

## 4. IMPLEMENTASI SOFTWARE

### 4.1.Desain Program

Dalam program ini ada beberapa bagian yang hanya bisa diakses oleh *user* tertentu saja, dimana aksesnya dilindungi oleh *password user* yang terenkripsi dengan menggunakan *metode* enkripsi RSA (*Rivest-Shamir Adleman*).

Program ini terdiri dari beberapa unit yang tergabung dalam sebuah *project* dengan nama **TA\_GPS**. Unit-unit yang terdapat dalam *project* ini antara lain adalah:

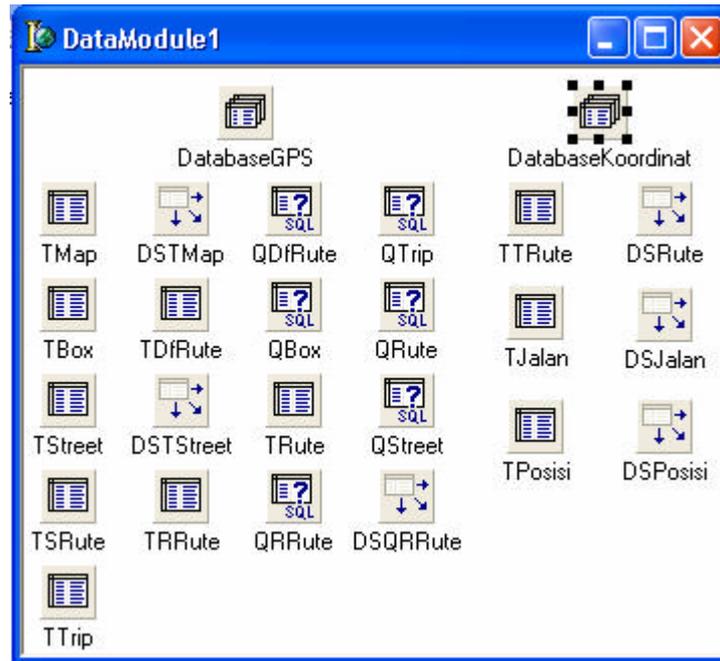
- Unit UMain: berisi program menu tampilan utama *software*.
- Unit ULogin: berisi program *login user* dan *password*.
- USettings: berisi program yang digunakan untuk mengatur *SIM card*, nomer *handphone* penerima yang digunakan dan juga *serial number* dimana *handphone* sistem dihubungkan dengan PC.
- Unit Global: merupakan unit yang dapat diakses oleh seluruh unit-unit lain.

Beberapa prosedur yang ada adalah:

- Procedure FastExponentiation(X,Z,N : LongInt; Var Y : LongInt; Sym1,Sym2,Sym3 : String)
- Procedure enkripsi(EN:string;var tanya:string)

Kedua prosedur di atas merupakan prosedur yang digunakan untuk mengenkripsi *password* dari *user* dengan menggunakan *metode* RSA (*Rivest-Shamir-Adleman Cryptosystem*)

- Unit UDataModule: berisi data *source* dan *query* yang digunakan pada *software* ini.



Gambar 4.1. Unit UDataModule

- Unit UTracking: berisi program yang digunakan untuk mengumpulkan data-data koordinat jalan-jalan dan point-point tertentu di Surabaya.
- Unit URuteTrack: berisi program yang digunakan untuk memasukkan rute baru yang digunakan oleh unit UTracking.
- Unit UKalibrasi: berisi program yang digunakan untuk kalibrasi peta Surabaya agar diperoleh batas-batas dari peta yang digunakan.
- Unit UPoint: berisi program untuk memilih point tertentu untuk digunakan pada unit UKalibrasi.
- Unit USimulation: berisi program simulasi dari program UCommunication dimana program ini mensimulasikan komunikasi GPS untuk melacak posisi mobil dengan menggunakan SMS.
- UMapping: merupakan unit awal untuk memasukkan, mengubah, dan menampilkan nama jalan, rute dan kotak jalan.
- UStreet: berisi program untuk memasukkan dan mengubah nama jalan.
- UBox: berisi program untuk memasukkan, mengubah dan menampilkan kotak jalan.
- URute: berisi program untuk memasukkan dan mengubah rute perjalanan.
- USMS: berisi program untuk mengirimkan pesan SMS.

- UView: berisi program untuk menampilkan data-data lokasi yang telah dilalui mobil berdasarkan tanggal.
- Unit UCommunication: berisi program komunikasi GPS untuk melacak posisi mobil dengan menggunakan SMS. Beberapa prosedur penting dalam unit ini adalah:
  - Program yang digunakan untuk melakukan koneksi antara Borland Delphi 6.0 dengan GPS *receiver*, dimana *serial port* yang digunakan adalah COM6 dan *baudrate*-nya adalah 4800 bps.

```

mscommGPS.CommPort:=6;
mscommGPS.InBufferSize:=1024;
mscommGPS.Settings:='4800,n,8,1';
if not mscommGPS.PortOpen then mscommGPS.PortOpen:=true;
mscommGPS.InBufferCount:=0;
mscommGPS.RThreshold:=512;
mscommGPS.InputLen:=0;

```

- Program yang digunakan untuk men-*filter* data dari GPS *receiver*, sehingga yang diterima hanya data dengan format \$GPRMC.

```

gpsstr:=mscommGPS.input;
gpsstrcpy:=gpsstr;
tmpstr:=copy(gpsstr,pos('$GPRMC',gpsstr),pos(#13,gpsstr)-1);

```

- Program yang digunakan untuk mengolah data \$GPRMC yang didapat dari GPS *receiver*, sehingga didapatkan data-data berupa jam, tanggal serta data bujur dan lintang.

```

if copy(tmpstr,1,pos(',',tmpstr)-1)='$GPRMC' then
begin
delete(tmpstr,1,pos(',',tmpstr));
//mendapatkan data jam dari $GPRMC
jam:=copy(tmpstr,1,2); mnt:=copy(tmpstr,3,2); dt:=copy(tmpstr,5,2);
EJam.Text:=jam+':' +mnt+':' +dt;
delete(tmpstr,1,pos(',',tmpstr));
//hanya data $GPRMC yang valid yang diproses
if copy(tmpstr,1,pos(',',tmpstr)-1)='A' then
begin
delete(tmpstr,1,pos(',',tmpstr));
if length(tmpstr)>=24 then
begin
// mendapatkan data lintang dari $GPRMC
latitudestr:=copy(tmpstr,1,pos(',',tmpstr)-1);
ELatitude.Text:="";
strq1:=copy(latitudestr,1,2); strq2:=copy(latitudestr,3,6);
ELatitude.Text:=strq1+'°'+strq2+'";
LintangY:=strtofloat(latitudestr);
delete(tmpstr,1,pos(',',tmpstr));
// menentukan apakah posisi Lintang di Utara atau Selatan
if copy(tmpstr,1,1)='S' then LLintang.Caption:='LS'
else LLintang.Caption:='LU';
delete(tmpstr,1,pos(',',tmpstr));
// mendapatkan data bujur dari $GPRMC
longitudestr:=copy(tmpstr,1,pos(',',tmpstr)-1);
ELongitude.Text:="";
strq1:=copy(longitudestr,1,3); strq2:=copy(longitudestr,4,6);
ELongitude.Text:=strq1+'°'+strq2+'";
BujurX:=strtofloat(longitudestr);
delete(tmpstr,1,pos(',',tmpstr));
// menentukan apakah posisi Bujur di Timur atau Barat
if copy(tmpstr,1,1)='E' then LBujur.Caption:='BT'
else LBujur.Caption:='BB';
delete(tmpstr,1,pos(',',tmpstr));
delete(tmpstr,1,pos(',',tmpstr));
delete(tmpstr,1,pos(',',tmpstr));
// mendapatkan data tanggal dari $GPRMC
tgl:=copy(tmpstr,1,2); bln:=copy(tmpstr,3,2); thn:=copy(tmpstr,5,2);
ETgl.Text:=tgl+'/' +bln+'/' +thn;
delete(tmpstr,1,pos(',',tmpstr));
delete(tmpstr,1,pos(',',tmpstr));
end;
end;
end;

```

- Program yang digunakan untuk mengolah SMS *request* dengan format “SEND”, untuk menghasilkan SMS *response* sesuai dengan yang diminta oleh *user* mengenai data posisi mobil.

```

setting:=Text;
RSMS:=copy(setting,1,4);
delete(setting,1,5);
if RSMS='SEND' then
begin
if copy(setting,3,1)= ' ' then
begin
waktu:=strtoint(copy(setting,1,2));
// SMS request yang meminta data setiap beberapa menit
if copy(setting,4,3)='MNT' then
begin
waktu:=waktu*60*1000;
timerSend.Enabled:=true;
timerSend.Interval:=waktu;
end else
// SMS request yang meminta data setiap beberapa detik
if copy(setting,4,3)='SCD' then
begin
waktu:=waktu*1000;
timerSend.Enabled:=true;
timerSend.Interval:=waktu;
end;
tipe:='waktu';
end;
// SMS request yang meminta data setiap posisi mobil pindah jalan
if copy(setting,1,pos('; ',setting)-1)='STREET' then
begin
timerSend.Enabled:=false;
tipe:='jalan';
end else
// SMS request yang meminta data setiap posisi mobil pindah kotak jalan
if copy(setting,1,pos('; ',setting)-1)='BOXES' then
begin
timerSend.Enabled:=false;
tipe:='box';
end;
end;
end;

```

- Program yang digunakan untuk mengolah SMS *request* dengan format “RUTE”, dimana *user* memberikan perintah kepada sistem agar sistem menggunakan rute perjalanan yang diberikan.

```

setting:=Text;
RSMS:=copy(setting,1,4);
delete(setting,1,5);
if RSMS='RUTE' then
begin
ind:=strtoint(copy(setting,1,pos('; ',setting)-1));
MStatus.Lines.Add('Use Rute: '+Rute[ind])
End;

```

- Program yang digunakan untuk mengolah SMS *request* dengan format “ALRM”, dimana *user* mengaktifkan *alarm* pada sistem.

```

setting:=Text;
RSMS:=copy(setting,1,4);
delete(setting,1,5);
if RSMS='ALRM' then
begin
  MediaPlayer1.Open;
  MediaPlayer1.Play;
End;

```

- Program yang digunakan untuk mengolah SMS *request* dengan format “STOP”, dimana *user* memberikan perintah kepada sistem agar berhenti mengirimkan SMS tentang posisi mobil.

```

setting:=Text;
RSMS:=copy(setting,1,4);
delete(setting,1,5);
if RSMS='STOP' then
begin
  timerSend.Enabled:=false;
  tipe:='jalan';
end;

```

- Program yang digunakan untuk mengolah SMS *request* dengan format “MESG”, dimana *user* memberikan perintah kepada sistem untuk menampilkan isi SMS dalam bentuk pesan-pesan singkat kepada pengemudi mobil, dengan cara menampilkan isi SMS tersebut dalam bentuk message dialog.

```

setting:=Text;
RSMS:=copy(setting,1,4);
delete(setting,1,5);
if RSMS='MESG' then
begin
  MessageDlg(setting,MtInformation,[MbOK],0);
end;

```

- Program yang digunakan untuk melakukan koneksi dengan *handphone* Nokia 6110 dimana *serial port* yang dapat digunakan adalah COM4 dan COM5 dengan *default port* COM5.

```
SMS.ComNumber:=ComHP;
// model koneksi DAU-9P
SMS.ConnectionMode:=0;
// model handphone seri 61xx
SMS.Model:=2;
SMS.Open;
```

#### 4.2.Desain *User Interface*

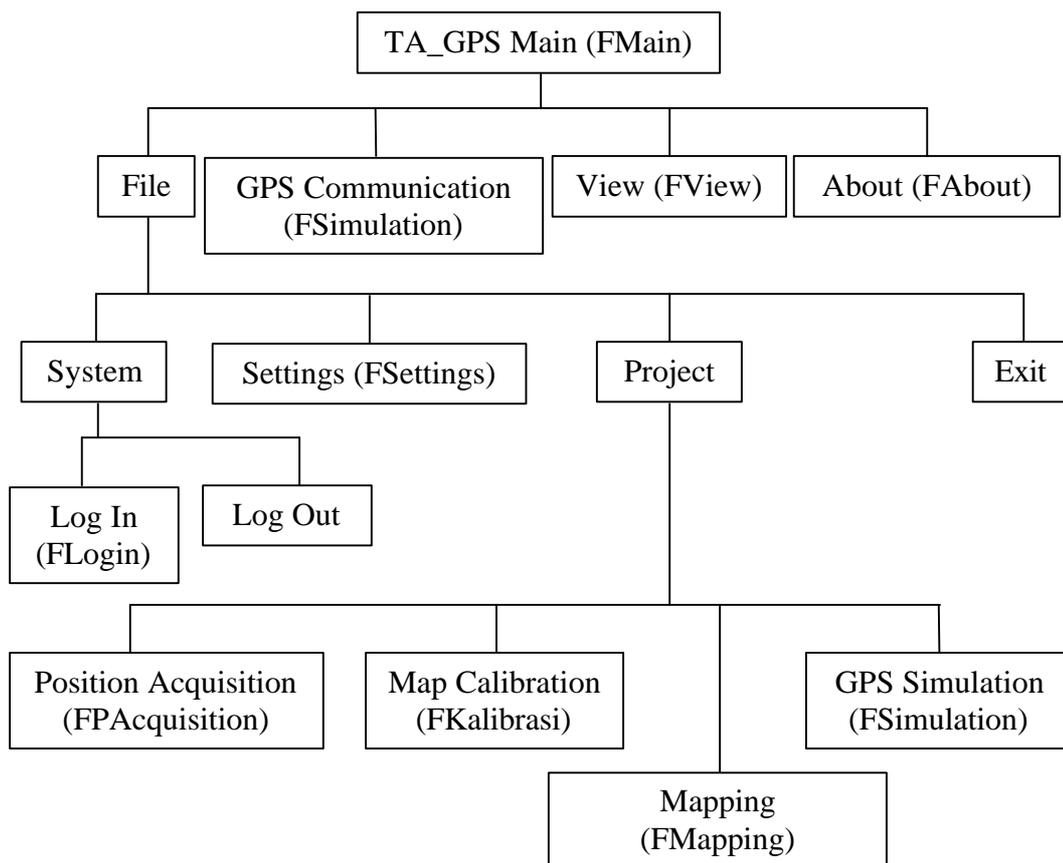
Desain *user interface* adalah bentuk masing-masing menu dan struktur menu dari program yang telah dirancang sebelumnya. Dalam mendesain *interface* program, hal-hal yang harus diperhatikan adalah:

1. Navigasi antar dan di dalam layer harus jelas,
2. Pengelompokan fungsi yang sejenis,
3. Pembukaan *window* atau layer di tengah,
4. Penggunaan kata yang jelas pada label dan pesan,
5. Penggunaan standar yang telah ada dan konsistensi.

Dalam *software* ini, pembuatan desain *user interface* dilakukan dengan pembuatan main menu untuk memudahkan dalam proses pemilihan kegiatan dalam *software* ini karena semua fasilitas yang ada pada *software* ini dapat dengan mudah diakses melalui *main* menu ini. Bentuk form FMain dapat dilihat pada gambar 4.2., sedangkan bagan main menu dapat dilihat pada gambar 4.3.



Gambar 4.2. Desain Form Menu Utama

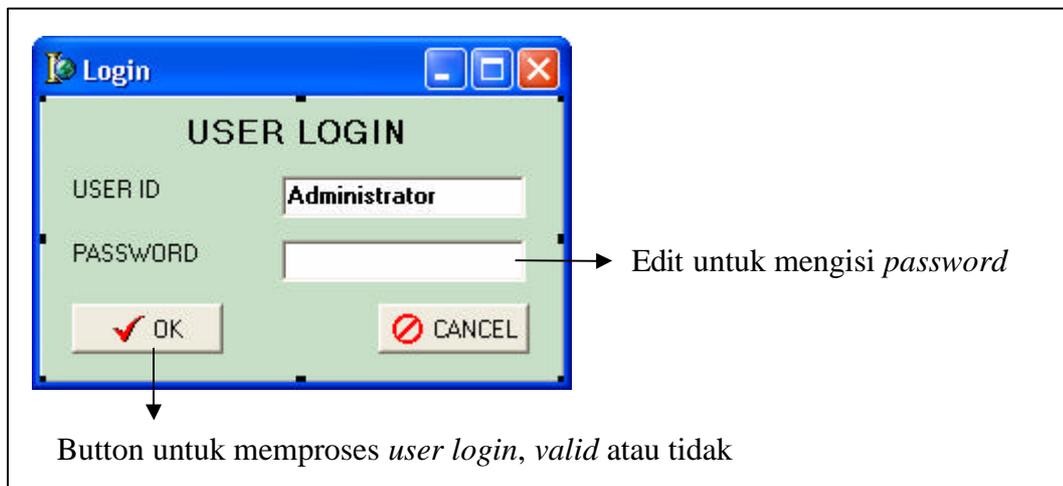


Gambar 4.3. Bagan Main Menu

Desain form pada *software* ini menggunakan button-button dengan gambar-gambar yang konsisten, sehingga memudahkan *user* untuk mengingatnya. Desain Form menu utama di atas merupakan gambaran dasar dari tampilan awal yang akan muncul jika *software* ini dijalankan. Sedangkan untuk form-form dengan fasilitas yang lain yang terdapat pada program ini untuk lebih jelasnya akan diterangkan sebagai berikut ini :

#### 4.2.1. Form FLogin

Form ini digunakan untuk akses *user* untuk beberapa bagian pada *software* ini.



Gambar 4.4. Desain dan Penjelasan Form FLogin

Cara kerja dari form FLogin adalah sebagai berikut:

1. *User* mengisi *password* pada edit Password, kemudian tekan Button Ok,
2. Sistem akan memprosesnya. Bila *password* yang di-input-kan benar maka akan muncul message dialog seperti terlihat pada gambar 4.5.



Gambar 4.5. Pesan *Password* Benar

Sedangkan bila *password* yang di-*input*-kan salah maka akan muncul message dialog seperti terlihat pada gambar 4.6.



Gambar 4.6. Pesan *Password* Salah

Berikut ini beberapa form yang membutuhkan form FLogin agar dapat diakses adalah:

- Form FPAcquisition
- Form FKalibrasi
- Form Mapping
- Form Settings
- Form View

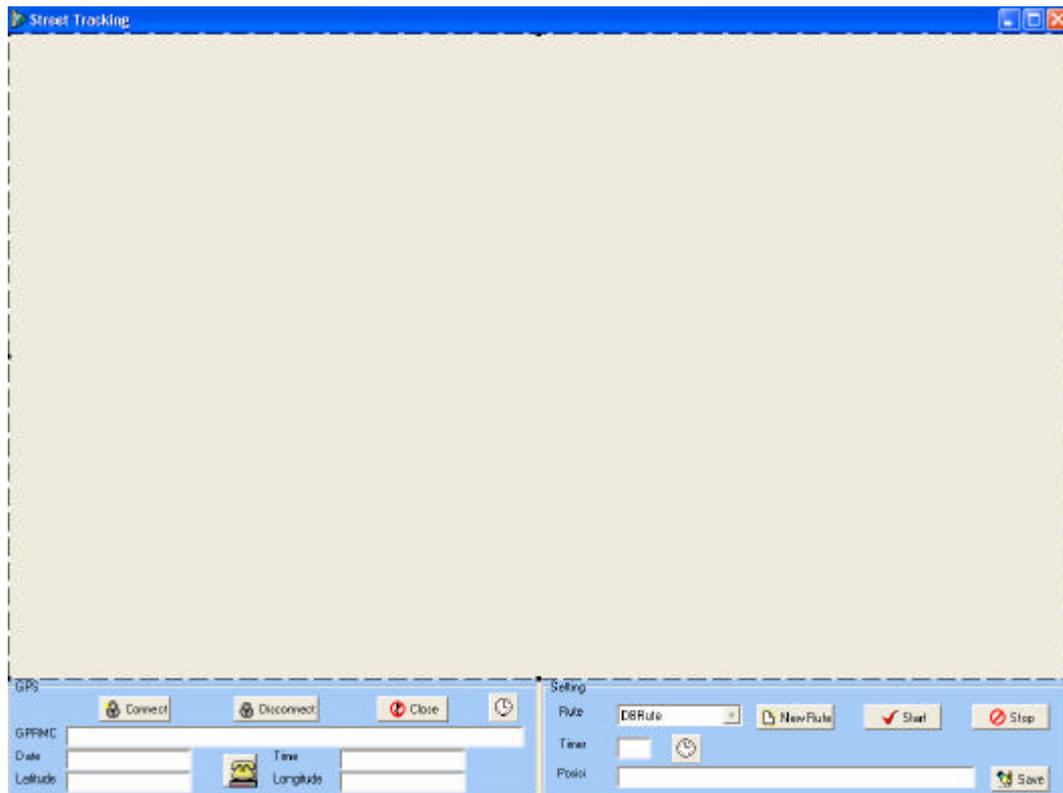
Apabila *user* selesai menggunakan form-form tersebut diatas, maka *user* perlu melakukan logout untuk melindungi hak akses dari form-form tersebut.



Gambar 4.7. Pesan Logout

#### 4.2.2. Form FPAcquisition

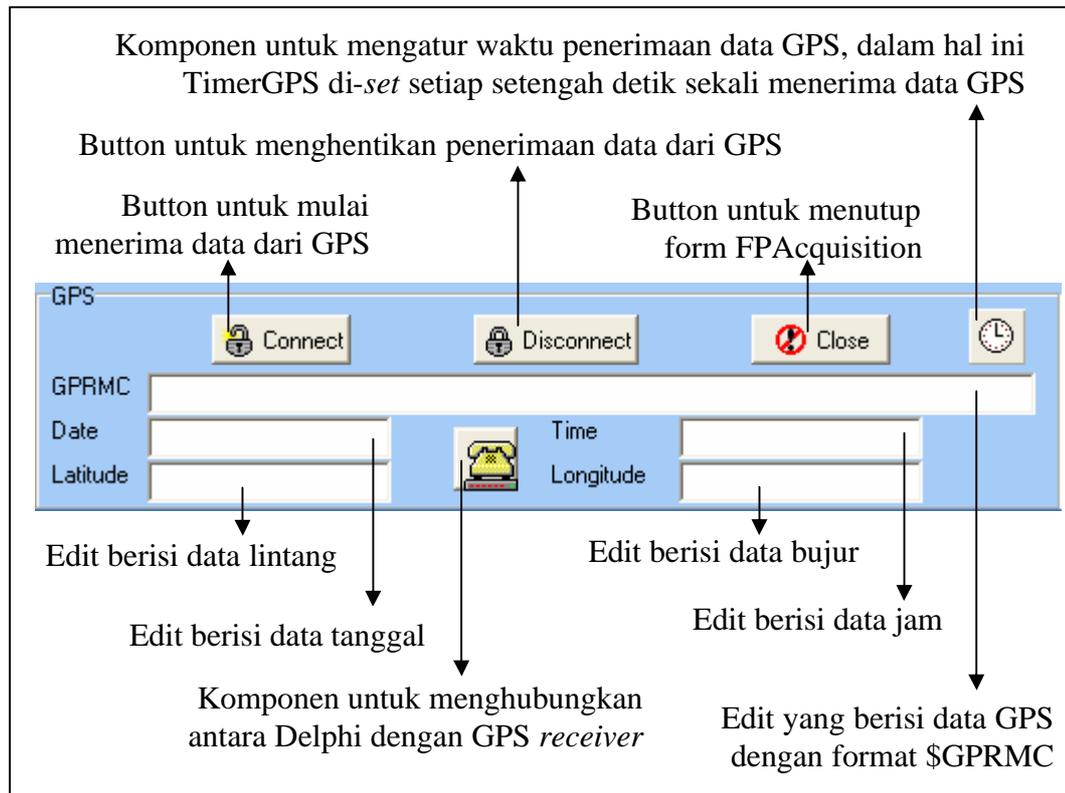
Form ini digunakan untuk mengumpulkan data-data koordinat jalan-jalan dan *point-point* tertentu di Surabaya. Tidak semua *user* dapat mengakses form ini. Apabila *user* belum login maka secara otomatis form FLogin akan muncul. Desain Form FPAcquisition dapat dilihat pada gambar 4.8.



Gambar 4.8. Desain Form FPAcquisition

Form FPAcquisition terbagi menjadi tiga bagian yaitu:

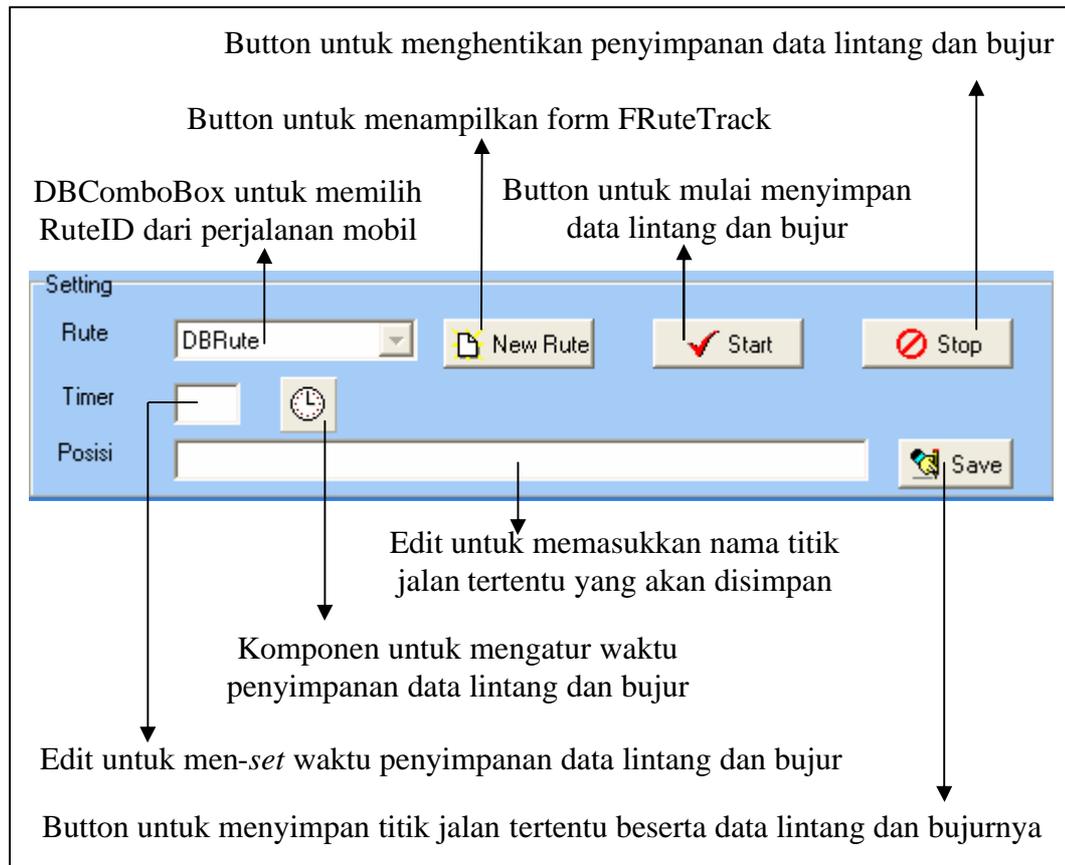
- Image  
Image digunakan untuk menggambarkan *route* perjalanan yang telah dilalui oleh mobil. Pada *image* akan tampak garis-garis yang menunjukkan bahwa mobil telah melalui jalan-jalan tersebut.
- Panel GPS  
Di dalam Panel GPS terdapat beberapa komponen yang berhubungan dengan data-data GPS seperti terlihat pada gambar 4.9.



Gambar 4.9. Desain dan Penjelasan Panel GPS Form FPAcquisition

- Panel Setting

Di dalam Panel Setting ini terdapat beberapa komponen yang digunakan untuk menyimpan data GPS yang diterima ke dalam *database* seperti terlihat pada gambar 4.10.



Gambar 4.10. Desain dan Penjelasan Panel Setting Form FPAcquisition

Cara kerja dari form FPAcquisition adalah sebagai berikut:

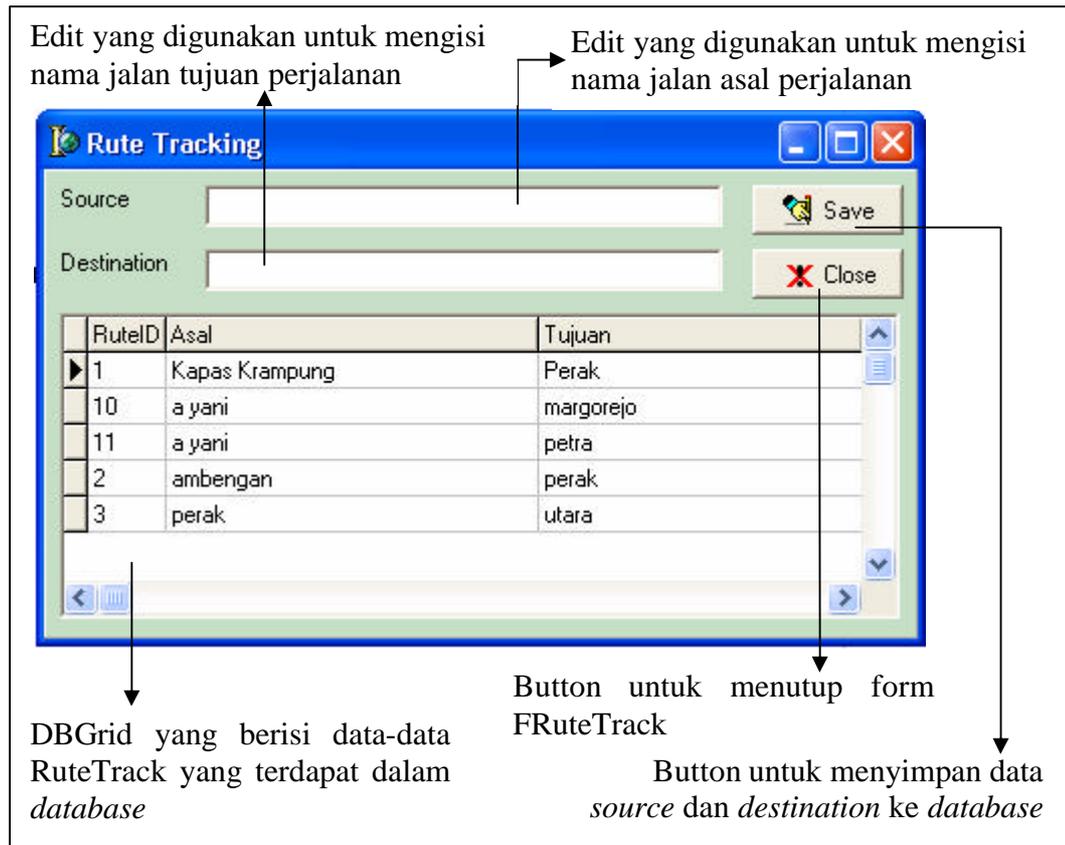
1. Untuk dapat menerima data-data dari GPS *receiver* maka perlu dilakukan koneksi antara Delphi dengan GPS *receiver* yaitu dengan cara menekan button Connect.
2. Setelah Delphi dan GPS *receiver* terhubung, maka data GPS dengan format \$GPRMC akan ditampilkan pada edit \$GPRMC.
3. Data \$GPRMC tersebut kemudian akan diproses untuk mendapatkan data-data waktu, lintang dan bujur, dimana data-data tersebut akan ditampilkan pada edit Date, edit Time, edit Latitude dan edit Longitude.
4. Untuk memasukkan rute perjalanan mobil yang baru, ditekan button New Rute, maka akan muncul form FRuteTrack.
5. Agar sistem dapat mulai menyimpan data lintang dan bujur posisi mobil ke dalam *database*, maka pertama-tama *user* memilih RuteID pada

DBComboBox RuteID, kemudian ditekan button Start. Maka sistem akan mulai menyimpan data-data GPS setiap lima detik sekali (*default*).

6. Untuk mengubah waktu penyimpanan data dapat dilakukan dengan mengisi angka tertentu pada edit timer. Sebagai contoh bila *user* menginginkan agar sistem menyimpan data-data GPS setiap satu detik sekali, maka pada edit timer *user* mengisikan angka satu.
7. Untuk menyimpan data lintang dan bujur suatu titik, pertama-tama *user* perlu meng-*input*-kan nama titik tersebut, kemudian ditekan button Save.
8. Untuk menghentikan proses penyimpanan data oleh sistem, ditekan tombol Stop.
9. Untuk memutuskan koneksi antara Delphi dengan GPS *receiver*, ditekan button Disconnect.
10. Untuk menutup form FPAcquisition, ditekan button Close.

#### 4.2.3. Form FRuteTrack

Form ini adalah form yang digunakan oleh form FTracking untuk memasukkan *rute* perjalanan baru. Desain form FRuteTrack beserta penjelasannya dapat dilihat pada gambar 4.11.



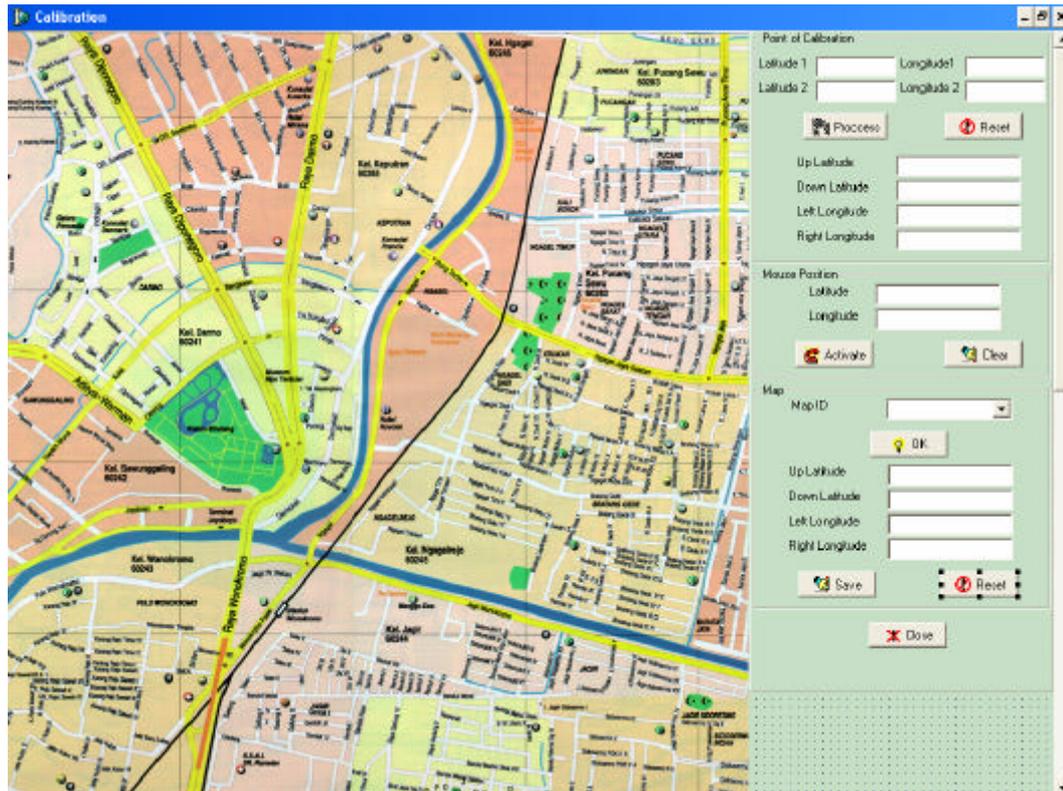
Gambar 4.11. Desain dan Penjelasan Form FRuteTrack

Cara kerja dari form FRuteTrack adalah sebagai berikut:

1. *User* memasukkan nama jalan yang menjadi asal perjalanan pada edit Source.
2. Kemudian *user* memasukkan nama jalan yang menjadi tujuan perjalanan pada edit Destination.
3. Setelah asal dan tujuan perjalanan dimasukkan, untuk menyimpannya ke dalam *database* ditekan button Save.
4. Untuk menutup form FRuteTrack, ditekan button Close.

#### 4.2.4. Form FKalibrasi

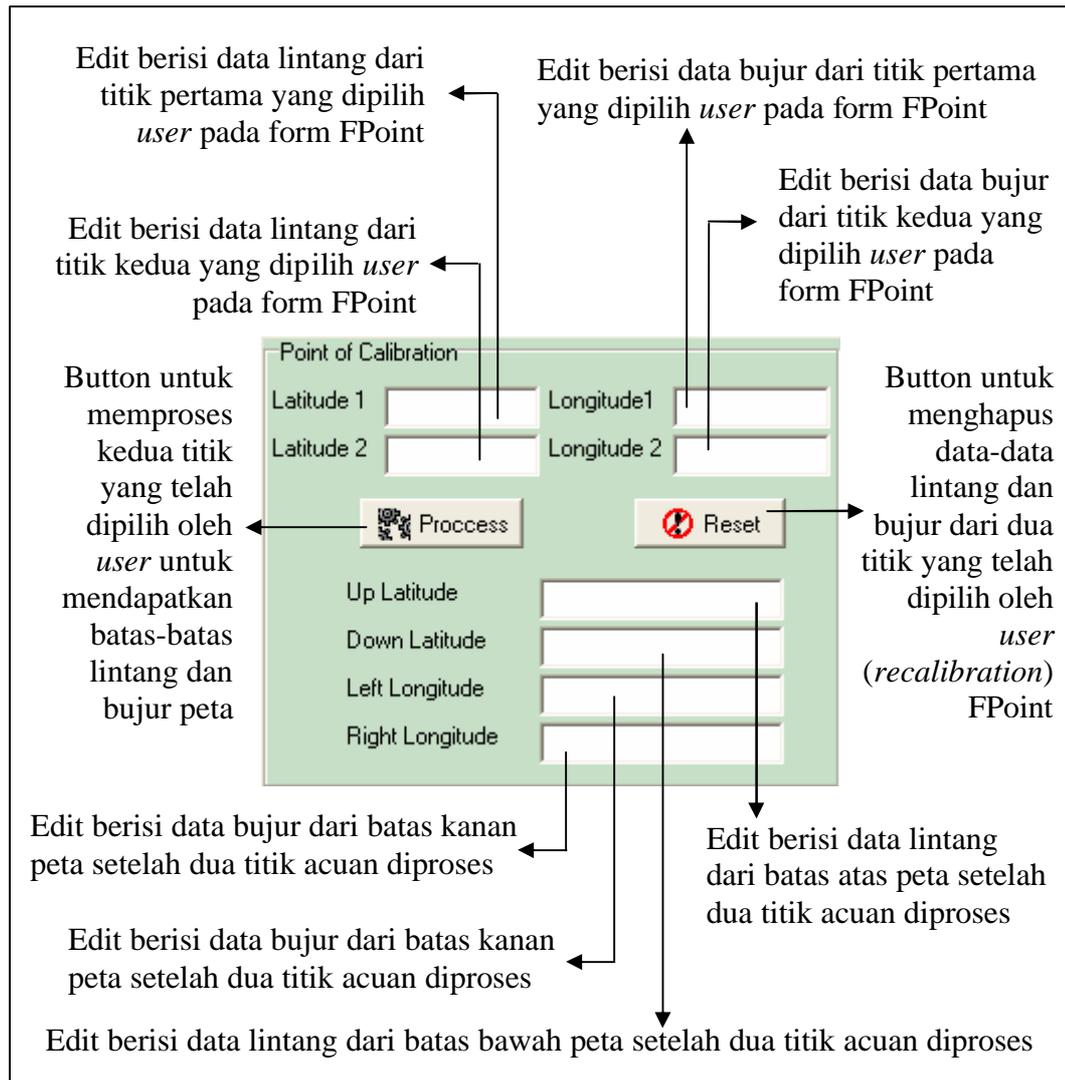
Image peta Surabaya yang digunakan dalam *software* dibagi menjadi tiga bagian. Oleh karena itu, perlu dilakukan penyimpanan data batas lintang dan bujur dari peta-peta tersebut. Form FKalibrasi inilah yang digunakan untuk menentukan batas-batas dari peta. Desain form FKalibrasi dapat dilihat pada gambar 4.12.



Gambar 4.12. Desain Form FKalibrasi

Form FKalibrasi terbagi menjadi empat bagian yaitu:

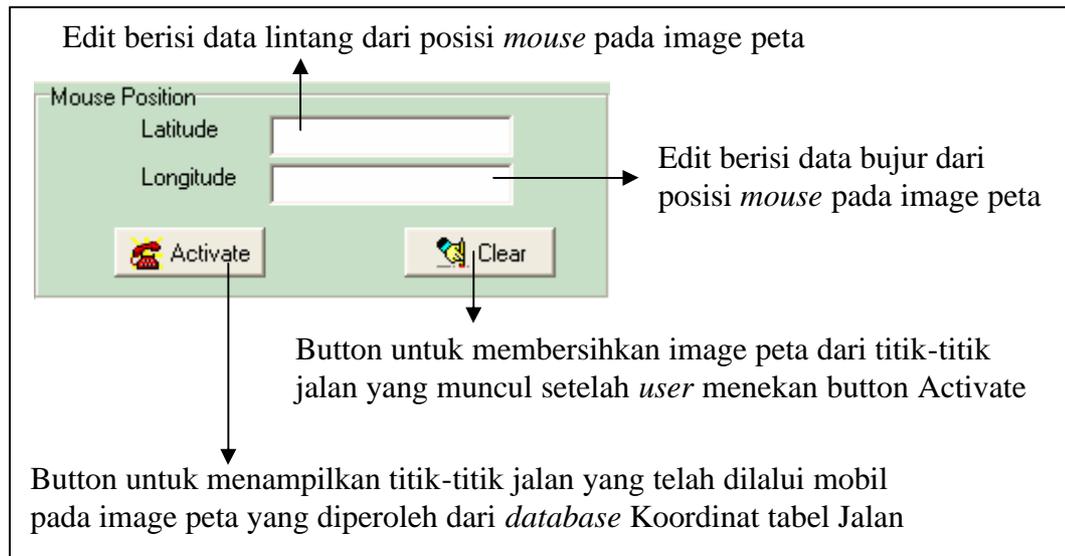
- Image  
Image berisi sebagian peta Surabaya yang digunakan dalam *software* ini.
- Panel Point of Calibration  
Di dalam Panel Point of Calibration terdapat beberapa komponen yang berhubungan dengan data-data dari dua titik yang dipilih oleh *user* sebagai titik acuan untuk menentukan batas-batas peta seperti terlihat pada gambar 4.13.



Gambar 4.13. Desain dan Penjelasan Panel Point of Calibration Form FKalibrasi

- Panel Mouse Position

Di dalam Panel Mouse Position terdapat beberapa komponen yang berhubungan dengan data-data lintang dan bujur dari posisi *mouse* pada image peta seperti terlihat pada gambar 4.14.

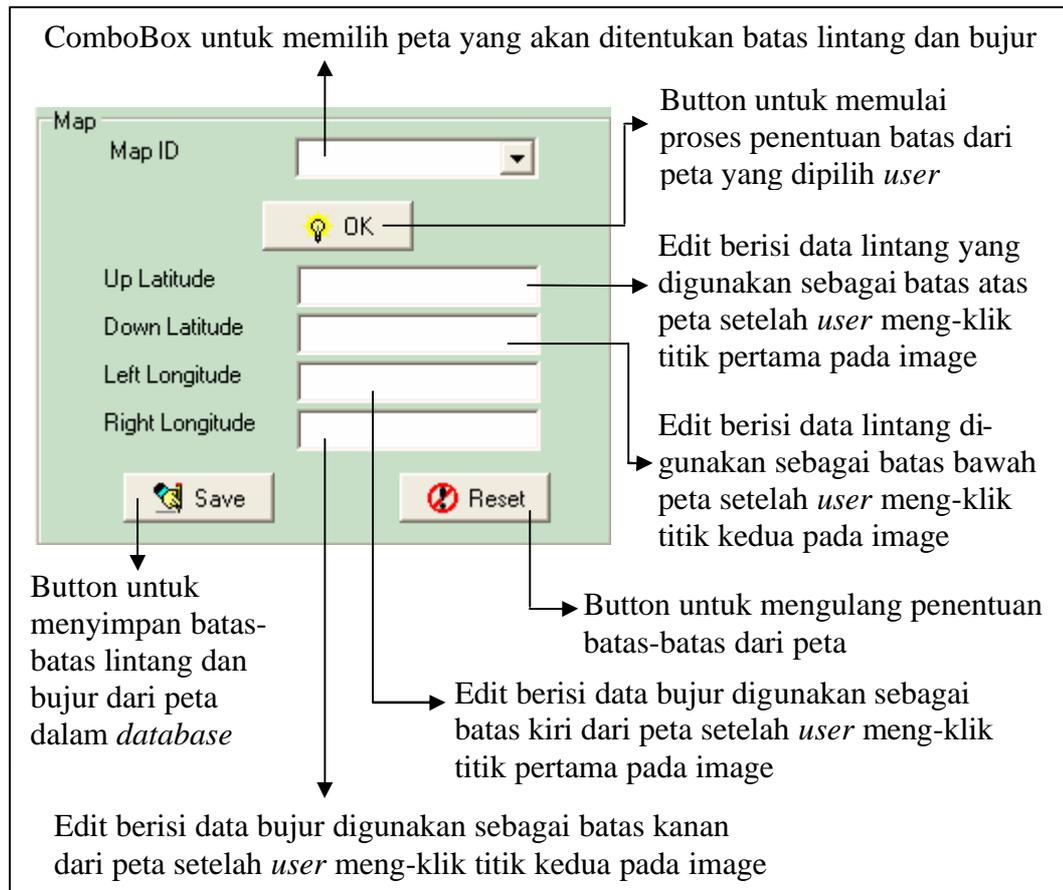


Gambar 4.14. Desain dan Penjelasan Panel Mouse Position Form FKalibrasi

Dengan menampilkan titik-titik jalan pada image peta, *user* dapat mengetahui apakah batas-batas peta yang telah diproses tersebut akurat atau tidak. Bila *user* merasa batas-batas peta tersebut tidak akurat maka dapat melakukan *recalibration* dengan menekan button Reset pada panel Point of Calibration.

- Panel Map

Di dalam Panel Map terdapat beberapa komponen yang berhubungan dengan batas-batas lintang dan bujur dari peta-peta yang ada seperti terlihat pada gambar 4.15.

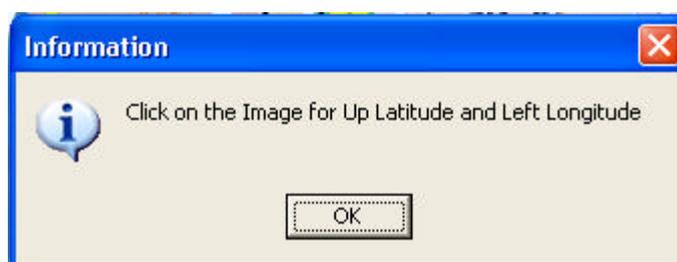


Gambar 4.15. Desain dan Penjelasan Panel Map Form FKalibrasi

Cara kerja dari form FKalibrasi ini adalah sebagai berikut:

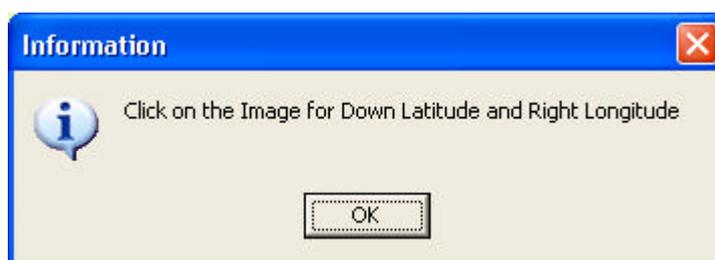
1. Pertama-tama untuk memilih dua titik acuan pada image, *user* mengklik pada image peta. Setelah itu akan muncul form FPoint. Pada form FPoint, *user* memilih nama dari titik tersebut yang kemudian secara otomatis akan ditentukan lintang dan bujur dari titik tersebut berdasarkan namanya. Data lintang dan bujur tersebut yang kemudian ditampilkan pada edit Latitude dan edit Longitude. Kedua titik acuan tersebut merupakan titik-titik yang digunakan untuk menentukan batas-batas lintang dan bujur dari peta.
2. Setelah *user* mengklik dua titik, ditekan button Process untuk memproses kedua titik acuan tersebut sehingga didapatkan batas lintang dan bujur dari image peta. Batas lintang atas peta ditampilkan pada edit Up Latitude, batas lintang bawah peta ditampilkan pada edit Down Latitude, batas bujur kiri peta ditampilkan pada edit Left Longitude, sedangkan batas bujur kanan peta ditampilkan pada edit Right Longitude.

3. Untuk mengulang proses nomer satu dan dua, ditekan button Reset, sistem akan menghapus semua data lintang dan bujur dari dua titik acuan serta data batas lintang dan bujur peta. Ini berarti bahwa *user* melakukan *recalibration*.
4. Bila telah didapatkan batas lintang dan bujur dari peta, *user* dapat mengecek keakuratan dari batas-batas tersebut dengan cara menggunakan data yang didapatkan dari form FPAcquisition. Untuk itu button Activate ditekan, maka posisi titik-titik yang telah dilalui oleh mobil dari form FPAcquisition akan ditampilkan pada peta.
5. Untuk membersihkan peta dari titik-titik jalan, ditekan button Clear.
6. Kemudian untuk menentukan batas-batas dari peta yang terdapat pada *database*, pertama-tama *user* memilih MapID yang akan ditentukan batas-batasnya pada combobox MapID. Setelah MapID dipilih, *user* menekan button OK, maka akan muncul message dialog seperti terlihat pada gambar 4.16.



Gambar 4.16. Pesan untuk Menentukan Batas Atas Peta

Kemudian *user* mengklik titik pada image. Titik yang diklik tersebut akan disimpan sebagai batas lintang atas dan batas bujur kiri dari peta yang akan ditampilkan pada edit Up Latitude dan edit Left Longitude. Setelah itu akan muncul message dialog seperti terlihat pada gambar 4.17.



Gambar 4.17. Pesan untuk Menentukan Batas Bawah Peta

Kemudian *user* mengklik titik pada image. Titik yang diklik tersebut akan disimpan sebagai batas lintang bawah dan batas bujur kanan dari peta yang akan ditampilkan pada edit Down Latitude dan edit Right Longitude. Titik yang dipilih tersebut nilainya harus lebih besar daripada titik pertama. Bila data yang dimasukkan salah, maka akan muncul message dialog seperti terlihat pada gambar 4.18.



Gambar 4.18. Pesan Data Titik Salah

7. Setelah batas-batas peta dimasukkan, untuk menyimpan data-data tersebut ditekan button Save. Maka batas-batas peta tersimpan dalam *database*, dan akan muncul message dialog seperti terlihat pada gambar 4.19.

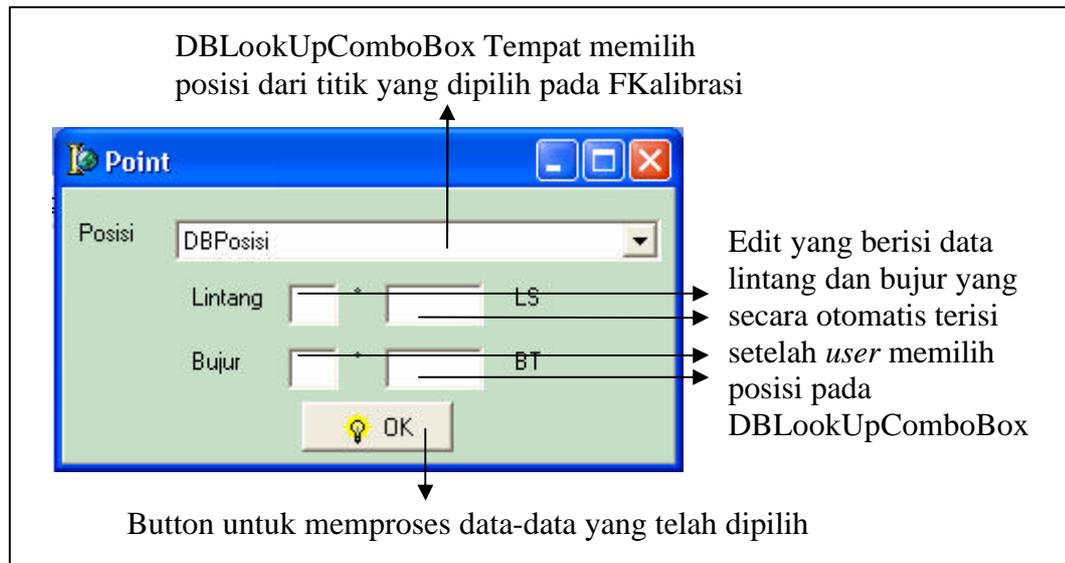


Gambar 4.19. Pesan Batas Peta telah Disimpan

8. Untuk menutup form FKalibrasi, tekan tombol Close.

#### 4.2.5. Form FPoint

Form ini adalah form yang digunakan oleh form FKalibrasi untuk menentukan nama, posisi lintang dan bujur dari titik yang dipilih pada form FKalibrasi. Desain form FPoint beserta penjelasannya dapat dilihat pada gambar 4.20.



Gambar 4.20. Desain dan Penjelasan Form FPoint

Cara kerja dari form FPoint adalah sebagai berikut:

1. Pertama-tama *user* memilih nama posisi dari titik yang telah diklik pada form FKalibrasi pada DBLookupComboBox Posisi.
2. Setelah nama posisi dipilih, maka secara otomatis data lintang dan bujur dari posisi tersebut akan ditampilkan pada edit lintang dan edit bujur.
3. Setelah itu tekan button OK, maka data lintang dan bujur dari posisi tersebut akan digunakan sebagai data lintang dan bujur dari titik yang telah diklik pada form FKalibrasi.

#### 4.2.6. Form FMapping

Form FMapping merupakan form awal yang digunakan untuk memasukkan, mengubah dan menampilkan nama jalan, rute dan juga kotak jalan. Desain dari form FMapping dapat dilihat pada gambar 4.21., sedangkan bagan menu yang terdapat pada form FMapping dapat dilihat pada gambar 4.22.

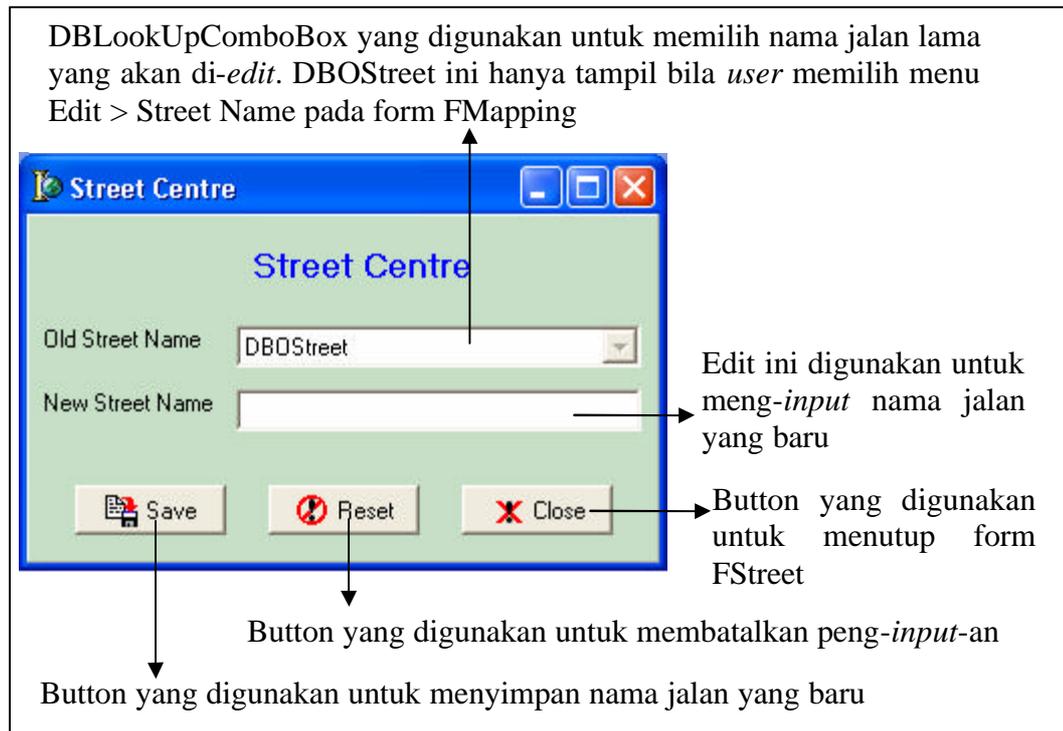


Penjelasan dari form FMapping adalah sebagai berikut:

- Menu Insert merupakan menu untuk memasukkan data yang baru, mempunyai tiga buah sub menu yaitu:
  - Sub menu Street: digunakan untuk memasukkan data nama jalan yang baru.
  - Sub menu Box Street: digunakan untuk memasukkan data kotak jalan yang baru.
  - Sub menu Rute: digunakan untuk memasukkan data rute perjalanan yang baru.
- Menu Edit merupakan menu untuk mengubah data-data, mempunyai tiga buah sub menu yaitu:
  - Sub menu Street Name: digunakan untuk mengubah data nama jalan yang terdapat dalam *database*.
  - Sub menu Delete Box Street: digunakan untuk menghapus kotak jalan yang terdapat dalam *database*.
  - Sub menu Rute: digunakan untuk mengubah rute perjalanan yang terdapat dalam *database*.
- Menu View merupakan menu untuk menampilkan data-data, mempunyai dua sub menu yaitu:
  - Sub menu Box: digunakan untuk menampilkan kotak jalan pada peta. Terdapat dua pilihan yaitu menampilkan semua kotak jalan atau menampilkan kotak jalan berdasarkan nama jalan.
  - Sub menu Street: digunakan untuk menampilkan data-data dari nama jalan yang terdapat dalam *database*.

#### 4.2.7. Form FStreet

Form FStreet merupakan form yang digunakan untuk memasukkan ataupun mengubah nama jalan. Desain dan penjelasan dari form FStreet dapat dilihat pada gambar 4.23.



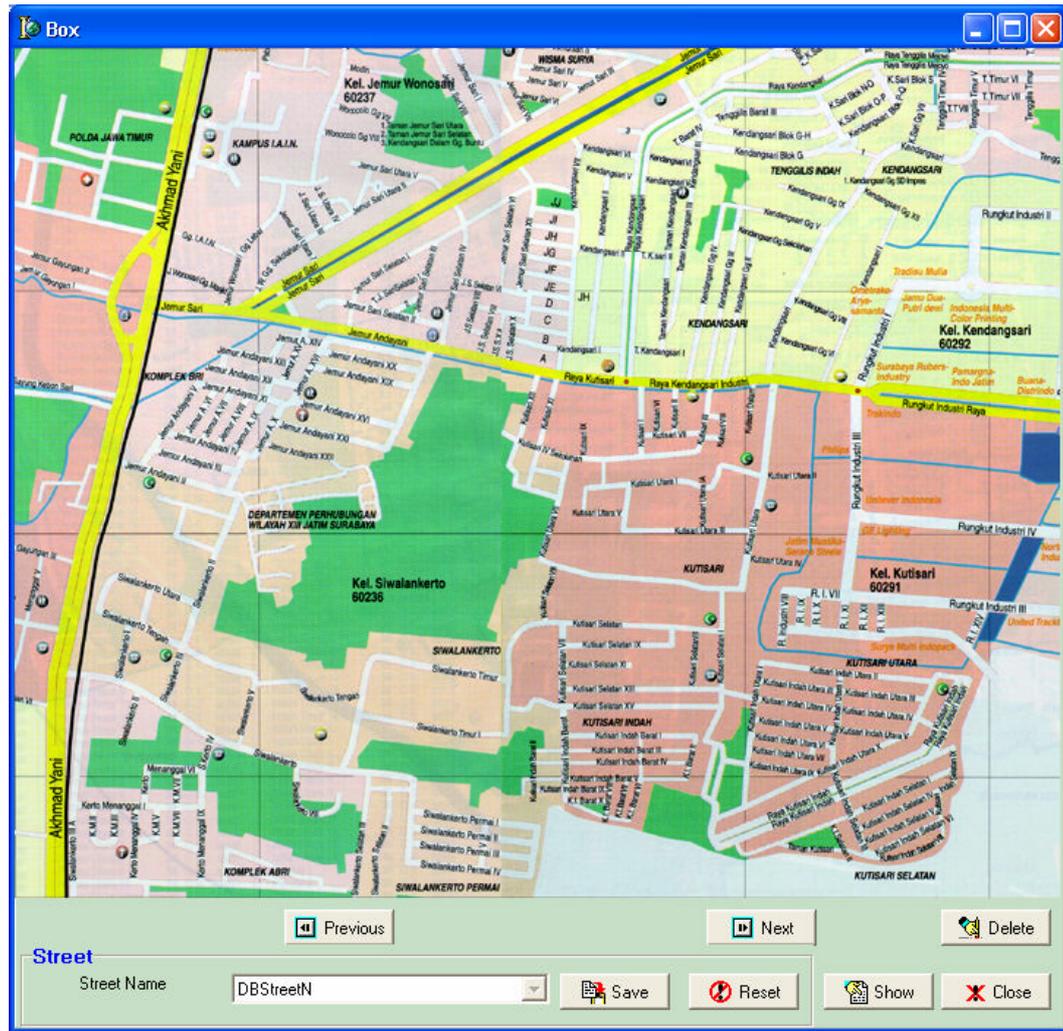
Gambar 4.23. Desain dan Penjelasan Form FStreet

Cara kerja dari form FStreet adalah sebagai berikut:

1. Untuk memasukkan data nama jalan yang baru, *user* meng-*input* nama jalan yang baru tersebut pada edit New Street Name, kemudian setelah itu tekan button Save untuk menyimpannya dalam *database*.
2. Untuk meng-*edit* data nama jalan yang sudah ada dalam *database*, pertama-tama *user* memilih nama jalan yang akan di-*edit* pada DBOStreet, kemudian *user* meng-*input* nama jalan yang baru pada edit New Street Name, lalu tekan button Save untuk mengganti nama jalan yang lama dengan nama jalan yang baru.
3. Untuk menutup form FStreet, tekan button Close.

#### 4.2.8. Form FBox

Form FBox merupakan form yang digunakan untuk memasukkan, mengubah dan menampilkan kotak jalan. Desain dari form FBox dapat dilihat pada gambar 4.24.



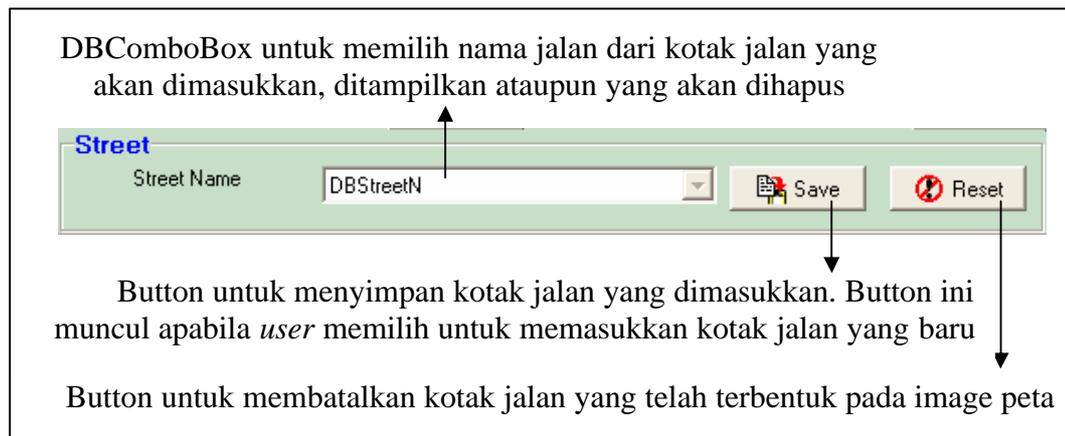
Gambar 4.24. Desain Form FBox

Form FBox terbagi menjadi tiga bagian yaitu:

- Image
  - Image ini menampilkan gambar peta Surabaya.
- Panel Utama
  - Komponen-komponen yang terdapat pada panel utama ini antara lain:
    - Button Next, digunakan untuk menampilkan urutan peta Surabaya selanjutnya yang tersimpan dalam *database*.
    - Button Previous, digunakan untuk menampilkan urutan peta Surabaya sebelumnya yang tersimpan dalam *database*.
    - Button Show, digunakan untuk menampilkan kotak jalan. Button ini muncul apabila *user* memilih untuk menampilkan kotak jalan yang ada.

- Button Delete, digunakan untuk menghapus kotak jalan. Button ini muncul apabila *user* memilih untuk menghapus kotak jalan yang ada.
- Button Close, digunakan untuk menutup form FBox.
- Panel Street
 

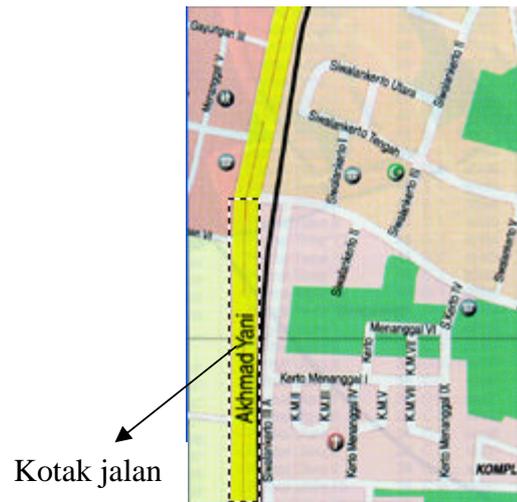
Di dalam Panel Street terdapat beberapa komponen yang berhubungan dengan jalan yang terdapat pada peta seperti terlihat pada gambar 4.25.



Gambar 4.25. Desain dan Penjelasan Panel Street Form FBox

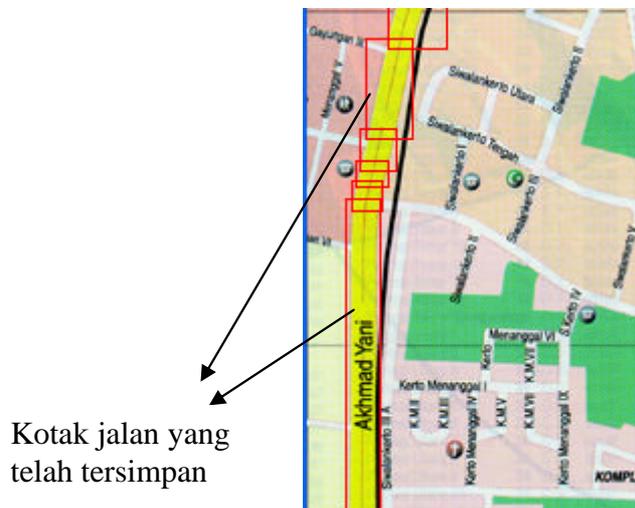
Cara kerja dari form FBox untuk memasukkan kotak jalan yang baru adalah sebagai berikut:

1. Pertama-tama *user* memilih nama jalan dari kotak jalan yang akan dimasukkan pada DBStreetN.
2. Kemudian *user* mulai membuat kotak jalan pada image peta yaitu dengan cara mengarahkan *mouse* pada image peta, tekan *left mouse* pada titik yang merupakan batas pertama dari kotak jalan, tahan *left mouse* sampai titik batas kedua dari kotak jalan kemudian baru lepas *left mouse*, maka kotak jalan akan terbentuk, seperti terlihat pada gambar 4.26.



Gambar 4.26. Kotak Jalan telah Terbentuk pada Image Peta

3. Kemudian setelah kotak jalan terbentuk pada image peta, tekan button Save untuk menyimpan data-data dari kotak jalan tersebut. Bila kotak jalan berhasil tersimpan dalam *database*, kotak jalan tersebut akan berubah warna menjadi merah, seperti terlihat pada gambar 4.27.



Gambar 4.27. Kotak Jalan telah Tersimpan dalam *Database*

Cara kerja dari form FBox untuk menghapus kotak jalan yang ada adalah sebagai berikut:

1. Pertama-tama *user* memilih nama jalan yang akan dihapus pada DBStreetN, kemudian tekan button Show. Maka kotak-kotak jalan dari nama jalan pada DBStreetN akan ditampilkan.
2. Setelah itu, klik daerah kotak jalan yang akan dihapus pada image peta dan tekan button Delete. Maka kotak jalan yang dipilih tersebut dihapus dari *database*.

Untuk menampilkan kotak jalan ada dua macam yaitu menampilkan semua kotak jalan yang ada atau menampilkan kotak jalan dari jalan tertentu. Untuk menampilkan semua kotak jalan, *user* memilih menu View > Box > All pada form FMapping, maka kotak-kotak jalan yang terdapat dalam *database* akan ditampilkan semuanya pada image peta. Sedangkan untuk menampilkan kotak jalan dari jalan tertentu caranya adalah dengan memilih nama jalan pada DBStreetN yang kotak jalannya ingin ditampilkan, kemudian tekan button Show. Maka kotak jalan dari jalan tersebut akan ditampilkan pada image peta.

#### 4.2.9. Form FRute

Form FRute merupakan form yang digunakan untuk memasukkan dan mengubah rute perjalanan. Desain form FRute dapat dilihat pada gambar 4.28.

Gambar 4.28. Desain Form FRute

Form FRute terbagi menjadi enam bagian yaitu:

- Panel Trip

Di dalam Panel Trip terdapat beberapa komponen yang berhubungan dengan perjalanan mobil seperti terlihat pada gambar 4.29.

DBCombobox untuk memilih nama jalan yang merupakan asal dari perjalanan

DBCombobox untuk memilih nama jalan yang merupakan tujuan dari perjalanan

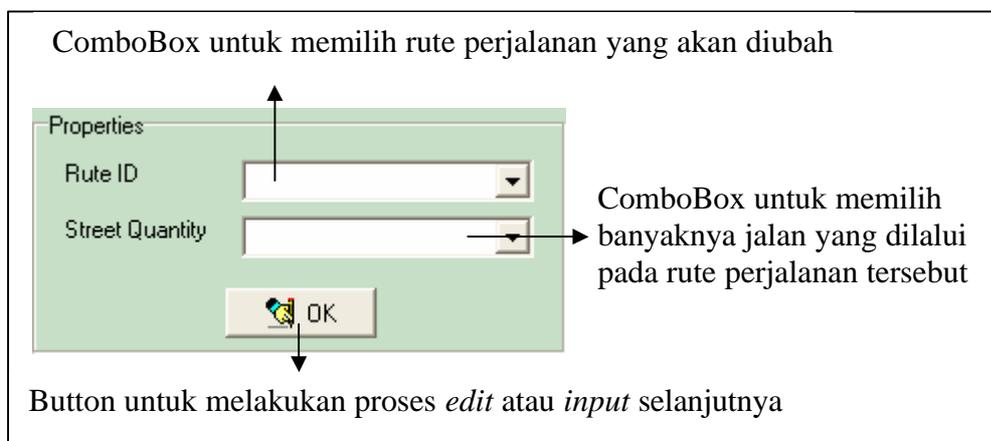
Button untuk mencari apakah perjalanan dengan asal DBSource dan tujuan DBDest rute perjalanannya telah ada pada *database*

Gambar 4.29. Desain dan Penjelasan Panel Trip Form FRute

Hasil pencarian tersebut kemudian ditampilkan pada memo Rute. Bila rute perjalanan ada maka semua rute perjalanan dari perjalanan tersebut akan ditampilkan pada memo Rute, tetapi bila tidak ada maka pada memo Rute tertulis “No Trip Found”.

- Panel Properties

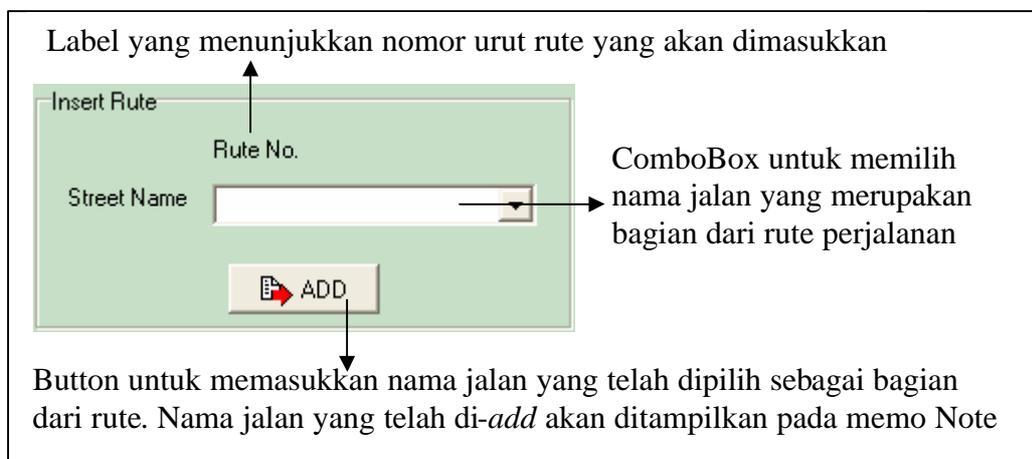
Di dalam Panel Properties terdapat beberapa komponen yang berhubungan dengan *property* dari perjalanan mobil yang dipilih pada panel Trip seperti terlihat pada gambar 4.30.



Gambar 4.30. Desain dan Penjelasan Panel Properties Form FRute

- Panel Insert Rute

Beberapa komponen yang terdapat pada panel Insert Rute dapat dilihat pada gambar 4.31.



Gambar 4.31. Desain dan Penjelasan Panel Insert Rute Form FRute

- Panel Utama

Pada panel Utama terdapat beberapa komponen yang berhubungan dengan semua bagian pada form FRute, yaitu:

- Button Save, digunakan untuk menyimpan rute perjalanan yang telah *edit* atau *di-input* oleh *user*.
- Button Reset, digunakan untuk membatalkan rute perjalanan yang telah *edit* atau *di-input* oleh *user*.
- Button Close, digunakan untuk menutup form FRute.

Berikut ini langkah-langkah untuk memasukkan rute perjalanan yang baru:

1. *User* memilih asal dan tujuan dari perjalanan pada DBSource dan DBDest yang terdapat pada panel Trip, kemudian tekan button Search. Maka sistem akan melakukan pencarian pada *database* apakah rute perjalanan tersebut sudah ada atau tidak dan menampilkannya pada memo Rute.
2. Setelah itu *user* memasukkan jumlah jalan pada combobox StreetQuantity di panel Properties yang terdapat pada rute perjalanan yang baru, lalu tekan button OK.
3. Setelah jumlah jalan ditentukan, *user* memilih nama jalan yang merupakan bagian dari rute perjalanan pada ComboBox Street Name di panel Insert Rute, lalu tekan button Add. Misalnya perjalanan dari jalan Jemur Sari menuju ke jalan Raya Darmo, jumlah jalannya ada empat yaitu jalan Jemur Sari – Ahmad Yani – Raya Wonokromo – Raya Darmo.
4. Kemudian setelah semua jalan dimasukkan, tekan button Save untuk menyimpan rute perjalanan baru tersebut.

Langkah-langkah untuk mengubah rute perjalanan hampir sama dengan langkah-langkah untuk memasukkan rute perjalanan yang baru, bedanya hanyalah sebelum menekan button OK, *user* terlebih dahulu memilih RuteID yang akan diubah pada combobox Rute ID. Langkah-langkah selanjutnya sama dengan memasukkan rute perjalanan baru.

#### 4.2.10. Form FView

Form FView merupakan form yang digunakan untuk menampilkan data-data lokasi yang telah dilalui mobil berdasarkan tanggal. Selain itu form FView juga dapat digunakan untuk menampilkan data-data jalan yang tersimpan dalam *database*. Desain dari form FView dapat dilihat pada gambar 4.32.



Gambar 4.32. Desain Form FView

Komponen-komponen yang terdapat pada form FView adalah sebagai berikut:

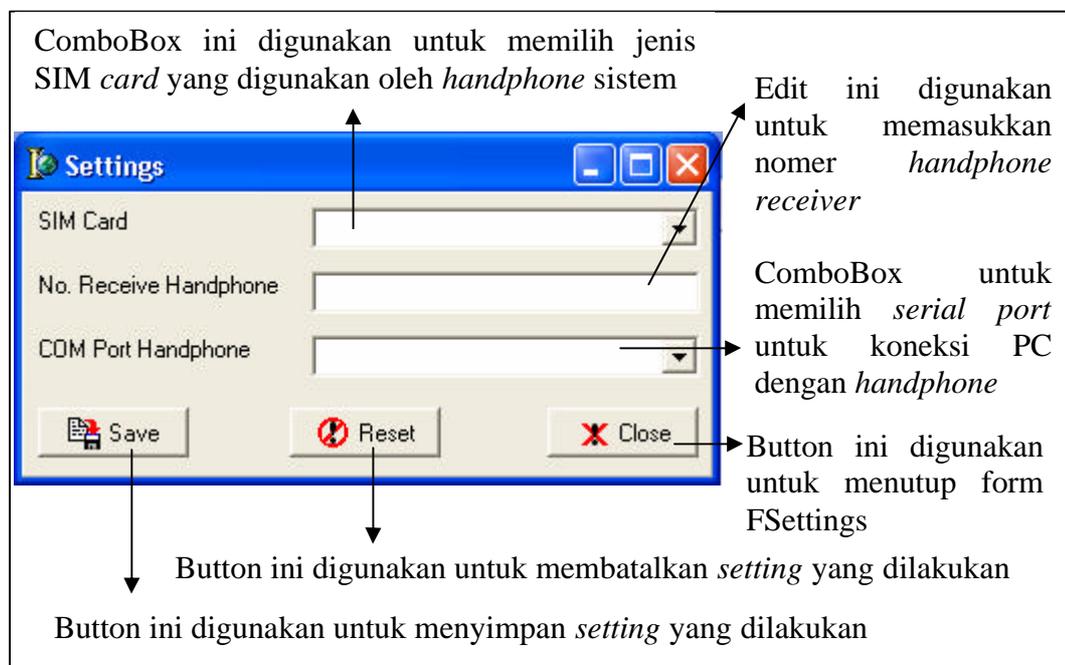
- DateTimePicker, digunakan untuk memilih tanggal tertentu dari rute perjalanan mobil yang ingin ditampilkan. Komponen DateTimePicker ini tidak akan tampil apabila *user* memilih untuk menampilkan data nama jalan.
- Button Show, digunakan untuk memproses data rute perjalanan mobil dengan tanggal tertentu yang akan ditampilkan. Sama dengan komponen DateTimePicker, button Show tidak akan tampil apabila *user* memilih untuk menampilkan data nama jalan.
- DBGrid, digunakan untuk menampilkan data-data baik data posisi mobil sesuai dengan tanggal yang dipilih pada DateTimePicker, maupun data nama jalan.
- Button Close, digunakan untuk menutup form FView.

Berikut ini langkah-langkah untuk menampilkan data rute perjalanan yang telah dilalui oleh mobil:

1. Pertama-tama *user* memilih tanggal dari perjalanan mobil yang ingin ditampilkan pada DateTimePicker.
2. Kemudian tekan button Show, maka data-data rute perjalanan mobil dengan tanggal DateTimePicker tersebut ditampilkan pada DBGrid.

#### 4.2.11. Form FSettings

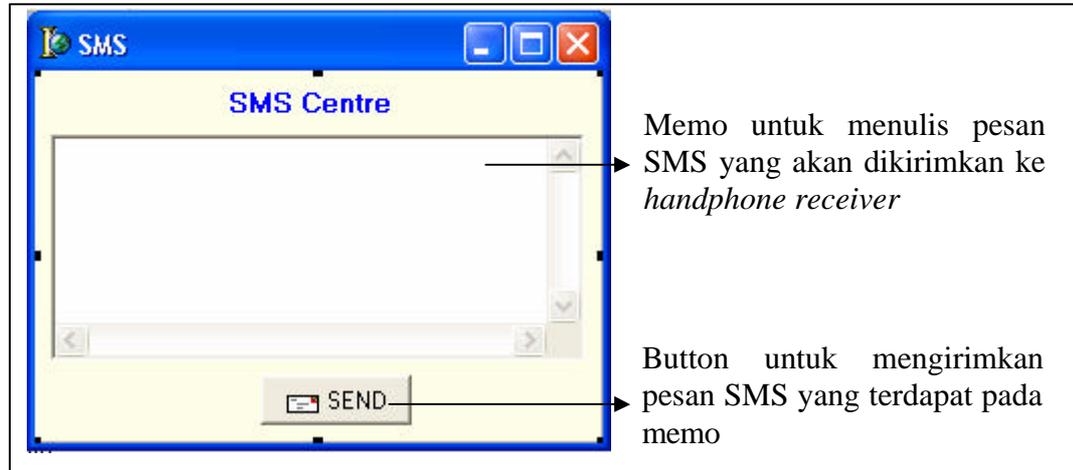
Form FSettings merupakan form yang digunakan untuk mengatur SIM card, nomer *handphone* penerima dan juga *serial port* dimana sistem dihubungkan dengan PC. Desain dan juga penjelasan dari form FSettings dapat dilihat pada gambar 4.33.



Gambar 4.33. Desain dan Penjelasan Form FSettings

#### 4.2.12. Form FSMS

Form FSMS merupakan form yang digunakan untuk mengirimkan pesan SMS kepada *user* diluar mobil. Desain form dan juga penjelasan dari form FSMS dapat dilihat pada gambar 4.34.



Gambar 4.34. Desain dan Penjelasan Form FSMS

Cara mengirimkan pesan SMS kepada *handphone receiver* adalah sebagai berikut:

1. Tulis pesan SMS yang ingin dikirimkan kepada *handphone receiver* pada memo.
2. Tekan button Send, maka pesan SMS pada memo dikirimkan kepada *handphone receiver*. Dalam hal ini nomer *handphone receiver* adalah nomer *handphone* yang di-*set* oleh *user* pada form FSettings.

#### 4.2.13. Form FCommunication

Form FCommunication merupakan program utama dari *software* ini dimana dalam form ini data lintang dan bujur posisi mobil yang diperoleh dari data GPS *receiver* diproses sehingga dapat menentukan secara pasti posisi mobil pada peta dan juga dapat menentukan nama jalan dari posisi mobil saat itu. Selain itu posisi mobil tersebut juga dapat dikirimkan kepada *user* yang berada di luar mobil sehingga *user* tersebut dapat terus memantau mobil. Desain dari form FCommunication dapat dilihat pada gambar 4.35.



Gambar 4.35. Desain Form FCommunication

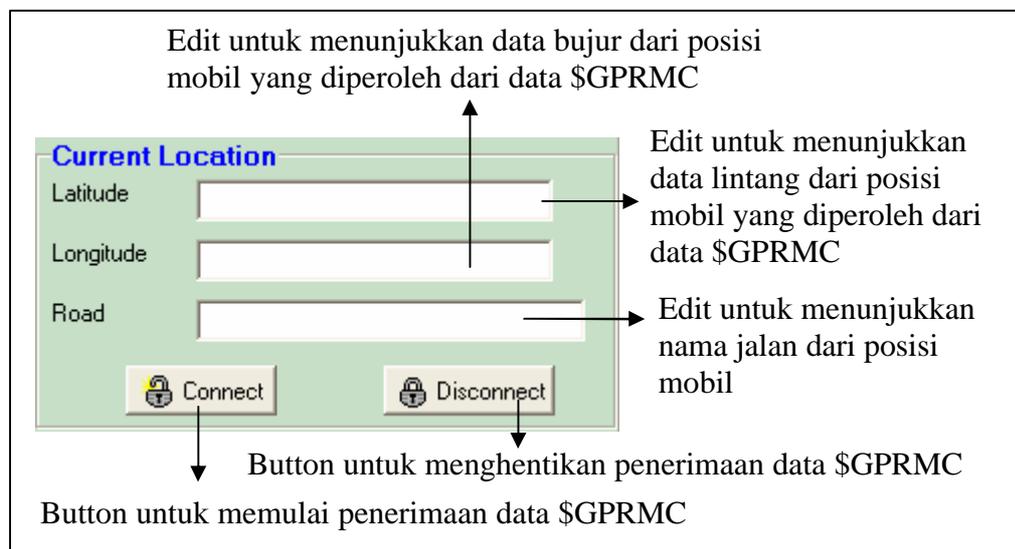
Form FCommunication terdiri dari lima bagian yaitu:

- Image  
Image peta digunakan untuk menunjukkan posisi mobil.
- Panel Utama  
Panel Utama mempunyai beberapa komponen sebagai berikut:
  - Edit Date, menunjukkan tanggal saat itu yang diperoleh dari data \$GPRMC.
  - Edit Time, menunjukkan waktu saat itu yang diperoleh dari data \$GPRMC.
  - Button SMS, digunakan untuk menampilkan form FSMS apabila akan mengirimkan SMS kepada *handphone receiver*.
  - Button Close, digunakan untuk menutup form FCommunication.
  - MediaPlayer1, digunakan untuk menyalakan alarm apabila *handphone receiver* mengirimkan SMS dengan format “ALRM”.
  - Panel StatusBar, digunakan untuk menunjukkan kegiatan yang dilakukan oleh sistem yang berhubungan dengan *handphone* sistem, misalnya bila koneksi antara Delphi dengan *handphone* sistem berhasil dilakukan, maka panel StatusBar akan menunjukkan tulisan “Connected” atau bila sistem

mengirimkan SMS kepada *handphone receiver*, maka panel StatusBar akan menunjukkan tulisan “Send SMS <kataSMS>”.

- Komponen TimerSend, digunakan untuk mengatur waktu pengirim SMS berisi informasi data lokasi mobil, dalam hal ini waktu dari TimerSend ditentukan oleh *handphone receiver*.
  - Komponen Timer1 digunakan untuk mengatur waktu penerimaan data GPS, dalam hal ini TimerGPS di-*set* setiap setengah detik sekali merima data GPS.
  - Komponen SMS, digunakan sebagai penghubung antara Delphi dengan *handphone* Nokia untuk mengirim dan menerima SMS.
  - Komponen MsCommGPS, digunakan sebagai penghubung antara Delphi dengan GPS *receiver*.
  - Komponen Shape1, digunakan sebagai *icon* penunjuk posisi mobil pada image peta.
- Panel Current Location

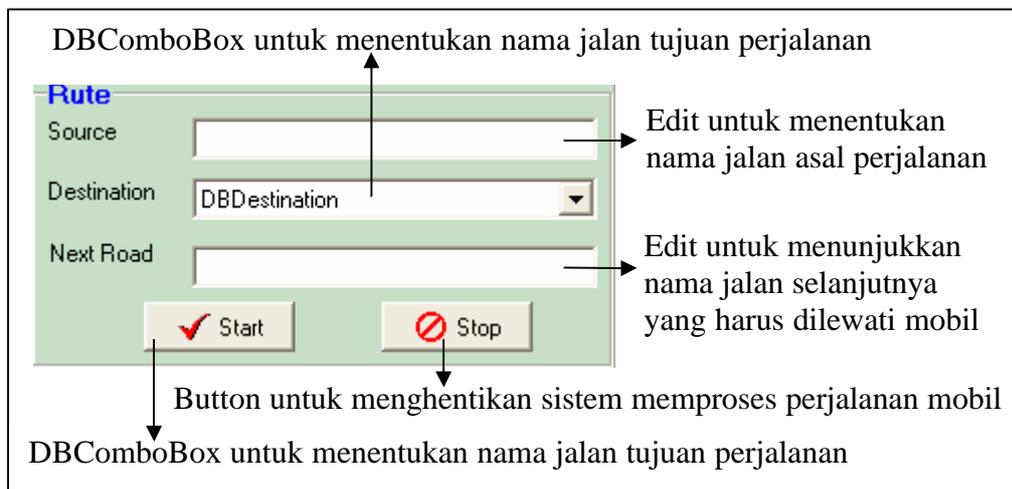
Panel Current Location terdiri dari beberapa komponen seperti terlihat pada gambar 4.36.



Gambar 4.36. Desain dan Penjelasan Panel Current Location Form  
FCommunication

- Panel Rute

Panel Rute terdiri dari beberapa komponen seperti terlihat pada gambar 4.37.



Gambar 4.37. Desain dan Penjelasan Panel Rute Form FCommunication

Button stop hanya dapat digunakan apabila mobil berada pada jalan yang merupakan tujuan perjalanan.

- Panel Status

Panel Status berisi memo yang menunjukkan informasi mengenai posisi mobil saat itu apakah mobil di jalan yang sesuai atau tidak dengan rute perjalanan yang ditentukan.

Cara kerja dari form FCommunication adalah sebagai berikut:

1. Pertama-tama untuk mulai menerima data GPS, *user* menekan button Connect, maka data \$GPRMC diproses oleh sistem menjadi data tanggal, jam, lintang, bujur, dan nama jalan yang masing-masing ditunjukkan pada edit Date, Time, Latitude, Longitude dan Road pada panel Current Location. Sedangkan posisi mobil pada peta ditunjukkan oleh komponen Shape1.
2. Kemudian untuk memulai perjalanan mobil, *user* memilih tujuan dari perjalanan tersebut pada DBDestination di panel Rute. *User* tidak perlu memasukkan asal dari perjalanan karena secara otomatis sistem akan menentukan asal perjalanan mobil berdasarkan posisi mobil saat itu.
3. Setelah *user* memilih tujuan perjalanan ditekan button Start, sistem akan mulai memproses perjalanan mobil dan mengirimkan SMS mengenai asal dan tujuan

perjalanan serta rute yang digunakan oleh sistem kepada *handphone receiver*. *Handphone receiver* dapat juga menentukan rute perjalanan yang digunakan oleh sistem dengan cara mengirimkan SMS *request* dengan format “RUTE”.

4. Kemudian setiap mobil berpindah jalan sistem akan mengirimkan SMS berisi informasi mengenai nama jalan dari posisi mobil saat itu dan juga nama jalan selanjutnya yang harus ditempuh oleh mobil. *Handphone receiver* dapat juga mengatur waktu pengiriman SMS dengan cara mengirimkan SMS *request* dengan format “SEND”. Misal, *handphone receiver* menginginkan agar sistem mengirimkan informasi posisi mobil setiap sepuluh menit sekali, maka SMS *request* yang dikirim oleh *handphone receiver* adalah “SEND 10 MNT”. Jika *handphone receiver* ingin menghentikan pengiriman SMS informasi posisi mobil, maka dikirimkan SMS *request* dengan format “STOP”.
5. *Handphone receiver* juga bisa mengaktifkan alarm, dengan cara mengirimkan SMS *request* dengan format “ALRM”, maka sistem akan mengaktifkan komponen MediaPlayer.
6. *User* di dalam mobil juga dapat mengirimkan SMS kepada *handphone receiver* dengan cara menekan button SMS. Form FSMS akan muncul dan *user* tinggal memasukkan kata-kata yang ingin dikirimkan.
7. Agar sistem berhenti menerima data \$GPRMC, ditekan button Disconnect.

Untuk mendapatkan data lintang dan bujur, mula-mula satelit mengirimkan data posisi yang kemudian diterima oleh GPS *receiver*. Data tersebut di-*transfer* oleh GPS *receiver* kepada sistem. Dan setelah data diterima, sistem men-*filter*-nya untuk mendapatkan data dengan format \$GPRMC. Sebagai contoh data GPS dengan format \$GPRMC adalah sebagai berikut:

```
$GPRMC,044935,A,0720.3983,S,11243.8911,E,000.0,000.0,050603,000.9,E*71
```

Dari data \$GPRMC diatas sistem akan memprosesnya sebagai berikut:

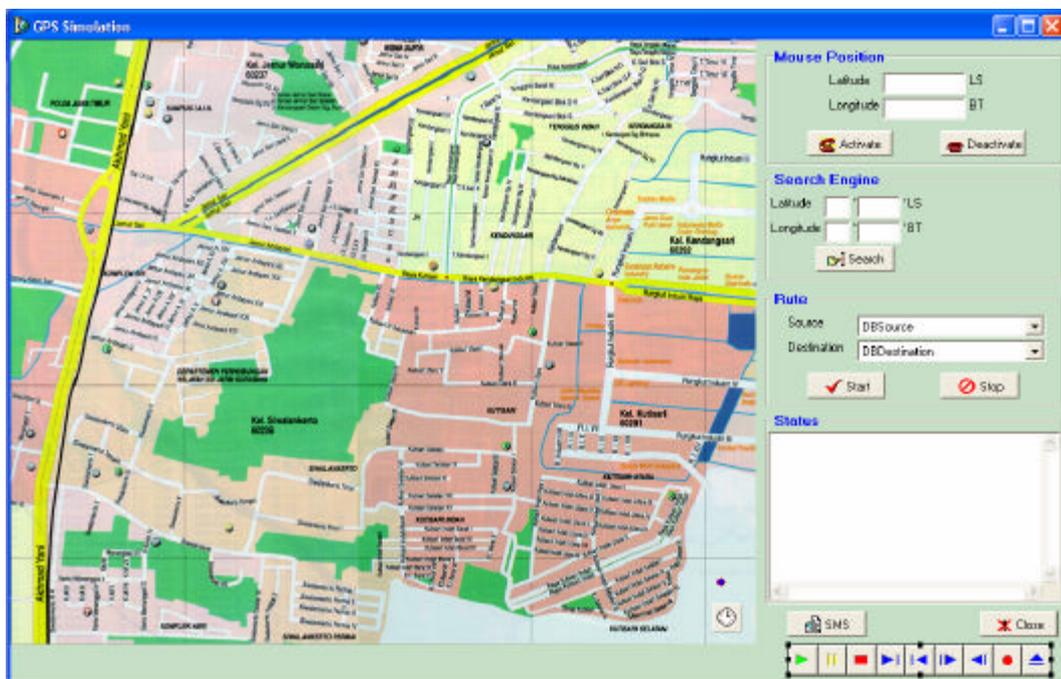
1. Field pertama menunjukkan jam dengan satuan atom. 044935 menunjukkan bahwa saat itu waktu menunjukkan pukul 04:49:35.
2. Field kedua menunjukkan apakah data posisi yang diterima oleh GPS *receiver* valid atau tidak. Jika data yang diterima tidak valid, maka sistem tidak akan memprosesnya. A menunjukkan bahwa data yang diterima valid.

3. Field ketiga dan keempat menunjukkan posisi dan arah lintang. 0720.3983 menunjukkan bahwa saat itu mobil berada pada posisi lintang  $07^{\circ}20.3983'$  dan S menunjukkan arah lintang berada di Lintang Selatan.
4. Field kelima dan keenam menunjukkan posisi dan arah bujur. 11243.8911 menunjukkan bahwa saat itu mobil berada pada posisi bujur  $112^{\circ}43.8911'$  dan E menunjukkan arah bujur berada di Bujur Timur.

Kemudian setelah diketahui posisi lintang dan bujurnya, sistem memprosesnya untuk mendapatkan nama jalan dari posisi mobil saat ini berada. Dan setelah sistem memprosesnya diperoleh data bahwa saat ini mobil berada di jalan Ahmad Yani.

#### 4.2.14. Form FSimulation

Form FSimulation merupakan simulasi dari form FCommunication, dimana seakan-akan mobil melakukan perjalanan. Fungsi dan kegunaannya hampir sama dengan form FCommunication. Bedanya, di form FSimulation mobil dijalankan dengan *mouse* dengan cara mengklik titik-titik pada image peta. Desain dari form FSimulation dapat dilihat pada gambar 4.38.



Gambar 4.38. Desain Form FSimulation

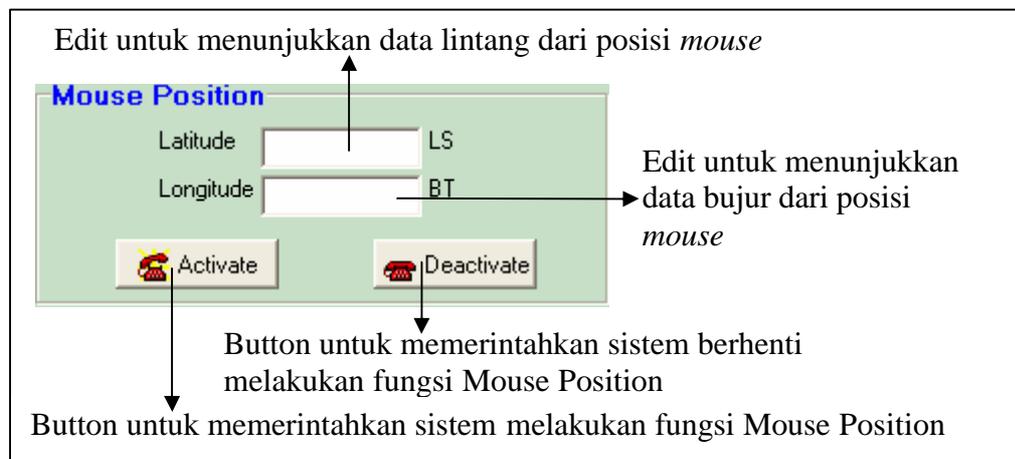
Form FSimulation terbagi menjadi enam bagian yaitu:

- Image peta Surabaya yang digunakan untuk menentukan posisi mobil.
- Panel Utama

Panel Utama mempunyai beberapa komponen sebagai berikut:

- Button SMS, digunakan untuk menampilkan form FSMS apabila akan memberikan SMS *request* kepada sistem.
  - Button Close, digunakan untuk menutup form FCommunication.
  - MediaPlayer1, digunakan untuk menyalakan alarm.
  - Komponen Timer1, digunakan untuk mengatur waktu pengiriman SMS simulasi berisi informasi data lokasi mobil.
  - Komponen Shape1, digunakan sebagai *icon* penunjuk titik yang dicari oleh *user* (fungsi *search engine*) pada image peta.
- Panel Mouse Position

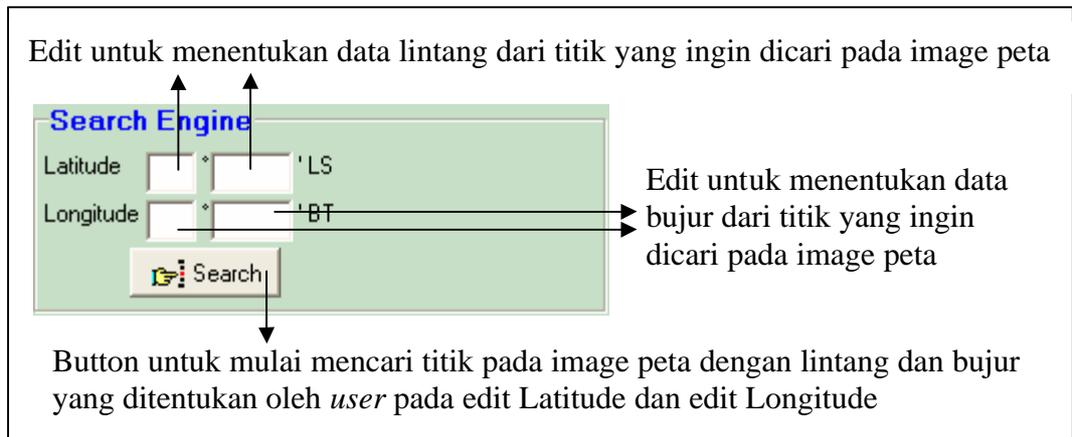
Beberapa komponen yang terdapat pada panel Mouse Position dapat dilihat pada gambar 4.39.



Gambar 4.39. Desain dan Penjelasan Panel Mouse Position Form FSimulation

- Panel Search Engine

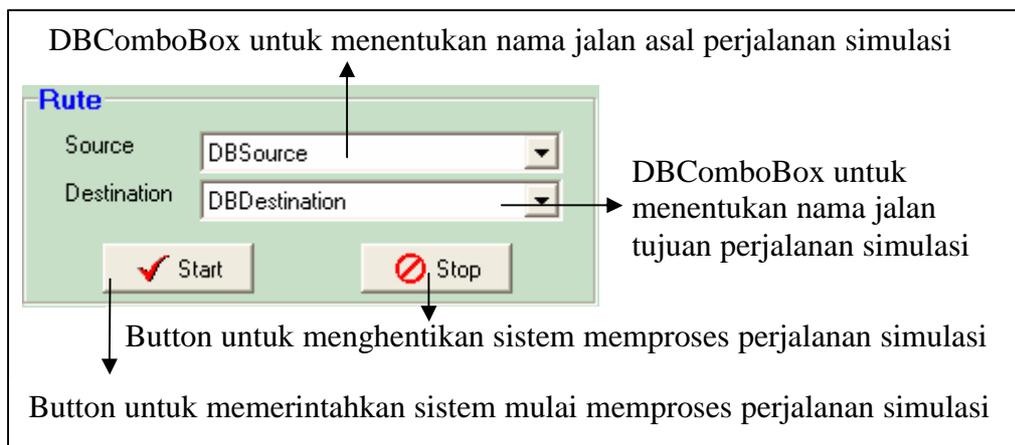
Panel Search Engine memiliki beberapa komponen seperti terlihat pada gambar 4.40.



Gambar 4.40. Desain dan Penjelasan Panel Search Engine Form FSimulation

- Panel Rute

Komponen-komponen yang terdapat pada panel Rute dapat dilihat pada gambar 4.41.



Gambar 4.41. Desain dan Penjelasan Panel Rute Form FSimulation

Button stop hanya dapat digunakan apabila mobil berada pada jalan yang merupakan tujuan perjalanan.

- Panel Status

Panel Status ini berisi memo yang menunjukkan informasi mengenai posisi mobil dari perjalanan simulasi apakah mobil di jalan yang sesuai atau tidak dengan rute perjalanan yang ditentukan.

Cara kerja dari form FSimulationa adalah sebagai berikut:

1. Untuk mengetahui data lintang dan bujur dari titik yang ditunjuk oleh *mouse* pada image peta, caranya dengan menekan button Activate kemudian mengarahkan *mouse* ke titik tertentu pada image peta. Maka data lintang dan bujur dari titik tersebut ditunjukkan pada edit Latitude dan edit Longitude pada panel Mouse Position.
2. Untuk mencari titik pada image peta yang diketahui data lintang dan bujurnya, caranya dengan mengisi data lintang dan bujur pada edit Latitude dan edit Longitude pada panel Search Engine, kemudian tekan button Search. Maka titik yang dicari tersebut akan ditunjukkan oleh komponen Shape1. Fungsi *search engine* ini hanya bisa digunakan apabila fungsi *mouse position* dalam keadaan tidak aktif (dengan cara menekan button Deactivate).
3. Untuk mulai perjalanan mobil simulasi, pertama-tama *user* memilih asal dan tujuan dari perjalanan simulasi, kemudian tekan button Start. Maka sistem akan mulai memproses perjalanan simulasi. Setelah itu *user* mulai melakukan perjalanan simulasi dengan mengklik titik-titik pada image peta seakan-akan mobil sedang melakukan perjalanan. Fungsinya hampir sama dengan form FCommunication. Agar sistem berhenti memproses perjalanan simulasi, ditekan button Stop.
4. Untuk memberikan SMS *request* pada sistem, *user* dapat menekan button SMS, maka akan muncul form FSMS. Pada memo yang tersedia *user* tinggal memasukkan SMS *request*, dan sistem akan meresponnya.
5. Untuk menutup form FSimulation, ditekan button Close.