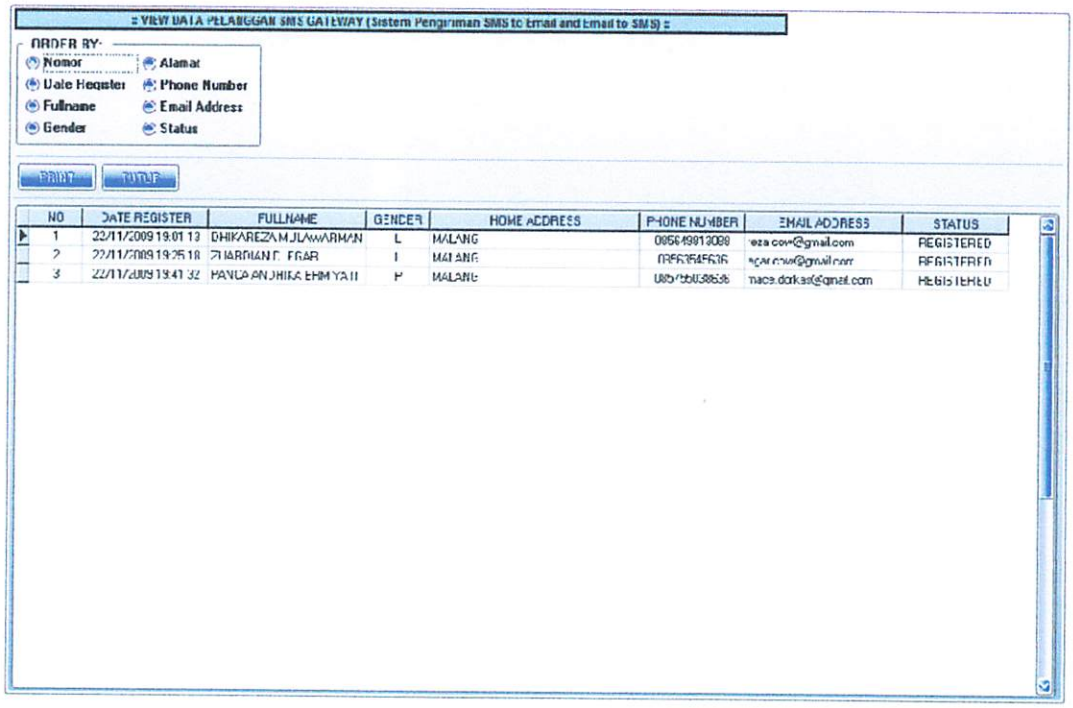


Gambar 4.38 Tampilan Data *Login* Sesuai dengan *Fullname* (Nama Lengkap)

Pengujian melihat data saat *login* / siapa saja yang pernah menggunakan aplikasi sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" pada Form Menu *View* Data *Login* telah sukses dan berjalan dengan baik.

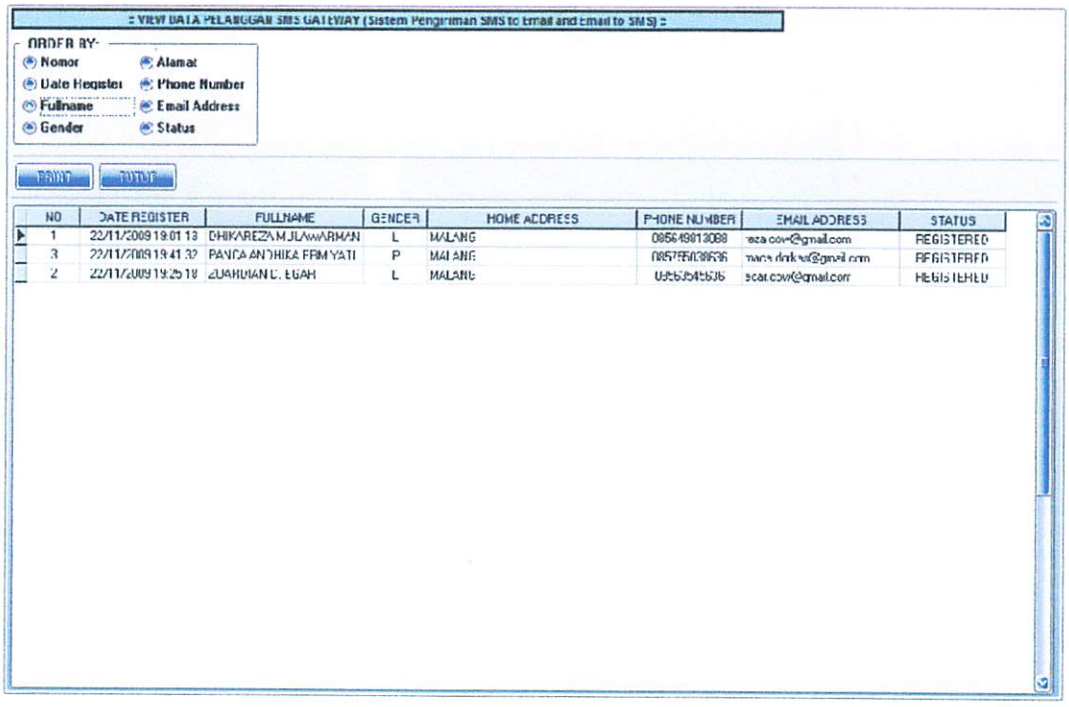
4.2.2.11. Pengujian Form Menu *View* Data Pelanggan

Pengujian disini tujuannya untuk melihat data orang yang sudah melakukan registrasi sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" Gambar 4.39 di bawah ini merupakan tampilan data pelanggan sesuai dengan nomor.



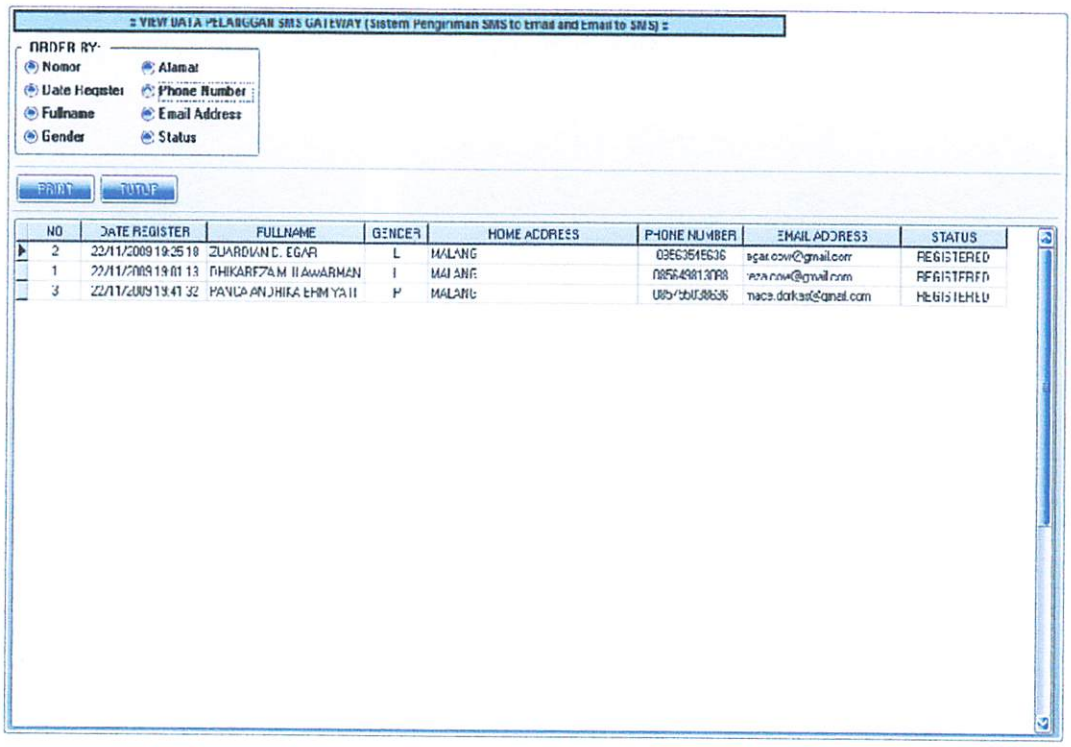
Gambar 4.39 Tampilan Data Pelanggan Sesuai dengan Nomor

Gambar 4.40 di bawah ini merupakan tampilan data pelanggan sesuai dengan *fullname* (Nama Lengkap).



Gambar 4.40 Tampilan Data Pelanggan Sesuai dengan *Fullname* (Nama Lengkap)

Gambar 4.41 di bawah ini merupakan tampilan data pelanggan sesuai dengan *phone number* (Nomor Telepon).

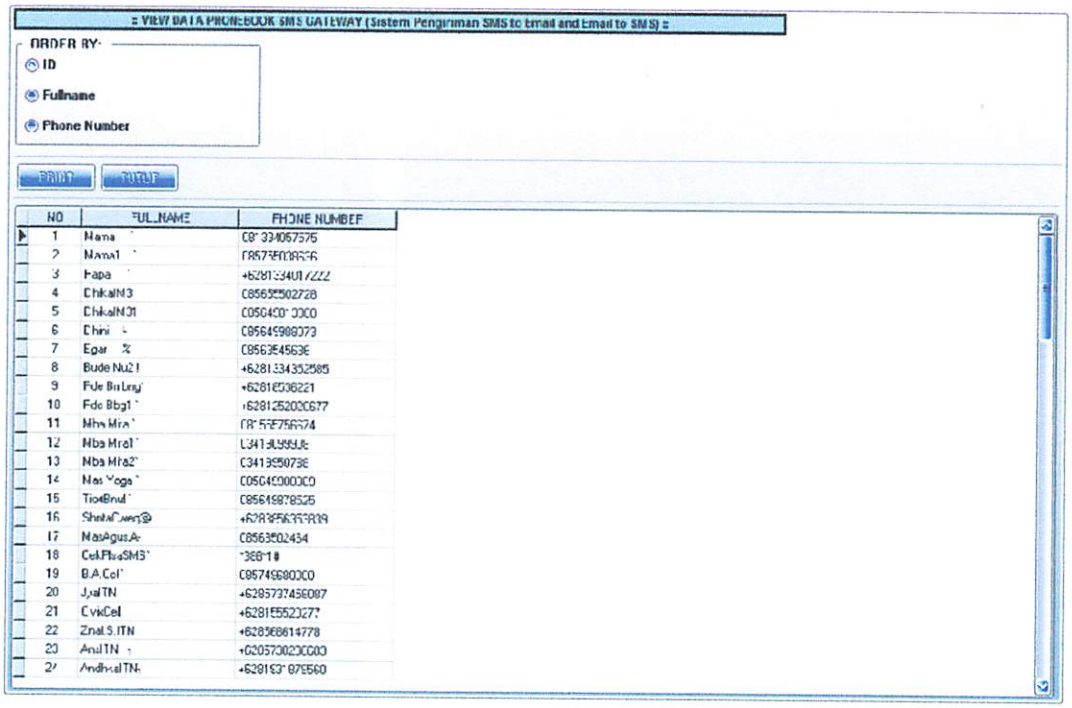


Gambar 4.41 Tampilan Data Pelanggan Sesuai dengan *Phone Number* (Nomor Telepon)

Pengujian melihat data orang yang sudah melakukan registrasi sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" pada Form Menu *View* Data Pelanggan telah sukses dan berjalan dengan baik.

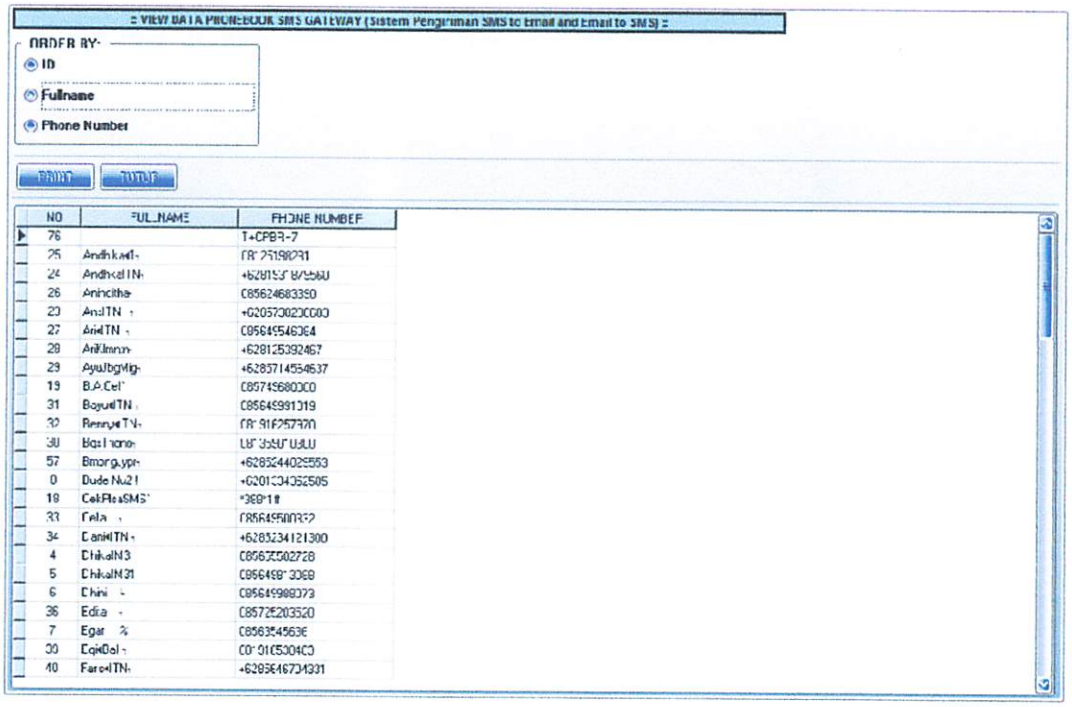
4.2.2.12. Pengujian Form Menu *View* Data *Phonebook*

Untuk Form Menu *View* Data *Phonebook*, pengujian dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada *component RadioGroup* "*Order By*:". Form Menu *View* Data *Phonebook* ini digunakan untuk melihat daftar buku telepon pada HP SMS *Gateway*. Gambar 4.42 di bawah ini merupakan tampilan data *Phonebook* sesuai dengan *ID*.



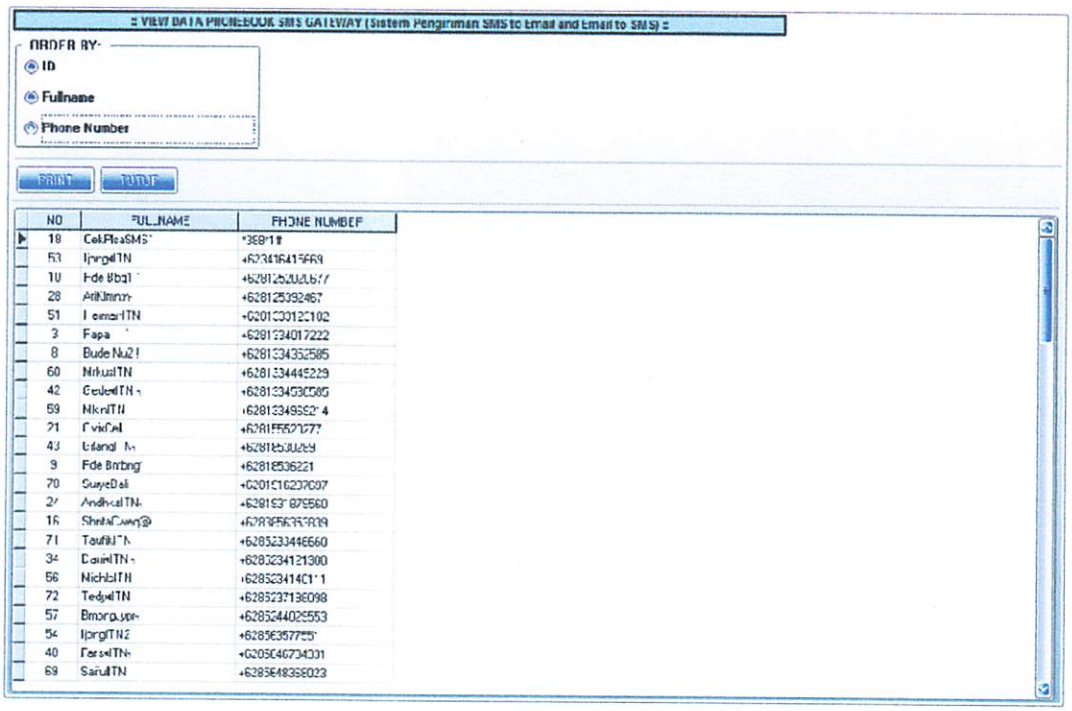
Gambar 4.42 Tampilan Data *Phonebook* Sesuai dengan *ID*

Gambar 4.43 di bawah ini merupakan tampilan data *Phonebook* sesuai dengan *Fullname* (Nama Lengkap).



Gambar 4.43 Tampilan Data *Phonebook* Sesuai dengan *Fullname* (Nama Lengkap)

Gambar 4.44 di bawah ini merupakan tampilan data *Phonebook* sesuai dengan *Phone Number* (Nomor Telepon).

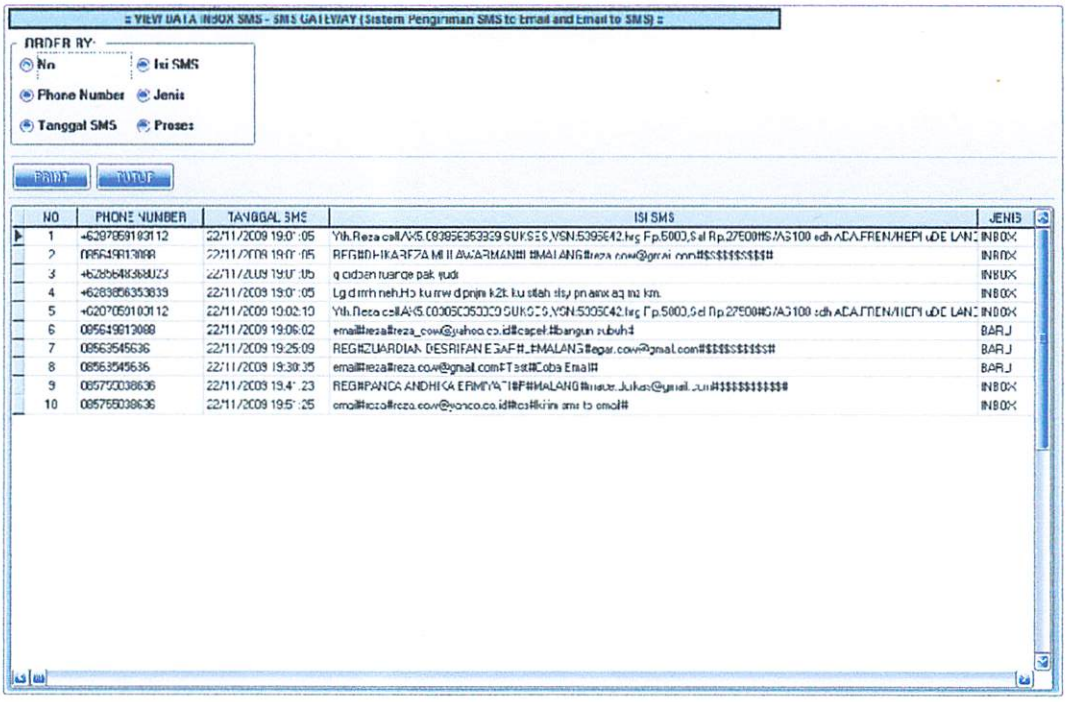


Gambar 4.44 Tampilan Data *Phonebook* Sesuai dengan *Phone Number* (Nomor Telepon)

Pengujian melihat daftar buku telepon pada HP SMS *Gateway* sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" pada Form Menu *View* Data *Phonebook* telah sukses dan berjalan dengan baik.

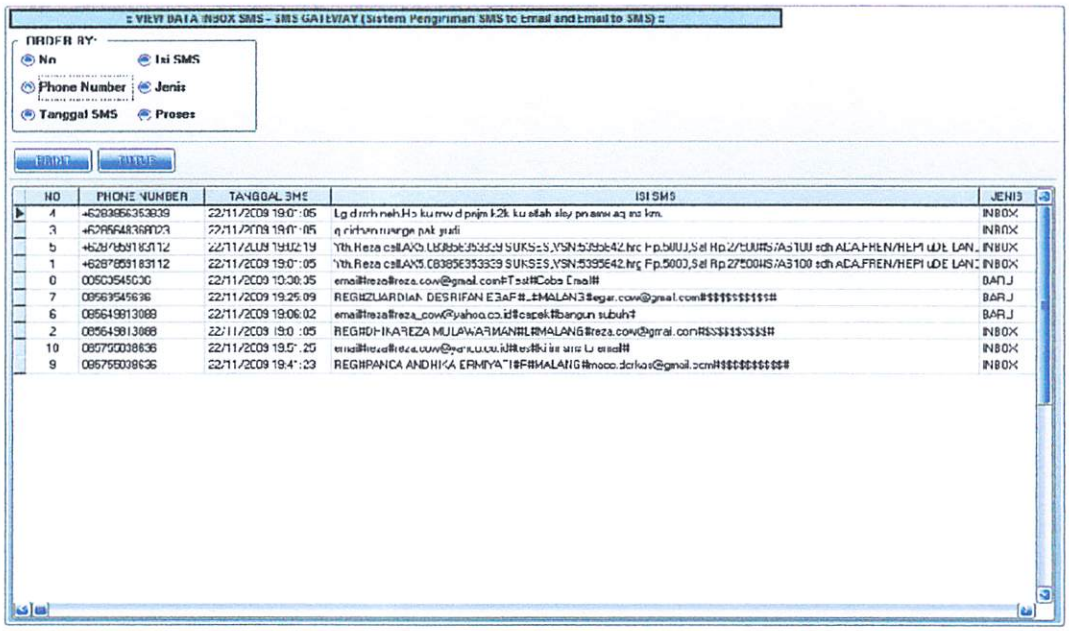
4.2.2.13. Pengujian Form Menu *View* Data *Inbox* SMS

Pengujian Form Menu *View* Data *Inbox* SMS dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada *component RadioGroup* "*Order By*:". Form Menu *View* Data *Inbox* SMS ini digunakan untuk melihat data SMS masuk yang ada pada HP SMS *Gateway*. Gambar 4.45 di bawah ini merupakan tampilan data *Inbox* SMS sesuai dengan Nomor.



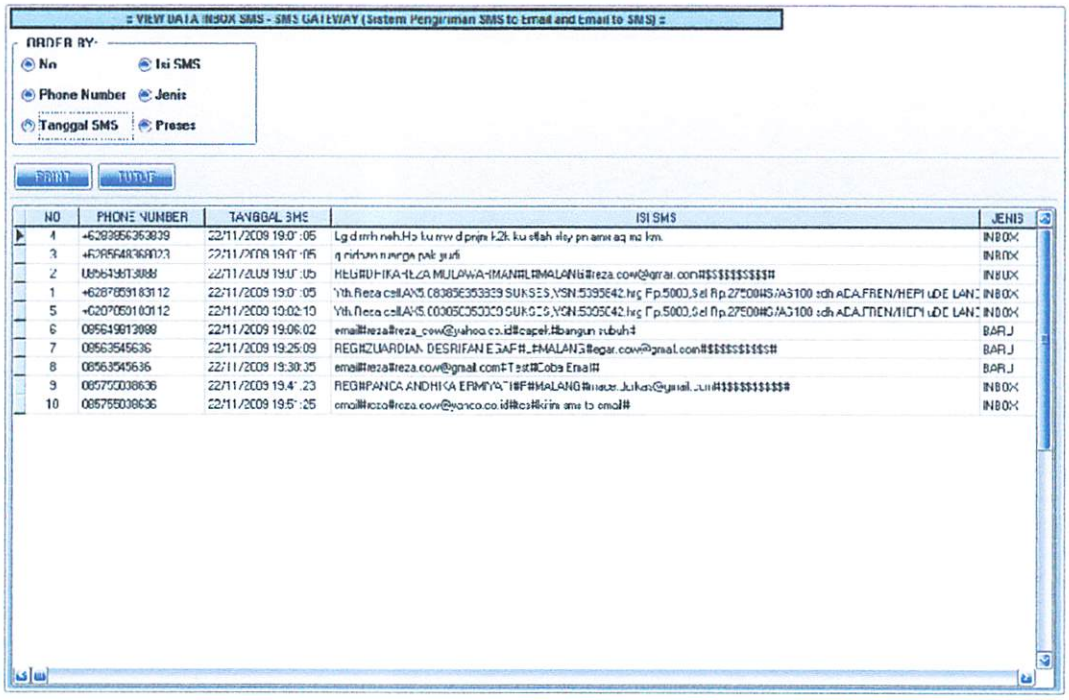
Gambar 4.45 Tampilan Data *Inbox* SMS Sesuai dengan Nomor

Gambar 4.46 di bawah ini merupakan tampilan data *Inbox* SMS sesuai dengan *Phone Number* (Nomor Telepon).



Gambar 4.46 Tampilan Data *Inbox* SMS Sesuai dengan *Phone Number* (Nomor Telepon)

Gambar 4.47 di bawah ini merupakan tampilan data *Inbox* SMS sesuai dengan Tanggal SMS.

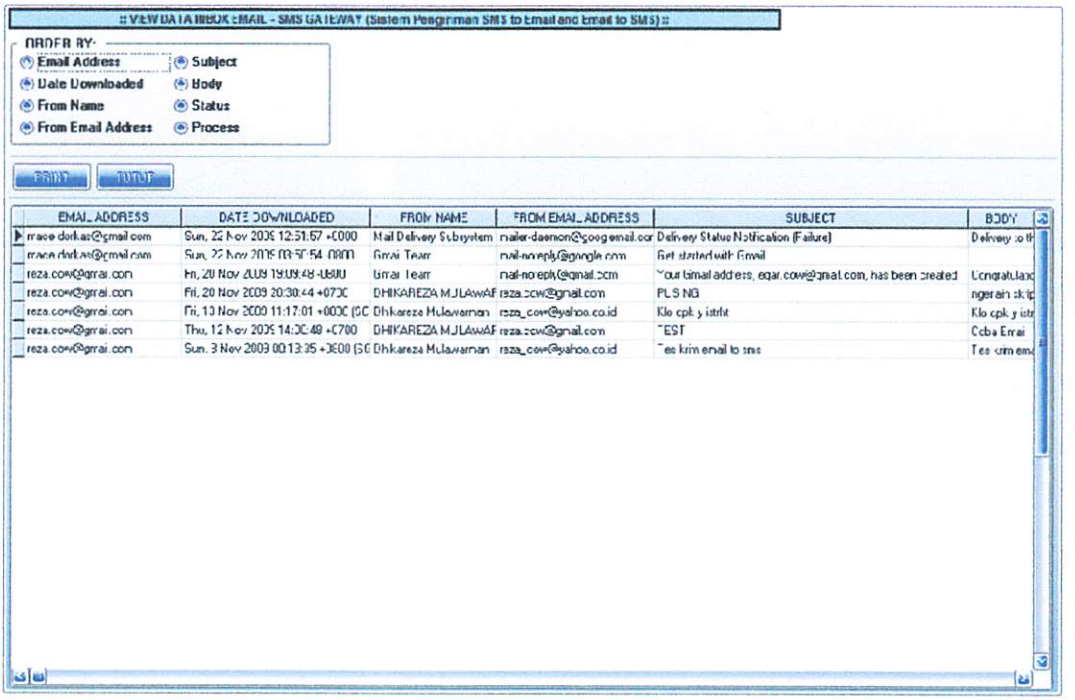


Gambar 4.47 Tampilan Data *Inbox* SMS Sesuai dengan Tanggal SMS

Pengujian melihat data SMS masuk yang ada pada HP SMS *Gateway* sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By:*" pada Form Menu *View* Data *Inbox* SMS telah sukses dan berjalan dengan baik.

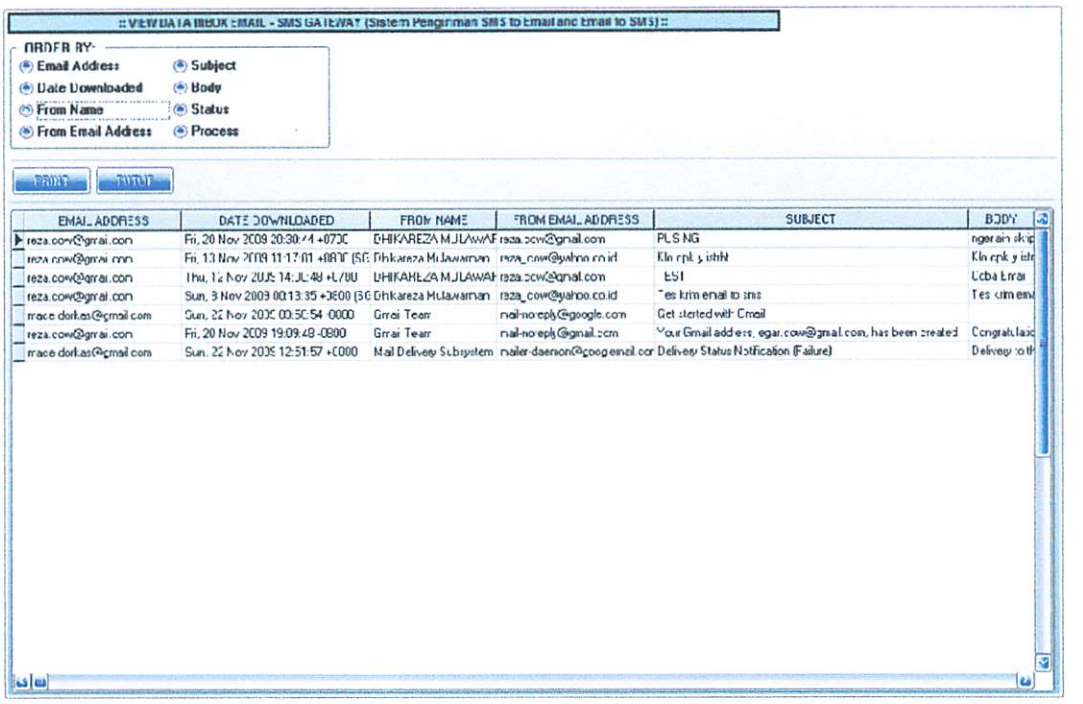
4.2.2.14. Pengujian Form Menu *View* Data *Inbox E-mail*

Untuk pengujian Form Menu *View* Data *Inbox E-mail* dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada component *RadioGroup* "*Order By:*". Form Menu *View* Data *Inbox E-mail* ini digunakan untuk melihat data *E-mail* yang masuk hasil *download E-mail* dari *webmail* gmail.com. Gambar 4.48 di bawah ini merupakan tampilan data *Inbox E-mail* sesuai dengan *E-mail Address* (Alamat *E-mail*).



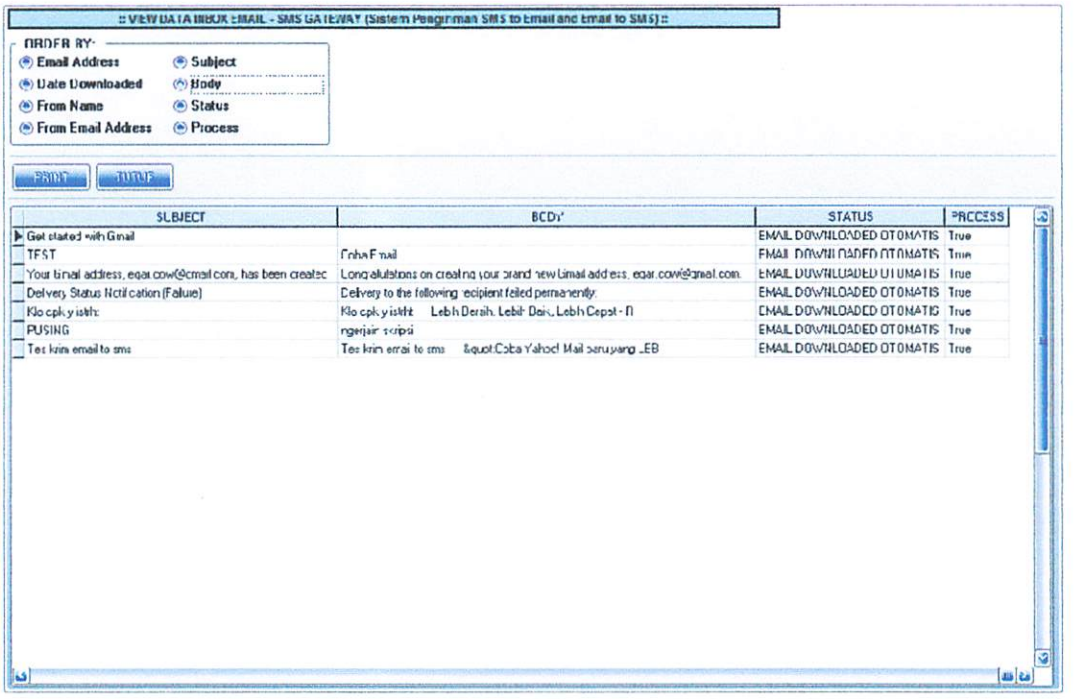
Gambar 4.48 Tampilan Data *Inbox E-mail* Sesuai dengan *E-mail Address* (Alamat *E-mail*)

Gambar 4.49 di bawah ini merupakan tampilan data *Inbox E-mail* sesuai dengan *From Name* (Dari Nama).



Gambar 4.49 Tampilan Data *Inbox E-mail* Sesuai dengan *From Name* (Dari Nama)

Gambar 4.50 di bawah ini merupakan tampilan data *Inbox E-mail* sesuai dengan *Body E-mail*.

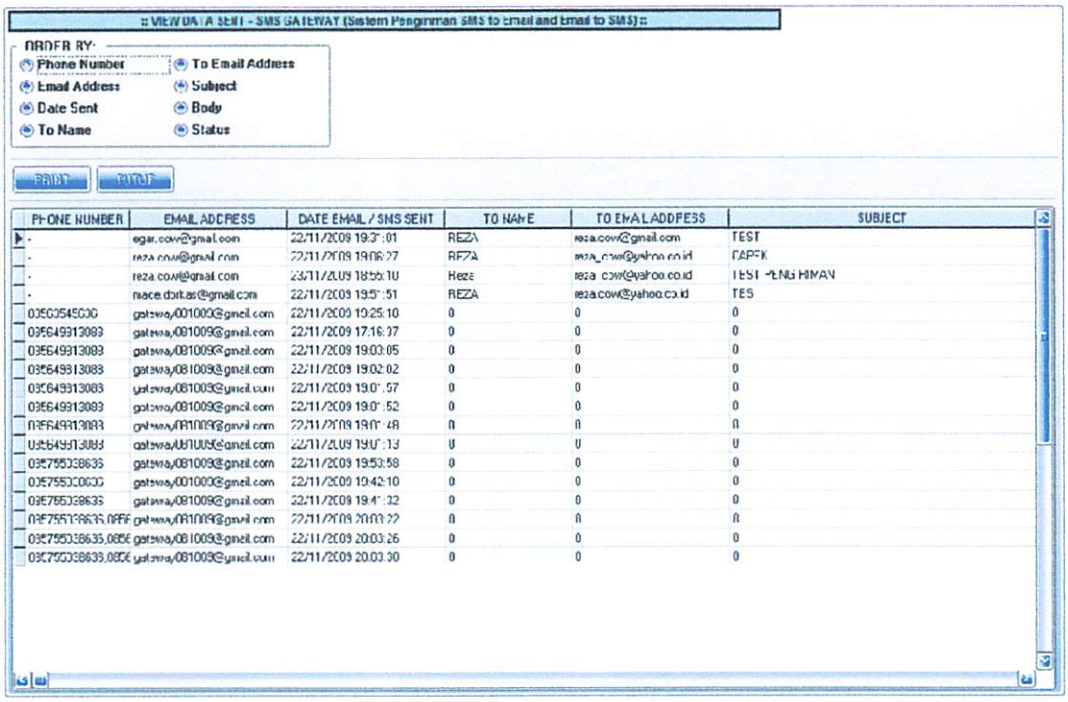


Gambar 4.50 Tampilan Data *Inbox E-mail* Sesuai dengan *Body E-mail*

Pengujian melihat data *E-mail* yang masuk hasil *download E-mail* dari *webmail* gmail.com sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" pada Form Menu *View* Data *Inbox E-mail* telah sukses dan berjalan dengan baik.

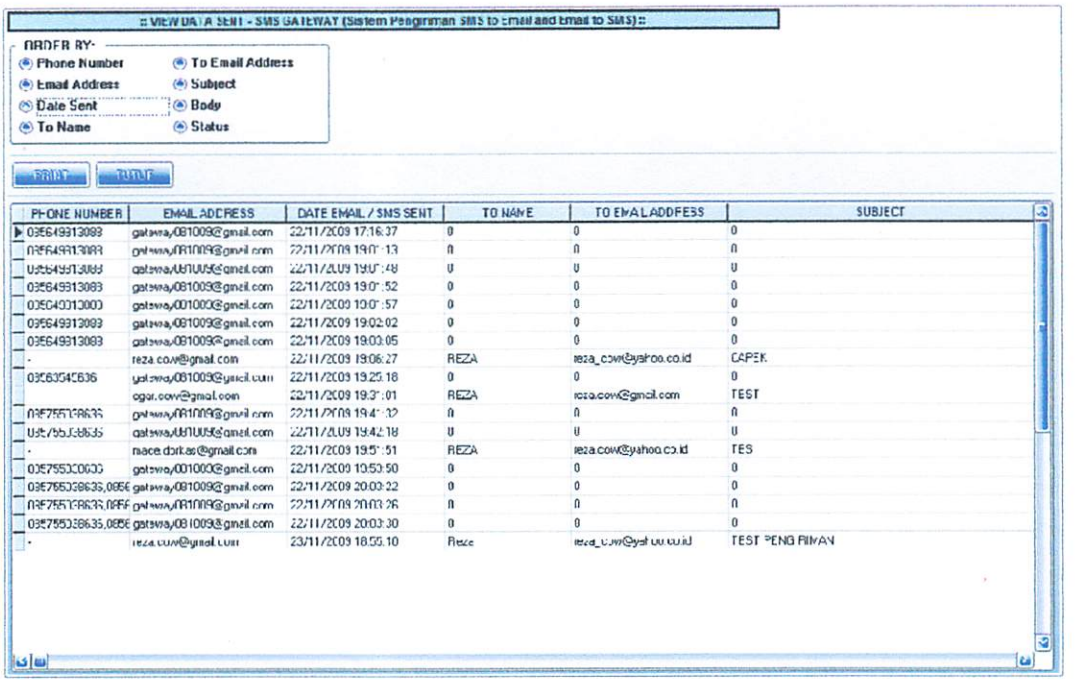
4.2.2.15. Pengujian Form Menu *View* Data *Sent*

Pengujian disini tujuannya untuk melihat data *E-mail* maupun SMS yang terkirim sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By*:" Gambar 4.51 di bawah ini merupakan tampilan data *Sent* sesuai dengan *Phone Number* (Nomor Telepon).



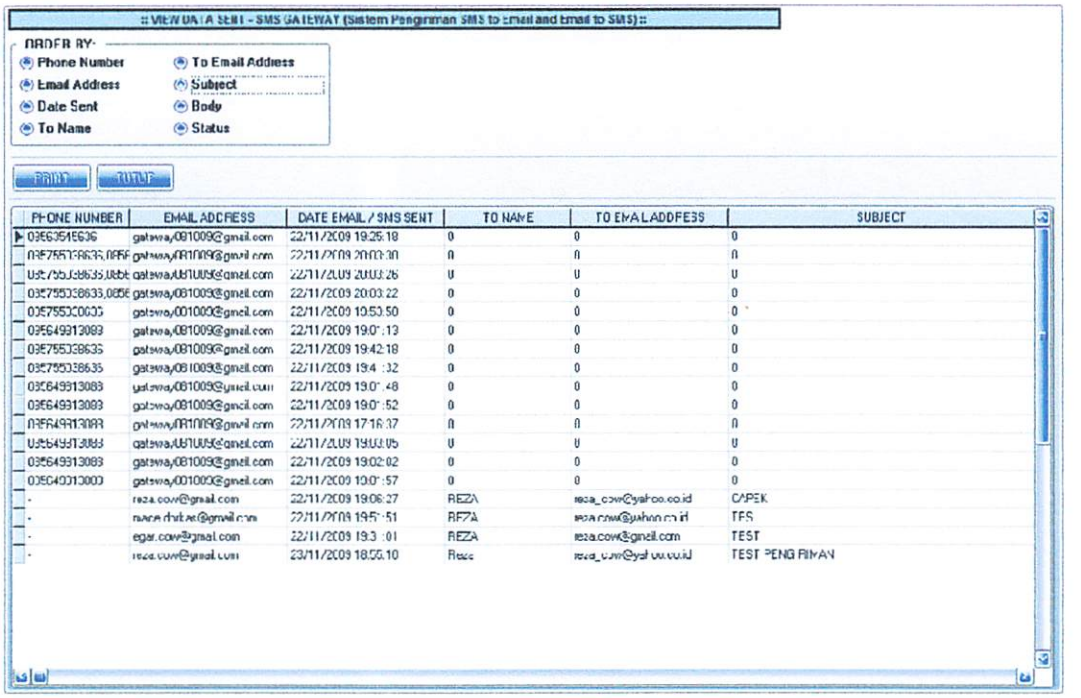
Gambar 4.51 Tampilan Data *Sent* Sesuai dengan *Phone Number* (Nomor Telepon)

Gambar 4.52 di bawah ini merupakan tampilan data *Sent* sesuai dengan *Date Sent* (Tanggal Mengirim).



Gambar 4.52 Tampilan Data *Sent* Sesuai dengan *Date Sent* (Tanggal Mengirim)

Gambar 4.53 di bawah ini merupakan tampilan data *Sent* sesuai dengan *Subject*.



Gambar 4.53 Tampilan Data *Sent* Sesuai dengan *Subject*

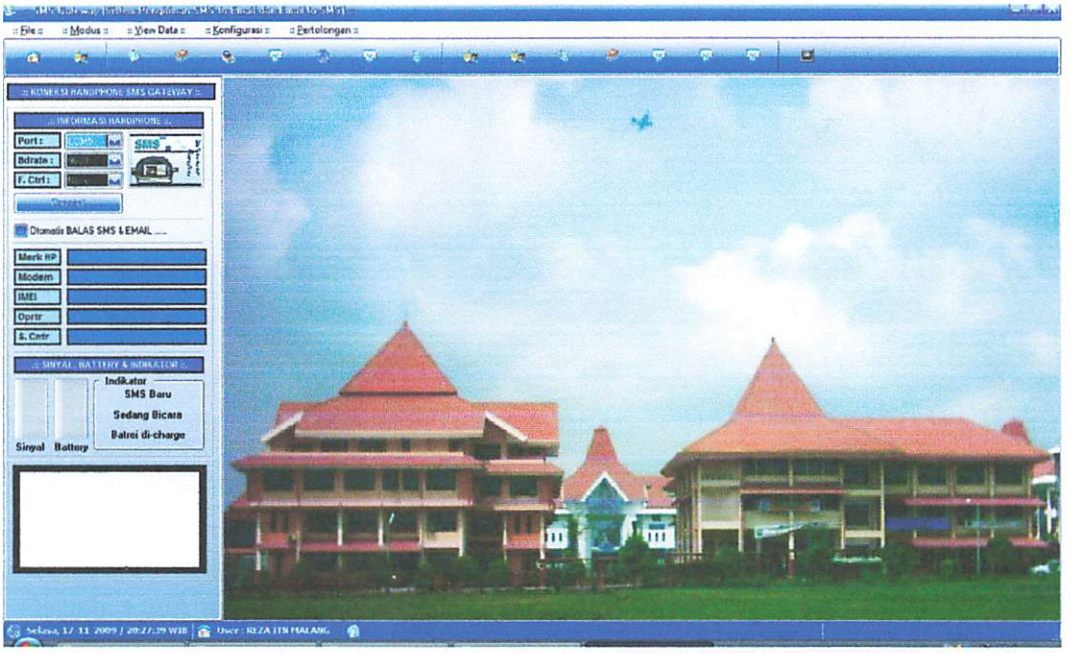
Pengujian melihat data *E-mail* maupun SMS yang terkirim sesuai dengan *RadioButton* yang di-klik pada *component RadioGroup* "*Order By:*" pada Form Menu *View* Data *Sent* telah sukses dan berjalan dengan baik.

Pengujian semua menu-menu yang ada di dalam Form Halaman Utama Administrator / Operator telah sukses dan berjalan dengan baik.

4.3. Pengujian Proses Pengiriman SMS *to E-mail* dan *E-mail to* SMS

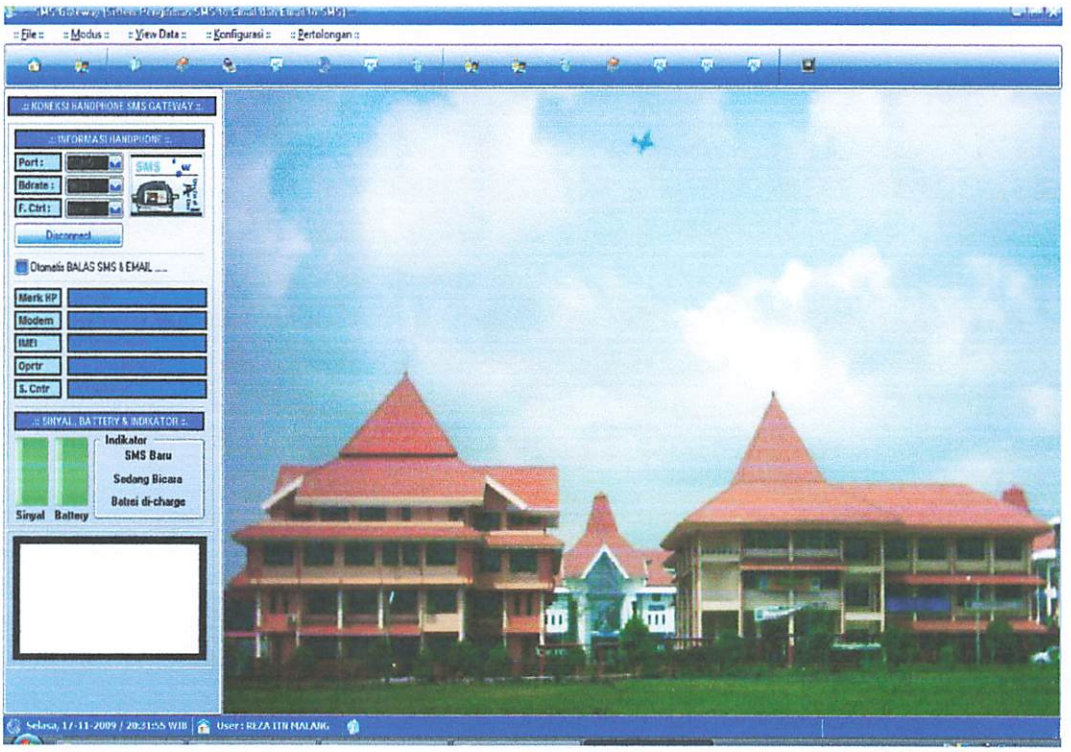
Untuk menguji proses pengiriman SMS *to E-mail* dan *E-mail to* SMS, diharapkan memenuhi prosedur yang telah ditetapkan. Prosedur yang ditetapkan adalah sebagai berikut :

1. Komputer yang digunakan pada *server* harus sudah terkoneksi ke INTERNET dan HP SMS *Gateway*, setelah itu jalankan aplikasi.
2. Melakukan proses *Login* sesuai dengan hak akses pada aplikasi.
3. Agar dapat mengoneksikan HP SMS *Gateway* dengan aplikasi, maka harus menyetting *Port Number* di COM 5 & *Baudrate* dengan nilai 9600 kemudian klik *button Connect* pada Form Halaman Utama. Gambar 4.54 di bawah ini menampilkan Form Halaman Utama sebelum terkoneksi dengan HP SMS *Gateway*.



Gambar 4.54 Form Halaman Utama Sebelum Terkoneksi dengan HP SMS *Gateway*

Gambar 4.55 di bawah ini menampilkan Form Halaman Utama sesudah terkoneksi dengan HP SMS *Gateway*.



Gambar 4.55 Form Halaman Utama Sesudah Terkoneksi dengan HP SMS *Gateway*

4. Mencentang *CheckBox* "Otomatis Balas SMS & *E-mail*" pada Form Halaman Utama agar aplikasi membalas otomatis SMS-SMS / *E-mail* - *E-mail* yang masuk ke dalam aplikasi.

Setelah semua prosedur di atas dipatuhi, maka pengujian proses pengiriman SMS *to E-mail* dan *E-mail to* SMS bisa dimulai.

4.3.1. Pengujian Proses Pengiriman SMS *to E-mail*

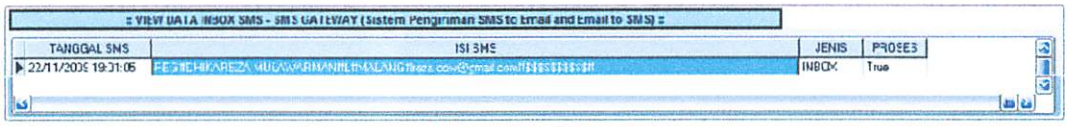
Pengujian proses pengiriman SMS *to E-mail* dilakukan dengan menerima SMS dari orang luar / calon pelanggan yang akan melakukan registrasi serta mengirimkan *E-mail* ke **TO_NAME** & **TO_EMAIL_ADDRESS** yang dituju berdasarkan nomor HP yang mengirim format SMS *to E-mail*.

Orang luar / calon pelanggan yang akan menggunakan fasilitas dari aplikasi ini harus melakukan registrasi terlebih dahulu dengan mengetik format SMS berikut:

REG#NAMA_LENGKAP#JENIS_KELAMIN#ALAMAT_LENGKAP# GMAIL_ADDRESS#GMAIL_PASSWORD# dan dikirimkan ke nomor HP SMS *Gateway* dari aplikasi ini, seperti yang terlihat pada Gambar 4.56:

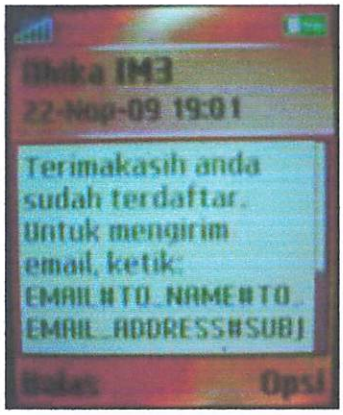
Gambar 4.56 Format SMS Registrasi pada HP Calon Pelanggan

Kemudian, format SMS Registrasi yang dikirimkan oleh calon pelanggan akan masuk ke TBL_SMS_INBOX yang bisa dilihat melalui Form Menu *View* Data *Inbox* SMS seperti yang terlihat pada Gambar 4.57:



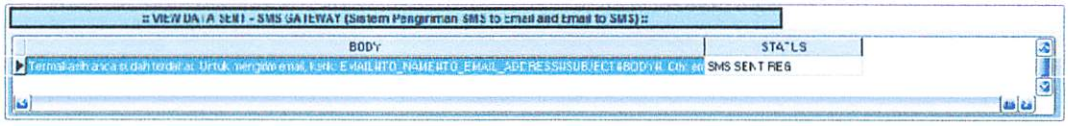
Gambar 4.57 Format SMS Registrasi Masuk ke Form Menu *View* Data *Inbox* SMS

Setelah calon pelanggan mengirimkan format SMS untuk registrasi ke nomor HP SMS *Gateway*, calon pelanggan tersebut akan menerima SMS *Reply* dari aplikasi seperti yang terlihat pada Gambar 4.58:



Gambar 4.58 SMS *Reply* yang Diterima oleh Pelanggan

Hasil dari pengiriman SMS *Reply* ke nomor HP pelanggan akan masuk ke TBL_SENT yang bisa dilihat pada Form Menu *View* Data *Sent* seperti yang terlihat pada Gambar 4.59:



Gambar 4.59 Hasil Pengiriman SMS *Reply* Masuk ke Form Menu *View* Data *Sent*

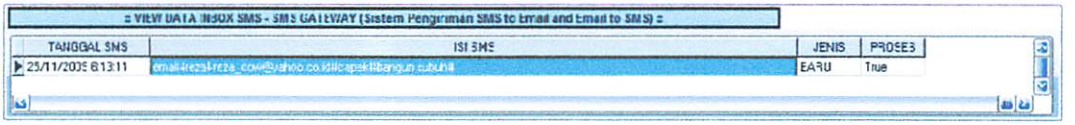
Kemudian, setelah melakukan registrasi, barulah pelanggan tersebut bisa mengirimkan SMS *to E-mail* dengan format SMS sebagai berikut:

EMAIL#TO_NAME#TO_EMAIL_ADDRESS#TO_SUBJECT#TO_BODY#

dan dikirimkan ke nomor HP SMS *Gateway* dari aplikasi ini, seperti yang terlihat pada Gambar 4.60:

Gambar 4.60 Format SMS untuk Kirim SMS *to E-mail* pada HP Pelanggan

Format SMS *to E-mail* yang dikirimkan oleh pelanggan akan masuk ke TBL_SMS_INBOX yang bisa dilihat melalui Form Menu *View* Data *Inbox* SMS seperti yang terlihat pada Gambar 4.61:

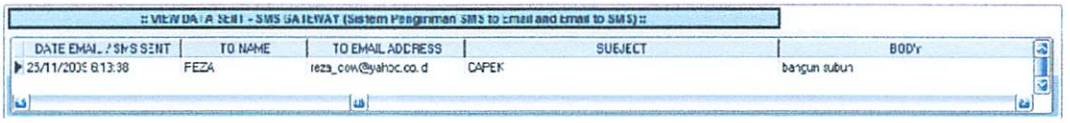


Gambar 4.61 Format SMS *to E-mail* Masuk ke Form Menu *View* Data *Inbox* SMS

Gambar 4.62 di bawah ini menampilkan hasil pengujian proses pengiriman SMS *to E-mail* pada *Account E-mail* tujuan.

Gambar 4.62 Hasil Pengujian Proses Pengiriman SMS to E-mail pada *Account E-mail* Tujuan

Hasil dari pengiriman SMS *to E-mail* ke alamat *E-mail* tujuan akan masuk ke TBL_SENT yang bisa dilihat pada Form Menu *View* Data *Sent* seperti yang terlihat pada Gambar 4.63:



DATE EMAIL / SMS SENT	TO NAME	TO EMAIL ADDRESS	SUBJECT	BODY
25/11/2005 8:13:38	FEZA	reza_cow@yahoo.co.id	CAPEK	bangun subuh

Gambar 4.63 Hasil Pengiriman SMS *to E-mail* Masuk ke Form Menu *View* Data *Sent*

Pengujian menerima SMS dari orang luar / calon pelanggan yang akan melakukan registrasi serta mengirimkan *E-mail* ke **TO_NAME** & **TO_EMAIL_ADDRESS** yang dituju berdasarkan nomor HP yang mengirim format SMS *to E-mail* telah sukses dan berjalan dengan baik.

4.3.2. Pengujian Proses Pengiriman *E-mail to* SMS

Untuk pengujian proses pengiriman *E-mail to* SMS dilakukan dengan men*download E-mail* pelanggan serta mengirimkannya ke nomor HP pelanggan. Bagi orang yang sudah registrasi, *E-mail* orang itu akan di*download* secara otomatis oleh aplikasi dan langsung dikirimkan ke nomor HP orang tersebut. Proses *E-mail transfer*nya sebagai berikut:

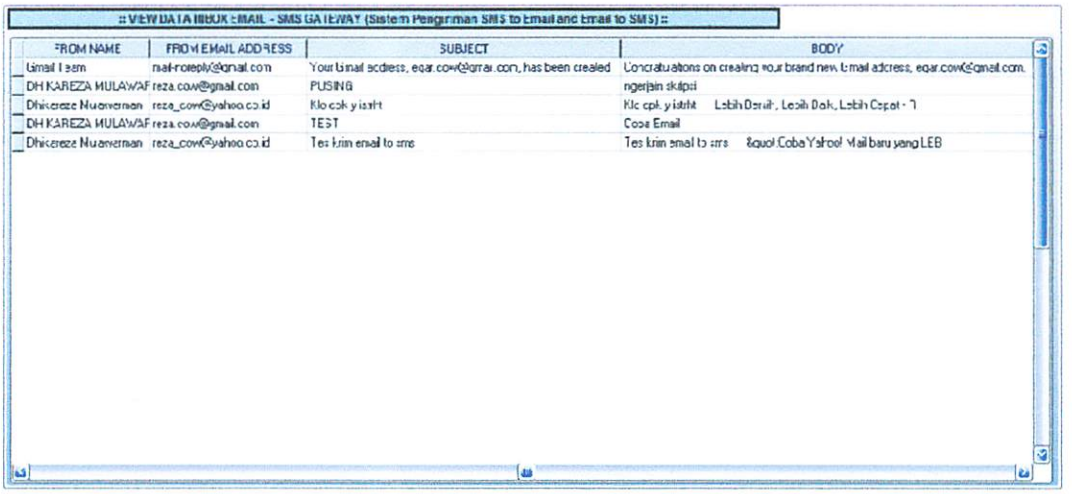
1. Ketika pelanggan registrasi ada inputan *gmail address* & *gmail password*nya. Format SMS *to E-mail*nya : **REG#NAMA_LENGKAP#JENIS_KELAMIN#ALAMAT_LENGKAP#GMAIL_ADDRESS#GMAIL_PASSWORD#**
2. Dengan *gmail address* & *gmail password* tersebut aplikasi *login* ke *server* gmail & mengambil *E-mail* pelanggan tersebut yang kemudian akan dikirimkan ke nomor HP pelanggan.

3. Untuk *login* ke *server* gmail tersebut aplikasi membutuhkan *username*, *password*, *incoming* & *outgoing* IMAP / POP3 *server address*, & *port number*.
4. Setelah itu aplikasi baru bisa *download E-mail – E-mail* pelanggan.

Gambar 4.64 di bawah ini menampilkan *E-mail - E-mail* yang terdapat pada *Inbox E-mail* salah satu *Account E-mail* pelanggan yang akan di*download* oleh aplikasi satu per satu dan kemudian akan dikirimkan ke nomor HP pelanggan dalam bentuk SMS.

Gambar 4.64 *E-mail - E-mail* yang Terdapat pada *Inbox E-mail* Salah Satu *Account E-mail* Pelanggan

E-mail-E-mail yang sudah di*download* oleh aplikasi, akan disimpan terlebih dahulu di TBL_EMAIL_INBOX dalam database, yang kemudian nanti *E-mail-E-mail* tersebut akan dikirimkan ke nomor HP pelanggan dalam bentuk SMS, seperti yang terlihat pada Gambar 4.65:

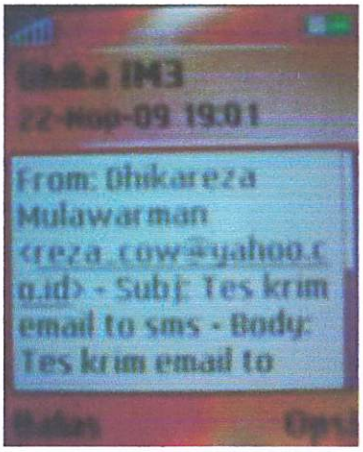


:: VIEW DATA INBOX E-MAIL - SMS GATEWAY (Sistem Pengirman SMS to Email and Email to SMS) ::

FROM NAME	FROM EMAIL ADDRESS	SUBJECT	BODY
Gmail Team	mail-noreply@gmail.com	Your Gmail address, eqar.cow@gmail.com, has been created	Congratulations on creating your brand new Gmail address, eqar.cow@gmail.com.
DH KAREZA MULAWAF	reza.cow@gmail.com	PUSING	ngerjain skripsi
Dhikereze Mulawarman	reza_cow@yahoo.co.id	Klo cpk yi istrht	Klo cpk yi istrht Lebih Dekat, Lebih Dok, Lebih Cepat - 1
DH KAREZA MULAWAF	reza.cow@gmail.com	TEST	Coba Email
Dhikereze Mulawarman	reza_cow@yahoo.co.id	Tes krim email to sms	Tes krim email to sms "Coba Yahoo! Mail baru yang LEB

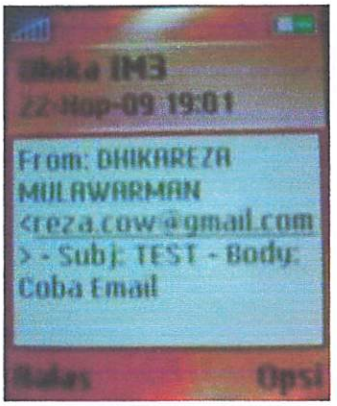
Gambar 4.65 Hasil *Download E-mail* dari *Server* GMAIL pada Form Menu *View* Data *Inbox E-mail*

Gambar 4.66 di bawah ini menampilkan salah satu HP pelanggan menerima *E-mail* ke 1 dalam bentuk SMS.



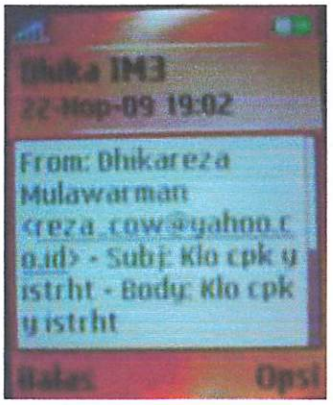
Gambar 4.66 Salah Satu HP Pelanggan Menerima *E-mail* ke 1 dalam Bentuk SMS

Gambar 4.67 di bawah ini menampilkan salah satu HP pelanggan menerima *E-mail* ke 2 dalam bentuk SMS.



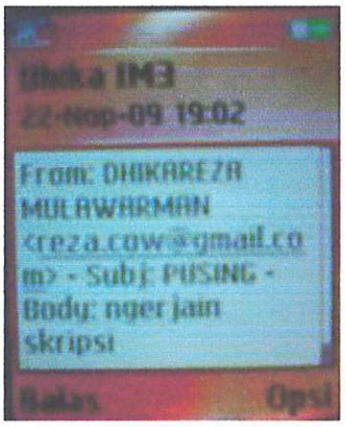
Gambar 4.67 Salah Satu HP Pelanggan Menerima *E-mail* ke 2 dalam Bentuk SMS

Gambar 4.68 di bawah ini menampilkan salah satu HP pelanggan menerima *E-mail* ke 3 dalam bentuk SMS.



Gambar 4.68 Salah Satu HP Pelanggan Menerima *E-mail* ke 3 dalam Bentuk SMS

Gambar 4.69 di bawah ini menampilkan salah satu HP pelanggan menerima *E-mail* ke 4 dalam bentuk SMS.

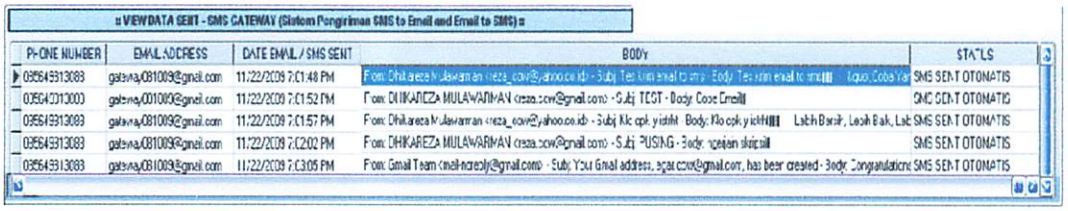


Gambar 4.69 Salah Satu HP Pelanggan Menerima *E-mail* ke 4 dalam Bentuk SMS

Gambar 4.70 di bawah ini menampilkan salah satu HP pelanggan menerima *E-mail* ke 5 dalam bentuk SMS.

Gambar 4.70 Salah Satu HP Pelanggan Menerima *E-mail* ke 5 dalam Bentuk SMS

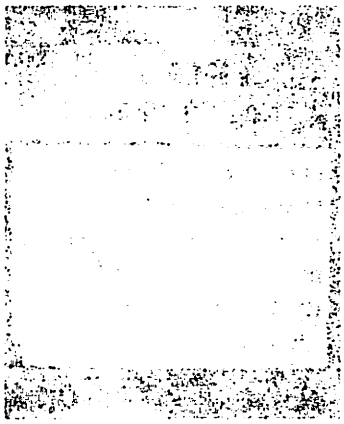
Hasil dari pengiriman semua *E-mail-E-mail* yang ada pada *Account E-mail* pelanggan ke nomor HP pelanggan, akan masuk ke TBL_SENT yang bisa dilihat melalui Form Menu *View* Data *Sent* seperti Gambar 4.71 berikut ini:

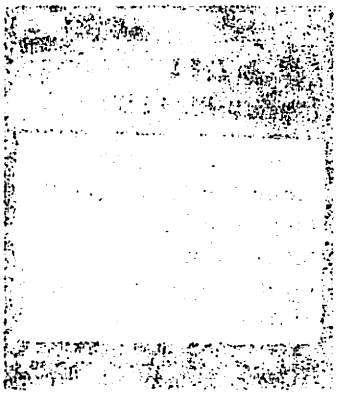


:: VIEW DATA SENT - SMS GATEWAY (Sistem Pengiriman SMS to Email and Email to SMS) ::

PHONE NUMBER	EMAIL ADDRESS	DATE EMAIL / SMS SENT	BODY	STATUS
085643913083	gateway001083@gmail.com	11/22/2009 7:01:48 PM	From: Dhikareza Mulawarman <reza_cow@yahoo.co.id> - Subj: Test kirim email to sms - Body: Test kirim email to sms!!!!!! - Kaus Coba Ya	SMS SENT OTOMATIS
085643913083	gateway001083@gmail.com	11/22/2009 7:01:52 PM	From: DHIKAREZA MULAWARMAN <reza.cow@gmail.com> - Subj: TEST - Body: Coba Email!	SMS SENT OTOMATIS
085643913083	gateway001083@gmail.com	11/22/2009 7:01:57 PM	From: Dhikareza Mulawarman <reza_cow@yahoo.co.id> - Subj: Klo cpk yi ichhh - Body: Klo cpk yi ichhhhh!!!! Lebih Baik, Lebih Baik, Lebi	SMS SENT OTOMATIS
085643913083	gateway001083@gmail.com	11/22/2009 7:02:02 PM	From: DHIKAREZA MULAWARMAN <reza.cow@gmail.com> - Subj: PUSING - Body: ngerjain skripsi!	SMS SENT OTOMATIS
085643913083	gateway001083@gmail.com	11/22/2009 7:03:05 PM	From: Gmail Team <mail-noreply@gmail.com> - Subj: Your Gmail address. egar.cow@gmail.com, has been created - Body: Congratulations	SMS SENT OTOMATIS

Gambar 4.71 *E-mail-E-mail* yang Berhasil Dikirim ke Pelanggan Masuk ke Form Menu *View* Data *Sent*





Tabel 4.2 di bawah ini merupakan hasil pengujian proses pengiriman *E-mail to* SMS.

Tabel 4.2 Hasil Pengujian Proses Pengiriman *E-mail to* SMS

E-mail Ke	Dari Alamat *E-mail*	Ke Nomor HP	Tanggal & Jam *E-mail* Terkirim	Perbedaan Waktu
1	Dhikareza Mulawarman <reza_cow@yahoo.co.id>	085649813088	22/11/2009 19:01:48	4 detik
2	DHIKAREZA MULAWARMAN <reza.cow@gmail.com>		22/11/2009 19:01:52	5 detik
3	Dhikareza Mulawarman <reza_cow@yahoo.co.id>		22/11/2009 19:01:57	
4	DHIKAREZA MULAWARMAN <reza.cow@gmail.com>		22/11/2009 19:02:02	1 menit 3 detik
5	Gmail Team <mail-noreply@gmail.com		22/11/2009 19:03:05	

Pengujian men*download* *E-mail* pelanggan serta mengirimkannya ke nomor HP pelanggan telah sukses dan berjalan dengan baik.

4.3.3. Pengujian Proses Unreg

Pengujian Proses *Unreg* dilakukan dengan cara mengetik format SMS untuk *Unreg*, kemudian mengirimkannya ke nomor HP SMS *Gateway* seperti Gambar 4.72 berikut ini:

Gambar 4.72 Format SMS untuk Proses *Unreg*

SMS *Unreg* yang dikirimkan oleh pelanggan akan masuk ke TBL_SMS_INBOX dalam database yang bisa dilihat melalui Form *View* Data *Inbox* SMS seperti Gambar 4.73 berikut ini:

:: VIEW DATA INBOX SMS - SMS GATEWAY (Sistem Pengirimen SMS to Email and Email to SMS) ::

TANGGAL SMS	ISI SMS	JENIS	PROSES
12/10/2009 0:32:20 PM	unreg#smsemail#	INBOX	True

Gambar 4.73. SMS *Unreg* Pelanggan Masuk ke Form *View* Data *Inbox* SMS

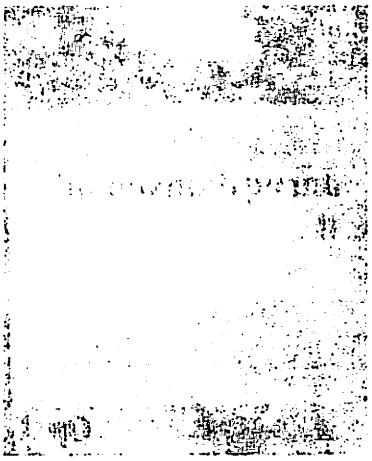
Selanjutnya, aplikasi akan memproses SMS *Unreg* tersebut dengan cara merubah status pelanggan yang masih *Registered* menjadi status *Unregistered* pada TBL_PELANGGAN yang bisa dilihat pada Form *View* Data Pelanggan, seperti Gambar 4.74 berikut ini:

:: VIEW DATA PELANGGAN SMS GATEWAY (Sistem Pengirimen SMS to Email and Email to SMS) ::

NO	DATE REGISTER	FULLNAME	GENDER	HOME ADDRESS	PHONE NUMBER	EMAIL ADDRESS	STATUS
1	12/19/2009 6:39:52 PM	DHIKA REZA M. ILAWARMAN	1	MALANG	+6285649813088	reza.cow@gmail.com	UNREGISTERED

Gambar 4.74. Status Pelanggan Berubah Menjadi Unregistered pada Form *View* Data Pelanggan

Kemudian, pelanggan akan mendapatkan SMS *Reply* sebagai informasi bahwa pelanggan tersebut sudah tidak menjadi pelanggan dari aplikasi SMS *Gateway* seperti Gambar 4.75 berikut ini:



Gambar 4.12 Form Ganti Password Pengguna

Setelah proses yang dilakukan oleh pengguna akan masuk ke [illegible] dalam gambar yang ini adalah model form dari SMS pengguna Gambar 4.13 berikut ini

[illegible]

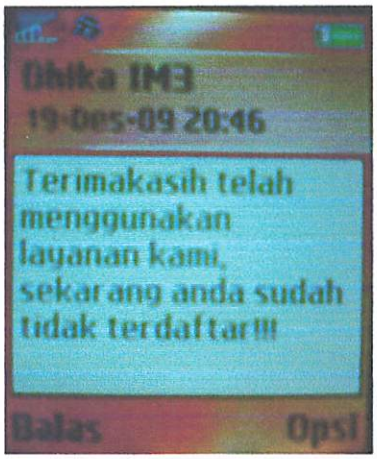
Gambar 4.13 SMS yang Dikirim Admin ke Pengguna Lain Saat SMS

Ketiduran dipilih dan mengirim SMS yang berisi deskripsi dengan cara memilih salah pengguna yang masih Aktivasi [illegible] pada [illegible] yang hal diatur oleh [illegible] Data pengguna sepert [illegible] 4.14 berikut ini

[illegible]

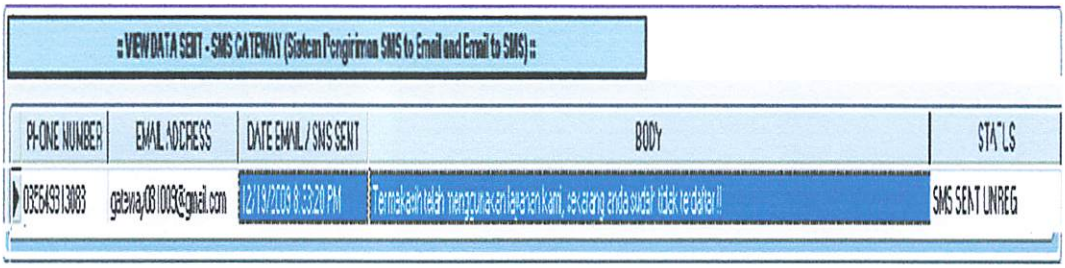
Gambar 4.14 Hasil Pengguna Berhasil Mengirim Informasi pada Form Data Pengguna

Ketika pengguna akan melihat informasi yang [illegible] pengguna lain [illegible] SMS yang sepert Gambar 4.15 berikut ini



Gambar 4.75 SMS *Reply* dari Aplikasi SMS *Gateway* yang Diterima Pelanggan

SMS *Reply* yang dikirimkan ke nomor HP pelanggan akan masuk ke TBL_SENT yang bisa dilihat melalui Form *View* Data *Sent* seperti yang terlihat pada Gambar 4.76 berikut ini:



Gambar 4.76 SMS *Reply* Masuk ke Form *View* Data *Sent*

Pengujian proses *Unreg* dan meng-*update* status pelanggan pada TBL_PELANGGAN serta mengirimkan SMS *Reply* ke nomor HP pelanggan telah sukses dan berjalan dengan baik.

4.4. Pengujian Seberapa Cepat *E-mail* Terkirim ke Alamat *E-mail* Tujuan

Pengujian ini dilakukan dengan cara mengirim *E-mail* secara:

1. Manual, melalui aplikasi SMS *Gateway* dengan menggunakan alamat *E-mail* asal reza.cow@gmail.com / w.leethpx@gmail.com
2. Otomatis, melalui 3 HP pelanggan dengan menggunakan 3 alamat *E-mail* pelanggan yaitu: mace.dorkas@gmail.com, egar.cow@gmail.com, dan

r3z4.c0w0@gmail.com ke 6 alamat *E-mail* tujuan serta ke 3 *webmail* yang berbeda-beda.

Tabel 4.3 di bawah ini menjelaskan hasil dari pengujian seberapa cepat *E-mail* terkirim ke alamat *E-mail* tujuan.

Tabel 4.3 Hasil dari Pengujian Seberapa Cepat *E-mail* Terkirim ke Alamat *E-mail* Tujuan

Alamat Email Asal	Tanggal / Waktu Dikirim	Ke Nama	Ke Alamat Email Tujuan	Subject Email	Body Email	Tanggal / Waktu Terkirim
reza.cow@gmail.com	14/02/2010 05:56:44 WIB	Ermiyati	mace.dorkas@gmail.com	Pengujian Seberapa Cepat E-mail Terkirim ke Alamat E-mail Tujuan.	Pengujian ini dilakukan dengan cara mengirim E-mail secara manual melalui aplikasi SMS Gateway dan menggunakan 3 nomor HP pelanggan ke 3 alamat E-mail yang berbeda-beda.	14/02/2010 05:56:45 WIB
ace.dorkas@gmail.com	14/02/2010 06:10:55 WIB	Egar	egar.cow@gmail.com	Berita Dari Egar Kemarin	Ke rumah teman...habis kecelakaan,, Di balehan...Dekat jalan ciliwung,,	14/02/2010 06:10:56 WIB
egar.cow@gmail.com	14/02/2010 06:22:15 WIB	Reza	reza.cow@gmail.com	Pngjian Sbrp Cpt Email Trkrm k Almt Email Tjuan	Pngjian Ni Dlkkn dgn Cra Mngrm Email Scra Mnual Mllui Aplksi SMS Gtwy	14/02/2010 06:22:16 WIB
3z4.c0w0@gmail. com	14/02/2010 19:14:35 WIB	Wily	w.leethpx@gmail.com			14/02/2010 19:14:36 WIB
reza.cow@gmail.com	14/02/2010 19:29:11 WIB	Reza	reza_cow@hotmail.com	Pengujian Proses Pengiriman E-mail to SMS	Untuk pengujian proses pengiriman E-mail to SMS dilakukan dengan mendownload E-mail pelanggan serta mengirimkannya ke nomor HP pelanggan.	14/02/2010 19:29:13 WIB
3z4.c0w0@gmail. com	14/02/2010 19:59:53 WIB			Pengujian Form Menu *View* Data *Sent*	Pengujian disini tujuannya untuk melihat data *E-mail* maupun SMS yang terkirim	14/02/2010 19:59:55 WIB
ace.dorkas@gmail.com	14/02/2010 20:05:37 WIB			Pengujian Proses Pengiriman SMS *to E-mail*	Dlkkn dngn mnrma SMS dr og lwr / cln plnggn yg akn mlkkn rgstrsi	14/02/2010 20:05:39 WIB
v.leethpx@gmail. com	14/02/2010 20:29:51 WIB	Reza	reza_cow@yahoo.co.id	Pengujian Form Menu *View* Data *Inbox E-mail*	Untuk pengujian Form Menu *View* Data *Inbox E-mail* dilakukan dengan meng-klik satu per satu *RadioButton* yang terdapat pada component *RadioGroup* "*Order By*:"	14/02/2010 20:29:54 WIB
ace.dorkas@gmail.com	14/02/2010 20:37:57 WIB			Pengujian Form Menu *View* Data *Inbox* SMS	dlkkn dngn mngklik 1 / 1 RdioBtton yg trdpt pd cmpnnt RdioGroup Order By:	14/02/2010 20:38:00 WIB
3z4.c0w0@gmail. com	14/02/2010 20:50:37 WIB			Pengujian Form Menu *View* Data *Phonebook*		14/02/2010 20:50:40 WIB

4.5. Pengujian Kapasitas Berapa Maksimal Isi *Body E-mail* Bisa Terkirim ke Alamat *E-mail* Tujuan dengan Settingan 80 Detik

Untuk pengujian ini dilakukan dengan mengirim *E-mail* secara otomatis melalui 3 HP pelanggan dan menggunakan 3 alamat *E-mail* pelanggan ke 3 alamat *E-mail* tujuan yang berbeda-beda.

Tabel 4.4 di bawah ini menjelaskan hasil dari pengujian kapasitas berapa maksimal isi *body E-mail* bisa terkirim ke alamat *E-mail* tujuan dengan settingan 80 detik.

Tabel 4.4 Hasil dari Pengujian Kapasitas Berapa Maksimal Isi *Body E-mail* Bisa Terkirim ke Alamat *E-mail* Tujuan dengan Settingan 80 detik

Kasus dan Hasil Uji (Data yang Dimasukkan < = 160 karakter)						
Alamat Email Asal	Tanggal / Waktu Dikirim	Ke Nama	Ke Alamat Email Tujuan	Subject Email	Body Email	Tanggal / Waktu Terkirim
ıce.dorkas@gmail. com	14/02/2010 06:10:55 WIB	Egar (4 karakter)	egar.cow@gmail.com (18 karakter)	Berita Dari Egar Kemarin (24 karakter)	Ke rumah teman...habis kecelakaan,, Di balehan...Dekat jalan ciliwung,, (71 karakter)	14/02/2010 06:10:56 WIB
egar.cow@gmail.com	14/02/2010 06:22:15 WIB	Reza (4 karakter)	reza.cow@gmail.com (18 karakter)	Pngjian Sbrp Cpt Email Trkrm k Almt Email Tjuan (47 karakter)	Pngjian Ni Dlkkn dgn Cra Mngrm Email Scra Mnual Mllui Aplksi SMS Gtwy (69 karakter)	14/02/2010 06:22:16 WIB
3z4.c0w0@gmail. com	14/02/2010 19:14:35 WIB	Wily (4 karakter)	w.leethpx@gmail.com (19 karakter)			14/02/2010 19:14:36 WIB
Hasil: *E-mail* sukses terkirim ke alamat *E-mail* tujuan						
Kasus dan Hasil Uji (Data yang Dimasukkan > 160 karakter)						
Alamat Email Asal	Tanggal / Waktu Dikirim	Ke Nama	Ke Alamat Email Tujuan	Subject Email	Body Email	Hasil
ıce.dorkas@gmail. com	15/02/2010 20:52:45 WIB	Egar (4 karakter)	egar.cow@gmail.com (18 karakter)	Pengujian Sistem (16 karakter)	Pengujian ini bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini bila diterapkan dalam sistem..... (113 karakter)	*E-mail* tidak terkirim dan aplikasi mengirimkan SMS *Reply* "Maaf, format yang anda kirimkan salah. Ketik: EMAIL#TO_NAME#TO_EMAIL_ADDRESS#SUBJECT#BODY#. Cth: email#reza#reza_itn@gmail.com#Test#Coba Email#" berulang-ulang kali ke nomor HP pelanggan yang mengirim *E-mail*
egar.cow@gmail.com	15/02/2010 22:31:48 WIB	Reza (4 karakter)	reza.cow@gmail.com (18 karakter)	Metode Pengoperasian (20 karakter)	Pada pengoperasian aplikasi ini diharapkan pengoperasi aplikasi memenuhi prosedur yang ditetapkan, yaitu..... (109 karakter)	
3z4.c0w0@gmail. com	15/02/2010 22:34:47 WIB	Wily (4 karakter)	w.leethpx@gmail.com (19 karakter)	Pngjian Aplksi Srvr (19 karakter)	Pngjian pd aplksi srvr dlkkn agr kmngknn trjdi kslhan pada alikasi server dapat diidentifikasikan sejak awal (109 karakter)	

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari perancangan dan implementasi Aplikasi Layanan Pengiriman *E-mail to* SMS dan SMS *to E-mail* yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Dalam proses *Registrasi*, apabila orang yang melakukan *registrasi* tersebut banyak dan waktu (pada detik ke 20) untuk aplikasi SMS *Gateway* memproses *registrasi* tersebut habis, maka aplikasi akan mengulanginya lagi pada detik ke 20 berikutnya. Selama menunggu ke detik 20, aplikasi akan men*download E-mail-Email* yang terdapat pada *account gmail* pelanggan yang sudah *registrasi* kemudian mengirimnya ke nomor HP pelanggan.
2. Pada proses SMS *to E-mail*, tidak bisa melampirkan gambar / meng-*attachment* gambar, dan jumlah karakter dalam menulis format SMS *to E-mail* hingga isi *body E-mail* tidak lebih dari 160 karakter.
3. Dalam proses *E-mail to* SMS, *E-mail-E-mail* yang dapat ditampilkan *body E-mail*-nya hanyalah *E-mail-E-mail* yang ber-*text* biasa, dengan jumlah karakter tidak lebih dari 160 karakter, dan tidak ber-*attachment*. Apabila data pelanggannya banyak, maka aplikasi akan men*download E-mail* pelanggan-pelanggan tersebut satu per satu (*looping* berdasarkan ID *auto number* di TBL_PELANGGAN).

4. Untuk pelanggan yang sudah tidak mau menggunakan layanan dari aplikasi ini dapat melakukan proses *Unreg* dengan mengirimkan format SMS *Unreg* ke nomor HP SMS *Gateway*nya.
5. Pada pengujian seberapa cepat *E-mail* dapat terkirim ke alamat *E-mail* tujuan ditentukan dari modem yang digunakan untuk terhubung ke internet. Apabila menggunakan modem *Speedy*, *E-mail* dapat segera terkirim ke alamat *E-mail* tujuan dengan perbedaan waktu 1 detik antara waktu mengirim *E-mail* dengan terkirimnya *E-mail* tersebut, sedangkan apabila menggunakan modem HP atau USB modem GSM / CDMA, *E-mail* yang dikirim tidak dapat segera terkirim ke alamat *E-mail* tujuan, karena tergantung sinyal yang diterima oleh modem-modem tersebut dari BTS (*Base Transceiver Station*) dan kecepatan *upload E-mail* yang dimiliki oleh modem-modem tersebut. Perbedaan waktu antara waktu mengirim *E-mail* dengan terkirimnya *E-mail* tersebut ke alamat *E-mail* tujuan menggunakan modem HP atau USB modem GSM / CDMA mencapai 2 – 3 detik.
6. Dalam pengujian kapasitas berapa maksimal isi *body E-mail* bisa terkirim ke alamat *E-mail* tujuan dengan settingan 80 Detik adalah berjumlah 160 karakter dihitung dari jumlah karakter format SMS *to E-mail*. Misalnya ada format SMS *to E-mail* seperti berikut:

   email#egar#egar.cow@gmail.com#Pengujian Sistem#Pengujian ini bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini bila diterapkan dalam sistem.....#

Jumlah karakter dari format SMS *to E-mail* di atas adalah 161 karakter, dan apabila format tersebut dikirim oleh salah satu pelanggan, maka *E-mail*nya tidak bisa dikirim oleh aplikasi serta aplikasi akan mengirimkan SMS *Reply* "**Maaf, format yang anda kirimkan salah. Ketik: EMAIL#TO_NAME#TO_EMAIL_ADDRESS#SUBJECT#BODY#. Cth: email#reza#reza_itn@gmail.com#Test#Coba Email#**" berulang-ulang kali ke nomor HP pelanggan tersebut.

5.2. Saran

Aplikasi ini masih jauh dari sempurna dan masih banyak kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah:

1. Proses pendaftaran pelanggan yang dapat menggunakan aplikasi ini hendaknya disediakan pendaftaran dalam bentuk *E-mail*.
2. Pada Proses Pengiriman *Email to* SMS dan SMS *to E-mail* hendaknya tidak lagi harus terbatasi dengan 160 karakter dan bisa meng-*attachment* gambar.
3. Kerjasama dengan operator jaringan telepon seluler agar aplikasi dapat diimplementasikan dengan baik, serta dapat memudahkan pemberitahuan informasi tentang aplikasi ini ke seluruh pelanggan salah satu operator jaringan telepon seluler yang ada di Indonesia.
4. Hendaknya dipertimbangkan trafik SMS atau *Email* masuk dan keluar guna mengoptimalkan layanan ini.

DAFTAR PUSTAKA

1. **Indriyawan Eko**, "*membangun Sistem andal Delphi*", Andi, 2008.

2. **Kosasih Asep**, "*Algoritma & Pemrograman dengan Bahasa Delphi 5.0*", Yrama Widya, Bandung, 2006.

3. **Malik Jamaludin Jaja**, "*Kumpulan Latihan Program Delphi*", Andi Offset, 2006.

4. http://brightside.wordpress.com/2006/01/20/aspicore-sms-gateway-support-page/.

5. http://www.globalkomputer.com/Bahasan/Komunikasi-Data.

6. http://www.globalkomputer.com/Bahasan/KomunikasiData/Topik/PDU/Subtopik/Command.

LAMPIRAN-LAMPIRAN

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK KOMPUTER & INFORMATIKA S-1
Jl. Karanglo KM. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama	: DHIKAREZA MULAWARMAN
NIM	: 05 12 555
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer & Informatika
Judul Skripsi	: Pembuatan Aplikasi Layanan Pengiriman Email to SMS dan SMS to Email Berbasis SMS Gateway

Dipertahankan di hadapan majelis penguji Skripsi jenjang Strata satu (S-1) pada :

Hari	: Senin
Tanggal	: 8 Februari 2010
Dengan Nilai	: 82,7 (A)

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji	Sekretaris Majelis Penguji
(Ir. H. Sidik Noertjahjono, MT)	**(Ir. F. Yudi Limpraptono, MT)**
NIP.Y. 1028700163	**NIP.Y. 1039500274**

ANGGOTA PENGUJI

Dosen Penguji I	Dosen Penguji II
(M. Ibrahim Ashari, ST.MT)	**(Sandy Nataly M, S.Kom)**
NIP.P. 1030100358	

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK KOMPUTER & INFORMATIKA S-1
Jl. Karanglo KM. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	: Dhikareza Mulawarman
NIM	: 05 12 555
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer & Informatika
Masa Bimbingan	: 03 Agustus 2009 s/d 03 Februari 2010.
Judul Skripsi	: Pembuatan Aplikasi Layanan Pengiriman Email to SMS dan SMS to Email Berbasis SMS Gateway

Penguji / Tanggal	Uraian	Paraf
Penguji I 8 Februari 2010	Tambahkan pengujian ; Tambahkan hasil pengujian di kesimpulan ; Beri pengantar pada tabel dan gambar ; Batasan masalah.	
	Diuji dengan kapasitas berapa maksimal isi Email bisa terkirim dengan settingan 80 detik ; Dilakukan uji coba seberapa cepat Email terkirim ke alamat Email tujuan.	
Penguji II 8 Februari 2010	Tambahkan hasil pengujian sistem pada Bab IV dengan cara percobaan selama beberapa kali ; Saran ditambahkan pengembangan aplikasi selanjutnya harus bagaimana / rencana pengembangan.	

Mengetahui,

Dosen Pembimbing I

(Ir. F. Yudi Limpraptono, MT)
NIP.Y.1039500274

Dosen Pembimbing II

(Dr. Eng. Aryuanto Soetedjo, ST.MT)
NIP.Y.1030800417

Dosen Penguji,

Penguji I

(M. Ibrahim Ashari, ST.MT)
NIP.P. 1030100358

Penguji II

(Sandy Nataly M, S.Kom)

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura – gura No.2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Dhikareza Mulawarman
Nim : 05.12.555
Masa Bimbingan : 03 Agustus 2009 s/d 03 Februari 2010
Judul Skripsi : Pembuatan Aplikasi Layanan Pengiriman E-Mail to SMS dan SMS to E-Mail Berbasis SMS Gateway

No	Tanggal	Uraian	Paraf Pembimbing
1	11/11 09	Perbaiki Bab III	
2	27/11 09	Bab II – III	
3	10/12 09	Bab IV / V Perbaiki kesimpulan	
4	10/12 09	Demo Program Skripsi	
5	12/12 09	Makalah Seminar Hasil	
6	12/1 2010	Acc akhir	
7			
8			
9			
10			

Malang, 12 Januari 2010
Dosen pembimbing I

Ir. F. Yudi Limpraptono, MT
NIP .Y.1039500274

Form S-4b

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura – gura No.2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Dhikareza Mulawarman
Nim : 05.12.555
Masa Bimbingan : 03 Agustus 2009 s/d 03 Februari 2010
Judul Skripsi : Pembuatan Aplikasi Layanan Pengiriman E-Mail to SMS dan SMS to E-Mail Berbasis SMS Gateway

No	Tanggal	Uraian	Paraf Pembimbing
1	14/8 2009	Revisi Bab I	
2	9/9/09	Bab 1 + "Proses Email 2 SMS dan sebaliknya" !!	
3	10/11/09	Proses email transfer ???	
4	8/12/09	Bab 4-5 ✓	
5	11/12/09	Demo S/W ✓	
6	17/12/2009	Makalah Seminar Hasil ✓	
7	14/1/2001	Laporan Skripsi ✓	
8			
9			
10			

Malang, 14 Januari 2010
Dosen pembimbing II

Dr. Eng. Aryuanto Soetedjo, ST. MT
NIP.Y. 1030800417

Form S-4b

LISTING PROGRAM

```
unit U_Utama;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, dxBarExtItems, dxBar,
dxDockControl, DB, ADODB, XComDrv,
  ExtCtrls, dxfBackGround, Menus,
cxGraphics, cxControls, dxStatusBar,
  ImgList, dxDockPanel, OleCtrls,
ShockwaveFlashObjects_TLB, StdCtrls,
  JLAqua, ButtonComps, ComCtrls,
SXSkinControl, SXSkinPanel, jpeg,
gsm_sms, IniFiles,
  ClipBrd, StrUtils, DateUtils, Registry,
  QDialogs, Mask, DBCtrls, MPlayer,
SUIDBCtrls, dxfCheckBox, Grids,
  DBGrids, dxfProgressBar, sSkinManager,
ToolWin, sToolBar, sStatusBar,
  sCheckBox, sLabel, Buttons, sBitBtn,
sPanel, sMemo, sEdit, sGroupBox,
  sBevel, CHILKATMAILLib2_TLB,
QuickRpt, QRCtrls;

type
  TF_Utama = class(TForm)
    PopupMenu1: TPopupMenu;
    Delete1: TMenuItem;
    Timer1: TTimer;
    XComm1: TXComm;
    ADOConnection1: TADOConnection;
    TBalasOtomatis: TTimer;
    ImageList1: TImageList;
    Image1: TImage;
    ADOQuery1: TADOQuery;
    Timer2: TTimer;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Login2: TMenuItem;
    N1: TMenuItem;
    Logout2: TMenuItem;
    N2: TMenuItem;
    Keluar2: TMenuItem;
    Modus1: TMenuItem;
    InfoSMS2: TMenuItem;
    KirimSMS2: TMenuItem;
    Phonebook2: TMenuItem;
    Panggilan2: TMenuItem;
    Broadcast2: TMenuItem;
    Unreg: TMenuItem;
    N4: TMenuItem;
    Konfigurasi1: TMenuItem;
    OtomatisBalasSMS1: TMenuItem;
    N5: TMenuItem;
    Skins1: TMenuItem;
    Pertolongan1: TMenuItem;
    Help2: TMenuItem;
    N7: TMenuItem;
    About1: TMenuItem;
    sToolBar1: TsToolBar;
    bLogin: TToolButton;
    ToolButton4: TToolButton;
    tbPhonebook: TToolButton;
    tbPhonecall: TToolButton;
    tbKirimSMS: TToolButton;
    tbKirimEmail: TToolButton;
    tbRegistrasi: TToolButton;
    ToolButton15: TToolButton;
    tbInfoSMS: TToolButton;
    SB1: TsStatusBar;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    mLatar: TMenuItem;
    mButtonShow: TMenuItem;
    mButtonKlik: TMenuItem;
    MenuSuara1: TMenuItem;
    pInfoSMS: TsPanel;
    SelButton23: TSelButton;
    SelButton24: TSelButton;
    cbSumber1: TComboBox;
    Bevel24: TBevel;
    prgBar2: TdxfProgressBar;
    Bevel21: TBevel;
    lvSMS1: TListView;
    lblSMS1: TMemo;
    swf3: TShockwaveFlash;
    lbJumlah: TsLabel;
    pRegistrasi: TsPanel;
    SelButton11: TSelButton;
    SelButton25: TSelButton;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    SelButton26: TSelButton;
    SelButton27: TSelButton;
    DBEdit3: TDBEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Button7: TButton;
    Bevel1: TBevel;
    DBGrid2: TDBGrid;
    bCekSMS: TsBitBtn;
    bTutupPanel: TsBitBtn;
    pBroadcast: TsPanel;
```

Bevel74: TBevel;
Bevel75: TBevel;
Bevel78: TBevel;
Bevel76: TBevel;
swf7: TShockwaveFlash;
SelButton157: TSelButton;
DBGrid16: TDBGrid;
txtSisaBroadcast: TSelButton;
txtPesanBroadcast: TMemo;
txtNomorBroadcast: TEdit;
SelButton154: TSelButton;
SelButton155: TSelButton;
SelButton156: TSelButton;
cbReport2: TdxfCheckBox;
cbReply2: TdxfCheckBox;
cbFlash2: TdxfCheckBox;
pKirimSMS: TsPanel;
Bevel11: TBevel;
Bevel12: TBevel;
Bevel13: TBevel;
Bevel14: TBevel;
swf4: TShockwaveFlash;
SelButton12: TSelButton;
SelButton13: TSelButton;
SelButton14: TSelButton;
cbReport1: TdxfCheckBox;
cbReply1: TdxfCheckBox;
cbFlash1: TdxfCheckBox;
txtNomor1: TEdit;
Label5: TLabel;
txtPesan1: TMemo;
bSisa: TSelButton;
SelButton21: TSelButton;
DBGrid1: TDBGrid;
SelButton22: TSelButton;
DBGrid4: TDBGrid;
pPhonebook: TsPanel;
lvPBook1: TListView;
Bevel17: TBevel;
Bevel18: TBevel;
swf5: TShockwaveFlash;
txtNamaPB1: TEdit;
txtNomorPB1: TEdit;
SelButton18: TSelButton;
SelButton17: TSelButton;
SelButton16: TSelButton;
cbJenisPB1: TComboBox;
SelButton15: TSelButton;
prgBar1: TdxfProgressBar;
Bevel16: TBevel;
pTelepon: TsPanel;
DBGrid3: TDBGrid;
Bevel19: TBevel;
Bevel20: TBevel;
swf6: TShockwaveFlash;
txtNomorCall1: TEdit;
SelButton20: TSelButton;
SelButton19: TSelButton;
pKirimEmail: TsPanel;
SelButton28: TSelButton;
N8: TMenuItem;
KirimEMAIL1: TMenuItem;
BroadcastEMAIL1: TMenuItem;
SelButton29: TSelButton;
SelButton30: TSelButton;
SelButton31: TSelButton;
SelButton32: TSelButton;
SelButton33: TSelButton;
eUsername: TsEdit;
ePassword: TsEdit;
eToEmail: TsEdit;
eSubject: TsEdit;
eBody: TsMemo;
bSendEmail: TsBitBtn;
sBitBtn1: TsBitBtn;
bLoginEmail: TsBitBtn;
sBevel1: TsBevel;
DBGrid5: TDBGrid;
sBevel2: TsBevel;
DBGrid6: TDBGrid;
sBevel3: TsBevel;
sLabel1: TsLabel;
SelButton34: TSelButton;
eToName: TsEdit;
sLabel2: TsLabel;
ChilkatMailMan21: TChilkatMailMan2;
sPanel1: TsPanel;
Bevel4: TBevel;
SelButton1: TSelButton;
SelButton2: TSelButton;
SelButton3: TSelButton;
SelButton4: TSelButton;
cbFC: TComboBox;
cbBaud: TComboBox;
swf2: TShockwaveFlash;
bKonek: TsBitBtn;
Bevel7: TBevel;
cbAuto: TCheckBox;
Bevel5: TBevel;
SelButton6: TSelButton;
SelButton7: TSelButton;
SelButton8: TSelButton;
SelButton9: TSelButton;
SelButton10: TSelButton;
bService: TSelButton;
bOperator: TSelButton;
bIMEI: TSelButton;
bModem: TSelButton;
bMerk: TSelButton;
Bevel6: TBevel;

SelButton5: TSelButton;
pbSinyal: TProgressBar;
pbBattery: TProgressBar;
GroupBox1: TGroupBox;
indSMSBaru: TLabel;
indPanggilan: TLabel;
indCharged: TLabel;
Label8: TLabel;
Label7: TLabel;
Bevel9: TBevel;
SXSkinPanel1: TSXSkinPanel;
swfDesktop: TShockwaveFlash;
Bevel8: TBevel;
mpLatar: TMediaPlayer;
mp1: TMediaPlayer;
ListSMS: TListView;
SelButton35: TSelButton;
sSkinManager1: TsSkinManager;
N9: TMenuItem;
tbBroadcastSMS: TToolButton;
tbData1: TToolButton;
tbData2: TToolButton;
tbData4: TToolButton;
tbData5: TToolButton;
tbData3: TToolButton;
bLogoutEmail: TsBitBtn;
bKirimBroadcast: TsBitBtn;
bClearBroadcast: TsBitBtn;
sBitBtn4: TsBitBtn;
bKirim: TsBitBtn;
bClear: TsBitBtn;
sBitBtn3: TsBitBtn;
bBaca1: TsBitBtn;
bSimpan1: TsBitBtn;
bBaru1: TsBitBtn;
sBitBtn7: TsBitBtn;
sBitBtn2: TsBitBtn;
bStop: TsBitBtn;
sBitBtn6: TsBitBtn;
tbUserAccount: TToolButton;
sLabel3: TsLabel;
bAttachFile: TsBitBtn;
OpenDialog1: TOpenDialog;
Background1: TMenuItem;
N3: TMenuItem;
Browse1: TMenuItem;
N10: TMenuItem;
DBComboBox1: TDBComboBox;
DBEdit4: TDBEdit;
DBEdit5: TDBEdit;
DBComboBox2: TDBComboBox;
DBNavigator1: TDBNavigator;
Bevel2: TBevel;
SelButton36: TSelButton;
SelButton37: TSelButton;
cbPort: TComboBox;
SelButton38: TSelButton;
SelButton39: TSelButton;
DBEdit6: TDBEdit;
Label1: TLabel;
ViewData1: TMenuItem;
Data1: TMenuItem;
Data3: TMenuItem;
Data2: TMenuItem;
Data5: TMenuItem;
Data6: TMenuItem;
Data4: TMenuItem;
Data7: TMenuItem;
N11: TMenuItem;
N12: TMenuItem;
UserAccount1: TMenuItem;
Cascade1: TMenuItem;
Tile1: TMenuItem;
CloseAll1: TMenuItem;
pUserAccount: TPanel;
SelButton40: TSelButton;
SelButton41: TSelButton;
SelButton42: TSelButton;
SelButton43: TSelButton;
SelButton44: TSelButton;
SelButton45: TSelButton;
SelButton46: TSelButton;
SelButton47: TSelButton;
Label2: TLabel;
Bevel3: TBevel;
SelButton48: TSelButton;
Bevel10: TBevel;
eVCH_FULLNAME: TDBEdit;
eVCH_GENDER: TDBComboBox;
eDTM_DATE_REGISTER: TDBEdit;
eVCH_PASSWORD: TDBEdit;
eVCH_USER_ID: TDBEdit;
eVCH_EMAIL_ADDRESS: TDBEdit;
eVCH_PHONE_NUMBER: TDBEdit;
eVCH_ROLE: TDBComboBox;
DBNavigator2: TDBNavigator;
bInsertUserAccount: TButton;
bEditUserAccount: TButton;
bOKUserAccount: TButton;
bCancelUserAccount: TButton;
bRefreshUserAccount: TButton;
bTutupUserAccount: TButton;
bDeleteUserAccount: TButton;
eVCH_HOME_ADDRESS: TDBEdit;
DBGrid7: TDBGrid;
pViewDataUser: TPanel;
SelButton49: TSelButton;
vDataUserOrderBy: TRadioGroup;
Bevel15: TBevel;
DBGrid8: TDBGrid;

bPrintViewDataUser: TsBitBtn;
sBitBtn8: TsBitBtn;
Bevel22: TBevel;
pViewDataLogin: TPanel;
SelButton50: TSelButton;
Bevel23: TBevel;
Bevel25: TBevel;
vDataLoginOrderBy: TRadioGroup;
sBitBtn5: TsBitBtn;
sBitBtn9: TsBitBtn;
DBGrid9: TDBGrid;
pViewDataPelanggan: TPanel;
SelButton51: TSelButton;
Bevel26: TBevel;
Bevel27: TBevel;
vDataPelangganOrderBy: TRadioGroup;
sBitBtn10: TsBitBtn;
sBitBtn11: TsBitBtn;
DBGrid10: TDBGrid;
pViewDataPhonebook: TPanel;
SelButton52: TSelButton;
Bevel28: TBevel;
Bevel29: TBevel;
vDataPhonebookOrderBy: TRadioGroup;
sBitBtn12: TsBitBtn;
sBitBtn13: TsBitBtn;
DBGrid11: TDBGrid;
pViewDataInboxSMS: TPanel;
SelButton53: TSelButton;
Bevel30: TBevel;
Bevel31: TBevel;
vDataInboxSMSOrderBy: TRadioGroup;
sBitBtn14: TsBitBtn;
sBitBtn15: TsBitBtn;
DBGrid12: TDBGrid;
pViewDataInboxEmail: TPanel;
SelButton54: TSelButton;
Bevel32: TBevel;
Bevel33: TBevel;
vDataInboxEmailOrderBy: TRadioGroup;
sBitBtn16: TsBitBtn;
sBitBtn17: TsBitBtn;
DBGrid13: TDBGrid;
pViewDataSent: TPanel;
SelButton55: TSelButton;
Bevel34: TBevel;
Bevel35: TBevel;
vDataSentOrderBy: TRadioGroup;
sBitBtn18: TsBitBtn;
sBitBtn19: TsBitBtn;
DBGrid14: TDBGrid;
pQR: TsPanel;
QR_VIEWDATAUSER: TQuickRep;
QRSubDetail1: TQRSubDetail;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRLabel6: TQRLabel;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
QRLabel9: TQRLabel;
QRLabel10: TQRLabel;
QRLabel11: TQRLabel;
QRBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRSubDetail2: TQRSubDetail;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText8: TQRDBText;
QRDBText9: TQRDBText;
QRBand2: TQRBand;
QRLabel12: TQRLabel;
QR_VIEWDATALOGIN: TQuickRep;
QRSubDetail3: TQRSubDetail;
QRLabel13: TQRLabel;
QRLabel14: TQRLabel;
QRLabel17: TQRLabel;
QRLabel18: TQRLabel;
QRLabel19: TQRLabel;
QRLabel15: TQRLabel;
QRBand3: TQRBand;
QRLabel22: TQRLabel;
QRLabel23: TQRLabel;
QRSubDetail4: TQRSubDetail;
QRDBText10: TQRDBText;
QRDBText11: TQRDBText;
QRDBText13: TQRDBText;
QRDBText15: TQRDBText;
QRDBText16: TQRDBText;
QRDBText17: TQRDBText;
QRBand4: TQRBand;
QRLabel24: TQRLabel;
QR_VIEWDATAPELANGGAN: TQuickRep;
QRSubDetail5: TQRSubDetail;
QRLabel16: TQRLabel;
QRLabel20: TQRLabel;
QRLabel21: TQRLabel;
QRLabel27: TQRLabel;
QRLabel28: TQRLabel;
QRLabel29: TQRLabel;
QRLabel30: TQRLabel;
QRLabel34: TQRLabel;
QRBand5: TQRBand;

```
  QRLabel31: TQRLabel;
  QRLabel32: TQRLabel;
  QRSubDetail6: TQRSubDetail;
  QRDBText12: TQRDBText;
  QRDBText14: TQRDBText;
  QRDBText18: TQRDBText;
  QRDBText21: TQRDBText;
  QRDBText22: TQRDBText;
  QRDBText24: TQRDBText;
  QRDBText25: TQRDBText;
  QRDBText26: TQRDBText;
  QRBand6: TQRBand;
  QRLabel33: TQRLabel;
  QR_VIEWDATAPHONEBOOK:
TQuickRep;
  QRSubDetail7: TQRSubDetail;
  QRLabel25: TQRLabel;
  QRLabel35: TQRLabel;
  QRLabel38: TQRLabel;
  QRBand7: TQRBand;
  QRLabel39: TQRLabel;
  QRLabel40: TQRLabel;
  QRSubDetail8: TQRSubDetail;
  QRDBText19: TQRDBText;
  QRDBText23: TQRDBText;
  QRDBText27: TQRDBText;
  QRBand8: TQRBand;
  QRLabel41: TQRLabel;
  QR_VIEWDATAINBOXSMS:
TQuickRep;
  QRSubDetail9: TQRSubDetail;
  QRLabel26: TQRLabel;
  QRLabel36: TQRLabel;
  QRLabel37: TQRLabel;
  QRLabel45: TQRLabel;
  QRLabel46: TQRLabel;
  QRLabel47: TQRLabel;
  QRBand9: TQRBand;
  QRLabel42: TQRLabel;
  QRLabel43: TQRLabel;
  QRSubDetail10: TQRSubDetail;
  QRDBText20: TQRDBText;
  QRDBText28: TQRDBText;
  QRDBText29: TQRDBText;
  QRDBText30: TQRDBText;
  QRDBText31: TQRDBText;
  QRDBText32: TQRDBText;
  QRBand10: TQRBand;
  QRLabel44: TQRLabel;
  QR_VIEWDATAINBOXEMAIL:
TQuickRep;
  QRSubDetail11: TQRSubDetail;
  QRLabel48: TQRLabel;
  QRLabel53: TQRLabel;
  QRLabel54: TQRLabel;
  QRLabel55: TQRLabel;
  QRLabel56: TQRLabel;
  QRBand11: TQRBand;
  QRLabel57: TQRLabel;
  QRLabel58: TQRLabel;
  QRSubDetail12: TQRSubDetail;
  QRDBText33: TQRDBText;
  QRDBText38: TQRDBText;
  QRDBText39: TQRDBText;
  QRDBText40: TQRDBText;
  QRDBText41: TQRDBText;
  QRBand12: TQRBand;
  QRLabel59: TQRLabel;
  QRLabel52: TQRLabel;
  QRDBText34: TQRDBText;
  QRLabel49: TQRLabel;
  QRDBText35: TQRDBText;
  QR_VIEWDATASENT: TQuickRep;
  QRSubDetail13: TQRSubDetail;
  QRLabel50: TQRLabel;
  QRLabel51: TQRLabel;
  QRLabel60: TQRLabel;
  QRLabel61: TQRLabel;
  QRLabel62: TQRLabel;
  QRLabel63: TQRLabel;
  QRLabel64: TQRLabel;
  QRBand13: TQRBand;
  QRLabel65: TQRLabel;
  QRLabel66: TQRLabel;
  QRSubDetail14: TQRSubDetail;
  QRDBText36: TQRDBText;
  QRDBText37: TQRDBText;
  QRDBText42: TQRDBText;
  QRDBText43: TQRDBText;
  QRDBText44: TQRDBText;
  QRDBText45: TQRDBText;
  QRDBText46: TQRDBText;
  QRBand14: TQRBand;
  QRLabel67: TQRLabel;
  QRLabel68: TQRLabel;
  QRDBText47: TQRDBText;
  tbData6: TToolButton;
  tbData7: TToolButton;
  ToolButton1: TToolButton;
  ToolButton2: TToolButton;
  tbExit: TToolButton;
  procedure FormCreate(Sender: TObject);
  procedure Timer1Timer(Sender:
TObject);
  procedure Timer2Timer(Sender:
TObject);
  procedure FormActivate(Sender:
TObject);
  procedure FormClose(Sender: TObject;
var Action: TCloseAction);
```

```
  procedure XComm1Data(Sender:
TObject; const Received: Cardinal);
  procedure cbAutoClick(Sender: TObject);
  procedure TBalasOtomatisTimer(Sender:
TObject);
  procedure lvSMS1SelectItem(Sender:
TObject; Item: TListItem;
    Selected: Boolean);
  procedure lvSMS1ColumnClick(Sender:
TObject; Column: TListColumn);
  procedure DBGrid4CellClick(Column:
TColumn);
  procedure lvPBook1ColumnClick(Sender:
TObject; Column: TListColumn);
  procedure lvPBook1SelectItem(Sender:
TObject; Item: TListItem;
    Selected: Boolean);
  procedure DBGrid3CellClick(Column:
TColumn);
  procedure mpLatarNotify(Sender:
TObject);
  procedure txtPesan1Change(Sender:
TObject);
  procedure
txtPesanBroadcastChange(Sender: TObject);
  procedure Button4Click(Sender:
TObject);
  procedure Button1Click(Sender:
TObject);
  procedure Button2Click(Sender:
TObject);
  procedure Button3Click(Sender:
TObject);
  procedure Button6Click(Sender:
TObject);
  procedure Button5Click(Sender:
TObject);
  procedure Login2Click(Sender: TObject);
  procedure Keluar2Click(Sender:
TObject);
  procedure InfoSMS2Click(Sender:
TObject);
  procedure KirimSMS2Click(Sender:
TObject);
  procedure Phonebook2Click(Sender:
TObject);
  procedure Panggilan2Click(Sender:
TObject);
  procedure Broadcast2Click(Sender:
TObject);
  procedure UnregClick(Sender: TObject);
  procedure bKonekClick(Sender:
TObject);
  procedure About1Click(Sender: TObject);
  procedure Help2Click(Sender: TObject);
  procedure mLatarClick(Sender: TObject);
  procedure mButtonShowClick(Sender:
TObject);
  procedure mButtonKlikClick(Sender:
TObject);
  procedure bCekSMSClick(Sender:
TObject);
  procedure KirimEMAIL1Click(Sender:
TObject);
  procedure bLoginEmailClick(Sender:
TObject);
  procedure bSendEmailClick(Sender:
TObject);
  procedure bLogoutEmailClick(Sender:
TObject);
  procedure bKirimBroadcastClick(Sender:
TObject);
  procedure bClearBroadcastClick(Sender:
TObject);
  procedure bTutupPanelClick(Sender:
TObject);
  procedure bKirimClick(Sender: TObject);
  procedure bClearClick(Sender: TObject);
  procedure bSimpan1Click(Sender:
TObject);
  procedure bBaru1Click(Sender: TObject);
  procedure bBaca1Click(Sender: TObject);
  procedure bStopClick(Sender: TObject);
  procedure cbReport1Click(Sender:
TObject);
  procedure cbReport2Click(Sender:
TObject);
  procedure bAttachFileClick(Sender:
TObject);
  procedure UserAccount1Click(Sender:
TObject);
  procedure Data1Click(Sender: TObject);
  procedure Cascade1Click(Sender:
TObject);
  procedure
bInsertUserAccountClick(Sender: TObject);
  procedure
bEditUserAccountClick(Sender: TObject);
  procedure
bDeleteUserAccountClick(Sender:
TObject);
  procedure bOKUserAccountClick(Sender:
TObject);
  procedure
bCancelUserAccountClick(Sender:
TObject);
  procedure
bRefreshUserAccountClick(Sender:
TObject);
```

```
  procedure Browse1Click(Sender: TObject);
  procedure vDataUserOrderByClick(Sender: TObject);
  procedure bPrintViewDataUserClick(Sender: TObject);
  procedure vDataLoginOrderByClick(Sender: TObject);
  procedure vDataPelangganOrderByClick(Sender: TObject);
  procedure vDataPhonebookOrderByClick(Sender: TObject);
  procedure vDataInboxSMSOrderByClick(Sender: TObject);
  procedure vDataInboxEmailOrderByClick(Sender: TObject);
  procedure vDataSentOrderByClick(Sender: TObject);

 private
  { Private declarations }
 public
  procedure SetTerkoneksi;
  procedure getInfo(buffer: string);
  function GoKonek: Boolean;
  procedure FilterDisplay();
  procedure HapusItem(kode: string);
  procedure SetPC(kode: string);
  procedure SetJenis(kode, jenis: string);
  { Public declarations }
  procedure CekSMS;
  procedure BalasSMS;
  procedure ForwardSMS;
  procedure HapusSMS;
  procedure PindahSMS;
  procedure BacaSMS;
  procedure BacaPB;
  procedure SimpanSetting;
  procedure BacaSetting;
  procedure BukaDatabase;
  procedure TutupDatabase;
  procedure MasukkanKeInbox;
  procedure MasukkanKeSent(VCH_EMAIL_ADDRESS, DTM_EMAIL_DATE, VCH_TO_NAME, VCH_TO_EMAIL_ADDRESS, VCH_SUBJECT, VCH_BODY, VCH_STATUS, VCH_PHONE_NUMBER: String);
  procedure ViewSent(Status: String);
  procedure PilihSemua(Sender: TObject);
  procedure HapusSMSdariHP(Sender: TObject);
  procedure KirimSMSOtomatisCekInbox(Sender: TObject);
  procedure KirimEmailOtomatisCekInbox(txtToName, txtToEmailAddress, txtToSubject,
    txtToBody, txtUsername, txtPassword, txtFromName: String);
  procedure CekInboxReg(Sender: TObject);
  procedure CekInboxSMSEmail(Sender: TObject);
  procedure CekInboxEmailSMS(Sender: TObject);
  procedure CekInboxUnreg(Sender: TObject);

  function rubahKeNama(nomor: string): string;
  function rubahKeNomor(nama: string): string;
  function KirimSMS(Tujuan, Isi: string): boolean;
  function KirimSMSBroadcast(Tujuan, Isi: string): boolean;
  function SendGetData(Teks, Batas: String): String;
  function TrimAll(t: string): string;
  Function CreateTokens(divider: char; s: String): String;

  procedure Suara1(Sender: TObject);
  procedure Suara2(Sender: TObject);
  procedure SuaraLatarMulai(Sender: TObject; var S: String);
  procedure LihatHint(Sender: TObject);
  procedure setVisiblePanel(Panel: String; setVisible: Boolean);
  procedure SkinChange(Sender:TObject);
  procedure BackgroundChange(Sender:TObject);
 end;

Type
 tLokasiSMS = (LokasiPC, LokasiHP);
 tJenisSMS = (jBaru, jInbox, jDraft, jSent, jSemua);

const
 sOK = #13#10'OK';
 sERROR = #13#10'ERROR';
```

```
 arSumber: array[0..2] of String[2] = ('SM',
'ME', 'MT');
 arJenisSMS: array[0..3] of String[9] =
('Baru', 'Inbox', 'Draft', 'Terkirim');
 arPhoneInfo: array[1..10] of string[20] =
('battchg', 'signal', 'batterywarning',
'chargerconnected', 'service', 'sounder',
'message', 'call', 'roam', 'callsetup');
 SA=2500;

var
 F_Utama: TF_Utama;
 Path: String;
 SPos: integer = 0;
 SText: string = '
.:: ';
 SText2: String = '.: APLIKASI SMS
GATEWAY SISTEM PENGIRIMAN
EMAIL TO SMS DAN SMS TO EMAIL,
CREATED BY DHIKAREZA
MULAWARMAN, ITN MALANG ::.';
 NamaHariPanjang: Array [1..7] Of String
= ('Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis',
'Jum"at', 'Sabtu');
 NamaBulanPanjang: Array [1..12] Of
String = ('Januari', 'Februari', 'Maret', 'April',
'Mei', 'Juni', 'Juli', 'Agustus', 'September',
'Oktober', 'November', 'Desember');
 ArrayPass: Array [1..1] of String =
('access_pjk');

 //
 SMSC: String;
 PBCache, Item, List: TStrings;
 fSMS, fPB: file;
 LokasiDipilih: Set of tLokasiSMS;
 JenisSMS: tJenisSMS;
 CallInProgress: Boolean;
 LastSortedColumn: integer;
 Ascending: boolean;
 Konek: Boolean;
 Buffer: WideString;
 ReadyState: Boolean;
 BatasStr: String;
 NamaLama, PBID: String;
 PhoneInfo : array[1..20] of string;
 JumlahPhoneInfo : Integer;
 SedangBaca : Boolean;
 //

 MulaiPengecekan: Integer;
 JumlahItem: Integer;

 CekIsi, CekLokasi, CekNomor, CekJenis:
String;
 txtNomor, txtPesan: String;
 SND_SMS2EMAIL_TO_NAME,
SND_SMS2EMAIL_TO_EMAILADDRES
S, SND_SMS2EMAIL_TO_SUBJECT,
SND_SMS2EMAIL_TO_BODY,
 SND_SMS2EMAIL_FROM_NAME,
SND_SMS2EMAIL_FROM_EMAILADDR
ESS,
SND_SMS2EMAIL_FROM_EMAILPASS
WORD: String;
 CekReg: array[1..6] of String;
 CekSMS2Email: array[1..5] of String;
 CekEmail2SMS: String;
 EmailAttachment: String;
 CekNomorInbox, PasswordData: String;

CekProses1,CekProses2,CekProses3,CekPro
ses4: String;
 CekID, NoReg: Integer;
 Ini: TIniFile;
 NCEK: INTEGER;

SuaraLatar,SuaraButtonShow,SuaraButtonK
lik,WarnaLatar,GambarBackground,BalasOt
omatis,ReportSMS: String;

Cek_Nmr_Teregistrasi_INBOX,Cek_Nmr_
Teregistrasi_REGISTRASI: String;
 Play, Stop: String;
 UsernameGMail, PasswordGMail,
ToNameGMail, ToEmailGMail,
SubjectGMail, BodyGMail: String;
 SR,SR2: TSearchRec;
 TMnSkin,TMnBackground: TMenuItem;
 S1,S2: String;
 txtTokenSource, txtTokenResult: String;

implementation

uses U_Password, U_DataModule,
U_About;

{$R *.dfm}

function SortByColumn(Item1, Item2:
TListItem; Data: integer):
  integer; stdcall;
begin
 if Data = 0 then Result :=
AnsiCompareText(Item1.Caption,
Item2.Caption)
 else
 Result :=
AnsiCompareText(Item1.SubItems[Data-
1],Item2.SubItems[Data-1]);
```

```
 if not Ascending then Result := -Result;
end;

function    TF_Utama.TrimAll(t:    string):
string;
var
 s: string;
begin
 s := trim(t);
 s := copy(s, 2, length(s) - 2);
 result := s;
end;

function        GetNextLongSMSRefference:
string;
var
 mref: integer;
begin
 randomize;
 mref := 1 + random(252);
 with TRegistry.Create do
  try
   if
OpenKey('\Software\SMSGateway.com\SM
U', true) then
    try
     if ValueExists('MessageRef') then
      mref := ReadInteger('MessageRef');
     if (mref < 0) or (mref >= 255) then
mref := 0;
     inc(mref);
     WriteInteger('MessageRef',mref);
    finally
     CloseKey;
    end;
  finally
   Free;
  end;
 Result := IntToHex(mref, 2);
end;

function    Tf_Utama.rubahKeNama(nomor:
string): string;
var
 s, sn: string;
begin
 if nomor[1] = '+' then sn := '%' +
copy(nomor, 4, length(nomor)) else
 if (copy(nomor, 1, 2) = '62') and
(length(nomor) > 4) then sn := '%' +
copy(nomor, 3, length(nomor)) else
 if (nomor[1] = '0') then sn := '%' +
copy(nomor, 2, length(nomor)) else sn :=
nomor;
 s := PBCache.Values[nomor];
 if s <> '' then
 begin
  Result := s;
  Exit;
 end;
 while        ADOQuery1.Active        do
ADOQuery1.Close;
 ADOQuery1.SQL.Text        :=        'select
VCH_FULLNAME from TBL_Phonebook
where VCH_PHONE_NUMBER like ' +
QuotedStr(sn);
 ADOQuery1.Open;
 if not ADOQuery1.Eof then
 begin
  Result                          :=
ADOQuery1.FieldValues['VCH_FULLNA
ME'];
  PBCache.Values[nomor]           :=
ADOQuery1.FieldValues['VCH_FULLNA
ME'];
 end else
 Result := nomor;
end;

function    Tf_Utama.rubahKeNomor(nama:
string): string;
var
 i : integer;
begin
 i := PBCache.IndexOf(nama);
 if i >= 0 then
 begin
  Result := PBCache.Strings[i];
  Exit;
 end;
 while        ADOQuery1.Active        do
ADOQuery1.Close;
 ADOQuery1.SQL.Text        :=        'select
VCH_PHONE_NUMBER          from
TBL_Phonebook where VCH_FULLNAME
like ' + QuotedStr(nama);
 ADOQuery1.Open;
 if not ADOQuery1.Eof then Result :=
ADOQuery1.FieldValues['VCH_PHONE_N
UMBER'] else Result := nama;
end;

function Tf_Utama.KirimSMS(Tujuan, Isi:
string): boolean;
var
 PDU, udhi, smstotal, smsref, tempisi, s:
string;
 sms: tSMS;
 i, p: integer;
begin
```

```
 if not konek then
 begin
  Result := false;
  exit;
 end;
sms := TSMS.Create;
 sms.Number := tujuan;
 sms.StatusRequest := cbReport1.Checked;
 sms.dcs := -1;
 Result := true;
 if length(isi) <= 160 then
 begin
  sms.Text := isi;
  sms.UDHI := '';
  pdu := sms.PDU;
  s := SendGetData('AT+CMGS=' +
inttostr(sms.tplength) + #13, '>');
  s := SendGetData(PDU + #$1A, sOK);
  Result := pos(sOK, s) > 0;
 end else
 begin
  p := 153;
  smstotal := IntToHex((length(isi) div p) +
1, 2);
  smsRef := GetNextLongSMSRefference;
  udhi := '050003' + smsRef + smstotal;
  for i := 1 to StrToInt('$' + smstotal) do
  begin
   tempisi := Copy(isi, 1, p);
   Delete(isi, 1, p);
   sms.Text := tempisi;
   sms.UDHI := udhi + IntToHex(i, 2);
   pdu := sms.PDU;
   s := SendGetData('AT+CMGS=' +
inttostr(sms.tplength) + #13, '>');
   s := SendGetData(PDU + #$1A, sOK);
   Result := pos(sOK, s) > 0;
  end;
 end;
 sms.Free;
end;

function
Tf_Utama.KirimSMSBroadcast(Tujuan, Isi:
string): boolean;
var
 PDU, udhi, smstotal, smsref, tempisi, s:
string;
 sms: tSMS;
 i, p: integer;
begin
 if not konek then
 begin
  Result := false;
  exit;
 end;
 sms := TSMS.Create;
 sms.Number := tujuan;
 sms.StatusRequest := cbReport2.Checked;
 sms.dcs := -1;
 Result := true;
 if length(isi) <= 160 then
 begin
  sms.Text := isi;
  sms.UDHI := '';
  pdu := sms.PDU;
  s := SendGetData('AT+CMGS=' +
inttostr(sms.tplength) + #13, '>');
  s := SendGetData(PDU + #$1A, sOK);
  Result := pos(sOK, s) > 0;
 end else
 begin
  p := 153;
  smstotal := IntToHex((length(isi) div p) +
1, 2);
  smsRef := GetNextLongSMSRefference;
  udhi := '050003' + smsRef + smstotal;
  for i := 1 to StrToInt('$' + smstotal) do
  begin
   tempisi := Copy(isi, 1, p);
   Delete(isi, 1, p);
   sms.Text := tempisi;
   sms.UDHI := udhi + IntToHex(i, 2);
   pdu := sms.PDU;
   s := SendGetData('AT+CMGS=' +
inttostr(sms.tplength) + #13, '>');
   s := SendGetData(PDU + #$1A, sOK);
   Result := pos(sOK, s) > 0;
  end;
 end;
 sms.Free;
end;

function Tf_Utama.SendGetData;
var
 waktu: TDateTime;
begin
 ReadyState := False;
 BatasStr := Batas;
 Buffer := '';
 waktu := now;
 XComm1.SendString(Teks);
 while (Not ReadyState) and
(SecondsBetween(waktu, Now) < 10)
 do Application.ProcessMessages;
 Result := Buffer;
end;
```

```
function Tf_Utama.GoKonek;
begin
  Konek := false;
  If Not f_Utama.XComm1.Opened then
f_Utama.XComm1.OpenDevice;
  if XComm1.SendString('ATE1'#13) and
(XComm1.WaitForString(['OK'], 2000) <> -
1) then
  Konek := True;
  Result := Konek;
end;

Function TF_Utama.CreateTokens(divider:
char; s: String): String;
var
  it: Integer;
  ps: String;
begin
  SetLength(result,0);
  it:=1;
  ps:='';
  while it<=Length(s) do
  begin
    if s[it]<>divider then
      ps:=ps+s[it]
    else begin
      if Length(ps)>0 then
      begin
        SetLength(result, Length(result)+1);
        txtTokenResult:=ps;
        ps:='';
      end;
    end;
    Inc(it);
  end;
  Setlength(result, Length(result)+1);
end;

procedure TF_Utama.SetTerkoneksi;
begin
  If not Konek then
  begin
    Exit;
  end;
  getInfo(SendGetData('AT+CGMI'#13,
sOK));
  getInfo(SendGetData('AT+GMM'#13,
sOK));
  getInfo(SendGetData('AT+CGSN'#13,
sOK));
  getInfo(SendGetData('AT+COPS?'#13,
sOK));
  getInfo(SendGetData('AT+CSCA?'#13,
sOK));
  getInfo(SendGetData('AT+CIND=?'#13,
sOK));
  Timer1.Enabled := Konek;
  bKonek.Caption := 'Disconnect';
  cbPort.Enabled := False;
  cbBaud.Enabled := False;
  cbFC.Enabled := False;
  SedangBaca := False;
  SimpanSetting;
end;

procedure TF_Utama.getInfo;
var
  c, s: string;
  p, n, i: integer;
begin
  List.Text := buffer;
  s := List.Strings[1];
  if Pos('AT+CGMI', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('AT+CGMI',
Buffer) + 2, length(Buffer));
    Item.Text := c;
    bMerk.Caption:=Item.Strings[2];
  end;
  if Pos('AT+GMM', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('AT+GMM',
Buffer) + 2, length(Buffer));
    Item.Text := c;
    bModem.Caption:=Item.Strings[2];
  end;
  if Pos('AT+CGSN', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('AT+CGSN',
Buffer) + 2, length(Buffer));
    Item.Text := c;
    bIMEI.Caption:=Item.Strings[2];
  end;
  if Pos('+COPS:', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('+COPS:', Buffer) +
7, length(Buffer));
    c := AnsiReplaceStr(c, ',', #13);
    Item.Text := c;
    bOperator.Caption:=Item.Strings[2];
  end;
  if Pos('+CSCA:', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('+CSCA:', Buffer) +
7, length(Buffer));
    c := AnsiReplaceStr(c, ',', #13);
    Item.Text := c;
    SMSC := TrimAll(Item.Strings[0]);
    bService.Caption:=SMSC;
```

```
  end;
  if Pos('+CIND:', Buffer) > 0 then
  begin
    c := copy(Buffer, pos('+CIND:', Buffer) +
7, length(Buffer));
    i := 0;
    p := pos('(''', c);
    while p > 0 do
    begin
      n := pos(''',', c);
      s := copy(c, p + 2, n - p - 2);
      PhoneInfo[i] := s;
      Inc(i);
      Delete(c, p, n + 2);
      p := pos('(''', c);
    end;
    JumlahPhoneInfo := i;
  end;
end;

procedure TF_Utama.FilterDisplay();
var
  i: integer;
  a, l: TListItem;
begin
  lvSMS1.Items.Clear;
  for i := 0 to ListSMS.Items.Count - 1 do
  begin
    l := ListSMS.Items.Item[i];
    if ((l.SubItems[2] = 'PC') and (LokasiPC
in LokasiDipilih)) or
      ((l.SubItems[2] = 'Phone') and
(LokasiHP in LokasiDipilih)) then
      if (l.SubItems[3] =
arJenisSMS[Ord(JenisSMS)]) or (JenisSMS
= jSemua) then
      begin
        a := lvSMS1.Items.Add;
        a.Caption := rubahKeNama(l.Caption);
        a.SubItems := l.SubItems;
      end;
  end;
  lbJumlah.Caption := 'SMS: ' +
IntToStr(lvSMS1.Items.Count);
  JumlahItem:=lvSMS1.Items.Count;
end;

procedure TF_Utama.HapusItem;
var
  i: integer;
begin
  for i := ListSMS.Items.Count - 1 downto 0
do
  if (ListSMS.Items.Item[i].SubItems[5] =
kode) and
(ListSMS.Items.Item[i].SubItems[2] =
'Phone') then
  begin
    ListSMS.Items.Item[i].Delete;
    exit;
  end;
end;

procedure TF_Utama.SetPC;
var
  i: integer;
begin
  for i := ListSMS.Items.Count - 1 downto 0
do
  if (ListSMS.Items.Item[i].SubItems[5] =
kode) and
(ListSMS.Items.Item[i].SubItems[2] =
'Phone') then
  begin
    ListSMS.Items.Item[i].SubItems[2] :=
'PC';
    if ListSMS.Items.Item[i].SubItems[3] =
'Baru' then
ListSMS.Items.Item[i].SubItems[3] :=
'Inbox';
    exit;
  end;
end;

procedure TF_Utama.SetJenis;
var
  i: integer;
begin
  for i := ListSMS.Items.Count - 1 downto 0
do
  if (ListSMS.Items.Item[i].SubItems[5] =
kode) and
(ListSMS.Items.Item[i].SubItems[2] =
'Phone') then
  begin
    ListSMS.Items.Item[i].SubItems[3] :=
Jenis;
    exit;
  end;
end;

procedure TF_Utama.CekSMS;
var
  nomer, tgl, s : string;
  i,j: integer;
  a, l: TListItem;
  sms : tSMS;
begin
  if not Konek then exit;
```

```
 s    :=    sendGetData('AT+CSMS=0'#13,
sOK);
 if Pos(sOK, s) = 0 then
 begin
  exit;
 end;
 s    :=    sendGetData('AT+CPMS="    +
arSumber[cbSumber1.ItemIndex]    +'"'#13,
sOK);
 if Pos(sOK, s) = 0 then
 begin
  exit;
 end;
 s    :=    sendGetData('AT+CMGF=0'#13,
sOK);
 if Pos(sOK, s) = 0 then
 begin
  exit;
 end;
 List.Clear;
 sms := TSMS.Create;
 for j := 1 downto 0 do
 begin
  List.Text  :=  SendGetData('AT+CMGL='
+ IntToStr(j) +#13, sOK);
  i := 0;
  while i < List.Count-2 do
  begin
   s := List.Strings[i];
   if copy(s, 1, 7) = '+CMGL: ' then
   begin
    nomer := copy(s, 8, pos(',', s) - 8);
    inc(i);
    s := List.Strings[i];
    sms.PDU := s;
    l := ListSMS.Items.Add;
    l.Caption := sms.number;
    if sms.TimeStamp > 0 then tgl :=
DateTimeToStr(sms.TimeStamp)
    else tgl := '-';
    l.SubItems.Add(tgl);
    l.SubItems.Add(sms.Text);
    l.SubItems.Add('Phone');
    l.SubItems.Add(arJenisSMS[j]);
    l.SubItems.Add(nomer);
    l.SubItems.Add(nomer);
    a := lvSMS1.Items.Add;
    a.Caption := rubahKeNama(l.Caption);
    a.SubItems := l.SubItems;
   end;
   inc(i);
  end;
 end;
 sms.Free;
end;

procedure TF_Utama.BalasSMS;
var
 l: TListItem;
begin
 l := lvSMS1.Selected;
 if l <> nil then
 begin
  txtNomor1.Text                    :=
rubahKeNomor(l.Caption);
  txtPesan1.Text := l.SubItems[1];
  txtPesan1.SetFocus;
 end;
end;

procedure TF_Utama.ForwardSMS;
var
 l: TListItem;
begin
 l := lvSMS1.Selected;
 if l <> nil then
 begin
  txtNomor1.Text := '';
  txtPesan1.Text := l.SubItems[1];
  txtNomor1.SetFocus;
 end;
end;

procedure TF_Utama.HapusSMS;
var
 i: integer;
 l: TListItem;
 s, n: string;
begin
 Timer1.Enabled := False;
 for i := lvSMS1.Items.Count - 1 downto 0
do
 begin
  l := lvSMS1.Items.Item[i];
  if (l <> nil) and (l.Selected) then
  begin
   If l.SubItems[2] = 'PC' Then
   begin
    ADOConnection1.Execute('delete from
TBL_SMS_INBOX    where    id='    +
l.SubItems[5]);
    HapusItem(l.SubItems[5]);
    l.Delete;
   end else
   if konek then
   begin
    n := l.SubItems[4];
    s := SendGetData('AT+CMGD=' + n +
#13, sOK);
    if Pos(sOK, s) > 0 then
    begin
```

```
      HapusItem(l.SubItems[5]);
      l.Delete;
     end;
    end;
   end;
  end;
  lbJumlah.Caption := 'SMS: ' +
IntToStr(lvSMS1.Items.Count);
  Timer1.Enabled := True;
end;

procedure TF_Utama.PindahSMS;
var
  i: integer;
  l: TListItem;
  s, n: string;
begin
  if not Konek then
  begin
    exit;
  end;
  Timer1.Enabled := False;
  prgbar2.Max := lvSMS1.Items.Count - 1;
  prgbar2.Min := 0;
  prgbar2.Position := 0;
  prgbar2.Visible := True;
  for i := lvSMS1.Items.Count - 1 downto 0
do
  begin
    l := lvSMS1.Items.Item[i];
    if (l <> nil) and (l.Selected) then
    begin
      if l.SubItems[2] = 'Phone' then
      begin
        n := l.SubItems[4];
        s := SendGetData('AT+CMGD=' + n +
#13, sOK);
        if Pos(sOK, s) > 0 then
        begin
          l.SubItems[2] := 'PC';
          SetPC(l.SubItems[5]);
          ADOConnection1.Execute('insert into
TBL_SMS_INBOX(VCH_PHONE_NUMB
ER,VCH_DATE,VCH_ISI_SMS,VCH_JEN
IS,VCH_PROCESS) values('+
QuotedStr(rubahKeNomor(copy(l.Caption,
1, 20))) + ',' +
QuotedStr(copy(l.SubItems[0], 1, 20)) + ','
+ QuotedStr(copy(l.SubItems[1], 1, 160)) +
',' + QuotedStr(copy(l.SubItems[3], 1, 10))
+ ',' + QuotedStr('False')+')');
ADOQuery1.SQL.Text := 'select max(id) as
baru from TBL_SMS_INBOX';
ADOQuery1.Open;
          l.SubItems[5] :=
ADOQuery1.FieldValues['baru'];
        end;
      end;
    end;
    prgbar2.Position := (lvSMS1.Items.Count
- i) - 1;
  end;
  prgbar2.Visible := False;
  FilterDisplay;
  Timer1.Enabled := True;
end;

procedure TF_Utama.BacaSMS;
var
  l: TListItem;
begin
  while ADOQuery1.Active do
ADOQuery1.Close;
  ADOQuery1.SQL.Text := 'select * from
TBL_SMS_INBOX';
  ADOQuery1.Open;
  While Not ADOQuery1.Eof do
  begin
    l := ListSMS.Items.Add;
    l.Caption :=
ADOQuery1.FieldValues['VCH_PHONE_N
UMBER'];
l.SubItems.Add(ADOQuery1.FieldValues['
VCH_DATE']);
l.SubItems.Add(ADOQuery1.FieldValues['
VCH_ISI_SMS']);
    l.SubItems.Add('PC');
l.SubItems.Add(ADOQuery1.FieldValues['
VCH_JENIS']);
    l.SubItems.Add('0');
l.SubItems.Add(ADOQuery1.FieldValues['I
D']);
    ADOQuery1.Next;
  end;
end;

procedure TF_Utama.BacaPB;
var
  l: TListItem;
begin
  while ADOQuery1.Active do
ADOQuery1.Close;
  ADOQuery1.SQL.Text := 'select * from
TBL_Phonebook';
  ADOQuery1.Open;
  While Not ADOQuery1.Eof do
  begin
    l := lvPBook1.Items.Add;
```

```
  l.Caption                                   :=
ADOQuery1.FieldValues['VCH_FULLNA
ME'];
l.SubItems.Add(ADOQuery1.FieldValues['
VCH_PHONE_NUMBER']);
l.SubItems.Add(ADOQuery1.FieldValues['I
D']);
  ADOQuery1.Next;
 end;
end;

procedure TF_Utama.SimpanSetting;
var
 Reg : TRegistry;
begin
 Reg := TRegistry.Create;
 try
  if
Reg.OpenKey('\Software\SMSGateway.com
\SMU', True) then
   begin
    Reg.WriteInteger('BaudRate',
cbBaud.ItemIndex);
    Reg.WriteInteger('Port',
cbPort.ItemIndex);
    Reg.WriteInteger('Flowcontrol',
cbFC.ItemIndex);
    Reg.CloseKey;
   end;
 finally
   Reg.Free;
   inherited;
 end;
end;

procedure TF_Utama.BacaSetting;
var
 Reg : TRegistry;
begin
 Reg := TRegistry.Create;
 try
  if
Reg.OpenKey('\Software\SMSGateway.com
\SMU', True) then
   begin
    if Reg.ValueExists('BaudRate') then
     cbBaud.ItemIndex                         :=
Reg.ReadInteger('BaudRate')
    else
     cbBaud.ItemIndex := 0;
    if Reg.ValueExists('Port') then
     cbPort.ItemIndex                         :=
Reg.ReadInteger('Port')
    else
     cbPort.ItemIndex := 0;
    if Reg.ValueExists('Flowcontrol') then
     cbFC.ItemIndex                           :=
Reg.ReadInteger('Flowcontrol')
    else
     cbFC.ItemIndex := 0;
    Reg.CloseKey;
   end;
 finally
   Reg.Free;
   inherited;
 end;
end;

procedure Tf_Utama.BukaDatabase;
var
 NamaFileDB: String;
begin
NamaFileDB:=ExtractFilePath(Application.
ExeName)+'Data\datasms.mdb';
 if not FileExists(NamaFileDB) then
 begin
  ShowMessage('File database tidak
ditemukan: ' + NamaFileDB);
  exit;
 end;
 TutupDatabase;
 ADOConnection1.ConnectionString :=
'Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=' + NamaFileDB + ';Persist Security
Info=False';
 ADOConnection1.Open;
end;

procedure Tf_Utama.TutupDatabase;
begin
 while ADOConnection1.Connected do
 ADOConnection1.Close;
end;

procedure TF_Utama.FormCreate(Sender:
TObject);
var
 i : integer;
 A,B,A2,B2: Integer;
 S,SS: String;
begin
Path:=ExtractFilePath(Application.ExeNam
e);
 Application.OnHint := LihatHint;
 bMerk.Caption:='';
bModem.Caption:='';
 bIMEI.Caption:='';
bOperator.Caption:='';
 bService.Caption:='';
```

```
swf2.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim2.swf';
 Ini                                    :=
TIniFile.Create(ExtractFilePath(Application.
ExeName) + '\DATA\NO_REG.INI');
 NoReg      :=      Ini.ReadInteger('REG',
'NOREG', 0);
 Play:='Play';
 Stop:='Stop';
 SuaraLatar  :=  Ini.ReadString('SUARA',
'LATAR', '');
 if SuaraLatar='ON'  then
 begin
   mLatar.Checked:=True;
   SuaraLatarMulai(Sender    as    TObject,
Play);
 end else
 if SuaraLatar='OFF' then
 begin
   mLatar.Checked:=False;
   SuaraLatarMulai(Sender    as    TObject,
Stop);
 end;
 SuaraButtonShow                        :=
Ini.ReadString('SUARA',
'BUTTONSHOW', '');
 if SuaraButtonShow='ON'  then
 begin
   mButtonShow.Checked:=True;
 end else
 if SuaraButtonShow='OFF' then
 begin
   mButtonShow.Checked:=False;
 end;
 SuaraButtonKlik                        :=
Ini.ReadString('SUARA',  'BUTTONKLIK',
'');
 if SuaraButtonKlik='ON'  then
 begin
   mButtonKlik.Checked:=True;
 end else
 if SuaraButtonKlik='OFF' then
 begin
   mButtonKlik.Checked:=False;
 end;
 ReportSMS                              :=
Ini.ReadString('SMS','REPORT','');
 if ReportSMS='1' then
 begin
   cbReport1.Checked:=True;
   cbReport2.Checked:=True;
 end else
 begin
   cbReport1.Checked:=False;
   cbReport2.Checked:=False;
 end;
 UsernameGMail                          :=
Ini.ReadString('EMAIL','USERNAME','');
 if        UsernameGMail<>''        then
eUsername.Text:=UsernameGMail;
 ToNameGMail                            :=
Ini.ReadString('EMAIL','TO_NAME','');
 if        ToNameGMail<>''        then
eToName.Text:=ToNameGMail;
 ToEmailGMail                           :=
Ini.ReadString('EMAIL','TO_EMAIL','');
 if        ToEmailGMail<>''        then
eToEmail.Text:=ToEmailGMail;
 SubjectGMail                           :=
Ini.ReadString('EMAIL','SUBJECT','');
 if        SubjectGMail<>''        then
eSubject.Text:=SubjectGMail;
 BodyGMail                              :=
Ini.ReadString('EMAIL','BODY','');
 if        BodyGMail<>''        then
eBody.Text:=BodyGMail;
S1:=ExtractFilePath(Application.ExeName)
+'Skins\';
 sSkinManager1.SkinDirectory:=S1;
 WarnaLatar  :=  Ini.ReadString('WARNA',
'LATAR', '');
 sSkinManager1.SkinName := WarnaLatar;
 A:=0;
 repeat
  Inc(A);
  if              A=1              then
B:=FindFirst(S1+'*.asz',faAnyFile,SR) else
   B:=FindNext(SR);
  if  (B=0)  and  (SR.Name<>'.')  and
(SR.Name<>'..') then
  begin
   S:=Copy(SR.Name,1,Length(SR.Name)-
4);
   TMnSkin:=TMenuItem.Create(Skins1);
   TMnSkin.Caption:=S;
   TMnSkin.Tag:=A;
   Skins1.Add(TMnSkin);
   TMnSkin.OnClick:=SkinChange;
  end;
 until B<>0;
 FindClose(SR);
S2:=ExtractFilePath(Application.ExeName)
+'Gambar\Background\';
 GambarBackground                       :=
Ini.ReadString('BACKGROUND',
'LATAR',                                ');
Image1.Picture.LoadFromFile(S2+GambarB
ackground+'.jpg');
 A2:=0;
```

```
 repeat
  Inc(A2);
  if A2=1 then
B2:=FindFirst(S2+'*.jpg',faAnyFile,SR2)
else
   B2:=FindNext(SR2);
  if (B2=0) and (SR2.Name<>'.') and
(SR2.Name<>'..') then
  begin
SS:=Copy(SR2.Name,1,Length(SR2.Name)-
4);
TMnBackground:=TMenuItem.Create(Back
ground1);
    TMnBackground.Caption:=SS;
    TMnBackground.Tag:=A2;
    Background1.Add(TMnBackground);
TMnBackground.OnClick:=BackgroundCha
nge;
  end;
 until B2<>0;
 FindClose(SR2);
    Item := TStringList.Create;
    List := TStringList.Create;
    PBCache := TStringList.Create;
    Konek := False;
    Buffer := '';
    Timer1.Enabled := False;
    cbPort.Items.Clear;
    for i := 1 to 30 do
      cbPort.Items.Add('COM' + IntToStr(i));
    cbPort.ItemIndex := 0;
    cbBaud.Items.Clear;
    cbBaud.Items.Add('9600');
    cbBaud.Items.Add('19200');
    cbBaud.Items.Add('57600');
    cbBaud.Items.Add('115200');
    cbBaud.Items.Add('128000');
    cbBaud.Items.Add('256000');
    cbBaud.ItemIndex := 0;
    cbFC.Clear;
    cbFC.Items.Add('None');
    cbFC.Items.Add('RTS-CTS');
    cbFC.Items.Add('DTR-DSR');
    cbFC.Items.Add('Software');
    cbFC.ItemIndex := 0;
    cbSumber1.ItemIndex := 0;
    cbJenisPB1.ItemIndex := 0;
    LokasiDipilih := [LokasiPC, LokasiHP];
    JenisSMS := jSemua;
    CallInProgress := False;
    XComm1.BaudRate := brCustom;
    XComm1.FlowControl := fcNone;
    BukaDatabase;
    BacaPB;
    ListSMS.Items.Clear;
    FilterDisplay;
    BacaSetting;
    LastSortedColumn := -1;
    Ascending := True;
    PBID := '';
end;

procedure TF_Utama.LihatHint(Sender:
TObject);
begin
  SB1.Panels[2].Text:='
'+GetLongHint(Application.Hint);
end;

procedure TF_Utama.Timer1Timer(Sender:
TObject);
var
  s, c: string;
  p : integer;
function getPhoneInfoIndex(info: string):
integer;
var
  i, r : integer;
begin
  r := -1;
  for i := 0 to JumlahPhoneInfo - 1 do
     if PhoneInfo[i] = info then r := i;
  Result := r;
end;
begin
  if not Konek then exit;
    if SedangBaca then exit;
    SedangBaca := True;
    List.Text :=
SendGetData('AT+CIND?'+#13, sOk);
    s := List.Strings[2];
    c := copy(s, pos(':', s) + 2, length(s));
    c := AnsiReplaceStr(c, ',', #13#10);
    Item.Text := c;
    if c = '' then exit;
    if Item.Count < 2 then exit;
    p := getPhoneInfoIndex('signal');
    if p <> - 1 then
      pbSinyal.Position :=
StrToInt(Item.Strings[p]);
    p := getPhoneInfoIndex('battchg');
    if p <> - 1 then
      pbBattery.Position :=
StrToInt(Item.Strings[p]);
    p := getPhoneInfoIndex('message');
    if p <> - 1 then
      If Item.Strings[p] = '1' Then begin
           indSMSBaru.Color := clRed;
      end
      else indSMSBaru.Color := clSkyBlue;
```

```
  p := getPhoneInfoIndex('call');
  if p <> - 1 then
    If Item.Strings[p] = '1' Then
indPanggilan.Color := clRed
    else indPanggilan.Color := clSkyBlue;
  p :=
getPhoneInfoIndex('chargerconnected');
    if p <> - 1 then
    If Item.Strings[p] = '1' Then
indCharged.Color := clRed
        else indCharged.Color :=
clSkyBlue;
  SedangBaca := False;
end;

procedure TF_Utama.Timer2Timer(Sender:
TObject);
begin
 SB1.Panels[0].Text:='
'+FormatDateTime('dddd, d-mm-yyyy /
hh:nn:ss WIB',Now);
end;

procedure TF_Utama.FormActivate(Sender:
TObject);
var
 K: Integer;
begin
 For K:=1 To 7 Do LongDayNames[K] :=
NamaHariPanjang[K];
 For K:=1 To 12 Do LongMonthNames[K]
:= NamaBulanPanjang[K];
end;

procedure TF_Utama.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
 if XComm1.Opened then
 while XComm1.Opened do
XComm1.CloseDevice;
end;

procedure
TF_Utama.XComm1Data(Sender: TObject;
const Received: Cardinal);
var
 data : string;
begin
 XComm1.ReadString(data);
 Buffer := Buffer + data;
 if (Not ReadyState) and (Pos(BatasStr,
Buffer) > 0) then
 begin
  ReadyState := True;
 end;
end;

procedure TF_Utama.cbAutoClick(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 case (TControl(Sender).Tag) of
 10: begin
     if cbAuto.Checked then
     begin
       TBalasOtomatis.Enabled:=False;
cbAuto.Checked:=False;
       SB1.Panels[3].Text:='';
Ini.WriteString('OTOMATISASI','OTOMA
TIS','0');
     end else
     begin
       TBalasOtomatis.Enabled:=True;
MulaiPengecekan:=0;
       cbAuto.Checked:=True;
Ini.WriteString('OTOMATISASI','OTOMA
TIS','1');
     end;
    end;
 11: begin
     if cbAuto.Checked then
     begin
       TBalasOtomatis.Enabled:=True;
MulaiPengecekan:=0;
Ini.WriteString('OTOMATISASI','OTOMA
TIS','1');
     end else
     begin
       TBalasOtomatis.Enabled:=False;
SB1.Panels[3].Text:='';
Ini.WriteString('OTOMATISASI','OTOMA
TIS','0');
     end;
    end;
 end;
end;

procedure TF_Utama.MasukkanKeInbox;
begin
DataModule1.QMasukkanKeInbox.SQL.Cle
ar;
DataModule1.QMasukkanKeInbox.SQL.Ad
d('INSERT INTO
TBL_SMS_INBOX(VCH_PHONE_NUMB
ER,VCH_DATE,VCH_ISI_SMS,VCH_JEN
IS,VCH_PROCESS) VALUES('
+QuotedStr(CekNomor)+','
+QuotedStr(DateTimeToStr(Now)) + ','
+QuotedStr(CekIsi)+','
+QuotedStr(CekJenis)+','
```

```
+QuotedStr('False')+')');
DataModule1.QMasukkanKeInbox.ExecSQ
L;
end;

procedure
TF_Utama.MasukkanKeSent(VCH_EMAIL
_ADDRESS,          DTM_EMAIL_DATE,
VCH_TO_NAME,
VCH_TO_EMAIL_ADDRESS,
VCH_SUBJECT,          VCH_BODY,
VCH_STATUS, VCH_PHONE_NUMBER:
String);
begin
DataModule1.QMasukkanKeSent.SQL.Clea
r;
DataModule1.QMasukkanKeSent.SQL.Add(
'INSERT          INTO          TBL_SENT
(VCH_EMAIL_ADDRESS,DTM_EMAIL_
DATE,VCH_TO_NAME,VCH_TO_EMAI
L_ADDRESS,VCH_SUBJECT,VCH_BOD
Y,VCH_STATUS,VCH_PHONE_NUMBE
R)                    VALUES('
+QuotedStr(VCH_EMAIL_ADDRESS)+','
+QuotedStr(DTM_EMAIL_DATE)+','
+QuotedStr(VCH_TO_NAME)+','
+QuotedStr(VCH_TO_EMAIL_ADDRESS)
+','          +QuotedStr(VCH_SUBJECT)+','
+QuotedStr(VCH_BODY)+','
+QuotedStr(VCH_STATUS)+','
+QuotedStr(VCH_PHONE_NUMBER)+')');
DataModule1.QMasukkanKeSent.ExecSQL;
end;

procedure        TF_Utama.ViewSent(Status:
String);
begin
 DataModule1.QViewSent.SQL.Clear;
 DataModule1.QViewSent.SQL.Add('select
* from tbl_sent where vch_status =
'+QuotedStr(Status));
 DataModule1.QViewSent.ExecSQL;
 DataModule1.QViewSent.Open;
end;

procedure    TF_Utama.PilihSemua(Sender:
TObject);
var
 i : integer;
begin
 for i:=0 to lvSMS1.Items.Count - 1 do
lvSMS1.Items[i].Selected := True;
end;

procedure
TF_Utama.HapusSMSdariHP(Sender:
TObject);
var
 i: integer;
 l: TListItem;
 s, n: string;
begin
 for i := lvSMS1.Items.Count - 1 downto 0
do
 begin
  l := lvSMS1.Items.Item[i];
  if (l <> nil) and (l.Selected) then
  begin
   if l.SubItems[2] = 'Phone' then
   begin
    n := l.SubItems[4];
    s := SendGetData('AT+CMGD=' + n +
#13, sOK);
   end;
  end;
 end;
end;

procedure
TF_Utama.TBalasOtomatisTimer(Sender:
TObject);
var
 I: Integer;
 L: TListItem;
begin
 MulaiPengecekan:=MulaiPengecekan+1;
SB1.Panels[3].Text:=IntToStr(MulaiPengec
ekan);
 if MulaiPengecekan=10 then
 begin
  cbSumber1.ItemIndex:=0;
  BCekSMSClick(Sender as TObject);
  for I:=0 to JumlahItem-1 do
  begin
   L          :=          lvSMS1.Items.Item[I];
CekNomor:=Uppercase(Copy(rubahKeNom
or(L.Caption),1,20));
CekIsi:=Copy(lvSMS1.Items[I].SubItems[1]
,1,160);
CekJenis:=Uppercase(Copy(lvSMS1.Items[I
].SubItems[3],1,8));
   MasukkanKeInbox;
  end;
  PilihSemua(Sender as TObject);
  HapusSMSdariHP(Sender as TObject);
 end;
 if          MulaiPengecekan=20          then
CekInboxReg(Sender as TObject);
```

```
 if MulaiPengecekan=50 then
CekInboxSMSEmail(Sender as TObject);
 if MulaiPengecekan=80 then
CekInboxEmailSMS(Sender as TObject);
 if MulaiPengecekan=110 then
CekInboxUnreg(Sender as TObject);
 if MulaiPengecekan>=140 then
MulaiPengecekan:=0;
end;

procedure
TF_Utama.lvSMS1SelectItem(Sender:
TObject; Item: TListItem;
 Selected: Boolean);
begin
 lblSMS1.Text:=Item.SubItems[1];
end;

procedure
TF_Utama.lvSMS1ColumnClick(Sender:
TObject; Column: TListColumn);
begin
 if Column.Index = LastSortedColumn then
Ascending := not Ascending
 else
 begin
  LastSortedColumn := Column.Index;
  Ascending := True;
 end;
TListView(Sender).CustomSort(@SortByC
olumn, Column.Index);
end;

procedure
TF_Utama.DBGrid4CellClick(Column:
TColumn);
begin
 txtNomor1.Text :=
DBGrid4.SelectedField.AsString;
end;

procedure
TF_Utama.lvPBook1ColumnClick(Sender:
TObject;
 Column: TListColumn);
begin
 if Column.Index = LastSortedColumn then
Ascending := not Ascending
 else
 begin
  LastSortedColumn := Column.Index;
  Ascending := True;
 end;
TListView(Sender).CustomSort(@SortByC
olumn, Column.Index);
end;

procedure
TF_Utama.lvPBook1SelectItem(Sender:
TObject; Item: TListItem;
 Selected: Boolean);
begin
 txtNamaPB1.Text := Item.Caption;
 txtNomorPB1.Text := Item.SubItems[0];
 NamaLama := Item.Caption;
 PBID := Item.SubItems[1];
end;

procedure
TF_Utama.DBGrid3CellClick(Column:
TColumn);
begin
 txtNomorCall1.Text :=
DBGrid3.SelectedField.AsString;
end;

procedure TF_Utama.Suara1(Sender:
TObject);
begin
mp1.FileName:=Path+'\Suara\Tampilkan.wa
v';
 mp1.Open;
 if mButtonshow.Checked=True then
mp1.Play else
 if mButtonshow.Checked=False then
mp1.Stop;
end;

procedure TF_Utama.Suara2(Sender:
TObject);
begin
 mp1.FileName:=Path+'\Suara\Klik.wav';
 mp1.Open;
 if mButtonklik.Checked=True then
mp1.Play else
 if mButtonklik.Checked=False then
mp1.Stop;
end;

procedure
TF_Utama.SuaraLatarMulai(Sender:
TObject; var S: String);
begin
mpLatar.FileName:=Path+'\Suara\Latar.mid'
;
 mpLatar.Open;
 if S='Play' then mpLatar.Play
 else mpLatar.Stop;
end;
```

```
procedure
TF_Utama.mpLatarNotify(Sender:
TObject);
var
  CekMPLatarStatus: String;
begin
CekMPLatarStatus:=Ini.ReadString('SUAR
A','LATAR','');
  if CekMPLatarStatus='ON' then
  begin
  with Sender as TMediaPlayer do
  begin
    Notify:=True;
    if (Sender as
TMediaPlayer).Mode=mpStopped then
    begin
mpLatar.FileName:=Path+'\Suara\Latar.mid'
;
      mpLatar.Open;
      mpLatar.Play;
    end;
  end;
  end else
  begin
  end;
end;

procedure
TF_Utama.txtPesan1Change(Sender:
TObject);
var
  p, s : integer;
begin
  p := ((length(txtPesan1.Text) - 1) div 160)
+ 1;
  s := length(txtPesan1.Text) mod 160;
  if (s = 0) and (length(txtPesan1.Text) <> 0)
then s := 160;
  bSisa.Caption := ' Sisa = '+IntToStr(160 -
s) + ' (' + IntToStr(p) + ' sms)';
end;

procedure
TF_Utama.txtPesanBroadcastChange(Sende
r: TObject);
var
  p, s : integer;
begin
  p := ((length(txtPesanBroadcast.Text) - 1)
div 160) + 1;
  s := length(txtPesanBroadcast.Text) mod
160;
  if (s = 0) and
(length(txtPesanBroadcast.Text) <> 0) then
s := 160;
  txtSisaBroadcast.Caption := ' Sisa =
'+IntToStr(160 - s) + ' (' + IntToStr(p) + '
sms)';
end;

procedure TF_Utama.Button4Click(Sender:
TObject);
begin
  DataModule1.T_Registrasi.Post;
  Button1.Enabled:=True;
  Button2.Enabled:=True;
  Button3.Enabled:=True;
  Button4.Enabled:=False;
  Button5.Enabled:=False;
  Button6.Enabled:=True;
  dbEdit1.ReadOnly:=True;
  dbEdit2.ReadOnly:=True;
  dbEdit3.ReadOnly:=True;
  dbEdit4.ReadOnly:=True;
  dbEdit5.ReadOnly:=True;
  dbEdit6.ReadOnly:=True;
  DBCombobox1.ReadOnly:=True;
  DBCombobox2.ReadOnly:=True;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Button1Click(Sender:
TObject);
begin
  DataModule1.T_Registrasi.Insert;
  Button1.Enabled:=False;
  Button2.Enabled:=False;
  Button3.Enabled:=False;
  Button4.Enabled:=True;
  Button5.Enabled:=True;
  Button6.Enabled:=False;
  dbEdit1.ReadOnly:=False;
  dbEdit2.ReadOnly:=False;
  dbEdit3.ReadOnly:=False;
  dbEdit4.ReadOnly:=False;
  dbEdit5.ReadOnly:=False;
  dbEdit6.ReadOnly:=False;
  dbEdit6.Text:=DateTimeToStr(Now);
  DBCombobox1.ReadOnly:=False;
  DBCombobox2.ReadOnly:=False;
  dbEdit1.SetFocus;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Button2Click(Sender:
TObject);
begin
  DataModule1.T_Registrasi.Edit;
  Button1.Enabled:=False;
  Button2.Enabled:=False;
```

```
  Button3.Enabled:=False;
  Button4.Enabled:=True;
  Button5.Enabled:=True;
  Button6.Enabled:=False;
  dbEdit1.ReadOnly:=False;
  dbEdit2.ReadOnly:=False;
  dbEdit3.ReadOnly:=False;
  dbEdit4.ReadOnly:=False;
  dbEdit5.ReadOnly:=False;
  dbEdit6.ReadOnly:=False;
  DBCombobox1.ReadOnly:=False;
  DBCombobox2.ReadOnly:=False;
  dbEdit1.SetFocus;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Button3Click(Sender:
TObject);
begin
  Button1.Enabled:=False;
  Button2.Enabled:=False;
  Button3.Enabled:=False;
  Button4.Enabled:=False;
  Button5.Enabled:=False;
  Button6.Enabled:=False;
  if MessageDlg('Apa anda yakin ingin
menghapus record ini ???',
    mtConfirmation, [mbOK, mbCancel], 0)
= mrOK then
  begin
    DataModule1.T_Registrasi.Delete;
    Button1.Enabled:=True;
    Button2.Enabled:=True;
    Button3.Enabled:=True;
    Button4.Enabled:=False;
    Button5.Enabled:=False;
    Button6.Enabled:=True;
  end else
  begin
    Button1.Enabled:=True;
    Button2.Enabled:=True;
    Button3.Enabled:=True;
    Button4.Enabled:=False;
    Button5.Enabled:=False;
    Button6.Enabled:=True;
  end;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Button6Click(Sender:
TObject);
begin
  DataModule1.T_Registrasi.Active:=false;
  DataModule1.T_Registrasi.Active:=true;
  DataModule1.T_Registrasi.Refresh;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Button5Click(Sender:
TObject);
begin
  DataModule1.T_Registrasi.Cancel;
  Button1.Enabled:=True;
  Button2.Enabled:=True;
  Button3.Enabled:=True;
  Button4.Enabled:=False;
  Button5.Enabled:=False;
  Button6.Enabled:=True;
  dbEdit1.ReadOnly:=True;
  dbEdit2.ReadOnly:=True;
  dbEdit3.ReadOnly:=True;
  dbEdit4.ReadOnly:=True;
  dbEdit5.ReadOnly:=True;
  dbEdit6.ReadOnly:=True;
  DBCombobox1.ReadOnly:=True;
  DBCombobox2.ReadOnly:=True;
  Suara2(Sender as TObject);
end;

procedure TF_Utama.Login2Click(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  if Login2.Enabled=False then
  begin
    F_Password.LogoutDiklik(Sender as
TObject);
    bLogin.Caption:='Login';
    DataModule1.QLogin.SQL.Clear;
DataModule1.QLogin.SQL.Add('UPDATE
TBL_LOGIN SET VCH_STATUS = '+
QuotedStr('LOGOUT')+', '
+'DTM_LOGOUT =
'+QuotedStr(DateTimeToStr(Now))+
'WHERE VCH_USER_ID =
'+QuotedStr(F_Password.eUserID.Text)+
'AND VCH_ROLE =
'+QuotedStr(F_Password.eRole.Text)+
'AND VCH_STATUS =
'+QuotedStr('LOGIN'));
    DataModule1.QLogin.ExecSQL;
    F_Password.Show;
    F_Password.eRole.ItemIndex:=0;
    F_Utama.Enabled:=False;
  end else
  if Login2.Enabled=True then
  begin
    F_Password.Show;
    F_Password.eRole.ItemIndex:=0;
    F_Utama.Enabled:=False;
```

```
  bLogin.Caption:='Logout';
 end;
end;

procedure TF_Utama.Keluar2Click(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 Application.Terminate;
end;

procedure
TF_Utama.InfoSMS2Click(Sender:
TObject);
begin
 Suara1(Sender as TObject);
Swf3.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim3.swf';
 pInfoSMS.Align:=alClient;
 setVisiblePanel('pInfoSMS',True);
end;

procedure
TF_Utama.KirimSMS2Click(Sender:
TObject);
begin
Swf4.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim4.swf';
 Suara1(Sender as TObject);
 pKirimSMS.Align:=alClient;
 setVisiblePanel('pKirimSMS',True);
 ViewSent('SMS MANUAL');
end;

procedure
TF_Utama.Phonebook2Click(Sender:
TObject);
begin
Swf5.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim5.swf';
 Suara1(Sender as TObject);
 pPhonebook.Align:=alClient;
 setVisiblePanel('pPhonebook',True);
 DataModule1.QPhonebook.SQL.Clear;
DataModule1.QPhonebook.SQL.Add('SELE
CT
VCH_FULLNAME,VCH_PHONE_NUMB
ER,ID FROM TBL_PHONEBOOK
ORDER BY VCH_FULLNAME');
 DataModule1.QPhonebook.ExecSQL;
 DataModule1.QPhonebook.Open;
end;

procedure
TF_Utama.Panggilan2Click(Sender:
TObject);
begin
Swf6.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim6.swf';
 Suara1(Sender as TObject);
 pTelepon.Align:=alClient;
 setVisiblePanel('pTelepon',True);
 DataModule1.QPhonebook.SQL.Clear;
DataModule1.QPhonebook.SQL.Add('SELE
CT
VCH_FULLNAME,VCH_PHONE_NUMB
ER,ID FROM TBL_PHONEBOOK
ORDER BY VCH_FULLNAME');
 DataModule1.QPhonebook.ExecSQL;
 DataModule1.QPhonebook.Open;
end;

procedure
TF_Utama.Broadcast2Click(Sender:
TObject);
var
 TempTambah1,TempTambah2: String;
begin
 Suara1(Sender as TObject);
Swf7.Movie:=ExtractFilePath(Application.E
xeName)+'\Gambar\Anim7.swf';
 pBroadcast.Align:=alClient;
 setVisiblePanel('pBroadcast',True);
 ViewSent('SMS BROADCAST
MANUAL');
 txtNomorBroadcast.Text:='';
 Suara1(Sender as TObject);
 Datamodule1.QBroadcast.SQL.Clear;
Datamodule1.QBroadcast.SQL.Add('SELE
CT * FROM TBL_PELANGGAN');
 Datamodule1.QBroadcast.Open;
 Datamodule1.QBroadcast.First;
 while not Datamodule1.QBroadcast.Eof do
begin
txtNomorBroadcast.Text:=Datamodule1.QB
roadcast.FieldByName('VCH_PHONE_NU
MBER').AsString;
TempTambah1:=txtNomorBroadcast.Text;
TempTambah2:=TempTambah1+','+TempT
ambah2;
  TempTambah1:='';
txtNomorBroadcast.Text:=TempTambah2;
  Datamodule1.QBroadcast.Next;
 end;
end;
```

```
procedure    TF_Utama.UnregClick(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  pRegistrasi.Align:=alClient;
  setVisiblePanel('pRegistrasi',True);
end;

procedure  TF_Utama.bKonekClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  if Konek then
  begin
    Timer1.Enabled := false;
    XComm1.CloseDevice;
    bKonek.Caption := 'Connect';
    Konek := False;
    cbPort.Enabled := True;
    cbBaud.Enabled := True;
    cbFC.Enabled := True;
    pbSinyal.Position := 0;
    pbBattery.Position := 0;
    bMerk.Caption:='';
bModem.Caption:='';
    bImei.Caption:='';
bOperator.Caption:='';
    bService.Caption:='';
  end else
  begin
    XComm1.BaudValue                    :=
StrToInt(cbBaud.Text);
    XComm1.DeviceName  := cbPort.Text;
    XComm1.FlowControl                  :=
TFlowcontrol(cbFC.ItemIndex);
    If Not GoKonek Then
    begin
      bMerk.Caption:='';
bModem.Caption:='';
      bImei.Caption:='';
bOperator.Caption:='';
      bService.Caption:='';
    end else SetTerkoneksi;
  end;
end;

procedure  TF_Utama.About1Click(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  F_About:=TF_About.Create(self);
  F_About.ShowModal;
end;

procedure    TF_Utama.Help2Click(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  Application.HelpFile                  :=
ExtractFilePath(Application.ExeName)+'/He
lp/help.hlp';
Application.HelpCommand(HELP_CONTE
NTS, 0);
end;

procedure    TF_Utama.mLatarClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  if mLatar.Checked=False then
  begin
    mLatar.Checked:=True;
    Ini.WriteString('SUARA','LATAR','ON');
    SuaraLatarMulai(Sender    as    TObject,
Play);
  end else
  if mLatar.Checked=True then
  begin
    mLatar.Checked:=False;
Ini.WriteString('SUARA','LATAR','OFF');
SuaraLatarMulai(Sender as TObject, Stop);
  end;
end;

procedure
TF_Utama.mButtonShowClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  if mButtonShow.Checked=False then
  begin
    mButtonShow.Checked:=True;
Ini.WriteString('SUARA','BUTTONSHOW',
'ON');
  end else
  begin
    mButtonShow.Checked:=False;
Ini.WriteString('SUARA','BUTTONSHOW',
'OFF');
  end;
end;

procedure
TF_Utama.mButtonKlikClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  if mButtonKlik.Checked=False then
  begin
```

```
  mButtonKlik.Checked:=True;
Ini.WriteString('SUARA','BUTTONKLIK','
ON');
 end else
 begin
  mButtonKlik.Checked:=False;
Ini.WriteString('SUARA','BUTTONKLIK','
OFF');
 end;
end;

procedure
TF_Utama.bCekSMSClick(Sender:
TObject);
begin
 if not Konek then
 begin
  exit;
 end;
 Timer1.Enabled := False;
 lblSMS1.Text := '';
 ListSMS.Items.Clear;
 CekSMS;
 FilterDisplay;
 Timer1.Enabled := True;
 Suara2(Sender as TObject);
end;

procedure TF_Utama.setVisiblePanel(Panel:
String; setVisible: Boolean);
begin
 if Panel='pInfoSMS' then
 begin
  pInfoSMS.Visible  := setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pKirimSMS' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pPhonebook' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pBroadcast' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
```

```
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pTelepon' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pRegistrasi' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pKirimEmail' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pUserAccount' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataUser' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=setVisible;
  pViewDataLogin.Visible:=not setVisible;
```

```
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataLogin' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataPelanggan' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataPhonebook' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
pViewDataPhonebook.Visible:=setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataInboxSMS' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
pViewDataInboxSMS.Visible:=setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataInboxEmail' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
```

```
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
pViewDataInboxEmail.Visible:=setVisible;
  pViewDataSent.Visible:=not setVisible;
 end else
 if Panel='pViewDataSent' then
 begin
  pInfoSMS.Visible  := not setVisible;
  pKirimSMS.Visible  := not setVisible;
  pBroadcast.Visible := not setVisible;
  pPhonebook.Visible := not setVisible;
  pTelepon.Visible  := not setVisible;
  pRegistrasi.Visible:= not setVisible;
  pKirimEmail.Visible:= not setVisible;
  pUserAccount.Visible:=not setVisible;
  pViewDataUser.Visible:=not setVisible;
  pViewDataLogin.Visible:=not setVisible;
  pViewDataPelanggan.Visible:=not
setVisible;
  pViewDataPhonebook.Visible:=not
setVisible;
  pViewDataInboxSMS.Visible:=not
setVisible;
  pViewDataInboxEmail.Visible:=not
setVisible;
  pViewDataSent.Visible:=setVisible;
 end else
 begin
  pInfoSMS.Visible  := setVisible;
  pKirimSMS.Visible  := setVisible;
  pBroadcast.Visible := setVisible;
  pPhonebook.Visible := setVisible;
  pTelepon.Visible  := setVisible;
  pRegistrasi.Visible:= setVisible;
  pKirimEmail.Visible:= setVisible;
  pUserAccount.Visible:=setVisible;
  pViewDataUser.Visible:=setVisible;
  pViewDataLogin.Visible:=setVisible;
  pViewDataPelanggan.Visible:=setVisible;
pViewDataPhonebook.Visible:=setVisible;
pViewDataInboxSMS.Visible:=setVisible;
pViewDataInboxEmail.Visible:=setVisible;
  pViewDataSent.Visible:=setVisible;
 end;
end;

procedure
TF_Utama.KirimEMAIL1Click(Sender:
TObject);
begin
 Suara1(Sender as TObject);
 pKirimEmail.Align:=alClient;
 setVisiblePanel('pKirimEmail',True);
end;

procedure
TF_Utama.bLoginEmailClick(Sender:
TObject);
var
 mailman: TChilkatMailMan2;
 success: Integer;
 bundle:
CHILKATMAILLib2_TLB.IChilkatEmailB
undle2;
 i, jmlNumber: Integer;
 email:
CHILKATMAILLib2_TLB.IChilkatEmail2;
 cekDate: String;
 getSubj160, getBody160: String;
begin
 if        ((eUsername.Text<>'')        and
(ePassword.Text<>'')) then
 begin
Ini.WriteString('EMAIL','USERNAME',eUs
ername.Text);
  mailman                          :=
TChilkatMailMan2.Create(Self);
  success                          :=
mailman.UnlockComponent('30-day trial');
  if (success <> 1) then
  begin
   exit;
  end;
  UsernameGMail := eUsername.Text;
  PasswordGMail := ePassword.Text;
  mailman.MailHost := 'pop.gmail.com';
  mailman.PopUsername              :=
UsernameGMail;
  mailman.PopPassword              :=
PasswordGMail;
  mailman.PopSsl := 1;
  mailman.MailPort := 995;
  bundle := mailman.GetAllHeaders(1);
  if (bundle = nil ) then
  begin
   exit;
  end else
  begin
   eToName.Enabled    := True;
   eToEmail.Enabled   := True;
   eSubject.Enabled   := True;
   eBody.Enabled      := True;
   bSendEmail.Enabled  := True;
   bLoginEmail.Enabled := False;
   bLogoutEmail.Enabled:= True;
   bAttachFile.Enabled := True;
  end;
  for i := 0 to bundle.MessageCount - 1 do
```

```
  begin
    email := bundle.GetEmail(i);
    getSubj160:=Copy(email.subject,1,160);
    getBody160:=Copy(email.Body,1,160);
    DataModule1.QEmail.SQL.Clear;
    DataModule1.QEmail.SQL.Add('select
DTM_EMAIL_DATE from
tbl_email_inbox where
DTM_EMAIL_DATE =
'+QuotedStr(email.GetHeaderField('Date')));
    DataModule1.QEmail.ExecSQL;
    DataModule1.QEmail.Open;
    cekDate :=
DataModule1.QEmail.FieldByName('DTM_
EMAIL_DATE').AsString;
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT MAX(NMB_IDX) AS JUM FROM
TBL_EMAIL_INBOX WHERE
VCH_EMAIL_ADDRESS =
'+QuotedStr(UsernameGMail));
    DataModule1.QCekInboxReg.ExecSQL;
    DataModule1.QCekInboxReg.Open;
    jmlNumber :=
DataModule1.QCekInboxReg.FieldByName
('JUM').AsInteger+1;
    if
(cekDate<>email.GetHeaderField('Date'))
then
    begin
      DataModule1.QEmail.Close;
      DataModule1.QEmail.SQL.Clear;
DataModule1.QEmail.SQL.Add('INSERT
INTO TBL_EMAIL_INBOX (NMB_IDX,
VCH_EMAIL_ADDRESS,DTM_EMAIL_
DATE,VCH_FROM_NAME,VCH_FROM_
EMAIL_ADDRESS,VCH_SUBJECT,VCH
_BODY,VCH_STATUS,VCH_PROCESS)
VALUES ('+
      IntToStr(jmlNumber)+','+
      QuotedStr(UsernameGMAIL)+','
+QuotedStr(email.GetHeaderField('Date'))+
','
+QuotedStr(email.fromName)+','
+QuotedStr(email.FromAddress)+','
+QuotedStr(getSubj160)+','
+QuotedStr(getBody160)+','
+QuotedStr('EMAIL DOWNLOADED
MANUAL')+','
+QuotedStr('False')+')');
DataModule1.QEmail.ExecSQL;
    end;
  end;
DataModule1.QEmailDownload.SQL.Clear;
DataModule1.QEmailDownload.SQL.Add('
select * from tbl_email_inbox where
VCH_EMAIL_ADDRESS =
'+QuotedStr(UsernameGMail)+
'and VCH_STATUS = '+QuotedStr('EMAIL
DOWNLOADED MANUAL')+
'order by nmb_idx');
DataModule1.QEmailDownload.ExecSQL;
  DataModule1.QEmailDownload.Open;
  DataModule1.QEmailSent.SQL.Clear;
DataModule1.QEmailSent.SQL.Add('select
* from tbl_sent where
VCH_EMAIL_ADDRESS =
'+QuotedStr(UsernameGMail)+
'and VCH_STATUS = '+QuotedStr('EMAIL
SENT MANUAL'));
  DataModule1.QEmailSent.ExecSQL;
  DataModule1.QEmailSent.Open;
 end else ShowMessage('Username dan
Password harus diisi!!!');
end;

procedure
TF_Utama.bSendEmailClick(Sender:
TObject);
var
 mailman: TChilkatMailMan2;
 success: Integer;
 email:
CHILKATMAILLib2_TLB.IChilkatEmail2;
begin
 if ((eToName.Text<>'') and
(eToEmail.Text<>'') and
   (eSubject.Text<>'') and
(eBody.Text<>'')) then
 begin
Ini.WriteString('EMAIL','TO_NAME',eToN
ame.Text);
Ini.WriteString('EMAIL','TO_EMAIL',eToE
mail.Text);
Ini.WriteString('EMAIL','SUBJECT',eSubje
ct.Text);
Ini.WriteString('EMAIL','BODY',eBody.Te
xt);
  mailman :=
TChilkatMailMan2.Create(Self);
  success :=
mailman.UnlockComponent('30-day trial');
  if (success <> 1) then
  begin
    exit;
  end;
  mailman.SmtpHost := 'smtp.gmail.com';
```

```
  mailman.SmtpUsername :=
UsernameGMail;
  mailman.SmtpPassword :=
PasswordGMail;
  mailman.StartTLS := 25;
  email := CoChilkatEmail2.Create();
  if (EmailAttachment<>'') then
  begin
email.AddFileAttachment(EmailAttachment
);
  end;
  email.From := 'SMS GATEWAY
<'+UsernameGMail+'>';
  email.Subject := eSubject.Text;
  email.Body := eBody.Text;
email.AddTo(eToName.Text,eToEmail.Text
);
  success := mailman.SendEmail(email As
CHILKATMAILLib2_TLB.IChilkatEmail2)
;
  if (success <> 1) then
  begin
   exit;
  end else
  begin
   DataModule1.QEmail.Close;
   DataModule1.QEmail.SQL.Clear;
DataModule1.QEmail.SQL.Add('INSERT
INTO TBL_SENT
(VCH_EMAIL_ADDRESS,DTM_EMAIL_
DATE,VCH_TO_NAME,VCH_TO_EMAI
L_ADDRESS,VCH_SUBJECT,VCH_BOD
Y,VCH_STATUS,VCH_PHONE_NUMBE
R) VALUES ('+
   QuotedStr(UsernameGMAIL)+','+
   QuotedStr(DateTimeToStr(Now))+','+
   QuotedStr(eToName.Text)+','+
   QuotedStr(eToEmail.Text)+','+
   QuotedStr(eSubject.Text)+','+
   QuotedStr(eBody.Text)+','+
   QuotedStr('EMAIL SENT
MANUAL')+','+
   QuotedStr('-')+')');
   DataModule1.QEmail.ExecSQL;
  end;
  success :=
mailman.CloseSmtpConnection();
  if (success <> 1) then
  begin
   exit;
  end;
  DataModule1.QEmailSent.SQL.Clear;
DataModule1.QEmailSent.SQL.Add('select
* from tbl_sent where
VCH_EMAIL_ADDRESS =
'+QuotedStr(UsernameGMail)+
' and VCH_STATUS
= '+QuotedStr('EMAIL SENT MANUAL'));
  DataModule1.QEmailSent.ExecSQL;
  DataModule1.QEmailSent.Open;
 end else ShowMessage('To Name, To
Email Address, Subject dan Body tidak
boleh kosong!!!');
end;

procedure
TF_Utama.bLogoutEmailClick(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 eUsername.Text := '';
 ePassword.Text := '';
 eToName.Enabled := False;
 eToEmail.Enabled := False;
 eSubject.Enabled := False;
 eBody.Enabled := False;
 bSendEmail.Enabled := False;
 bLoginEmail.Enabled := True;
 bLogoutEmail.Enabled:= False;
 bAttachFile.Enabled := False;
DataModule1.QEmailDownload.SQL.Clear;
DataModule1.QEmailDownload.SQL.Add('
select * from tbl_email_inbox where
vch_email_address =
'+QuotedStr('NO_DATA'));
 DataModule1.QEmailDownload.ExecSQL;
 DataModule1.QEmailDownload.Open;
 DataModule1.QEmailSent.SQL.Clear;
 DataModule1.QEmailSent.SQL.Add('select
* from tbl_sent where vch_email_address =
'+QuotedStr('NO_DATA'));
 DataModule1.QEmailSent.ExecSQL;
 DataModule1.QEmailSent.Open;
end;

procedure
TF_Utama.bKirimBroadcastClick(Sender:
TObject);
var
 c : string;
 i : integer;
 a, l: TListItem;
begin
 Suara2(Sender as TObject);
 if txtNomorBroadcast.Text = '' then
 begin
  exit;
 end;
 if not Konek then
```

```
  begin
    ShowMessage('Belum terkoneksi ke HP');
    exit;
  end;
  Timer1.Enabled := False;
  c                                        :=
AnsiReplaceStr(txtNomorBroadcast.Text, ',',
#13#10);
  Item.Text := c;
  for i := 0 to Item.Count - 1 do
  begin
    c := trim(Item.Strings[i]);
    if                (KirimSMSBroadcast(c,
txtPesanBroadcast.Text)) then
    begin
      l := ListSMS.Items.Add;
      l.Caption := rubahKeNama(c);
      l.SubItems.Add(DateTimeToStr(Now));
l.SubItems.Add(txtPesanBroadcast.Text);
      l.SubItems.Add('PC');
      l.SubItems.Add('Terkirim');
      l.SubItems.Add('0');
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesanBroadcast.Text,              'SMS
BROADCAST                  MANUAL',
txtNomorBroadcast.Text);
      ViewSent('SMS          BROADCAST
MANUAL');
      a := lvSMS1.Items.Add;
      a.Caption := l.Caption;
      a.SubItems := l.SubItems;
    end else
  end;
  FilterDisplay;
  Timer1.Enabled := True;
end;

procedure
TF_Utama.bClearBroadcastClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  txtPesanBroadcast.Text:='';
end;

procedure
TF_Utama.bTutupPanelClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  setVisiblePanel('Lain-Lain',False);
end;

procedure   TF_Utama.bKirimClick(Sender:
TObject);
var
  c : string;
  i : integer;
  a, l: TListItem;
begin
  Suara2(Sender as TObject);
  if txtNomor1.Text = '' then
  begin
    ShowMessage('Nomor harus diisikan');
    exit;
  end;
  if not Konek then
  begin
    ShowMessage('Belum terkoneksi ke HP');
    exit;
  end;
  Timer1.Enabled := False;
  c  :=  AnsiReplaceStr(txtNomor1.Text,  ',',
#13#10);
  Item.Text := c;
  for i := 0 to Item.Count - 1 do
  begin
    c := trim(Item.Strings[i]);
    if (KirimSMS(c, txtPesan1.Text)) then
    begin
      l := ListSMS.Items.Add;
      l.Caption := rubahKeNama(c);
      l.SubItems.Add(DateTimeToStr(Now));
      l.SubItems.Add(txtPesan1.Text);
      l.SubItems.Add('PC');
      l.SubItems.Add('Terkirim');
      l.SubItems.Add('0');
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan1.Text,      'SMS      MANUAL',
txtNomor1.Text);
      ViewSent('SMS MANUAL');
      a := lvSMS1.Items.Add;
      a.Caption := l.Caption;
      a.SubItems := l.SubItems;
    end else
  end;
  FilterDisplay;
  Timer1.Enabled := True;
end;

procedure   TF_Utama.bClearClick(Sender:
TObject);
begin
  Suara2(Sender as TObject);
  txtNomor1.Text:='';
  txtPesan1.Text:='';
end;
```

```
procedure
TF_Utama.bSimpan1Click(Sender:
TObject);
var
 l : TListItem;
begin
 Suara2(Sender as TObject);
 if (txtNamaPB1.Text = '') or
(txtNomorPB1.Text = '') then
 begin
  ShowMessage('Isikan nama dan nomor');
  exit;
 end;
 while ADOQuery1.Active do
ADOQuery1.Close;
 if PBID <> '' then
 begin
  ADOConnection1.Execute('UPDATE
TBL_Phonebook set
VCH_PHONE_NUMBER='
+QuotedStr(txtNomorPB1.Text) +
',VCH_FULLNAME='
+QuotedStr(txtNamaPB1.Text) + ' where
id=' + PBID);
l := lvPBook1.FindCaption(0, NamaLama,
False, True, True);
  If l <> Nil then
  begin
   l.Caption := txtNamaPB1.Text;
   l.SubItems[0] := txtNomorPB1.Text;
  end;
 end else
 begin
  ADOConnection1.Execute('INSERT
INTO
TBL_Phonebook(VCH_FULLNAME,VCH
_PHONE_NUMBER) values('
+QuotedStr(txtNamaPB1.Text) + ','
+QuotedStr(txtNomorPB1.Text) + ')');
ADOQuery1.SQL.Text := 'select max(id) as
baru from TBL_Phonebook';
  ADOQuery1.Open;
  l := lvPBook1.Items.Add;
  l.Caption := txtNamaPB1.Text;
  l.SubItems.Add(txtNomorPB1.Text);
l.SubItems.Add(ADOQuery1.FieldValues['b
aru'])
 end;
 FilterDisplay;
 If l <> Nil then
 begin
  l.MakeVisible(False);
  l.Selected := True;
 end;
 txtNamaPB1.Clear;
 txtNomorPB1.Clear;
 PBID := '';
end;

procedure TF_Utama.bBaru1Click(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 txtNamaPB1.Clear;
 txtNomorPB1.Clear;
 PBID := '';
end;

procedure TF_Utama.bBaca1Click(Sender:
TObject);
var
 l: TListItem;
 s, c: string;
 i, j, l1, l2, p1, p2 : integer;
begin
 Suara2(Sender as TObject);
 if not Konek then
 begin
  ShowMessage('Belum terkoneksi ke HP');
  exit;
 end;
 s := SendGetData('AT+CPBS="' +
arSumber[cbJenisPB1.itemindex]+'"'#13,
sOK);
 if Pos(sOK, s) = 0 then
 begin
  ShowMessage('Tidak dapat memilih
media ' + cbJenisPB1.Text);
  exit;
 end;
 s := SendGetData('AT+CPBR=?'+#13,
sOK);
 if Pos(sOK, s) = 0 then
 begin
  ShowMessage('Tidak dapat membaca
buku kontak');
  exit;
 end;
 p1 := pos('(', s) + 1;
 p2 := pos(')', s);
 c := copy(s, p1, p2 - p1);
 p1 := pos('-', c);
 if p1 = 0 Then exit;
 Timer1.Enabled := False;
 l1 := StrToInt(copy(c, 1, p1 - 1));
 l2 := StrToInt(copy(c, p1 + 1, length(c)));
 prgbar1.Min := l1;
 prgbar1.Max := l2;
 prgbar1.Visible := true;
 j := 0;
```

```
 for i := l1 to l2 do
 begin
  prgbar1.Position := i;
  s := SendGetData('AT+CPBR=' +
IntToStr(i)+#13, sOK);
  p1 := pos(':', s) + 2;
  s := Copy(s, p1, length(s));
  List.Text := AnsiReplaceStr(s, ',',
#13#10);
  If List.Count > 3 then
  begin
   l := lvPBook1.FindCaption(0,
TrimAll(List.Strings[3]), False, True, True);
   If l = Nil then
   begin
    ADOConnection1.Execute('INSERT
INTO
TBL_Phonebook(VCH_FULLNAME,VCH
_PHONE_NUMBER) values('
+QuotedStr(TrimAll(List.Strings[3])) + ','
+QuotedStr(TrimAll(List.Strings[1])) + ')');
    ADOQuery1.SQL.Text := 'select
max(id) as baru from TBL_Phonebook';
    ADOQuery1.Open;
    l := lvPBook1.Items.Add;
    l.Caption := TrimAll(List.Strings[3]);
l.SubItems.Add(TrimAll(List.Strings[1]));
l.SubItems.Add(ADOQuery1.FieldValues['b
aru']);
    Inc(j);
   end;
  end;
 end;
 prgbar1.Visible := false;
 ShowMessage(IntToStr(j) + ' phonebook
terbaca');
 Timer1.Enabled := True;
 DataModule1.QPhonebook.SQL.Clear;
DataModule1.QPhonebook.SQL.Add('SELE
CT
VCH_FULLNAME,VCH_PHONE_NUMB
ER,ID FROM TBL_PHONEBOOK
ORDER BY VCH_FULLNAME');
 DataModule1.QPhonebook.ExecSQL;
 DataModule1.QPhonebook.Open;
end;

procedure TF_Utama.bStopClick(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 if not Konek then
 begin
  ShowMessage('Belum terkoneksi ke HP');
  exit;
 end;
 if TButton(Sender).Tag = 2 then
 begin
  SendGetData('ATH'#13, sOK);
 end else
 begin
  if txtNomorCall1.Text = '' then
  begin
   ShowMessage('Nomor tujuan belum
diisikan');
   exit;
  end;
  SendGetData('AT&FE0'#13, sOK);
  SendGetData('ATD ' +
txtNomorCall1.Text + ';'#13, sOK);
 end;
end;

procedure
TF_Utama.cbReport1Click(Sender:
TObject);
begin
 if cbReport1.Checked then
Ini.WriteString('SMS','REPORT','1')else
 Ini.WriteString('SMS','REPORT','0');
end;

procedure
TF_Utama.cbReport2Click(Sender:
TObject);
begin
 if cbReport2.Checked then
Ini.WriteString('SMS','REPORT','1')else
 Ini.WriteString('SMS','REPORT','0');
end;

procedure
TF_Utama.KirimSMSOtomatisCekInbox(Se
nder: TObject);
var
 c : string;
 i : integer;
 a, l: TListItem;
begin
 Timer1.Enabled := False;
 c := AnsiReplaceStr(txtNomor, ',', #13#10);
 Item.Text := c;
 for i := 0 to Item.Count - 1 do
 begin
  c := trim(Item.Strings[i]);
  if (KirimSMS(c, txtPesan)) then
  begin
   l := ListSMS.Items.Add;
   l.Caption := rubahKeNama(c);
   l.SubItems.Add(DateTimeToStr(Now));
```

```
    l.SubItems.Add(txtPesan);
    l.SubItems.Add('PC');
    l.SubItems.Add('Terkirim');
    l.SubItems.Add('0');
    a := lvSMS1.Items.Add;
    a.Caption := l.Caption;
    a.SubItems := l.SubItems;
   end else
  end;
  FilterDisplay;
  Timer1.Enabled := True;
end;

procedure
TF_Utama.KirimEmailOtomatisCekInbox(t
xtToName,          txtToEmailAddress,
txtToSubject,
 txtToBody,  txtUsername,  txtPassword,
txtFromName: String);
var
 mailman: TChilkatMailMan2;
 success: Integer;
 email:
CHILKATMAILLib2_TLB.IChilkatEmail2;
begin
 mailman                          :=
TChilkatMailMan2.Create(Self);
 success                          :=
mailman.UnlockComponent('30-day trial');
 if (success <> 1) then
 begin
  exit;
 end;
 mailman.SmtpHost := 'smtp.gmail.com';
 mailman.SmtpUsername := txtUsername;
 mailman.SmtpPassword := txtPassword;
 mailman.StartTLS := 25;
 email := CoChilkatEmail2.Create();
 email.From          :=  txtFromName+'
<'+txtUsername+'>';
 email.Subject := txtToSubject;
 email.Body          :=  txtToBody;
email.AddTo(txtToName,txtToEmailAddres
s);
 success := mailman.SendEmail(email As
CHILKATMAILLib2_TLB.IChilkatEmail2)
;
 if (success <> 1) then
 begin
  exit;
 end else
 begin
  DataModule1.QEmail.Close;
  DataModule1.QEmail.SQL.Clear;
  DataModule1.QEmail.SQL.Add('INSERT
INTO                TBL_SENT
(VCH_EMAIL_ADDRESS,DTM_EMAIL_
DATE,VCH_TO_NAME,VCH_TO_EMAI
L_ADDRESS,VCH_SUBJECT,VCH_BOD
Y,VCH_STATUS,VCH_PHONE_NUMBE
R) VALUES ('+
  QuotedStr(txtUsername)+','+
  QuotedStr(DateTimeToStr(Now))+','+
  QuotedStr(txtToName)+','+
  QuotedStr(txtToEmailAddress)+','+
  QuotedStr(txtToSubject)+','+
  QuotedStr(txtToBody)+','+
  QuotedStr('EMAIL          SENT
OTOMATIS')+','+
  QuotedStr('-')+')');
  DataModule1.QEmail.ExecSQL;
 end;
 success                          :=
mailman.CloseSmtpConnection();
 if (success <> 1) then
 begin
  exit;
 end;
end;

procedure TF_Utama.CekInboxReg(Sender:
TObject);
var
 CekValidEmail,      CekStatusPelanggan:
String;
 I, getLength: Integer;
 REG,            NAMA_LENGKAP,
JENIS_KELAMIN,          ALAMAT,
GMAIL_ADDRESS,
GMAIL_PASSWORD: String;
begin
 DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT  *  FROM  TBL_SMS_INBOX
WHERE      VCH_ISI_SMS      LIKE
'+QuotedStr('Reg%')+'          AND
VCH_PROCESS = '+QuotedStr('False'));
 DataModule1.QCekInboxReg.ExecSQL;
 DataModule1.QCekInboxReg.Open;
txtTokenSource:=DataModule1.QCekInbox
Reg.FieldByName('VCH_ISI_SMS').AsStri
ng;
CekProses1:=DataModule1.QCekInboxReg.
FieldByName('VCH_PROCESS').AsString;
CekNomorInbox:=DataModule1.QCekInbox
Reg.FieldByName('VCH_PHONE_NUMB
ER').AsString;
 if CekProses1='False' then
 begin
```

```
 for I:=1 to 6 do
 begin
  CreateTokens('#', txtTokenSource);
  getLength:=Length(txtTokenResult);
txtTokenSource:=Copy(txtTokenSource,1,L
ength(txtTokenSource)-(getLength+1));
  CekReg[I]:=txtTokenResult;
 end;
 REG:=Uppercase(CekReg[6]);
NAMA_LENGKAP:=Uppercase(CekReg[5]
);
JENIS_KELAMIN:=Uppercase(CekReg[4])
;
 ALAMAT:=Uppercase(CekReg[3]);
 GMAIL_ADDRESS:=CekReg[2];
///Uppercase(CekReg[2]);
 GMAIL_PASSWORD:=CekReg[1];
///Uppercase(CekReg[1]);
CekValidEmail:=Copy(GMAIL_ADDRESS
,Length(GMAIL_ADDRESS)-8,9);
 if (REG='REG') and (CekProses1='False')
then
 begin
  if (CekValidEmail='gmail.com') then
  begin
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
SELECT * FROM TBL_PELANGGAN
WHERE VCH_PHONE_NUMBER =
'+QuotedStr(CekNomorInbox));
DataModule1.QUpdateInboxReg.ExecSQL;
   DataModule1.QUpdateInboxReg.Open;
CekStatusPelanggan:=DataModule1.QUpdat
eInboxReg.FieldByName('VCH_STATUS').
AsString;
   if CekStatusPelanggan='REGISTERED'
then
   begin
    txtPesan:='Data anda sudah ada. Untuk
mengirim email, ketik:
EMAIL#TO_NAME#TO_EMAIL_ADDRE
SS#SUBJECT#BODY#. Cth:
email#reza#reza_itn@gmail.com#Test#Cob
a Email#';
    txtNomor:=CekNomorInbox;
    KirimSMSOtomatisCekInbox(Sender
as TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);
   end else
   if
CekStatusPelanggan='UNREGISTERED'
then
   begin
    txtPesan:='Terimakasih anda sudah
terdaftar. Untuk mengirim email, ketik:
EMAIL#TO_NAME#TO_EMAIL_ADDRE
SS#SUBJECT#BODY#. Cth:
email#reza#reza_itn@gmail.com#Test#Cob
a Email#';
    txtNomor:=CekNomorInbox;
    KirimSMSOtomatisCekInbox(Sender
as TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);

DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_PELANGGAN SET
VCH_STATUS =
'+QuotedStr('REGISTERED')+
'WHERE VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor)+
'AND VCH_STATUS =
'+QuotedStr('UNREGISTERED'));
DataModule1.QUpdateInboxReg.ExecSQL;
   end else
   begin
    txtPesan:='Terimakasih anda sudah
terdaftar. Untuk mengirim email, ketik:
EMAIL#TO_NAME#TO_EMAIL_ADDRE
SS#SUBJECT#BODY#. Cth:
email#reza#reza_itn@gmail.com#Test#Cob
a Email#';
    txtNomor:=CekNomorInbox;
    KirimSMSOtomatisCekInbox(Sender
as TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
INSERT INTO TBL_PELANGGAN
(VCH_FULLNAME,VCH_GENDER,VCH
_PHONE_NUMBER,VCH_HOME_ADDR
ESS,VCH_EMAIL_ADDRESS,VCH_EMA
IL_PASSWORD,VCH_STATUS,DTM_DA
TE_REGISTER) VALUES('+
QuotedStr(NAMA_LENGKAP)+','
+QuotedStr(Copy(JENIS_KELAMIN,1,1))+
','
+QuotedStr(txtNomor)+','
+QuotedStr(ALAMAT)+','
+QuotedStr(GMAIL_ADDRESS)+','
+QuotedStr(GMAIL_PASSWORD)+','
```

```
+QuotedStr('REGISTERED')+','
+QuotedStr(DateTimeToStr(Now))+')');
DataModule1.QUpdateInboxReg.ExecSQL;
    end;
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE      TBL_SMS_INBOX      SET
VCH_PROCESS = '+QuotedStr('True')+', '+
                        'VCH_ISI_SMS =
'+QuotedStr(REG+'#'+NAMA_LENGKAP+
'#'+Copy(JENIS_KELAMIN,1,1)+'#'+
ALAMAT+'#'+GMAIL_ADDRESS+'#'+'$$
$$$$$$$$$#')+
'WHERE      VCH_ISI_SMS      LIKE
'+QuotedStr('Reg%')+'          AND
VCH_PHONE_NUMBER              =
'+QuotedStr(txtNomor));
DataModule1.QUpdateInboxReg.ExecSQL;
    txtNomor:='';
    txtPesan:='';
   end else
   begin
    txtPesan:='Maaf, email address harus
menggunakan ACCOUNT GMAIL.';
    txtNomor:=CekNomorInbox;
    KirimSMSOtomatisCekInbox(Sender as
TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE      TBL_SMS_INBOX      SET
VCH_PROCESS = '+QuotedStr('True')+', '+
                        'VCH_ISI_SMS =
'+QuotedStr(REG+'#'+NAMA_LENGKAP+
'#'+Copy(JENIS_KELAMIN,1,1)+'#'+
ALAMAT+'#'+GMAIL_ADDRESS+'#'+'$$
$$$$$$$$$#')+
'WHERE      VCH_ISI_SMS      LIKE
'+QuotedStr('Reg%')+'          AND
VCH_PHONE_NUMBER              =
'+QuotedStr(txtNomor));
DataModule1.QUpdateInboxReg.ExecSQL;
   end;
  end else
  if (REG<>'REG') and (CekProses1='False')
then
  begin
   txtPesan:='Maaf, format yang anda
kirimkan salah. Ketik:
REG#NAMA_LENGKAP#JENIS_KELAM
IN(L/P)#ALAMAT#GMAIL_ADDRESS#G
MAIL_PASSWORD#.          Cth:
reg#reza#L#malang#reza_itn@gmail.com#
maafRahasia#';
   txtNomor:=CekNomorInbox;
   KirimSMSOtomatisCekInbox(Sender as
TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE      TBL_SMS_INBOX      SET
VCH_PROCESS = '+QuotedStr('True')+', '
+'VCH_ISI_SMS                  =
'+QuotedStr(Uppercase(CekReg[6])+'#'+Up
percase(CekReg[5])+'#'+Uppercase(Copy(C
ekReg[4],1,1))+'#'+
Uppercase(CekReg[3])+'#'+Uppercase(Cek
Reg[2])+'#'+'$$$$$$$$$$$#')+
'WHERE      VCH_ISI_SMS      LIKE
'+QuotedStr('Reg%')+'          AND
VCH_PHONE_NUMBER              =
'+QuotedStr(txtNomor));
DataModule1.QUpdateInboxReg.ExecSQL;
  end;
  end;
end;

procedure
TF_Utama.CekInboxSMSEmail(Sender:
TObject);
var
  I, getLength: Integer;
  EMAIL,                TO_NAME,
TO_EMAIL_ADDRESS,   TO_SUBJECT,
TO_BODY: String;
  FROM_EMAIL_ADDRESS,
FROM_EMAIL_PASSWORD,
FROM_FULLNAME: String;
begin
  DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT  *  FROM  TBL_SMS_INBOX
WHERE      VCH_ISI_SMS      LIKE
'+QuotedStr('Email%')+'          AND
VCH_PROCESS = '+QuotedStr('False'));
  DataModule1.QCekInboxReg.ExecSQL;
  DataModule1.QCekInboxReg.Open;
txtTokenSource:=DataModule1.QCekInbox
Reg.FieldByName('VCH_ISI_SMS').AsStri
ng;
CekProses2:=DataModule1.QCekInboxReg.
FieldByName('VCH_PROCESS').AsString;
CekNomorInbox:=DataModule1.QCekInbox
```

```
Reg.FieldByName('VCH_PHONE_NUMB
ER').AsString;
 txtNomor:=CekNomorInbox;
 if CekProses2='False' then
 begin
 for I:=1 to 5 do
 begin
  CreateTokens('#', txtTokenSource);
  getLength:=Length(txtTokenResult);
txtTokenSource:=Copy(txtTokenSource,1,L
ength(txtTokenSource)-(getLength+1));
  CekSMS2Email[I]:=txtTokenResult;
 end;
 EMAIL:=Uppercase(CekSMS2Email[5]);
TO_NAME:=Uppercase(CekSMS2Email[4]
);
TO_EMAIL_ADDRESS:=CekSMS2Email[
3];        ///Uppercase(CekSMS2Email[3]);
TO_SUBJECT:=Uppercase(CekSMS2Email
[2]);
 TO_BODY:=CekSMS2Email[1];
 if        (EMAIL='EMAIL')        and
(CekProses2='False') then
 begin
  DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT * FROM TBL_PELANGGAN
WHERE VCH_PHONE_NUMBER =
'+QuotedStr(CekNomorInbox)+'  AND
VCH_STATUS =
'+QuotedStr('REGISTERED'));
  DataModule1.QCekInboxReg.ExecSQL;
  DataModule1.QCekInboxReg.Open;
FROM_EMAIL_ADDRESS:=DataModule1
.QCekInboxReg.FieldByName('VCH_EMA
IL_ADDRESS').AsString;
FROM_EMAIL_PASSWORD:=DataModul
e1.QCekInboxReg.FieldByName('VCH_EM
AIL_PASSWORD').AsString;
FROM_FULLNAME:=DataModule1.QCek
InboxReg.FieldByName('VCH_FULLNAM
E').AsString;
KirimEmailOtomatisCekInbox(TO_NAME,
TO_EMAIL_ADDRESS, TO_SUBJECT,
   TO_BODY,
FROM_EMAIL_ADDRESS,
FROM_EMAIL_PASSWORD,
FROM_FULLNAME);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_SMS_INBOX SET
VCH_PROCESS = '+QuotedStr('True')+
'WHERE VCH_ISI_SMS =
'+QuotedStr(EMAIL+'#'+TO_NAME+'#'+T
O_EMAIL_ADDRESS+'#'+TO_SUBJECT
+'#'+TO_BODY+'#')+
'AND VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor)+
'AND VCH_PROCESS =
'+QuotedStr('False'));
DataModule1.QUpdateInboxReg.ExecSQL;
 end else
 if        (EMAIL<>'EMAIL')        and
(CekProses2='False') then
 begin
  txtPesan:='Maaf, format yang anda
kirimkan salah. Ketik:
EMAIL#TO_NAME#TO_EMAIL_ADDRE
SS#SUBJECT#BODY#. Cth:
email#reza#reza_itn@gmail.com#Test#Cob
a Email#';
  txtNomor:=CekNomorInbox;
  KirimSMSOtomatisCekInbox(Sender as
TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT REG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_SMS_INBOX SET
VCH_PROCESS = '+QuotedStr('True')+
'WHERE VCH_ISI_SMS =
'+QuotedStr(EMAIL+'#'+TO_NAME+'#'+T
O_EMAIL_ADDRESS+'#'+TO_SUBJECT
+'#'+TO_BODY+'#')+
'AND VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor)+
'AND VCH_PROCESS =
'+QuotedStr('False'));
DataModule1.QUpdateInboxReg.ExecSQL;
 end;
 end;
end;

procedure
TF_Utama.CekInboxEmailSMS(Sender:
TObject);
var
 I, J, jmlPelanggan, jmlNumber: Integer;
 GET_VCH_PHONE_NUMBER,
GET_VCH_EMAIL_ADDRESS,
GET_VCH_EMAIL_PASSWORD: String;
 mailman: TChilkatMailMan2;
 success: Integer;
 bundle:
CHILKATMAILLib2_TLB.IChilkatEmailB
undle2;
```

```
 email:
CHILKATMAILLib2_TLB.IChilkatEmail2;
 cekDate: String;
 getSubj160,getBody160: String;
begin
 DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT MAX(ID) AS JUM FROM
TBL_PELANGGAN WHERE
VCH_STATUS =
'+QuotedStr('REGISTERED'));
 DataModule1.QCekInboxReg.ExecSQL;
 DataModule1.QCekInboxReg.Open;
jmlPelanggan:=DataModule1.QCekInboxRe
g.FieldByName('JUM').AsInteger;
 for I:=1 to jmlPelanggan do
 begin
  DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT * FROM TBL_PELANGGAN
WHERE ID = '+IntToStr(I));
DataModule1.QCekInboxReg.ExecSQL;
DataModule1.QCekInboxReg.Open;
GET_VCH_PHONE_NUMBER:=DataMod
ule1.QCekInboxReg.FieldByName('VCH_P
HONE_NUMBER').AsString;
GET_VCH_EMAIL_ADDRESS:=DataMod
ule1.QCekInboxReg.FieldByName('VCH_E
MAIL_ADDRESS').AsString;
GET_VCH_EMAIL_PASSWORD:=DataM
odule1.QCekInboxReg.FieldByName('VCH
_EMAIL_PASSWORD').AsString;
        mailman :=
TChilkatMailMan2.Create(Self);
  success :=
mailman.UnlockComponent('30-day trial');
  if (success <> 1) then
  begin
   Exit;
  end;
  mailman.MailHost := 'pop.gmail.com';
  mailman.PopUsername :=
GET_VCH_EMAIL_ADDRESS;
  mailman.PopPassword :=
GET_VCH_EMAIL_PASSWORD;
  mailman.PopSsl := 1;
  mailman.MailPort := 995;
  bundle := mailman.GetAllHeaders(1);
  if (bundle = nil ) then
  begin
   Exit;
  end;
  try
   for J:=0 to bundle.MessageCount - 1 do
   begin
    email := bundle.GetEmail(J);
getSubj160:=Copy(email.subject,1,160);
    getBody160:=Copy(email.Body,1,160);
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('sel
ect DTM_EMAIL_DATE from
tbl_email_inbox where
DTM_EMAIL_DATE =
'+QuotedStr(email.GetHeaderField('Date')));
DataModule1.QCekInboxReg.ExecSQL;
    DataModule1.QCekInboxReg.Open;
      cekDate :=
DataModule1.QCekInboxReg.FieldByName
('DTM_EMAIL_DATE').AsString;
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT MAX(NMB_IDX) AS JUM FROM
TBL_EMAIL_INBOX WHERE
VCH_EMAIL_ADDRESS =
'+QuotedStr(GET_VCH_EMAIL_ADDRES
S));
DataModule1.QCekInboxReg.ExecSQL;
    DataModule1.QCekInboxReg.Open;
    jmlNumber :=
DataModule1.QCekInboxReg.FieldByName
('JUM').AsInteger+1;
    if
(cekDate<>email.GetHeaderField('Date'))
then
    begin
     DataModule1.QCekInboxReg.Close;
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('IN
SERT INTO TBL_EMAIL_INBOX
(NMB_IDX,
VCH_EMAIL_ADDRESS,DTM_EMAIL_
DATE,VCH_FROM_NAME,VCH_FROM_
EMAIL_ADDRESS,VCH_SUBJECT,VCH
_BODY,VCH_STATUS,VCH_PROCESS)
VALUES ('+
     IntToStr(jmlNumber)+','+
QuotedStr(GET_VCH_EMAIL_ADDRESS
)+','
+QuotedStr(email.GetHeaderField('Date'))+'
,'
+QuotedStr(email.fromName)+','
+QuotedStr(email.FromAddress)+','
+QuotedStr(getSubj160)+','
+QuotedStr(getBody160)+','
+QuotedStr('EMAIL DOWNLOADED
OTOMATIS')+','
+QuotedStr('False')+')');
DataModule1.QCekInboxReg.ExecSQL;
    end;
   end;
```

```
  except
    exit;
  end;
  try
    for J:=1 to jmlNumber - 1 do
    begin
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT * FROM TBL_EMAIL_INBOX
WHERE VCH_EMAIL_ADDRESS =
'+QuotedStr(GET_VCH_EMAIL_ADDRES
S)+
'AND NMB_IDX = '+IntToStr(J)+' AND
VCH_STATUS = '+QuotedStr('EMAIL
DOWNLOADED OTOMATIS')+
'AND VCH_PROCESS =
'+QuotedStr('False'));
DataModule1.QCekInboxReg.ExecSQL;
      DataModule1.QCekInboxReg.Open;
      txtPesan:='From:
'+DataModule1.QCekInboxReg.FieldByNa
me('VCH_FROM_NAME').AsString+'
<'+DataModule1.QCekInboxReg.FieldByNa
me('VCH_FROM_EMAIL_ADDRESS').As
String+'> - '+
            'Subj:
'+DataModule1.QCekInboxReg.FieldByNa
me('VCH_SUBJECT').AsString+' - '+
            'Body:
'+DataModule1.QCekInboxReg.FieldByNa
me('VCH_BODY').AsString;
txtNomor:=GET_VCH_PHONE_NUMBER
;
      if
(DataModule1.QCekInboxReg.FieldByNam
e('VCH_FROM_EMAIL_ADDRESS').AsSt
ring<>'') then
      begin
        KirimSMSOtomatisCekInbox(Sender
as TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT OTOMATIS',
txtNomor);
DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('UP
DATE TBL_EMAIL_INBOX SET
VCH_PROCESS = '+QuotedStr('True')+
'WHERE VCH_EMAIL_ADDRESS =
'+QuotedStr(GET_VCH_EMAIL_ADDRES
S)+
'AND NMB_IDX = '+IntToStr(J)+' AND
VCH_STATUS = '+QuotedStr('EMAIL
DOWNLOADED OTOMATIS')+
'AND VCH_PROCESS =
'+QuotedStr('False'));
DataModule1.QCekInboxReg.ExecSQL;
      end;
    end;
  except
    exit;
  end;
 end;
end;

procedure
TF_Utama.CekInboxUnreg(Sender:
TObject);
var
 I, getLength: Integer;
 UNREG, SHORTCODE: String;
begin
 DataModule1.QCekInboxReg.SQL.Clear;
DataModule1.QCekInboxReg.SQL.Add('SE
LECT * FROM TBL_SMS_INBOX
WHERE VCH_ISI_SMS LIKE
'+QuotedStr('Unreg%')+' AND
VCH_PROCESS = '+QuotedStr('False'));
 DataModule1.QCekInboxReg.ExecSQL;
 DataModule1.QCekInboxReg.Open;
txtTokenSource:=DataModule1.QCekInbox
Reg.FieldByName('VCH_ISI_SMS').AsStri
ng;
CekProses4:=DataModule1.QCekInboxReg.
FieldByName('VCH_PROCESS').AsString;
CekNomorInbox:=DataModule1.QCekInbox
Reg.FieldByName('VCH_PHONE_NUMB
ER').AsString;
 if CekProses4='False' then
 begin
 for I:=1 to 2 do
 begin
  CreateTokens('#', txtTokenSource);
  getLength:=Length(txtTokenResult);
txtTokenSource:=Copy(txtTokenSource,1,L
ength(txtTokenSource)-(getLength+1));
  CekReg[I]:=txtTokenResult;
 end;
 UNREG:=Uppercase(CekReg[2]);
 SHORTCODE:=Uppercase(CekReg[1]);
 if (UNREG='UNREG') and
(CekProses4='False') and
(SHORTCODE='SMSEMAIL') then
 begin
  txtPesan:='Terimakasih telah
menggunakan layanan kami, sekarang anda
sudah tidak terdaftar!!!';
  txtNomor:=CekNomorInbox;
```

```
  KirimSMSOtomatisCekInbox(Sender as
TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT UNREG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_SMS_INBOX SET
VCH_PROCESS = '+QuotedStr('True')+
'WHERE VCH_ISI_SMS LIKE
'+QuotedStr('Unreg%')+' AND
VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor));
DataModule1.QUpdateInboxReg.ExecSQL;
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_PELANGGAN SET
VCH_STATUS =
'+QuotedStr('UNREGISTERED')+
'WHERE VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor)+' AND
VCH_STATUS = '+
QuotedStr('REGISTERED'));
DataModule1.QUpdateInboxReg.ExecSQL;
  txtNomor:='';
  txtPesan:='';
 end else
 if (UNREG<>'UNREG') and
(CekProses4='False') and
(SHORTCODE='SMSEMAIL') then
 begin
  txtPesan:='Maaf, format yang anda
kirimkan salah. Ketik:
UNREG#SMSEMAIL#. Cth:
unreg#smsemail#';
  txtNomor:=CekNomorInbox;
  KirimSMSOtomatisCekInbox(Sender as
TObject);
MasukkanKeSent('gateway081009@gmail.c
om', DateTimetoStr(Now), '0', '0', '0',
txtPesan, 'SMS SENT UNREG', txtNomor);
DataModule1.QUpdateInboxReg.SQL.Clear
;
DataModule1.QUpdateInboxReg.SQL.Add('
UPDATE TBL_SMS_INBOX SET
VCH_PROCESS = '+QuotedStr('True')+
'WHERE VCH_ISI_SMS LIKE
'+QuotedStr('Unreg%')+' AND
VCH_PHONE_NUMBER =
'+QuotedStr(txtNomor));
DataModule1.QUpdateInboxReg.ExecSQL;
 end;
 end;
end;

procedure
TF_Utama.bAttachFileClick(Sender:
TObject);
begin
 Suara2(Sender as TObject);
 if (OpenDialog1.Execute) then
 begin
EmailAttachment:=OpenDialog1.FileName;
 end;
end;

procedure
TF_Utama.SkinChange(Sender:TObject);
var
 A: Integer;
 S: String;
begin
 Suara2(Sender as TObject);
 if not (Sender is TMenuItem) then exit;
 for A:=0 to
TMenuItem(Sender).Parent.Count-1 do
TMenuItem(Sender).Parent.Items[A].Check
ed:=False;
 TMenuItem(Sender).Checked:=True;
 if TMenuItem(Sender).Tag<0 then exit;
S:=Copy(TMenuItem(Sender).Caption,1,Le
ngth(TMenuItem(Sender).Caption));
 sSkinManager1.SkinName:=S;
 Ini.WriteString('WARNA','LATAR',S);
end;

procedure
TF_Utama.BackgroundChange(Sender:TObj
ect);
var
 A: Integer;
 S: String;
begin
 Suara2(Sender as TObject);
 Image1.Visible:=False;
 if not (Sender is TMenuItem) then exit;
 for A:=0 to
TMenuItem(Sender).Parent.Count-1 do
TMenuItem(Sender).Parent.Items[A].Check
ed:=False;
 TMenuItem(Sender).Checked:=True;
 if TMenuItem(Sender).Tag<0 then exit;
S:=Copy(TMenuItem(Sender).Caption,1,Le
ngth(TMenuItem(Sender).Caption));
Image1.Picture.LoadFromFile(S2+S+'.jpg');
 Image1.Visible:=True;
Ini.WriteString('BACKGROUND','LATAR',
S);
```

```
end;

procedure
TF_Utama.UserAccount1Click(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  pUserAccount.Align:=alClient;
  setVisiblePanel('pUserAccount',True);
end;

procedure    TF_Utama.Data1Click(Sender:
TObject);
begin
  Suara1(Sender as TObject);
  case (sender as TComponent).Tag of
  1: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
VCH_FULLNAME');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataUser.Align:=alClient;
      setVisiblePanel('pViewDataUser',True);
    end;
  2: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
DTM_LOGIN');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataLogin.Align:=alClient;
setVisiblePanel('pViewDataLogin',True);
    end;
  3: begin
      DataModule1.QViewData.SQL.Clear;

DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY ID');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataPelanggan.Align:=alClient;
setVisiblePanel('pViewDataPelanggan',True
);
    end;
  4: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PHONEBOOK ORDER
BY ID');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataPhonebook.Align:=alClient;
setVisiblePanel('pViewDataPhonebook',Tru
e);
    end;
  5: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY ID');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataInboxSMS.Align:=alClient;
setVisiblePanel('pViewDataInboxSMS',True
);
    end;
  6: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_EMAIL_INBOX
ORDER BY VCH_EMAIL_ADDRESS');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataInboxEmail.Align:=alClient;
setVisiblePanel('pViewDataInboxEmail',Tru
e);
    end;
  7: begin
      DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_PHONE_NUMBER');
      DataModule1.QViewData.ExecSQL;
      DataModule1.QViewData.Open;
      pViewDataSent.Align:=alClient;
      setVisiblePanel('pViewDataSent',True);
    end;
  end;
end;

procedure
TF_Utama.Cascade1Click(Sender:
TObject);
var
  I: integer;
begin
  Suara1(Sender as TObject);
  case (Sender as TComponent).Tag of
  1: Cascade;
  2: Tile;
  3: For i:=mdichildcount-1 DownTo 0 Do
MdiChildren[i].Close;
  end;
end;
```

```
procedure
TF_Utama.bInsertUserAccountClick(Sender
: TObject);
begin
 DataModule1.T_USERACCOUNT.Insert;
 bInsertUserAccount.Enabled :=False;
 bEditUserAccount.Enabled   :=False;
 bDeleteUserAccount.Enabled :=False;
 bOKUserAccount.Enabled     :=True;
 bCancelUserAccount.Enabled :=True;
 bRefreshUserAccount.Enabled :=False;
 eVCH_FULLNAME.ReadOnly     :=False;
 eVCH_USER_ID.ReadOnly      :=False;
 eVCH_PASSWORD.ReadOnly
:=False;
 eVCH_ROLE.ReadOnly         :=False;
 eVCH_GENDER.ReadOnly       :=False;
eDTM_DATE_REGISTER.ReadOnly:=Fals
e;
 eDTM_DATE_REGISTER.Text
:=DateTimeToStr(Now);
 eVCH_EMAIL_ADDRESS.ReadOnly
:=False;
 eVCH_PHONE_NUMBER.ReadOnly
:=False;
 eVCH_HOME_ADDRESS.ReadOnly
:=False;
 eVCH_FULLNAME.SetFocus;
 F_Utama.Suara2(Sender as TObject);
end;

procedure
TF_Utama.bEditUserAccountClick(Sender:
TObject);
begin
 DataModule1.T_USERACCOUNT.Edit;
 bInsertUserAccount.Enabled :=False;
 bEditUserAccount.Enabled   :=False;
 bDeleteUserAccount.Enabled :=False;
 bOKUserAccount.Enabled     :=True;
 bCancelUserAccount.Enabled :=True;
 bRefreshUserAccount.Enabled :=False;
 eVCH_FULLNAME.ReadOnly     :=False;
 eVCH_USER_ID.ReadOnly      :=False;
 eVCH_PASSWORD.ReadOnly
:=False;
 eVCH_ROLE.ReadOnly         :=False;
 eVCH_GENDER.ReadOnly       :=False;
eDTM_DATE_REGISTER.ReadOnly:=Fals
e;
 eVCH_EMAIL_ADDRESS.ReadOnly
:=False;
 eVCH_PHONE_NUMBER.ReadOnly
:=False;
 eVCH_HOME_ADDRESS.ReadOnly
:=False;
 eVCH_FULLNAME.SetFocus;
 F_Utama.Suara2(Sender as TObject);
end;

procedure
TF_Utama.bDeleteUserAccountClick(Sende
r: TObject);
begin
 bInsertUserAccount.Enabled :=False;
 bEditUserAccount.Enabled   :=False;
 bDeleteUserAccount.Enabled :=False;
 bOKUserAccount.Enabled     :=False;
 bCancelUserAccount.Enabled :=False;
 bRefreshUserAccount.Enabled :=False;
 if MessageDlg('Apa anda yakin ingin
menghapus record ini ???',
   mtConfirmation, [mbOK, mbCancel], 0)
= mrOK then
 begin
DataModule1.T_USERACCOUNT.Delete;
   bInsertUserAccount.Enabled :=True;
   bEditUserAccount.Enabled   :=True;
   bDeleteUserAccount.Enabled :=True;
   bOKUserAccount.Enabled     :=False;
   bCancelUserAccount.Enabled :=False;
   bRefreshUserAccount.Enabled :=True;
 end else
 begin
   bInsertUserAccount.Enabled :=True;
   bEditUserAccount.Enabled   :=True;
   bDeleteUserAccount.Enabled :=True;
   bOKUserAccount.Enabled     :=False;
   bCancelUserAccount.Enabled :=False;
   bRefreshUserAccount.Enabled :=True;
 end;
 F_Utama.Suara2(Sender as TObject);
end;

procedure
TF_Utama.bOKUserAccountClick(Sender:
TObject);
var
 cekPass: String;
begin
cekPass:=Copy(eVCH_PASSWORD.Text,L
ength(eVCH_PASSWORD.Text)-1,2);
 if cekPass<>'==' then
eVCH_PASSWORD.Text:=F_Password.En
crypt(eVCH_PASSWORD.Text)+'==';
 DataModule1.T_USERACCOUNT.Post;
 bInsertUserAccount.Enabled :=True;
 bEditUserAccount.Enabled   :=True;
 bDeleteUserAccount.Enabled :=True;
```

```
 bOKUserAccount.Enabled      :=False;
 bCancelUserAccount.Enabled  :=False;
 bRefreshUserAccount.Enabled :=True;
 eVCH_FULLNAME.ReadOnly      :=True;
 eVCH_USER_ID.ReadOnly       :=True;
 eVCH_PASSWORD.ReadOnly      :=True;
 eVCH_ROLE.ReadOnly          :=True;
 eVCH_GENDER.ReadOnly        :=True;
eDTM_DATE_REGISTER.ReadOnly:=Tru
e;
 eVCH_EMAIL_ADDRESS.ReadOnly
:=True;
 eVCH_PHONE_NUMBER.ReadOnly
:=True;
 eVCH_HOME_ADDRESS.ReadOnly
:=True;
 eVCH_FULLNAME.SetFocus;
 F_Utama.Suara2(Sender as TObject);
end;

procedure
TF_Utama.bCancelUserAccountClick(Send
er: TObject);
begin
 DataModule1.T_USERACCOUNT.Cancel;
 bInsertUserAccount.Enabled  :=True;
 bEditUserAccount.Enabled    :=True;
 bDeleteUserAccount.Enabled  :=True;
 bOKUserAccount.Enabled      :=False;
 bCancelUserAccount.Enabled  :=False;
 bRefreshUserAccount.Enabled :=True;
 eVCH_FULLNAME.ReadOnly      :=True;
 eVCH_USER_ID.ReadOnly       :=True;
 eVCH_PASSWORD.ReadOnly      :=True;
 eVCH_ROLE.ReadOnly          :=True;
 eVCH_GENDER.ReadOnly        :=True;
eDTM_DATE_REGISTER.ReadOnly:=Tru
e;
 eVCH_EMAIL_ADDRESS.ReadOnly
:=True;
 eVCH_PHONE_NUMBER.ReadOnly
:=True;
 eVCH_HOME_ADDRESS.ReadOnly
:=True;
 eVCH_FULLNAME.SetFocus;
 Suara2(Sender as TObject);
end;

procedure
TF_Utama.bRefreshUserAccountClick(Send
er: TObject);
begin
DataModule1.T_USERACCOUNT.Active:=
false;
DataModule1.T_USERACCOUNT.Active:=
true;
DataModule1.T_USERACCOUNT.Refresh;
 Suara2(Sender as TObject);
end;

procedure TF_Utama.Browse1Click(Sender:
TObject);
var
 FileSource, CopyFileKe, S: String;
 A,B: Integer;
begin
 if (OpenDialog1.Execute) then
 begin
  FileSource:=OpenDialog1.FileName+'\';
  CreateTokens('\',          FileSource);
CopyFileKe:=ExtractFilePath(Application.E
xeName)+'Gambar\Background\'+txtToken
Result;
  Image1.Visible:=False;
Image1.Picture.LoadFromFile(OpenDialog1
.FileName);
  Image1.Visible:=True;
  Image1.Picture.SaveToFile(CopyFileKe);
FileSource:=Copy(txtTokenResult,1,Length(
txtTokenResult)-4);
  Ini.WriteString('BACKGROUND',
'LATAR', FileSource);
  A:=0;
  repeat
   Inc(A);
   if          A=1          then
B:=FindFirst(S+'*.jpg',faAnyFile,SR) else
   B:=FindNext(SR);
   if (B=0) and (SR.Name<>'.') and
(SR.Name<>'..') then
   begin
S:=Copy(SR.Name,1,Length(SR.Name)-4);
TMnBackground:=TMenuItem.Create(Back
ground1);
    TMnBackground.Caption:=S;
    TMnBackground.Tag:=A;
    Background1.Add(TMnBackground);
TMnBackground.OnClick:=BackgroundCha
nge;
   end;
  until B<>0;
  FindClose(SR);
 end;
end;

procedure
TF_Utama.vDataUserOrderByClick(Sender:
TObject);
begin
```

```
 case vDataUserOrderBy.ItemIndex of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
VCH_FULLNAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
VCH_USER_ID');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
VCH_ROLE');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
VCH_GENDER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_USER ORDER BY
DTM_DATE_REGISTER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;

procedure
TF_Utama.bPrintViewDataUserClick(Sende
r: TObject);
begin
 case (sender as TComponent).Tag of
 0: QR_ViewDataUser.Preview;
 1: QR_ViewDataLogin.Preview;
 2: QR_ViewDataPelanggan.Preview;
 3: QR_ViewDataPhonebook.Preview;
 4: QR_ViewDataInboxSMS.Preview;
 5: QR_ViewDataInboxEmail.Preview;
 6: QR_ViewDataSent.Preview;
 end;
end;

procedure
TF_Utama.vDataLoginOrderByClick(Sende
r: TObject);
begin
 case vDataLoginOrderBy.ItemIndex of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
DTM_LOGIN');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
DTM_LOGOUT');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
VCH_FULLNAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
VCH_USER_ID');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
VCH_ROLE');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 5: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_LOGIN ORDER BY
VCH_STATUS');
    DataModule1.QViewData.ExecSQL;
```

```
    DataModule1.QViewData.Open;
   end;
  end;
end;

procedure
TF_Utama.vDataPelangganOrderByClick(S
ender: TObject);
begin
 case vDataPelangganOrderBy.ItemIndex of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY ID');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY DTM_DATE_REGISTER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_FULLNAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_GENDER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_HOME_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 5: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_PHONE_NUMBER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 6: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_EMAIL_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 7: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PELANGGAN ORDER
BY VCH_STATUS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;

procedure
TF_Utama.vDataPhonebookOrderByClick(S
ender: TObject);
begin
 case   vDataPhonebookOrderBy.ItemIndex
of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PHONEBOOK ORDER
BY ID');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PHONEBOOK ORDER
BY VCH_FULLNAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_PHONEBOOK ORDER
BY VCH_PHONE_NUMBER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;
```

```
procedure
TF_Utama.vDataInboxSMSOrderByClick(S
ender: TObject);
begin
 case   vDataInboxSMSOrderBy.ItemIndex
of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY ID');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY VCH_PHONE_NUMBER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY VCH_DATE');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY VCH_ISI_SMS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY VCH_JENIS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 5: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SMS_INBOX ORDER
BY VCH_PROCESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;

procedure
TF_Utama.vDataInboxEmailOrderByClick(
Sender: TObject);
begin
 case  vDataInboxEmailOrderBy.ItemIndex
of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER BY VCH_EMAIL_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER BY DTM_EMAIL_DATE');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER BY VCH_FROM_NAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER                          BY
VCH_FROM_EMAIL_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER BY VCH_SUBJECT');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 5: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT   *   FROM   TBL_EMAIL_INBOX
ORDER BY VCH_BODY');
    DataModule1.QViewData.ExecSQL;
```

```
    DataModule1.QViewData.Open;
   end;
 6: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_EMAIL_INBOX
ORDER BY VCH_STATUS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 7: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_EMAIL_INBOX
ORDER BY VCH_PROCESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;

procedure
TF_Utama.vDataSentOrderByClick(Sender:
TObject);
begin
 case vDataSentOrderBy.ItemIndex of
 0: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_PHONE_NUMBER');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 1: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_EMAIL_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 2: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
DTM_EMAIL_DATE');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 3: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_TO_NAME');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 4: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_TO_EMAIL_ADDRESS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 5: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_SUBJECT');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 6: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_BODY');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 7: begin
    DataModule1.QViewData.SQL.Clear;
DataModule1.QViewData.SQL.Add('SELE
CT * FROM TBL_SENT ORDER BY
VCH_STATUS');
    DataModule1.QViewData.ExecSQL;
    DataModule1.QViewData.Open;
   end;
 end;
end;

end.
```