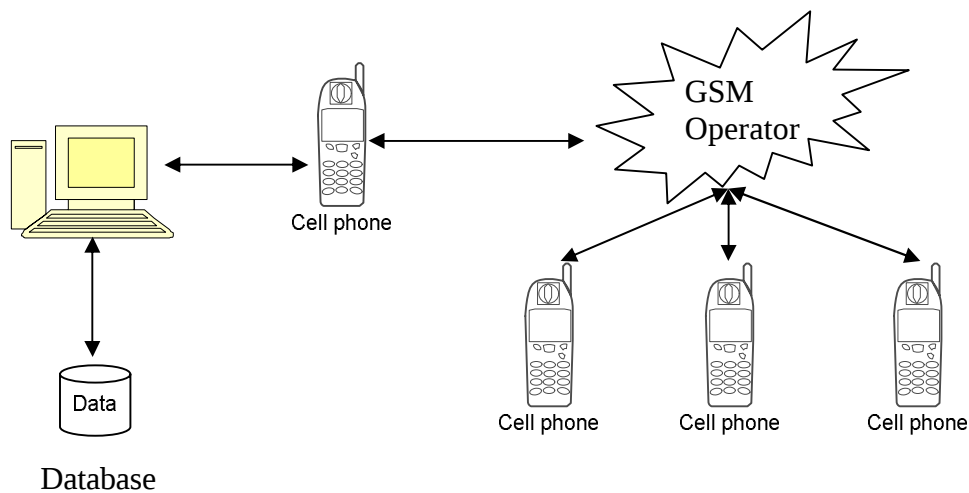


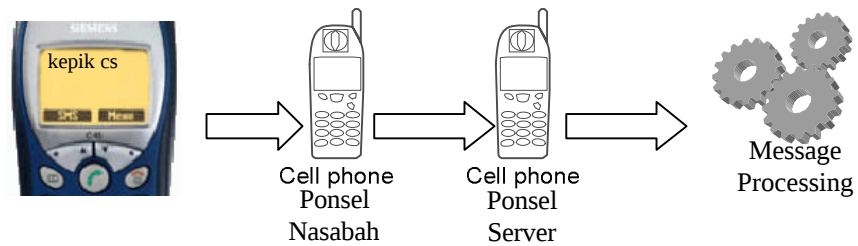
3. PERENCANAAN SISTEM

Pada bab ini akan dibahas mengenai perencanaan sistem. Pembahasan ini akan meliputi dua bagian besar, yaitu perencanaan perangkat lunak utama dan perencanaan perangkat lunak penunjang. Gambar berikut ini memperlihatkan *block diagram* dari “*Prototype Mobile Banking*” secara keseluruhan.



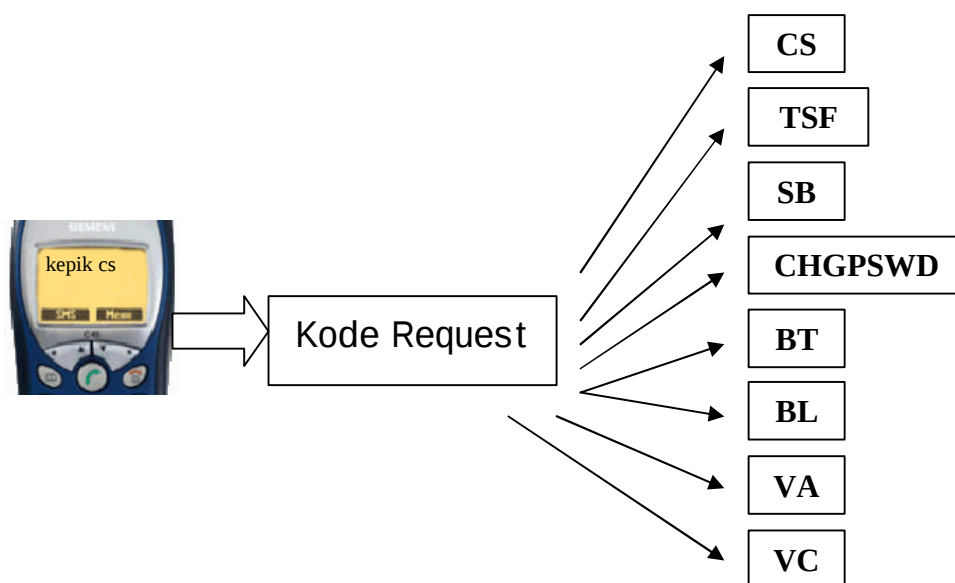
Gambar 3.1. *Block System*

Prosesnya adalah nasabah mengirim sebuah SMS yang berisi password dengan kode *request* yang diinginkan sesuai dengan format (lihat Bab 3.4.5). SMS tersebut kemudian dikirim lewat ponsel nasabah, yang mana nomor ponselnya sesuai dengan nomor ponsel pada saat pembukaan rekening. Kemudian SMS tersebut dikirim ke nomor ponsel server. Setelah sampai maka SMS tersebut akan di-*encode* untuk kemudian dibaca isi dari SMS tersebut kemudian melakukan proses sesuai dengan kode yang di-*request*. Setelah itu SMS balasan akan dikirim kembali ke ponsel nasabah. Prosesnya adalah sebagai berikut:



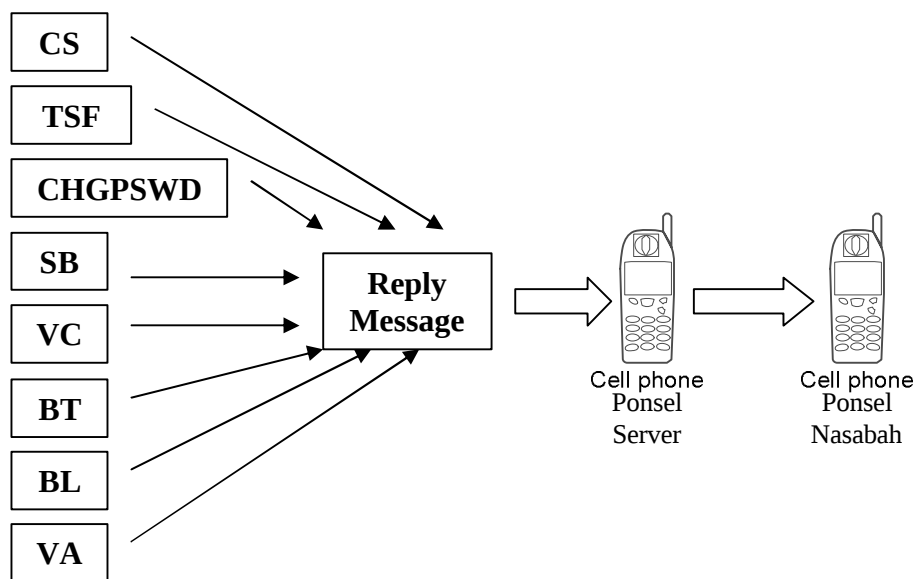
Gambar 3.2. Proses pengiriman *request* SMS

Setelah SMS yang dikirim oleh nasabah sampai ke ponsel server, maka sekarang kita sampai pada bagian *message processing*. Pada bagian ini SMS tersebut, yang masih dalam bentuk bilangan heksadesimal, akan diolah. Setelah isi SMS diketahui maka data tersebut akan diolah menjadi beberapa bagian, yaitu password dan kode *request*. Sesuai dengan kode request maka sistem akan memprosesnya sampai menghasilkan hasilnya. Untuk lebih jelasnya proses tersebut bisa dilihat pada gambar 3.3 berikut:



Gambar 3.3. Proses pengolahan kode *request*

Setelah kode *request* diproses, sistem akan menghasilkan SMS balasan sesuai dengan yang diminta. SMS balasan itu kemudian akan diproses untuk kembali dikirim ke ponsel nasabah yang tadi mengirim SMS *request*. Prosesnya adalah seperti pada gambar 3.4 berikut:



Gambar 3.4. Proses pengiriman *Reply* SMS

Setelah proses diatas, maka nasabah akan menerima *reply* SMS yang sesuai dengan yang diminta, sebagai contoh jika nasabah ingin mengecek saldo maka dia akan mengirim *password* beserta kode “CS”. Jika *password*, nomor ponsel, serta kodenya benar maka SMS balasan yang akan diterima adalah “Saldo anda Rp.....”. Jika format *request* SMS-nya salah maka nasabah akan menerima *reply* SMS yang menyatakan bahwa formatnya salah.

3.1. Perencanaan Perangkat Lunak Penunjang

Dalam perencanaan perangkat lunak ini perlu diketahui terlebih dahulu fasilitas-fasilitas dan data-data apa saja yang perlu ditampilkan agar operator dapat mengetahui informasi apa saja yang ada di sistem *mobile banking*. Fasilitas yang dibutuhkan oleh operator untuk memperoleh data dan informasi yang dibutuhkan adalah sebagai berikut:

a. Penjelasan mengenai menu New Account

Menu ini digunakan untuk mengisi data-data dari nasabah yang dibutuhkan untuk membuka rekening. Setelah proses memasukkan data selesai maka sistem secara otomatis akan mengirim *password* ke nomor ponsel yang telah dimasukkan sebelumnya pada menu New Account.

b. Penjelasan mengenai menu Edit Account

Menu ini digunakan untuk mengubah beberapa data dari nasabah. Tidak semua data bisa diubah karena bisa mengganggu sistem, seperti nomor rekening dan nama nasabah.

c. Penjelasan mengenai menu Account List

Menu ini digunakan untuk melihat daftar rekening. Dimana bisa dicari menurut nomor rekening, nama, kota, jenis kelamin, status perkawinan, agama, pekerjaan, pendidikan terakhir dan penghasilan.

d. Penjelasan mengenai menu SMS List

Menu ini bisa digunakan untuk melihat daftar SMS yang masuk dan keluar dari sistem. Semua SMS yang dikirim oleh nasabah dan balasannya juga bisa dilihat disini. Dimana SMS itu bisa dicari berdasarkan tanggal, nomor pengirim dan status pengiriman.

e. Penjelasan mengenai menu Transaction List

Menu ini digunakan untuk melihat semua transaksi yang dilakukan oleh para nasabah dalam jangka waktu tertentu. Semua transaksi itu bisa dicari berdasarkan tanggal dan nomor rekening.

f. Penjelasan mengenai menu Port Setting

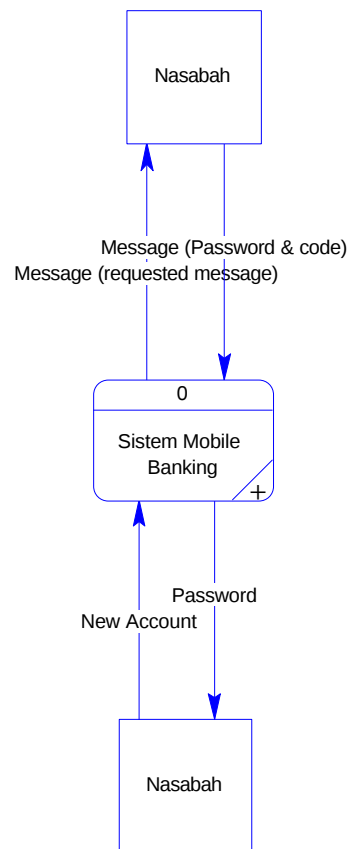
Menu ini digunakan untuk melakukan *setting port* yang tersambung dengan ponsel server dan jenis *SIM-card* yang digunakan oleh ponsel server. Selain itu juga bisa digunakan untuk mengecek kondisi *port* apakah sudah tersambung atau belum.

g. Penjelasan mengenai menu SMS Simulation

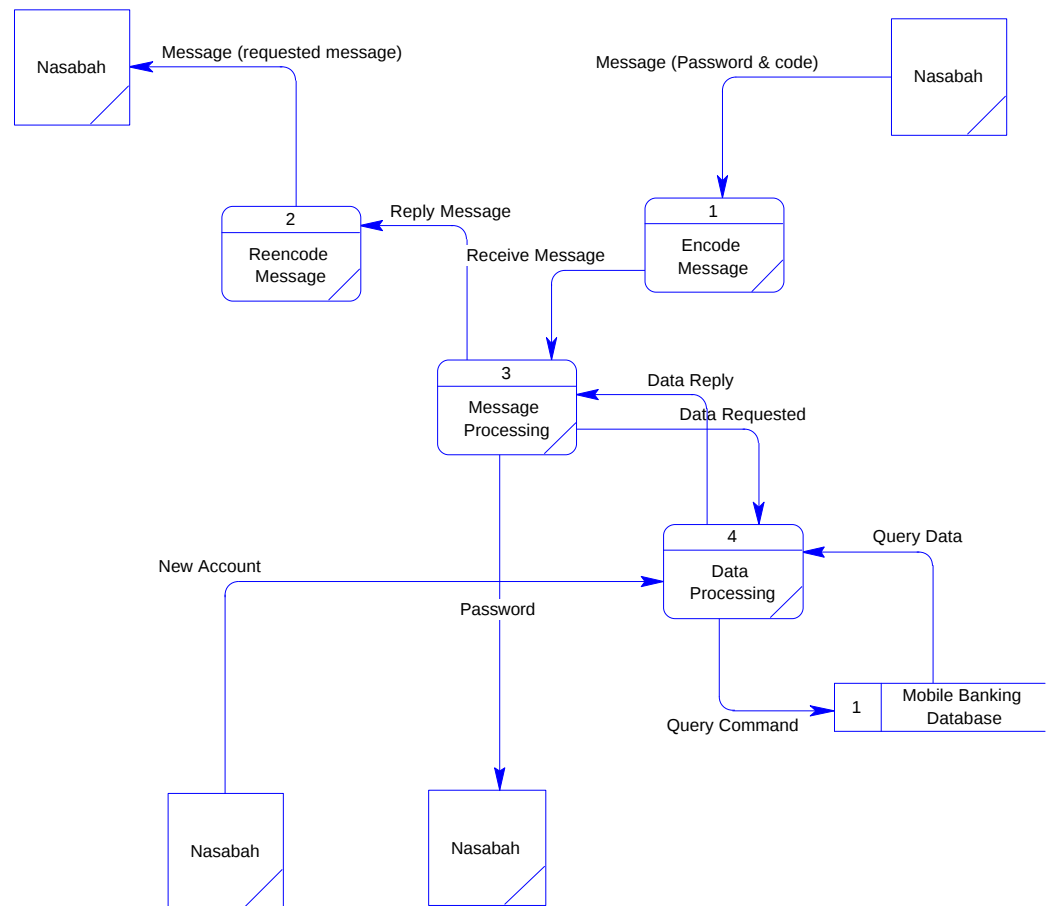
Menu ini digunakan untuk simulasi pengiriman dan penerimaan SMS. Menu ini lebih kepada untuk mengecek *message processing*. Semua SMS yang di-“kirim” lewat menu ini juga akan dimasukkan ke database.

3.2. Perencanaan Database

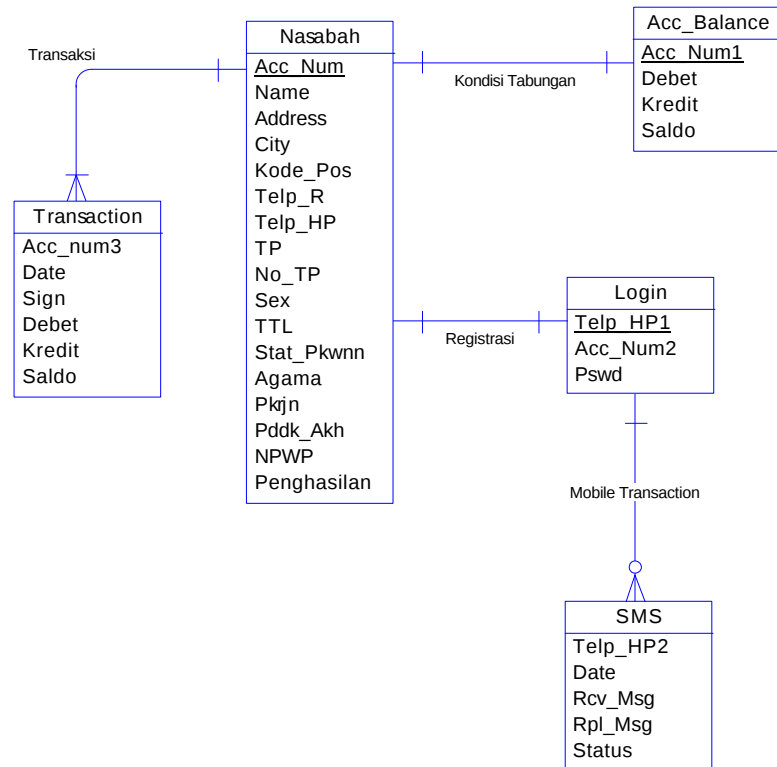
Untuk menyimpan data pada aplikasi *mobile banking* ini, maka diperlukan sebuah *database* dengan beberapa buah tabel didalamnya. Desain *database*-nya hanya merupakan rekayasa dari desain *database* pada sistem perbankan. Desain ini hanya menampilkan tabel-tabel yang dibutuhkan untuk sistem *mobile banking*. Program *database* yang digunakan adalah Microsoft Access 2002, dan nama *database* yang digunakan adalah MobileBanking.mdb. Berikut ini adalah rancangan DFD dan ERD-nya seperti pada gambar 3.5, 3.6 dan 3.7.



Gambar 3.5. DFD level 0



Gambar 3.6. DFD level 1



Gambar 3.7. ERD

Database MobileBanking.mdb terdiri dari tabel-tabel sebagai berikut:

a. Tabel **Account**

Tabel ini digunakan untuk mencatat semua data dari nasabah yang diminta pada saat para nasabah membuka rekening.

Tabel 3.1. Tabel Account

Field	Type	Size	Keterangan
Acc_Num	Text	15	Nomor rekening
Jns_Tab	Text	25	Jenis tabungan
Name	Text	50	Nama nasabah
Address	Text	75	Alamat nasabah
City	Text	50	Kota
Kode_Pos	Text	5	Kode pos
Telp_R	Text	15	Telepon rumah
Telp_HP	Text	15	Nomor ponsel
TP	Text	20	Jenis tanda pengenal
No_TP	Text	20	Nomor tanda pengenal
Sex	Text	10	Jenis kelamin
TTL	Text	50	Tempat tanggal lahir
Stat_Pkwnn	Text	20	Status perkawinan
Agama	Text	50	Agama
Pkrjn	Text	50	Pekerjaan
Pddk_Akh	Text	25	Pendidikan terakhir
NPWP	Text	20	Nomor pokok wajib pajak
Penghasilan	Text	50	Jumlah penghasilan

b. Tabel **Acc_Balance**

Tabel ini digunakan untuk mendata total transaksi dari setiap nasabah.

Tabel 3.2. Tabel Acc_Balance

Field	Type	Size	Keterangan
Acc_Num	Text	20	Nomor rekening
Debet	Number	18	Jumlah transaksi debet
Kredit	Number	18	Jumlah transaksi kredit
Saldo	Number	18	Jumlah saldo akhir

c. Tabel **Jenis_Tabungan**

Tabel ini digunakan untuk menyimpan semua jenis tabungan beserta suku bunganya.

Tabel 3.3. Tabel Jenis_Tabungan

Field	Type	Size	Keterangan
Jns_tab	Text	25	Jenis tabungan
SB	Text	5	Suku bunga

d. Tabel **Listrik**

Tabel ini digunakan untuk menyimpan tagihan listrik.

Tabel 3.4. Tabel Listrik

Field	Type	Size	Keterangan
No_lggn	Text	50	Nomor langganan
Tagihan	Text	50	Jumlah tagihannya
Rgtr	Yes/No	Yes/No	Register untuk otomatis debet
No_Rek	Text	15	Nomor rekening untuk auto debet
Bayar	Yes/No	Yes/No	Kondisi pembayaran

e. Tabel **Login**

Tabel ini digunakan untuk menyimpan segala informasi mengenai nasabah yang bisa menggunakan *mobile banking*.

Tabel 3.5. Tabel Login

Field	Type	Size	Keterangan
Pswd	Text	50	Password dari setiap nasabah
Acc_Num	Text	50	Nomor rekening
Phone_Num	Text	15	Nomor ponsel

f. Tabel **SMS**

Tabel ini digunakan untuk menyimpan semua SMS yang masuk ataupun dikirim oleh sistem.

Tabel 3.6. Tabel SMS

Field	Type	Size	Keterangan
Phone_Num	Text	15	Nomor ponsel
Date	Date/Time	Date/Time	Tanggal diterimanya SMS
Rcv_Msg	Text	160	SMS yang diterima
Rpl_Msg	Text	160	SMS yang dikirim
Status	Text	10	Status SMS

g. Tabel **Telepon**

Tabel ini digunakan untuk menyimpan tagihan rekening telepon.

Tabel 3.7. Tabel Telepon

Field	Type	Size	Keterangan
No_Telp	Text	15	Nomor telepon
Tagihan	Number	18	Jumlah tagihan
Bayar	Yes/No	Yes/No	Kondisi pembayaran

h. Tabel **Transaction**

Tabel ini digunakan untuk menyimpan semua transaksi yang dilakukan oleh nasabah.

Tabel 3.8. Tabel Transaction

Field	Type	Size	Keterangan
Acc_Num	Text	15	Nomor rekening
Date	Date/Time	Date/Time	Tanggal transaksi
Sign	Text	2	Kode transaksi
Debet	Number	18	Transaksi debet
Kredit	Number	18	Transaksi kredit
Saldo	Number	18	Jumlah saldo setelah transaksi

i. Tabel **Valas**

Tabel ini digunakan untuk menyimpan jenis mata uang dan kursnya.

Tabel 3.9. Tabel Valas

Field	Type	Size	Keterangan
Valuta	Text	10	Jenis valuta asing
Harga	Number	18	Kursnya terhadap rupiah

j. Tabel **Voucher**

Tabel ini digunakan untuk menyimpan kode pengisian ulang voucher.

Tabel 3.10. Tabel Voucher

Field	Type	Size	Keterangan
No_Voucher	Text	15	Nomor voucher untuk pengisian
Status	Yes/No	Yes/No	Status pembelian

3.3. Koneksi ke *Database Microsoft Access 2002*

Untuk dapat mengakses ke *database*, maka perlu adanya koneksi ke *database*. Koneksi ini dimulai dengan men-setting ODBC. Pilih tab *System DSN*, kemudian tekan tombol “ADD”. Kemudian pilih *driver* “Microsoft Access Driver” kemudian tekan tombol “FINISH”. Kemudian pilih tombol “SELECT” untuk menunjukkan dimana letak *database*. Koneksi ini memerlukan nama *data source*-nya dan lokasi *database*-nya. Pada Delphi kita cukup menggunakan

component ADOConnection untuk melakukan koneksi ke *database*. Pada *Properties Connection String* ketikkan:

```
Provider=MSDASQL.1;Persist Security Info=False;Data Source=TA;Extended
Properties="DSN=TA;DBQ=..\Database TA\MobileBanking.mdb;DriverId=25;FIL=MS
Access;MaxBufferSize=2048;PageTimeout=5;"
```

Kemudian pada *Properties Default Database* ketikan:

```
..\Database TA\MobileBanking.mdb
```

Selanjutnya untuk menginputkan data baru ke dalam tabel cukup menggunakan *component* ADOTable. Kemudian *properties Connection*-nya kita isi dengan ADOConnection dan pada *properties Table Name* kita isi dengan nama tabel yang ingin kita masukkan datanya dan perintah untuk memasukkan data dapat dilakukan seperti pada contoh berikut:

```
ADOTable.Append;
ADOTable['Pswd']:= '*****';
ADOTable['Acc_Num']:= '591-02-51459-2';
ADOTable['Phone_Num']:= '628115429685';
ADOTable.Post;
```

Kemudian untuk melakukan query cukup dengan menambahkan *component* ADOQuery, kemudian ubahlah *properties connection*-nya menjadi ADOConnection dan ditambahi dengan contoh perintah sebagai berikut:

```
Query.SQL.Clear;
Query.SQL.Append('SELECT *');
Query.SQL.Append('From Account');
Query.SQL.Append('WHERE Acc_Num = ""591-02-15495-2""');
Query.ExecSQL;
```

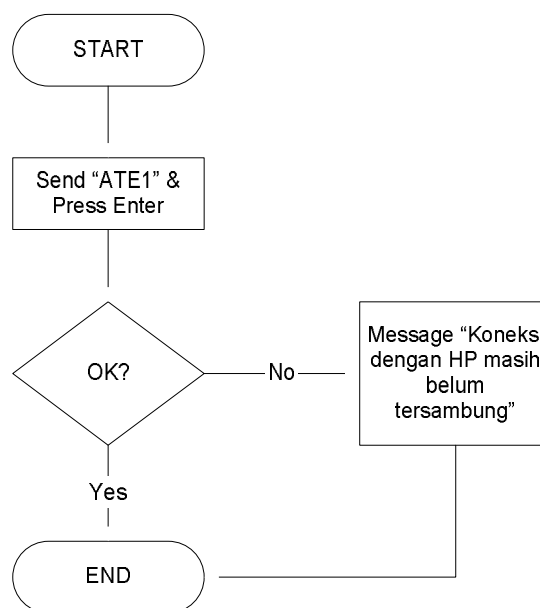
3.4. Perencanaan SMS

Dalam hubungan antara PC komputer dengan *handphone*, pengiriman data melalui *serial port*-nya untuk kemudian disalurkan ke *handphone*. Serial port yang digunakan untuk sistem ini adalah port 1 dengan *baud rate* 19200, sesuai dengan ketentuan untuk ponsel *server* yang dipakai, yaitu Siemens C45. *Baud rate* itu yang dipilih karena *baud rate* untuk Siemens C45 telah ditentukan 19200. Penggunaan *baud rate* diatas maupun dibawah angka 19200 tidak akan direspon

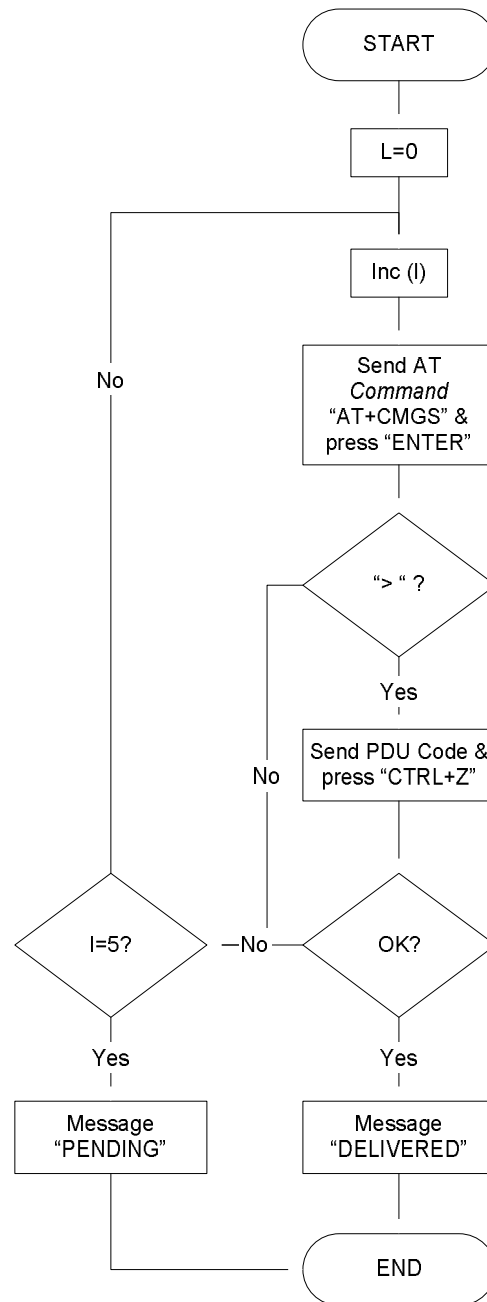
oleh ponsel tersebut. Untuk dapat menggunakan serial port maka harus digunakan *component* tambahan dari Delphi, yaitu Vacomm.

3.4.1. Pengecekan Port

Untuk pengecekan serial port dapat dilakukan dengan menjalankan *script* seperti pada lampiran 5. Script tersebut akan mengirimkan perintah AT Command “ATE1” dengan diakhiri penekanan “ENTER” dan menunggu respon dari *handphone*. Ada dua macam respon yang diberikan oleh *handphone*, yaitu: “OK” atau “ERROR”. Apabila komunikasi telah siap maka *handphone* akan mengirim respon “OK”, sedangkan jika masih ada kesalahan maka *handphone* akan memberikan respon “ERROR” yang kemudian oleh *script* diatas akan menampilkan pesan “Koneksi dengan HP masih belum tersambung”. Prosesnya adalah sebagai berikut seperti pada gambar 3.8.

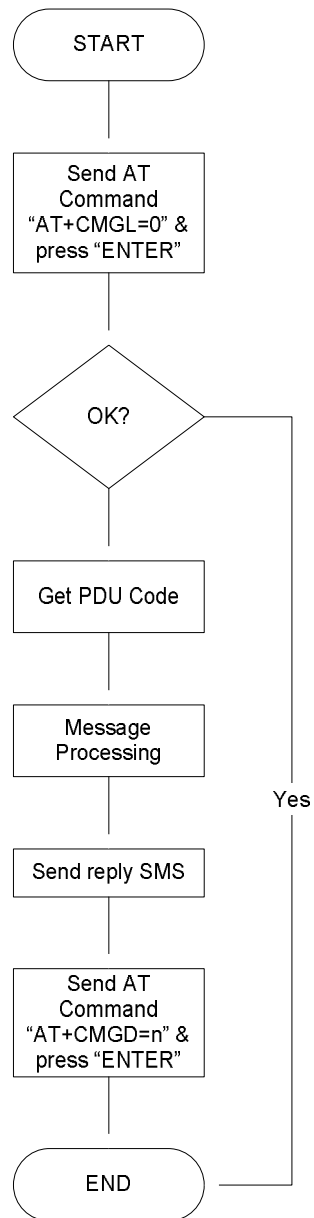


perintah sukses dijalankan atau akan memberikan respon “ERROR” jika gagal. Setelah muncul prompt “> ” maka akan diisi dengan kode PDU dari SMS yang akan dikirim. Respon yang dikirim oleh *handphone* juga ada dua, yaitu “OK” jika berhasil dikirim atau “ERROR” jika proses pengiriman gagal. Setiap kali terjadi “ERROR” maka *script* diatas akan memulai dari awal lagi, akan tetapi semua proses ini hanya akan diulang sebanyak lima kali. Jika dalam lima kali percobaan gagal terus maka SMS itu akan berstatus “PENDING”. Lain halnya jika berhasil maka statusnya “DELIVERED”. Proses untuk mengirim SMS adalah seperti pada gambar 3.9.



component Timer. Sehingga *script* tersebut hanya akan dijalankan setiap 10 detik. *Script* tersebut adalah seperti pada lampiran 1.

Script tersebut akan mengirim perintah AT *Command* “AT+CMGL=0”. Kemudian menunggu respon dari *handphone*. Respon ini terdiri dari dua macam, yaitu “OK” jika tidak ada SMS yang baru dan sekumpulan bilangan heksadesimal ada SMS baru yang masuk. Sekumpulan bilangan heksa tersebut akan diolah oleh procedure ReceiveSMS. Setelah diolah SMS tersebut akan dihapus pada saat *script* diatas mengirim perintah AT *Command* “AT+CMGD=n” dimana nilai n adalah nomor referensinya yang telah diambil sebelumnya. Respon yang diberikan juga terdiri dari dua macam, yaitu “OK” jika SMS tersebut berhasil dihapus dan “ERROR” jika SMS tersebut tidak berhasil dihapus. Proses untuk memeriksa dan menghapus SMS adalah seperti pada gambar 3.10.



3.4.5. Format SMS

SMS yang dikirim oleh para nasabah bisa saja terdiri dari dua sampai empat *header*. Minimum ada dua *header*. *Header-header* tersebut, yang dipisahkan oleh satu spasi, adalah seperti pada gambar 3.11.

<i>Password</i>	Kode perintah	<Optional 1>	<Optional 2>
-----------------	---------------	--------------	--------------

Gambar 3.11. Header *Requested SMS*

Berikut ini adalah penjelasan mengenai *header-header* tersebut:

a. *Password*

Password ini adalah kode yang dikirimkan kepada nasabah setelah proses pembukaan rekening baru. *Password* dan nomor ponsel harus sesuai dengan pada saat pembukaan rekening kecuali telah terjadi penggantian nomor ponsel atau *password*.

b. Kode Perintah

Kode ini tergantung dengan transaksi apa yang diinginkan. Berikut ini adalah jenis kode dan penjelasannya:

- CS

Untuk mengecek jumlah saldo akhir dari rekening nasabah yang bersangkutan.

- VA

Untuk mengetahui valuta asing yang dilayani oleh pihak bank dan kurs-nya pada saat itu.

- SB

Untuk mengetahui jenis tabungan dan suku bunganya.

- CHGPSWD

Untuk mengganti password.

Contoh:

FAOFL CHGPSWD kepek kepek

Password aslinya adalah “FAOFL” yang kemudian diganti dengan “kepek” yang diverifikasi ulang.

- TSF

Untuk mentransfer sejumlah uang ke rekening dalam satu bank.

Contoh:

FAOFL TSF 591-01-12345-2 500000

Rekening nasabah yang bersangkutan akan didebet sebesar Rp. 500,000 untuk kemudian ditransfer ke rekening 591-02-12345-2.

- BT

Kode untuk membayar rekening telepon. Kode ini selalu diikuti oleh nomor telepon yang akan dibayar rekeningnya.

Contoh:

FAOFL BT 031-8476596

Rekening nasabah yang bersangkutan akan didebet sesuai dengan tagihan dari nomor telepon 031-8476596.

- BL

Kode untuk membayar rekening listrik. Sama dengan telepon untuk pembayaran listrik juga harus diikuti oleh nomor langganan dari yang akan dibayar.

Contoh:

FAOFL BL AA123456-7

Rekening nasabah yang bersangkutan akan didebet sesuai dengan tagihan dari rekening listrik dengan nomor langganan AA123456-7.

- VC

Kode untuk pembelian voucher. Nasabah akan memperoleh nomor untuk pengisian pulsa. Fasilitas ini hanya untuk kartu pra-bayar.

c. Optional

Sebagai *field* tambahan seperti yang digunakan untuk BL, BT, CHGPSWD dan TSF.

Untuk lebih jelasnya mengenai kode *request*, format penulisan, isi SMS balasan dan kesalahan pengiriman dapat dilihat pada tabel 3.11 berikut ini:

Tabel 3.11. SMS *Command* dan Penjelasannya

Keterangan	Kode <i>Request</i>	Format Penulisan	Isi SMS balasan
Pengecekan saldo	CS	<password> CS	Saldo anda adalah Rp. <Jumlah saldo>
Transfer antar rekening	TSF	<password> TSF <rekening tujuan> <jumlah dana yang ingin dipindahbukukan>	Dana anda telah dialokasikan sejumlah Rp. <jumlah dana> ke rekening <rekening tujuan>
Perubahan password	CHGPSWD	<password> CHGPSWD <password baru> <verifikasi password baru>	Password anda telah diubah
Pembayaran telepon	BT	<password> BT <nomor telepon>	Rekening anda telah didebet untuk pembayaran telepon
Pembayaran listrik	BL	<password> BL <nomor langganan listrik>	Rekening anda telah didebet untuk pembayaran listrik
Pembelian voucher	VC	<password> VC	Nomor voucher untuk pengisian ulang adalah <nomor voucher>

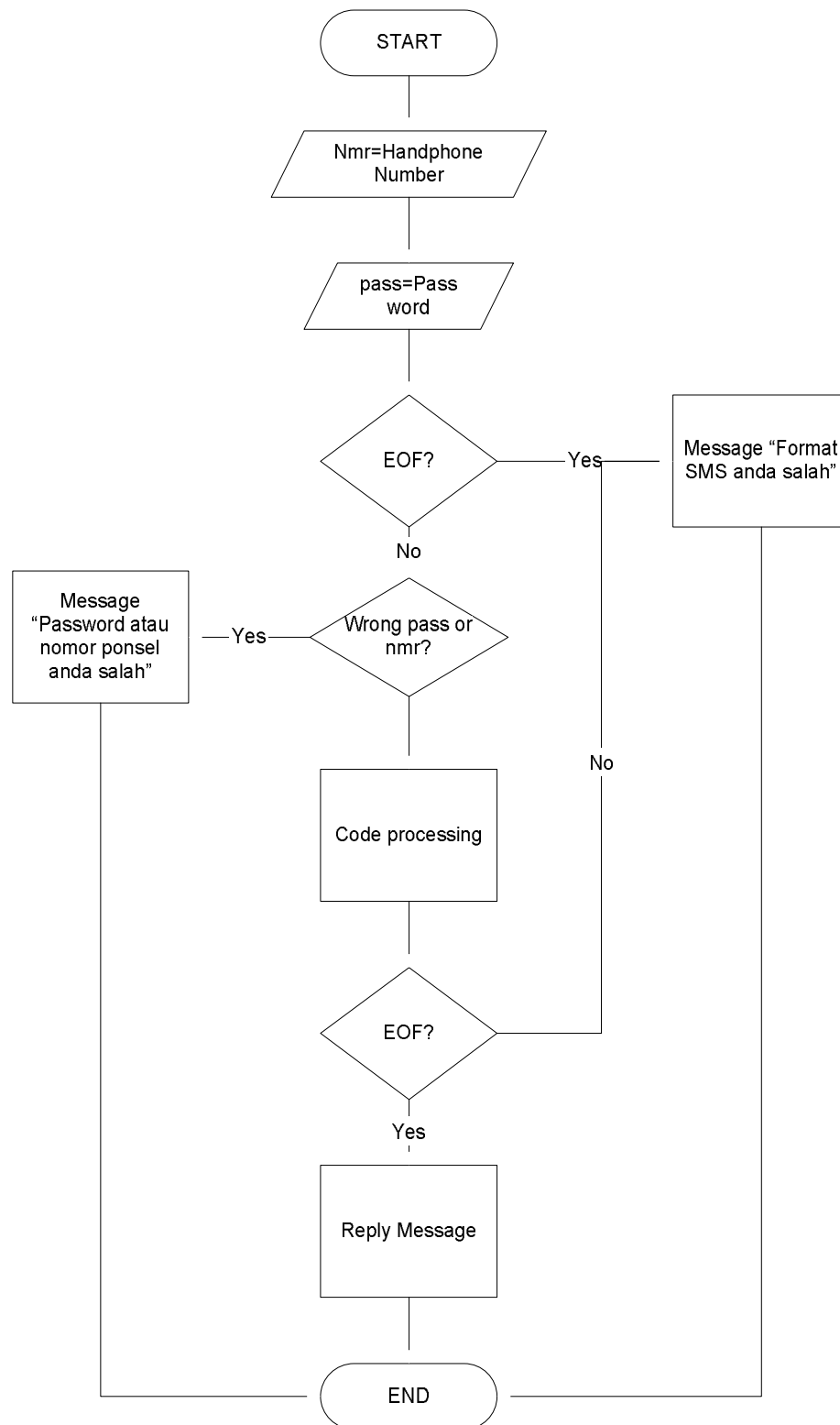
Tabel 3.11. SMS Command dan Penjelasannya (Samb.)

Keterangan	Kode Request	Format Penulisan	Isi SMS balasan
Data valuta asing	VA	<password> VA	AUD Rp. 5428 CHF Rp. 6347 EUR Rp. 9697 GBP Rp. 13521 HKD Rp. 1059 JPY Rp. 69 NZD Rp. 4798 SGD Rp. 4790 THB Rp. 198 USD Rp. 8220
Data suku bunga	SB	<password> SB	BCADOLLAR : 1.25% DEPOSITO : 9% TAHAPAN : 6,5% TAPRES : 7%
Kesalahan format SMS	-	-	Format SMS yang anda kirim salah
Kesalahan nomor rekening tujuan (untuk transfer)	-	-	Nomor rekening tujuan masih salah
Kurangnya dana untuk melakukan transaksi	-	-	Dana anda tidak mencukupi untuk melakukan transaksi
Password yang diubah memiliki kembaran	-	-	Password anda kembar

Tabel 3.11. SMS *Command* dan Penjelasannya (Samb.)

Keterangan	Kode <i>Request</i>	Format Penulisan	Isi SMS balasan
Rekening yang akan dibayar sudah terbayar	-	-	Rekening <jenis tagihan> tersebut telah dibayar
Rekening yang akan dibayar tidak ada	-	-	Nomor telepon yang anda masukkan masih salah
Kesalahan <i>password</i> atau nomor ponsel	-	-	Permintaan anda tidak dapat dilakukan karena kesalahan <i>password</i> atau belum terdaftarnya nomor handphone anda

Script untuk memproses isi dari SMS seperti pada lampiran 3. *Script* tersebut akan mengolah isi dari SMS menjadi beberapa bagian, yaitu *password* dan kode *request*. Seandainya salah maka sistem akan mengirim SMS yang menyatakan bahwa *password* atau nomor ponselnya masih salah. Begitu juga jika kode *request*-nya yang salah. Prosesnya adalah seperti pada gambar 3.12 berikut ini:



3.4.6. Proses *Parsing Message*

Setelah pengiriman *AT Command* “AT+CMGL=n” maka akan ada balasan dari ponsel berupa PDU atau tanda OK. PDU yang masuk masih berupa bilangan heksadesimal. Oleh karena itu *PDU Code* tersebut masih harus diproses untuk mendapatkan nomor pengirim dan isi SMS tersebut. Berikut ini adalah proses yang harus dilakukan beserta contohnya. Contoh *PDU Code* yang masuk adalah:

07912658050000F0040C9126581610739800003070225123800005E8329BFD06

a. *Parsing nomor SMS-Centre*

Pertama-tama yang dilakukan adalah membaca 2 digit pertama dari *PDU Code* yang merupakan petunjuk jumlah pasangan bilangan heksa dari *SMS-Centre*. Pada contoh angka yang diambil adalah 07. Bilangan ini masih dalam bentuk bilangan heksadesimal sehingga harus diubah terlebih dahulu. Hasil yang didapat adalah 7, berarti jumlah pasangan bilangan heksa untuk *SMS-Centre* terdiri dari 7 pasang. Kemudian adalah pembacaan nomor *SMS-Centre* yang terdiri dari 14 digit (7 pasang), berarti mulai dari digit ke-3 sampai dengan digit ke-16. Pasangan pertama merupakan kode dari nomor *SMS-Centre*, dimana *PDU Code* yang didapat adalah 91 yang merupakan kode internasional (nomor ponsel dimulai dengan angka “62”). Kemudian setiap pasangan heksa sisanya dibalik-balik, sehingga nomor *SMS-centre* yang didapat adalah 62855000000. Untuk digit terakhir yang berpasangan dengan “F” karena tidak memiliki pasangan cukup dihilangkan huruf “F” tersebut.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BFD06

b. *Parsing tipe SMS*

Untuk bagian ini pembacaan *PDU Code* dilanjutkan dari posisi terakhir setelah pembacaan *SMS-Centre*, yaitu mulai digit ke-17 s/d 18. Untuk tipe SMS hanya terdiri dari dua bilangan heksa. Jika *PDU Code* yang dibaca tersebut adalah 04 itu merupakan SMS-Terima.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

c. *Parsing* nomor pengirim

Parsing untuk bagian ini sama dengan untuk SMS-Centre. Proses pembacaan dimulai dari urutan terakhir setelah *parsing* tipe SMS, yaitu mulai digit ke-19. Pertama-tama yang kita baca mulai digit ke-19 s/d 20, yang hasilnya adalah 0C, yang mana berarti ada 12 digit. Digit-digit tersebut merupakan jumlah nomor ponsel pengirim. Kemudian mulai digit ke-21 s/d 22 merupakan kode dari nomor ponsel dimana hasilnya adalah 91 (kode internasional). Kemudian proses pembacaan nomor ponsel pengirim yang dimulai dari digit ke-23 s/d 34. Setiap PDU Code tersebut dipasang-pasangkan tiap dua bilangan untuk kemudian dibalik-balik, hasilnya adalah 628561013789.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

d. *Parsing* bentuk SMS

Pembacaan dimulai dari digit ke-35 s/d 36. PDU Code ini adalah bentuk dari SMS. Bilangan heksa yang didapat adalah 00 yang berarti SMS yang dikirim dalam bentuk SMS.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

e. *Parsing* skema encoding

Pembacaan dimulai dari digit ke-37 s/d 38. Bilangan heksa yang didapat adalah 00, yang berarti skema yang digunakan adalah skema 7 bit.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

f. *Parsing* tanggal dan waktu SMS diterima di SMS-Centre

Pembacaan dimulai dari posisi pembacaan terakhir, yaitu mulai digit ke-39 s/d 44. PDU Code yang diambil adalah PDU Code untuk tanggal dimana cara pembacaannya mulai dari belakang. PDU Code yang didapat adalah 307022, dan hasil yang didapat adalah 22/07/03, yang mana merupakan tanggal diterimanya SMS di SMS-Centre. Pembacaan berikutnya dimulai dari digit ke-45 s/d 50. PDU Code yang diambil adalah PDU Code untuk waktu dimana cara pembacaannya berlainan dengan tanggal. Setiap pasangan bilangan heksa yang terdiri dari dua bilangan heksa dibalik-balik, sehingga dengan bilangan heksa yang didapat, yaitu 512380, kita akan mendapatkan waktu diterimanya SMS di SMS-Centre, yaitu 15:32:08.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

g. *Parsing* batas waktu validitas

Pembacaan untuk bagian dimulai dari digit ke-51 s/d 52. Data yang didapat adalah 00 yang mana berarti SMS tersebut tidak dibatasi waktu validnya.

Contoh:

07912658050000F0040C9126581610739800003070225123800005E8329BF
D06

h. *Parsing* Isi SMS

Untuk pembacaan isi SMS, yang pertama harus dilakukan adalah membaca PDU Code yang dimulai dari digit ke-53 s/d 54, yang mana merupakan jumlah huruf dari isi SMS dalam bentuk heksadesimal. Dalam contoh bilangan heksa yang didapat adalah 05, yang mana berarti isi SMS terdiri dari lima karakter. Setelah itu pembacaan dilanjutkan dari digit ke-55 sampai akhir dari PDU Code secara keseluruhan. PDU Code yang diambil tersebut terdiri dari bilangan heksa sejumlah dua kali jumlah karakter dari isi SMS. Dalam contoh jumlah bilangan PDU Code yang diambil terdiri dari 10 bilangan heksa. Bilangan heksa tersebut dipasangkan tiap dua bilangan. Kemudian dalam tiap

pasangan, tiap bilangan heksa diubah menjadi bilangan biner. Setelah menghasilkan bilangan biner, maka bilangan biner dalam setiap pasangan digabungkan. Dari hasil penggabungan dalam tiap pasangan akan diambil tujuh digit, dimana digit yang tersisa ditambahkan pada akhir dari penggabungan bilangan biner dari pasangan berikut. Setiap tujuh digit yang diambil merupakan kode ASCII dari karakter yang diwakilkan. Pada akhirnya akan tersisa bilangan biner “0” sebanyak jumlah karakter. Bilangan-bilangan ini bisa dihapus. Untuk lebih jelasnya dipaparkan seperti berikut ini.

Heksadesimal	Biner 8 bit
E 8	1110 1000
3 2	0011 0010
9 B	1001 1011
F D	1111 1101
0 6	0000 0110

<u>1110</u> 1000		h
0011 <u>0010</u>	→ 1	e
<u>1001</u> 1011	→ 00	l
1111 <u>1101</u>	→ 100	l
<u>0000</u> 0110	→ 1111	o

Contoh:

07912658050000F0040C912658161073980000307022512380000**05E8329BF**

D06