

**PENGEMBANGAN SISTEM INFORMASI LAYANAN 108  
PSTN ( PUBLIC SWITCH TELEPHONE NETWORK )  
MENGUNAKAN SMS GATEWAY**

**SKRIPSI**



**Disusun Oleh :  
PEDRICO KARLINOVIANDY DA SILVA  
NIM 05.12.529**

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

3818

EXHIBIT REPORTER MOUNTAIN VIEW  
SYSTEMS REPORTER MOUNTAIN  
VIEW REPORTER MOUNTAIN VIEW  
VIEW REPORTER MOUNTAIN VIEW

1992 04 18 000  
REPORTER MOUNTAIN VIEW  
REPORTER MOUNTAIN VIEW

REPORTER MOUNTAIN VIEW  
REPORTER MOUNTAIN VIEW  
REPORTER MOUNTAIN VIEW

3818

REPORTER MOUNTAIN VIEW  
REPORTER MOUNTAIN VIEW  
REPORTER MOUNTAIN VIEW

## LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM INFORMASI LAYANAN 108  
PSTN (PUBLIC SWITCH TELEPHONE NETWORK)  
MENGUNAKAN SMS GATEWAY

### SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)*

Disusun Oleh :

**PEDRICO KARLINOVIANDY DA SILVA**

NIM : 05.12.529

Diperiksa dan Disetujui

Mengetahui

Ketua Jurusan Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189



Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT

NIP Y. 1018800189

Dosen Pembimbing II

I Komang Somawirata, ST. MT

NIP.Y.1030100361

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

2010

**PENGEMBANGAN SISTEM INFORMASI LAYANAN 108 PSTN  
(PUBLIC SWITCH TELEPHONE NETWORK)  
MENGUNAKAN SMS GATEWAY**

**Pedrico Karlinoviandy Da Silva**  
Email : [ped.karlino@gmail.com](mailto:ped.karlino@gmail.com)

**Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer dan Informatika**

**Fakultas Teknologi Industri, Institut Teknologi Nasional Malang  
Jl. Raya Karanglo Km 2 Malang  
Email : [itn@itn.ac.id](mailto:itn@itn.ac.id)**

*Abstrak*

*Teknologi hadir untuk memberikan kemudahan-kemudahan terhadap suatu masalah yang dihadapi. Keberadaan sistem informasi layanan 108 pada PT. TELKOM menjadi salah satu faktor penting untuk meningkatkan pelayanannya dan kini telah menjadi salah satu standar mutu. Apalagi didukung dengan fasilitas SMS (Short Message Service) untuk melayani kebutuhan akan informasi yang diinginkan para pelanggan.*

*Penerapan aplikasi ini adalah untuk menyediakan dan memberikan informasi untuk para pelanggan. Informasi yang bisa didapatkan oleh pelanggan antara lain informasi mengenai nomor telephone pelanggan beserta nama pemiliknya, informasi kode area, informasi kode negara dan informasi nomor darurat yang bersifat umum.*

*Dengan keberadaan sistem informasi dengan fasilitas SMS ini diharapkan dapat memberikan informasi yang cepat tepat akurat dan efisien.*

**Kata Kunci :** Sistem informasi layanan 108, SMS

*Abstract*

*Technology exists to provide easiness of a problem to be faced. The presence information system 108 service at PT. TELKOM be one important factor to improve its service and has now become one of the basic quality standards. Moreover, the facility supported by SMS (Short Message Service) to serve the needs of our customers desired information.*

*Application of this application is to provide and providing information to the customer. The information can be obtained by customers, among others, information about customers and their telephone number owner's name, area code information, information on the country code and the emergency number information of a general nature.*

*With the existence of information systems with the SMS facility is expected to provide accurate information fast, accurate and efficient.*

**Keyword :** Service Information System 108, SMS

## **KATA PENGANTAR**

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa yang dengan segala Kasih dan Anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul :

### **PENGEMBANGAN SISTEM INFORMASI LAYANAN 108 PSTN ( PUBLIC SWITCH TELEPHONE NETWORK ) MENGUNAKAN SMS GATEWAY**

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE., selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahyono, MT., selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT., selaku Ketua Jurusan Teknik Elektro S – 1 ITN Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT., selaku Dosen Pembimbing I
5. Bapak I Komang Somawirata, ST. MT., selaku Dosen Pembimbing II.
6. Papa , Mama dan semua anggota keluargaku tercinta yang selalu memberikan semangat, dukungan dan doa untuk dapat meraih cita – cita.
7. Kakakku Mas Doni yang selalu memberikan semangat dan solusi, terutama lewat udara.
8. Teman - teman ITN Angkatan 2005 khususnya Danang, Viktor, Happy, Thariq, Pandu, Putra, Eko yang memberikam support dan dukungan.
9. Teman – teman Telkom khususnya It.6 yang selalu memberikan semangat ketika penulis membutuhkan dukungan mental.
10. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis juga berharap agar laporan tugas akhir ini dapat dikembangkan lebih lanjut sehingga dapat benar-benar digunakan sebaik-baiknya untuk mendukung

perkembangan ilmu pengetahuan.. Oleh sebab itu kritik dan saran yang bersifat membangun sangat penulis harapkan.

Akhir kata, penulis mohon maaf kepada semua pihak apabila selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat baik bagi pembaca maupun penulis sendiri. Amin.

Malang, Agustus 2010

Penulis

## DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	i
ABSTRAK .....	ii
KATA PENGANTAR .....	iii
DAFTAR ISI .....	v
DAFTAR GAMBAR .....	viii
DAFTAR TABEL .....	xi
<b>BAB I : PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan.....	3
<b>BAB II : LANDASAN TEORI .....</b>	<b>4</b>
2.1 SMS Gateway.....	4
2.2 BORLAND DELPHY 7 .....	5
2.3 MY SQL.....	6
2.4 Data Flow Diagram (DFD) .....	8
2.5 AT Command .....	9
2.6 PDU ( Protocol Data Unit ) .....	12
2.7 Metode Penelitian .....	15
<b>BAB III : PERANCANGAN SISTEM.....</b>	<b>17</b>
3.1 Perancangan Aplikasi .....	17
3.1.1 Aplikasi SMS Gateway .....	17

3.1.2 Penggunaan Sistem.....	18
3.1.3 Spesifikasi Sistem.....	18
3.2 Perancangan Sistem .....	18
3.2.1 Alur Proses Aplikasi.....	18
3.2.1.1 <i>Flowchart</i> SMS.....	19
3.2.1.2 <i>Flowchart</i> Login Admin .....	20
3.2.1.3 <i>Flowchart</i> Memasukkan Data .....	21
3.2.1.4 <i>Flowchart</i> Merubah Data.....	21
3.2.2 <i>Data Flow Diagram</i> (DFD).....	22
3.2.2.1 <i>Contex Diagram</i> .....	23
3.2.2.2 DFD level 1 .....	23
3.2.3 <i>Desain Basisdata</i> .....	24
3.2.3.1 <i>Tabel data Pada Aplikasi Administrator</i> .....	24
3.2.4 <i>Desain AntarMuka Aplikasi</i> .....	26
3.2.4.1 <i>Desain Halaman Login</i> .....	26
3.2.4.2 <i>Desain Halaman Menu Utama Aplikasi</i> .....	26
3.2.4.3 <i>Desain List SMS Masuk (Inbox)</i> .....	27
3.2.4.4 <i>Desain List SMS Replay (Sent Item)</i> .....	27
3.2.4.5 <i>Desain Baca SMS Masuk</i> .....	27
3.2.4.6 <i>Desain Pengiriman SMS</i> .....	28
3.2.4.7 <i>Desain Data Pelanggan</i> .....	28
3.2.4.8 <i>Desain Kode Area</i> .....	29
3.2.4.9 <i>Desain Kode Negara</i> .....	29
3.2.4.10 <i>Desain Data Admin</i> .....	30
3.2.4.11 <i>Desain Nomor Darurat</i> .....	30



<b>BAB IV : IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>31</b>
<b>4.1 Implementasi Sistem .....</b>	<b>31</b>
4.1.1 Koneksi Basis Data.....	31
4.1.2 Pengoperasian Basis Data.....	31
4.1.3 Kebutuhan Sistem.....	31
4.1.3.1 Perangkat Keras .....	32
4.1.3.2 Perangkat Lunak .....	32
4.1.4 Konfigurasi Basis Data .....	32
4.2 Pengujian Hasil.....	33
<b>BAB V : PENUTUP .....</b>	<b>67</b>
5.1 Kesimpulan .....	67
5.2 Saran .....	67
<b>DAFTAR PUSTAKA .....</b>	<b>68</b>
<b>LAMPIRAN .....</b>	<b>69</b>

## DAFTAR GAMBAR

Gambar 3.1 Desain Sistem .....	18
Gambar 3.2 <i>Flowchart</i> Alur SMS .....	19
Gambar 3.3 <i>Flowchart</i> Login Administrator.....	20
Gambar 3.4 <i>Flowchart</i> Memasukkan Data.....	21
Gambar 3.5 <i>Flowchart</i> Merubah Data.....	22
Gambar 3.6 <i>DFD Level 0</i> .....	23
Gambar 3.7 <i>DFD Level 1</i> .....	23
Gambar 3.8 Desain Halaman Login .....	26
Gambar 3.9 Desain Menu Halaman Aplikasi SMS Gateway.....	27
Gambar 3.10 Desain List SMS Masuk .....	27
Gambar 3.11 Desain List SMS Replay.....	27
Gambar 3.12 Desain Baca SMS Masuk .....	28
Gambar 3.13 Desain pengiriman SMS .....	28
Gambar 3.14 Desain Data Pelanggan .....	28
Gambar 3.15 Desain Kode Area.....	29
Gambar 3.16 Desain Kode Negara .....	29
Gambar 3.17 Desain Data Admin.....	30
Gambar 3.18 Desain Nomor Darurat.....	30
Gambar 4.1 Membuat <i>database</i> baru.....	33
Gambar 4.2 Pengisian <i>database</i> .....	33
Gambar 4.3 Login Administrator .....	34
Gambar 4.4 Tampilan Menu Utama .....	34
Gambar 4.5 Data Pelanggan .....	35
Gambar 4.6 Data Kode Area .....	36
Gambar 4.7 Data Kode Negara.....	37
Gambar 4.8 Data Nomor Darurat .....	37
Gambar 4.9 Data Administrator .....	38
Gambar 4.10 Menu Aplikasi <i>Request</i> .....	38
Gambar 4.11 Menu Aplikasi <i>Replay</i> .....	39

**DAFTAR TABEL**

Tabel 2.1 Perintah – Perintah AT Command.....	11
Tabel 2.2 PDU untuk SMS Kirim .....	12
Tabel 2.2 PDU untuk SMS Terima .....	13
Tabel 3.1 Tabel Data Pelanggan.....	24
Tabel 3.2 Tabel Data Admin .....	24
Tabel 3.3 Tabel Kode Area .....	25
Tabel 3.4 Tabel Kode Negara .....	21
Tabel 3.5 Tabel Nomor Darurat .....	26

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan alat komunikasi saat ini, handphone bukanlah suatu barang yang mewah lagi dan sudah termasuk kebutuhan primer. Hampir setiap individu membutuhkan handphone sebagai alat komunikasi yang mudah dan cepat. Selain sebagai alat komunikasi handphone juga bisa digunakan sebagai alat untuk mendapatkan informasi yang di butuhkan. Sebagai contoh untuk mendapatkan informasi nomor telepon rumah beserta nama pemiliknya, mengetahui kode area kota, kode negara dan mengetahui nomor-nomor darurat yang bersifat umum.

Jadi di saat kita hanya mengetahui alamat rumah seseorang dan ingin mengetahui nomor teleponnya, kita tidak perlu menghubungi layanan 108 lewat telephone. Pada saat ini aplikasi pada handphone yang sering digunakan adalah aplikasi sms karena tarif yang relatif murah. Teknologi sms merupakan teknologi selular digital dalam penyampaian pesan singkat cepat dan efektif. Untuk mendapatkan informasi sms lebih sering digunakan karena singkat dan efektif.

Dalam menanggapi kasus - kasus di ini munculah ide untuk menciptakan aplikasi sms gateway dengan tujuan untuk memudahkan mendapatkan informasi nomor telepon beserta nama pemiliknya, mengetahui kode area, kode negara dan mengetahui nomor-nomor darurat yang bersifat umum .

Dalam pembuatan aplikasi sms gateway ini, penulis menggunakan bahasa pemrograman Delphi. Sampai saat ini Delphi 7.0 merupakan salah satu bahasa pemrograman yang sangat populer didunia. Selai didukung dengan IDE (Integrated Development Environment) atau lingkungan pengembangan yang terintegrasi, yang sangat mudah digunakan. Selain menggunakan bahasa pemrograman Delphi 7.0, aplikasi ini menggunakan database yaitu dengan MySQL. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang software dan aplikasi baik di platform web maupun desktop. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil namu perusahaan seperti Yahoo pun juga menggunakannya. Dengan adanya aplikasi SMS Gateway ini diharapkan kita tidak perlu lagi bingung untuk mendapatkan berbagai informasi dengan mudah dan cepat.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas maka timbulah suatu permasalahan bagaimana merancang suatu aplikasi untuk system informasi yang dapat memudahkan untuk mendapatkan nomor telepon beserta nama pemiliknya, mengetahui kode area,kode negara dan mengetahui nomor-nomor darurat yang bersifat umum.

## **1.3 Tujuan**

Tujuan dari perancangan aplikasi SMS Gateway ini adalah sebagai sarana pendukung bagi pelanggan layanan 108 dalam mendapatkan informasi melalui media SMS.

## **1.4 Batasan Masalah**

Agar permasalahan yang di angkat tidak meluas dan lebih terarah sesuai dengan tujuan, maka :

1. Pada program aplikasi sms gateway ini, hanya diperuntukkan untuk mendapatkan nomor telepon beserta nama pemiliknya dengan input alamat, permintaan kode area,kode negara dan permintaan nomor-nomor darurat yang bersifat umum.
2. Tidak menerima permintaan jika diketahui hanya nomor telepon pelanggan saja karena merupakan kode etik layanan 108 itu sendiri atas peraturan PT. Telkom.
3. Aplikasi ini menggunakan input alamat hanya dengan rincian nama jalan,nomor gang dan nomor rumah saja.
4. Tidak membahas nomor telephone pelanggan yang di privasi.
5. Tidak membahas masalah sistem jaringan komputer yang digunakan pada PT. Telkom
6. Tidak membahas masalah sistem keamanan yang di gunakan.
7. Program dibuat dengan menggunakan bahasa pemrograman Delphi 7.0, MySQL sebagai databasenya dan satu buah handphone yang di gunakan untuk menerima dan mengirimkan sms.
8. Tidak membahas masalah jaringan GSM.

### **1.5 Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut:

#### **BAB I : PENDAHULUAN**

Bab ini berisi tentang latar belakang, tujuan, permasalahan, batasan masalah, dan sistematika pembahasan dari skripsi ini.

#### **BAB II : LANDASAN TEORI**

Bab ini berupa penjelasan tentang teori – teori yang mendukung dalam perancangan sistem yang meliputi, teori SMS

#### **BAB III : PERANCANGAN DAN PEMBUATAN SISTEM**

Bab ini berisi tentang membangun SMS Gateway serta pembuatan database

#### **BAB IV : HASIL DAN ANALISA**

Bab ini berisi pembahasan hasil pengujian serta pembahasan dari hasil analisa mengenai cara kerja dari sistem.

#### **BAB V : PENUTUP**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan pada skripsi ini.

## BAB II

### LANDASAN TEORI

#### 2.1 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*. Pada prinsipnya, *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. Secara khusus, sistem ini akan memiliki fungsi-fungsi sebagai berikut:

##### 1. *Message Management* dan *Delivery*

- a. Pengaturan pesan yang meliputi manajemen prioritas pesan, manajemen pengiriman pesan, dan manajemen antrian.
- b. Pesan yang dilalukan harus sedapat mungkin *fail safe*. Artinya, jika terdapat gangguan pada jaringan telekomunikasi, maka sistem secara otomatis akan mengirim ulang pesan tersebut.

##### 2. 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. Call Center dan SMS Pengaduan, misalnya Polisi, PLN, PDAM, 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. Beberapa kemampuan *SMS Gateway*, yaitu untuk:

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.

## **2.2 BORLAND DELPHI 7.0**

Borland Delphi 7 adalah paket bahasa pemrograman yang bekerja dalam system operasi Windows, yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Beberapa jenis aplikasi yang dapat kita buat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web.

Secara umum kemampuan Delphi adalah menyediakan komponen – komponen dan bahasa pemrograman yang andal, sehingga memungkinkan kita untuk membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah pemrogram dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman



tersebut dibagi dalam dua kelompok, yaitu object dan bahasa pemrograman. Secara ringkas, object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan – batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman Object Pascal yang sudah sangat dikenal dikalangan pemrograman professional. Gabungan dari object dan bahasa pemrograman ini disebut sebagai bahasa pemrograman berorientasi object atau Object Oriented Programming (OOP).

Khusus untuk pemrograman database, Delphi menyediakan object yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemrograman dalam merancang, membuat dan menyelesaikan aplikasi database yang diinginkan. Selain itu juga Delphi dapat menangani data dalam berbagai format database, misalnya format MS-Acces, Sybase, Oracle, FoxPro, Informix, DB2 dan lain – lain. Format database yang dianggap asli dari Delphi adalah Paradox dan dBase.

### 2.3 MYSQL

MySQL merupakan Relational Database Management Sistem (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structure Query Language).

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu system database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan dengan database server yang lainnya dalam query data.

Sebagai database yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL :

- **Portability.** MySQL dapat berjalan stabil pada berbagai sistem operasi di antaranya adalah seperti Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, HP-UX dan masih banyak lagi.
  - **Open Source.** MySQL didistribusikan secara open source (gratis), di bawah lisensi GPL.
  - **Multiuser.** MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan.
  - **Performance Tuning.** MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
  - **Column Types.** MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, year, set serta enum.
  - **Command dan Function.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.
  - **Security.** MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan user dengan system perizinan yang mendetail serta password terenkripsi.
  - **Stability dan Limits.** MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu, batas indeks yang dapat di tampung mencapai 32 indeks pada tiap tabelnya.
  - **Connectivity.** MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).
10. Localisation MySQL dapat mendeteksi pesan kesalahan (error code) pada client dengan

menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.

- Localisation. MySQL dapat mendeteksi pesan kesalahan (error code) pada client dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.
- Interface. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
- Client dan Tools. MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertai petunjuk online.

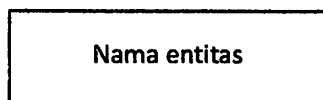
Struktur Tabel. MySQL memiliki struktur table yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

#### **2.4 Data Flow Diagram (DFD)**

Adalah suatu model diagram yang menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan proses serta penyimpanan dan aliran data. DFD sering digunakan untuk menggambarkan system yang telah ada atau system yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Ada 4 komponen dalam model DFD, yaitu (*Fatansyah, 1999*):

##### **1. Entitas luar (external entity)**

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Entitas luar merupakan kesatuan diluar lingkungan sistem yang dapat berupa orang, organisasi atau sistem yang lainnya berada dilingkungan luar yang akan memberikan input atau menerima output dari sistem.



## 2. Arus data

Arus data diberi symbol suatu panah. Arus data ini mengalir diantara proses, simpanan dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk system atau hasil dari proses system.

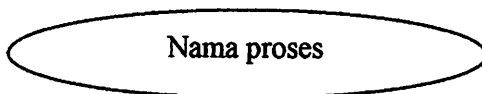


## 3. Proses

Proses menunjukkan transformasi dari masukan sampai keluaran. Suatu proses terjadi karena adanya arus data yang masuk dan hasil dari proses merupakan arus data yang mengalir.

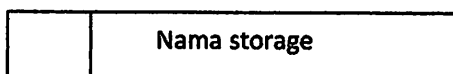
Beberapa hal yang harus dihindari dalam menggambarkan suatu proses adalah :

- Proses yang mempunyai masukan tapi tidak memiliki keluaran, kesalahan ini disebut black hole
- Proses yang menghasilkan keluaran tetapi tidak mendapatkan masukan, kesalahan ini disebut miracle
- Proses yang menghasilkan keluaran lengkap tetapi masukan dengan data terbatas atau masukan yang kurang lengkap, kesalahan ini disebut Gray hole.



## 4. Penyimpanan data (data storage)

Digunakan untuk menyimpan data hasil proses maupun menyediakan data untuk diproses pada waktu lain



## 2.5. AT Command

AT Command adalah perintah-perintah yang digunakan dalam komunikasi dengan serial port. Dengan AT Command kita dapat mengetahui vendor dari

Handphone yang digunakan, kekuatan sinyal, membaca pesan yang ada pada SIM Card, mengirim pesan, mendeteksi pesan SMS baru yang masuk secara otomatis, menghapus pesan pada SIM Card dan masih banyak lagi.

Dalam program SMS Server yang akan kita buat nanti, tidak semua perintah AT digunakan. Kita hanya menggunakan beberapa perintah AT yang ada hubungannya dengan sistem kerja dari program SMS Server. Adapun perintah yang akan digunakan adalah sebagai berikut :

AT Command	Keterangan
AT	Mengecek apakah Handphone telah terhubung
AT+CMGF	Untuk menetapkan format mode dari terminal
AT+CSCS	Untuk menetapkan jenis encoding
AT+CNMI	Untuk mendeteksi pesan SMS baru masuk secara otomatis
AT+CMGL	Membuka daftar SMS yang ada pada SIM Card
AT+CMGS	Mengirim pesan SMS
AT+CMGR	Membaca pesan SMS
AT+CMGD	Menghapus pesan SMS
ATE1	Mengatur ECHO
ATV1	Mengatur input dan output berupa naskah
AT+CGMI	Mengecek Merek HP
AT+CGMM	Mengecek Seri HP
AT+CGMR	Mengecek Versi Keluaran HP

AT Command	Keterangan
AT+CBC	Mengecek Baterai
AT+CSQ	Mengecek Kualitas Sinyal
AT+CCLK?	Mengecek Jam (waktu) pada HP
AT+CALM=<n>	Mengecek Suara/dering HP saat di Telepon (ada Telepon Masuk)  'n' adalah adalah angka yang menunjukkan jenis dering  0 = berdering  1 dan 2 = Silent (Diam)
AT^SCID	Mengecek ID SIM CARD
AT+CGSN	Mengecek Nomor IMEI
AT+CLIP=1	Menampilkan nomor telepon pemanggil
AT+CLCC	Menampilkan nomor telepon yang sedang memanggil
AT+COPN	Menampilkan Nama Sumua Operator di dunia
AT+COPS?	Menampilkan nama operator dari SIM yang digunakan
AT+CPBR=<n>	Membaca nomor telepon yang disimpan pada buku telepon (SIM CARD)  'n' adalah nomor urut penyimpanan
AT+CPMS=<md>	Mengatur Memori dari HP  'md' adalah memori yang digunakan  ME = Memori HP  SM = Memori SIM CARD

Tabel 2.1 Tabel Perintah - Perintah *AT Command*

## 2.6. PDU (Protocol Data Unit)

PDU (Protocol Data Unit) adalah protokol data dalam suatu SMS, berupa pasangan-pasangan karakter ASCII yang mencerminkan representasi angka heksadesimal dari informasi yang ada dalam suatu SMS, misalnya nomor pengirim, nomor tujuan, waktu pengiriman dan isi pesan SMS itu sendiri. PDU ini harus dipahami sebelum mengimplementasikannya ke dalam program di komputer/mikrokontroler.

Terus setelah kita memahami PDU, maka kita dapat membuat program untuk konvert PDU to Text dan Text to PDU. Bagaimana membuatnya ? disini akan saya ulas lengkap dengan code programnya dengan bahasa pemrograman Delphi.

### 2.6.1 PDU untuk SMS Kirim

Contoh:

```
06 91 2618010000 01 00 0C 91 261832547698 00 00 05 E8329BFD06
```

Beberapa pasangan di atas harus kita baca secara dibalik-balik, misalnya 26 adalah 62, dst. Arti dari data PDU di atas adalah sebagai berikut.

Bagian	Arti
06	Jumlah pasangan nomor SMS Center (6 pasang = 1 pasang jenis penomoran + 5 pasang nomor SMSC)
91	Jenis penomoran SMS Center (91 = menggunakan penomoran internasional)
2618010000	Nomor SMS Center (6281100000 = SMSC Telkomsel)
01	Tipe SMS (01 = SMS kirim)
00	Nomor Referensi SMS (otomatis jadi biarkan 00)

0C	Jumlah digit nomor tujuan dalam bilangan heksa (0C = 12 digit)
91	Jenis penomoran pengirim (91 = menggunakan penomoran internasional)
261832547698	Nomor tujuan pengiriman SMS (628123456789)
00	Bentuk SMS (00 = SMS teks)
00	Skema encoding (00 = skema 7 bit)
05	Jumlah karakter isi pesan dalam heksa (5 karakter)
C8329BFD06	Isi pesan dalam susunan encoding yang dipilih (E8329BFD06 jika diterjemahkan 7 bit -7 bit adalah 'Hello')

### 2.6.2 PDU untuk SMS Terima

Contoh:

```
06 91 2618010000 24 0C 91 261832547698 00 00 807060 510321 80 05
E8329BFD06
```

Beberapa pasangan di atas harus kita baca secara dibalik-balik, misalnya 26 adalah 62, dst. Arti dari data PDU di atas adalah sebagai berikut.

Bagian	Arti
06	Jumlah pasangan nomor SMS Center (6 pasang = 1 pasang jenis penomoran + 5 pasang nomor SMSC)



91	Jenis penomoran SMS Center (91 = menggunakan penomoran internasional)
2618010000	Nomor SMS Center (6281100000 = SMSC Telkomsel)
24	Tipe SMS (24 = SMS terima)
0C	Jumlah digit nomor pengirim dalam bilangan heksa (0C = 12 digit)
91	Jenis penomoran tujuan (91 = menggunakan penomoran internasional)
261832547698	Nomor pengirim SMS (628123456789)
00	Bentuk SMS (00 = SMS teks)
00	Skema encoding (00 = skema 7 bit)
807060	Tanggal pengiriman SMS yaitu 6/7/2008
510321	Jam pengiriman SMS yaitu 15:30:12
80	Standar waktu yang digunakan (GMT+7 Indonesia)
05	Jumlah karakter isi pesan dalam heksa (5 karakter)
C8329BFD06	Isi pesan dalam susunan encoding yang dipilih (E8329BFD06 jika diterjemahkan 7 bit -7 bit adalah 'Hello')

## 2.7 Metode Penelitian

Untuk membuat aplikasi *sms* ini, peneliti menggunakan metode *waterfall*<sup>1</sup> (air terjun) dalam metode pengembangan sistemnya. Adapun langkah-langkah yang dilakukan dalam pengembangan system adalah sebagai berikut :

### 1. Rekayasa Sistem

Pada tahap ini dilakukan beberapa pengumpulan kebutuhan yang diperlukan untuk membuat sistem, pengguna dan analisa sistem mendefinisikan konsep sistem serta spesifikasi sistem yang akan digunakan. Pada tahap ini hal yang terpenting adalah adanya keterlibatan antara kedua belah pihak.

### 2. Analisa Sistem

Tahap ini melakukan pengumpulan kebutuhan elemen-elemen ditingkat perangkat lunak. Dengan analisa harus dapat ditentukan kebutuhan data, fungsi, proses atau procedure yang diperlukan beserta unjuk kerjanya, dan *interfaces*. Hasil akhir tahap ini adalah spesifikasi kebutuhan perangkat lunak.

### 3. Desain

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidak sesuain desain antara user dan analyst. Tahap ini terdiri dari dua tahap, yaitu *preliminary design* dan *detailed design*.

Subtahap pertama menghasilkan rancangan yang bersifat global, sedangkan subtahap yang kedua menghasilkan rancangan detil sehingga semua kelas, tipe data, fungsi dan prosedurenya terdefinisi.

### 4. Coding

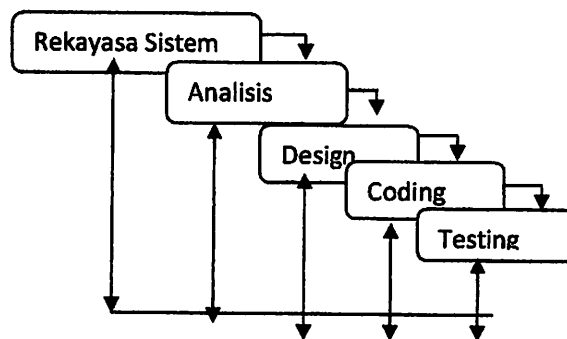
Pada tahap ini, apabila design dari sistem telah ditentukan, maka programmer mengembangkan desain yang ada menjadi suatu program dimana Tahap ini dilakukan implementasi hasil rancangan kedalam baris-baris kode program yang dapat dimengerti oleh mesin.

---

<sup>1</sup>metode waterfall : <http://bluewarrior.wordpress.com/2009/10/12/waterfall-model-vs-v-model/>

## 5. Testing

Pengujian dilakukan untuk setiap modul. Jika hasil pengujian tidak menemukan adanya masalah, modul-modul yang terpisah tersebut diintegrasikan untuk mendapatkan perangkat lunak yang utuh. Kemudian, dilakukan pengujian ditingkat perangkat lunak yang memfokuskan pada masalah-masalah logika internal, fungsi eksternal, potensi masalah yang mungkin terjadi dan pemeriksaan hasil.



**Gambar1. Pengembangan Sistem dengan metode *Waterfall***

## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1 Perancangan Aplikasi**

Aplikasi yang direncanakan ini bermanfaat untuk memberikan pelayanan dan informasi bagi pelanggan, dengan cepat, tepat serta murah. Sehingga kualitas pelayanan dapat dimaksimalkan, dan mempermudah pelanggan dalam memperoleh informasi yang dibutuhkan.

Dalam merancang aplikasi pada tugas akhir ini terlebih dahulu dilakukan pembuatan desain proses, desain data, serta desain antar muka aplikasi. Desain proses berguna untuk mengintegrasikan semua proses yang terjadi dalam aplikasi yang akan dibuat. Desain data berguna untuk mengetahui data apa saja yang dibutuhkan dalam proses yang akan dikerjakan. Sedangkan perancangan antarmuka berfungsi sebagai antar muka interaksi antara pengguna dengan sistem aplikasi yang dibuat, sehingga pengguna dapat mengoperasikan aplikasi yang dibuat.

Perancangan aplikasi ini terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Perancangan perangkat keras hanya terdiri dari perancangan kabel data yang berguna untuk menghubungkan antara PC (*Personal Computer*) dengan HP (*Handphone*). Sedangkan perancangan perangkat lunak terdiri dari perancangan aplikasi *gateway*.

##### **3.1.1 Aplikasi SMS Gateway**

Aplikasi ini merupakan aplikasi yang berhubungan dengan perangkat diluar komputer. Perangkat yang dimaksud adalah *Handphone*. Dimana terhubung dengan kabel data. Perangkat tambahan ini merupakan pintu masuk dan keluarnya sms yang diolah. Dengan fasilitas sms berupa permintaan informasi berupa informasi tagihan dan jatuh tempo, informasi tempat pembayaran terdekat dan melakukan komplain atau pengaduan.

### 3.1.2 Penggunaan Sistem

Pengguna sistem informasi ini antara lain :

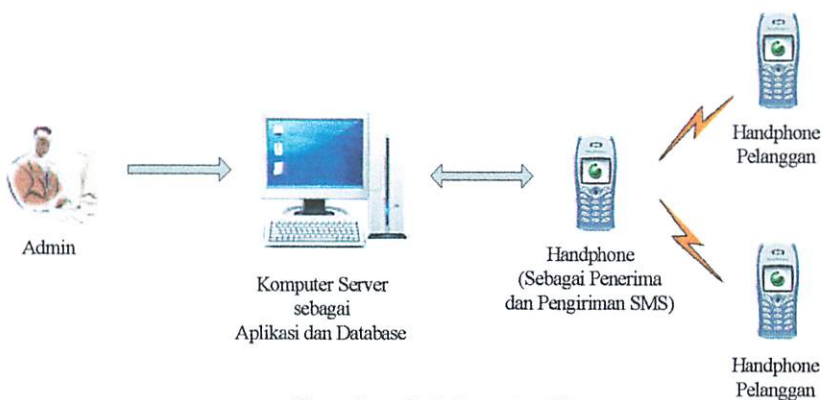
- a. *Administrator*, dalam hal ini adalah *administrator*, yaitu pengguna yang memiliki hak akses untuk mengupdate database yang di butuhkan dan mengoperasikannya.
- b. Pelanggan, yaitu sebagai pihak yang ingin mendapatkan informasi.

### 3.1.3 Spesifikasi Sistem

Sesuai dengan tujuan dari pengembangan aplikasi SMS Gateway untuk sistem informasi pada PT. TELKOM yang didukung dengan informasi berbasis sms maka spesifikasi sistem yang diperlukan antara lain sebagai berikut:

1. Sistem informasi dengan model *client - server* berbasis aplikasi.
2. Sistem *basisdata server* menggunakan MySQL.
3. Aplikasi SMS Gateway ini, terutama bagian aplikasi client diharapkan dapat dengan mudah dipakai oleh banyak orang dan operator yang mengoperasikan sistem ini.

Desain sistem aplikasi SMS Gateway di jelaskan pada gambar 3.1



**Gambar 3.1 Desain Sistem**

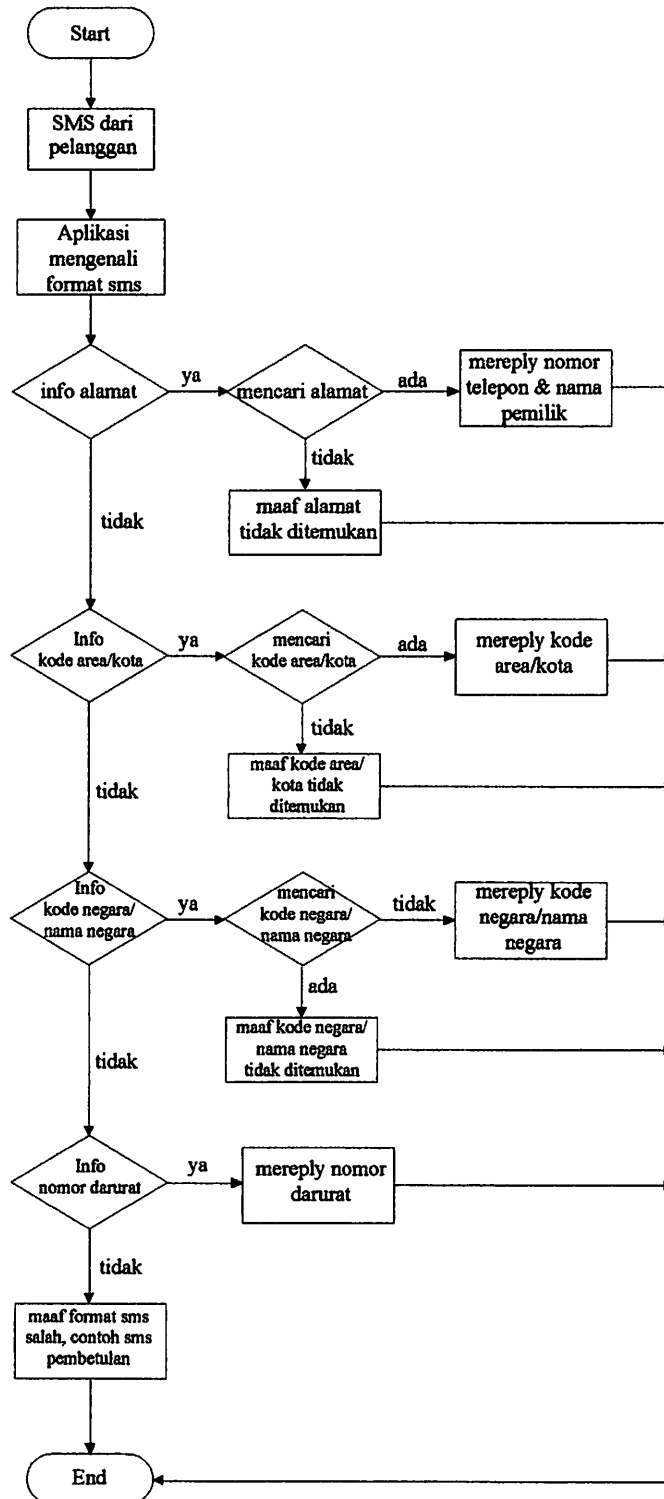
## 3.2 Perancangan Sistem

### 3.2.1 Alur Proses Aplikasi

Untuk lebih memperjelas alur dari sistem, maka alur proses aplikasi dapat digambarkan pada *flowchart* tersebut dibawah. *Flowchart* alur proses tersebut menjelaskan rancangan urutan proses yang terjadi pada aplikasi.

### 3.2.1.1 Flowchart SMS

Flowchart Alur SMS ditunjukkan pada gambar 3.2.



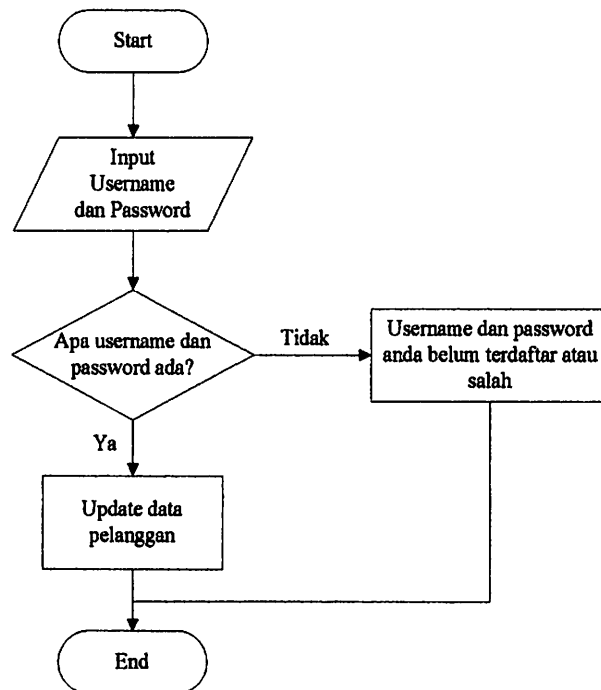
Gambar 3.2 Flowchart Alur SMS

Penjelasan *flowchart* alur aplikasi pada gambar 3.2 adalah sebagai berikut:

- a. Pelanggan menginputkan alamat untuk mendapatkan informasi mengenai nomor telepon dan pemilik telepon yang di inginkan.
- b. Pelanggan menginputkan kode area atau nama kota untuk mendapatkan informasi mengenai nama kota atau kode area.
- c. Pelanggan menginputkan kode negara atau nama negara untuk mendapatkan informasi mengenai nama negara atau kode negara.
- d. Pelanggan menginputkan permintaan nomor darurat untuk mendapatkan informasi mengenai nomor-nomor darurat yang bersifat umum.

### 3.2.1.2 *Flowchart* Login Administrator

*Flowchart* Login Administrator ditunjukkan pada gambar 3.3.



Gambar 3.3 *Flowchart* Login Administrator

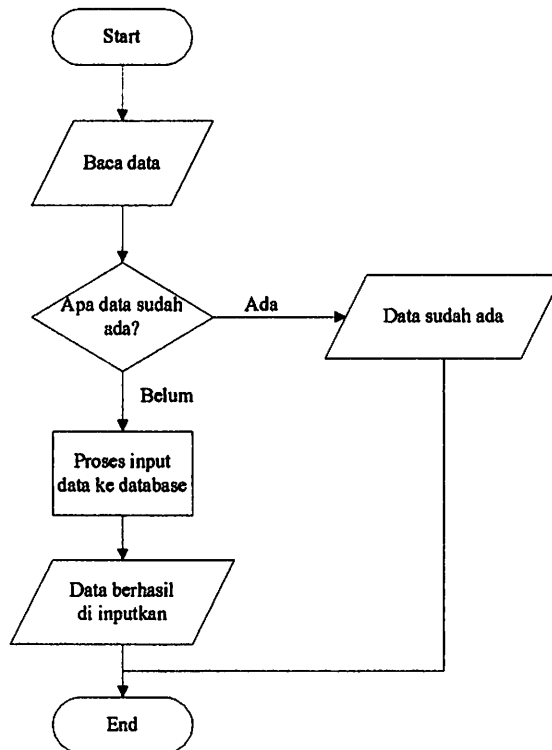
Penjelasan *flowchart* alur aplikasi pada gambar 3.3 adalah sebagai berikut:

- a. Administrator melakukan login dengan username dan password untuk dapat masuk dan mengoperasikan aplikasi.
- b. Administrator tidak akan bisa masuk dan mengoperasikan aplikasi ini jika username dan password belum terdaftar atau salah.

- c. Jika administrator berhasil melakukan login akan langsung masuk pada aplikasi dan mengoperasikannya.

### 3.2.1.3 Flowchart Memasukkan data

Flowchart Memasukkan Data ditunjukkan pada gambar 3.4.



Gambar 3.4 Flowchart Memasukkan Data

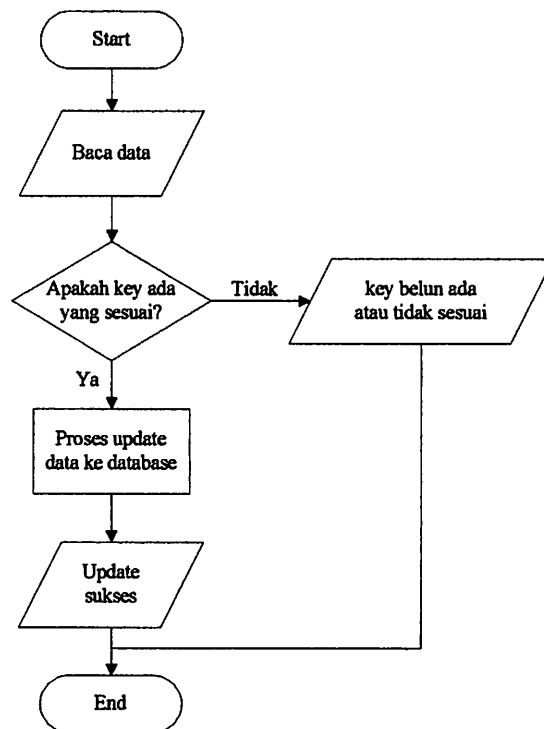
Penjelasan *flowchart* alur aplikasi pada gambar 3.4 adalah sebagai berikut:

- Administrator memasukkan data sesuai dengan form yang telah ada atau tersedia.
- Jika data yang dimasukkan sudah ada maka data akan ditolak.
- Data-data yang di *input*-kan pada form akan terkirim ke *database* untuk disimpan

### 3.2.1.4 Flowchart Merubah data

Flowchart Merubah Data akan ditunjukkan pada gambar 3.5





Gambar 3.5 *Flowchart* Merubah Data

Penjelasan *flowchart* alur aplikasi pada gambar 3.5 adalah sebagai berikut:

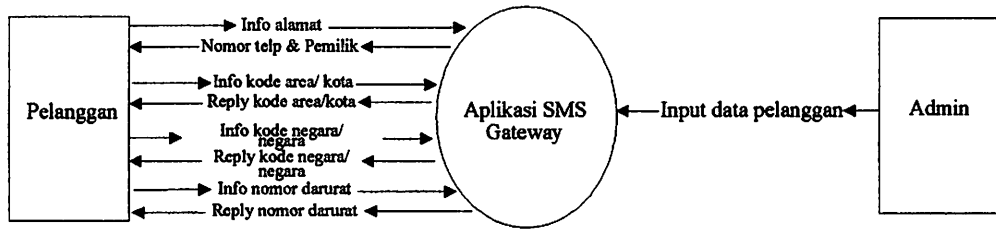
- Perubahan isi data diidentifikasi dari *key* pada masing-masing form.
- Jika sesuai maka data dapat di proses, jika tidak proses gagal.

### 3.2.2 *Data Flow Diagram (DFD)*

Dalam merancang suatu aplikasi, diperlukan suatu desain sistem yang dibuat dengan pemodelan proses. Model proses dari sistem informasi ini dibuat dalam bentuk *Data Flow Diagram (DFD)*. DFD menunjukkan bagaimana alur kerja sistem informasi ini secara nyata.

#### 3.2.2.1 *Contex Diagram*

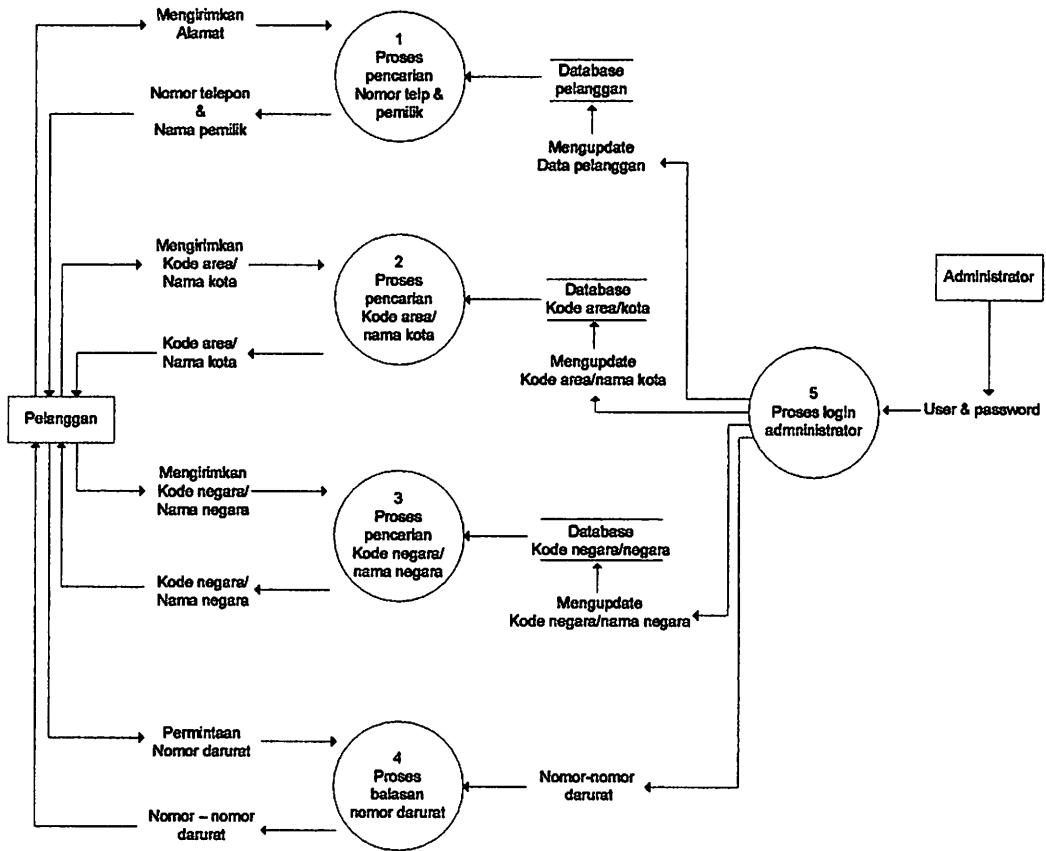
*Contex Diagram* menjelaskan hubungan sistem dengan lingkungan atau kesatuan luar. Pada sistem ini, *contex diagram* melibatkan dua kesatuan luar, yaitu Pelanggan dan Admin. *Contex diagram* sistem ini ditunjukkan pada gambar 3.6 .



Gambar 3.6 DFD Level 0

3.2.2.2 DFD Level 1

Level 1 merupakan penjabaran proses pada diagram konteks (*context diagram*) yang memuat proses-proses yang ada dalam sistem secara garis besar dan keseluruhan. Diagram arus data level 1 juga mencantumkan kesatuan luar yang berhubungan dengan sistem. Diagram arus data level 1 ini ditunjukkan pada Gambar 3.7 .



Gambar 3.7 DFD Level 1

### 3.2.3 Desain Basisdata

Terdapat dua macam *basisdata* yang digunakan pada aplikasi sistem informasi ini, yaitu *basisdata* untuk admin dan *basisdata* untuk sistem informasi yang menjadi satu bagian dari *basisdata* sistem informasi itu sendiri.

*Basisdata* untuk admin berfungsi untuk menyimpan data-data diri administrator. Sedangkan *basisdata* utama yaitu *basisdata* sistem informasi layanan 108 itu sendiri berfungsi untuk menyimpan semua informasi yang akan dikelola oleh sistem.

#### 3.2.3.1 Tabel Data Pada Aplikasi Administrator

➤ Tabel Data Pelanggan

Fungsi : Menyimpan data pelanggan

Primary key : Alamat

Tabel 3.1 menerangkan Tabel Data Pelanggan.

Tabel 3.1 Tabel Data Pelanggan

No	Nama Kolom	Type Data	Keterangan
1	Kode Area	Varchar(10)	Kode Area Kota
2	No.Telepon	Decimal(15)	No Telp Pelanggan
3	Nama	Varchar(20)	Nama Pelanggan
4	Alamat	Varchar(50)	Alamat Pelanggan,Primary key
5	No_Identitas	Varchar (20)	No.KTP/SIM Pelanggan
6	Jenis_Kelamin	Varchar(1)	Jenis Kelamin Pelanggan
7	Tempat_Lahir	Varchar(15)	Tempat Lahir Pelanggan
8	Tanggal_Lahir	Date	Tanggal Lahir Pelanggan

➤ Tabel Data Admin

Fungsi : Menyimpan data Admin,username dan password

Primary key : No.Identitas

Tabel 3.2 menerangkan Tabel Data Admin

Tabel 3.2 Tabel Data Admin

No	Nama Kolom	Type Data	Keterangan
1	Nama	Varchar(15)	Nama Admin

2	Alamat	Varchar(50)	Alamat Admin
3	Jenis_Kelamin	Varchar(15)	Jenis kelamin admin
4	No_Identitas	Varchar (15)	No.KTP/SIM Admin
5	Username	Varchar(10)	Username Administrator
6	Password	Varchar(10)	Password Administrator

➤ Tabel Kode Area

Fungsi : Menyimpan data nama kota dan kode areanya

Primary key : Wilayah

Tabel 3.3 menerangkan Tabel Kode Area

Tabel 3.3 Tabel Kode Area

No	Nama Kolom	Type Data	Keterangan
1	Kode_Area	Varchar(6)	Kode area
2	Wilayah	Varchar(30)	Nama kota

➤ Tabel Kode Negara

Fungsi : Menyimpan data nama negara dan kode negaranya

Primary key : Nama Negara

Tabel 3.4 menerangkan Tabel Kode Negara

Tabel 3.4 Tabel Kode Negara

No	Nama Kolom	Type Data	Keterangan
1	Kode_Negara	Varchar(6)	Kode negara
2	Nama_Negara	Varchar(30)	Nama negara

➤ Tabel Nomor Darurat

Fungsi : Menyimpan data 7 nomor darurat yang hanya bersifat umum

Primary key : Kode

Tabel 3.5 menerangkan Tabel Nomor Darurat

Tabel 3.5 Tabel Nomor Darurat

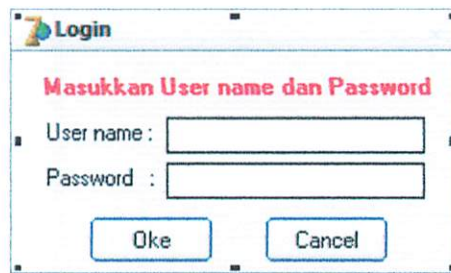
No	Nama Kolom	Type Data	Keterangan
1	Nomor_Darurat	Varchar(30)	Jenis pelayanan umum
2	Kode	Varchar(15)	Nomor kontak

### 3.2.4 Desain Antarmuka Aplikasi

Sesuai dengan spesifikasi sistem di atas, maka aplikasi ini diharapkan dapat dengan mudah dipakai oleh banyak orang dan operator / administrator yang mengoperasikan sistem ini. Untuk itu harus dibuat desain antarmuka yang mudah dipahami dan tidak terlalu rumit.

#### 3.2.4.1. Desain Halaman Login

Gambar 3.8 merupakan gambar Desain Halaman Login.

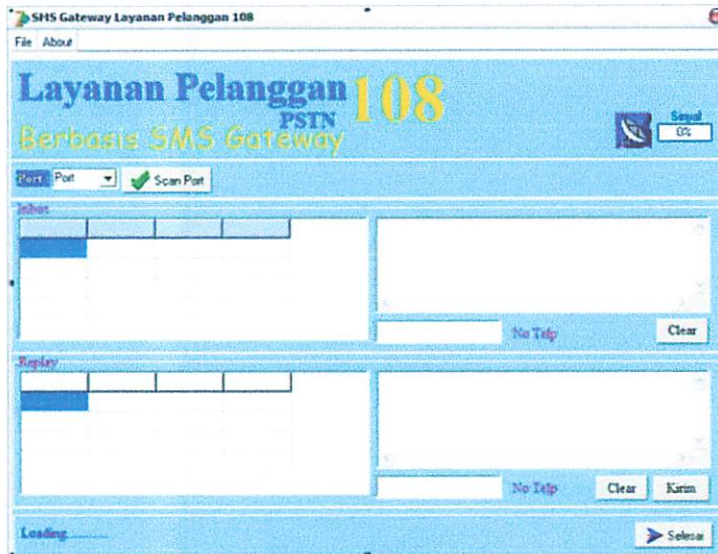


Gambar 3.8 Desain Halaman Login

Desain halaman login ini dipergunakan untuk administrator untuk mengupdate data. Dalam login hanya terdapat dua entri untuk nama pengguna dan password. Kemudian dua tombol untuk login dan cancel halaman.

#### 3.2.4.2 Desain Halaman Menu Utama Aplikasi

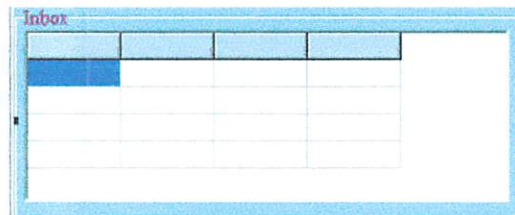
Gambar 3.9 merupakan gambar Desain Menu Halaman Aplikasi SMS Gateway.



Gambar 3.9 Desain Menu Halaman Aplikasi SMS Gateway

#### 3.2.4.3 Desain List SMS Masuk (*Inbox*)

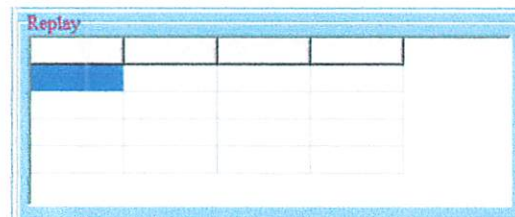
Gambar 3.10 merupakan gambar Desain List SMS Masuk.



Gambar 3.10 Desain List SMS Masuk

#### 3.2.4.4 Desain List SMS Replay (*Sent Item*)

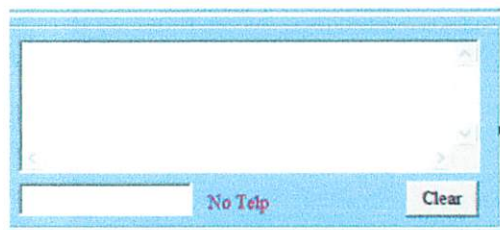
Gambar 3.11 merupakan gambar Desain List SMS Replay.



Gambar 3.11 Desain List SMS Replay

#### 3.2.4.5 Desain Baca SMS Masuk

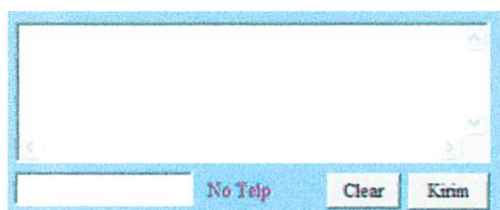
Gambar 3.12 merupakan gambar Desain Baca SMS Masuk.



Gambar 3.12 Desain Baca SMS Masuk

### 3.2.4.6 Desain pengiriman SMS

Gambar 3.13 merupakan gambar Desain pengiriman SMS.



Gambar 3.13 Desain pengiriman SMS

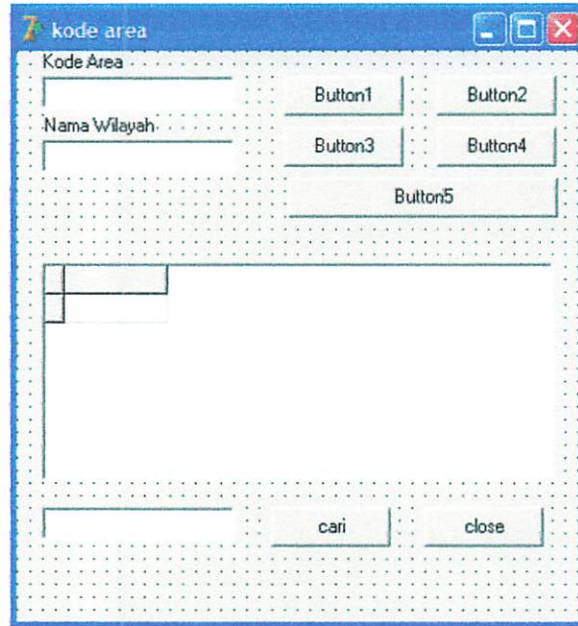
### 3.2.4.7 Desain Data Pelanggan

Gambar 3.14 merupakan gambar Desain Data Pelanggan.

Gambar 3.14 Desain Data Pelanggan

### 3.2.4.8 Desain Kode Area

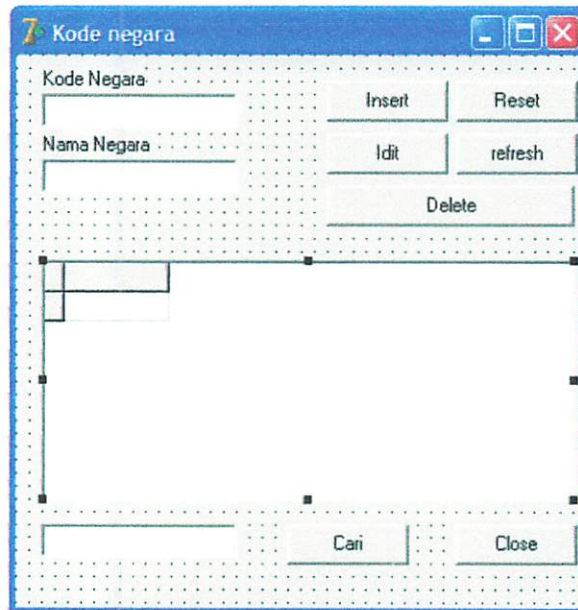
Gambar 3.15 merupakan gambar Desain Kode Area.



Gambar 3.15 Desain Kode Area

### 3.2.4.9 Desain Kode Negara

Gambar 3.16 merupakan gambar Desain Kode Negara.

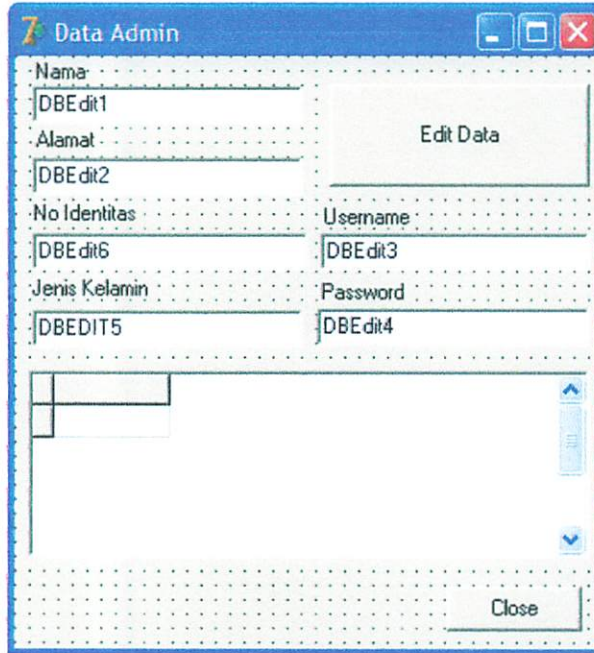


Gambar 3.16 Desain Kode Negara



### 3.2.4.10 Desain Data Admin

Gambar 3.17 merupakan gambar Desain Data Admin.

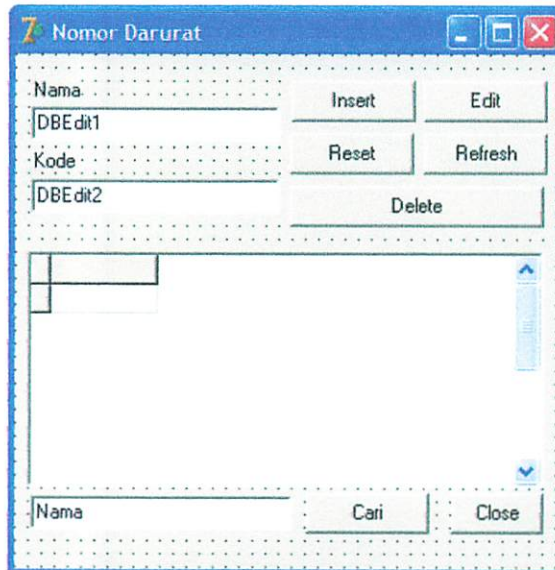


The screenshot shows a Windows-style window titled "Data Admin". It contains several text input fields for data entry: "Nama" (DBEdit1), "Alamat" (DBEdit2), "No Identitas" (DBEdit6), "Jenis Kelamin" (DBEDIT5), "Username" (DBEdit3), and "Password" (DBEdit4). A large "Edit Data" button is positioned to the right of the first two fields. Below the input fields is a table with a few rows and columns. At the bottom right of the window is a "Close" button.

Gambar 3.17 Desain Data Admin

### 3.2.4.11 Desain Nomor Darurat

Gambar 3.18 merupakan gambar Desain nomor Darurat



The screenshot shows a Windows-style window titled "Nomor Darurat". It features two text input fields: "Nama" (DBEdit1) and "Kode" (DBEdit2). To the right of these fields are several buttons: "Insert", "Edit", "Reset", "Refresh", and "Delete". Below the input fields is a table with a few rows and columns. At the bottom of the window, there is a search section with a "Nama" input field, a "Cari" button, and a "Close" button.

Gambar 3.18 Desain Nomor Darurat

## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi Sistem

Tahap implementasi pengembangan perangkat lunak merupakan proses perubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

Aplikasi sistem informasi layanan 108 ini menggunakan basis data MySQL yang berfungsi sebagai media penyimpanan data atau informasi yang terkumpul, yang terdiri dari beberapa tabel yang saling berhubungan. Implementasi dari desain aplikasi berupa implementasi struktur data dari masing-masing proses. Program bantu untuk mengimplementasikan struktur data pada masing-masing proses menggunakan Borland Delphi 7.0

#### 4.1.1 Koneksi Basis Data

Teknik pemrograman dengan menggunakan aplikasi yang berbasis *Windows* adalah teknik pemrograman yang berorientasi pada objek. Untuk itu, dibuat suatu *Module* untuk membuat program utama koneksi database, dalam hal ini adalah *ODBC*.

#### 4.1.2 Pengoperasian Basis Data

Seperti proses koneksi di atas, pengoperasian basis data juga menggunakan *Module*, dalam hal ini dibuat *Module* sama dengan *Module* koneksi database. Proses utama dari pengoperasian basis data meliputi pembacaan data, penyimpanan, pengubahan, dan penghapusan data.

#### 4.1.3 Kebutuhan Sistem

Sebelum menjalankan program atau aplikasi, ada beberapa hal yang perlu diperhatikan, antara lain kebutuhan sistem akan perangkat keras (*hardware*) dan perangkat lunak (*software*), serta langkah-langkah yang harus dilakukan untuk dapat melakukan instalasi aplikasi agar dapat berfungsi sebagaimana mestinya. Dalam perancangan dan pembuatan aplikasi ini ada beberapa perangkat keras dan lunak komputer yang dibutuhkan antara lain :

#### 4.1.3.1 Perangkat Keras

Perangkat keras komputer adalah komponen-komponen fisik peralatan yang membentuk suatu sistem komputer, serta peralatan-peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun perangkat keras yang diperlukan dalam aplikasi ini adalah :

1. CPU dengan processor 1200 Mhz atau lebih
2. Monitor XGA
3. Memory 256 MB atau lebih.
4. VGA Card dengan memory 8 MB atau lebih.
5. Printer
6. Mouse, Keyboard dan CDROM.
7. Satu buah Hp Siemens M35
8. Satu buah kabel data

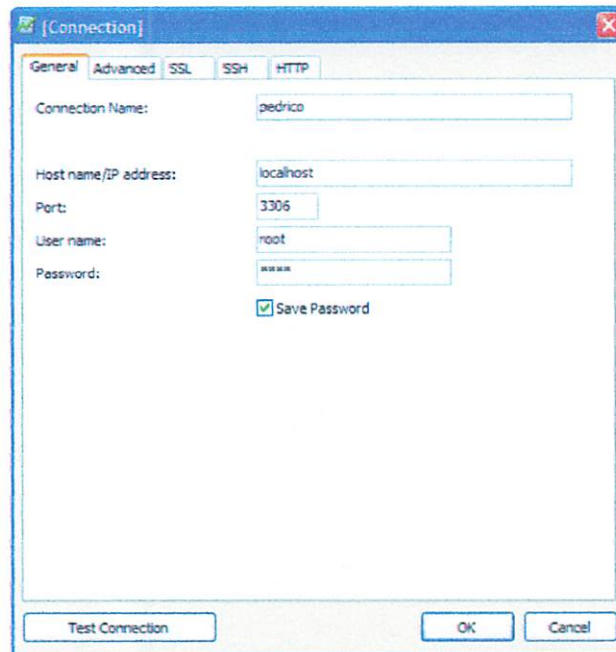
#### 4.1.3.2 Perangkat Lunak

Perangkat lunak yang diperlukan adalah program komputer yang diperlukan untuk mengoperasikan fungsi dari perangkat keras. Adapun perangkat lunak yang diperlukan dalam perancangan dan pembuatan aplikasi ini adalah :

1. Sistem Operasi Windows xp
2. Borland Delphi 7
3. MySQL

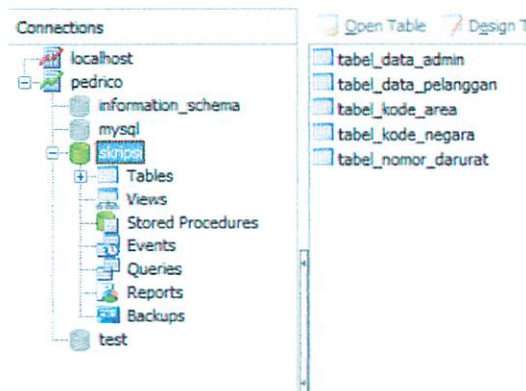
#### 4.1.4 Konfigurasi *BasisData*

Setelah semua kebutuhan perangkat keras dan lunak terpenuhi, maka langkah awal yang harus dilakukan adalah mengatur konfigurasi koneksi basis data / *database*. Maka buatlah *database* baru di MySQL dengan nama Database. Form untuk membuat database baru dijelaskan pada gambar 4.14



Gambar 4.1 Membuat *database* baru.

Setelah *database* jadi maka isi dengan tabel – tabel yang di perlukan dalam pembuatan *database* ini. Pengisian *database* ditunjukkan pada gambar 4.2



Gambar 4.2 Pengisian *database*..

Untuk membuat tabel-tabel yang dibutuhkan, cukup dengan mengklik kanan lalu *new table* setelah itu buatlah desain tabel seperti yang di inginkan.

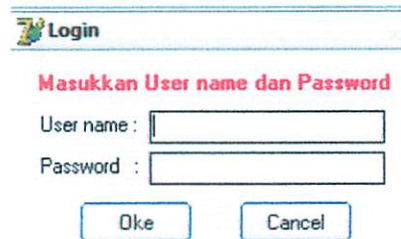
## 4.2 Pengujian Hasil

### 4.2.1 Pengujian Aplikasi

Pada sistem informasi ini merupakan aplikasi untuk penyampaian informasi yang meliputi informasi tentang nama pelanggan beserta nomor teleponnya, kode area, kode negara dan informasi nomor darurat yang bersifat umum. Pengoperasian aplikasi berupa pengolahan data pelanggan , data kode area, data data kode negara dan data

nomor darurat yang bersifat umum ,oleh karena itu pengujian hasil di fokuskan pada proses entri data, hak akses pengguna terhadap aplikasi dan pengujian hasil pada *windows*.

#### 4.2.1.1 Halaman Login



Gambar 4.3 Login Administrator

Digunakan untuk login ke form database jika ingin mengupdate data, sedangkan yang bisa login ke aplikasi adalah administrator dengan memasukkan username dan password. Halaman Login Administrator ditunjukkan pada gambar 4.3

#### 4.2.1.2 Tampilan Menu Utama

Tampilan menu utama merupakan tampilan pembuka setelah melakukan login. Di dalamnya terdapat sub menu dari masing-masing menu yang lainnya. Antara lain menu file dan menu about. Tampilan Menu Utama ditunjukkan oleh gambar 4.4.



Gambar 4.4 Tampilan Menu Utama

Tombol Scan Port : untuk mendeteksi port yang digunakan handphone saat aplikasi ini akan digunakan.

Tombol Selesai : untuk memutuskan koneksi dengan handphone pada saat aplikasi ini sudah selesai di gunakan.

Inbox : sebagai list sms masuk, memo1 untuk membaca sms baru, no.pusat pesan dan cek sinyal sedangkan tombol clear untuk membersihkan atau menghapus memo yang di baca pada memo1.

Replay : sebagai list sms replay atau balasan, memo2 untuk mengirim sms, tombol clear untuk menghapus sms yang sudah dikirim yang diketik, sedangkan tombol kirim untuk melakukan perintah pengiriman sms.

#### 4.2.1.3 Tampilan Data Pelanggan

Di dalam halaman ini digunakan untuk mengolah data para pelanggan, dalam mengolah data pelanggan ada beberapa menu yang dapat dilakukan antara lain:

Insert : untuk memasukkan data pelanggan baru

Edit : untuk meng-update data pelanggan yang sudah ada

Delete : untuk menghapus data pelanggan

Reset : untuk mengeset ulang atau mengosongkan kolom tempat mengisi data

Refresh : untuk merefresh data yang baru di inputkan atau di update

Tampilan Data Pelanggan ditunjukkan oleh gambar 4.5

Kode_Area	Nomor_Telepon	Nama	Alamat	Jenis_Kelamin	No_Iden
0354	414789	HASTORO BUD	JL PISANG KIPAS 34A I L	L	00000000
0341	428899	ELLY SETIAWA	JL THAMRIN 27 MALANCP	L	00000000
0341	415168	SUTRISNO	JL KEMBANG TURI 6 ML	L	00000000
0341	415101	MOCH TALHA	JL ARUT 10 MALANG	L	00000000
0341	428899	ELLY SETIAWA	JL THAMRIN 27 MALAN L	L	00000000

Gambar 4.5 Data Pelanggan

#### 4.2.1.4 Tampilan Data Kode Area

Di dalam halaman ini digunakan untuk mengolah data kode area , ada beberapa menu yang dapat dilakukan antara lain:

Insert : untuk memasukkan data kode area beserta nama kotanya.

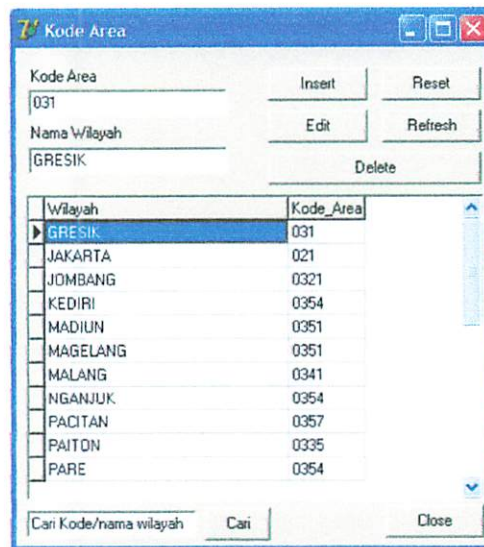
Edit : untuk meng-update data kode area yang sudah ada.

Delete : untuk menghapus data kode area.

Reset : untuk mengeset ulang atau mengosongkan kolom tempat mengisi data setelah melakukan update atau insert data.

Refresh : untuk merefresh data kode area yang baru di inputkan atau di update.

Tampilan Data Kode Area ditunjukkan oleh gambar 4.6



Gambar 4.6 Data Kode Area

#### 4.2.1.5 Tampilan Data Kode Negara

Di dalam halaman ini digunakan untuk mengolah data kode negara , dalam mengolah data ada beberapa menu yang dapat dilakukan antara lain:

Insert : untuk memasukkan data kode negara beserta kode negaranya.

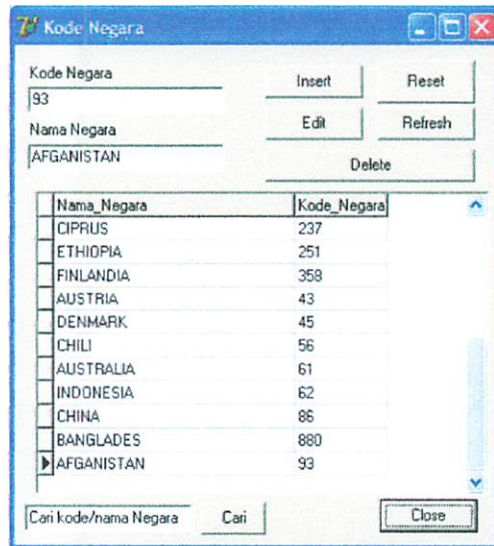
Edit : untuk meng-update data kode negara.

Delete : untuk menghapus data kode negara.

Reset : untuk mengeset ulang atau mengosongkan kolom tempat mengisi data setelah melakukan update atau insert data.

Refresh : untuk merefresh data kode negara yang baru di inputkan atau di update.

Tampilan Data Kode Negara Ditunjukkan oleh gambar 4.7



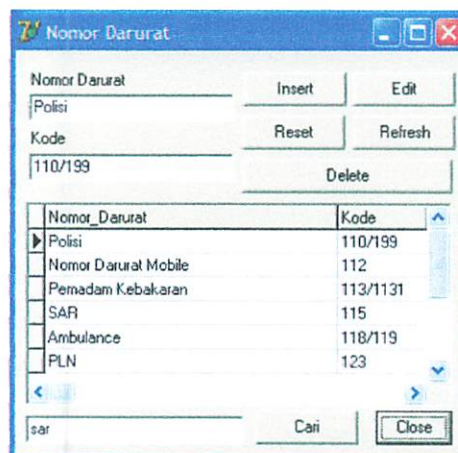
Gambar 4.7 Data Kode Negara

#### 4.2.1.6 Tampilan Data Nomor Darurat

Di dalam halaman ini digunakan untuk mengolah data nomor darurat yang bersifat umum saja, dalam mengolah data ada beberapa menu yang dapat dilakukan antara lain:

- Insert : untuk memasukkan data nomor darurat beserta kodenya
- Edit : untuk meng-update data nomor darurat yang sudah ada di database
- Delete : untuk menghapus data nomor darurat
- Reset : untuk mengeset ulang atau mengosongkan kolom tempat mengisi data setelah melakukan update atau insert data
- Refresh : untuk mererefresh data nomor darurat yang baru di inputkan atau di update

Tampilan Data Nomor Darurat ditunjukkan oleh gambar 4.8



Gambar 4.8 Data Nomor Darurat



#### 4.2.1.7 Tampilan Data Administrator

Di dalam halaman ini digunakan untuk mengolah data administrator, dalam mengolah data ada beberapa menu yang dapat dilakukan antara lain:

Edit Data : untuk meng-update data administrator yang sudah ada di database  
Tampilan Data Admin Ditunjukkan oleh gambar 4.9

Nama	Alamat
Pedrico k. Da Silva	JL SENGGANI 33 MALANG

Gambar 4.9 Data Administrator

#### 4.2.2 Pengujian SMS Gateway

##### 4.2.2.1 Proses SMS masuk

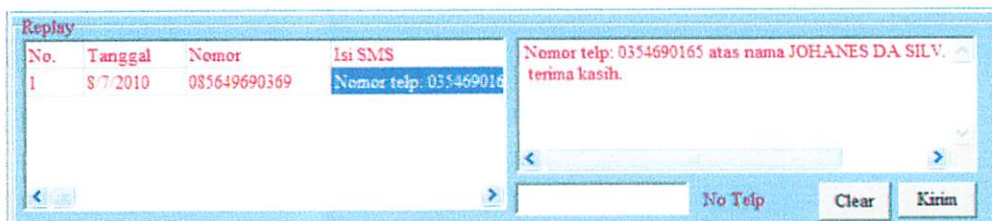
Menu *request* ini berfungsi untuk mengontrol sms yang masuk dari pelanggan ke dalam sms gateway. Menu Aplikasi Reques ditunjukkan oleh gambar 4.10

No.	Tanggal	Nomor	Isi SMS
1	06-08-2010	085731236654	info kode area 0354
2	07-08-2010	085649690369	Info Alamat Jl Pahl...

Gambar 4.10 Menu Aplikasi *Request*

##### 4.2.2.2 Proses SMS Keluar

Sedangkan menu *replay* ini untuk menampilkan informasi yang dibalas secara otomatis oleh komputer berdasarkan permintaan pelanggan. Menu Aplikasi Replay ditunjukkan oleh gambar 4.11



Gambar 4.11 Menu Aplikasi *Replay*

#### 4.2.3 Penjelasan Tentang PDU, AT Command Pada Implementasi Software

Software melakukan pengecekan SMS pada telepon celluler Siemens M35 dengan cara mengirim perintah AT COMMAND melalui kabel serial. Command dikirim dari PC ke HP dengan kecepatan 19200Bps ( berdasarkan ketentuan interface serial pada Siemens) secara serial. Jika perintah yang diterima benar, maka HP merespon Command tersebut dengan balasan “OK”, jika perintah salah atau tidak sesuai, maka HP akan merespon “ERROR”. Pada proses pembacaan SMS masuk, software harus mengirimkan perintah “AT+CMGL=0” (ketentuan AT COMMAND Set Siemens untuk baca sms masuk belum terbaca), selanjutnya jika SMS pada HP Siemens tersebut ditemukan, maka balasan dari HP akan dikirimkan dengan format PDU (Protocol Data Unit) SMS. PDU SMS merupakan kumpulan atau format penyusunan karakter dan seting SMS dari operator celluler yang dikemas menjadi paket data. Selanjutnya PDU tersebut harus dikonversi untuk mendapatkan hasil SMS yang sebenarnya.

- Request Untuk Info Alamat

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

```
07912658050000F0240D91265846690963F90000018081223001822149B7F9  
ED0AB2C3ED30DDA564BBA061F7999D6E8741B51A685D26A7E569
```

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

```
07 91 2658050000F0 24 0D 91 265846690963F9 00 00 018081223001 82  
2149B7F9ED0AB2C3ED30DDA564BBA061F7999D6E8741B51A685D26A7E569
```

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah "07", berarti 7 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 58 05 00 00 F0 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi *628505000000*, sementara Kode *91* merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu "24" menandakan bahwa PDU tersebut merupakan SMS diterima.

Kode selanjutnya dari PDU adalah *0D 91 265846690963F9* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0D* = 13 digit), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi *6285649690369*, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Urutan selanjutnya dari PDU diatas yaitu *018081223001* yang merupakan waktu pengiriman PDU SMS dari operator. Urutan diatas merupakan pasangan PDU yang dibalik dengan urutan tahun, bulan, tanggal, jam, menit dan detik. Dengan demikian maka PDU SMS tersebut dikirim pada tanggal 18-08-2010 pada jam 22:03:10.

Kode berikutnya merupakan validitas waktu SMS yaitu kode 82, dimana waktu pengiriman SMS dinyatakan valid sejak 82 x 5menit terhitung sejak tanggal dan waktu kirim SMS. Kode ini dikirim oleh operator.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

***21 49,B7,F9,ED,0A,B2,C3,ED,30,DD,A5,64,BB,A0,61,F7,99,9D,6E,87,41,B5,  
1A,68,5D,26,A7,E5,69***

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 21=33 pasang dan dikonversi dengan cara berikut:

49	= <u>01001001</u>	→	1001001	= I
B7	= <u>10110111</u>	→	1101110	= n
F9	= <u>11111001</u>	→	1100110	= f
ED	= <u>11101101</u>	→	1101111	= o
0A	= <u>00001010</u>	→	0101110	= .
B2	= <u>10110010</u>	→	1000001	= A
C3	= <u>11000011</u>	→	1101100	= l
		→	1100001	= a
ED	= <u>11101101</u>	→	1101101	= m
30	= <u>00110000</u>	→	1100001	= a
DD	= <u>11011101</u>	→	1110100	= t
A5	= <u>10100101</u>	→	0101110	= .
64	= <u>01100100</u>	→	1001010	= J
BB	= <u>10111011</u>	→	1101100	= l
A0	= <u>10100000</u>	→	0101110	= .
		→	1010000	= P
61	= <u>01100001</u>	→	1100001	= a
F7	= <u>11110111</u>	→	1101110	= n
99	= <u>10011001</u>	→	1100111	= g
9D	= <u>10011011</u>	→	1011100	= l
6E	= <u>01101110</u>	→	1101001	= i
87	= <u>10000111</u>	→	1101101	= m
41	= <u>01000001</u>	→	1100001	= a
		→	0100000	= ( spasi )
B5	= <u>10110101</u>	→	0110101	= 5

1A	= <u>00011010</u>	→	0110101	= 5
68	= <u>01101000</u>	→	0100000	= ( spasi )
5D	= <u>01011101</u>	→	1101011	= k
26	= <u>00100110</u>	→	1100101	= e
A7	= <u>10100111</u>	→	1100100	= d
E5	= <u>11100101</u>	→	1101001	= i
		→	1110010	= r
69	= <u>01101001</u>	→	1101001	= i

Hasil akhir dari konversi PDU diatas adalah :

**“Info.Alamat.Jl.Panglima 55 kediri”**

Hasil sms diatas selanjutnya diproses oleh software dengan cara mencari posisi titik (.) kemudian informasi yang diminta berdasarkan sms tersebut. Jika informasi yang diminta berupa alamat, maka software selanjutnya mencari informasi alamat yang ditulis sesudah karakter titik. Jika ada kecocokan pada database, maka software akan mengirim SMS sesuai permintaan pelanggan.

- Replay atau SMS Balasan Untuk Info Alamat

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**0691261801000001000D91265846690963F9000054CE77FB2D07D1CB6CB80E049  
BD568369CCC26CB81C2F4F01CE40EB7C3A0E9B3B87C12954F16885E96A7DB6  
1D03A3C4FA35D**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**06 91 2618010000 01 00 0D 91 265846690963F9 00 00  
39CE77FB2D07D1CB6CB80E049BD568369CCC26CB81C2F4F01CE40EB7C3A0E  
9B3B87C12954F16885E96A7DB61D03A3C4FA35D**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah "06", berarti 6 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 18 01 00 00 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi 6281100000, sementara Kode 91 merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu "01" menandakan bahwa PDU tersebut merupakan SMS dikirim.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Kode selanjutnya dari PDU adalah *0D 91 265846690963F9* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0D* = 13 digit), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi 6285649690369, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

**39 CE,77,FB,2D,07,D1,CB,6C,B8,0E,04,9B,D5,68,36,9C,CC,26,CB,81,  
C2,F4,F0,1C,E4,0E,B7,C3,A0,E9,B3,B8,7C,12,95,4F,16,88,5E,96,A7,DB,61,D0,3A,3  
C,4F,A3,5D**

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 39 =57 pasang dan dikonversi dengan cara berikut:

*CE* = 11001110 → 1001110 = *N*

*77* = 01110111 → 1101111 = *o*

$t =$	1110100	←	$\overline{11110100} =$	F4
$v =$	1100001	←		
$(\text{spas}) =$	0100000	←	$\overline{11000010} =$	C2
$9 =$	0111001	←	$\overline{10000001} =$	81
$2 =$	0110010	←	$\overline{11001011} =$	CB
$6 =$	0110110	←	$\overline{00100110} =$	26
$2 =$	0110010	←	$\overline{11001100} =$	CC
$8 =$	0111000	←	$\overline{10011100} =$	9C
$6 =$	0110110	←	$\overline{00110110} =$	36
$4 =$	0110100	←		
$5 =$	0110101	←	$\overline{01101000} =$	68
$3 =$	0110011	←	$\overline{11010101} =$	D5
$0 =$	0110000	←	$\overline{10011011} =$	9B
$(\text{spas}) =$	0100000	←	$\overline{00000100} =$	04
$:$	0111010	←	$\overline{00001110} =$	0E
$d =$	1110000	←	$\overline{10111000} =$	B8
$1 =$	1101100	←	$\overline{01101100} =$	6C
$e =$	1100101	←		
$t =$	1110100	←	$\overline{11001011} =$	CB
$(\text{spas}) =$	0100000	←	$\overline{11010001} =$	D1
$r =$	1110010	←	$\overline{00000111} =$	07
$o =$	1011111	←	$\overline{00101011} =$	2D
$m =$	1101101	←	$\overline{11111011} =$	FB

F0	= <u>11110000</u>	→	1100001	= a
1C	= <u>00011100</u>	→	1110011	= s
E4	= <u>11100100</u>	→	0100000	= (spasi)
0E	= <u>00001110</u>	→	1101110	= n
B7	= <u>10110111</u>	→	1100001	= a
C3	= <u>11000011</u>	→	1101101	= m
		→	1100001	= a
A0	= <u>10100000</u>	→	0100000	= (spasi)
E9	= <u>11101001</u>	→	1010011	= S
B3	= <u>10110011</u>	→	1001111	= O
B8	= <u>10111000</u>	→	1000101	= E
7C	= <u>01111100</u>	→	1001011	= K
12	= <u>00010010</u>	→	1001111	= O
95	= <u>10010101</u>	→	1000100	= D
		→	1001010	= J
4F	= <u>01001111</u>	→	1001111	= O
16	= <u>00010110</u>	→	0101100	= ,
88	= <u>10001000</u>	→	0100000	= (spasi)
5E	= <u>01011110</u>	→	1110100	= t
96	= <u>10010110</u>	→	1100101	= e
A7	= <u>10100111</u>	→	1110010	= r
DB	= <u>11011011</u>	→	1101001	= i
		→	1101101	= m



61	= <u>01100001</u>	→	1100001	= a
D0	= <u>11010000</u>	→	0100000	= (spasi)
3A	= <u>00111010</u>	→	1101011	= k
3C	= <u>00111100</u>	→	1100001	= a
4F	= <u>01001111</u>	→	1110011	= s
A3	= <u>10100011</u>	→	1101001	= i
5D	= <u>01011101</u>	→	1101000	= h
		→	0101110	= .

“Nomor telp : 0354682629 atas nama SOEKODJO, terima kasih.”

- Request Untuk Info Kode Area

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**079126 58050000F0240D91265846690963F90000018080910590821349**

**B7F9ED5ABEC96550585E0EBB9665725A9E06**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**07 91 2658050000F0 24 0D 91 265846690963F9 00 00 018080910590 82**

**1349B7F9ED5ABEC96550585E0EBB9665725A9E06**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah “07”, berarti 7 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 58 05 00 00 0F dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi 628505000000, sementara Kode 91 merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu “24” menandakan bahwa PDU tersebut merupakan SMS diterima.

Kode selanjutnya dari PDU adalah *0D 91 265846690963F9* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0D = 13 digit*), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi 6285649690369, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Urutan selanjutnya dari PDU diatas yaitu *018080910590* yang meruapakan waktu pengiriman PDU SMS dari operator. Urutan diatas merupakan pasangan PDU yang dibalik dengan urutan tahun, bulan, tanggal, jam, menit dan detik. Dengan demikian maka PDU SMS tersebut dikirim pada tanggal 08-08-2010 pada jam 19:50:09.

Kode berikutnya merupakan validitas waktu SMS yaitu kode 82, dimana waktu pengiriman SMS dinyatakan valid sejak 82 x 5menit terhitung sejak tanggal dan waktu kirim SMS. Kode ini dikirim oleh operator.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang meruapakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

***13,49,B7,F9,ED,5A,BE,C9,65,50,58,5E,0E,BB, 96,65,72,5A,9E,06***

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 13=19 pasang dan dikonversi dengan cara berikut:

<i>49</i>	<i>=<u>01001001</u></i>	<i>→</i>	<i>1001001</i>	<i>= I</i>
<i>B7</i>	<i>=<u>10110111</u></i>	<i>→</i>	<i>110111<u>0</u></i>	<i>= n</i>
<i>F9</i>	<i>=<u>11111001</u></i>	<i>→</i>	<i>11001<u>10</u></i>	<i>= f</i>
<i>ED</i>	<i>=<u>11101101</u></i>	<i>→</i>	<i>1101<u>111</u></i>	<i>= o</i>
<i>5A</i>	<i>=<u>01011010</u></i>	<i>→</i>	<i>0101<u>110</u></i>	<i>= .</i>

BE	= <u>10111110</u>	→	1001011	= K
C9	= <u>11001001</u>	→	1101111	= o
	Sisa	→	1100100	= d
65	= <u>01100101</u>	→	1100101	= e
50	= <u>01010000</u>	→	0100000	= (spasi)
58	= <u>01011000</u>	→	1100001	= a
5E	= <u>01011110</u>	→	1110010	= r
0E	= <u>00001110</u>	→	1100101	= e
BB	= <u>10111011</u>	→	1100001	= a
96	= <u>10010110</u>	→	1011000	= .
	Sisa	→	1001011	= K
65	= <u>01100101</u>	→	1100101	= e
72	= <u>01110010</u>	→	0011010	= d
5A	= <u>01011010</u>	→	1101001	= i
9E	= <u>10011110</u>	→	1110010	= r
06	= <u>00000110</u>	→	1101001	= i

Hasil akhir dari konversi PDU diatas adalah :

**“Info.Kode area.Kediri”**

Hasil sms diatas selanjutnya diproses oleh software dengan cara mencari posisi titik (.) kemudian informasi yang diminta berdasarkan sms tersebut. Jika informasi yang diminta berupa kode area, maka software selanjutnya mencari kode area yang ditulis sesudah karakter titik dan dilanjutkan kemudian dengan pengecekan nomor kode area kota pada karakter titik terakhir. Jika ada kecocokan pada database, maka software akan mengirim SMS sesuai permintaan pelanggan. Begitu juga sebaliknya jika yang diminta berupa info nama kotanya, maka software selanjutnya mencari nomor kode area kota yang diminta pelanggan.

- Replay atau SMS Balasan Untuk Info Kode Area

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**0691261801000001000D91265846690963F9000050CB37B90C0ACACB61D0  
35CD0EE7C368D0B2484C4A93A02039CC0EA341B0598DC602D1CBF2743B0C5A  
87E769B40B**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**06 91 2618010000 01 00 0D 91 265846690963F9 00 00  
34CB37B90C0ACACB61D035CD0EE7C368D0B2484C4A93A02039CC0EA341B059  
8DC602D1CBF2743B0C5A87E769B40B**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah "06", berarti 6 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 18 01 00 00 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi *6281100000*, sementara Kode *91* merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu "01" menandakan bahwa PDU tersebut merupakan SMS dikirim.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Kode selanjutnya dari PDU adalah *0D 91 265846690963F9* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0D* = 13 digit), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi *6285649690369*, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

**34 CB,37,B9,0C,0A,CA,CB,61,D0,35,CD,0E,E7,C3,68,D0,B2,48,4C,4A,93,A0,  
20,39,CC,0E,A3,41,B0,59,8D,C6,02,D1,CB,F2,74,3B,0C,5A,87, E7,69,B4,0B**

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 52 pasang dan dikonversi dengan cara berikut:

CB	= <u>11001011</u>	→	1001011	= K
37	= <u>00110111</u>	→	110111 <u>1</u>	= o
F9	= <u>11111001</u>	→	11001 <u>00</u>	= d
0C	= <u>00001100</u>	→	1100 <u>111</u>	= e
0A	= <u>00001010</u>	→	010 <u>0000</u>	= (spasi)
CA	= <u>11001010</u>	→	100 <u>0001</u>	= A
CB	= <u>11001011</u>	→	<u>1110010</u>	= r
Sisa		→	<u>1100101</u>	= e
61	= <u>01100001</u>	→	1100001	= a
D0	= <u>11010000</u>	→	010000 <u>0</u>	= (spasi)
35	= <u>00110101</u>	→	1010 <u>111</u>	= W
CD	= <u>11001101</u>	→	1101 <u>001</u>	= i
0E	= <u>00001110</u>	→	110 <u>1100</u>	= l
E7	= <u>11100111</u>	→	1100001	= a
C3	= <u>11000011</u>	→	<u>1110001</u>	= y
Sisa		→	<u>1100001</u>	= a
68	= <u>01101000</u>	→	1101000	= h
D0	= <u>11010000</u>	→	01101 <u>10</u>	= (spasi)
B2	= <u>10110010</u>	→	10010 <u>11</u>	= K
48	= <u>01001000</u>	→	1000 <u>101</u>	= E

4C	= <u>01001100</u>	→	<u>1000100</u>	= D
4A	= <u>01001010</u>	→	<u>1001001</u>	= I
93	= <u>10010011</u>	→	<u>1010010</u>	= R
	Sisa	→	<u>1001001</u>	= I
A0	= <u>10100000</u>	→	<u>0100000</u>	= (spasi)
20	= <u>00100000</u>	→	<u>1000001</u>	= A
39	= <u>00111001</u>	→	<u>1100100</u>	= d
CC	= <u>11001100</u>	→	<u>1100001</u>	= a
0E	= <u>00001110</u>	→	<u>1101100</u>	= l
A3	= <u>10100011</u>	→	<u>1100001</u>	= a
41	= <u>01000001</u>	→	<u>1101000</u>	= h
	Sisa	→	<u>0100000</u>	= (spasi)
B0	= <u>10110000</u>	→	<u>0110000</u>	= 0
59	= <u>01011001</u>	→	<u>0110011</u>	= 3
8D	= <u>10001101</u>	→	<u>0110101</u>	= 5
C6	= <u>11000110</u>	→	<u>0110100</u>	= 4
02	= <u>00000010</u>	→	<u>0101100</u>	= ,
D1	= <u>11010001</u>	→	<u>0100000</u>	= (space)
CB	= <u>11001011</u>	→	<u>0110000</u>	= t
	Sisa	→	<u>1100101</u>	= e
F2	= <u>11110010</u>	→	<u>1110010</u>	= r
74	= <u>01110100</u>	→	<u>1101001</u>	= i
3B	= <u>00111011</u>	→	<u>1101101</u>	= m
0C	= <u>00001100</u>	→	<u>1100001</u>	= a
5A	= <u>01011010</u>	→	<u>0100000</u>	= (space)
87	= <u>10000111</u>	→	<u>1101011</u>	= k

*E7* = 11100111 → 1100001 = *a*  
*Sisa* → 1110011 = *s*  
*69* = 01101001 → 1101001 = *i*  
*B4* = 10110100 → 0110100 = *h*  
*0B* = 00001011 → 0101110 = *.*

Hasil akhir dari konversi PDU diatas adalah :

**“Kode Area Wilayah KEDIRI Adalah 0354, terima kasih.”**

- Request Untuk Info Kode Negara

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**06912618010000240C912618321431120000018011303211821349B7F9ED5A  
BFC96590BB7C0ECBC32E9B0C**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**06 91 2618010000 24 0C 91 261832143112 00 00 018011303211 82 13  
49B7F9ED5ABFC96590BB7C0ECBC32E9B0C**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah “06”, berarti 6 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 18 01 00 00 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi *6281100000*, sementara Kode *91* merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu “24” menandakan bahwa PDU tersebut merupakan SMS diterima.

Kode selanjutnya dari PDU adalah *0C 91 261832143112* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0C* = 12 digit), kode *91* merupakan kode internasional yang dikirim perator

celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi 628123411321, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Urutan selanjutnya dari PDU diatas yaitu *018011303211* yang merupakan waktu pengiriman PDU SMS dari operator. Urutan diatas merupakan pasangan PDU yang dibalik dengan urutan tahun, bulan, tanggal, jam, menit dan detik. Dengan demikian maka PDU SMS tersebut dikirim pada tanggal 11-08-2010 pada jam 03:23:11.

Kode berikutnya merupakan validitas waktu SMS yaitu kode 82, dimana waktu pengiriman SMS dinyatakan valid sejak 82 x 5menit terhitunga sejak tanggal dan waktu kirim SMS. Kode ini dikirim oleh operator.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

***13 49,B7,F9,ED,5A,BF,C9,65,90,BB,7C,0E,CB,C3,2E,9B,0C***

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 19 pasang dan dikonversi dengan cara berikut:

<i>49</i>	<i>=<u>01001001</u></i>	<i>→</i>	<i>1001001</i>	<i>= I</i>
<i>B7</i>	<i>=<u>10110111</u></i>	<i>→</i>	<i>110111<u>0</u></i>	<i>= n</i>
<i>F9</i>	<i>=<u>11111001</u></i>	<i>→</i>	<i>110011<u>0</u></i>	<i>= f</i>
<i>ED</i>	<i>=<u>11101101</u></i>	<i>→</i>	<i>1101111</i>	<i>= o</i>
<i>5A</i>	<i>=<u>01011010</u></i>	<i>→</i>	<i>0101110</i>	<i>= .</i>
<i>BF</i>	<i>=<u>10111111</u></i>	<i>→</i>	<i>1101011</i>	<i>= k</i>
<i>C9</i>	<i>=<u>11001001</u></i>	<i>→</i>	<i>1101111</i>	<i>= o</i>
	<i>Sisa</i>	<i>→</i>	<i><u>1100100</u></i>	<i>= d</i>
<i>65</i>	<i>=<u>01100101</u></i>	<i>→</i>	<i>1100101</i>	<i>= e</i>



90	= <u>10010000</u>	→	<u>0100000</u>	= (spasi)
BB	= <u>10111011</u>	→	<u>1101110</u>	= n
7C	= <u>01111100</u>	→	<u>1100101</u>	= e
0E	= <u>00001110</u>	→	<u>1100111</u>	= g
CB	= <u>11001011</u>	→	<u>1100001</u>	= a
C3	= <u>11000011</u>	→	<u>1110010</u>	= r
	Sisa	→	<u>1100001</u>	= a
2E	= <u>00101110</u>	→	<u>0101110</u>	= .
9B	= <u>10011011</u>	→	<u>0110110</u>	= 6
0C	= <u>00001100</u>	→	<u>0110010</u>	= 2

Hasil akhir dari konversi PDU diatas adalah :

**“Info.kode negara.62”**

Hasil sms diatas selanjutnya diproses oleh software dengan cara mencari posisi titik (.) kemudian informasi yang diminta berdasarkan sms tersebut. Jika informasi yang diminta berupa kode negara, maka software selanjutnya mencari kode negara yang ditulis sesudah karakter titik dan dilanjutkan kemudian dengan pengecekan nomor kode negara pada karakter titik terakhir. Jika ada kecocokan pada database, maka software akan mengirim SMS sesuai permintaan pelanggan. Begitu juga sebaliknya jika yang diminta berupa info nama negaranya, maka software selanjutnya mencari nomor kode negara yang diminta pelanggan.

- Replay atau SMS Balasan Untuk Info Kode Negara

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**0691261801000001000C91261832143112000028EB37B90C7296CF61791894  
74129FCEE2341904F54036190B442FCBD3ED30681D9EA7D12E**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**06 91 261801000001 00 0C 91 261832143112 00 00  
28EB37B90C7296CF6179189474129FCEE2341904F54036190B442FCBD3ED3068  
1D9EA7D12E**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah "06", berarti 6 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 18 01 00 00 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi *6281100000*, sementara Kode *91* merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya dari PDU adalah *0C 91 261832143112* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0C* = 12 digit), kode *91* merupakan kode internasional yang dikirim perator celluler, sementara kode *26* dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi *628123411321*, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah *00* dan *00* yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. *00* menandakan bahwa informasi PDU berupa teks SMS, jika *01* maka berupa email, dan jika *02* berupa FAX.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

**28**        ***EB,37,B9,0C,72,96,CF,61,79,18,94,74,12,9F,CE,E2,34,19,04,F5,40,36,19,0B,44,2F,CB,D3,ED,30,68,1D,9E,A7,D1,2E***

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 52 pasang dan dikonversi dengan cara berikut:

<i>EB</i>	= <u>11101011</u>	→	<i>1101011</i>	= <i>k</i>
<i>37</i>	= <u>00110111</u>	→	<i>1101111</i>	= <i>o</i>
<i>B9</i>	= <u>10111001</u>	→	<i>1100100</i>	= <i>d</i>
<i>0C</i>	= <u>00001100</u>	→	<i>1100101</i>	= <i>e</i>
<i>72</i>	= <u>01110010</u>	→	<i>0100000</i>	= ( <i>spasi</i> )
<i>96</i>	= <u>10010110</u>	→	<i>1001110</i>	= <i>N</i>

CF	= <u>11001111</u>	→	<u>1100101</u>	= e
	Sisa	→	<u>1100111</u>	= g
61	= <u>01100001</u>	→	<u>1100001</u>	= a
79	= <u>01111001</u>	→	<u>1110010</u>	= r
18	= <u>00011000</u>	→	<u>1100001</u>	= a
94	= <u>10010100</u>	→	<u>0100000</u>	= (spasi)
74	= <u>01110100</u>	→	<u>1001001</u>	= l
12	= <u>00010010</u>	→	<u>1001110</u>	= N
9F	= <u>10011111</u>	→	<u>1000100</u>	= D
	Sisa	→	<u>1001111</u>	= O
CE	= <u>11001110</u>	→	<u>1001110</u>	= N
E2	= <u>11100010</u>	→	<u>1000101</u>	= E
34	= <u>00110100</u>	→	<u>0100011</u>	= S
19	= <u>00011001</u>	→	<u>1001001</u>	= l
04	= <u>00000100</u>	→	<u>1000001</u>	= A
F5	= <u>11110101</u>	→	<u>0100000</u>	= (spasi)
40	= <u>01000000</u>	→	<u>0111101</u>	= =
	Sisa	→	<u>0100000</u>	= (spasi)
36	= <u>00110110</u>	→	<u>0110110</u>	= 6
19	= <u>00011001</u>	→	<u>0110010</u>	= 2
0B	= <u>00001011</u>	→	<u>0101100</u>	= ,
44	= <u>01000100</u>	→	<u>0100000</u>	= (spasi)
2F	= <u>00101111</u>	→	<u>1110100</u>	= t
CB	= <u>11001011</u>	→	<u>1100101</u>	= e
D3	= <u>11010011</u>	→	<u>1110010</u>	= r
	Sisa	→	<u>1101001</u>	= i

<i>ED</i>	= <u>11101101</u>	→	<i>1101101</i>	= <i>m</i>
<i>30</i>	= <u>00110000</u>	→	<i>1100001</i>	= <i>a</i>
<i>68</i>	= <u>01101000</u>	→	<i>0100000</i>	= ( <i>spasi</i> )
<i>1D</i>	= <u>00011101</u>	→	<i>1101011</i>	= <i>k</i>
<i>9E</i>	= <u>10011110</u>	→	<i>1100001</i>	= <i>a</i>
<i>A7</i>	= <u>10100111</u>	→	<i>1110011</i>	= <i>s</i>
<i>D1</i>	= <u>11010001</u>	→	<i>1101001</i>	= <i>i</i>
<i>Sisa</i>		→	<i>1101000</i>	= <i>h</i>
<i>2E</i>	= <u>00101110</u>	→	<i>0101110</i>	= .

Hasil akhir dari konversi PDU diatas adalah :

**“kode Negara INDONESIA = 62, terima kasih”**

- Request Untuk Info Nomor Darurat

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

***07912658050000F0240D91265846690963F9 00 00 018081223370 82***  
***1249B7F9ED72BEDB6F39881C96D7E5613A***

Pada urutan PDU diatas setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

***07 91 2658050000F0 24 0D 91 265846690963F9 00 00 018081223370 82***  
***1249B7F9ED72BEDB6F39881C96D7E5613A***

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah “07”, berarti 7 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 58 05 00 00 0F dan selanjutnya

masing-masing pasangan tersebut dibalik oleh software menjadi 628505000000, sementara Kode 91 merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya yaitu "24" menandakan bahwa PDU tersebut merupakan SMS diterima.

Kode selanjutnya dari PDU adalah *0D 91 265846690963F9* yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim (*0D* = 13 digit), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi 6285649690369, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Urutan selanjutnya dari PDU diatas yaitu *018081223370* yang merupakan waktu pengiriman PDU SMS dari operator. Urutan diatas merupakan pasangan PDU yang dibalik dengan urutan tahun, bulan, tanggal, jam, menit dan detik. Dengan demikian maka PDU SMS tersebut dikirim pada tanggal 18-08-2010 pada jam 22:33:07.

Kode berikutnya merupakan validitas waktu SMS yaitu kode 82, dimana waktu pengiriman SMS dinyatakan valid sejak 82 x 5menit terhitung sejak tanggal dan waktu kirim SMS. Kode ini dikirim oleh operator.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

***12 49,B7,F9,ED,72,BE,DB,6F,39,88,1C,96,D7,E5,61,3A***

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 12=18 pasang dan dikonversi dengan cara berikut:

<i>49</i>	= <u>0</u> 1001001	→	1001001	= <i>l</i>
<i>B7</i>	= <u>10</u> 110111	→	110111 <u>0</u>	= <i>n</i>
<i>F9</i>	= <u>1111</u> 1001	→	110011 <u>0</u>	= <i>f</i>

ED	= <u>11101101</u>	→	<u>1101111</u>	= o
72	= <u>01110010</u>	→	<u>0101110</u>	= .
BE	= <u>10111110</u>	→	<u>1001110</u>	= N
DB	= <u>11011011</u>	→	<u>1101111</u>	= o
	Sisa	→	<u>1101101</u>	= m
6F	= <u>01101111</u>	→	<u>1101111</u>	= o
39	= <u>00111001</u>	→	<u>1110010</u>	= r
88	= <u>10001000</u>	→	<u>0100000</u>	= (spasi)
1C	= <u>00011100</u>	→	<u>1100100</u>	= d
96	= <u>10010110</u>	→	<u>1100001</u>	= a
D7	= <u>11010111</u>	→	<u>1110010</u>	= r
E5	= <u>11100101</u>	→	<u>1110101</u>	= u
	Sisa	→	<u>1110010</u>	= r
61	= <u>01100001</u>	→	<u>1100001</u>	= a
3A	= <u>00111010</u>	→	<u>1110100</u>	= t

Hasil akhir dari konversi PDU diatas adalah :

**“Info.Nomor darurat”**

Hasil sms diatas selanjutnya diproses oleh software dengan cara mencari posisi titik (.) kemudian informasi yang diminta berdasarkan sms tersebut. Jika informasi yang diminta berupa nomor darurat, maka software selanjutnya mencari data nomor-nomor darurat . Jika ada kecocokan pada database, maka software akan mengirim SMS sesuai permintaan pelanggan.

- **Replay atau SMS Balasan Untuk Info Kode Negara**

Pada pengujian alat ini pembacaan PDU didapat data sebagaimana berikut:

**0691261801000001000D91265846690963F900008CD0373B3D4FEB40B118  
 EC15CBE51AC1B6B87E0EBBE73A502C867BC5623957631A94EA40B158CDD  
 580BEE7EB37485C768FC3EE3028CC0EB775A0982CE76A40994E1D28269BB  
 91AD0723B4C0EB741CBB238BC0ECBC36E1D28169BBD62B159CCD570BED  
 B6F39881C96D7E561FA6B1EA697D969BA0E148BC940**

Pada urutan PDU diatas, setelah dipilah-pilah menjadi beberapa bagian oleh software pada PC, maka urutan PDU diatas menjadi:

**06 91 261801000001 00 0D 91 265846690963F9 00 00  
 87D0373B3D4FEB40B118EC15CBE51AC1B6B8CE0EBBE73A502C867BC562  
 3957631A94EA40B158CDD580BEE7EB37485C768FC3EE3028CC0EB775A098  
 2CE76A40994E1D28269BB91AD0723B4C0EB741CBB238BC0ECBC36E1D281  
 69BBD62B159CCD570BEDB6F39881C96D7E561FA6B1EA697D969BA0E148  
 BC940201008040281402010080402814020100804028140**

Baris PDU ini mempunyai beberapa informasi dari kode yang harus dipecahkan dengan cara sebagai berikut:

2 karakter pertama pada urutan PDU merupakan jumlah pasangan nomor SMS Center, jika pada contoh diatas, nilainya adalah "06", berarti 6 pasangan karakter sesudahnya merupakan pasangan Nomor SMS Center yang dibolak-balik, dengan demikian, pasangan SMS-Center pada PDU diatas adalah 91 26 18 01 00 00 01 dan selanjutnya masing-masing pasangan tersebut dibalik oleh software menjadi 628110000001, sementara Kode 91 merupakan kode internasional yang tidak perlu dibalik.

Kode selanjutnya dari PDU adalah 0D 91 265846690963F9 yang merupakan PDU untuk nomor pengirim SMS, 2 karakter terdepan adalah jumlah digit nomor pengirim ( $0C = 12$  digit), kode 91 merupakan kode internasional yang dikirim perator celluler, sementara kode 26 dan selanjutnya adalah nomor pengirim yang dibolak-balik. Setelah diterjemahkan dan dibalik kembali oleh software maka menjadi 6285649690369, yang menandakan bahwa SMS dikirim oleh nomor tersebut.

Urutan selanjutnya dari PDU diatas adalah 00 dan 00 yang merupakan kode default yang diisi operator sebagai tanda skema encoding 7 bit dan type sms. 00 menandakan bahwa informasi PDU berupa teks SMS, jika 01 maka berupa email, dan jika 02 berupa FAX.

Kode PDU terakhir dari urutan diatas adalah kode PDU isi SMS yang merupakan kode PDU 7-bit yang ditulis dalam bentuk 8-bit. Dengan demikian, maka PDU tersebut harus dikonversi kembali ke 7-bit Dengan cara sebagai berikut:

**87 D0,37,3B,3D,4F,EB,40,B1,18,EC,15,CB,E5,1A,C1,B6,B8,CE,0E,BB,E7,3A,  
50,2C,86,7B,C5,62,39,57,63,1A,94,EA,40,B1,58,CD,D5,80,BE,E7,EB,37,48,5C,  
76,8F,C3,EE,30,28,CC,0E,B7,75,A0,98,2C,E7,6A,40,99,4E,1D,28,26,9B,B9,1A,D0,72  
,3B,4C,0E,B7,41,CB,B2,38,BC,0E,CB,C3,6E,1D,28,16,9B,BD,62,B1,59,CC,D5,70,BE  
,DB,6F,39,88,1C,96,D7,E5,61,FA,6B,1E,A6,97,D9,69,BA,0E,14,8B,C9,40**

Kode 2 karakter terdepan merupakan jumlah pasangan PDU isi SMS yang berarti: 135 pasang dan dikonversi dengan cara berikut:

<i>D0</i>	= <u>11010000</u>	→	<i>1010000</i>	= <i>P</i>
<i>37</i>	= <u>00110111</u>	→	<i>1101111</i>	= <i>o</i>
<i>3B</i>	= <u>00111011</u>	→	<i>1101100</i>	= <i>l</i>
<i>3D</i>	= <u>00111101</u>	→	<i>1101001</i>	= <i>i</i>
<i>4F</i>	= <u>01001111</u>	→	<i>1110011</i>	= <i>s</i>
<i>EB</i>	= <u>11101011</u>	→	<i>1101001</i>	= <i>i</i>
<i>40</i>	= <u>01000000</u>	→	<i>0111010</i>	= <i>:</i>
<i>Sisa</i>		→	<u>0100000</u>	= ( <i>Spasi</i> )
<i>B1</i>	= <u>10110001</u>	→	<i>0110001</i>	= <i>1</i>
<i>18</i>	= <u>00011000</u>	→	<i>0110001</i>	= <i>1</i>
<i>EC</i>	= <u>11101100</u>	→	<i>0110000</i>	= <i>0</i>
<i>15</i>	= <u>00010101</u>	→	<i>0101111</i>	= <i>/</i>
<i>CB</i>	= <u>11001011</u>	→	<i>0110001</i>	= <i>1</i>
<i>E5</i>	= <u>11100101</u>	→	<i>0111001</i>	= <i>9</i>
<i>1A</i>	= <u>00011010</u>	→	<i>0111001</i>	= <i>9</i>
<i>Sisa</i>		→	<u>0001101</u>	= <i>CR (Carriage Return)</i>
<i>C1</i>	= <u>11000001</u>	→	<i>1000001</i>	= <i>A</i>



B6	= <u>10110110</u>	→	<u>1101101</u>	= m
B8	= <u>10111000</u>	→	<u>1100010</u>	= b
CE	= <u>11001110</u>	→	<u>1110101</u>	= u
0E	= <u>00001110</u>	→	<u>1101100</u>	= l
BB	= <u>10111011</u>	→	<u>1100001</u>	= a
E7	= <u>11100111</u>	→	<u>1101110</u>	= n
	Sisa	→	<u>1110011</u>	= s
3A	= <u>00111010</u>	→	<u>0111010</u>	= :
50	= <u>01010000</u>	→	<u>0100000</u>	= (spasi)
2C	= <u>00101100</u>	→	<u>0110001</u>	= 1
86	= <u>10000110</u>	→	<u>0110001</u>	= 1
7B	= <u>01111011</u>	→	<u>0111000</u>	= 8
C5	= <u>11000101</u>	→	<u>0101111</u>	= /
62	= <u>01100010</u>	→	<u>0110001</u>	= 1
	Sisa	→	<u>0110001</u>	= 1
39	= <u>00111001</u>	→	<u>0111001</u>	= 9
57	= <u>01010111</u>	→	<u>0101110</u>	= .
63	= <u>01100011</u>	→	<u>0001101</u>	= CR (Carriage Return)
1A	= <u>00011010</u>	→	<u>1010011</u>	= S
94	= <u>10010100</u>	→	<u>1000001</u>	= A
EA	= <u>11101010</u>	→	<u>1010010</u>	= R
40	= <u>01000000</u>	→	<u>0111010</u>	= :
	Sisa	→	<u>0100000</u>	= (spasi)
B1	= <u>10110001</u>	→	<u>0110001</u>	= 1
58	= <u>01011000</u>	→	<u>0110001</u>	= 1
CD	= <u>11001101</u>	→	<u>0110101</u>	= 5

D5	= <u>11010101</u>	→	<u>0101110</u>	= .
80	= <u>10000000</u>	→	<u>0001101</u>	= CR (Carriage Return)
BE	= <u>10111110</u>	→	<u>1010000</u>	= p
E7	= <u>11100111</u>	→	<u>1101111</u>	= o
	Sisa	→	<u>1110011</u>	= s
EB	= <u>11101011</u>	→	<u>1101011</u>	= k
37	= <u>00110111</u>	→	<u>1101111</u>	= o
48	= <u>01001000</u>	→	<u>0100000</u>	= (spasi)
5C	= <u>01011100</u>	→	<u>1100010</u>	= b
76	= <u>01110110</u>	→	<u>1100101</u>	= e
8F	= <u>10001111</u>	→	<u>1101110</u>	= n
C3	= <u>11000011</u>	→	<u>1100011</u>	= c
	Sisa	→	<u>1100001</u>	= a
EE	= <u>11101110</u>	→	<u>1101110</u>	= n
30	= <u>00110000</u>	→	<u>1100001</u>	= a
28	= <u>00101000</u>	→	<u>0100000</u>	= (spasi)
CC	= <u>11001100</u>	→	<u>1100001</u>	= a
0E	= <u>00001110</u>	→	<u>1101100</u>	= l
B7	= <u>10110111</u>	→	<u>1100001</u>	= a
75	= <u>01110101</u>	→	<u>1101101</u>	= m
	Sisa	→	<u>0111010</u>	= :
A0	= <u>10100000</u>	→	<u>0100000</u>	= (spasi)
98	= <u>10011000</u>	→	<u>0110001</u>	= 1
2C	= <u>00101100</u>	→	<u>0110010</u>	= 2
E7	= <u>11100111</u>	→	<u>0111001</u>	= 9
6A	= <u>01101010</u>	→	<u>0101110</u>	= .

40	= <u>01000000</u>	→	<u>0001101</u>	= CR ( <i>Carriage Return</i> )
99	= <u>10011001</u>	→	<u>1010000</u>	= P
	<i>Sisa</i>	→	<u>1001100</u>	= L
4E	= <u>01001110</u>	→	<u>1001110</u>	= N
1D	= <u>00011101</u>	→	<u>0111010</u>	= :
28	= <u>00101000</u>	→	<u>0100000</u>	= ( <i>spasi</i> )
26	= <u>00100110</u>	→	<u>0110001</u>	= 1
9B	= <u>10011011</u>	→	<u>0110010</u>	= 2
B9	= <u>10111001</u>	→	<u>0110011</u>	= 3
1A	= <u>00011010</u>	→	<u>0101110</u>	= .
	<i>Sisa</i>	→	<u>0001101</u>	= CR ( <i>Carriage Return</i> )
D0	= <u>11010000</u>	→	<u>1010000</u>	= P
72	= <u>01110010</u>	→	<u>1100101</u>	= e
3B	= <u>00111011</u>	→	<u>1101101</u>	= m
4C	= <u>01001100</u>	→	<u>1100001</u>	= a
0E	= <u>00001110</u>	→	<u>1100100</u>	= d
B7	= <u>10110111</u>	→	<u>1100001</u>	= a
41	= <u>01000001</u>	→	<u>1101101</u>	= m
	<i>Sisa</i>	→	<u>0100000</u>	= ( <i>spasi</i> )
CB	= <u>11001011</u>	→	<u>1001011</u>	= K
B2	= <u>10110010</u>	→	<u>1100101</u>	= e
38	= <u>00111000</u>	→	<u>1100010</u>	= b
BC	= <u>10111100</u>	→	<u>1100001</u>	= a
0E	= <u>00001110</u>	→	<u>1101011</u>	= k
CB	= <u>11001011</u>	→	<u>1100001</u>	= a
C3	= <u>11000011</u>	→	<u>1110010</u>	= r

	<i>Sisa</i>	→	<u>1100001</u>	= a
6E	= <u>01101110</u>	→	1101110	= n
ID	= <u>00011101</u>	→	011101 <u>0</u>	= :
28	= <u>00101000</u>	→	01000 <u>00</u>	= (spasi)
16	= <u>00010110</u>	→	01100 <u>01</u>	= l
9B	= <u>10011011</u>	→	01100 <u>01</u>	= l
BD	= <u>10111101</u>	→	01 <u>10011</u>	= 3
62	= <u>01100010</u>	→	0 <u>101111</u>	= /
	<i>Sisa</i>	→	<u>0110001</u>	= l
B1	= <u>10110001</u>	→	0110001	= l
59	= <u>01011001</u>	→	011001 <u>1</u>	= 3
CC	= <u>11001100</u>	→	01100 <u>01</u>	= l
D5	= <u>11010101</u>	→	0101 <u>110</u>	= .
70	= <u>01110000</u>	→	000 <u>1101</u>	= CR (Carriage Return)
BE	= <u>10111110</u>	→	100 <u>1110</u>	= N
DB	= <u>11011011</u>	→	110 <u>1111</u>	= o
	<i>Sisa</i>	→	<u>1101101</u>	= m
6F	= <u>01101111</u>	→	1101111	= o
39	= <u>00111001</u>	→	111001 <u>0</u>	= r
88	= <u>10001000</u>	→	01000 <u>00</u>	= (spasi)
1C	= <u>00011100</u>	→	1100 <u>100</u>	= d
96	= <u>10010110</u>	→	1100 <u>001</u>	= a
D7	= <u>11010111</u>	→	11 <u>10010</u>	= r
E5	= <u>11100101</u>	→	11 <u>10101</u>	= u
	<i>Sisa</i>	→	<u>1110010</u>	= r
61	= <u>01100001</u>	→	0110001	= a

FA	= <u>11111010</u>	→	<u>1110100</u>	= t
6B	= <u>01101011</u>	→	<u>0101111</u>	= /
1E	= <u>00011110</u>	→	<u>1110011</u>	= s
A6	= <u>10100110</u>	→	<u>1100001</u>	= a
97	= <u>10010111</u>	→	<u>1110100</u>	= t
D9	= <u>11011001</u>	→	<u>1100101</u>	= e
	Sisa	→	<u>1101100</u>	= l
69	= <u>01101001</u>	→	<u>1101001</u>	= i
BA	= <u>10111010</u>	→	<u>1110100</u>	= t
0E	= <u>00001110</u>	→	<u>0111010</u>	= :
14	= <u>00010100</u>	→	<u>0100000</u>	= ( spasi)
8B	= <u>10001011</u>	→	<u>0110001</u>	= 1
C9	= <u>11001001</u>	→	<u>0110001</u>	= 1
40	= <u>1000000</u>	→	<u>0110010</u>	= 2

Hasil akhir dari konversi PDU diatas adalah :

**Polisi: 110/119**

**Ambulans: 118/119.**

**SAR: 115.**

**Posko bencana alam: 129**

**PLN: 123.**

**Pemadam Kebakaran: 113/1131.**

**Nomor darurat/satelit: 112**

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Hasil dari konsep informasi yang dibuat ini dihasilkan adalah:

- Bagi pengguna handphone, informasi bisa didapatkan melalui SMS.
- Dapat diakses kapan pun dan dimana saja tanpa harus menggunakan telepon rumah atau wartel.
- Pada aplikasi ini dapat digunakan para pelanggan untuk mendapatkan informasi mengenai nomor telepon beserta nama pemiliknya dengan input alamat, permintaan kode area, kode negara dan permintaan nomor-nomor darurat yang bersifat umum.
- Pada pengujian aplikasi ini, belum bisa menerima sms dari pelanggan yang datang secara bersamaan.
- Dapat menggunakan *SIM Card* yang berbeda (*all operator*).

#### **5.2 Saran**

Aplikasi ini masih banyak sekali kekurangan dan masih bisa di kembangkan lagi. Untuk pengembangan aplikasi ini diharapkan memperhatikan beberapa faktor meliputi :

- Kualitas *device* seperti HP dan kabel data yang digunakan.
- Trafik SMS pada jaringan operator cellular.
- Dengan adanya aplikasi ini dapat digunakan sebagai kajian ilmiah untuk generasi mendatang.
- Pada pengembangan selanjutnya, diharapkan aplikasi ini dapat menangani proses SMS pelanggan yang datang dalam waktu yang bersamaan.

## DAFTAR PUSTAKA

- [1] **Andi dan Wahana Komputer.** *Panduan Praktis Pemrograman Borland Delphi 7.0.* Yogyakarta : Andi Offset. 2003
- [2] **Kadir, Abdul.** *Dasar Aplikasi Database MySql Delphi.* Yogyakarta : andi.2008
- [3] **Alam, M. Agus J.** *Belajar Sendiri Mengolah Database dengan Borland Delphi 7.0.* PT Elex Media Komputindo. Gramedia-Jakarta. 2004.
- [4] **Watequlis, Yan.** *Diktat kuliah Sistem Informasi.* Malang. 2006
- [5] <http://id.wikipedia.org>
- [6] <http://www.planetsourcecode.com>
- [7] <http://bluewarrior.wordpress.com/2009/10/12/waterfall-model-vs-v-model>
- [8] <http://www.mysql.com>
- [9] <http://www.newgrounds.com/portal/view/496311>
- [10] <http://elektroindonesia.com/elektro/komp27.html>
- [11] [http://parno.staff.gunadarma.ac.id/Downloads/files/4395/SI\\_03\\_DFD.pdf](http://parno.staff.gunadarma.ac.id/Downloads/files/4395/SI_03_DFD.pdf)
- [12] <http://www.pdf-search-engine.com/pengertian-dvd-pdf.html>
- [13] <http://www.telkom.ac.id>

# **LAMPIRAN**



```

1.PDU
unit text2pdu;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, Math;

type
  Tpdu = Record
    pdu : string;
    smslen : integer;
  End;
  Tsms = Record
    smsc : string;
    smsn : string;
    timestamp : string;
    textlen : integer;
    text : string;
  End;
  TText2pdu = class(TComponent)
  private
    { Private declarations }
    prKillEvent: THandle;
  protected
    { Protected declarations }
    procedure Execute;
  public
    { Public declarations }
    constructor Create(aOwner: TComponent);
    destructor Destroy; override;

  published

```

```

    { Published declarations }
    function reversenum(num: string) : string;
    function hextoint(hex:string) : byte;

    function texttosms(sms:Tsms): Tpdu;
    function smstotext(pdu: string): Tsms;
    function texttopdu(text : string) : string;
    function pdutotext(pdu:string) : string;
  end;

  procedure Register;

  implementation

  procedure Register;
  begin
    RegisterComponents('SMS', [TText2pdu]);
  end;

  constructor TText2pdu.Create(aOwner: TComponent);
  begin
    prKillEvent := CreateEvent(nil, false, false, nil);
  end;

  destructor TText2pdu.Destroy;
  begin
    SetEvent(prKillEvent);
    CloseHandle(prKillEvent);
    inherited;
  end;

  procedure TText2pdu.Execute;
  begin

```

```

end;

function TText2pdu.reversenum(num:string):string;
var
  rnum, tnum : string;
  i : byte;
begin
  {complete smsc with even digits}
  if length(num) mod 2 = 1 then
    tnum := num + 'F'
  else
    tnum := num;

  {reverse digits}
  rnum := "";
  i := 1;
  while i < length(tnum) do
    begin
      rnum := rnum + tnum[i+1] + tnum[i];
      inc(i,2);
    end;

  {remove trailing F}
  if Copy(rnum,length(rnum),1) = 'F' then
    rnum := Copy(rnum,1,length(rnum)-1);

  result := rnum;
end;

function TText2pdu.texttopdu(text : string) : string;
var
  data : string;
  tmp : array [1..255] of byte;
  shlbit,

```

```

  orbit,
  i,len : byte;
begin
  for i := 1 to length(text) do
    tmp[i] := ord(text[i]) and 127;

  shlbit := 7;
  i := 1;
  while i <= length(text) do
    begin
      if shlbit = 0 then
        begin
          shlbit := 7;
        end
      else
        begin
          if i < length(text) then
            begin
              orbit := tmp[i+1] shl shlbit;
              tmp[i] := tmp[i] or orbit;
              tmp[i+1]:= tmp[i+1] shr (8 - shlbit);
              dec(shlbit);
            end;
          end;
          inc(i);
        end;
      len := 0;
      i := 1;
      while i <= length(text) do
        begin
          inc(len);
          tmp[len] := tmp[i];
          inc(i);
          if i mod 8 = 0 then {lewati item ke 8}

```

```

    inc(i);
end;
data := "";
for i:=1 to len do
    data := data + intohex(tmp[i],2);
    result := data;
end;

function TText2pdu.texttosms(sms:Tsms): Tpdu;
var
    rsmc,rsmsn,pdu,smscoctets,smsoctets : string;
    ret : Tpdu;
begin
    rsmc := reversenum(sms.smc);
    rsmsn := reversenum(sms.smsn);
    pdu := texttopdu(sms.text);
    smscoctets := intohex((length(rsmc) div 2)+2,2) +
        '91' +
        rsmc;
    smsoctets := '1100' +
        intohex(length(sms.smsn),2)+
        '91' + rsmsn +
        '0000A7' +
        intohex(length(sms.text),2) +
        pdu;
    ret.pdu := smscoctets + smsoctets;
    ret.smlen := length(smsoctets);
    result := ret;
end;

function TText2pdu.hextoint(hex:string) : byte;
var
    ret, code : integer;
begin

```

```

    val('$'+hex,ret,code);
    result := io(ret);
end;

function TText2pdu.pdutotext(pdu:string) : string;
var
    data : string;
    octets,
    shlbit,
    andbit,
    orbit,
    i : byte;
begin
    shlbit := 7;
    i := 1;
    data := "";
    while i <= length(pdu) do
        begin
            if shlbit = 7 then
                begin
                    orbit := 0;
                    andbit:= 127;
                    shlbit:= 0;
                end;
            octets := hextoint(copy(pdu,i,2));
            data := data + chr(((octets and andbit) shl shlbit) or orbit);
            inc(i,2);
            inc(shlbit);
            andbit := andbit shr 1;
            orbit := octets shr (8 - shlbit);
            if shlbit = 7 then
                data := data + chr(orbit);
            end;
        end;
    result := data;
end;

```

```

end;

function TText2pdu.smstotext(pdu: string): Tsms;
var
  pos : integer;
  smsclen : integer;
  smsnlen : integer;
  smstext : string;
  ret : Tsms;
begin
  {extract smsc}
  pos := 1;
  smsclen := HexToInt(copy(pdu,1,2));
  pos := pos + 2 + 2;
  ret.smsc := reversenum(copy(pdu,pos,smsclen*2-2));
  pos := pos + smsclen*2 - 2;
  {04}
  pos := pos + 2;
  {extract smsn}
  smsnlen := HexToInt(copy(pdu,pos,2));
  pos := pos + 2;
  {91}
  pos := pos + 2;
  if smsnlen mod 2 = 1 then
    inc(smsnlen);
  ret.smsn := reversenum(copy(pdu,pos,smsnlen));
  pos := pos + smsnlen;
  {0000}
  pos := pos + 4;
  {yymmddhhmmssrr}
  ret.timestamp := reversenum(copy(pdu,pos,14));
  pos := pos + 14;
  {:extlen}
  ret.textlen := HexToInt(copy(pdu,pos,2));

```

```

  pos := pos + 2;
  {smstext}
  smstext := pdu_totext(Copy(pdu,pos,length(pdu)-pos+1));
  ret.text := Copy(smstext,1,ret.textlen);
  result := ret;
end;

end.

```

**2.Menu awal**  
**unit Unit1;**

**interface**

**uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
 Dialogs, Menus, Grids, StdCtrls, Buttons, VaClasses, VaComm, ComCtrls,  
 ExtCtrls, text2pdu, DBGrids, DB, ADOdb, Registry, Gauges, SUIForm, SUIDlg;

**type**

```

TForm1 = class(TForm)
  Panel1: TPanel;
  Label7: TLabel;
  Label9: TLabel;
  Label2: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Panel2: TPanel;
  Label3: TLabel;
  ComboPort: TComboBox;
  BitBtn1: TBitBtn;
  GroupBox1: TGroupBox;

```

```

Label10: TLabel;
Memo1: TMemo;
Button3: TButton;
Edit2: TEdit;
gridinbox: TStringGrid;
GroupBox2: TGroupBox;
Label11: TLabel;
Button1: TButton;
Button2: TButton;
Memo_sms: TMemo;
Edltnomor: TEdit;
Gridoutbox: TStringGrid;
GroupBox3: TGroupBox;
Label1: TLabel;
BitBtn2: TBitBtn;
MainMenu1: TMainMenu;
File1: TMenuItem;
Exit1: TMenuItem;
DataPelanggan1: TMenuItem;
KodeArea1: TMenuItem;
NomorDarurat1: TMenuItem;
KodeNegara1: TMenuItem;
Timer1: TTimer;
Memo3: TMemo;
Memo2: TMemo;
VaComm1: TVaComm;
Image1: TImage;
progress_sinyal: TGauge;
suiForm1: TsuiForm;
DataAdmin1: TMenuItem;
About1: TMenuItem;
Label6: TLabel;
Button8: TButton;
Edit1: TEdit;

```

```

About2: TMenuItem;
procedure BomboranClick(Sender: TObject);
procedure ComboBoxChar(Sender: TObject); Count: Integer;
procedure Button1Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure TIMER1BITTimer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure DataPelangganClick(Sender: TObject);
procedure KodeAreaClick(Sender: TObject);
procedure KodeNegaraClick(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure DataAdminClick(Sender: TObject);
procedure NomorDaruratClick(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure GridoutboxSelectClick(Sender: TObject); Allow: Integer;
var CanSelect: Boolean;
procedure gridinboxSelectClick(Sender: TObject); AC: ARow; Integer;
var CanSelect: Boolean;
procedure About2Click(Sender: TObject);
private
{ Private declaration;
public
{ Public declaration;
procedure Smpan_Sinyal;
procedure Baca_Sinyal;
procedure RefreshDataGrid(DaGrid: TGrid);
procedure MatchVid(DaGrid: TGrid);
procedure cek_por;
end;

```

```

var
  Form1: TForm1;
  hitung,nomor:integer;

AsPDU,PDU_kirim,nomor_SC,tanggalinbox,verifikasiinbox,isi_smsinbox:string;
tanggaloutbox,verifikasioutbox,isi_smsoutbox:STRING;
sms:ttext2pdu;
flag_kirim,sukses,normal,hasil:boolean;
// NOMOR_SENDING

const
sOK='OK'+#13+#10;
sERROR='ERROR'+#13+#10;
Swarning ='Format SMS salah'+#13+'ketik
info.alamat.(alamat)+'#13+'info.kode area.(nm kota/kode
kota)+'#13+'info.kode negara.(nm negara/kode negara)+'#13+'nfo.nomor
darurat ';
implementation

uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8;

{$R *.dfm}
procedure TForm1.cek_port;
var Reg : TRegistry;
    i:integer;
begin
  comboport.Clear;
  Reg := TRegistry.Create;
  reg.RootKey:=HKEY_LOCAL_MACHINE;
  try
    if Reg.OpenKey("\HARDWARE\DEVICEMAP\SERIALCOMM", True) then
      begin
        for i:=0 to 30 do

```

```

begin
  if Reg.ValueExists('\Device\Serial'+inttostr(i)) then
    begin
      Comboport.Items.Add( Reg.ReadString('\Device\Serial'+inttostr(i)));
      comboport.ItemIndex:=0;
      comboport.Refresh;
    end;
  end;
  Reg.CloseKey;
end;
finally
  Reg.Free;
inherited;
end;
end;

procedure update_inbox;
var i:integer;
    data:string;
begin
  form1.gridinbox.RowCount:=form1.Memo3.Lines.Count+1;
  for i:=1 to form1.Memo3.Lines.Count do
    begin
      data:=form1.Memo3.Lines.Strings[i-1];
      // showmessage(data);
      if data<>"" then
        begin
          form1.gridinbox.Cells[0,i]:=inttostr(i);
          form1.gridinbox.Cells[1,i]:=copy(data,1,pos(char($F1),data)-1);
          delete(data,1,pos(char($f1),data));
          form1.gridinbox.Cells[2,i]:=copy(data,1,pos(char($F2),data)-1);
          delete(data,1,pos(char($f2),data));
          form1.gridinbox.Cells[3,i]:=copy(data,1,length(data));

```

```

// delete(data,1,pos(char($f1),data));
end;
end;
end;

procedure update_outbox;
var l:integer;
data:string;
begin
form1.gridoutbox.RowCount:=form1.Memo2.Lines.Count+1;
for l:=1 to form1.Memo2.Lines.Count do
begin
data:=form1.Memo2.Lines.Strings[l-1];
if data<>'' then
begin
form1.gridoutbox.Cells[0,l]:=inttostr(l);
form1.gridoutbox.Cells[1,l]:=copy(data,1,pos(char($f1),data)-1);
delete(data,1,pos(char($f1),data));
form1.gridoutbox.Cells[2,l]:=copy(data,1,pos(char($f2),data)-1);
delete(data,1,pos(char($f2),data));
form1.gridoutbox.Cells[3,l]:=copy(data,1,length(data));
end;
end;
end;

// delete(data,1,pos(char($f1),data));
end;
end;
end;

procedure kirim_sms(teks_sms:string;nomor:string);
var temp:string;
jumlah:integer;
begin
verifikasioutbox:=nomor;
isi_smsoutbox:=teks_sms;
tanggaloutbox:=datetostr(date);

```

```

if teks_sms<>'' then
begin
teks_sms:=teks_sms+'';
if length(teks_sms)>160 then teks_sms:=copy(teks_sms,1,160);
teks_sms:=sms.texttopdu(teks_sms);
jumlah:=length(teks_sms) div 2;
teks_sms:=format('%0.2X',[jumlah])<>teks_sms;
if nomor<>'' then
begin
if nomor[1]='0' then
begin
delete(nomor,1,1);
insert('62',nomor,1);
end;
temp:=format('%0.2X',[length(nomor)]);
nomor:=sms.reverseenum(nomor);
nomor:=temp+'91'+nomor;
end;
end;
temp:='0100'+nomor+'0000'+teks_sms;
jumlah:=length(temp) div 2;
temp:=format('%0.2X',[jumlah]);
pdu_kirim:=nomor_SC+'0100'+nomor+'0000'+teks_sms+#26;
flag_kirim:=true;
form1.Timer1.Enabled:=false;
FORM1.vacom1.WriteText('AT+CMGS='+temp+#13);
end;

procedure cek_sms(teks_sms:string;nomor_hp:string);
var perintah,hasil:string;
scan:string;
begin
teks_sms:=uppercase(teks_sms);
// showmessage(teks_sms);

```

```

if pos('.',teks_sms)>0then
begin
perintah:=copy(teks_sms,1,pos('.',teks_sms)-1);
delete(teks_sms,1,pos('.',teks_sms));
if perintah='INFO' then
begin
IF pos('.',teks_sms)>0 then
begin
perintah:=copy(teks_sms,1,pos('.',teks_sms)-1);
delete(teks_sms,1,pos('.',teks_sms));
IF teks_sms[LENGTH(teks_sms)]=#0 then
delete(teks_sms,length(teks_sms),1);
if perintah='ALAMAT' then
begin
//cek ALAMAT
perintah:=teks_sms;
form2.ADOQuery1.Close;
form2.ADOQuery1.SQL.Clear;
form2.ADOQuery1.SQL.Add('select*from tabel_data_pelanggan ' +
'where Alamat = :Alamat');
form2.ADOQuery1.Parameters.ParamByName('Alamat').Value :=
perintah;
form2.ADOQuery1.Open;
perintah:=form2.DBGrid1.Fields[1].AsString; //no tlp
if perintah<>" then
begin
hasil:='Nomor telp: '+form2.DBGrid1.Fields[0].AsString+perintah+' atas
nama '+form2.DBGrid1.Fields[2].AsString+' , terima kasih.';
end else
begin
hasil:='Maaf! Alamat yang diminta tidak tersedia / Format SMS salah
contoh: Info.Alamat.jl.dau 45 malang atau Info.Alamat.jl.MT.Haryono II/12
malang,Terima kasih. ';
end;

```

```

//cek nomor ID
// hasil:='Alamat anda tidak dikenali';
end else
if perintah='KODE AREA' then
begin
perintah:=teks_sms;
form3.ADOQuery1.Close;
form3.ADOQuery1.SQL.Clear;
form3.ADOQuery1.SQL.Add('select*from tabel_kode_area ' +
'where Kode_Area = :Kode_Area');
form3.ADOQuery1.Parameters.ParamByName('Kode_Area').Value :=
perintah;
form3.ADOQuery1.Open;
perintah:=form3.DBGrid1.Fields[1].AsString;
if perintah<>" then
begin
perintah:="";
repeat
scan:=form3.DBGrid1.Fields[0].AsString;
perintah:=perintah+scan+' ';
form3.ADOQuery1.Next;
until scan=form3.DBGrid1.Fields[0].AsString;
delete(perintah,length(perintah)-1,2);

hasil:='Kode Area '+form3.DBGrid1.Fields[1].AsString+' = 'Wilayah
'+perintah+'. terima kasih.';

end else
begin
//scan kode berdasarkan nama wilayah
perintah:=teks_sms;
form3.ADOQuery1.Close;
form3.ADOQuery1.SQL.Clear;

```



```

form3.ADOQuery1.SQL.Add('select*from tabel_kode_area ' +
'where Wilayah = :Wilayah');
form3.ADOQuery1.Parameters.ParamByName('Wilayah').Value :=
perintah;
form3.ADOQuery1.Open;
perintah:=form3.DBGrid1.Fields[0].AsString;
if perintah<>" then
begin
hasil:='Kode Area Wilayah '+form3.DBGrid1.Fields[0].AsString+' Adalah
'+form3.DBGrid1.Fields[1].AsString+', terima kasih.';
end else hasil:='Maaf! Kode Area/nama Wilayah yang anda minta tidak
tersedia pada database kami, Terima kasih.';
end;
end else
if perintah='KODE NEGARA' then
begin
perintah:=teks_sms;
form4.ADOQuery1.Close;
form4.ADOQuery1.SQL.Clear;
form4.ADOQuery1.SQL.Add('select*from tabel_kode_negara " +
'where kode_negara = :kode_negara');
form4.ADOQuery1.Parameters.ParamByName('kode_negara').Value :=
perintah;
form4.ADOQuery1.Open;
perintah:=form4.DBGrid1.Fields[1].AsString;
if perintah<>" then
begin
hasil:='kode Negara '+form4.DBGrid1.Fields[1].AsString+' =
'+form4.DBGrid1.Fields[0].AsString+', terima kasih.';
end else
begin
//scan kode berdasarkan nama negara
perintah:=teks_sms;
form4.ADOQuery1.Close;

```

```

form4.ADOQuery1.SQL.Clear;
form4.ADOQuery1.SQL.Add('select*from tabel_kode_negara ' +
'where nama_negara = :nama_negara');
form4.ADOQuery1.Parameters.ParamByName('nama_negara').Value :=
perintah;
form4.ADOQuery1.Open;
perintah:=form4.DBGrid1.Fields[0].AsString;
if perintah<>" then
begin
hasil:='Kode Negara '+form4.DBGrid1.Fields[0].AsString+' Adalah
'+form4.DBGrid1.Fields[1].AsString+', terima kasih.';
end else hasil:='Maaf! Kode Negara yang anda minta tidak tersedia pada
database kami, Terima kasih.';
end;
end else hasil:=swarning;
end else
begin //jika sesudah info tidak ada titik
if teks_sms='NOMOR DARURAT' then
begin //++++
perintah:=form3.DBGrid1.Fields[1].AsString;
if perintah<>" then
begin
perintah:="";
form7.ADOQuery1.Close;
form7.ADOQuery1.SQL.Clear;
form7.ADOQuery1.SQL.Add('select*from tabel_normor_darura:');
form7.ADOQuery1.Open;

repeat
scan:=form7.DBGrid1.Fields[0].AsString;
perintah:=perintah+scan+' '+form7.DBGrid1.Fields[1].AsString+',
'+#13;
form7.ADOQuery1.Next;
until scan=form7.DBGrid1.Fields[0].AsString;

```

```

delete(perintah,length(perintah)-1,2);

Hasil:=perintah;
end;
end else Hasil:=swarning;
end;
end else Hasil:=swarning;
end else Hasil:=swarning;

kirim_sms(hasil,nomor_hp);

// showmessage(hasil);
end;

procedure proses_sms(kode_PDU:string);
var temp,ascii,nomor_phone,tanggal,isi_sms:string;
jumlah,code:integer;
begin
  ascii:=kode_PDU;
  //nomor sms center
  temp:=copy(ascii,1,2); //ambil karakter jumlah sms center
  val ('$'+temp,jumlah,code);
  jumlah:=jumlah * 2;
  delete(ascii,1,jumlah+2); //hapus nomor sms center

  //type sms
  delete(ascii,1,2); //hapus 2 karakter type sms

  //nomor pengirim sms
  temp:=copy(ascii,1,2); //ambil karakter jumlah nomor pengirim sms
  val ('$'+temp,jumlah,code);
  jumlah:=jumlah;
  delete(ascii,1,4); //hapus 4 karakter jumlah nmrpengirim dan kode
  intrnsional

```

```

temp:=copy(ascii,1,jumlah); //baca PDU nomor sms center
if (length(temp)>0) and (temp[length(:temp)]='F') then
temp:=copy(ascii,1,jumlah+1); //jika ganjil trmbh k:akter briku:tx
delete(ascii,1,length(temp));
temp:=sms.reversenum(temp); //nomor pengirim sms
delete(temp,1,2); //hapus 62
insert('0',temp,1); // tambah 0
nomor_phone:=temp;
//showmessage(temp);
//bentuk sms dan skema encoding
delete(ascii,1,4); //hapus bentuk sms dn skema encoding sms

//tanggal kirim sms
temp:=copy(ascii,1,6); //baca PDU tanggal kirim sms
temp:=sms.reversenum(temp); //balik PDU tanggal
delete(ascii,1,6);
temp:= copy(temp,5,2)+'-' +copy(temp,3,2)+'-' +!0'+copy(temp,1,2);
//tanggal kirim
tanggal:=temp;
// showmessage(temp);
//jam kirim sms
delete(ascii,1,6); //hapus PDU waktu sms (jam)

//validitas sms
delete(ascii,1,2); //hapus PDU waktu vliditas sms (batas)

//isi sms
temp:=copy(ascii,1,2); //ambil karakter jumlah PDU teks sms
val ('$'+temp,jumlah,code);
jumlah:=jumlah*2;
delete(ascii,1,2); //hapus 2 karakter PDU jumlah teks PDU sms
temp:=copy(ascii,1,jumlah); //baca PDU teks SMS
temp:=sms.pdutotext(temp); //k: SMS
isi_sms:=temp;

```

```

PDU_kirim:=";
form1.memo1.Text:=isi_sms;
form1.edit2.text:=nomor_phone;

form1.Memo3.Lines.Add(tanggal+char($f1)+nomor_phone+char($f2)+isi_sms);
;
form1.Memo3.Lines.SaveToFile('D:\inbox.sms');
update_inbox;
cek_sms(isi_sms,nomor_phone);
end;

procedure cek_Command(data_PDU:string);
var ascii,temp:string;
    jumlah,sinyal:integer;
begin
    if data_PDU<>" then
        begin
            temp:=copy(data_PDU,1,pos(char(#13),data_PDU)-1); //baca echo
command
            temp:=uppercase(temp);
            if temp='AT+CMGL=0'then
                begin
                    //form1.memo5.text:=data_PDU; //UNTUK CEK PDU SMS MASUK-----
                    ---
                    delete (data_PDU,1,pos(char(#10),data_PDU)); //hapus echo hingga
Line Feed char
                    temp:=copy(data_pdu,1,pos(char(#13),data_pdu)-1); //cek karakter di
baris2 IPDU
                    delete (data_PDU,1,pos(char(#10),data_PDU)); //hapus balasan
command hingga Line Feed char ke2
                    ascii:=copy(data_pdu,1,pos(char(#13),data_pdu)-1); //cek karakter PDU
// form1.Memo3.Lines.Add(temp);
                    if ascii="" then //jika PDU=OK isi sms kosong
                        begin

```

```

                delete (data_PDU,1,pos(char(#10),data_PDU)); //hapus OK pada akhir
PDU
                form1.Timer1.Enabled:=true;
                form1.label1.Caption:='SMS Kosong';

                end;// else

                ascii:=copy(data_PDU,1,pos(char(#13),data_PDU)-1); //ambil data
signal hingga enter
                // showmessage(ascii);
                if ascii="" then exit;
                data_PDU:="";
                form1.label1.Caption:='SMS ditemukan. proses.....';
                proses_sms(ascii);

                end else
                if temp='AT+CSQ' then
                    begin
                        delete (data_PDU,1,pos(':',data_PDU)+1); //hapus command hingga :
                        data_PDU:=copy(data_PDU,1,pos(char(#13),data_PDU)-1); //ambil data
signal hingga enter
                        if pos(':',data_PDU)>0 then
                            data_PDU:=copy(data_PDU,1,pos(':',data_PDU)-1); //bung koma
                            sinyal:=strtoint(data_PDU);
                            form1.progress_sinyal.Progress:=sinyal;

                        end else
                            if temp='AT+CSCA?'then
                                begin

                                    delete (data_PDU,1,pos(':',data_PDU)+1); //hapus command hingga :
                                    data_PDU:=copy(data_PDU,1,pos(char(#13),data_PDU)-1); //ambil data
signal hingga enter
                                    delete(data_PDU,1,pos('+',data_PDU)); //hapus hingga +

```

```

data_PDU:=copy(data_PDU,1,pos('','',data_PDU)-1);
data_PDU:='91'+sms.reversenum(data_PDU);
jumlah:=length(data_PDU) div 2;
nomor_SC:=format('%0.2X',[jumlah])+data_PDU;
form1.label1.Caption:='SMS Center : '+nomor_SC;
end;
end;
end;

```

```

procedure TForm1.simpan_setting_serial;
var Reg : TRegistry;
begin
  Reg := TRegistry.Create;
  try
    if Reg.OpenKey('\Software\Program_sms\serial_seting', True) then
      begin
        Reg.WriteInteger('Port', comboPort.ItemIndex);

        Reg.CloseKey;
      end;
    finally
      Reg.Free;
    inherited;
  end;
end;

```

```

procedure TForm1.Baca_Setting;
var
  Reg : TRegistry;
begin
  Reg := TRegistry.Create;

```

```

try
  If Reg.OpenKey('\Software\Program_sms\serial_seting', True) then
    begin
      If Reg.ValueExists('Port') then
        Comboport.ItemIndex := Reg.ReadInteger('Port')
      else
        comboPort.ItemIndex := 0;
      Reg.CloseKey;
    end;
  finally
    Reg.Free;
  inherited;
end;
end;

```

```

procedure TForm1.MatchWidthDbGrid(DBG:TDBGrid);
var x,y,wd,TmpWd:Integer;
begin
  for x:=0 to DBG.Columns.Count-1 do
    begin
      wd:=length(DBG.Columns.Items[x].Title.Caption);
      DBG.DataSource.DataSet.First;
      for y:=0 to DBG.DataSource.DataSet.RecordCount-1 do
        begin
          TmpWd:=Length(trim(DBG.DataSource.DataSet.Fields.Fields[x].Text));
          if TmpWd>wd then
            begin
              wd:=TmpWd;
            end;
          DBG.DataSource.DataSet.Next;
        end;// akhir for field
      DBG.Columns.Items[x].Width:=wd*10;
      DBG.Columns.Items[x].Title.Alignment:=taCenter;
    end;
  end;

```

```

end;//akhir for coloumn
DBG.DataSource.DataSet.First;
end;

procedure TForm1.RefreshDbGird(DBG:TDBGrid);
begin
    DBG.DataSource.DataSet.Active:=false;
    DBG.DataSource.DataSet.Active:=true;
    MatchWidthDbGird(DBG);
end;

procedure TForm1.ComboportClick(Sender: TObject);
var port:string;
begin
    if Comboport.Items.Count =0 then
        begin
            showmessage('Koneksi HP terhubung pada '+port);
        end else
            begin
                vacomm1.Close;
                sleep(100);
                port:=Comboport.Items.Strings[Comboport.itemindex];
                vacomm1.DeviceName:=port;
                // vacomm1.PortNum:=4;
                showmessage('Koneksi HP terhubung pada '+port);
                vacomm1.AutoOpen:=true;
                vacomm1.Open;
            end;
end;

procedure TForm1.VaComm1RxChar(Sender: TObject; Count: Integer);
VaR Data:string;
begin
    data:=vacomm1.ReadText;

```

```

asPDU:=asPDU+data;
// memo1.Text:=memo1.Text+data;
if (pos('>',asPDU)>1)and (flag_kirim=true) then
begin
    asPDU:="";
    // memo4.Text:=PDU_kirim; //UNTUK CEK PDU KIMIN(JAWAB)
    flag_kirim:=false;
    vacomm1.WriteText(PDU_kirim);
    sukses:=true;
end;// else
if pos(sOK,asPDU)>1 then
begin
    if sukses =true then
        begin
            sukses:=false;
            memo2.Lines.Add(tanggaloutbox+char($f1)+verifikasioutbox+char($f2)+Isi_sm
soutbox);
            memo2.Lines.SaveToFile('D:\Outbox.sms');
            vacomm1.WriteText('AT+CMGD=1'+#13);
            update_outbox;
            //vacomm1.WriteText('AT+CMGD=2'+#13);
            // MEMO3.Lines.Add('BERHASIL');
            EXIT;
        end;
    // memo1.Text:=asPDU;
    cek_Command(asPDU);
    asPDU:="";
    timer1.Enabled:=true;
    // break;
end;// else
if pos(sERROR,asPDU)>1 then
begin
    asPDU:="";

```

```

if sukses =true then
begin
  vacomm1.WriteText('AT+CMGD=1'+#13);
  sukses:=false;
end;
timer1.Enabled:=true;
// break;
end;

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  memo1.Clear;
  edit2.Clear;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  timer1.Enabled:=false;
  kirim_sms(memo_sms.Text,editnomor.Text);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  memo_sms.Clear;
  editnomor.Clear;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  close;
end;

```

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if hitung=1 then //cek signal
  begin
    vacomm1.WriteText('AT+CSCA?'+#13);
    label1.Caption:='Cek SMS Center.....';
  end;
  if hitung=2 then //cek level signal
  begin
    vacomm1.WriteText('AT+CSQ'+#13);
    label1.Caption:='Cek Signal.....';
  end;
  if hitung=3 then //cek level bateray
  begin
    // memo1.Clear;
    vacomm1.WriteText('AT+CMGL=0'+#13);
    hitung:=0;
    label1.Caption:='Cek SMS baru.....';
  end;
  hitung:=hitung+1;

end;

procedure TForm1.TIMERKIRIMTimer(Sender: TObject);
begin
{ timerkirim.Enabled:=false;
  sleep(20);
  flag_kirim:=true;
  sukses:=true;
  vacomm1.WriteText(PDU_kirim);
  // showmessage(PDU_kirim);
  // vacomm1.AutoOpen:=false;
  // sleep(200);
  // vacomm1.AutoOpen:=true;

```

```

timer1.Enabled:=true;}

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
memo3.Lines.LoadFromFile('D:\inbox.sms');
memo2.Lines.LoadFromFile('D:\outbox.sms');
update_inbox;
update_outbox;
gridinbox.ColWidths[0]:=40;
gridinbox.ColWidths[1]:=70;
gridinbox.ColWidths[2]:=100;
gridinbox.ColWidths[3]:=2700;
gridinbox.Cells[0,0]:='No.';
gridinbox.Cells[1,0]:='Tanggal';
gridinbox.Cells[2,0]:='Nomor';
gridinbox.Cells[3,0]:='Isi SMS';

gridoutbox.ColWidths[0]:=40;
gridoutbox.ColWidths[1]:=70;
gridoutbox.ColWidths[2]:=100;
gridoutbox.ColWidths[3]:=2700;
gridoutbox.Cells[0,0]:='No.';
gridoutbox.Cells[1,0]:='Tanggal';
gridoutbox.Cells[2,0]:='Nomor';
gridoutbox.Cells[3,0]:='Isi SMS';

sukses:=false;
hitung:=2;
nomor:=1;
cek_port;
end;

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
cek_port;
end;

procedure TForm1.DataPelanggan1Click(Sender: TObject);
begin
form6.ShowModal;
if hasil=true then form2.showmodal;
end;

procedure TForm1.KodeArea1Click(Sender: TObject);
begin
form6.ShowModal;
if hasil=true then form3.ShowModal;
end;

procedure TForm1.KodeNegara1Click(Sender: TObject);
begin
form6.ShowModal;
if hasil=true then form4.ShowModal;
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.DataAdmin1Click(Sender: TObject);
begin
form6.ShowModal;
if hasil=true then form5.ShowModal;
end;

```

```

end;

procedure TForm1.NomorDarurat1Click(Sender: TObject);
begin
  form6.ShowModal;
  if hasil=true then form7.ShowModal;
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
cek_sms(edit1.text,'081');
end;

procedure TForm1.GridoutboxSelectCell(Sender: TObject; ACol, ARow:
Integer;
var CanSelect: Boolean);
begin
  memo_sms.Text:=gridoutbox.Cells[3,ARow];
  editnomor.Text:=gridoutbox.Cells[2,ARow];
end;

procedure TForm1.gridinboxSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
  memo1.Text:=gridinbox.Cells[3,ARow];
  edit2.Text:=gridinbox.Cells[2,ARow];
end;

procedure TForm1.About2Click(Sender: TObject);
begin
  form8.ShowModal;
end;

end.

```

### 3.Data Pelanggan unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, StdCtrls, ADOdb, DB, Mask, DBCtrls;

type

```

TForm2 = class(TForm)
  ADOConnection1: TADOConnection;
  DataSource1: TDataSource;
  ADOQuery1: TADOQuery;
  ADOQuery1Nama: TWideStringField;
  ADOQuery1Alamat: TWideStringField;
  ADOQuery1Tempat_Lahir: TStringField;
  ADOQuery1Tanggal_Lahir: TDateField;
  ADOCommand1: TADOCommand;
  Button7: TButton;
  Button6: TButton;
  Edit1: TEdit;
  Edit2: TEdit;
  Button5: TButton;
  Button4: TButton;
  Button3: TButton;
  Button2: TButton;
  Button1: TButton;
  DBGrid1: TDBGrid;
  Label1: TLabel;
  DBEdit1: TDBEdit;

```



```

Label2: TLabel;
DBEdit2: TDBEdit;
Label3: TLabel;
DBEdit3: TDBEdit;
Label4: TLabel;
DBEdit4: TDBEdit;
Label5: TLabel;
DBEdit5: TDBEdit;
Label6: TLabel;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
Label7: TLabel;
DBEdit8: TDBEdit;
Label8: TLabel;
ADOQuery1Kode_Area: TStringField;
ADOQuery1Nomor_Telepon: TBCDField;
ADOQuery1No_Identitas: TStringField;
ADOQuery1Jenis_Kelamin: TStringField;
Button8: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Edit1Click(Sender: TObject);
procedure Edit2Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

```

```

var
Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
adoquery1.Insert;
adoquery1Kode_Area.Value:=dbedit1.Text;
adoquery1Nomor_Telepon.AsString:=dbedit2.Text;
adoquery1Nama.Value:=dbedit3.Text;
adoquery1Alamat.Value:=dbedit4.Text;
adoquery1Jenis_Kelamin.Value:=dbedit5.Text;
adoquery1No_Identitas.Value:=dbedit6.Text;
adoquery1Tempat_Lahir.Value:=dbedit8.Text;
adoquery1Tanggal_Lahir.AsString:=dbedit7.Text;
ADOQuery1.Close;
ADOQuery1.Open;
End;

procedure TForm2.Button2Click(Sender: TObject);
begin
adoquery1.Edit;
adoquery1kode_area.Value:=dbedit1.Text;
adoquery1nomor_telepon.Value:=strtocurr(dbedit2.Text);
adoquery1Nama.Value:=dbedit3.Text;
adoquery1kode_area.Value:=dbedit1.Text;
adoquery1Alamat.Value:=dbedit4.Text;
adoquery1Jenis_kelamin.Value:=dbedit5.Text;
adoquery1No_identitas.Value:=dbedit6.Text;
adoquery1Tempat_lahir.Value:=dbedit8.Text;

```

```

adoquery1Tanggal_lahir.Value:=strtodate(dbedit7.Text);

adoquery1.Refresh;

end;

procedure TForm2.Button3Click(Sender: TObject);
begin
With ADOCommand1 Do
IF MessageDlg('Anda yakin data'+ adoquery1Nama.value+ ' akan Dihapus?',
mtConfirmation,[mbYes,mbNo],0 ) = mrYes
Then
begin
adoquery1.Delete;
end;
ADOQuery1.Close;
ADOQuery1.Open;
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
DBEdit1.Text := "";
DBEdit2.Text := "";
DBEdit3.Text := "";
DBEdit4.Text := "";
DBEdit5.Text := "";
DBEdit6.Text := "";
DBEdit7.Text := "";
DBEdit8.Text := "";
end;

procedure TForm2.Button5Click(Sender: TObject);
begin
ADOQuery1.Close;

```

```

ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select*from tabel_data_pelanggan');
ADOQuery1.Open;
Edit1.Text := 'Search Nama';
end;

procedure TForm2.Button6Click(Sender: TObject);
begin
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select*from tabel_data_pelanggan ' +
'where Nama = :Nama');
ADOQuery1.Parameters.ParamByName('Nama').Value := Edit1.Text;
ADOQuery1.Open;

end;

procedure TForm2.Button7Click(Sender: TObject);
begin
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select*from tabel_data_pelanggan ' +
'where Alamat = :Alamat');
ADOQuery1.Parameters.ParamByName('Alamat').Value := Edit2.Text;
ADOQuery1.Open;
end;

procedure TForm2.Edit1Click(Sender: TObject);
begin
edit1.SelectAll;
end;

procedure TForm2.Edit2Click(Sender: TObject);

```

```
begin
edit2.SelectAll;
end;

end.
```

#### 4. Kode Area unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ADOdb, DB, StdCtrls, Mask, DBCtrls, Grids, DBGrids;

type

```
TForm3 = class(TForm)
DBGrid1: TDBGrid;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button5: TButton;
Edit1: TEdit;
Button6: TButton;
ADOConnection1: TADOConnection;
DataSource1: TDataSource;
```

```
ADOQuery1: TADOQuery;
ADOCommand1: TADOCommand;
Label1: TLabel;
Label2: TLabel;
ADOQuery1Wilayah: TStringField;
ADOQuery1Kode_Area: TStringField;
Button7: TButton;
procedure Button7Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Edit1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.Button7Click(Sender: TObject);
begin
close;
end;
```

```

procedure TForm3.Button1Click(Sender: TObject);
begin
with ADOCommand1 do
Begin
CommandText := 'INSERT INTO tabel_kode_area'+
'(Kode_Area, Wilayah)+'
'VALUES ( :Kode_Area, :Wilayah)';

Parameters.ParamByName('Kode_Area').Value := DBEdit1.Text;
Parameters.ParamByName('Wilayah').Value := DBEdit2.Text;

CommandType := cmdText;
Execute;
End;
ADOQuery1.Close;
ADOQuery1.Open;
ADOQuery1.Locate('Kode_area',DBEdit1.Text,[loPartialKey])
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
with adocommand1 do
begin
adoquery1.Edit;
adoquery1Wilayah.Value:=dbedit2.Text;
adoquery1Kode_Area.Value:=dbedit1.Text;
adoquery1.Refresh;
end;
end;

procedure TForm3.Button3Click(Sender: TObject);

```

```

begin
With ADOCommand1 Do
IF MessageDlg('Anda yakin data '+ adoquery1wilayah.value+' akan Dihapus?',
mtConfirmation,[mbYes,mbNo],0) = mrYes
Then
begin
CommandText :='DELETE From tabel_kode_area ' +
'WHERE wilayah = :wilayah ';
Parameters.ParamByName('wilayah').Value := DBEdit2.Text;
CommandType := cmdText;
Execute;
end;
ADOQuery1.Close;
ADOQuery1.Open;
end;

procedure TForm3.Button5Click(Sender: TObject);
begin
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select*from tabel_Kode_Area');
ADOQuery1.Open;
Edit1.Text := 'Cari kode/nama wilayah';

end;

procedure TForm3.Button6Click(Sender: TObject);
var scan:string;
begin
form3.ADOQuery1.Close;
form3.ADOQuery1.SQL.Clear;
form3.ADOQuery1.SQL.Add('select*from tabel_kode_area ' +
'where Kode_Area = :Kode_Area');

```

```

form3.ADOQuery1.Parameters.ParamByName('Kode_Area').Value :=
edit1.Text;
form3.ADOQuery1.Open;
scan:=form3.DBGrid1.Fields[1].AsString;
if scan="" then
begin
//scan kode berdasarkan nama wilayah
form3.ADOQuery1.Close;
form3.ADOQuery1.SQL.Clear;
form3.ADOQuery1.SQL.Add('select*from tabel_kode_area ' +
"where Wilayah = :Wilayah");
form3.ADOQuery1.Parameters.ParamByName('Wilayah').Value :=
edit1.Text;
form3.ADOQuery1.Open;
end;
end;

procedure TForm3.Button8Click(Sender: TObject);
var scan,hasil:string;
begin
showmessage(hasil);
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
dbedit1.Clear;
dbedit2.Clear;
end;

procedure TForm3.Edit1Click(Sender: TObject);
begin
edit1.SelectAll;
end;

```

end.

**5. Kode Negara**  
unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, ADODB, StdCtrls, Grids, DBGrids, Mask, DBCtrls;

type

```

TForm4 = class(TForm)
DBEdit1: TDBEdit;
Label1: TLabel;
Label2: TLabel;
DBEdit2: TDBEdit;
Button2: TButton;
Button1: TButton;
Button4: TButton;
Button5: TButton;
Button3: TButton;
DBGrid1: TDBGrid;
Button6: TButton;
Edit1: TEdit;
Button7: TButton;
ADOCCommand1: TADOCCommand;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
ADOConnection1: TADOConnection;
ADOQuery1Kode_Negara: TStringField;
ADOQuery1Nama_Negara: TStringField;

```

```

procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Edit1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;

implementation

uses Unit3;

{$R *.dfm}

procedure TForm4.Button6Click(Sender: TObject);
var scan:string;
begin
  form4.ADOQuery1.Close;
  form4.ADOQuery1.SQL.Clear;
  form4.ADOQuery1.SQL.Add('select*from tabel_kode_negara ' +
  'where Kode_negara = :Kode_negara');
  form4.ADOQuery1.Parameters.ParamByName('Kode_negara').Value :=
  edit1.Text;
  form4.ADOQuery1.Open;
  scan:=form4.DBGrid1.Fields[1].AsString;

```

```

if scan="" then
begin
  //scan kode berdasarkan nama wilayah
  form4.ADOQuery1.Close;
  form4.ADOQuery1.SQL.Clear;
  form4.ADOQuery1.SQL.Add('select*from tabel_kode_negara ' +
  'where nama_negara = :nama_negara');
  form4.ADOQuery1.Parameters.ParamByName('nama_negara').Value :=
  edit1.Text;
  form4.ADOQuery1.Open;
end;
end;

```

```

procedure TForm4.Button7Click(Sender: TObject);
begin
  close;
end;

```

```

procedure TForm4.Edit1Click(Sender: TObject);
begin
  edit1.SelectAll;
end;

```

```

procedure TForm4.Button4Click(Sender: TObject);
begin
  dbedit1.Clear;
  dbedit2.Clear;
end;

```

```

procedure TForm4.Button5Click(Sender: TObject);
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;

```

```

ADOQuery1.SQL.Add('select*from tabel_Kode_Negara');
ADOQuery1.Open;
Edit1.Text := 'Cari kode/nama Negara';
end;

procedure TForm4.Button3Click(Sender: TObject);
begin
With ADOCommand1 Do
IF MessageDlg('Anda yakin data '+ adoquery1Nama_Negara.value+' akan
Dihapus?',
mtConfirmation,[mbYes,mbNo],0) = mrYes
Then
begin
CommandText :='DELETE From tabel_kode_negara ' +
'WHERE Kode_Negara = :Kode_Negara ';
Parameters.ParamByName('Kode_Negara').Value := DBEdit1.Text;
CommandType := cmdText;
Execute;
end;
ADOQuery1.Close;
ADOQuery1.Open;
end;

```

```

procedure TForm4.Button1Click(Sender: TObject);
begin
with ADOCommand1 do
Begin
CommandText := 'INSERT INTO tabel_kode_negara'+
'(Kode_Negara, Nama_Negara)'+
'VALUES ( :Kode_Negara, :Nama_Negara)';
Parameters.ParamByName('Kode_Negara').Value := DBEdit1.Text;
Parameters.ParamByName('Nama_Negara').Value := DBEdit2.Text;

```

```

CommandType := cmdText;
Execute;
End;
ADOQuery1.Close;
ADOQuery1.Open;
ADOQuery1.Locate('Kode_Negara',DBEdit1.Text,[loPartialKey])
end;

```

```

procedure TForm4.Button2Click(Sender: TObject);
begin
adoquery1.Edit;
adoquery1Kode_Negara.Value:=dbedit1.Text;
adoquery1Kode_Negara.Value:=dbedit2.Text;
adoquery1.Refresh;
end;

end.

```

## 6. Data Admin unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, StdCtrls, Mask, DBCtrls, ADODB, Grids, DBGrids;

```

type
  TForm5 = class(TForm)
    DBGrid1: TDBGrid;
    ADOCommand1: TADOCommand;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    ADOConnection1: TADOConnection;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    ADOQuery1Nama: TWideStringField;
    ADOQuery1Alamat: TWideStringField;
    ADOQuery1No_Identitas: TWideStringField;
    Button6: TButton;
    ADOQuery1Username: TWideStringField;
    ADOQuery1Password: TWideStringField;
    ADOQuery1Jenis_Kelamin: TWideStringField;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    button1: TButton;
    procedure Button6Click(Sender: TObject);
    procedure button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

```

```

end;
var
  Form5: TForm5;

Implementation

{$R *.dfm}

procedure TForm5.button1Click(Sender: TObject);
begin
  adoquery1.Edit;
  adoquery1nama.Value:=dbedit1.Text;
  adoquery1alamat.Value:=dbedit2.Text;
  adoquery1no_identitas.Value:=dbedit6.Text;
  adoquery1jenis_kelamin.Value:=dbedit5.Text;
  adoquery1username.Value:=dbedit3.Text;
  adoquery1password.Value:=dbedit4.Text;
  adoquery1.Refresh;
end;
procedure TForm5.Button2Click(Sender: TObject);
begin
  { ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('select*from tabel_data_admin');
  ADOQuery1.Open; }
end;

end.
7. Login
unit Unit6;

interface

uses

```



```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, SUIForm, Mask, SUIEdit, SUIButton;
type
TForm6 = class(TForm)
  pass: TsuiMaskEdit;
  suiForm1: TsuiForm;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  editnama: TsuiMaskEdit;
  suiButton1: TsuiButton;
  suiButton2: TsuiButton;
  procedure suiButton1Click(Sender: TObject);
  procedure suiButton2Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormShow(Sender: TObject);
  procedure passKeyPress(Sender: TObject; var Key: Char);
  procedure editnamaKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }

end;

var
  Form6: TForm6;
implementation

uses Unit5, Unit4, Unit1;

{$R *.dfm}

procedure TForm6.suiButton1Click(Sender: TObject);

```

```

  var user,password:string;
  begin
    hasil:=false;
    form5.ADOQuery1.Close;
    form5.ADOQuery1.SQL.Clear;
    form5.ADOQuery1.SQL.Add('select*from tabel_data_admin ' +
'where username = :username');
    form5.ADOQuery1.Parameters.ParamByName('username').Value :=
Editnama.Text;
    form5.adOQuery1.Open;
    user:=form5.DBGrid1.Fields[4].AsString; //username
    if user<>>" then
      begin
        password:=form5.DBGrid1.Fields[5].AsString; //pwd
        if uppercase(password)=uppercase(pass.Text) then
          begin
            hasil:=true;
            close;
          end else showmessage('Maaf Password anda salah. !');
        end else showmessage('Maaf Username anda salah. !');
      end;

    procedure TForm6.suiButton2Click(Sender: TObject);
      begin
        close;
      end;

    procedure TForm6.FormClose(Sender: TObject; var Action: TCloseAction);
      begin
        close;
      end;

    procedure TForm6.FormShow(Sender: TObject);
      begin

```

```
hasil:=false;
pass.Text:="";
editnama.Text:="";
editnama.SetFocus;
end;
```

```
procedure TForm6.passKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then suibutton1.Click;
end;
```

```
procedure TForm6.editnamaKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then suibutton1.Click;
end;
```

end.

### 8. Nomor Darurat

```
unit Unit7;
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, Mask, DBCtrls;
```

```
type
```

```
TForm7 = class(TForm)
  Button1: TButton;
  Button4: TButton;
  Button2: TButton;
  Button5: TButton;
  Button3: TButton;
```

```
Label1: TLabel;
DBEdit1: TDBEdit;
Label2: TLabel;
DBEdit2: TDBEdit;
ADOCommand1: TADOCommand;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
ADOConnection1: TADOConnection;
DBGrid1: TDBGrid;
Button6: TButton;
ADOQuery1Nomor_Darurat: TStringField;
ADOQuery1Kode: TStringField;
Label6: TLabel;
Button7: TButton;
Edit1: TEdit;
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
```

```
Form7: TForm7;
```

```
implementation
```

```
{$R *.dfm}
```

```

procedure TForm7.Button1Click(Sender: TObject);
begin
  with ADOCommand1 do
  Begin
    CommandText := 'INSERT INTO tabel_nomor_darurat'+
      '{nomor_darurat, kode}'+
      'VALUES ( :nomor_darurat, :kode)';
    Parameters.ParamByName('nomor_darurat').Value := DBEdit1.Text;
    Parameters.ParamByName('kode').Value := DBEdit2.Text;

    CommandType := cmdText;
    Execute;
  End;
end;

```

```

procedure TForm7.Button4Click(Sender: TObject);
begin
  dbedit1.Clear;
  dbedit2.Clear;
end;

```

```

procedure TForm7.Button7Click(Sender: TObject);
begin
  close;
end;

```

```

procedure TForm7.Button5Click(Sender: TObject);
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('select*from tabel_nomor_darurat');
  ADOQuery1.Open;
end;

```

```

procedure TForm7.Button6Click(Sender: TObject);

```

```

var scan:string;
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('select*from tabel_nomor_darurat ' +
    'where nomor_darurat = :nomor_Darurat');
  ADOQuery1.Parameters.ParamByName('Nomor_Darurat').Value :=
    edit1.Text;
  ADOQuery1.Open;
  scan:=DBGrid1.Fields[1].AsString;
end;
procedure TForm7.Button3Click(Sender: TObject);
begin
  With ADOCommand1 Do
  IF MessageDlg('Anda yakin data '+ adoquery1nomor_darurat.value+' akan
  Dihapus?',
    mtConfirmation,[mbYes,mbNo],0 ) = mrYes
  Then
  begin
    CommandText :='DELETE From tabel_nomor_darurat ' +
      'WHERE nomor_darurat = :nomor_darurat ';
    Parameters.ParamByName('nomor_darurat').Value := DBEdit1.Text;
    CommandType := cmdText;
    Execute;
  end;
  ADOQuery1.Close;
  ADOQuery1.Open;
end;
procedure TForm7.Button2Click(Sender: TObject);
begin
  adoquery1.Edit;
  adoquery1nomor_darurat.Value:=dbedit1.Text;
  adoquery1kode.Value:=dbedit2.Text;
  adoquery1.Refresh;

```

end;

end.

**9.About**  
**unit Unit8;**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls, jpeg;

type

TForm8 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Button1: TButton;

Image1: TImage;

Label9: TLabel;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form8: TForm8;

implementation

{ \$R \*.dfm }

procedure TForm8.Button1Click(Sender: TObject);

begin

close;



end;

end.



## LEMBAR PENGAJUAN JUDUL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi: Teknik Energi Listrik/Teknik Elektronika/Teknik Komputer & Informatika\*)

1.	Nama Mahasiswa: <u>PEDRICO K. DA SILVA</u>	Nim: <u>0512529</u>
2.	Waktu Pengajuan	Tanggal: <u>23</u> Bulan: <u>April</u> Tahun: <u>2010</u>
3.	Spesifikasi Judul (berilah tanda silang)**)	
	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi
	d. Sistem Kendali Industri	h. lainnya .....
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*)  <u>Ir. Muisan M. MT</u>	Ketua Jurusan   <u>Ir. F. Yudi Limpraptono, MT</u> NIP. P. 1039500274
5.	Judul yang diajukan mahasiswa	<u>PENGEMBANGAN SISTEM INFORMASI 108 PSTM (PUBLIC SWITCH TELEPHONE NETWORK) MENGGUNAKAN SMS GATEWAY</u>
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	.....
	Catatan:	.....
7.	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui Dosen <span style="float: right;">200</span>  

**Perhatian:**

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: \*) Coret yang tidak perlu  
\*\*) dilingkari a, b, c, ..... atau g sesuai bidang keahlian

Lampiran : 1 (satu) berkas  
**Pembimbing Skripsi**

Kepada : Yth. Ir. TH. Mimien Mustikawati, MT  
Dosen Institut Teknologi Nasional  
MALANG

Yang bertanda tangan di bawah ini:

Nama : PEDRICO KARLINOVIANDY DA SILVA  
Nim : 05.12.529  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika S-1

Dengan ini mengajukan permohonan, kiranya Bapak / Ibu bersedia menjadi Dosen Pembimbing Utama / ~~Pendamping~~ \*), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

**PENGEMBANGAN SISTEM INFORMASI 108 PSTN**  
**(PUBLIC SWITCH TELEPHONE NETWORK)**  
**MENGGUNAKAN SMS GATEWAY**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan ini kami buat.

Mengetahui



**Ir. F. Yudi Limpraptono, MT**

**NIP Y. 1039500274**

Malang, 23 Maret 2010

Hormat Kami,



**Pedrico K. Da Silva**

**05.12.529**

**Catatan :**

**\*) Coret yang tidak perlu**

Form S-3 a

**INSTITUT TEKNOLOGI NASIONAL**  
**Jl. Raya Karanglo Km 2**  
**MALANG**

---

**PERNYATAAN KESEDIAAN DALAM PEMBIMBING SKRIPSI**

Sesuai permohonan dari Mahasiswa :

Nama : PEDRICO KARLINOVIANDY DA SILVA  
Nim : 05.12.529  
Semester : X ( Sepuluh )  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika S-1

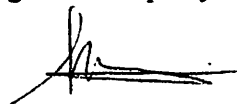
Dengan ini menyatakan bersedia / ~~tidak bersedia~~ \*) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**PENGEMBANGAN SISTEM INFORMASI 108 PSTN  
(PUBLIC SWITCH TELEPHONE NETWORK)  
MENGUNAKAN SMS GATEWAY**

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 23 Maret 2010

Kami yang membuat pernyataan,



**Ir. TH. Mimien Mustikawati, MT**

**NIP.P.1030000352**

**Catatan :**

Setelah disetujui formulir  
Diserahkan mahasiswa/ I yang bersangkutan  
Kepada Jurusan untuk diproses lebih lanjut  
\*) Coret yang tidak perlu

Form S-3 b

**PERNYATAAN KESEDIAAN DALAM PEMBIMBING SKRIPSI**

Sesuai permohonan dari Mahasiswa :

Nama : PEDRICO KARLINOVIANDY DA SILVA

Nim : 05.12.529

Semester : X (sepuluh)

Jurusan : Teknik Komputer & Informatika S-1

Dengan ini menyatakan bersedia / ~~tidak bersedia~~ \*) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**PENGEMBANGAN SISTEM INFORMASI 108 PSTN  
(PUBLIC SWITCH TELEPHONE NETWORK)  
MENGUNAKAN SMS GATEWAY**

Dengan surat pernyataan ini kami buat agar dapat dapat dipergunakan seperlunya.

Malang, 23 Maret 2010

Kami yang membuat pernyataan,



**I Komang Somawirata, ST.MT**  
**NIP.Y.1030100361**

**Catatan :**  
Setelah disetujui formulir  
Diserahkan mahasiswa / I yang bersangkutan  
Kepada Jurusan untuk diproses lebih lanjut  
\*) Coret yang tidak perlu

Form S-3 b





**FORMULIR BIMBINGAN SKRIPSI**

Nama : Pedrico Karlinoviandy Da Silva  
Nim : 06.12.529  
Masa Bimbingan : 30 Maret 2010 s/d 30 September 2010  
Judul Skripsi : PENGEMBANGAN SISTEM INFORMASI LAYANAN 108 PSTN  
(PUBLIC SWITCH TELEPHONE NETWORK MENGGUNAKAN  
SMS GATEWAY)

No	Tanggal	Uraian	Paraf Pembimbing
1	03/08/10	BAB I	
2	03/08/10	BAB II ( Tambahkan Landasan Teori )	
3	03/08/10	BAB III ( Perbaiki Penyusunan Tabel )	
4	03/08/10	BAB IV ( Tambahkan Hasil Pengujian )	
5	03/08/10	BAB V	
6	04/08/10	DEMO PROGRAM	
7	06/08/10	REVISI PROGRAM	
8	06/08/10	ACC MAKALAH SEMINAR HASIL	
9	10/08/10	ACC LEMBAR PUBLIKASI	
10	14/08/10	ACC LAPORAN	

Malang,  
Dosen Pembimbing I

(Ir. Yusuf Ismail Nakhoda, MT)

NIP.Y.1018800189



**FORMULIR BIMBINGAN SKRIPSI**

Nama : PEDRICO KARLINOVIANDY DA SILVA  
NIM : 05.12.529  
Masa Bimbingan : 30 Maret 2010 s/d 30 September 2010  
Judul Skripsi : PENGEMBANGAN SISTEM INFORMASI LAYANAN 108 PSTN  
( PUBLIC SWITCH TELEPHONE NETWORK ) MENGGUNAKAN  
SMS GATEWAY

No	Tanggal	Uraian	Paraf Pembimbing
1	19-07-2010	Konsultasi dan Revisi Bab I dan II	
2	23-07-2010	Konsultasi Bab III, Bab I & II Acc	
3	02-08-2010	Revisi Bab III	
4	07-08-2010	Acc Bab III & Konsultasi Bab IV	
5	21-08-2010	Acc Bab IV & Bab V	
6			
7			
8			
9			
10			

Malang,  
Dosen Pembimbing II

**I Komang Somawirata, ST MT**  
**NIP.Y.1030100361**



**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**


Nama : PEDRICO KARLINOVIANDY DA SILVA  
NIM : 05.12.529  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer dan Informatika S-1  
Judul Skripsi : PENGEMBANGAN SISTEM INFORMASI LAYANAN 108 PSTN  
( PUBLIC SWITCH TELEPHONE NETWORK ) MENGGUNAKAN SMS  
GATEWAY

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Senin  
Tanggal : 23 Agustus 2010  
Dengan Nilai : 84,95 (A) / 2


**PANITIA UJIAN SKRIPSI**

**KETUA,**



**Ir. Yusuf Ismail Nakhoda, MT**  
NIP. Y. 1018800189


**SEKRETARIS,**



**Dr. Eng. Aryanto S, ST, MT**  
NIP. Y. 1030800417


**ANGGOTA PENGUJI**

**PENGUJI I,**



**Ibrahim Ashari, ST, MT**  
NIP. P. 1030000365

**PENGUJI II,**



**Sonny Prasetyo, ST, MT**  
NIP. 1031000433



**PERKUMPULAN PENGELOLAH PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG**  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

**FAKULTAS TEKNOLOGI INDUSTRI**  
**FAKULTAS TEKNIK SIPIL DAN PERENCANAAN**  
**PROGRAM PASCA SARJANA MAGISTER TEKNIK**

PT.BNI PERSERO MALANG  
 BANK NIAGA MALANG

KAMPUS I : Jl. Bendungan Sigura-gura No.2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145  
 KAMPUS II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**FORMULIR PERBAIKAN SKRIPSI**

Nama : PEDRICO KARLINOVIANDY DA SILVA  
 NIM : 05.12.529  
 Jurusan : Teknik Elektro S-1  
 Konsentrasi : Teknik Komputer dan Informatika S-1  
 Masa Bimbingan : 30 Maret 2010 s/d 30 September 2010  
 Judul Skripsi : Pengembangan Sistem Informasi Layanan 108 PSTN ( Public Switch Telephone Network ) Menggunakan SMS Gateway

Tanggal	Penguji	Uraian	Paraf
23 Agustus 2010	Penguji I	1. Daftar isi diberi nomor 2. Tambahkan kalimat pengantar pada tabel dan gambar 3. Kesimpulan ditambahkan pengujian sistem	
	Penguji II	1. Abstraksi 2. Latar Belakang 3. Keterangan pada gambar 4. Kesimpulan	25/ 2011 

*Disetujui,*

**PENGUJI I,**

**Ibrahim Ashari, ST. MT**  
 NIP. P. 1030000365

**PENGUJI II,**

**Sonny Prasetyo, ST. MT**  
 NIP. 1031000433

*Mengetahui,*

**Pembimbing I,**

**Ir. Yusuf Ismail Nakhoda, MT**  
 NIP. Y. 1018800189

**Pembimbing II,**

**I Komang Somawirata, ST MT**  
 NIP.Y.1030100361