



UNIVERSITAS INDONESIA

**PERANCANGAN LAYANAN “SEND TO MANY” BERBASIS
VB DALAM MENCEGAH PENYEBARAN KASUS DBD**

SKRIPSI

SUWARDJONO ADIPUTRO

04 05 03 07 45

**FAKULTAS TEKNIK
PROGRAM SARJANA TEKNIK ELEKTRO
DEPOK
JULI 2009**



UNIVERSITAS INDONESIA

**PERANCANGAN LAYANAN “SEND TO MANY” BERBASIS
VB DALAM MENCEGAH PENYEBARAN KASUS DBD**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Teknik**

SUWARDJONO ADIPUTRO

04 05 03 07 45

**FAKULTAS TEKNIK
PROGRAM SARJANA TEKNIK ELEKTRO
DEPOK
JULI 2009**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Suwardjono Adiputro

NPM : 0405030745

Tanda Tangan : 

Tanggal : 1 Juli 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Suwardjono Adiputro
NPM : 0405030745
Program Studi : Elektro
Judul Skripsi : Perancangan Layanan Send To Many Berbasis VB
Dalam Mencegah Penyebaran Kasus DBD

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir. Muhamad Asvial, M.Eng

Penguji : Prof. Dr. Ir. Eko Tjipto Rahardjo, M.Sc

Penguji : Ir. Arifin Djauhari, MT

Ditetapkan di : Depok

Tanggal : 1 Juli 2009

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan atas Kehadirat Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan buku skripsi ini dengan baik.

Penulisan buku skripsi ini dimaksudkan untuk melaporkan kepada pembimbing khususnya atas hasil bimbingan yang telah dilakukan, dan para dosen penguji umumnya untuk melakukan pengujian terhadap penulis dalam mata kuliah skripsi. Selain itu penulisan buku skripsi ini dilakukan sebagai salah satu syarat kelulusan untuk menjadi seorang sarjana teknik.

Ucapan terima kasih juga penulis tujukan kepada pihak-pihak yang telah memberikan dukungan dan motivasi kepada penulis, di antaranya :

1. Dr. Ir. Muhamad Asvial, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan skripsi ini,
2. Orang tua, yang telah memberikan dukungan baik secara moril maupun spiritual, dan
3. Rekan-rekan mahasiswa.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Suwardjono Adiputro
NPM : 0405030745
Program Studi : Elektro
Departemen : Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

PERANCANGAN LAYANAN “SEND TO MANY” BERBASIS VB DALAM
MENCEGAH PENYEBARAN KASUS DBD

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 1 Juli 2009

Yang menyatakan



(Suwardjono Adiputro)

ABSTRAK

Nama : Suwardjono Adiputro
Program Studi : Teknik Elektro
Judul : Perancangan Layanan “Send To Many” Berbasis VB Dalam Mencegah Penyebaran Kasus DBD

Kasus DBD sampai saat ini masih mengalami peningkatan. Hal ini dapat terjadi karena kurang cepatnya penyampaian kasus DBD kepada masyarakat sekitar. Oleh karena itu diperlukan cara untuk memberikan informasi secara cepat mengenai kasus DBD kepada masyarakat sekitar. Cara tersebut adalah dengan memanfaatkan layanan sms pada handphone. Proses pengiriman pesan dilakukan secara send to many yang memanfaatkan metode looping. Hal yang diperlukan adalah system database kelurahan dan system pengiriman pesan. Dari system kelurahan akan didapatkan nomor ID untuk setiap individu. Nomor ID inilah yang akan menjadi input pada system pengiriman pesan. Handphone hanya digunakan sebagai pintu gerbang keluarnya pesan. Dengan layanan send to many ini pesan dapat sampai di beberapa nomor tujuan dalam waktu yang relative singkat.

Kata kunci :

AT Command, nomor ID, database, layanan send to many

ABSTRACT

Name : Suwardjono Adiputro
Programme of Study : Electrical Engineering
Title : Design of Send To Many Service Based On VB to
Protect DBD Divergence

Nowadays, DBD cases still increase. This can occur because of less rapid delivery DBD case to the people around. Therefore it is needed a way to quickly provide information about the case to the people around. The way is done by using the sms service on mobile phones. The delivery message process is done using a send to many way which is using a looping method. The required thing are the database system and information system of DBD case. The database system products ID number of every individual. From ID number, we can get the individual information. Then, This number will be used as input into the information system of DBD. Handphone is just used as a gate to send a message. By Send to many service, a message will be received at the some destination number in the relatively short time.

Key word :

AT-Command, nomor ID, send to many service, database

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMAKASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR SINGKATAN	xii
BAB 1 PENDAHULUAN	1
1.1 LatarBelakang	1
1.2 Tujuan	2
1.3 Batasan Masalah	2
1.4 Sistematika Penulisan	3
BAB 2 SMS (Short Message Service).....	4
2.1 Arsitektur GSM.....	4
2.1.1 Mobile Station (MS)	5
2.1.2 Base Transmision Subsystem (BTS)	5
2.1.3 Base Station Controller (BSC).....	5
2.1.4 Home Location Register (HLR).....	6
2.1.5 Visitor Location Register (VLR)	6
2.1.6 Mobile Switching Centre (MSC).....	6
2.2 Arsitektur SMS	6
2.2.1 Elemen Utama Pada Arsitektur SMS.....	7
2.2.2 Basic Feature SMS.....	8
2.2.3 SMS Protocol Layer.....	9
2.3 AT Command.....	9
2.3.1 PDU Mode	15
2.3.2 Sistem Bilangan	18
2.4 Database.....	20

2.5 Dasar Pemrograman VB	21
BAB 3 PERANCANGAN LAYANAN SEND TO MANY	23
3.1 Komponen Perancangan	23
3.2 Koneksi PC to HP	23
3.3 Perancangan Database	24
3.4 SMS Sender	27
3.4.1 Diagram Alir	28
BAB 4 ANALISIS PROGRAM dan UJICOBA.....	31
4.1 Analisis Database.....	31
4.1.1 Analisis Database System Kelurahan	31
4.1.1.1 Analisis Data Kelurahan	32
4.1.1.2 Analisis Data RW	42
4.1.1.3 Analisis Data RT.....	45
4.1.1.4 Analisis Data KK.....	46
4.1.1.5 Analisis Data Individu	46
4.1.2 Analisis Database System Informasi DBD.....	47
4.2 Analisis SMS Sent	49
4.2.1 Mengatur Port Serial	49
4.2.2 Melakukan Settingan MsComm.....	49
4.2.3 Konversi Text Mode Menjadi PDU Mode	50
4.2.4 Koneksi Antara Database MySQL dan VB	53
4.2.5 Send To Many.....	53
BAB 5 KESIMPULAN.....	58
DAFTAR ACUAN	59
DAFTAR PUSTAKA	60
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 AT – Command	10
Tabel 2.2 Format PDU	16
Tabel 2.3 Rumus Validity Period (Vp).....	17
Tabel 2.4 Kode ASCII.....	17
Tabel 2.5 Convert Text to PDU	18
Tabel 2.6 Konversi Decimal-Biner (a), Decimal-octal (b), dan Decimal-Heksadecimal	18
Tabel 2.7 Konversi Biner-Decimal (a), Biner-octal (b), dan Biner-Heksadecimal	19
Tabel 2.8 Konversi Octal-Decimal (a), octal (b)-Biner, dan Octal-Heksadecimal.....	19
Tabel 2.9 Konversi Heksadecimal-Decimal (a), Heksadecimal-Biner (b), dan Heksadecimal-Octal	19
Tabel 3.1 Tabel Sementara Nomor Tujuan.....	30
Tabel 3.2 Tabel Sementara Pesan.....	30

DAFTAR GAMBAR

Gambar 2.1 Arsitektur GSM.....	5
Gambar 2.2 New Connection.....	11
Gambar 2.3 Pemilihan ComPort.....	12
Gambar 2.4 Port Setting.....	13
Gambar 2.5 Window Hyperterminal	14
Gambar 2.6 Intruksi AT-CMGEF	15
Gambar 2.7 Window Microsoft Visual Basic 6.0.....	21
Gambar 2.8 Toolbox Microsoft Visual Basic 6.0	22
Gambar 3.1 Koneksi PC to HP to MultiUser.....	23
Gambar 3.2 Diagram Alir SMS Sender	28
Gambar 4.1 Database System Kelurahan	31
Gambar 4.2 Database Kelurahan	32
Gambar 4.3 Database RW	42
Gambar 4.4 Entry Database RW	43
Gambar 4.5 Database RT	45
Gambar 4.6 Database KK	46
Gambar 4.7 Database Individu	46
Gambar 4.8 Database System Informasi DBD	47
Gambar 4.9 Komponen MsComm	49
Gambar 4.10 Sistem Informasi DBD	54
Gambar 4.11 Proses pengiriman Informasi DBD.....	56
Gambar 4.12 Hasil Pengiriman Informasi di Handphone Penerima.....	57

DAFTAR SINGKATAN

1. *AT- Command* = *Attention Command*
2. *BASIC* = *Beginners All-Purpose Symbolic Instruction Code*
3. *BSC* = *Base System Controlled*
4. *BSS* = *Base Station Subsystem*
5. *BTS* = *Base Transceiver Subsystem*
6. *CDMA* = *Code Division Multiple Access*
7. *CSPDN* = *Circuit Switched Public Data Network*
8. *DBD* = *Demam Berdarah Dengue*
9. *ESME* = *External Short Message Service*
10. *GSM* = *Global System For Mobile Communications*
11. *GUI* = *Graphical User Interface*
12. *HP* = *Handphone*
13. *HLR* = *Home Location Register*
14. *ISDN* = *Integrated Service Digital Network*
15. *ME* = *Mobile Equipment*
16. *MS* = *Mobile Station*
17. *MSC* = *Mobile Switching Centre*
18. *NSS* = *Network Subsystem*
19. *OSS* = *Operation Subsystem*
20. *PC* = *Personal Computer*
21. *PDU* = *Paket Data Unit*
22. *PSPDN* = *Packet Switched Public Data Network*
23. *PSTN* = *Public Switched Telephone Network*
24. *RDBMS* = *Relational Database Management System*
25. *SIM* = *Subscriber Identity Module*
26. *SME* = *Short Message Entity*
27. *SMS* = *Short Message Service*
28. *SMSC* = *Short Message Service Centre*
29. *SM – AL* = *Short Message – Application Layer*
30. *SM – LL* = *Short Message – Link Layer*

31. *SM – MO* = *Short Message – Mobile Originated*
32. *SM – MT* = *Short Message – Mobile Terminated*
33. *TE* = *Terminal Equipment*
34. *VB* = *Visual Basic*
35. *VLR* = *Visitor Location Register*
36. *VMS* = *Voice Mail System / Server*
37. *VP* = *Validity Period*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

DBD merupakan penyakit yang disebabkan oleh gigitan nyamuk *Aedes Aegypti* (nyamuk malaria). Penyakit DBD ini dapat menular melalui gigitan nyamuk lain yang menggigit penderita DBD. Di Indonesia, penyakit ini masih terus meningkat. Pada hakikatnya, cara yang paling efektif dalam melakukan pencegahan terhadap penyebaran kasus DBD ini adalah dengan memutus siklus hidup nyamuk *Aedes Aegypti* dan nyamuk biasa sebagai perantaranya. Masyarakat dapat melakukan usaha tersebut dengan cara kerja bakti secara berkala khususnya membersihkan saluran air, penampungan air, dan tempat-tempat lain yang memungkinkan nyamuk dapat berkembangbiak. Namun salah satu kendalanya adalah kurangnya kesadaran masyarakat untuk melakukan itu semua. Pada umumnya masyarakat akan melakukan hal tersebut setelah beberapa kasus DBD telah melanda di daerahnya. Hal ini dapat terjadi karena lambatnya penyampaian informasi kasus DBD ke masyarakat sekitar. Oleh karena itu, untuk menyampaikan informasi kasus DBD secara cepat, maka diperlukan suatu aplikasi yang *user friendly* yang mampu mengirimkan pesan kasus DBD dan dapat mudah dimengerti dalam pengoperasiannya.

Untuk memenuhi kriteria tersebut, maka aplikasi yang tepat adalah aplikasi yang berbasis sms. Dengan sms, penyampaian pesan akan cepat diterima ke tujuan dalam hitungan menit atau bahkan detik. Selain itu sms, sebagai salah satu layanan dalam handphone baik GSM maupun CDMA, telah meluas di Indonesia pada seluruh lapisan masyarakat. Hal ini dikarenakan kesederhanaan fitur-fiturnya, dan kemudahan dalam penggunaannya. Sistem informasi semacam ini telah dibuat oleh seorang dokter yang bernama Amnur Kayo. Beliau memanfaatkan proses penyampaian informasi dengan menggunakan layanan sms pada handphone yang dihubungkan ke pc, tujuannya adalah untuk dapat menampung data yang jauh lebih banyak dari memory sebuah handphone. Namun system yang beliau gunakan masih memanfaatkan sistem point to point sms yang berarti informasi yang dikirimkan hanya diperuntukkan oleh satu orang. Jika ingin mengirimkan lagi kepada orang lain maka dilakukan proses pengiriman seperti awal

kembali. Informasi mengenai DBD dibutuhkan oleh banyak pihak di lingkungan sekitar korban, oleh karena itu pemanfaatan sistem pengiriman pesan seperti ini kurang efektif.

Untuk itu penulis hendak mengembangkan system tersebut sehingga dalam satu kali proses pengiriman, pesan dapat sampai ke beberapa nomor tujuan. Proses ini dapat dikenal dengan istilah layanan "send to many ". Proses ini dapat terjadi dengan cara melakukan pembuatan system database kelurahan yang menghasilkan nomor ID bagi tiap individu, sehingga tidak ada individu yang memiliki nomor yang sama. Nomor ID akan merepresentasikan lokasi tempat tinggal individu tersebut. Dari nomor ID akan di pilih database tokoh-tokoh masyarakat (Lurah, RW, RT, KK) beserta nomor handphonenya sebagai objek tujuan pengiriman sms kasus DBD, dengan asumsi kasus DBD terjadi di lokasi tempat tinggal individu tersebut berada. Jadi hal utama dalam perancangan layanan send to many ini adalah system database dan system sms. System database berguna untuk menyimpan data-data pasien korban kasus DBD, dan system sms berguna untuk melakukan pengiriman informasi kasus DBD ke multiuser. Pemanfaatan system sms, yang penulis buat, memanfaatkan teknologi GSM yang sudah familiar di kalangan masyarakat. Hal ini dikarenakan oleh jangkauannya yang cukup luas. "Teknologi GSM sampai saat ini paling banyak digunakan di Dunia juga di Indonesia karena salah satu keunggulan dari GSM adalah kemampuan roaming yang luas sehingga dapat dipakai di berbagai Negara. Akibatnya mengalami pertumbuhan yang sangat pesat." [1]

1.2 Tujuan

Tujuan dari skripsi ini adalah membuat suatu aplikasi yang mampu mengirimkan sebuah pesan (kasus DBD) ke beberapa nomor tujuan dalam waktu yang relative singkat.

1.3 Batasan Masalah

Batasan masalah skripsi ini adalah :

1. Kasus DBD terjadi di tempat tinggal pasien DBD
2. Pengiriman pesan kasus DBD secara multiuser ditujukan kepada lurah, ketua rw, ketua rt, dan kk di sekitar tempat tinggal pasien.
3. Handphone berfungsi sebagai media pengiriman pesan melalui pc.

1.4 Sistematika Penulisan

Skripsi ini disusun dengan menggunakan metoda studi literatur dan simulasi dengan sistematika penulisan yang terbagi dalam lima bab. Sistematika penulisan sebagai berikut: Bab satu adalah pendahuluan yang menjelaskan mengenai latar belakang, tujuan, batasan masalah, dan sistematika penulisan. Bab dua menjelaskan teori-teori yang berhubungan dengan perancangan layanan send to many yang penulis buat. Bab tiga menjelaskan mengenai tahap-tahap perancangan layanan send to many dan beberapa komponen yang digunakan. Bab empat adalah analisis program dan ujicoba dari rancangan layanan send to many yang penulis buat. Bab lima adalah kesimpulan.

BAB 2

SMS (Short Message Service)

2.1 Arsitektur GSM

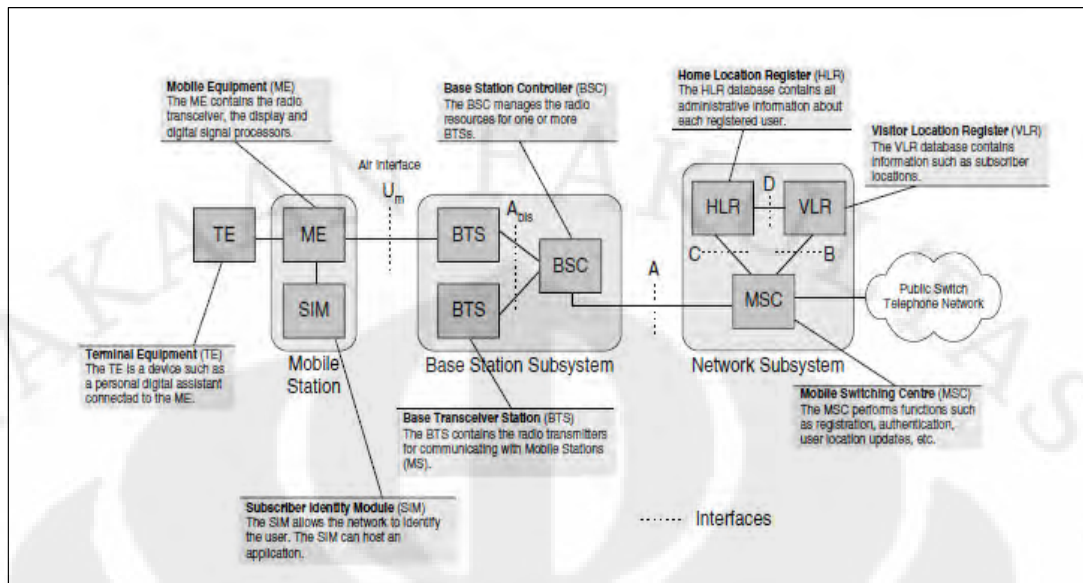
Global System For Mobile Communications (GSM) merupakan jaringan global yang digunakan untuk melakukan manajemen dalam komunikasi bergerak.

Unsur-unsur utama dalam arsitektur GSM ditunjukkan pada **Gambar 2.1**, terdiri atas tiga subsistem, yaitu

- a. Base Station Subsystem (BSS),
- b. Network Subsystem (NSS) dan
- c. Operation Subsystem (OSS).

OSS tidak dijelaskan lebih lanjut, unsur-unsur BSS dan NSS akan diuraikan lebih lanjut.

Berikut adalah gambar arsitektur jaringan GSM :



Sumber : Ekawati Prihatini
 Gambar 2.1 Arsitektur GSM

2.1.1 Mobile Station (MS)

MS adalah perangkat yang digunakan untuk mengirim dan menerima signal radio. MS dapat berupa Mobile Handset atau Personal Digital Assistant (PDA). MS terdiri dari Mobile Equipment (ME) dan Subscriber Identity Module (SIM). ME berisi transceiver radio, display dan digital signal processor. SIM sebagai identitas dari user sehingga network dapat mengenali user tersebut.

2.1.2 Base Transceiver Station (BTS)

BTS berfungsi sebagai interface komunikasi antar MS yang aktif dan berada dalam coverage area BTS tersebut. Di dalamnya termasuk modulasi signal, demodulasi, equalize signal dan error coding. Beberapa BTS terhubung pada satu BSC. Satu BTS biasanya mampu handle 20-40 komunikasi secara serentak.

2.1.3 Base Station Controller (BSC)

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

2.1.4 Home Location Register (HLR)

HLR adalah sebuah database yang berfungsi sebagai tempat penyimpanan informasi secara permanen mengenai data pelanggan / subscribers. Elemen ini berisi detail dari setiap subscriber. Sebuah HLR biasanya mampu mengatur ratusan bahkan ribuan subscribers.

2.1.5 Visitor Location Register (VLR)

VLR merupakan sebuah database tempat penyimpanan informasi sementara pelanggan dari sebuah HLR yang sedang roaming pada HLR lain. Informasi ini dibutuhkan oleh MSC untuk dapat melayani pelanggan yang sedang roaming tersebut. VLR ini berisi informasi dinamis tentang user yang terkoneksi dengan mobile network termasuk lokasi user tersebut. VLR biasanya terintegrasi dengan MSC.

2.1.6 Mobile Switching Centre (MSC)

MSC merupakan sebuah sistem yang melakukan fungsi switching dan pengontrolan panggilan telepon dalam sebuah jaringan komunikasi bergerak. MSC inilah yang akan mengirimkan sebuah short message ke suatu tujuan tertentu melalui base station yang sesuai. MSC melakukan fungsi registrasi, autentikasi, update lokasi user, billing service dan sebagai interface dengan jaringan lain. Selain itu MSC juga bertanggung jawab untuk call set-up, release dan routing. Melalui MSC, mobile network terhubung dengan jaringan lain seperti PSTN, ISDN, CSPDN ,dan PSPDN

2.2 Arsitektur SMS

Short Message Service (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel. Hal ini memungkinkan pengiriman pesan dalam bentuk alphanumeric antar terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti email, paging, voice mail, dan lain-lain.

SMS adalah protokol layanan pertukaran pesan text singkat (sebanyak 160 karakter per pesan) antar mobile phone. SMS tidak hanya melalui layanan sistem jaringan komunikasi GSM tetapi juga melalui layanan sistem standar komunikasi lain seperti CDMA, UMTS dan bahkan melalui jaringan telepon rumah (Fixed phone).

2.2.1 Elemen Utama pada Arsitektur SMS

Elemen utama pada arsitektur sms adalah sebagai berikut :

a. Short Message Entity (SME)

SME adalah elemen yang dapat mengirim atau menerima pesan singkat. SME dapat berupa software aplikasi pada mobile handset, dapat juga berupa perangkat facsimile, perangkat telex, remote internet server, dll. Sebuah SME dapat berupa server yang terkoneksi dengan SMS centre secara langsung atau melalui gateway. Dapat dikatakan bahwa SME merupakan entitas dalam sistem sms yang dapat berada di dalam jaringan berupa perangkat bergerak, atau merupakan service center yang berada di luar jaringan. ESME (External Short Message Entity) merupakan SME yang berada di luar jaringan sms. Beberapa macam ESME di antaranya adalah :

1. Voice Mail System / Server (VMS)
2. Web, WAP proxy / server
3. Email Gateway

b. SMS Service Centre (SMSC)

SMS Service Centre (SMSC) memegang peran kunci dalam arsitektur SMS. Fungsi utama SMSC adalah menyampaikan pesan singkat antara SME dengan MS, juga menyimpan, routing, dan meneruskan pesan singkat (menyimpan pesan jika penerima SME tidak tersedia) dari terminal sumber ke terminal tujuan. SMSC dapat terintegrasi sebagai bagian dari mobile network (terintegrasi dengan MSC) atau sebagai entitas network independen. Terminologi SMSC mengacu pada sesuatu yang berupa hardware dan software.

c. Email Gateway

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

2.2.2 Basic Features SMS

SMS mempunyai beberapa basic feature, seperti :

a. Message Submission and Delivery

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

Pada message delivery, pesan disampaikan oleh SMSC ke MS. Dikenal sebagai Short Message Mobile Terminated (SM-MT). SM-MO dan SM-MT dapat dikirim / diterima saat voice call atau koneksi data sedang berlangsung.

b. Status Report

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

c. Reply Path

Reply Path dapat diatur oleh SME asal (atau SMSC serving) untuk mengindikasikan bahwa SMSC serving mampu untuk handle secara langsung reply dari SME penerima.

d. Addressing Mode

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

e. Validity Period

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

2.2.3 SMS Protocol Layer

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

a. Application Layer

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

b. Transfer Layer

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

c. Relay Layer

Relay layer mengizinkan pengiriman pesan antar elemen network yang berbeda. Sebuah elemen network menyimpan pesan sementara jika elemen berikutnya dimana pesan akan diforward tidak tersedia.

d. Link Layer

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

2.3 AT Command

AT Command merupakan kependekkan dari attention command. “AT Command adalah perintah-perintah yang digunakan dalam komunikasi dengan serial port” (adipurnomo). AT Command merupakan suatu perintah yang dapat dimengerti oleh sebuah handphone. Dalam perancangan layanan send to many ini, penulis menggunakan HP jenis sony ericcson tipe W200i. Pada umumnya untuk setiap jenis Handphone memiliki jenis AT Command yang sama.

Pada umumnya AT command dapat digunakan untuk:

- a. melakukan konfigurasi handphone via kabel USB, infrared port, atau bluetooth

- b. mengetahui informasi tentang operational status dari sebuah handphone/ modem
Berikut adalah beberapa contoh AT Command beserta fungsinya masing-masing:

Tabel 2.1 AT-Command

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+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 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

AT Command	Keterangan
	ME = Memori HP
	SM = Memori SIM CARD

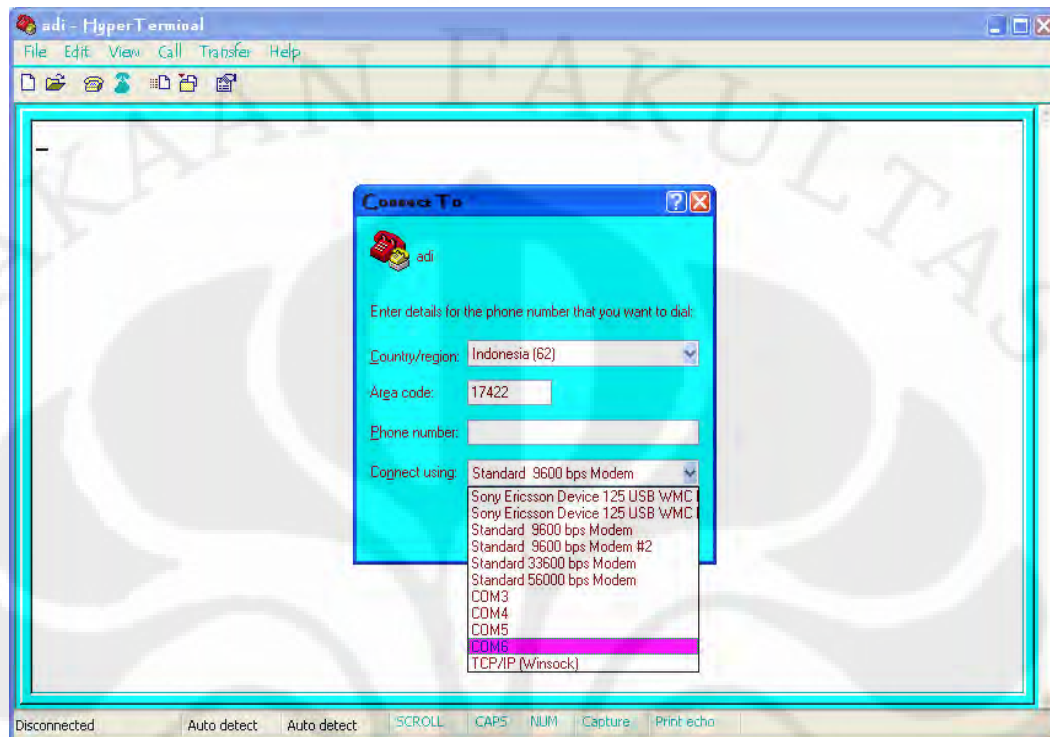
Sumber : <http://javaku.wordpress.com>

Dalam perancangan layanan send to many ini, penulis hanya menggunakan perintah AT + CMGF, AT + CMGS, dan AT + CMGS. AT Command ini dijalankan pada Hyperterminal yang sudah disediakan oleh windows. Untuk membukanya cukup melakukan klik **Start-All Program-Accessories-Communication-Hyperterminal**. Hyperterminal merupakan terminal yang digunakan untuk melakukan komunikasi antara dua device. Setelah mengklik hyperterminal maka akan muncul tampilan sebagai berikut :



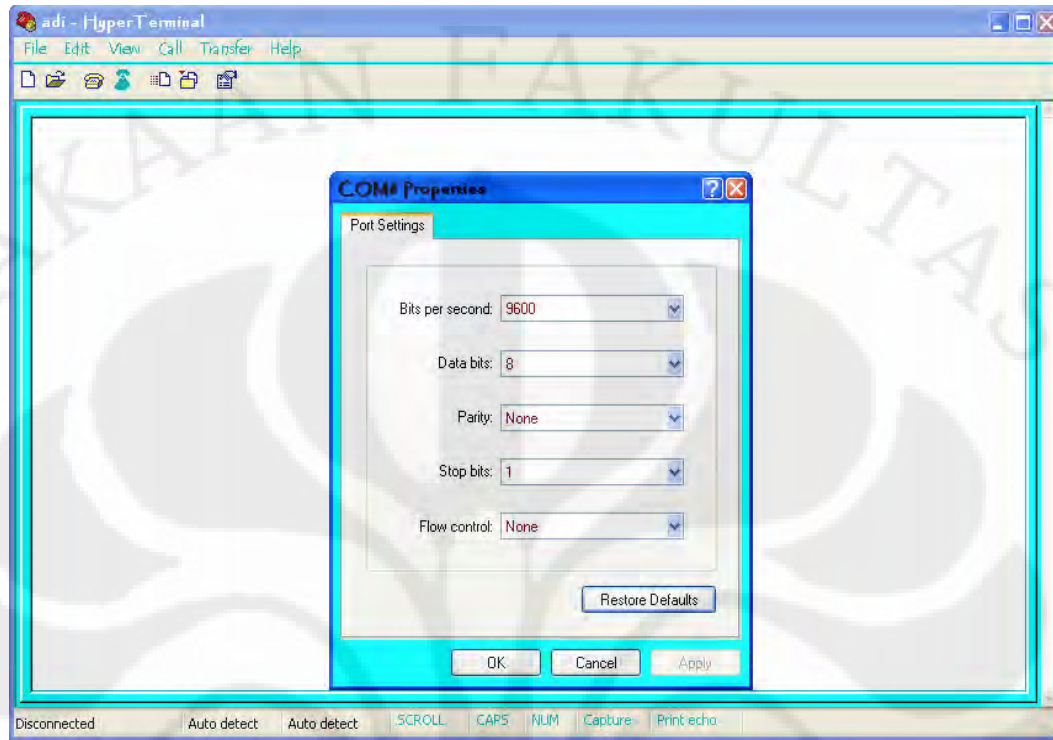
Gambar 2.2 New Connection

Tuliskan nama dan icon yang diinginkan. Setelah itu akan muncul tampilan sebagai berikut:



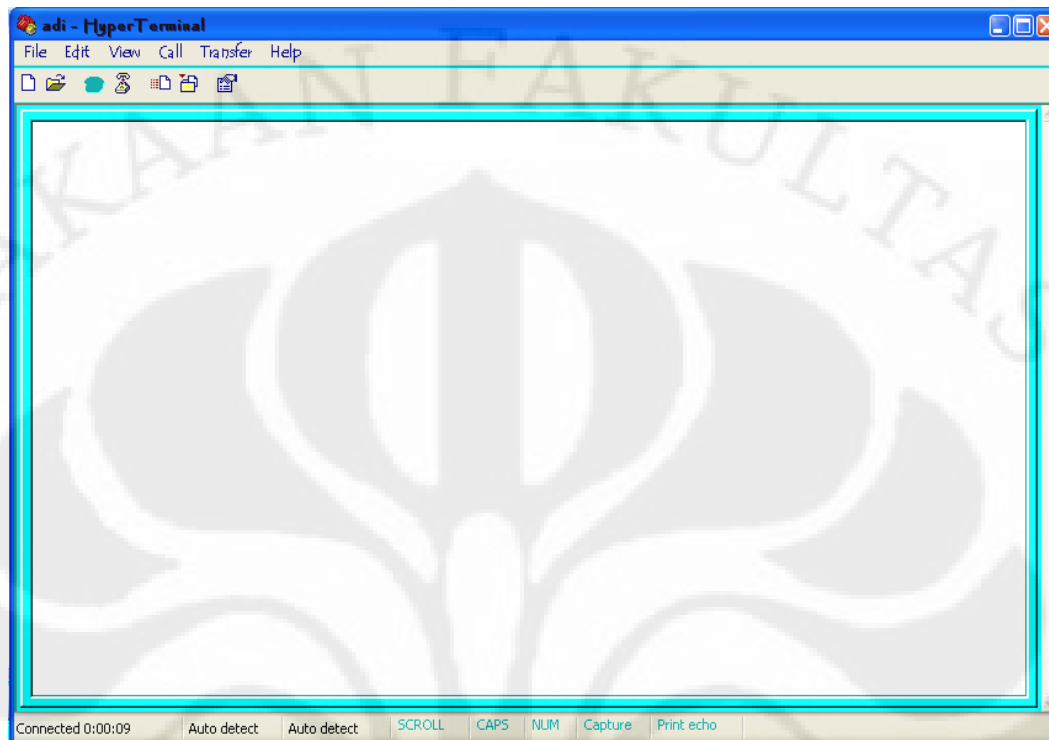
Gambar 2.3 Pemilihan ComPort

Dari tampilan ini yang penting adalah “connect using”. Jika koneksi antara computer dan handphone telah terhubung, maka akan tersedia port yang dapat digunakan untuk melakukan koneksi diantara keduanya. Pada perancangan layanan send to many ini penulis menggunakan kabel data untuk menghubungkan handphone ke pc. Dan comport yang digunakan adalah Com6



Gambar 2.4 Port Setting

Intruksi di atas meminta user untuk melakukan peyettingan port yang digunakan. Tekan restore default jika tidak mengetahui settingan-annya. Jika berhasil maka status “disconnect” akan berubah menjadi “connect”, artinya koneksi antara handphone dan pc sudah berhasil. Seperti tampilan berikut :



Gambar 2.5 Window Hyperteminal

Langkah selanjutnya adalah menuliskan perintah AT Command.

```

adi - HyperTerminal
File Edit View Call Transfer Help
at
OK
at+cmgf=?
+CMGF: (0)
OK
Connected 0:00:19 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

```

Gambar 2.6 Intruksi AT + CMGF

Dari tampilan di atas dapat dilihat, jika diketikkan “at” kemudian direspon “ok” maka artinya koneksi antara pc dan handphone telah berhasil dan siap dilakukan konfigurasi. Kemudian baris berikutnya diketikkan perintah “ at + cmgf ” yang berarti user ingin mengetahui bagaimana jenis mode yang digunakan oleh handphone. Perlu diketahui bahwa setiap handphone memiliki dua jenis mode yaitu mode text dan mode PDU. Ada handphone yang mendukung dua jenis mode tersebut, namun ada pula yang hanya memiliki salah satu jenis mode tersebut. Untuk mengetahuinya dapat dituliskan perintah “ at + cmgf ” seperti di atas. Jika direspon (0) maka berarti handphone yang digunakan hanya mendukung pdu mode, sedangkan jika direspon (1) maka handphone yang digunakan mendukung text mode. Jika direspon (0,1) maka berarti handphone tersebut dapat mendukung pdu mode maupun text mode. Dari tampilan di atas dapat dilihat bahwa handphone yang digunakan penulis hanya mendukung pdu mode. Mengenai pdu mode maupun text mode akan dijelaskan pada subbab berikut.

2.3.1 PDU Mode

Seperti telah dijelaskan sebelumnya bahwa setiap jenis handphone memiliki

dua jenis mode yang digunakan dalam proses pengiriman maupun penerimaan sms. Mode tersebut adalah mode text dan mode PDU. Mode text berarti penulisan pesan maupun nomor tujuan yang digunakan dalam pengiriman maupun penerimaan sms, dapat dilakukan dengan penulisan text seperti biasa. Berbeda dengan mode text, maka penulisan pesan maupun nomor tujuan dengan menggunakan mode PDU harus diubah ke dalam format PDU. PDU merupakan kependekkan dari paket data unit.

Berikut adalah contoh rangkaian PDU untuk mengirimkan sms dengan pesan “SMS Dengan VB” ke nomor tujuan 628561552200

Format PDU :

```
07912658050000F011000C912658165522000000AB0DD3E614442CBBCF6137C82
A04
```

Penjelasan format PDU :

Table 2.2 Format PDU

07	Panjang kode internasional ditambah nomor SMSC pengirim
91	kode yang menunjukkan bahwa penomoran menggunakan format penomoran internasional (62855000000), jika menggunakan penomoran nasional (0855000000) maka menggunakan kode 81
2658050000F0	Nomor SMSC pengirim yang dibalik setiap dua digit, karena ganjil maka nomor tersebut ditambah F
1100	“11 merupakan kode dalam hexa untuk pengiriman sms, 00 berarti membiarkan pengaturan pesan dilakukan sendiri oleh handphone tujuan”. [2]
0C	panjang kode internasional ditambah nomor handphone tujuan
91	kode yang menunjukkan bahwa penomoran menggunakan format penomoran internasional
265816552200	Nomor handphone tujuan
0000	00 berarti jenis pesan yang dikirim berupa text, “00 pesan SMS yang dikirim berupa teks standart” [2]
AB	Nilai validity period = 5 hari dalam hexa
0D	Panjang pesan heksadesimal

D3E614442CBBCF6137C82A04	Pesan berupa "SMS Dengan VB"
--------------------------	------------------------------

Berikut adalah rumus menghitung validity period (Vp)

Tabel 2.3 Rumus validity period (Vp)

Waktu VP	Nilai Validitas Periode
5 menit – 720 menit (12 jam)	$(\text{Waktu VP} / 5) - 1$
12,5 jam – 24 jam	$143 + ((\text{Waktu VP} - 12) * 2)$
2 – 30 hari	$166 + \text{Waktu VP}$
Lebih dari 4 minggu	$192 + \text{Waktu VP}$

Sumber : <http://javaku.wordpress.com>

Isi pesan “SMS Dengan VB” harus diubah ke dalam format PDU. Setiap karakter terlebih dahulu harus diubah kedalam kode ASCII kemudian diubah menjadi biner (7 bit). 7 bit ini kemudian diubah menjadi 8 bit dengan menambahkan 1 bit dari karakter sesudahnya. Berikut adalah kode ASCII yang digunakan untuk mengubah karakter menjadi decimal.

Tabel 2.4 Kode ASCII

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

Sumber : <http://javaku.wordpress.com>

Tabel 2.5 Convert Text To PDU

Nilai	Dec	Septet (7 bit)	Octet (8 bit)	Hasil
S	83	1010011	11010011	D3
M	77	1001101	11100110	E6
S	83	1010011	00010100	14
Spasi	32	0100000	01000100	44
D	68	1000100	00101100	2C
E	101	1100101	10111011	BB
n	110	1101110	11001111	CF
g	103	1100111		
a	97	1100001	01100001	61
n	110	1101110	00110111	37
Spasi	32	0100000	11001000	C8
V	86	1010110	00101010	2A
B	66	1000010	00000100	04

2.3.2 Sistem Bilangan

Pada umumnya manusia hanya mengenali system bilangan decimal, namun terdapat beberapa sistem bilangan lain seperti sistem bilangan biner, sistem bilangan octal, sistem bilangan heksadesimal

1. Konversi Sistem Bilangan Desimal ke Biner, Octal, Heksadesimal

Contoh : 36 desimal

Tabel 2.6 Konversi Desimal-Biner (a), Desimal-Octal (b), dan Desimal-Heksadesimal (c)

(a) Desimal – Biner

$36 \div 2 \rightarrow 18 \text{ sisa } 0$
$18 \div 2 \rightarrow 9 \text{ sisa } 0$
$9 \div 2 \rightarrow 4 \text{ sisa } 1$
$4 \div 2 \rightarrow 2 \text{ sisa } 0$
$2 \div 2 \rightarrow 1 \text{ sisa } 0$
1
Jadi $36 = 100100_2$

(b) Desimal – Octal

$36 \div 8 \rightarrow 4 \text{ sisa } 4$
4
Jadi $36 = 42_8$

(c) Desimal – Heksadesimal

$36 \div 16 \rightarrow 2 \text{ sisa } 4$
2
Jadi $36 = 24_{16}$

2. Sistem Bilangan Biner

Tabel 2.7 Konversi Biner-Decimal (a), Biner-Octal (b), dan Biner-Hexadecimal (c)

- (a) Biner – Decimal
 $100100_2 = 1x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 36$
- (b) Biner – Octal = Biner – Decimal – Octal
- (c) Biner – Heksadecimal = Biner – Decimal – Heksadecimal

3. Sistem Bilangan Octal

Tabel 2.8 Konversi Octal-Decimal (a), Octal-Biner (b), dan Octal-Heksadecimal (c)

- (a) Octal – Decimal
 $42 = 4x8^1 + 2x8^0 = 36$
- (b) Octal – Biner = Octal – Decimal – Biner
- (c) Octal - Heksadecimal = Octal – Decimal – Heksadecimal

4. Sistem Bilangan Heksadesimal

Tabel 2.9 Konversi Heksadecimal-Decimal (a), Heksadecimal-Biner (b), dan Heksadecimal-Octal (c)

- (a) Heksadecimal – Decimal
 $24 = 2x16^1 + 4x16^0 = 36$
- (b) Heksadecimal – Biner = Heksadecimal – Decimal – Biner
- (c) Heksadecimal – Octal = Heksadecimal – Decimal – Octal

2.4 Database

Salah satu keuntungan membuat aplikasi layanan send to many ini adalah kemampuan untuk menampung jumlah data baik berupa nomor tujuan, data sms yang dikirim maupun diterima dalam jumlah yang lebih banyak jika dibandingkan dengan menggunakan aplikasi handphone biasa. Hal ini jelas dikarenakan oleh kapasitas memori dalam computer yang lebih banyak dibandingkan handphone. Data yang digunakan dalam aplikasi ini diatur dengan menggunakan system database. Dengan Database ini data yang digunakan dalam aplikasi ini dikelompokkan ke dalam tabel-tabel. System database yang digunakan adalah system database relasional atau yang lebih dikenal dengan *RDBMS (Relational DataBase Management System)*. Seperti namanya, dalam system ini terdapat relasional antara tabel yang satu dengan tabel yang lain. Untuk dapat tercipta suatu relasional di antara tabel-tabel tersebut maka harus terdapat suatu relasi (penghubung) di antara tabel tersebut.

Pada database relasional ada tiga jenis model relasi, yaitu :

1. Hubungan *one-to-one*, artinya pada masing-masing tabel hanya terdapat satu data yang saling berhubungan.
2. Hubungan *one-to-many*, artinya pada sebuah tabel berelasi dengan banyak record pada tabel yang lain.
3. Hubungan *many-to-many*, artinya banyak record pada sebuah tabel berhubungan dengan banyak record pada tabel yang lain.[3]

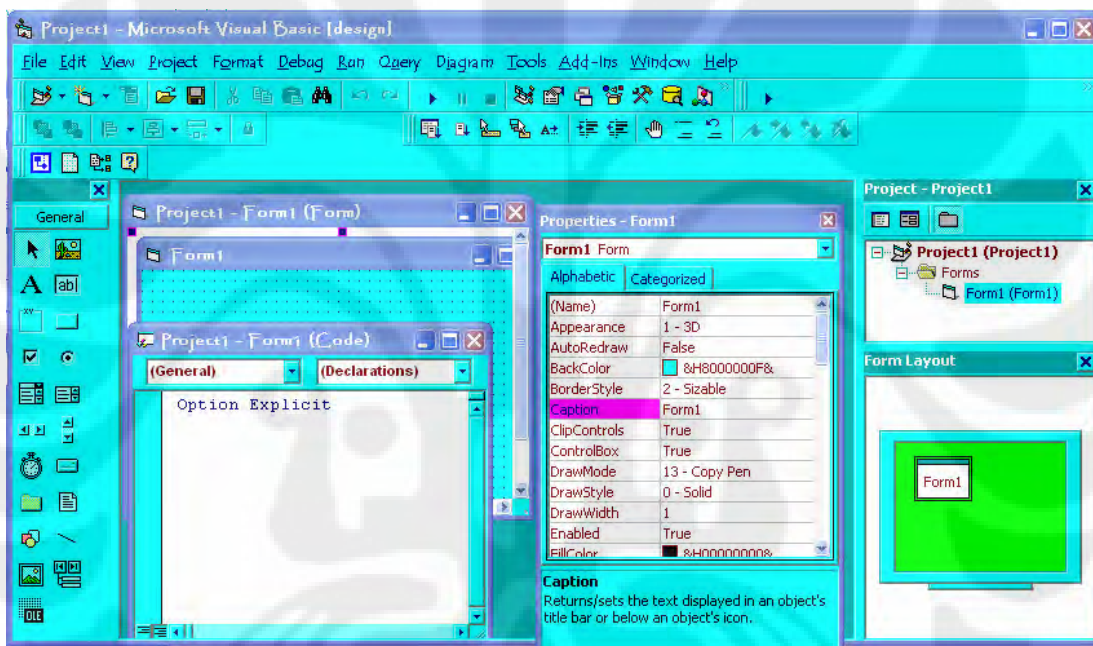
System database yang dibuat dalam perancangan layanan ini adalah dengan menggunakan database MySQL. Pada umumnya, “semua perintah pengolahan data pada MySQL hanya terdiri dari empat jenis permintaan” [3] di antaranya :

1. Menambah data ke dalam tabel, perintah yang digunakan adalah :
INSERT INTO namatabel (kolom1, kolom2,...) VALUES (nilai1, nilai2....)
2. Mengedit data yang telah ada, perintah yang digunakan adalah :
UPDATE namatabel SET namafield = ekspresi WHERE definisi
3. Menampilkan data, perintah yang digunakan :
SELECT * FROM namatabel
4. Menghapus data, perintah yang digunakan :
DELETE FROM namatabel

2.5 Dasar Pemrograman VB

Visual BASIC merupakan sebuah bahasa pemrograman yang dapat digunakan untuk membuat suatu aplikasi dalam Microsoft Windows. Visual BASIC menggunakan metode *Graphical User Interface* (GUI) dalam pembuatan program aplikasi (*project*). Istilah visual mengacu pada metode pembuatan tampilan program (Interface) atau objek pemrograman yang biasa dilakukan secara langsung terlihat oleh programmer.

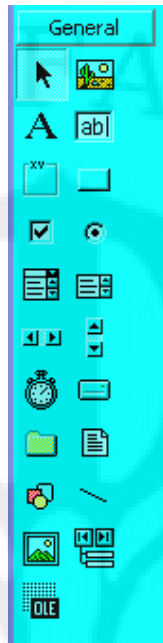
Berikut adalah tampilan window Microsoft visual basic 6.0 :



Gambar 2.7. Window Microsoft Visual Basic 6.0

Window Microsoft visual basic terdiri atas dua window, yaitu window form dan window code. Window form digunakan untuk membuat interface dengan user sedangkan window code berisi program dari interface tersebut.

Untuk membuat interface dengan user, berikut toolbox standard yang sering digunakan :



Gambar 2.8 Toolbox Microsoft Visual Basic 6.0

BAB 3 PERANCANGAN LAYANAN SEND TO MANY

3.1 Komponen Perancangan

Dalam melakukan perancangan aplikasi ini, dibutuhkan beberapa entitas sebagai berikut :

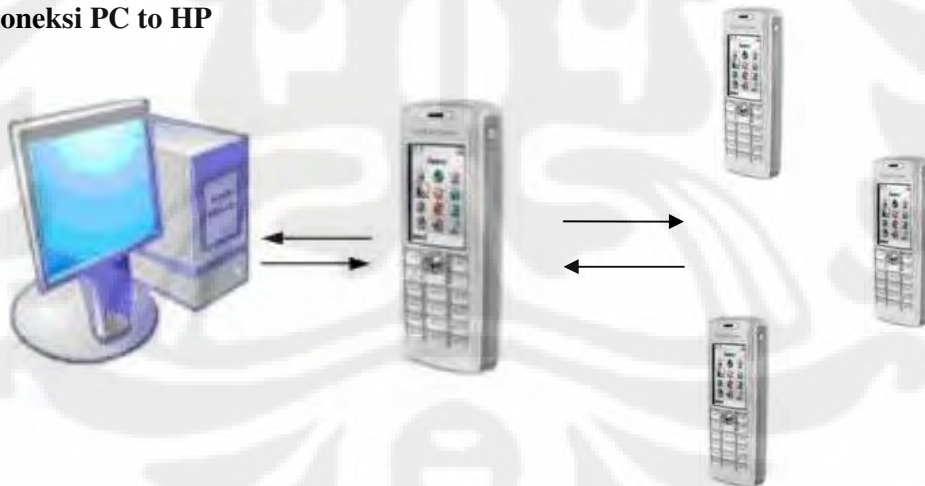
3.1.1 Hardware

- 3.1.1.1 PC
- 3.1.1.2 Handphone / Modem GSM
- 3.1.1.3 Kabel data

3.1.2 Software

- 3.1.2.1 Windows 2000,2003, XP
- 3.1.2.2 Micosoft Visual Basic 6.0
- 3.1.2.3 PHP
- 3.1.2.4 MySQL

3.2 Koneksi PC to HP



Gambar 3.1 Koneksi PC to HP to Multi User

PC dan handphone / modem GSM merupakan entitas utama dalam perancangan layanan “Send to Many ” ini. Dua entitas tersebut merupakan perangkat telekomunikasi yang dapat saling berkomunikasi akibat adanya suatu jaringan komputer. Dengan adanya

jaringan komputer, maka komputer tersebut mampu berhubungan dengan komputer lain atau dengan device lain termasuk jika dihubungkan ke handphone. Media yang digunakan untuk melakukan pertukaran informasi ini dapat dilakukan melalui kabel data atau melalui wireless. Dalam perancangan layanan send to many ini, penulis menggunakan penggunaan kabel data sebagai media transmisi antara handphone dan komputer, Sedangkan dari handphone ke handphone lain menggunakan jaringan wireless (GSM).

Dalam layanan “send to many”, terlebih dahulu dilakukan penyettingan antara komputer dengan handphone. Hal ini dapat dilakukan dengan melakukan settingan komponen HyperTerminal pada windows (start-allprogram-accessories-communications-HyperTerminal). Akibat peyettingan ini maka komputer dapat melakukan koneksi dengan handphone dan kemudian komputer dapat melakukan perintah-perintah terhadap handphone tersebut. Dalam hal ini handphone yang terhubung dengan komputer hanya berfungsi sebagai perantara dalam mengirim dan menerima sms ke handphone lain yang dilakukan secara broadcast atau send to many.

Dalam perancangan aplikasi yang penulis buat, pemrograman yang digunakan adalah visual basic versi 6.0. Komponen tambahan yang diperlukan untuk melakukan koneksi dengan handphone melalui komputer adalah MS Comm. MS Comm ini berfungsi seperti HyperTerminal pada windows. Setelah terbentuk koneksi antara komputer dengan handphone maka langkah selanjutnya adalah menuliskan perintah-perintah untuk mengendalikan handphone dari komputer. Perintah-perintah tersebut dinamakan AT Command. AT Command dapat berarti intruksi-intruksi atau perintah-perintah yang diberikan untuk mendapatkan perhatian dari handphone sehingga handphone dapat dikendalikan dari komputer.

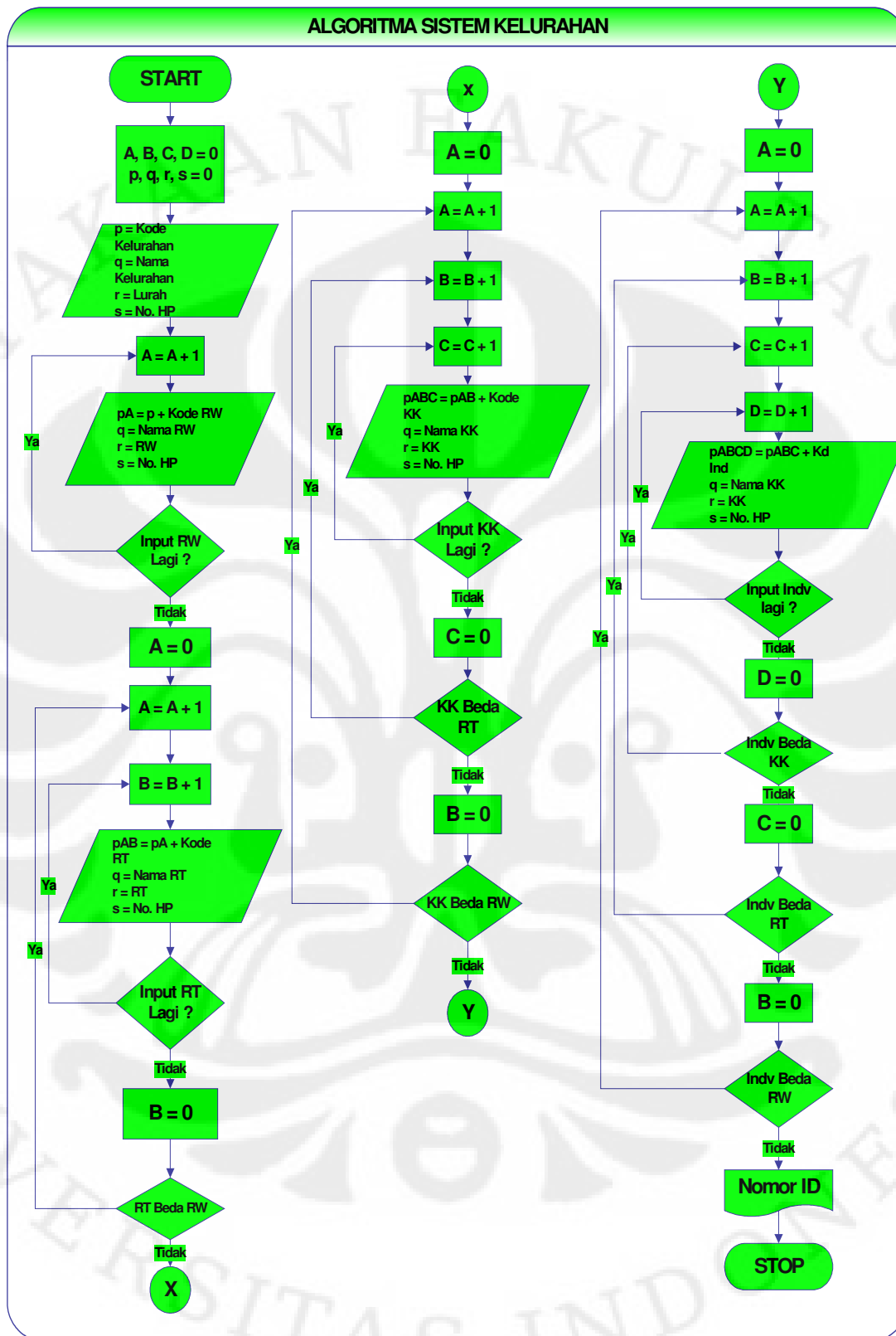
Dalam perancangan layanan send to many ini, 2 hal utama yang harus dilakukan adalah:

1. Perancangan Database
2. Perancangan SMS Sender

3.3 Perancangan Database

Untuk membuat perancangan layanan send to many ini dibutuhkan informasi mengenai individu yang bertindak sebagai pasien dan informasi mengenai tokoh masyarakat yang tinggal disekitar pasien khususnya lurah, rw,rt, dan kepala keluarga pasien itu sendiri.

Untuk memenuhi kebutuhan informasi tersebut maka haruslah dibuat database system kelurahan dan database mengenai informasi dbd. Database system kelurahan terdiri atas tabel negara, tabel propinsi, tabel kodya, tabel kecamatan, tabel kelurahan, tabel rw, tabel rt, tabel kk, dan tabel individu. Pembuatan database kelurahan tersebut telah dibuat oleh dr Amur Kayo. Dalam perancangan layanan send to many ini penulis hanya menggunakan tabel kelurahan, tabel rw, tabel rt, tabel kk, dan tabel individu. Informasi penting dari setiap tabel tersebut diantaranya kode, nama, dan nomor handphone. Dengan memanfaatkan konsep database relasional, setiap tabel saling memiliki relasi berdasarkan kode yang dimiliki, dimana kode dari setiap tabel merupakan penggabungan kode tabel tersebut dengan tabel sebelumnya. Tabel rw memiliki kode gabungan antara kode kelurahan dan kode rw, begitu juga tabel individu memiliki gabungan antara kode kelurahan, kode rw, kode rt, kode, kk, dan kode individu sehingga dihasilkan nomor ID untuk setiap individu. Dengan nomor ID inilah setiap tabel dapat saling berhubungan. Nomor ID inilah yang akan menjadi inputan ke dalam system database informasi dbd. Dalam database informasi dbd, informasi yang dibutuhkan adalah nomor ID, nama pasien, alamat pasien. Nama pasien didapatkan dari tabel individu, sedangkan alamat pasien didapatkan dari tabel kk, tabel rt, tabel rw, dan tabel lurah. Dari nomor ID tersebut juga dapat dibuat pesan kasus dbd beserta nomor handphone tujuan yang hendak dikirim. Dalam pembasan SMS Sender akan terlihat lebih jelas. Berikut adalah flowchart pengisian database system kelurahan yang menghasilkan nomor ID.



Gambar 3.2 Algoritma Sistem Kelurahan

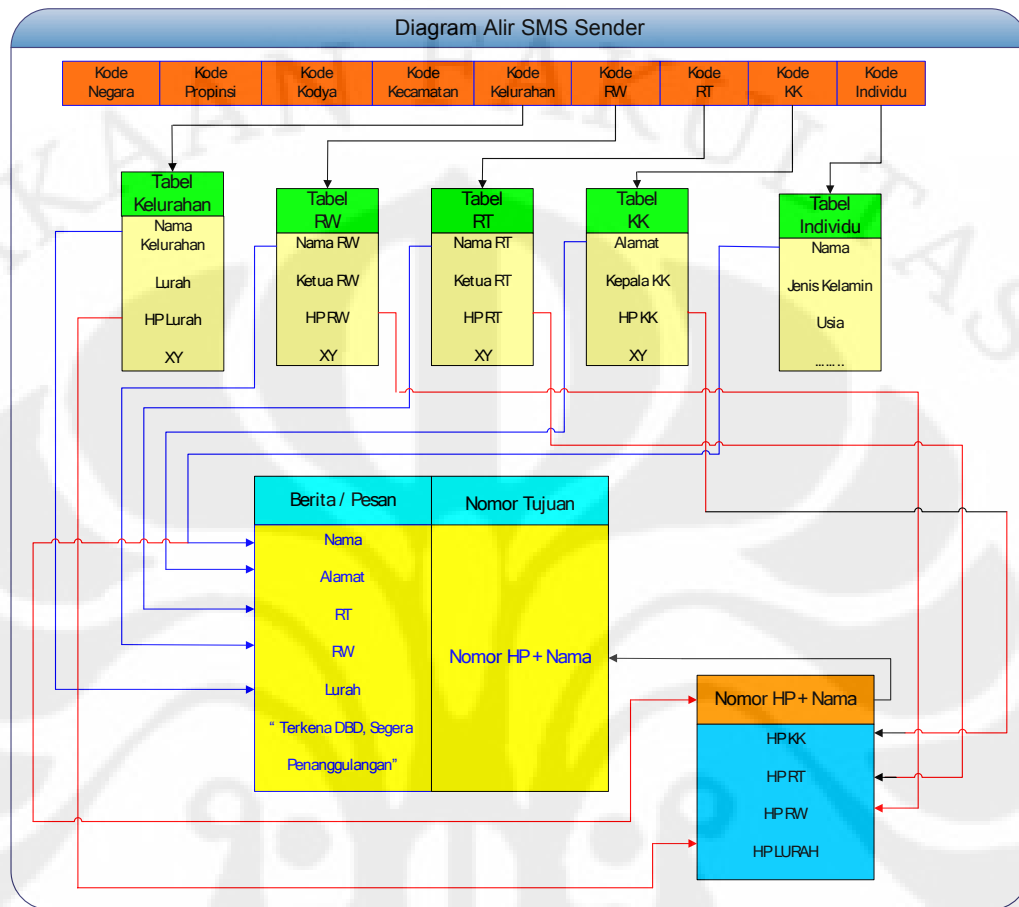
3.4 SMS Sender

3.4.1 Diagram Alir

Dalam membuat suatu perancangan aplikasi, terlebih dahulu penulis membuat diagram alir untuk memberikan gambaran alur proses yang nantinya akan dibuat. Berikut akan dijelaskan tentang diagram alir dari proses pengiriman sms dengan banyak tujuan.

Untuk melakukan pengiriman sebuah informasi melalui sms, maka terlebih dahulu harus diketahui pihak-pihak yang menjadi tujuan pengiriman informasi tersebut. Seperti telah disampaikan sebelumnya, bahwa perancangan yang penulis buat merupakan kelanjutan dari sistem informasi DBD berbasis sms yang dibuat oleh dr Amnur Kayo. Dari sistem informasi yang telah dibuat oleh beliau, setiap individu akan memiliki deretan nomor yang mengidentifikasi kode negara, kode provinsi, kode kota, kode kecamatan, kode kelurahan, kode rw, kode rt, kode kk, dan kode individu itu sendiri. Dan tentunya setiap individu pasti memiliki kode yang berbeda. Deretan nomor ini akan menjadi data input untuk melakukan registrasi ke dalam sistem informasi. Deretan nomor ini juga akan menjadi masukan untuk menentukan pihak-pihak yang akan menjadi tujuan pengiriman sms.

Berikut ini adalah diagram alir proses pengiriman sms ke multiuser.



Gambar 3.3 Diagram Alir SMS Sender

Dari diagram di atas dapat dilihat bahwa pihak yang akan menjadi tujuan penerima sms adalah kepala keluarga, rt, rw dan lurah dari individu yang terkena virus demam berdarah, dengan harapan pihak-pihak tersebut dapat segera melakukan tindakan penanggulangan / pencegahan penyebaran virus demam berdarah tersebut. Untuk mengetahui keterhubungan antara individu dengan pihak-pihak tersebut dapat dilihat dari nomor ID yang dimiliki oleh individu tersebut. Nomor ID dari setiap individu tersebut merupakan gabungan dari kode negara, propinsi hingga kode individu. Dari kode tersebut khususnya kode kelurahan, kode rw, kode rt, kode kk, dan kode individu dapat ditentukan pihak-pihak yang akan terlibat dalam proses ini. Informasi mengenai kelurahan terkait dapat dilakukan dengan melakukan searching pada tabel kelurahan berdasarkan kode kelurahan yang merupakan bagian dari kode individu yang telah dibuat dalam database. Begitu pula jika ingin mengetahui informasi mengenai rw, rt, kk, dan individu yang dimaksud. Dalam proses ini, hal utama yang diperlukan adalah

menentukan berita atau pesan yang akan dikirim dan nomor dari penerima yang akan menjadi tujuan pengiriman pesan tersebut.

a. Berita / Pesan

Berita / pesan meliputi informasi sebagai berikut :

1. Nama
2. Alamat
3. RT
4. RW
5. Lurah
6. Pesan

Informasi nama berasal dari tabel individu yang terkena virus demam berdarah, alamat terkait bersumber dari tabel kepala keluarga, dan nama RT, RW, dan Lurah masing-masing didapatkan dengan melakukan searching pada tabel RT, RW, dan Lurah. Sedangkan untuk pesan informasi telah dilakukan penyettingan yaitu :

“ Kasus DBD :..... Segera Penanggulangan”

Sebagai contoh dari pesan yang akan dikirim via sms adalah sebagai berikut :

“Kasus DBD : Suwardjono Adiputro Jln Raung Gang 1 RT 01 RW 01 Jati Asih, Segera Penanggulangan”

Dari informasi ini dapat diterjemahkan bahwa RT 01, RW 01 dan Kelurahan Jati Asih diharapkan dapat melakukan penanggulangan / pencegahan penyebaran virus demam berdarah setempat khususnya di Jln Raung Gang 1

b. Nomor Tujuan

Proses pengiriman ini dilakukan untuk beberapa tujuan (send to many), sehingga nomor tujuan pengiriman pesan meliputi beberapa nomor handphone. Untuk melakukannya, penulis mengelompokkan nomor-nomor tersebut ke dalam sebuah tabel sementara. Tabel ini berisi nama kelurahan, nama rw, nama rt, dan nama kk beserta nomor handphone masing-masing.

Dari diagram alir di atas, nomor handphone lurah, rt, rw, dan kepala keluarga yang di dapat masing-masing dari tabel lurah, rt, rw, dan kepala keluarga dimasukkan ke dalam tabel sementara dan akan menjadi nomor tujuan pengiriman sms ini.

Sebagai contoh adalah:

HP Lurah Jati Asih : 0811119116
 HP RW 01 : 085697178287
 HP RT 01 : 081399450484
 HP KK : 02196141770

Nomor-nomor tersebut akan masuk ke dalam tabel sementara. Tabel sementara ini hanya diperlukan untuk menampilkan nama lurah, rw, rt, dan kk beserta nomor handphonenya.

Tabel 3.1 Tabel Sementara Nomor Tujuan

Lurah Jati Asih	0811119116
RW 01	085697178287
RT 01	081399450484
KK	02196141770

Begitu juga dengan pesan yang akan dikirimkan ditampilkan ke dalam tabel yang bersifat sementara (hanya untuk menampilkan)

Tabel 3.2 Tabel Sementara Pesan

Pesan	Kasus DBD : Suwardjono Adiputro Jln Raung Gang 1 RT 01 RW 01 Jati Asih, Segera Penanggulangan
-------	---

BAB 4

ANALISIS PROGRAM dan UJICOBA

Tahap awal dalam pembuatan “layanan send to many” ini adalah membuat database yang berisi informasi mengenai pasien dan membuat interface antara user (pengguna program) dengan Microsoft VB. Kemudian dilakukan pemrograman untuk masing-masing interface. Dalam pembuatan “layanan send to many” ini sedikitnya diperlukan tiga buah program utama, diantaranya program pembuatan database, program untuk mengirim sms, dan program untuk menerima sms. Masing-masing program memiliki sub program-sub program yang bekerja berdasarkan prinsip pemrograman modular. Ujicoba keberhasilan program yang dibuat, penulis melakukan proses entry data dan mengirimkan pesan kasus dbd ke beberapa kerabat dekat.

Analisis perancangan dan ujicoba penulis buat menjadi satu dengan tujuan untuk lebih meringkas penulisan karena pada hakikatnya tidak jauh berbeda. Analisis Program dan Ujicoba ini dibagi atas menjadi dua sub bab yaitu analisis database dan analisis sms sent. Berikut adalah penjelasannya.

4.1 Analisis Database

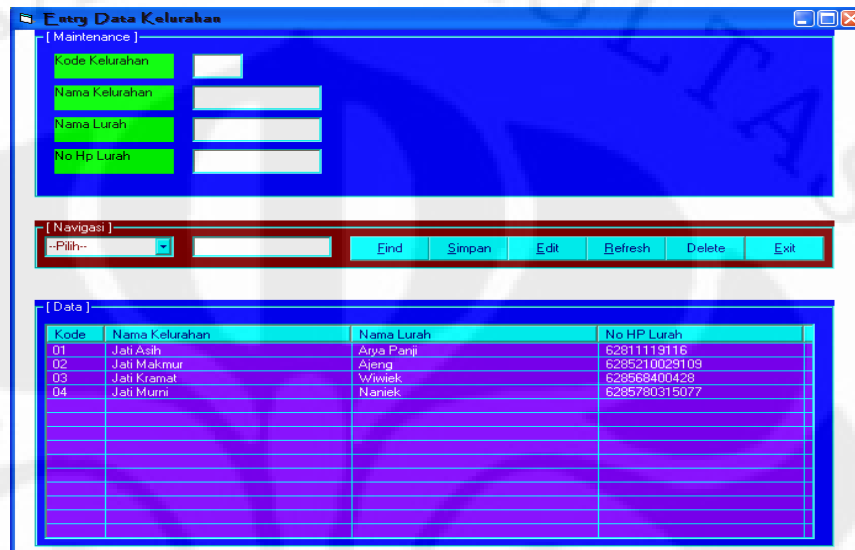
Seperti telah dijelaskan pada Bab 3 mengenai perancangan database, dalam perancangan layanan send to many ini dibutuhkan sedikitnya 2 database diantaranya database system kelurahan dan database system informasi DBD.

4.1.1 Database Sistem Kelurahan



Gambar 4.1 Database Sistem Kelurahan

4.1.1.1 Database Kelurahan



Gambar 4.2 Database Kelurahan

Berikut adalah listing program yang dibuat oleh dr Amnur Kayo dengan perubahan tampilan dan pengeditan.

```
Option Explicit
Dim conn As ADODB.Connection

Sub connect()
' connect to MySQL server using MySQL ODBC 5.0 Driver
Set conn = New ADODB.Connection
conn.Open "Dsn=ADI"
conn.CursorLocation = adUseClient
End Sub
```

Hal pertama yang dilakukan adalah melakukan inisialisasi. Option Explicit digunakan untuk melakukan pencegahan jika terdapat variable yang tidak dideklarasikan. Untuk mendeklarasikan suatu variabel maka digunakan “Dim”. Variabel conn dideklarasikan sebagai ADODB.Connection, ADODB merupakan control dalam VB untuk dapat berhubungan dengan database. (.).connection berarti variable conn digunakan untuk melakukan koneksi dengan database.. Variable-variabel ini hanya berfungsi jika dilakukan penyettingan selanjutnya. Seperti halnya pada baris program selanjutnya yaitu “Set conn = New

ADODB.Connection” . Jika tidak terdapat masalah maka setiap kali menuliskan conn dan dilanjutkan dengan dot (.), maka secara otomatis akan muncul control selanjutnya yang bersesuaian dengan variable tersebut. Conn.open ”Dsn=ADI” digunakan untuk memberikan perintah agar VB dapat berkomunikasi dengan servel local yang memiliki *data source name* (DSN) ADI.. Karena membutuhkan data dari server maka property conn.CursorLocation bertindak sebagai adUsedClient.

```

Private Sub cmdSimpan_Click()
Dim rsCek As New ADODB.Recordset

If txtKode.Text = "" Then
  MsgBox "Masukkan Kode Kelurahan", vbInformation + vbOKOnly, "information"
  txtKode.SetFocus
  Exit Sub
Else
  Call connect
  Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & """)
  If Not rsCek.EOF Then
    MsgBox "Maaf Kode Kelurahan Sudah Ada Dalam Database", vbInformation +
vbOKOnly, "information"
    txtKode.SetFocus
    Exit Sub
  End If
End If

If txtNmKelurahan = "" Then
  MsgBox "Masukkan Nama Kelurahan", vbInformation + vbOKOnly, "information"
  txtNmKelurahan.SetFocus
  Exit Sub
End If

```

Program di atas terjadi ketika tombol buton “Simpan” diklik. rsCek menggunakan (.)recordset artinya variable rsCek digunakan untuk melakukan penyimpanan sementara terhadap record-record tabel dalam database. Jika nomor kode belum dimasukkan maka VB akan mengeluarkan perintah “Masukkan Kode Kelurahan” . (.) SetFokus berarti VB akan memfokuskan kursor ke property text (txtKode) dengan harapan user memprioritaskan posisi tersebut agar perintah “Masukkan Kode Kelurahan” tidak muncul kembali. Kemudian VB melakukan pemanggilan connect (Call connect). Dalam hal ini

VB menggunakan konsep pemrograman modular, artinya hanya memanggil sub function dari suatu algoritma tanpa harus menuliskan kembali algoritma program tersebut. Dengan call connect, berarti sudah dapat melakukan proses yang dilakukan oleh sub connect. Untuk mencegah terjadinya duplikasi data yang akan disimpan, maka harus terlebih dahulu dilakukan pengecekan apakah di dalam database sudah terdapat data yang dimasukkan oleh user atau tidak. Untuk itu dilakukan proses pencarian data terlebih dahulu dengan perintah :

```
Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & """)
```

Artinya bahwa variable conn melakukan eksekusi terhadap database MySQL, berupa melakukan pencarian sekaligus menampilkan * (seluruh field) dari tabel kelurahan dimana field “fl_kd_kelurahan” sama dengan nilai yang dituliskan user pada property text. Proses pencarian ini tentunya pada database server yang telah terkoneksi. If Not EOF artinya sebelum mencapai EOF (end of file) telah ditemukan data yang dimaksud, maka VB mengeluarkan statement “Maaf Kode Kelurahan Sudah Ada Dalam Database”. Jika ternyata tidak ditemukan maka VB mempersilahkan user untuk melakukan pengisian data hingga selesai. Kemudian mengeksekusi perintah :

```
conn.Execute "INSERT INTO
tb_kelurahan(fl_kd_kelurahan,fl_nm_kelurahan,fl_nm_lurah,fl_hp_lurah ) " _&
"VALUES(" & (txtKode.Text) & "," & (txtNmKelurahan.Text) & "," &
(txtNmLurah.Text) & "," & (txtHpLurah.Text)
```

Artinya memasukkan ke dalam tabel kelurahan khususnya field atau kolom fl_kd_kelurahan,fl_nm_kelurahan,fl_nm_lurah, dan fl_hp_lurah dengan nilai-nilai yang telah dimasukkan oleh user. Hasil dari perubahan dapat terlihat pada database. Untuk dapat menampilkan pada aplikasi (VB) maka harus dibuat program tersendiri seperti berikut:


```

Sub tampil(key As String)
On Error GoTo pesan
Dim rsData As New ADODB.Recordset
Dim fieldKey As String
Dim istview
Call connect
If key = "LOAD" Then
    Set rsData = conn.Execute("SELECT*FROM tb_kelurahan ORDER BY fl_kd_kelurahan
ASC")
Else
Select Case cmbSearch.ListIndex
    Case 0
        fieldKey = "fl_kd_kelurahan"
    Case 1
        fieldKey = "fl_nm_kelurahan"
    Case 2
        fieldKey = "fl_nm_lurah"
    Case 3
        fieldKey = "fl_hp_lurah"
End Select

Set rsData = conn.Execute("SELECT*FROM tb_kelurahan WHERE " & fieldKey & " Like
%" & (txtSearch.Text) & "%' ORDER BY fl_kd_kelurahan Asc ")
If rsData.EOF Then
    MsgBox "Maaf Data yang Anda Cari Tidak DiTemukan", vbInformation + vbOKOnly,
"information"
End If
End If
Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
    Set istview = lvData.ListItems.Add(, , rsData(0))
    istview.ListSubItems.Add , , rsData(2)
    istview.ListSubItems.Add , , rsData(3)
    istview.ListSubItems.Add , , rsData(4)

rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

```

Sub function tampil di atas berfungsi sebagai sub function yang siap menjalani program jika terdapat pemanggilan atas dirinya sesuai key yang digunakan, jika dipilih Key LOAD, maka akan dieksekusi perintah:

```

Set rsData = conn.Execute("SELECT*FROM tb_kelurahan ORDER BY fl_kd_kelurahan
ASC")

```

Artinya dilakukan pencarian sekaligus menampilkan field-field pada tabel kelurahan yang diurutkan secara ascending (ASC) pada field fl_kd_kelurahan.

Jika tidak maka dilakukan program:

```
Select Case cmbSearch.ListIndex
  Case 0
    fieldKey = "fl_kd_kelurahan"
  Case 1
    fieldKey = "fl_nm_kelurahan"
  Case 2
    fieldKey = "fl_nm_lurah"
  Case 3
    fieldKey = "fl_hp_lurah"

End Select
Set rsData = conn.Execute("SELECT*FROM tb_kelurahan WHERE " & fieldKey & " Like
'%" & (txtSearch.Text) & "%' ORDER BY fl_kd_kelurahan Asc ")
If rsData.EOF Then
  MsgBox "Maaf Data yang Anda Cari Tidak DiTemukan", vbInformation + vbOKOnly,
"information"
```

Artinya dilakukan pencarian sekaligus menampilkan tabel kelurahan berdasarkan fieldKey yang dipilih oleh user. Jika ternyata data tidak ditemukan (If rsData.EOF) maka VB mengeluarkan statement “Maaf Data yang Anda Cari Tidak Ditemukan”

```
Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
  Set istview = lvData.ListItems.Add(, , rsData(0))
  istview.ListSubItems.Add , , rsData(2)
  istview.ListSubItems.Add , , rsData(3)
  istview.ListSubItems.Add , , rsData(4)
rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub
```

Kemudian dilakukan pemanggilan Call grid dengan listing program sebagai berikut:

```
Sub grid()
lvData.ColumnHeaders.Item(1).Text = "Kode"
lvData.ColumnHeaders.Item(1).Width = "700"
lvData.ColumnHeaders.Item(2).Text = "Nama Kelurahan"
lvData.ColumnHeaders.Item(2).Width = "3000"
lvData.ColumnHeaders.Item(3).Text = "Nama Lurah"
lvData.ColumnHeaders.Item(3).Width = "3000"
lvData.ColumnHeaders.Item(4).Text = "No HP Lurah"
lvData.ColumnHeaders.Item(4).Width = "2500"
End Sub
```

Fungsi grid ini untuk melakukan setting ukuran terhadap lvData (List View Data) sehingga ketika dieksekusi, lvData telah memiliki ukuran yang diharapkan. lvData merupakan property yang digunakan untuk menampilkan field-field database ke dalam aplikasi VB.

```
lvData.ListItems.Clear
Do While Not rsData.EOF
  Set istview = lvData.ListItems.Add(, , rsData(0))
  istview.ListSubItems.Add , , rsData(2)
  istview.ListSubItems.Add , , rsData(3)
  istview.ListSubItems.Add , , rsData(4)
rsData.MoveNext
Loop
Exit Sub
```

Perintah di atas melakukan pemindahan atas data yang ada di dalam database sehingga dapat ditampilkan ke dalam aplikasi VB. rsData(0) berarti field yang memiliki kolom ke 0 (array kolom database selalu diawali dengan 0) dalam database akan ditempatkan pada kolom lvData yang ke 1 (Kode). Begitu juga dengan rsData(4) (field ke 4 dalam database) ditempatkan pada kolom ke 4 (No Hp Lurah).

Program-program di atas akan dilakukan ketika user melakukan tombol “simpan” dalam Form Entry tabel kelurahan.

Ketika tombol “find” ditekan, maka listing programnya adalah sebagai berikut:

```
Private Sub cmdFind_Click()
Call Clear
If txtSearch.Text = "" Then
  MsgBox "Pilih Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
  "information"
  txtSearch.SetFocus
Else
Call tampil("CARI")
End If
```

Call Clear melakukan pembersihan terhadap data-data pada property text yang dimasukkan oleh user. Berikut listing programnya:

```
Sub Clear()
txtKode = ""
txtNmKelurahan = ""
txtNmLurah = ""
txtHpLurah = ""
txtKode.SetFocus
End Sub
```

Untuk melakukan pencarian (find) pada suatu data tentunya user harus memasukkan inisial data yang dicari sesuai dengan field yang telah ditentukan. Kemudian program akan membacanya sebagai berikut:

```
Select Case cmbSearch.ListIndex
  Case 0
    fieldKey = "fl_kd_kelurahan"
  Case 1
    fieldKey = "fl_nm_kelurahan"
  Case 2
    fieldKey = "fl_nm_lurah"
  Case 3
    fieldKey = "fl_hp_lurah"

End Select
Set rsData = conn.Execute("SELECT*FROM tb_kelurahan WHERE " & fieldKey & " Like
'%" & (txtSearch.Text) & "%' ORDER BY fl_kd_kelurahan Asc ")
If rsData.EOF Then
  MsgBox "Maaf Data yang Anda Cari Tidak DiTemukan", vbInformation + vbOKOnly,
"information"
End If
```

Program ini terletak pada sub function tampil sebagaimana telah dijelaskan sebelumnya. Kemudian dilakukan pemanggilan call tampil("CARI"). Program ini pun terletak pada sub function tampil. Namun pada sub function tampil tidak terdapat key "CARI", yang ada hanya "LOAD". Key "CARI" ini dapat di eksekusi bukan berdasarkan atas Key "CARI", tetapi berdasarkan Key selain "LOAD". Berikut listing programnya:

```
Call connect
If key = "LOAD" Then
    Set rsData = conn.Execute("SELECT*FROM tb_kelurahan ORDER BY fl_kd_kelurahan
ASC")
Else
    Select Case cmbSearch.ListIndex
    Case 0
        fieldKey = "fl_kd_kelurahan"
    Case 1
        fieldKey = "fl_nm_kelurahan"
    Case 2
        fieldKey = "fl_nm_lurah"
    Case 3
        fieldKey = "fl_hp_lurah"
    End Select
    Set rsData = conn.Execute("SELECT*FROM tb_kelurahan WHERE " & fieldKey & " Like
%" & (txtSearch.Text) & "%' ORDER BY fl_kd_kelurahan Asc ")
    If rsData.EOF Then
        MsgBox "Maaf Data yang Anda Cari Tidak DiTemukan", vbInformation + vbOKOnly,
"information"
    End If
End If
```

Ketika user menekan tombol Edit maka akan program akan melakukan proses pengeditan terhadap data yang telah tersimpan di dalam database. Berikut adalah listing programnya:

```

Private Sub cmdEdit_Click()
    On Error GoTo pesan
    Dim rsCek As New ADODB.Recordset
    If txtKode.Text = "" Then
        MsgBox "Maaf Kode Kelurahan Harus Anda Masukkan", vbInformation +
vbOKOnly, "information"
    Else
        Call connect
        Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE
fl_kd_kelurahan=" & (txtKode.Text) & """)
        If rsCek.EOF Then
            MsgBox "Maaf Kode Kelurahan yang Anda Masukkan Tidak Ada di Dalam
Database", vbInformation + vbOKOnly, "information"
        Else
            If txtNmKelurahan = "" Then
                MsgBox "Masukkan Nama Kelurahan", vbInformation + vbOKOnly,
"information"
                txtNmKelurahan.SetFocus
                Exit Sub
            End If

            If txtNmLurah = "" Then
                MsgBox "Masukkan Nama Lurah", vbInformation + vbOKOnly, "information"
                txtNmLurah.SetFocus
                Exit Sub
            End If
            If txtHpLurah = "" Then
                MsgBox "Masukkan Nomor HP Lurah", vbInformation + vbOKOnly,
"information"
                txtHpLurah.SetFocus
                Exit Sub
            Else
                MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"
                conn.Execute ("UPDATE tb_kelurahan SET fl_nm_kelurahan=" &
(txtNmKelurahan.Text) & ", fl_nm_Lurah =" & (txtNmLurah.Text) & ", fl_hp_Lurah
=" & (txtHpLurah.Text) & " WHERE fl_kd_kelurahan =" & (txtKode.Text) & """)
                Call tampil("LOAD")
                MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
            End If
        End If
    End If
End If

```

Seperti pada program sebelumnya, untuk program edit memiliki perintah-perintah yang hampir sama.

Ketika user menekan tombol Refresh, maka akan dijalankan program sebagai berikut:

```
Private Sub cmdRefresh_Click()
    Call Clear
    Call tampil("LOAD")
End Sub
```

Tombol Refresh akan membersihkan property text yang dimasukkan user dan menampilkan seluruh data yang ada dalam database ke lvData.

Ketika user menekan tombol Delete, maka akan dijalankan program sebagai berikut:

```
Private Sub cmdDelete_Click()
    On Error GoTo pesan
    Dim rsCek As New ADODB.Recordset
    If txtKode.Text = "" Then
        MsgBox "Maaf kode kelurahan harus dimasukkan", vbInformation + vbOKOnly,
        "information"
    Else
        Call connect
        Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
        (txtKode.Text) & """)
        If rsCek.EOF Then
            MsgBox "Maaf kode kelurahan tidak ada di dalam database", vbInformation +
            vbOKOnly, "information"
        Else
            conn.Execute ("DELETE FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
            (txtKode.Text) & """)
            Call tampil("LOAD")
            MsgBox "Data Kelurahan Sukses di DELETE", vbInformation + vbOKOnly,
            "Information"
        End If
    End If
    Exit Sub

pesan:
    MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub
```

Inti dari program ini adalah

```
conn.Execute ("DELETE FROM tb_kelurahan WHERE fl_kd_kelurahan=" & (txtKode.Text)
& """)
```

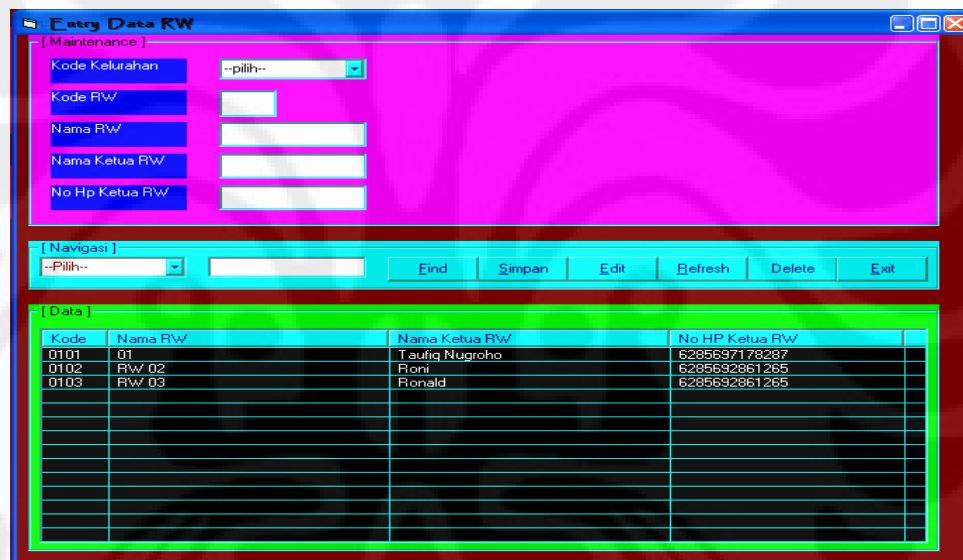
Artinya, fungsi melakukan Delete dari tabel kelurahan di mana field `fl_kd_kelurahan` sama dengan nilai yang diinputkan oleh user.

Ketika menekan tombol exit, maka listing programnya sebagai berikut:

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

Dalam VB “Unload Me” merupakan fungsi untuk keluar dari proses running.

4.1.1.2 Database RW



Gambar 4.3 Database RW

Listing program pada entry data RW tidak jauh berbeda dengan listing program pada entry data kelurahan. Dari gambar di atas dapat dilihat bahwa proses entry data RW sama dengan entry data kelurahan, begitu pula dengan command – command atau tombol-tombol yang digunakan. Namun terdapat perbedaan ketika melakukan entry data, berupa penambahan kode kelurahan. Dengan melakukan penambahan kode kelurahan ini memungkinkan terjadinya penggabungan kode kelurahan dengan kode rw, sehingga terbentuk kode yang mengidentifikasi kode kelurahan dan kode rw. Hal ini dapat dilihat pada `lvData` (Listview Data) terutama pada kolom Kode. Penggabungan nomor ini bertujuan untuk tercipta suatu relasi antara tabel kelurahan dengan tabel rw,

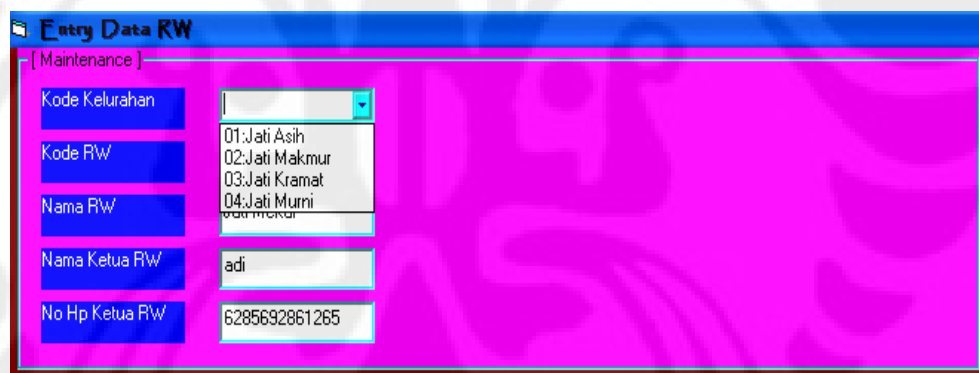
sehingga memungkinkan penulis untuk mendapatkan informasi baik informasi dari tabel kelurahan maupun tabel rw dari nomor kode tersebut.

Berikut adalah potongan listing program untuk melakukan penggabungan dua kode tersebut:

```
Private Sub cmdSimpan_Click()
Dim rsCek As New ADODB.Recordset
Dim kdKunci As String
kdKunci = Left(cmbKelurahan.Text, 2) & txtKode.Text
```

Pada program di atas terdapat penambahan variable kdKunci, dimana variable tersebut merupakan penggabungan kode kelurahan dengan kode rw yang diinputkan oleh user. Penambahan variable tersebut berlaku juga untuk sub function Find, Edit, Delete, dan Refresh.

Left(cmbKelurahan.Text,2) berarti mengambil dua digit dari kiri dari text yang terdapat pada kolom combo, seperti gambar berikut,



Gambar 4.4 Entry Data RW

Penulis menganalogikan kode kelurahan, kode rw, hingga kode individu dengan dua digit nilai hanya untuk memudahkan penulis. Dari gambar di atas terlihat pada kolom combo :



Terdapat masing-masing kode kelurahan beserta nama kelurahannya. Hal ini dapat dilakukan dengan program sebagai berikut:

```
Private Sub cmbKelurahan_GotFocus()
Call Combo
End Sub
```

“Event GotFocus dipicu ketika sebuah control menerima fokus”[3]. Event yang dimaksud adalah Call Combo. Berikut adalah listing program untuk Combo:

```
Sub Combo()
On Error GoTo pesan
Dim rsCb As New ADODB.Recordset
Call connect
Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_kelurahan,',',fl_nm_kelurahan) as
kelurahan FROM tb_kelurahan ORDER BY fl_kd_kelurahan ASC ")
cmbKelurahan.Clear
Do While Not rsCb.EOF
    cmbKelurahan.AddItem rsCb(0)
rsCb.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub
```

Inti dari listing proram ini adalah

```
Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_kelurahan,',',fl_nm_kelurahan) as
kelurahan FROM tb_kelurahan ORDER BY fl_kd_kelurahan ASC ")
cmbKelurahan.Clear
Do While Not rsCb.EOF
    cmbKelurahan.AddItem rsCb(0)
rsCb.MoveNext
```

Seperti penjelasan sebelumnya, rsCb dideklarasikan sebagai variable penyimpan recordset dari tabel kelurahan.

```
Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_kelurahan,',',fl_nm_kelurahan) as
kelurahan FROM tb_kelurahan ORDER BY fl_kd_kelurahan ASC ")
```

Perintah di atas melakukan pemilihan sekaligus menampilkan field pada tabel kelurahan . Field yang ditampilkan hanya kode beserta nama kelurahannya. Hal ini dapat dilakukan dengan perintah :

CONCAT(fl_kd_kelurahan,':'fl_nm_kelurahan).

Untuk melakukan penyimpanan ke dalam tabel rw dilakukan perintah sebagai berikut:

```
conn.Execute "INSERT INTO tb_rw( fl_kd_rw,fl_nm_rw,fl_nm_ketua_rw,fl_hp_ketua_rw ) "
-
      & "VALUES('" & (kdKunci) & "','" & (txtNmRW.Text) & "','" &
      (txtNmKetuaRW.Text) & "','" & (txtHpRW.Text) & "''")"
```

Penjelasannya sama seperti proses penyimpanan pada entry data tabel kelurahan, namun bedanya field fl_kd_rw berisi nomor kode kelurahan dan nomor kode rw yang dideklarasikan dengan variable kdKunci .

4.1.1.3 Database RT

Kode	Nama RT	Nama Ketua RT	No HP Ketua RT
010101	RT 01	Welman Hasiholan	6281399450484
010102	RT 01	adi	6285692861265
010103	RT 01	adi	6285692861265
020201	RT 03	Lukman	080970970970

Gambar 4.5 Database RT

Penjelasan untuk program entry data rt serupa dengan program entry data rw

4.1.1.4 Database KK

Kode	Nama KK	Nama Kepala KK	No HP Kepala KK
01010101	Jln Raung Gang 1 No 2	Haris	622196141770
01010202	Jln Raung Gang 1 No 2	adi	6285692861265
01010301	Jln Perjuangan no. 100	Harv	622196141770

Gambar 4.6 Database KK

Penjelasan program entry tabel kk serupa dengan entry tabel rw

4.1.1.5 Database Individu

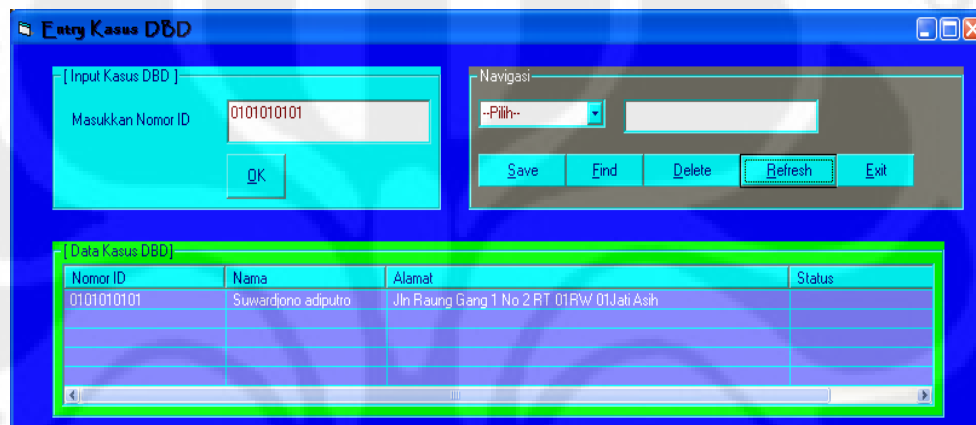
Kode	Nomor KTP	Nama Individu	No HP Individu	Jenis Kelamin
0101010101	10.5506.060687.1002	Suwardjono adiputro	6285692861265	Laki-Laki

Gambar 4.7 Database Individu

Penjelasan program entry tabel individu serupa dengan penjelasan program entry tabel rw.

4.1.2 Database Sytem Informasi DBD

Setelah melakukan proses entry data individu, maka langkah selanjutnya adalah membuat database system informasi dbd sebagai database untuk menyimpan identitas pasien yang terkena kasus demam berdarah. Nomor ID merupakan input dari proses ini. Nomor ID ini adalah kode individu yang merupakan gabungan dari kode kelurahan, kode rw, kode rt, kode kk, hingga kode individu.



Gambar 4.8 Database System Informasi DBD

Dari tabel di atas, dapat dilihat bahwa dengan melakukan inputan nomor ID maka akan tampil identitas pasien yang memiliki nomor tersebut. Identitas yang dimaksud berupa nomor ID, nama, dan alamat pasien. Nomor ID dan nama pasien didapat dari field kode individu dan field nama pada tabel individu, sedangkan alamat pasien didapat dari field nama kk pada tabel kk, nama rt pada tabel rt, nama rw pada tabel rw, dan nama lurah pada tabel lurah. Dengan menginput nomor ID, maka identitas pasien tersebut akan ditampilkan ke dalam lvData (listview data) sebagaimana tampil pada gambar di atas. Penyimpanan data pasien dbd dilakukan dengan membuat tabel baru bernama tabel entry_dbd. Untuk melakukan penyimpanan data tersebut, proses yang dilakukan tidak jauh berbeda dengan penyimpanan proses sebelumnya, berikut adalah listing program yang dilakukan,

```

Private Sub cmdSave_Click()
Dim rsHP As New ADODB.Recordset
Dim ID, Nama, Alamat

Call connect
Set rsHP = conn.Execute("SELECT
i.fl_kd_individu,i.fl_nama,CONCAT(k.fl_nm_kk,"t.fl_nm_rt","w.fl_nm_rw","l.fl_nm_keluraha
n) FROM tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l WHERE i.fl_kd_individu Like
%" & (txtID.Text) & "%' AND k.fl_kd_kk Like '%" & Left(txtID, 8) & "%' AND t.fl_kd_rt
Like '%" & Left(txtID, 6) & "%'AND w.fl_kd_rw Like '%" & Left(txtID, 4) & "%'AND
l.fl_kd_kelurahan Like '%" & Left(txtID, 2) & "%'")
Call grid
lvData.ListItems.Clear
If Not rsHP.EOF Then
    ID = "" & rsHP(0)
    Nama = "" & rsHP(1)
    Alamat = "" & rsHP(2)
    conn.Execute ("INSERT INTO tb_entry_dbd VALUES('" & ID & "','" & Nama & "','" &
Alamat & "')")
    rsHP.MoveNext
    Call tampil("Refresh")
    MsgBox "Data Sukses DiSimpan", vbOKOnly + vbInformation, "Information"
    Exit Sub
End If
End Sub

```

Bedanya dengan proses sebelumnya, untuk memudahkan melakukan penyimpanan pada tabel yang baru (tb_entry_dbd) maka identitas pasien tersebut harus ditampilkan terlebih dahulu ke lvData pada VB dengan program sebagai berikut :

```

Set rsHP = conn.Execute("SELECT
i.fl_kd_individu,i.fl_nama,CONCAT(k.fl_nm_kk,"t.fl_nm_rt","w.fl_nm_rw","l.fl_nm_kelura
han) FROM tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l WHERE i.fl_kd_individu
Like '%" & (txtID.Text) & "%' AND k.fl_kd_kk Like '%" & Left(txtID, 8) & "%' AND
t.fl_kd_rt Like '%" & Left(txtID, 6) & "%'AND w.fl_kd_rw Like '%" & Left(txtID, 4) &
"%'AND l.fl_kd_kelurahan Like '%" & Left(txtID, 2) & "%'")

```

Karena terjadi penggabungan data dari tiap tabel yang memiliki kolom-kolom yang berbeda maka proses pemilihan akan menjadi lebih kompleks dari sebelumnya, maka penamaan lain yang mewakili tabel tersebut. Hal ini dinamakan alias. Seperti pada potongan program di atas :

```

SELECT
i.fl_kd_individu,i.fl_nama,CONCAT(k.fl_nm_kk,"t.fl_nm_rt","w.fl_nm_rw","l.fl_nm_kelura
han) FROM tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l

```

Program di atas menampilkan fl_kd_individu dari tabel individu (disingkat dengan i.fl_kd_individu, dimana I alias dari tabel individu), fl_nama dari tabel individu, kemudian menampilkan data :

k.fl_nm_kk,", t.fl_nm_rt,",w.fl_nm_rw,",l.fl_nm_kelurahan secara berturut-turut menjadi satu kalimat. k.fl_nm_kk merupakan kolom fl_nm_kk dari tabel kk dengan alias k. Pendeklarasian symbol alias dengan masing-masing tabel dilakukan dengan penulisan sebagai berikut "tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l"

Setelah menampilkan data yang dimaksud, maka langkah selanjutnya adalah penyimpanan ke dalam tabel yang baru dengan nama tb_entry_dbd yang memiliki field Nomor ID, field Nama, dan field alamat, dengan perintah sebagai berikut:

```
conn.Execute ("INSERT INTO tb_entry_dbd VALUES(" & ID & "," &
Nama & "," & Alamat & " ")")
```

4.2 Analisis SMS Sent

Untuk membuat program mengirim sms ini, penulis membaginya menjadi subprogram-subprogram, di antaranya:

4.2.1 Mengatur Port Serial

Pengaturan port serial ini berdasarkan port koneksi yang tersedia untuk melakukan koneksi dari HP ke PC. Untuk mengetahuinya, lebih mudah dengan melihat pada Hyperterminal yang telah disediakan oleh windows. Penulis menggunakan COM 6.

4.2.2 Melakukan settingan MsComm

MsComm merupakan komponen yang digunakan dalam Microsoft visual basic sehingga dapat dilakukan proses komunikasi antara pc dengan devais lain (hanphone) melalui serial port.



Gambar 4.9 Komponen MsComm

Icon seperti ini merupakan icon komponen MsComm, kependekan dari Microsoft Communication Control yang digunakan untuk melakukan settingan “com-port (communication port)”, sehingga PC mampu melakukan koneksi dengan handphone. Penggunaan MsComm dalam VB ini seperti penggunaan Hyperterminal pada windows. Supaya komponen ini bekerja, maka harus dilakukan proses inisialisasi awal, seperti:

```

With MSComm1
  .CommPort = 6
  .Settings = "9600,N,8,1"
  .Handshaking = comRTS
  .RTSEnable = True
  .DTREnable = True
  .RThreshold = 1
  .SThreshold = 1
  .InputMode = comInputModeBinary
  .InputLen = 0
  .PortOpen = True 'must be the last
End With

```

Property comport digunakan untuk menentukan port yang digunakan untuk melakukan komunikasi. Property settings secara default bernilai 9600,N,8,1 9600 merupakan baud rate, N berarti non parity, 8 merupakan jumlah data bits, dan 1 merupakan jumlah stop bits. Handshaking merupakan property yang memungkinkan dua devais melakukan koneksi, comRTS berarti komunikasi untuk melakukan request to send .RTSEnable = True artinya mengenabledkan protocol RTS. DTREnable=True artinya mengenabledkan data terminal ready selama komunikasi. Property RThreshold menentukan jumlah karakter yang diterima oleh receive buffer sebelum event dipicu. InputMode menentukan tipe data input yang akan dibaca oleh CPU. InputLen =0 berarti menentukan jumlah karakter yang dibaca CPU .Jika bernilai 0 maka seluruh isi receive buffer akan dibaca CPU. property input untuk melakukan pembacaan terhadap receive buffer.

4.2.3 Konversi Text mode menjadi PDU mode

Sebagaimana telah diterangkan sebelumnya, dalam perancangan program mengirim sms ini, penulis menggunakan hanphone tipe sony ericsson w200i. Tipe handphone tersebut hanya support untuk pengiriman sms dengan tipe PDU

mode. Untuk itu penulis harus mengubah terlebih dahulu text mode yang diisikan oleh user menjadi PDU mode dan mengatur formatnya sehingga dapat diterapkan pada handphone tersebut. Berikut adalah contoh format pdu yang telah dijelaskan pada bab 2

Format PDU :

```
07912658050000F011000C912658165522000000AB0DD3E614442CBBCF613
7C82A04
```

Untuk memudahkan analisa, penulis memecahnya menjadi beberapa bagian:

```
07-91-2658050000F0-1100-0C-91-265816552200-0000-AB-0D-
D3E614442CBBCF6137C82A04
```

Yang diberi label kuning (07-0C-AB-0D) merupakan nilai-nilai dalam hexa, Label hijau (2658050000F0-265816552200) adalah nomor smsc dan nomor handphone yang dibalik susunannya dengan aturan sebelum dibalik nomor tersebut harus ditambahkan huruf F jika berjumlah ganjil. Label biru muda (D3E614442CBBCF6137C82A04) merupakan isi pesan yang sudah mengalami perubahan format dari dec-septet-oktet seperti telah dijelaskan pada bab 2 sub bab PDU. Sedangkan yang tidak diberi label (91-1100-0000) merupakan nilai default.

Untuk mendapatkan format seperti itu, perintah yang dibuat adalah sebagai berikut:

```
txtPDU.Text = SMSC(cbosmsc) & ("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") &
(Hex((Len(ASCII2PDU(txtMessage.Text)) / 2))) & ASCII2PDU(txtMessage.Text)
```

SMSC(cbosmsc) merupakan fungsi untuk melakukan perubahan nilai smsc sesuai dengan yang diinputkan oleh user pada control cbosmsc. Penulis sudah melakukan pembalikan nomor smsc sesuai dengan format yang digunakan, sekaligus menyertai panjang nomor smsc dari tiap operator di depan nomor smsc tersebut. Seperti

```

If cbosmsc = "Excelcom+62818445009" Then
SMSC = "07912618485400F9"
End If

If cbosmsc = "IM3+62855000000" Then
SMSC = "07912658050000F0"
End If

```

Untuk mengetahui panjang dari nomor handphone tujuan sekaligus mengubahnya ke dalam hexa, maka dilakukan perintah sebagai berikut:

```
(Hex(Len(txtHP.Text)))
```

Fungsi Len dalam VB melakukan perhitungan terhadap panjang nomor handphone, kemudian fungsi Hex mengubah nilai tersebut ke dalam bilangan hexa. Untuk melakukan pembalikan terhadap nomor handphone yang inputkan oleh user tidak dapat dilakukan secara manual seperti pada pembalikan nilai smsc karena tentunya program belum mengetahui nomor tersebut, sehingga dibuat sub function Reverse, dengan perintah sebagai berikut:

```

Function Reverse(ByVal strChar As String) As String
Dim i As Integer
Dim position() As String
Dim intLength As Integer

intLength = Len(strChar)
ReDim position(intLength + 1)

For i = 1 To (intLength / 2)
    position(i) = Mid(strChar, i * 2, 1)
    position(i + 1) = Mid(strChar, i * 2 - 1, 1)
    Reverse = Reverse & position(i) & position(i + 1)
Next i

End Function

```

Fungsi utama dalam program di atas adalah fungsi Mid. “Mid menghasilkan karakter yang dihitung dari tengah-tengah string, dimulai dari titik awal tertentu” (Michael Halvorson, p.347).

Mid(strChar,i*2,1) dimana i dimulai dari 1 berarti mengambil nilai 1 digit dari string yang dimulai dari digit atau karakter ke dua dari string. Sebagai contoh

jika strChar nya adalah kata “bisa” maka dihasilkan “i”. Untuk posisi kedua dilakukan hal yang sama namun “i*2-1”, sehingga dihasilkan “b”, begitu seterusnya hingga $i = \text{jumlah karakter string} / 2$. Akhirnya posisi-posisi tersebut digabungkan dan disimpan ke dalam fungsi Reverse.

Sedangkan untuk mendapatkan panjang PDU, dilakukan perintah sebagai berikut

```
txtPanjangPDU = Val(Len("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") & (Hex((Len(ASCII2PDU(txtMessage.Text))
/ 2))) & ASCII2PDU(txtMessage.Text)) / 2)
```

Fungsi yang digunakan adalah Val(Len(...)) artinya menghitung jumlah nilai yang ada di dalam kurung. Untuk mengubah pesan ke dalam format PDU, penulis menggunakan program yang sudah dipublikasikan oleh Daniel di <http://www.averagecoder.net/>.

4.2.4 Koneksi antara Database MySQL dan VB

Dalam program ini database berguna sebagai tempat penyimpanan data-data seperti Nomor ID, Nama, dan Alamat dari pasien penderita dbd. Dengan adanya database ini maka data-data tersebut dapat disimpan dalam wadah yang memiliki kapasitas yang sangat besar jika dibandingkan dengan memory handphone.

Listing program untuk melakukan koneksi ke database sama seperti pada program sebelumnya.

4.2.5 Send To Many

Dalam mengirimkan suatu pesan tentunya memerlukan nomor tujuan dan isi pesan yang akan dikirimkan. Dalam program ini isi pesan memiliki format seperti:

“Kasus DBD : Nama, Alamat, Segera Penanggulangan”

Nama dan Alamat didapatkan dari tabel tb_entry_dbd, Sehingga untuk mendapatkan informasi pesan seperti di atas, listing programnya adalah sebagai berikut :

```
Set rsCek = conn.Execute("SELECT*FROM tb_entry_dbd WHERE fl_id = " & (txtID.Text) & """)
txtMessage.Text = "Kasus DBD : " + " " + "" & rsCek(1) + " " + "" & rsCek(2) + " , " + "Segera
Penanggulangan"
```

Perintahnya sama seperti pada pembahasan sebelumnya dengan menggunakan SELECT..FROM. rsCek(1) merupakan record pada field pertama dari tabel tb_entr_dbd

yaitu kolom Nama, dan rsCek(2) merupakan record pada field kedua dari tabel tb_entry_dbd yaitu kolom Alamat.

Dalam layanan send to many ini, nomor tujuan yang hendak dikirimkan adalah 4 nomor yang meliputi nomor Lurah, RW, RT, dan Kepala Keluarga. Untuk itu perlu dilakukan penyettingan nomor-nomor tujuan yang bersesuaian dengan identitas pasien. Seperti pada tampilan berikut :

Objek Informasi DBD	
Lurah Jati Asih	62811119116
RW 01	6285697178287
RT 01	6281399450484
Haris	622196141770

Gambar 4.10 System Informasi DBD

Untuk mendapatkan nomor-nomor tersebut , maka dilakukan program sebagai berikut:

```

Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_entry_dbd WHERE fl_id = " & (txtID.Text) & """)
txtMessage.Text = "Kasus DBD : " + " " + "" & rsCek(1) + " " + "" & rsCek(2) + " , " + "Segera Penanggulangan"

Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan = " & Left(txtID, 2) & """)
lbl1.Caption = "Lurah" + " " & rsCek(2)
txt1.Text = "" & rsCek(4)

Set rsCek = conn.Execute("SELECT*FROM tb_rw WHERE fl_kd_rw = " & Left(txtID, 4) & """)
lbl2.Caption = "" & rsCek(2)
txt2.Text = "" & rsCek(4)

Set rsCek = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt = " & Left(txtID, 6) & """)
lbl3.Caption = "" & rsCek(2)
txt3.Text = "" & rsCek(4)

Set rsCek = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk = " & Left(txtID, 8) & """)
lbl4.Caption = "" & rsCek(3)
txt4.Text = "" & rsCek(4)

```

Perintah ke 2 hingga ke 5 memiliki banyak persamaan, Namun untuk masing-masing tabel dipilih berdasarkan kode tabel masing-masing yang direpresentasikan oleh nomor ID. Untuk menampilkan nama dan nomor handphone lurah dari seorang pasien, dilakukan perintah sebagai berikut :

```
Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan = " &
Left(txtID, 2) & """)
lbl1.Caption = "Lurah" + " " & rsCek(2)
txt1.Text = "" & rsCek(4)
```

Perintah melakukan pemilihan sekaligus menampilkan nama dan nomor handphone lurah dari tabel kelurahan dimana kode kelurahan didapat dari dua digit sebelah kiri dari nomor ID (Left(txtID,2)). Kemudian masing-masing ditampilkan pada property label (lbl1.Caption) dan textbox (txt1.Text). Hal serupa dilakukan hingga pada tabel tb_kk.

Karena nomor tujuannya lebih dari satu, maka untuk melakukan pengiriman sms penulis menyiasatinya dengan menambahkan variable dan melakukan looping, seperti berikut :

```
Dim x As Integer
Dim i
If x = 1 Then
    txtHP.Text = txt1.Text
Elseif x = 2 Then
    txtHP.Text = txt2.Text
Elseif x = 3 Then
    txtHP.Text = txt3.Text
Elseif x = 4 Then
    txtHP.Text = txt4.Text
End If

For x = 1 To 4
    If x = 1 Then
        txtHP.Text = txt1.Text
    Elseif x = 2 Then
        txtHP.Text = txt2.Text
    Elseif x = 3 Then
        txtHP.Text = txt3.Text
    Elseif x = 4 Then
        txtHP.Text = txt4.Text
    End If
    i = Len(Hex((Len(ASCII2PDU(txtMessage.Text)) / 2)))
    If i Mod 2 = 1 Then
        txtPDU.Text = SMSC(cbosmsc) & ("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") & (Hex((Len(ASCII2PDU(txtMessage.Text)) / 2))) &
ASCII2PDU(txtMessage.Text)
        txtPanjangPDU = Val(Len(("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") & (Hex((Len(ASCII2PDU(txtMessage.Text)) / 2))) &
ASCII2PDU(txtMessage.Text)) / 2)
```

```

Else
  txtPDU.Text = SMSC(cbosmsc) & ("11000") & (Hex(Len(txtHP.Text))) & ("91") &
  Reverse(InputHP(txtHP.Text)) & ("0000AB") & (Hex((Len(ASCII2PDU(txtMessage.Text)) / 2))) &
  ASCII2PDU(txtMessage.Text)
  txtPanjangPDU = Val(Len("11000") & (Hex(Len(txtHP.Text))) & ("91") &
  Reverse(InputHP(txtHP.Text)) & ("0000AB") & (Hex((Len(ASCII2PDU(txtMessage.Text)) / 2))) &
  ASCII2PDU(txtMessage.Text)) / 2)
End If

```

Keberhasilan pengiriman pesan ditandai dengan label hijau di bawah tombol End, seperti berikut:



Gambar 4.11 Proses Pengiriman Informasi DBD

Berikut adalah potongan programnya:

```

Call Send(txtPDU.Text & Chr(26))
If WaitForSuccess(buf) Then
  lbl7.BackColor = &H80000001 ' warna hijau
Else
  lbl8.BackColor = &H80000008 ' warna merah hati
End If

```

Jika pengiriman sukses maka label akan berwarna hijau, namun jika gagal maka akan diberi warna merah hati. Sukses maupun gagalnya pengiriman dapat terjadi sebagaimana

gagal ataupun suksesnya pengiriman sms pada umumnya.

Berikut adalah hasil pengiriman sms pada beberapa handphone penerima :



Gambar 4.12 Hasil Pengiriman Informasi DBD di Handphone Penerima

BAB 5

KESIMPULAN

1. Perancangan Layanan send to many ini memanfaatkan ponsel sony ericsson w200i hanya sebagai pintu keluar untuk mengirimkan pesan. Alur atau proses pengiriman pesan mengikuti procedure seperti sms pada umumnya.
2. Setelah dilakukan ujicoba terhadap pengiriman pesan ke beberapa nomor tujuan, program ini berjalan dengan baik.
3. Dalam uji coba pengiriman ke beberapa nomor tujuan, isi pesan yang dikirim oleh sms sender dapat diterima dengan baik di beberapa nomor tujuan tersebut.
4. Keunggulan layanan send to many ini diantaranya :
 - a. kemampuan menampung data yang lebih besar dibandingkan memory sebuah handphone.
 - b. efisiensi waktu pengiriman pesan

DAFTAR ACUAN

[1] Pasaribu, Parlin. (2006). *Evolusi Teknologi Telekomunikasi Bergerak: 1G – 4G*.

<http://ilmukomputer.com>

[2] Purnomo, Adi. *Sistem kerja sms*. <http://javaku.wordpress.com>

[3] Swastika, Windra. (2006). *VB & MySQL, proyek membuat program general ledger (seri 1)*. Malang : Dian Rakyat

DAFTAR PUSTAKA

Halvorson, Michael. *Step by step microsoft visual basic 6.0 professional*. Jakarta : PT Elex Media Komputindo

Nurjadi, Joko.(2008).*Konsep pembuatan sms gateway*. PCMedia

Pasaribu, Parlin. (2006).*Evolusi Teknologi Telekomunikasi Bergerak: 1G – 4G*.

<http://ilmukomputer.com>

Prihatini, Ekawati.(2006).*Aspek keamanan pada jalur komunikasi short message service*.

Bandung : ITB

Purnomo, Adi. *Sistem kerja sms*. <http://javaku.wordpress.com>

Rozidi, Romzi Imron.(2004).*Membuat sendiri sms gateway (ESME) berbasis protokol SMPP*. Yogyakarta : Penerbit ANDI

Swastika, Windra. (2006). *VB & MySQL, proyek membuat program general ledger(seri 1)*.

Malang : Dian Rakyat

(2007). *AT Commands For Sony Ericsson Phones*. Sony Ericsson

LAMPIRAN

Program Layanan Send To Many

3. Program Entry Data Kelurahan

```
Option Explicit
Dim conn As ADODB.Connection

Private Sub cmdDelete_Click()
On Error GoTo pesan
Dim rsCek As New ADODB.Recordset
If txtKode.Text = "" Then
MsgBox "Maaf kode kelurahan harus dimasukkan", vbInformation + vbOKOnly,
"information"
Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & "")
If rsCek.EOF Then
MsgBox "Maaf kode kelurahan tidak ada di dalam database", vbInformation +
vbOKOnly, "information"
Else
conn.Execute ("DELETE FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & "")
Call tampil("LOAD")
MsgBox "Data Kelurahan Sukses di DELETE", vbInformation + vbOKOnly, "Information"
End If
End If
Exit Sub

pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub cmdEdit_Click()
On Error GoTo pesan
Dim rsCek As New ADODB.Recordset
If txtKode.Text = "" Then
MsgBox "Maaf Kode Kelurahan Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & "")
If rsCek.EOF Then
MsgBox "Maaf Kode Kelurahan yang Anda Masukkan Tidak Ada di Dalam Database",
vbInformation + vbOKOnly, "information"
Else
If txtNmKelurahan = "" Then
MsgBox "Masukkan Nama Kelurahan", vbInformation + vbOKOnly, "information"
txtNmKelurahan.SetFocus
Exit Sub
End If

```

(lanjutan)

```
If txtNmLurah = "" Then
    MsgBox "Masukkan Nama Lurah", vbInformation + vbOKOnly, "information"
    txtNmLurah.SetFocus
    Exit Sub
End If

If txtHpLurah = "" Then
    MsgBox "Masukkan Nomor HP Lurah", vbInformation + vbOKOnly, "information"
    txtHpLurah.SetFocus
    Exit Sub
Else
    MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"

    conn.Execute ("UPDATE tb_kelurahan SET fl_nm_kelurahan=" &
    (txtNmKelurahan.Text) & ", fl_nm_Lurah =" & (txtNmLurah.Text) & ", fl_hp_Lurah =" &
    (txtHpLurah.Text) & " WHERE fl_kd_kelurahan =" & (txtKode.Text) & """)
    Call tampil("LOAD")
    MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
    End If
End If
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Private Sub cmdFind_Click()
    Call Clear
    If txtSearch.Text = "" Then
        MsgBox "Pilh Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
        "information"
        txtSearch.SetFocus
    Else
        Call tampil("CARI")
    End If
End Sub

Private Sub cmdRefresh_Click()
    Call Clear
    Call tampil("LOAD")
End Sub

Private Sub cmdSimpan_Click()
    Dim rsCek As New ADODB.Recordset

    If txtKode.Text = "" Then
        MsgBox "Masukkan Kode Kelurahan", vbInformation + vbOKOnly, "information"
        txtKode.SetFocus
    Exit Sub
```

(lanjutan)

```
Else
  Call connect
  Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan=" &
(txtKode.Text) & "")
  If Not rsCek.EOF Then
    MsgBox "Maaf Kode Kelurahan Sudah Ada Dalam Database", vbInformation +
vbOKOnly, "information"
    txtKode.SetFocus
    Exit Sub
  End If
End If

If txtNmKelurahan = "" Then
  MsgBox "Masukkan Nama Kelurahan", vbInformation + vbOKOnly, "information"
  txtNmKelurahan.SetFocus
  Exit Sub
End If

If txtNmLurah = "" Then
  MsgBox "Masukkan Nama Lurah", vbInformation + vbOKOnly, "information"
  txtNmLurah.SetFocus
  Exit Sub
End If

If txtHpLurah = "" Then
  MsgBox "Masukkan Nomor HP Lurah", vbInformation + vbOKOnly, "information"
  txtHpLurah.SetFocus
  Exit Sub
Else
  Call connect
  conn.Execute "INSERT INTO tb_kelurahan(
fl_kd_kelurahan,fl_nm_kelurahan,fl_nm_lurah,fl_hp_lurah ) " _
& "VALUES(" & (txtKode.Text) & "," & (txtNmKelurahan.Text) & "," &
(txtNmLurah.Text) & "," & (txtHpLurah.Text) & ")"
  Call tampil("LOAD")
End If
End Sub

Sub tampil(key As String)
  On Error GoTo pesan
  Dim rsData As New ADODB.Recordset
  Dim fieldKey As String
  Dim istview
  Call connect
  If key = "LOAD" Then

    Set rsData = conn.Execute("SELECT*FROM tb_kelurahan ORDER BY fl_kd_kelurahan
ASC")
  Else
    Select Case cmbSearch.ListIndex
    Case 0
      fieldKey = "fl_kd_kelurahan"
    Case 1
      fieldKey = "fl_nm_kelurahan"
    Case 2
      fieldKey = "fl_nm_lurah"
    Case 3
      fieldKey = "fl_hp_lurah"
```

(lanjutan)

```
End Select
Set rsData = conn.Execute("SELECT*FROM tb_kelurahan WHERE " & fieldKey & " Like
%" & (txtSearch.Text) & "%' ORDER BY fl_kd_kelurahan Asc ")
If rsData.EOF Then
    MsgBox "Maaf Data yang Anda Cari Tidak DiTemukan", vbInformation + vbOKOnly,
"information"
End If
End If
Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
    Set istview = lvData.ListItems.Add(, rsData(0))
    istview.ListSubItems.Add(, rsData(2))
    istview.ListSubItems.Add(, rsData(3))
    istview.ListSubItems.Add(, rsData(4))

rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSCComctlLib.ColumnHeader)
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
    If lvData.SortOrder = lvwAscending Then
        lvData.SortOrder = lvwDescending
    Else
        lvData.SortOrder = lvwAscending
    End If
Else
    lvData.Sorted = True
    lvData.SortKey = ColumnHeader.SubItemIndex
    lvData.SortOrder = lvwAscending
End If
End Sub

Sub grid()
lvData.ColumnHeaders.Item(1).Text = "Kode"
lvData.ColumnHeaders.Item(1).Width = "700"
lvData.ColumnHeaders.Item(2).Text = "Nama Kelurahan"
lvData.ColumnHeaders.Item(2).Width = "3000"
lvData.ColumnHeaders.Item(3).Text = "Nama Lurah"
lvData.ColumnHeaders.Item(3).Width = "3000"
lvData.ColumnHeaders.Item(4).Text = "No HP Lurah"
lvData.ColumnHeaders.Item(4).Width = "2500"

End Sub

Sub Clear()
txtKode = ""
txtNmKelurahan = ""
txtNmLurah = ""
txtHpLurah = ""
txtKode.SetFocus
```

(lanjutan)

```
End Sub
```

```
Sub connect()
```

```
'connect to MySQL server using MySQL ODBC 5.0 Driver
```

```
Set conn = New ADODB.Connection
```

```
conn.Open "Dsn=ADI"
```

```
conn.CursorLocation = adUseClient
```

```
End Sub
```

4. Program Entry Data RW

```
Option Explicit
```

```
Dim conn As New ADODB.Connection
```

```
Dim rs As New ADODB.Recordset
```

```
Sub Combo()
```

```
On Error GoTo pesan
```

```
Dim rsCb As New ADODB.Recordset
```

```
Call connect
```

```
Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_kelurahan,',';fl_nm_kelurahan) as  
kelurahan FROM tb_kelurahan ORDER BY fl_kd_kelurahan ASC ")
```

```
cmbKelurahan.Clear
```

```
Do While Not rsCb.EOF
```

```
    cmbKelurahan.AddItem rsCb(0)
```

```
rsCb.MoveNext
```

```
Loop
```

```
Exit Sub
```

```
pesan:
```

```
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
```

```
End Sub
```

```
Private Sub cmbKelurahan_GotFocus()
```

```
    Call Combo
```

```
End Sub
```

```
Private Sub cmdDelete_Click()
```

```
On Error GoTo pesan
```

```
Dim rsCek As New ADODB.Recordset
```

```
Dim kdKunci As String
```

```
kdKunci = Left(cmbKelurahan.Text, 2) & txtKode.Text
```

```
If txtKode.Text = "" Then
```

```
    MsgBox "Maaf Kode RW Harus Dimasukkan", vbInformation + vbOKOnly, "information"
```

```
    cmbKelurahan.SetFocus
```

(lanjutan)

```
Exit Sub
```

```
Else
```

```
Call connect
```

```
Set rsCek = conn.Execute("SELECT*FROM tb_rw WHERE fl_kd_rw=" & (kdKunci) & """)
```

```
If rsCek.EOF Then
```

```
MsgBox "Maaf Kode RW Tidak Ada di Dalam Database", vbInformation + vbOKOnly, "information"
```

```
Else
```

```
conn.Execute ("DELETE FROM tb_rw WHERE fl_kd_rw =" & (kdKunci) & """)
```

```
Call tampil("LOAD")
```

```
MsgBox "Data RW Sukses di DELETE", vbInformation + vbOKOnly, "Information"
```

```
End If
```

```
End If
```

```
Exit Sub
```

```
pesan:
```

```
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
```

```
End Sub
```

```
Private Sub cmdEdit_Click()
```

```
On Error GoTo pesan
```

```
Dim rsCek As New ADODB.Recordset
```

```
Dim kdKunci As String
```

```
kdKunci = Left(cmbKelurahan.Text, 2) & txtKode.Text
```

```
If cmbKelurahan.Text = "" Then
```

```
MsgBox "Maaf Kode Kelurahan Harus Anda Masukkan", vbInformation + vbOKOnly, "information"
```

```
cmbKelurahan.SetFocus
```

```
Exit Sub
```

```
End If
```

```
If txtKode.Text = "" Then
```

```
MsgBox "Maaf Kode Kelurahan Harus Anda Masukkan", vbInformation + vbOKOnly, "information"
```

```
cmbKelurahan.SetFocus
```

```
Exit Sub
```


(lanjutan)

```
Else
  Call connect
  Set rsCek = conn.Execute("SELECT*FROM tb_rw WHERE fl_kd_rw=" & (kdKunci) &
  "")
  If rsCek.EOF Then
    MsgBox "Maaf Kode RW yang Anda Masukkan Tidak Ada di Dalam Database",
    vbInformation + vbOKOnly, "information"
  Else
    If txtNmRW = "" Then
      MsgBox "Masukkan Nama RW", vbInformation + vbOKOnly, "information"
      txtNmRW.SetFocus
    Exit Sub
    End If

    If txtNmKetuaRW = "" Then
      MsgBox "Masukkan Nama Ketua RW", vbInformation + vbOKOnly, "information"
      txtNmKetuaRW.SetFocus
    Exit Sub
    End If

    If txtHpRW = "" Then
      MsgBox "Masukkan Nomor HP Ketua RW", vbInformation + vbOKOnly,
      "information"
      txtHpRW.SetFocus
    Exit Sub
    Else

      MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"

      conn.Execute ("UPDATE tb_rw SET fl_kd_kelurahan=" & (Left(cmbKelurahan, 2)) &
      ",fl_nm_rw=" & (txtNmRW.Text) & ", fl_nm_ketua_rw =" & (txtNmKetuaRW.Text) & ",
      fl_hp_ketua_rw =" & (txtHpRW.Text) & " WHERE fl_kd_rw =" & (kdKunci) & "")
      Call tampil("LOAD")
      MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
    End If
  End If
End If
Exit Sub
```

(lanjutan)

```
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Sub tampil(key As String)
    On Error GoTo pesan
    Dim rsData As New ADODB.Recordset
    Dim fieldKey As String
    Dim istview
    Call connect
    If key = "LOAD" Then
        Set rsData = conn.Execute("SELECT*FROM tb_rw ORDER BY fl_kd_rw ASC")
    Else
        If key = "COMBO" Then
            Set rsData = conn.Execute("SELECT*FROM tb_rw WHERE fl_kd_rw = " &
(Left(cmbKelurahan.Text, 2)) & " ORDER BY fl_kd_RW")
        Else
            Select Case cmbSearch.ListIndex
            Case 0
                fieldKey = "fl_kd_rw"
            Case 1
                fieldKey = "fl_nm_rw"
            Case 2
                fieldKey = "fl_nm_ketua_rw"
            Case 3
                fieldKey = "fl_hp_rw"

            End Select
            Set rsData = conn.Execute("SELECT*FROM tb_rw WHERE " & fieldKey & " Like '%" &
(txtSearch.Text) & "%' ORDER BY fl_kd_rw Asc ")

            End If
        End If
    Call grid
```

(lanjutan)

```
IvData.ListItems.Clear
Do While Not rsData.EOF
    Set istview = IvData.ListItems.Add(, , rsData(0))
    istview.ListSubItems.Add , , rsData(2)
    istview.ListSubItems.Add , , rsData(3)
    istview.ListSubItems.Add , , rsData(4)

rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub cmdFind_Click()
Call Clear
If txtSearch.Text = "" Then
    MsgBox "Pilih Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
"information"
    txtSearch.SetFocus
Else
Call tampil("CARI")
End If
End Sub

Private Sub cmdRefresh_Click()
    Call Clear
    Call tampil("LOAD")
End Sub

Private Sub cmdSimpan_Click()

Dim rsCek As New ADODB.Recordset
Dim kdKunci As String
kdKunci = Left(cmbKelurahan.Text, 2) & txtKode.Text

If cmbKelurahan.Text = "" Then
    MsgBox "Masukkan Pilihan Kode Kelurahan", vbInformation + vbOKOnly, "information"
```

(lanjutan)

```
cmbKelurahan.SetFocus
Exit Sub
End If
If cmbKelurahan.Text = "--pilih--" Then
    MsgBox "Masukkan Pilihan Kode Kelurahan", vbInformation + vbOKOnly, "information"
    cmbKelurahan.SetFocus
    Exit Sub
End If

If txtKode.Text = "" Then
    MsgBox "Masukkan Kode RW", vbInformation + vbOKOnly, "information"
    txtKode.SetFocus
    Exit Sub
Else
    Call connect
    Set rsCek = conn.Execute("SELECT*FROM tb_RW WHERE fl_kd_rw=" & (kdKunci) &
    "")
    If Not rsCek.EOF Then
        MsgBox "Maaf Kode RW Sudah Ada Dalam Database", vbInformation + vbOKOnly,
        "information"
        txtKode.SetFocus
        Exit Sub
    End If
End If

If txtNmRW.Text = "" Then
    MsgBox "Masukkan Nama RW", vbInformation + vbOKOnly, "information"
    txtNmRW.SetFocus
    Exit Sub
End If

If txtNmKetuaRW.Text = "" Then
    MsgBox "Masukkan Nama Ketua RW", vbInformation + vbOKOnly, "information"
    txtNmKetuaRW.SetFocus
    Exit Sub
End If

If txtHpRW.Text = "" Then
    MsgBox "Masukkan Nomor HP Ketua RW", vbInformation + vbOKOnly, "information"
```

(lanjutan)

```
txtHpRW.SetFocus
Exit Sub
Else
Call connect
conn.Execute "INSERT INTO tb_rw( fl_kd_rw,fl_nm_rw,fl_nm_ketua_rw,fl_hp_ketua_rw )
" _
    & "VALUES(" & (kdKunci) & "," & (txtNmRW.Text) & "," &
(txtNmKetuaRW.Text) & "," & (txtHpRW.Text) & ")"
Call tampil("LOAD")
End If

End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSCComctlLib.ColumnHeader)
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
    If lvData.SortOrder = lvwAscending Then
        lvData.SortOrder = lvwDescending
    Else
        lvData.SortOrder = lvwAscending
    End If
Else
    lvData.Sorted = True
    lvData.SortKey = ColumnHeader.SubItemIndex
    lvData.SortOrder = lvwAscending
End If
End Sub

Sub grid()
    lvData.ColumnHeaders.Item(1).Text = "Kode"
    lvData.ColumnHeaders.Item(1).Width = "700"
    lvData.ColumnHeaders.Item(2).Text = "Nama RW"
    lvData.ColumnHeaders.Item(2).Width = "3000"
    lvData.ColumnHeaders.Item(3).Text = "Nama Ketua RW"
    lvData.ColumnHeaders.Item(3).Width = "3000"
    lvData.ColumnHeaders.Item(4).Text = "No HP Ketua RW"
    lvData.ColumnHeaders.Item(4).Width = "2500"
End Sub
```

(lanjutan)

```
Sub Clear()  
    txtKode = ""  
    txtNmRW = ""  
    txtNmKetuaRW = ""  
    txtHpRW = ""  
    txtKode.SetFocus  
End Sub  
Sub connect()  
'connect to MySQL server using MySQL ODBC 3.51 Driver  
Set conn = New ADODB.Connection  
conn.Open "Dsn=ADI"  
conn.CursorLocation = adUseClient  
End Sub
```

5. Program Entry Data RT

```
Option Explicit  
Dim conn As New ADODB.Connection  
Sub connect()  
    Set conn = New ADODB.Connection  
    conn.Open "Dsn=ADI"  
    conn.CursorLocation = adUseClient  
End Sub  
  
Private Sub cmbRW_Click()  
    Call tampil("COMBO")  
End Sub  
  
Private Sub cmbRW_GotFocus()  
    Call Combo  
End Sub  
  
Private Sub cmdDelete_Click()  
    On Error GoTo pesan  
    Dim rsCek As New ADODB.Recordset  
    Dim kdKunci As String  
    kdKunci = Left(cmbRW.Text, 4) & txtKode.Text  
    If txtKode.Text = "" Then
```

(lanjutan)

```
MsgBox "Maaf Kode RT Harus Dimasukkan", vbInformation + vbOKOnly, "information"
txtKode.SetFocus
Exit Sub

Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt=" & (kdKunci) & "")
If rsCek.EOF Then
MsgBox "Maaf Kode RT Tidak Ada di Dalam Database", vbInformation + vbOKOnly,
"information"
Else
conn.Execute ("DELETE FROM tb_rt WHERE fl_kd_rt =" & (kdKunci) & "")
Call tampil("LOAD")
MsgBox "Data RT Sukses di DELETE", vbInformation + vbOKOnly, "Information"
End If

End If
Exit Sub

pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub cmdEdit_Click()
On Error GoTo pesan
Dim rsCek As New ADODB.Recordset
Dim kdKunci As String
kdKunci = Left(cmbRW.Text, 4) & txtKode.Text

If cmbRW.Text = "" Then
MsgBox "Maaf Kode RW Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
cmbRW.SetFocus
Exit Sub
End If

If cmbRW.Text = "--pilih--" Then
MsgBox "Maaf Kode RW Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
```

(lanjutan)

```
cmbRW.SetFocus
Exit Sub
End If

If txtKode.Text = "" Then
    MsgBox "Maaf Kode RT Harus Anda Masukkan", vbInformation + vbOKOnly,
    "information"
    txtKode.SetFocus
    Exit Sub
Else
    Call connect
    Set rsCek = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt=" & (kdKunci) & "")
    If rsCek.EOF Then
        MsgBox "Maaf Kode RT yang Anda Masukkan Tidak Ada di Dalam Database",
        vbInformation + vbOKOnly, "information"
    Else
        If txtNmRT = "" Then
            MsgBox "Masukkan Nama RT", vbInformation + vbOKOnly, "information"
            txtNmRT.SetFocus
            Exit Sub
        End If

        If txtNmKetuaRT = "" Then
            MsgBox "Masukkan Nama Ketua RT", vbInformation + vbOKOnly, "information"
            txtNmKetuaRT.SetFocus
            Exit Sub
        End If

        If txtHpRT = "" Then
            MsgBox "Masukkan Nomor HP Ketua RT", vbInformation + vbOKOnly, "information"
            txtHpRT.SetFocus
            Exit Sub
        Else
            MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"

            conn.Execute ("UPDATE tb_rt SET fl_kd_rw = " & (Left(cmbRW, 4)) & ",fl_nm_rt="
            & (txtNmRT.Text) & ", fl_nm_ketua_rt =" & (txtNmKetuaRT.Text) & ", fl_hp_ketua_rt ="
```


(lanjutan)

```
& (txtHpRT.Text) & "' WHERE fl_kd_rt =' & (kdKunci) & "'"
    Call tampil("LOAD")
    MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
End If
End If
End If
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Private Sub cmdFind_Click()
    Call Clear
    If txtSearch.Text = "" Then
        MsgBox "Pilih Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
        "information"
        txtSearch.SetFocus
    Else
        Call tampil("CARI")
    End If
End Sub

Private Sub cmdRefresh_Click()
    Call Clear
    Call tampil("LOAD")
End Sub

Private Sub cmdSimpan_Click()
    Dim rsCek As New ADODB.Recordset
    Dim kdKunci As String
    kdKunci = Left(cmbRW.Text, 4) & txtKode.Text

    If cmbRW.Text = "" Then
```

(lanjutan)

```
MsgBox "Masukkan Kode RW", vbInformation + vbOKOnly, "information"
cmbRW.SetFocus
Exit Sub
End If

If cmbRW.Text = "--pilih--" Then
MsgBox "Masukkan Kode RW", vbInformation + vbOKOnly, "information"
cmbRW.SetFocus
Exit Sub
End If

If txtKode.Text = "" Then
MsgBox "Masukkan Kode RT", vbInformation + vbOKOnly, "information"
txtKode.SetFocus
Exit Sub
Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt = " & (kdKunci) &
"")
If Not rsCek.EOF Then
MsgBox "Data RW Sudah Ada Di Dalam Database", vbInformation + vbOKOnly,
"information"
End If
End If

If txtNmRT.Text = "" Then
MsgBox "Masukkan Nama RT", vbInformation + vbOKOnly, "information"
txtNmRT.SetFocus
Exit Sub
End If

If txtNmKetuaRT.Text = "" Then
MsgBox "Masukkan Ketua RT", vbInformation + vbOKOnly, "information"
txtNmKetuaRT.SetFocus
Exit Sub
End If

If txtHpRT.Text = "" Then
MsgBox "Masukkan No HP Ketua RT", vbInformation + vbOKOnly, "information"
```

(lanjutan)

```
txtHpRT.SetFocus
Exit Sub
Else
    Call connect
    conn.Execute "INSERT INTO tb_rt( fl_kd_rt,fl_nm_rt,fl_nm_ketua_rt,fl_hp_ketua_rt ) " &
"VALUES(" & (kdKunci) & "," & (txtNmRT.Text) & "," & (txtNmKetuaRT.Text) & "," &
& (txtHpRT.Text) & ")"
    Call tampil("LOAD")
    MsgBox "Data Sukses diInput", vbInformation + vbOKOnly, "information"
End If
End Sub

Sub tampil(key As String)
On Error GoTo pesan
Dim rsData As New ADODB.Recordset
Dim fieldKey As String
Dim istview
Dim kdKunci As String
kdKunci = Left(cmbRW.Text, 4) & txtKode.Text

Call connect
If key = "LOAD" Then
    Set rsData = conn.Execute("SELECT*FROM tb_rt ORDER BY fl_kd_rt ASC")
Else
    If key = "COMBO" Then
        Set rsData = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt = " & (kdKunci) & "
ORDER BY fl_kd_rt ASC")
    Else
        Select Case cmbSearch.ListIndex
        Case 0
            fieldKey = "fl_kd_rt"
        Case 1
            fieldKey = "fl_nm_rt"
        Case 2
            fieldKey = "fl_nm_ketua_rt"
        Case 3
            fieldKey = "fl_hp_ketua_rt"
        End Select
    End If
End If
```

(lanjutan)

```
End Select
Set rsData = conn.Execute("SELECT*FROM tb_rt WHERE " & fieldKey & " Like '%" &
(txtSearch.Text) & "%' ORDER BY fl_kd_rt Asc ")
End If
End If

Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
Set istview = lvData.ListItems.Add(, , rsData(0))
istview.ListSubItems.Add , , rsData(2)
istview.ListSubItems.Add , , rsData(3)
istview.ListSubItems.Add , , rsData(4)
rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSCComctlLib.ColumnHeader)
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
If lvData.SortOrder = lvwAscending Then
lvData.SortOrder = lvwDescending
Else
lvData.SortOrder = lvwAscending
End If
Else
lvData.Sorted = True
lvData.SortKey = ColumnHeader.SubItemIndex
lvData.SortOrder = lvwAscending
End If
End Sub

Sub grid()
lvData.ColumnHeaders.Item(1).Text = "Kode"
lvData.ColumnHeaders.Item(1).Width = "700"
```

(lanjutan)

```
lvData.ColumnHeaders.Item(2).Text = "Nama RT"
lvData.ColumnHeaders.Item(2).Width = "3000"
lvData.ColumnHeaders.Item(3).Text = "Nama Ketua RT"
lvData.ColumnHeaders.Item(3).Width = "3000"
lvData.ColumnHeaders.Item(4).Text = "No HP Ketua RT"
lvData.ColumnHeaders.Item(4).Width = "2500"
End Sub

Sub Combo()
On Error GoTo pesan
Dim rsCb As New ADODB.Recordset
Call connect
Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_rw,':',fl_nm_rw) as rw FROM tb_rw
ORDER BY fl_kd_rw ASC")
cmbRW.Clear
Do While Not rsCb.EOF
    cmbRW.AddItem rsCb(0)
rsCb.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Sub Clear()
txtKode = ""
txtNmRT = ""
txtNmKetuaRT = ""
txtHpRT = ""
txtKode.SetFocus
End Sub
```

6. Program Entry Data KK

```
Option Explicit
Dim conn As New ADODB.Connection
Sub connect()
Set conn = New ADODB.Connection
conn.Open "Dsn=ADI"
conn.CursorLocation = adUseClient
```

(lanjutan)

```
End Sub
```

```
Private Sub cmbRT_Click()
```

```
    Call tampil("COMBO")
```

```
End Sub
```

```
Private Sub cmbRT_GotFocus()
```

```
    Call Combo
```

```
End Sub
```

```
Private Sub cmdDelete_Click()
```

```
On Error GoTo pesan
```

```
Dim rsCek As New ADODB.Recordset
```

```
Dim kdKunci As String
```

```
kdKunci = Left(cmbRT.Text, 6) & txtKode.Text
```

```
If txtKode.Text = "" Then
```

```
    MsgBox "Maaf Kode KK Harus Dimasukkan", vbInformation + vbOKOnly, "information"
```

```
    txtKode.SetFocus
```

```
    Exit Sub
```

```
Else
```

```
    Call connect
```

```
    Set rsCek = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk=" & (kdKunci) & """)
```

```
    If rsCek.EOF Then
```

```
        MsgBox "Maaf Kode KK Tidak Ada di Dalam Database", vbInformation + vbOKOnly, "information"
```

```
    Else
```

```
        conn.Execute ("DELETE FROM tb_kk WHERE fl_kd_kk =" & (kdKunci) & """)
```

```
        Call tampil("LOAD")
```

```
        MsgBox "Data KK Sukses di DELETE", vbInformation + vbOKOnly, "Information"
```

```
    End If
```

```
End If
```

```
Exit Sub
```

```
pesan:
```

```
    MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
```

```
End Sub
```

(lanjutan)

```
Private Sub cmdEdit_Click()
On Error GoTo pesan
Dim rsCek As New ADODB.Recordset
Dim kdKunci As String
kdKunci = Left(cmbRT.Text, 6) & txtKode.Text

If cmbRT.Text = "" Then
MsgBox "Maaf Kode RT Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
cmbRT.SetFocus
Exit Sub
End If

If cmbRT.Text = "--pilih--" Then
MsgBox "Maaf Kode RT Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
cmbRT.SetFocus
Exit Sub
End If

If txtKode.Text = "" Then
MsgBox "Maaf Kode KK Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
txtKode.SetFocus
Exit Sub
Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk=" & (kdKunci) &
"")
If rsCek.EOF Then
MsgBox "Maaf Kode KK yang Anda Masukkan Tidak Ada di Dalam Database",
vbInformation + vbOKOnly, "information"
Else
If txtNmKK = "" Then
MsgBox "Masukkan Nama KK (Alamat)", vbInformation + vbOKOnly, "information"
txtNmKK.SetFocus
Exit Sub
End If

```

(lanjutan)

```
If txtNmKepalaKK = "" Then
    MsgBox "Masukkan Nama Kepala KK", vbInformation + vbOKOnly, "information"
    txtNmKepalaKK.SetFocus
    Exit Sub
End If

If txtHpKK = "" Then
    MsgBox "Masukkan Nomor HP Kepala KK", vbInformation + vbOKOnly,
"information"
    txtHpKK.SetFocus
    Exit Sub
Else
    MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"

    conn.Execute ("UPDATE tb_kk SET fl_kd_rt =" & (Left(cmbRT, 6)) & ",fl_nm_kk="
& (txtNmKK.Text) & ", fl_nm_kepala_kk =" & (txtNmKepalaKK.Text) & ",
fl_hp_kepala_kk =" & (txtHpKK.Text) & " WHERE fl_kd_kk =" & (kdKunci) & """)
    Call tampil("LOAD")
    MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
    End If
End If
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Private Sub cmdFind_Click()
    Call Clear
    If txtSearch.Text = "" Then
        MsgBox "Pilih Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
"information"
```


(lanjutan)

```
txtSearch.SetFocus

Else
Call tampil("CARI")
End If
End Sub

Private Sub cmdRefresh_Click()
    Call Clear
    Call tampil("LOAD")
End Sub

Private Sub cmdSimpan_Click()
    Dim rsCek As New ADODB.Recordset
    Dim kdKunci As String
    kdKunci = Left(cmbRT.Text, 6) & txtKode.Text

    If cmbRT.Text = "" Then
        MsgBox "Masukkan Kode RT", vbInformation + vbOKOnly, "information"
        cmbRT.SetFocus
        Exit Sub
    End If

    If cmbRT.Text = "--pilih--" Then
        MsgBox "Masukkan Kode RT", vbInformation + vbOKOnly, "information"
        cmbRT.SetFocus
        Exit Sub
    End If

    If txtKode.Text = "" Then
        MsgBox "Masukkan Kode KK", vbInformation + vbOKOnly, "information"
        txtKode.SetFocus
        Exit Sub
    Else
        Call connect
        Set rsCek = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk = "" & (kdKunci) & """)
        If Not rsCek.EOF Then
            MsgBox "Data KK Sudah Ada Di Dalam Database", vbInformation + vbOKOnly,
```

(lanjutan)

```
"information"  
    End If  
End If  
  
If txtNmKK.Text = "" Then  
    MsgBox "Masukkan Nama KK", vbInformation + vbOKOnly, "information"  
    txtNmKK.SetFocus  
    Exit Sub  
End If  
  
If txtNmKepalaKK.Text = "" Then  
    MsgBox "Masukkan Kepala KK", vbInformation + vbOKOnly, "information"  
    txtNmKepalaKK.SetFocus  
    Exit Sub  
End If  
  
If txtHpKK.Text = "" Then  
    MsgBox "Masukkan No HP Kepala KK", vbInformation + vbOKOnly, "information"  
    txtHpKK.SetFocus  
    Exit Sub  
Else  
    Call connect  
    conn.Execute "INSERT INTO tb_kk(  
fl_kd_kk,fl_nm_kk,fl_nm_kepala_kk,fl_hp_kepala_kk ) " & "VALUES(" & (kdKunci) & ","  
& (txtNmKK.Text) & "," & (txtNmKepalaKK.Text) & "," & (txtHpKK.Text) & ")"  
    Call tampil("LOAD")  
    MsgBox "Data Sukses diInput", vbInformation + vbOKOnly, "information"  
End If  
End Sub  
  
Sub tampil(key As String)  
On Error GoTo pesan  
Dim rsData As New ADODB.Recordset  
Dim fieldKey As String  
Dim istview  
Dim kdKunci As String  
kdKunci = Left(cmbRT.Text, 6) & txtKode.Text
```

(lanjutan)

```
Call connect
If key = "LOAD" Then
    Set rsData = conn.Execute("SELECT*FROM tb_kk ORDER BY fl_kd_kk ASC")
Else
    If key = "COMBO" Then
        Set rsData = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk = '" & (kdKunci) &
        "' ORDER BY fl_kd_kk ASC")
    Else
        Select Case cmbSearch.ListIndex
            Case 0
                fieldKey = "fl_kd_kk"
            Case 1
                fieldKey = "fl_nm_kk"
            Case 2
                fieldKey = "fl_nm_kepala_kk"
            Case 3
                fieldKey = "fl_hp_kepala_kk"
        End Select
        Set rsData = conn.Execute("SELECT*FROM tb_kk WHERE " & fieldKey & " Like '%" &
        (txtSearch.Text) & "%' ORDER BY fl_kd_kk Asc ")
    End If
End If

Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
    Set istview = lvData.ListItems.Add(, , rsData(0))
    istview.ListSubItems.Add , , rsData(2)
    istview.ListSubItems.Add , , rsData(3)
    istview.ListSubItems.Add , , rsData(4)
    rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSComctlLib.ColumnHeader)
```

(lanjutan)

```
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
    If lvData.SortOrder = lvwAscending Then
        lvData.SortOrder = lvwDescending
    Else
        lvData.SortOrder = lvwAscending
    End If
Else
    lvData.Sorted = True
    lvData.SortKey = ColumnHeader.SubItemIndex
    lvData.SortOrder = lvwAscending
End If
End Sub

Sub grid()
    lvData.ColumnHeaders.Item(1).Text = "Kode"
    lvData.ColumnHeaders.Item(1).Width = "1000"
    lvData.ColumnHeaders.Item(2).Text = "Nama KK"
    lvData.ColumnHeaders.Item(2).Width = "4000"
    lvData.ColumnHeaders.Item(3).Text = "Nama Kepala KK"
    lvData.ColumnHeaders.Item(3).Width = "3000"
    lvData.ColumnHeaders.Item(4).Text = "No HP Kepala KK"
    lvData.ColumnHeaders.Item(4).Width = "3000"
End Sub

Sub Combo()
    On Error GoTo pesan
    Dim rsCb As New ADODB.Recordset
    Call connect
    Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_rt,','fl_nm_rt) as rt FROM tb_rt
    ORDER BY fl_kd_rt ASC")
    cmbRT.Clear
    Do While Not rsCb.EOF
        cmbRT.AddItem rsCb(0)
        rsCb.MoveNext
    Loop
    Exit Sub
pesan:
```

(lanjutan)

```
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"  
End Sub
```

```
Sub Clear()  
txtKode = ""  
txtNmKK = ""  
txtNmKepalaKK = ""  
txtHpKK = ""  
txtKode.SetFocus  
End Sub
```

7. Program Entry Data Individu

```
Option Explicit  
Dim conn As New ADODB.Connection  
Sub connect()  
Set conn = New ADODB.Connection  
conn.Open "Dsn=ADI"  
conn.CursorLocation = adUseClient  
End Sub  
  
Private Sub cmbKK_Click()  
Call tampil("COMBO")  
End Sub  
  
Private Sub cmbKK_GotFocus()  
Call Combo  
End Sub  
  
Private Sub cmdDelete_Click()  
On Error GoTo pesan  
Dim rsCek As New ADODB.Recordset  
Dim kdKunci As String  
kdKunci = Left(cmbKK.Text, 8) & txtKode.Text  
If txtKode.Text = "" Then  
MsgBox "Maaf Kode Individu Harus Dimasukkan", vbInformation + vbOKOnly,
```

(lanjutan)

```
"information"  
    txtKode.SetFocus  
    Exit Sub  
  
Else  
    Call connect  
    Set rsCek = conn.Execute("SELECT*FROM tb_individu WHERE fl_kd_individu=" &  
(kdKunci) & """)  
    If rsCek.EOF Then  
        MsgBox "Maaf Kode Individu Tidak Ada di Dalam Database", vbInformation +  
vbOKOnly, "information"  
    Else  
        conn.Execute ("DELETE FROM tb_individu WHERE fl_kd_individu =" & (kdKunci)  
& """)  
        Call tampil("LOAD")  
        MsgBox "Data Individu Sukses di DELETE", vbInformation + vbOKOnly, "Information"  
    End If  
  
End If  
Exit Sub  
  
pesan:  
    MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"  
End Sub  
  
Private Sub cmdEdit_Click()  
    On Error GoTo pesan  
    Dim rsCek As New ADODB.Recordset  
    Dim kdKunci As String  
    kdKunci = Left(cmbKK.Text, 8) & txtKode.Text  
  
    If cmbKK.Text = "" Then  
        MsgBox "Maaf Kode KK Harus Anda Masukkan", vbInformation + vbOKOnly,  
"information"  
        cmbKK.SetFocus  
        Exit Sub  
    End If  
  
    If cmbKK.Text = "--pilih--" Then
```

(lanjutan)

```
MsgBox "Maaf Kode KK Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
cmbKK.SetFocus
Exit Sub
End If

If txtKode.Text = "" Then
MsgBox "Maaf Kode Individu Harus Anda Masukkan", vbInformation + vbOKOnly,
"information"
txtKode.SetFocus
Exit Sub
Else
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_individu WHERE fl_kd_individu=" &
(kdKunci) & "")
If rsCek.EOF Then
MsgBox "Maaf Kode Individu yang Anda Masukkan Tidak Ada di Dalam Database",
vbInformation + vbOKOnly, "information"
Else
If txtNoKTP = "" Then
MsgBox "Masukkan Nomor KTP", vbInformation + vbOKOnly, "information"
txtNoKTP.SetFocus
Exit Sub
End If

If txtNmIndividu = "" Then
MsgBox "Masukkan Nama Individu", vbInformation + vbOKOnly, "information"
txtNmIndividu.SetFocus
Exit Sub
End If

If txtHpIndividu = "" Then
MsgBox "Masukkan Nomor HP Individu", vbInformation + vbOKOnly, "information"
txtHpIndividu.SetFocus
Exit Sub
Else

MsgBox "Data Complete ", vbInformation + vbOKOnly, "information"
```

(lanjutan)

```
conn.Execute ("UPDATE tb_individu SET fl_kd_kk =" & (Left(cmbKK, 8)) &
",fl_no_ktp=" & (txtNoKTP.Text) & ", fl_nama =" & (txtNmIndividu.Text) & ", fl_no_hp
=" & (txtHpIndividu.Text) & ",fl_jenis_kelamin=" & (cmbJenisKelamin.Text) &
",fl_agama=" & (txtAgama.Text) & " WHERE fl_kd_individu =" & (kdKunci) & """)
Call tampil("LOAD")
MsgBox "Data success diupdate", vbInformation + vbOKOnly, "information"
End If
End If
End If
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End Sub

Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdFind_Click()
Call Clear
If txtSearch.Text = "" Then
MsgBox "Pilih Keterangan yang Anda Ingin Tampilkan", vbInformation + vbOKOnly,
"information"
txtSearch.SetFocus
Else
Call tampil("CARI")
End If
End Sub

Private Sub cmdRefresh_Click()
Call Clear
Call tampil("LOAD")
End Sub

Private Sub cmdSimpan_Click()
Dim rsCek As New ADODB.Recordset
Dim kdKunci As String
kdKunci = Left(cmbKK.Text, 8) & txtKode.Text
```


(lanjutan)

```
If cmbKK.Text = "" Then
    MsgBox "Masukkan Kode KK", vbInformation + vbOKOnly, "information"
    cmbKK.SetFocus
    Exit Sub
End If

If cmbKK.Text = "--pilih--" Then
    MsgBox "Masukkan Kode KK", vbInformation + vbOKOnly, "information"
    cmbKK.SetFocus
    Exit Sub
End If

If txtKode.Text = "" Then
    MsgBox "Masukkan Kode Individu", vbInformation + vbOKOnly, "information"
    txtKode.SetFocus
    Exit Sub
Else
    Call connect
    Set rsCek = conn.Execute("SELECT*FROM tb_individu WHERE fl_kd_individu = " &
(kdKunci) & "")
    If Not rsCek.EOF Then
        MsgBox "Data Individu Sudah Ada Di Dalam Database", vbInformation + vbOKOnly,
"information"
    End If
End If

If txtNoKTP.Text = "" Then
    MsgBox "Masukkan Nomor KTP", vbInformation + vbOKOnly, "information"
    txtNoKTP.SetFocus
    Exit Sub
End If

If txtNmIndividu.Text = "" Then
    MsgBox "Masukkan Nama Individu", vbInformation + vbOKOnly, "information"
    txtNmIndividu.SetFocus
    Exit Sub
End If
```

(lanjutan)

```
If txtHpIndividu.Text = "" Then
    MsgBox "Masukkan No HP Individu", vbInformation + vbOKOnly, "information"
    txtHpIndividu.SetFocus
Exit Sub
Else
    Call connect
    conn.Execute "INSERT INTO tb_individu(
fl_kd_individu,fl_no_ktp,fl_nama,fl_no_hp,fl_jenis_kelamin,fl_agama) " & "VALUES(" &
(kdKunci) & "," & (txtNoKTP.Text) & "," & (txtNmIndividu.Text) & "," &
(txtHpIndividu.Text) & "," & (cmbJenisKelamin.Text) & "," & (txtAgama.Text) & ")"
    Call tampil("LOAD")
    MsgBox "Data Sukses diInput", vbInformation + vbOKOnly, "information"
End If
End Sub

Sub tampil(key As String)
    On Error GoTo pesan
    Dim rsData As New ADODB.Recordset
    Dim fieldKey As String
    Dim istview
    Dim kdKunci As String
    kdKunci = Left(cmbKK.Text, 8) & txtKode.Text

    Call connect
    If key = "LOAD" Then
        Set rsData = conn.Execute("SELECT*FROM tb_individu ORDER BY fl_kd_individu
ASC")
    Else

    If key = "COMBO" Then
        Set rsData = conn.Execute("SELECT*FROM tb_individu WHERE fl_kd_individu = " &
(kdKunci) & " ORDER BY fl_kd_individu ASC")
    Else
        Select Case cmbSearch.ListIndex
        Case 0
            fieldKey = "fl_kd_individu"
        Case 1
            fieldKey = "fl_no_ktp"
        Case 2
```

(lanjutan)

```
        fieldKey = "fl_nama"
    Case 3
        fieldKey = "fl_no_hp"

    End Select
    Set rsData = conn.Execute("SELECT*FROM tb_individu WHERE " & fieldKey & " Like
'%" & (txtSearch.Text) & "%' ORDER BY fl_kd_individu Asc ")
    End If
End If

Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
    Set istview = lvData.ListItems.Add(, , rsData(0))
        istview.ListSubItems.Add , , rsData(2)
        istview.ListSubItems.Add , , rsData(3)
        istview.ListSubItems.Add , , rsData(4)
        istview.ListSubItems.Add , , rsData(5)
        istview.ListSubItems.Add , , rsData(6)
    rsData.MoveNext
Loop
Exit Sub
pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSCComctlLib.ColumnHeader)
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
    If lvData.SortOrder = lvwAscending Then
        lvData.SortOrder = lvwDescending
    Else
        lvData.SortOrder = lvwAscending
    End If
Else
    lvData.Sorted = True
    lvData.SortKey = ColumnHeader.SubItemIndex
    lvData.SortOrder = lvwAscending
End If
```

(lanjutan)

End If

End Sub

Sub grid()

lvData.ColumnHeaders.Item(1).Text = "Kode"

lvData.ColumnHeaders.Item(1).Width = "2000"

lvData.ColumnHeaders.Item(2).Text = "Nomor KTP"

lvData.ColumnHeaders.Item(2).Width = "2000"

lvData.ColumnHeaders.Item(3).Text = "Nama Individu"

lvData.ColumnHeaders.Item(3).Width = "3000"

lvData.ColumnHeaders.Item(4).Text = "No HP Individu"

lvData.ColumnHeaders.Item(4).Width = "3000"

lvData.ColumnHeaders.Item(5).Text = "Jenis Kelamin"

lvData.ColumnHeaders.Item(5).Width = "3000"

lvData.ColumnHeaders.Item(6).Text = "Agama"

lvData.ColumnHeaders.Item(6).Width = "3000"

End Sub

Sub Combo()

On Error GoTo pesan

Dim rsCb As New ADODB.Recordset

Call connect

Set rsCb = conn.Execute("SELECT CONCAT(fl_kd_kk,':',fl_nm_kk) as kk FROM tb_kk
ORDER BY fl_kd_kk ASC")

cmbKK.Clear

Do While Not rsCb.EOF

 cmbKK.AddItem rsCb(0)

rsCb.MoveNext

Loop

Exit Sub

pesan:

MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End Sub

Sub Clear()

txtKode = ""

txtNoKTP = ""

txtNmIndividu = ""

txtHpIndividu = ""

(lanjutan)

```
cmbJenisKelamin = ""  
txtAgama = ""  
txtKode.SetFocus  
End Sub
```

8. Program Send To Many

```
Option Explicit  
Dim conn As New ADODB.Connection  
Private strBuffer As String 'receive buffer  
Sub connect()  
Set conn = New ADODB.Connection  
conn.Open "Dsn=ADI"  
conn.CursorLocation = adUseClient  
End Sub  
  
Private Sub cmdDelete_Click()  
Dim rsCek As New ADODB.Recordset  
If txtID.Text = "" Then  
MsgBox "Masukkan Nomor ID Hendak DiHapus", vbOKOnly + vbInformation,  
"Information"  
txtID.SetFocus  
Exit Sub  
Else  
Call connect  
Set rsCek = conn.Execute("SELECT*FROM tb_entry_dbd WHERE fl_id = " &  
(txtID.Text) & """)  
If rsCek.EOF Then  
MsgBox "Data Individu Tidak DiTemukan Dalam Database", vbInformation +  
vbOKOnly, "information"  
Else  
conn.Execute ("DELETE FROM tb_entry_dbd WHERE fl_id=" & (txtID.Text) & """)  
Call tampil("Refresh")  
MsgBox "Data Sukses Di DiHapus", vbOKOnly + vbInformation, "Information"  
End If  
End If  
End Sub  
  
Private Sub cmdEnd_Click()  
Unload Me
```

(lanjutan)

End Sub

Private Sub cmdExit_Click()

Unload Me

End Sub

Private Sub cmdOK_Click()

On Error GoTo pesan

Dim rsHP As New ADODB.Recordset

Dim istview

If txtID.Text = "" Then

MsgBox "Masukkan Kode Kode Individu(ID)", vbInformation + vbOKOnly, "Information"

txtID.SetFocus

Exit Sub

Else

Call connect

Set rsHP = conn.Execute("SELECT i.fl_kd_individu,i.fl_nama,CONCAT(k.fl_nm_kk,'t.fl_nm_rt','w.fl_nm_rw','l.fl_nm_kelurahan) FROM tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l WHERE i.fl_kd_individu Like '%" & (txtID.Text) & "%' AND k.fl_kd_kk Like '%" & Left(txtID, 8) & "%' AND t.fl_kd_rt Like '%" & Left(txtID, 6) & "%'AND w.fl_kd_rw Like '%" & Left(txtID, 4) & "%'AND l.fl_kd_kelurahan Like '%" & Left(txtID, 2) & "%")")

Call grid

lvData.ListItems.Clear

Do While Not rsHP.EOF

Set istview = lvData.ListItems.Add(, , rsHP(0))

istview.ListSubItems.Add , , rsHP(1)

istview.ListSubItems.Add , , rsHP(2)

rsHP.MoveNext

Loop

Exit Sub

pesan:

MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"

End If

End Sub

Private Sub cmdRefresh_Click()

(lanjutan)

```
Call tampil("Refresh")
```

```
End Sub
```

```
Private Sub cmdSave_Click()
```

```
Dim rsHP As New ADODB.Recordset
```

```
Dim ID, Nama, Alamat
```

```
Call connect
```

```
Set rsHP = conn.Execute("SELECT i.fl_kd_individu,i.fl_nama,CONCAT(k.fl_nm_kk,"t.fl_nm_rt","w.fl_nm_rw","l.fl_nm_kelurahan) FROM tb_individu i,tb_kk k,tb_rt t,tb_rw w,tb_kelurahan l WHERE i.fl_kd_individu Like '%" & (txtID.Text) & "%' AND k.fl_kd_kk Like '%" & Left(txtID, 8) & "%' AND t.fl_kd_rt Like '%" & Left(txtID, 6) & "%'AND w.fl_kd_rw Like '%" & Left(txtID, 4) & "%'AND l.fl_kd_kelurahan Like '%" & Left(txtID, 2) & "%")")
```

```
Call grid
```

```
lvData.ListItems.Clear
```

```
If Not rsHP.EOF Then
```

```
    ID = "" & rsHP(0)
```

```
    Nama = "" & rsHP(1)
```

```
    Alamat = "" & rsHP(2)
```

```
    conn.Execute ("INSERT INTO tb_entry_dbd VALUES('" & ID & "','" & Nama & "','" & Alamat & "')")
```

```
    rsHP.MoveNext
```

```
    Call tampil("Refresh")
```

```
    MsgBox "Data Sukses DiSimpan", vbOKOnly + vbInformation, "Information"
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
Private Sub cmdSMS_Click()
```

```
Dim rsCek As New ADODB.Recordset
```

```
Call connect
```

```
Set rsCek = conn.Execute("SELECT*FROM tb_entry_dbd WHERE fl_id = '" & (txtID.Text) & "'")
```

```
txtMessage.Text = "Kasus DBD : " + " " + "" & rsCek(1) + " " + "" & rsCek(2) + " , " + "Segera Penanggulangan"
```

(lanjutan)

```
Set rsCek = conn.Execute("SELECT*FROM tb_kelurahan WHERE fl_kd_kelurahan = " &  
Left(txtID, 2) & """)
```

```
lbl1.Caption = "Lurah" + " " & rsCek(2)
```

```
txt1.Text = "" & rsCek(4)
```

```
Set rsCek = conn.Execute("SELECT*FROM tb_rw WHERE fl_kd_rw = " & Left(txtID, 4) &  
""")
```

```
lbl2.Caption = "" & rsCek(2)
```

```
txt2.Text = "" & rsCek(4)
```

```
Set rsCek = conn.Execute("SELECT*FROM tb_rt WHERE fl_kd_rt = " & Left(txtID, 6) &  
""")
```

```
lbl3.Caption = "" & rsCek(2)
```

```
txt3.Text = "" & rsCek(4)
```

```
Set rsCek = conn.Execute("SELECT*FROM tb_kk WHERE fl_kd_kk = " & Left(txtID, 8) &  
""")
```

```
lbl4.Caption = "" & rsCek(3)
```

```
txt4.Text = "" & rsCek(4)
```

```
End Sub
```

```
Private Sub Find_Click()
```

```
Dim rsCek As New ADODB.Recordset
```

```
If cmbSearch.Text = "" Or cmbSearch.Text = "--Pilih--" Then
```

```
MsgBox "Pilih Keterangan Yang Hendak Anda Temukan", vbOKOnly + vbInformation,  
"Information"
```

```
cmbSearch.SetFocus
```

```
Exit Sub
```

```
End If
```

```
If txtSearch.Text = "" Then
```

```
MsgBox "Masukkan Inisial Keterangan Yang Hendak Anda Temukan", vbOKOnly +  
vbInformation, "Information"
```

```
txtSearch.SetFocus
```

```
Exit Sub
```

```
Else
```


(lanjutan)

```
Call connect
Set rsCek = conn.Execute("SELECT*FROM tb_entry_dbd WHERE fl_id = " &
(txtID.Text) & """)
If rsCek.EOF Then
MsgBox "Data Individu Tidak diTemukan Di Dalam Database", vbInformation +
vbOKOnly, "information"
Else
Call tampil("Find")
End If
End If
End Sub
Sub tampil(key As String)
On Error GoTo pesan
Dim rsData As New ADODB.Recordset
Dim fieldKey As String
Dim istview

Call connect
If key = "Refresh" Then
Set rsData = conn.Execute("SELECT*FROM tb_entry_dbd ORDER BY fl_id ASC")
Else
Select Case cmbSearch.ListIndex
Case 0
fieldKey = "fl_id"
Case 1
fieldKey = "fl_nama"
Case 2
fieldKey = "fl_alamat"

End Select
Call connect
Set rsData = conn.Execute("SELECT*FROM tb_entry_dbd WHERE " & fieldKey & "
Like '%" & (txtSearch.Text) & "%' ORDER BY fl_id Asc ")
End If

Call grid
lvData.ListItems.Clear
Do While Not rsData.EOF
Set istview = lvData.ListItems.Add(, , rsData(0))
```

(lanjutan)

```
        istview.ListSubItems.Add , , rsData(1)
        istview.ListSubItems.Add , , rsData(2)
    rsData.MoveNext
Loop
Exit Sub

pesan:
MsgBox Err.Description, vbCritical + vbOKOnly, "Pesan Error"
End Sub

Private Sub lvData_ColumnClick(ByVal ColumnHeader As MSCComctlLib.ColumnHeader)
Dim colVar As ColumnHeader
If lvData.Sorted = True And ColumnHeader.SubItemIndex = lvData.SortKey Then
    If lvData.SortOrder = lvwAscending Then
        lvData.SortOrder = lvwDescending
    Else
        lvData.SortOrder = lvwAscending
    End If
Else
    lvData.Sorted = True
    lvData.SortKey = ColumnHeader.SubItemIndex
    lvData.SortOrder = lvwAscending
End If
End Sub

Sub grid()
    lvData.ColumnHeaders.Item(1).Text = "Nomor ID"
    lvData.ColumnHeaders.Item(1).Width = "2000"
    lvData.ColumnHeaders.Item(2).Text = "Nama"
    lvData.ColumnHeaders.Item(2).Width = "2000"
    lvData.ColumnHeaders.Item(3).Text = "Alamat"
    lvData.ColumnHeaders.Item(3).Width = "5000"
    lvData.ColumnHeaders.Item(4).Text = "Status"
    lvData.ColumnHeaders.Item(4).Width = "3000"
End Sub
```

(lanjutan)

```
Private Sub MSComm1_OnComm()  
'SMS SEND TO MANY "  
Dim strMessage As String  
Select Case MSComm1.CommEvent  
' Event messages.  
Case comEvReceive  
    strMessage = StrConv(MSComm1.Input, vbUnicode)  
' Case comEvSend  
' Case comEvCTS  
    strMessage = "[Change in CTS Detected]"  
' Case comEvDSR  
    strMessage = "[Change in DSR Detected]"  
' Case comEvCD  
    strMessage = "[Change in CD Detected]"  
' Case comEvRing  
    strMessage = "[The Phone is Ringing]"  
' Case comEvEOF  
    strMessage = "[End of File Detected]"  
  
' Error messages.  
Case comBreak  
    strMessage = "[Break Received]"  
Case comCDTO  
    strMessage = "[Carrier Detect Timeout]"  
Case comCTSTO  
    strMessage = "[CTS Timeout]"  
Case comDCB  
    strMessage = "[Error retrieving DCB]"  
Case comDSRTO  
    strMessage = "[DSR Timeout]"  
Case comFrame  
    strMessage = "[Framing Error]"  
Case comOverrun  
    strMessage = "[Overrun Error]"  
Case comRxOver  
    strMessage = "[Receive Buffer Overflow]"  
Case comRxParity  
    strMessage = "[Parity Error]"  
Case comTxFull
```

(lanjutan)

```
        strMessage = "[Transmit Buffer Full]"
    ' Case Else
    '     strMessage = "[Unknown error or event: " & MSComm1.CommEvent & "]"
    End Select
    strBuffer = strBuffer & strMessage
End Sub

Private Sub Delay(ByVal HowLong As Date)
    Dim endDate As Date
    endDate = DateAdd("s", HowLong, Now)
    While endDate > Now
        DoEvents 'Allows windows to handle other stuff
    Wend
End Sub

Private Sub cmdSend_Click()
    Dim x As Integer
    Dim i
    If x = 1 Then
        txtHP.Text = txt1.Text
    ElseIf x = 2 Then
        txtHP.Text = txt2.Text
    ElseIf x = 3 Then
        txtHP.Text = txt3.Text
    ElseIf x = 4 Then
        txtHP.Text = txt4.Text
    End If

    For x = 1 To 4
        If x = 1 Then
            txtHP.Text = txt1.Text
        ElseIf x = 2 Then
            txtHP.Text = txt2.Text
        ElseIf x = 3 Then
            txtHP.Text = txt3.Text
        ElseIf x = 4 Then
            txtHP.Text = txt4.Text
        End If
    End For
End Sub
```

(lanjutan)

```
i = Len(Hex((Len(ASCII2PDU(txtMessage.Text)) / 2)))
If i Mod 2 = 1 Then
    txtPDU.Text = SMSC(cbosmsc) & ("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") & (Hex((Len(ASCII2PDU(txtMessage.Text))
/ 2))) & ASCII2PDU(txtMessage.Text)
    txtPanjangPDU = Val(Len(("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB0") & (Hex((Len(ASCII2PDU(txtMessage.Text))
/ 2))) & ASCII2PDU(txtMessage.Text)) / 2)

Else
    txtPDU.Text = SMSC(cbosmsc) & ("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB") & (Hex((Len(ASCII2PDU(txtMessage.Text)) /
2))) & ASCII2PDU(txtMessage.Text)
    txtPanjangPDU = Val(Len(("11000") & (Hex(Len(txtHP.Text))) & ("91") &
Reverse(InputHP(txtHP.Text)) & ("0000AB") & (Hex((Len(ASCII2PDU(txtMessage.Text)) /
2))) & ASCII2PDU(txtMessage.Text)) / 2)
End If
txtLog.Text = ""

txtHP.Enabled = False
txtMessage.Enabled = False
cmdSend.Enabled = False
cmdEnd.Enabled = False

On Error GoTo ErrorHandler
With MSCComm1
    .CommPort = 6
    .Settings = "19200,N,8,1"
    .Handshaking = comRTS
    .RTSEnable = True
    .DTREnable = True
    .RThreshold = 1
    .SThreshold = 1
    .InputMode = comInputModeBinary
    .InputLen = 0
    .PortOpen = True 'must be the last
End With
```

(lanjutan)

```
'Call Send(Chr(27)) 'Cancel active message input
```

```
'Verify connection
```

```
Call Send("AT" & vbCrLf)
```

```
If Not VerifySuccess(Receive, True) Then GoTo finally
```

```
'Enter text mode (phone must support it)
```

```
Call Send("AT+CMGF=0" & vbCrLf)
```

```
If Not VerifySuccess(Receive, True) Then GoTo finally
```

```
'Send the message
```

```
Dim buf As String
```

```
Call Send("AT+CMGS=" & txtPanjangPDU.Text & vbCrLf)
```

```
If Not VerifyEndsWith(Receive, "> ", True) Then GoTo finally
```

```
Call Send(txtPDU.Text & Chr(26))
```

```
If WaitForSuccess(buf) Then
```

```
    lbl7.BackColor = &H80000001
```

```
Else
```

```
    lbl8.BackColor = &H80000008
```

```
End If
```

```
GoTo finally
```

```
ErrorHandler:
```

```
Call MsgBox("Error: " & Err.Number & " - " & Err.Description, vbCritical + vbOKOnly)
```

```
finally:
```

```
If MSComm1.PortOpen Then MSComm1.PortOpen = False
```

```
txtHP.Enabled = True
```

```
txtMessage.Enabled = True
```

```
cmdSend.Enabled = True
```

```
cmdEnd.Enabled = True
```

```
Next x
```

```
End Sub
```

```
'Check for success
```

```
Private Function IsSuccess(ByVal data As String)
```

```
    IsSuccess = InStr(data, vbCrLf & "OK" & vbCrLf) > 0
```

(lanjutan)

End Function

'Check for comm error

Private Function IsCommError(ByVal data As String)

IsCommError = InStr(data, vbCrLf & "ERROR" & vbCrLf) > 0

End Function

'Check for network error

Private Function IsNetworkError(ByVal data As String)

IsNetworkError = InStr(data, vbCrLf & "+CMS ERROR:") > 0

End Function

'Check for any known error

Private Function IsError(ByVal data As String)

IsError = IsCommError(data) Or IsNetworkError(data)

End Function

'Call this function when response is not the expected one.

'It analyzes the response and displays an appropriate error message if showMessage is True.

Private Sub ErrorDetails(ByVal data As String, Optional ByVal showMessage As Boolean = False)

Dim msg As String

'Check if there is data at all

If Len(data) = 0 Then

msg = "No answer from phone"

Else

'Check if comm error

If IsCommError(data) Then

msg = "Phone returned an error."

Else

'Check if network error

If IsNetworkError(data) Then

msg = "A network error occurred."

Else

'Anything else: Unexpected

msg = "Unexpected response: "" & data & """

End If

End If

(lanjutan)

```
End If
Call Trace(msg)
If showMessage Then Call MsgBox(msg, vbCritical + vbOKOnly)
End Sub

'Ensures that the string contains a success message,
'If not, determines the error details.
Private Function VerifySuccess(ByVal data As String, Optional ByVal showMessage As
Boolean = False) As Boolean
    VerifySuccess = True
    If Not IsSuccess(data) Then
        VerifySuccess = False
        Call ErrorDetails(data, showMessage)
    End If
End Function

'Ensures that the string ends with a specific string.
'If not, determines the error details.
Private Function VerifyEndsWith(ByVal data As String, ByVal endsWith As String, Optional
ByVal showMessage As Boolean = False) As Boolean
    VerifyEndsWith = True
    If InStr(data, endsWith) <> (Len(data) - Len(endsWith) + 1) Then
        VerifyEndsWith = False
        Call ErrorDetails(data, showMessage)
    End If
End Function

'Sends data to the serial port
Private Sub Send(ByVal data As String)
    strBuffer = ""
    Call Trace("<<" & data)
    MSComm1.Output = data
End Sub

'Receives data from the serial port
Private Function Receive() As String
    Dim strPart As String
    Dim strInput As String
    strInput = ""
```


(lanjutan)

```
Do
    strPart = ""
    Call Delay(1)
    strPart = strBuffer
    strBuffer = ""
    If strPart = "" Then Exit Do
    strInput = strInput & strPart
Loop
If strInput <> "" Then Call Trace(">> " & strInput)
Receive = strInput
End Function

'Waits until a success message is received or a timeout occurred.
Private Function WaitForSuccess(ByRef data As String) As Boolean
    Dim i As Integer
    Dim strInput As String
    Dim strPart As String
    strInput = ""

    'try receive 5 times with 2 seconds delay between
    For i = 1 To 5
        strPart = Receive
        strInput = strInput & strPart
        If IsSuccess(strInput) Or IsError(strInput) Then Exit For
        If strPart = "" Then
            Call Trace("Nothing new received, waiting 2s...")
            Call Delay(2)
        End If
    Next
    data = strInput
    WaitForSuccess = IsSuccess(strInput)
End Function

Private Sub Trace(ByVal message As String)
    Dim strLine As String
    strLine = DateTime.Now & " " & message
    txtLog.Text = txtLog.Text & strLine & vbCrLf
    txtLog.SelStart = Len(txtLog.Text)
End Sub
```

(lanjutan)

Form Setting ComPort

```
Option Explicit
Private Sub Form_Load()
Dim i As Integer
cboPort.Clear
For i = 1 To 10
    cboPort.AddItem "COM" & i
Next i
cbosmsc.Clear
cbosmsc.AddItem "Telkomsel+6281100000"
cbosmsc.AddItem "Telkomsel+62811130004"
cbosmsc.AddItem "Satelindo+6281615"
cbosmsc.AddItem "Excelcom+62818445009"
cbosmsc.AddItem "IM3+62855000000"
cboPort.ListIndex = 5
cbosmsc.ListIndex = 0
End Sub
```

Module Send To Many

```
Option Explicit
'Fungsi ini digunakan untuk melakukan settingan no smsc sehingga dapat dikonvert ke PDU'
Function SMSC(cbosmsc As ComboBox) As String
If cbosmsc = "Telkomsel+6281100000" Then
    SMSC = "07912618010000"
End If
If cbosmsc = "Telkomsel+62811130004" Then
    SMSC = "07912618110300F4"
End If
If cbosmsc = "Satelindo+6281615" Then
    SMSC = "0791261816F5"
End If
If cbosmsc = "Excelcom+62818445009" Then
    SMSC = "07912618485400F9"
End If
If cbosmsc = "IM3+62855000000" Then
    SMSC = "07912658050000F0"
End If
End Function
' Fungsi ini digunakan untuk menginput no HP'
```

(lanjutan)

```
Function InputHP(ByVal NumHP As String) As String
```

```
If Len(NumHP) Mod 2 = 1 Then
```

```
    InputHP = NumHP + "F"
```

```
Else
```

```
    InputHP = NumHP
```

```
End If
```

```
End Function
```

'Fungsi ini digunakan untuk melakukan reverse no HP sehingga memenuhi format PDU'

```
Function Reverse(ByVal strChar As String) As String
```

```
Dim i As Integer
```

```
Dim position() As String
```

```
Dim intLength As Integer
```

```
intLength = Len(strChar)
```

```
ReDim position(intLength + 1)
```

```
For i = 1 To (intLength / 2)
```

```
    position(i) = Mid(strChar, i * 2, 1)
```

```
    position(i + 1) = Mid(strChar, i * 2 - 1, 1)
```

```
    Reverse = Reverse & position(i) & position(i + 1)
```

```
Next i
```

```
End Function
```

'Fungsi ini digunakan untuk mengubah Format text mode pada Message menjadi Format PDU'

```
Function ASCII2PDU(ByVal strASCII As String) As String
```

```
Dim i
```

```
Dim intLength As Integer
```

```
Dim slices() As String
```

```
Dim quart() As String
```

```
Dim result() As String
```

```
Dim temp As String
```

```
intLength = Len(strASCII)
```

```
ReDim slices(intLength + 1)
```

```
ReDim quart(intLength + 1, 3)
```

```
ReDim result(intLength + 1, 3)
```

```
temp = String$(intLength, "0")
```

```
For i = 1 To intLength
```

```
    temp = temp & BINARY7(ASCICODE(Mid(strASCII, intLength - i + 1, 1)))
```

```
Next i
```

```
For i = intLength To 1 Step -1
```

```
    slices(i) = Mid(temp, (8 * (i - 1)) + 1, 8)
```

(lanjutan)

```
quart(i, 1) = Mid(slices(i), 1, 4)
quart(i, 2) = Mid(slices(i), 5, 4)
result(i, 1) = Hex$(DEBINARY4(quart(i, 1)))
result(i, 2) = Hex$(DEBINARY4(quart(i, 2)))
ASCII2PDU = ASCII2PDU & result(i, 1) & result(i, 2)
Next i
```

End Function

```
Function BINARY7(ByVal Number As Integer) As String
```

```
    If Number > 1 Then
```

```
        Dim i As Integer
```

```
        Dim result(8) As Integer
```

```
        Dim SISA(8) As Integer
```

```
        i = 1
```

```
        Do
```

```
            result(i) = Int(Number / 2)
```

```
            SISA(i) = Number Mod 2
```

```
            Number = result(i)
```

```
            i = i + 1
```

```
            BINARY7 = SISA(i - 1) & BINARY7
```

```
        Loop Until result(i - 1) < 2
```

```
        BINARY7 = result(i - 1) & BINARY7
```

```
        BINARY7 = String$(7 - Len(BINARY7), "0") & BINARY7
```

```
    ElseIf Number = 1 Then
```

```
        BINARY7 = "0000001"
```

```
    ElseIf Number = 0 Then
```

```
        BINARY7 = "0000000"
```

```
    End If
```

End Function

```
Function ASCIICODE(ByVal strChar As String) As Integer
```

```
    Dim intCodeNum As Integer
```

```
    intCodeNum = Asc(strChar)
```

```
    If intCodeNum = 64 Then
```

```
        ASCIICODE = 0
```

```
    ElseIf intCodeNum = 36 Then
```

```
        ASCIICODE = 2
```

```
    ElseIf intCodeNum = 223 Then
```

(lanjutan)

```
    ASCIICODE = 30
Elseif intCodeNum >= 32 And intCodeNum <= 35 Then
    ASCIICODE = intCodeNum
Elseif intCodeNum >= 37 And intCodeNum <= 63 Then
    ASCIICODE = intCodeNum
Elseif intCodeNum = 95 Then
    ASCIICODE = 64
Elseif intCodeNum >= 65 And intCodeNum <= 90 Then
    ASCIICODE = intCodeNum
Elseif intCodeNum = 196 Then
    ASCIICODE = 91
Elseif intCodeNum = 214 Then
    ASCIICODE = 92
Elseif intCodeNum = 220 Then
    ASCIICODE = 94
Elseif intCodeNum = 168 Then
    ASCIICODE = 96
Elseif intCodeNum >= 97 And intCodeNum <= 122 Then
    ASCIICODE = intCodeNum
Elseif intCodeNum = 228 Then
    ASCIICODE = 123
Elseif intCodeNum = 246 Then
    ASCIICODE = 124
Elseif intCodeNum = 252 Then
    ASCIICODE = 126
Elseif intCodeNum = 10 Then
    ASCIICODE = intCodeNum
Elseif intCodeNum = 13 Then
    ASCIICODE = intCodeNum
End If
End Function

Function BINARY4(ByVal Number As Integer) As String
    If Number > 1 Then
        Dim i As Integer
        Dim result(5) As Integer
        Dim SISA(5) As Integer

        i = 1
```

(lanjutan)

```
Do
    result(i) = Int(Number / 2)
    SISA(i) = Number Mod 2
    Number = result(i)
    i = i + 1
    BINARY4 = SISA(i - 1) & BINARY4
Loop Until result(i - 1) < 2
BINARY4 = result(i - 1) & BINARY4
BINARY4 = String$(4 - Len(BINARY4), "0") & BINARY4
Elseif Number = 1 Then
    BINARY4 = "0001"
Elseif Number = 0 Then
    BINARY4 = "0000"
End If
End Function
Function DEBINARY4(ByVal strChar As String) As Integer
    Dim Number(5) As Integer
    Dim i
    For i = 1 To 4
        Number(i) = Val(Mid(strChar, i, 1))
        DEBINARY4 = DEBINARY4 + Number(i) * 2 ^ (4 - i)
    Next i
End Function
```