

BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak sebelumnya. Pembahasan terdiri dari penjelasan tentang spesifikasi lingkungan (spesifikasi perangkat keras dan perangkat lunak) di mana sistem diimplementasikan, implementasi basis data, implementasi algoritma dan implementasi antarmuka aplikasi.

5.1 Spesifikasi Sistem

Implementasi didasarkan atas hasil analisis kebutuhan dan perancangan yang diuraikan pada Bab 4. Implementasi ditujukan agar sistem ini bisa berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan untuk implementasi aplikasi ini ditunjukkan pada tabel 5.1 berikut.

Tabel 5.1 Spesifikasi komputer untuk implementasi

Spesifikasi Komputer	
Notebook	Toshiba Portege M300
Prosesor	Intel® Pentium M® Processor 1.10 GHz
Memori (RAM)	512 MB
Hardisk	IOMEGA kapasitas 60 GB
VGA Card	Intel® Extreme Graphics

Sumber: Implementasi

5.1.2 Spesifikasi Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak yang digunakan untuk implementasi aplikasi ini ditunjukkan pada tabel 5.2 berikut.

Tabel 5.2 Spesifikasi perangkat lunak untuk implementasi

Spesifikasi Perangkat Lunak	
Sistem Operasi	Windows XP Professional Service Pack 2
Bahasa Pemrograman	Visual Basic

Lingkungan Pemrograman	Visual Basic 6.0 Enterprise Edition ESRI MapObjects Version 2.4 MySQL 5.1.22-rc-community
IDE (<i>Integrated Development Environment</i>)	<i>Multiple Document Interface</i> (MDI)

Sumber: Implementasi

5.2 Implementasi Basis Data MySQL

Aplikasi ini dirancang untuk dapat terhubung ke basis data MySQL. Implementasi perancangan basis data skripsi dilakukan sesuai dengan *Entity Relationship Diagram*. Implementasi perancangan basis data menggunakan *query* SQL. *Query* SQL digunakan untuk mengimplementasikan rancangan basis data ke dalam sistem basis data MySQL.

Query SQL yang digunakan dalam membentuk basis data skripsi ditunjukkan pada gambar 5.1 berikut.

```
CREATE DATABASE skripsi;
```

Gambar 5.1 *Query* untuk membuat basis data skripsi

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel simpang ditunjukkan pada gambar 5.2 berikut.

```
CREATE TABLE simpang (
    nama_simpang VARCHAR(50) NOT NULL,
    jumlah_kaki VARCHAR(15) NOT NULL,
    kondisi VARCHAR(20) NOT NULL,
    keterangan VARCHAR(100) NOT NULL,
    PRIMARY KEY(nama_simpang)
);
```

Gambar 5.2 *Query* untuk membuat tabel simpang

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel waktu_siklus ditunjukkan pada gambar 5.3 berikut.

```
CREATE TABLE waktu_siklus (
    id_waktu INT(10) NOT NULL AUTO_INCREMENT,
    waktu VARCHAR(10) NOT NULL,
    siklus_merah VARCHAR(10) NOT NULL,
    siklus_kuning VARCHAR(10) NOT NULL,
    siklus_hijau VARCHAR(10) NOT NULL,
    nama_simpang varchar(50) NOT NULL,
    PRIMARY KEY(id_waktu),
    FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

Gambar 5.3 *Query* untuk membuat tabel waktu_siklus

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel `kaki_simpang` ditunjukkan pada gambar 5.4 berikut.

```
CREATE TABLE kaki_simpang(
  id_kaki_simpang INT(10) NOT NULL AUTO_INCREMENT,
  kode_pendekat VARCHAR(15) NOT NULL,
  jalan VARCHAR(25) NOT NULL,
  nama_simpang VARCHAR(50) NOT NULL,
  PRIMARY KEY(id_kaki_simpang),
  FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

Gambar 5.4 *Query* untuk membuat tabel `kaki_simpang`

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel `jam_puncak` ditunjukkan pada gambar 5.5 berikut.

```
CREATE TABLE jam_puncak(
  id_jam_puncak INT(10) NOT NULL AUTO_INCREMENT,
  jam_puncak VARCHAR(10) NOT NULL,
  kode_pendekat VARCHAR(15) NOT NULL,
  volume VARCHAR(10) NOT NULL,
  fase VARCHAR(10) NOT NULL,
  waktu_hijau VARCHAR(10) NOT NULL,
  kapasitas VARCHAR(10) NOT NULL,
  panjang_antrian VARCHAR(10) NOT NULL,
  derajat_kejenuhan VARCHAR(10) NOT NULL,
  tingkat_pelayanan VARCHAR(10) NOT NULL,
  nama_simpang VARCHAR(50) NOT NULL,
  PRIMARY KEY(id_jam_puncak),
  FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

Gambar 5.5 *Query* untuk membuat tabel `jam_puncak`

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel `lampu` ditunjukkan pada gambar 5.6 berikut.

```
CREATE TABLE lampu(
  id_lampu INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  tipe_lampu VARCHAR(20) NOT NULL,
  merk_lampu VARCHAR(20) NOT NULL,
  jenis_lampu VARCHAR(20) NOT NULL,
  diameter_lampu VARCHAR(10) NOT NULL,
  daya_lampu VARCHAR(10) NOT NULL,
  tegangan_lampu VARCHAR(10) NOT NULL,
  frekuensi_lampu VARCHAR(10) NOT NULL,
  nama_simpang VARCHAR(50) NOT NULL,
  PRIMARY KEY(id_lampu),
  FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

Gambar 5.6 *Query* untuk membuat tabel `lampu`

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel `marka` ditunjukkan pada gambar 5.7 berikut.


```
CREATE TABLE marka(  
    id_marka INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    jenis_marka VARCHAR(20) NOT NULL,  
    nama_simpang VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_marka),  
    FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Gambar 5.7 Query untuk membuat tabel marka
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel tiang ditunjukkan pada gambar 5.8 berikut.

```
CREATE TABLE tiang(  
    id_tiang INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    jenis_tiang VARCHAR(20) NOT NULL,  
    tinggi_tiang VARCHAR(10) NOT NULL,  
    diameter_tiang VARCHAR(10) NOT NULL,  
    nama_simpang VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_tiang),  
    FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Gambar 5.8 Query untuk membuat tabel tiang
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel kabel ditunjukkan pada gambar 5.9 berikut.

```
CREATE TABLE kabel(  
    id_kabel INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    tipe_kabel VARCHAR(20) NOT NULL,  
    jenis_kabel VARCHAR(20) NOT NULL,  
    ukuran_kabel VARCHAR(20) NOT NULL,  
    nama_simpang VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_kabel),  
    FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Gambar 5.9 Query untuk membuat tabel kabel
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel rambu ditunjukkan pada gambar 5.10 berikut.

```
CREATE TABLE rambu(  
    id_rambu INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    jenis_rambu VARCHAR(50) NOT NULL,  
    jumlah_rambu VARCHAR(10) NOT NULL,  
    nama_simpang VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id_rambu),  
    FOREIGN KEY (nama_simpang) REFERENCES simpang (nama_simpang)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Gambar 5.10 Query untuk membuat tabel rambu
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel pengguna ditunjukkan pada gambar 5.11 berikut.

```
CREATE TABLE pengguna(
  username VARCHAR(15) NOT NULL,
  password VARCHAR(10) NOT NULL,
  nama VARCHAR(20) NOT NULL,
  alamat VARCHAR(50) NOT NULL,
  email VARCHAR(20) NOT NULL,
  PRIMARY KEY(id_lampu),
);
```

Gambar 5.11 *Query* untuk membuat tabel pengguna
Sumber: Implementasi

Basis data skripsi yang telah diimplementasikan pada DBMS MySQL ditunjukkan dalam Gambar 5.12.

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.1.22-rc-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| coba      |
| mysql     |
| skripsi   |
| test      |
+-----+
5 rows in set (0.01 sec)

mysql> use skripsi;
Database changed
mysql> show tables;
+-----+
| Tables_in_skripsi |
+-----+
| jam_puncak        |
| kabel             |
| kaki_simpang      |
| lampu             |
| marka             |
| pengguna          |
| rambu             |
| simpang           |
| tiang             |
| waktu_siklus      |
+-----+
10 rows in set (0.24 sec)
```

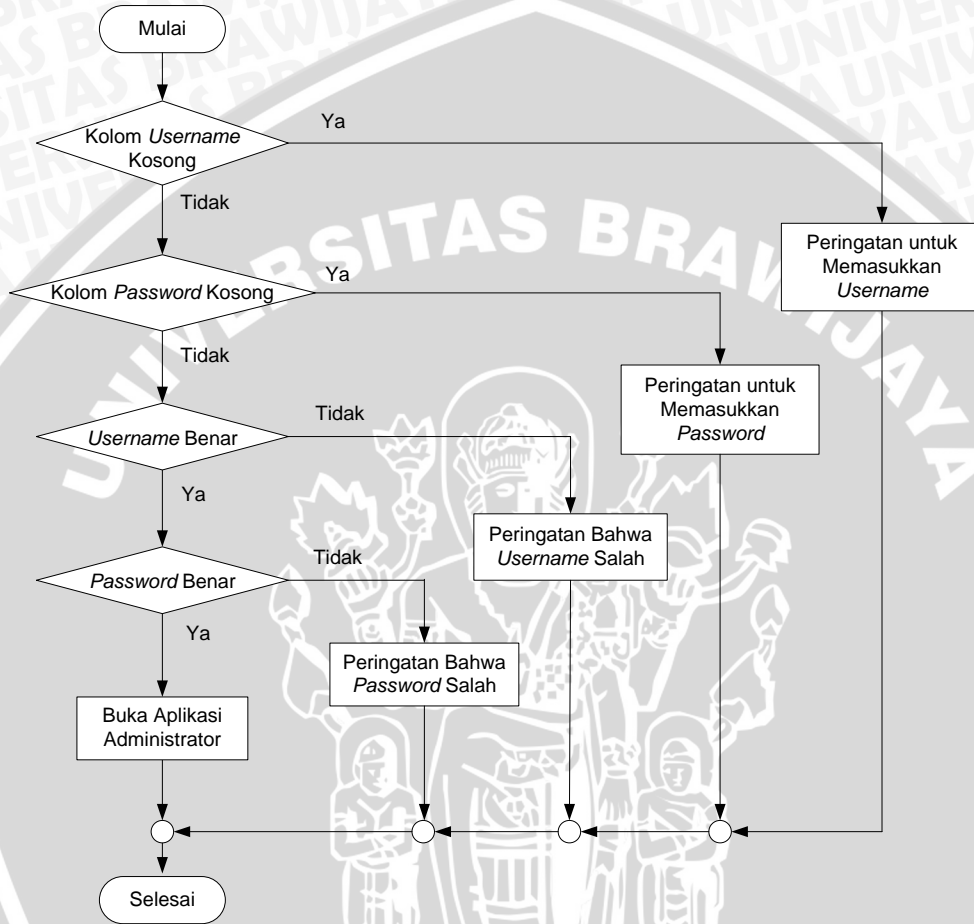
Gambar 5.12 Basis data skripsi yang telah diimplementasikan
Sumber: Implementasi

5.3 Implementasi Algoritma

Implementasi algoritma direpresentasikan dalam 2 tahap. Tahap pertama yaitu *flowchart* (diagram alir). Tahap kedua yaitu pendekatan representasi algoritma dalam format bahasa pemrograman Visual Basic. *Flowchart* adalah urutan aliran algoritma yang digambarkan dalam diagram. Implementasi algoritma yang direpresentasikan adalah implementasi algoritma dari prosedur yang ada pada tiap modul. Berikut beberapa prosedur utama pada sistem ini.

5.3.1 Implementasi Algoritma Login

Algoritma untuk proses validasi *login* dan menampilkan pesan hasil validasi login diimplementasikan pada prosedur `cmdOK_Click()`. Algoritma ini akan dijalankan jika tombol OK pada *form* Login di klik. Diagram alir prosedur `Simpan()` ditunjukkan pada gambar 5.13 berikut.



Gambar 5.13 Flowchart prosedur `cmdOK_Click()`
Sumber: Implementasi

Algoritma prosedur `cmdOK_Click()` yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.1.

```

Private Sub cmdOK_Click()
1   If txtUsername = "" Then
2     MsgBox "Isikan Username !", vbExclamation, "Validasi"
3   Exit Sub
4   End If
5   If txtPassword = "" Then
6     MsgBox "Isikan Password !", vbExclamation, "Validasi"
7     txtPassword.SetFocus
8     Exit Sub
9   End If
  
```




```

10  Katasql = "SELECT username FROM pengguna WHERE " & _
      "username='" & Trim(txtUsername) & "' "
11  TutupRs
12  rs.Open Katasql, Conn, adOpenStatic
13  If rs.EOF = True Or rs.BOF = True Then
14      MsgBox "Maaf Username yang Anda Isikan Salah", vbExclamation,
          "Ada Kesalahan"
15  Else
16      Katasql2 = "SELECT username FROM pengguna WHERE " "username='" & _
          & Trim(txtUsername) & "' And password='" & Trim(txtPassword) & "'"
17      TutupRs2
18      rs2.Open Katasql2, Conn, adOpenStatic
19      If rs2.EOF = True Or rs2.BOF = True Then
20          MsgBox "Maaf Password yang Anda Isikan Salah", vbExclamation,
              "Ada Kesalahan"
21          txtPassword.SetFocus
22      Else
23          Unload Me
24          BukaKoneksi
25          With simbolPencarian
26              .SymbolType = moFillSymbol
27              .Color = moYellow
28              .OutlineColor = moRed
29              .Style = 1
30              .Size = 5
31          End With
32          frmPeta.Show 1
33      End If
34  End If
End Sub

```

Algoritma 5.1 Implementasi algoritma `cmdOK_Click()`

Sumber: Implementasi

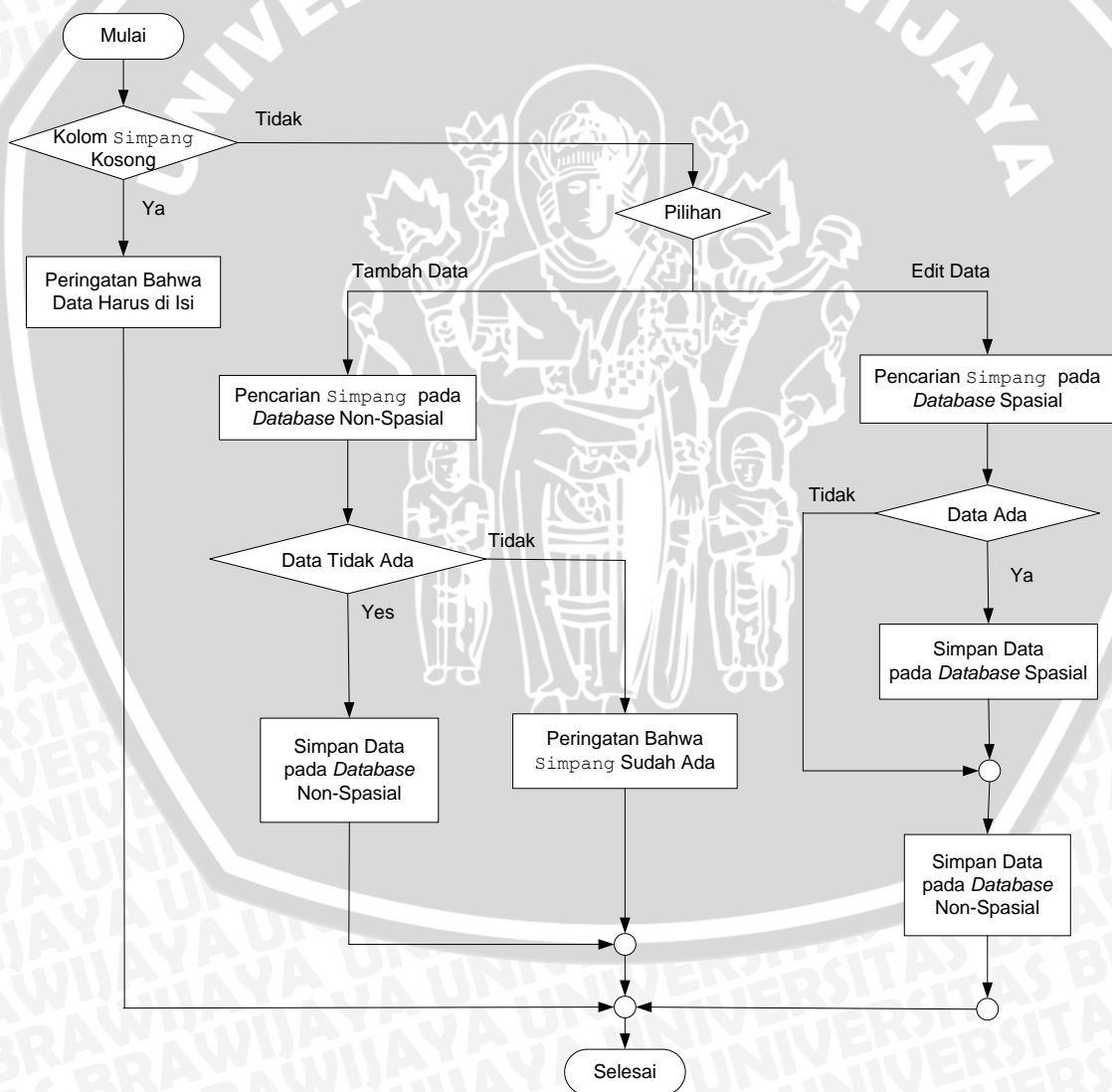
Penjelasan implementasi Algoritma 5.1 yaitu:

1. Baris 1-4, jika kolom `txtUsername` kosong maka akan muncul peringatan untuk memasukkan *username* dan keluar prosedur, jika tidak maka jalankan proses selanjutnya.
2. Baris 5-9, jika kolom `txtPassword` kosong maka akan muncul peringatan untuk memasukkan *password* dan keluar prosedur, jika tidak maka jalankan proses selanjutnya.
3. Baris 10-12, pencarian *username* yang telah diisi pada *database*.
4. Baris 13-14, jika *username* tidak ada pada *database* maka akan muncul peringatan bahwa *username* salah.
5. Baris 15-18, jika *username* ada maka dilakukan pencarian *password* dari *username* tersebut pada *database*.
6. Baris 19-20, jika *password* yang telah diisi tidak sesuai dengan *username*-nya maka akan muncul peringatan bahwa *password* salah.
7. Baris 21, fokus ke kolom *password*.

8. Baris 22-24, jika *password* yang telah diisi sesuai dengan *usernamenya* (valid) maka tampilan *Login* akan tertutup dan prosedur *BukaKoneksi* yang berfungsi untuk membuka *database* dijalankan.
9. Baris 15-21, pemberian nilai pada variabel *simbolPencarian*.
10. Baris 22, *form* aplikasi untuk Administrator ditampilkan.

5.3.2 Implementasi Algoritma Simpan Data Non-Spasial

Algoritma untuk menyimpan data-data non-spasial diimplementasikan pada prosedur *Simpan()*. Prosedur ini dijalankan jika tombol untuk menyimpan data pada *toolbar* non-spasial di klik. Diagram alir prosedur *Simpan()* ditunjukkan pada gambar 5.14 berikut.



Gambar 5.14 Flowchart prosedur *Simpan()*

Sumber: Implementasi

Algoritma prosedur `Simpan()` yang digunakan untuk menyimpan salah satu data non-spasial yaitu data simpang dan diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.2.

```

Sub Simpan()
  Dim fld As MapObjects2.Field
  1   If txtIdSimpang = "" Then
  2     MsgBox "Data Harus diisi !", vbExclamation, "Peringatan"
  3     Exit Sub
  4   End If
  5   Select Case Mode
  6     Case 0
  7       TutupRs
  8       Katasql = "SELECT * FROM simpang WHERE
                ucase(nama_simpang)='" & UCase(Trim(txtIdSimpang)) & "'"
  9       rs.Open Katasql, Conn, adOpenStatic
  10      If rs.EOF = True Or rs.BOF = True Then
  11        Katasql = "SELECT * FROM simpang"
  12        TutupRs
  13        rs.Open Katasql, Conn, adOpenDynamic, adLockOptimistic
  14        rs.AddNew
  15        rs.Fields("nama_simpang") = Trim(txtIdSimpang)
  16        rs.Fields("jumlah_kaki") = Trim(txtJmlhKaki)
  17        rs.Fields("kondisi") = Trim(txtKondisi)
  18        rs.Fields("keterangan") = Trim(txtKtrngn)
  19        rs.Update
  20        BrowseMode
  21      Else
  22        MsgBox "Nama Simpang " & txtIdSimpang & " Sudah Ada
                !", vbInformation, "Peringatan"
  23      End If
  24    Case 1
  25      Set rsPencarian = frmPeta.Map1.Layers("traffic light").
                SearchExpression("simpang='" & Trim(SavedKode) & "'")
  26      If rsPencarian.EOF = False Then
  27        rsPencarian.Edit
  28        rsPencarian.Fields("simpang").Value = Trim(txtIdSimpang)
  29        rsPencarian.Update
  30      End If
  31      Katasql = "SELECT * FROM simpang WHERE ucase(nama_simpang)='"
                & Trim(UCase(SavedKode)) & "'"
  32      TutupRs
  33      rs.Open Katasql, Conn, adOpenDynamic, adLockOptimistic
  34      rs.Fields("nama_simpang") = Trim(txtIdSimpang)
  35      rs.Fields("jumlah_kaki") = Trim(txtJmlhKaki)
  36      rs.Fields("kondisi") = Trim(txtKondisi)
  37      rs.Fields("keterangan") = Trim(txtKtrngn)
  38      rs.Update
  39      Set rsPencarian = Nothing
  40      Screen.MousePointer = vbNormal
  41      frmPeta.IsiListView
  42      BrowseMode
  43    End Select
End Sub

```

Algoritma 5.2 Implementasi algoritma `Simpan()`

Sumber: Implementasi

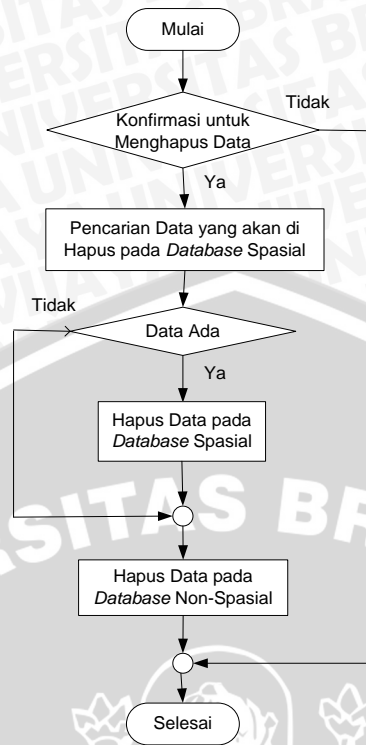
Penjelasan implementasi Algoritma 5.2 yaitu:

1. Baris 1-4, jika kolom `txtIdSimpang` (nama simpang) kosong maka akan muncul peringatan bahwa data tersebut harus di isi.

2. Baris 5, pilihan mode. Pilihan terdiri dari 2 mode yaitu mode 0 dan mode 1. Mode 0 digunakan untuk menyimpan data baru sedangkan mode 1 digunakan untuk mengedit data yang sudah ada.
3. Baris 6-9, jika mode 0 yang dipilih maka diperiksa terlebih dahulu apakah nama simpang pada `txtIdSimpang` sudah ada dalam *database*.
4. Baris 10-20, jika nama simpang pada `txtIdSimpang` tidak ada dalam *database* maka seluruh data yang telah diisi di simpan didalam *database*. Setelah itu jalankan prosedur `BrowseMode`.
5. Baris 21-23, jika nama simpang pada `txtIdSimpang` ada dalam *database* maka akan muncul peringatan bahwa data simpang tersebut sudah ada.
6. Baris 24-25, jika mode 1 yang dipilih maka diperiksa terlebih dahulu apakah nama simpang yang akan di edit ada pada *database* spasial layer *traffic light*.
7. Baris 26-30, jika nama simpang ada maka *field* simpang yang ada pada layer *traffic light* akan di *update* sesuai dengan perubahan nama simpang pada `txtIdSimpang`.
8. Baris 31-38, proses menampilkan atau memilih *record* simpang yang akan di edit dari *database* non-spasial kemudian data yang telah di edit disimpan pada *recordset* yang sesuai.
9. Baris 39-43, sebelum keluar dari prosedur, *record* hasil pencarian di set kosong dan tampilan tabel data disesuaikan dengan perubahan yang telah terjadi. Setelah itu jalankan prosedur `BrowseMode`.

5.3.3 Implementasi Algoritma Hapus Data Non-Spasial

Algoritma untuk menghapus data non-spasial diimplementasikan pada prosedur `Hapus()`. Prosedur ini dijalankan jika tombol untuk menghapus data pada *toolbar* non-spasial di klik. Diagram alir prosedur `Hapus()` ditunjukkan pada gambar 5.15.



Gambar 5.15 Flowchart prosedur Hapus ()
Sumber: Implementasi

Algoritma prosedur Hapus () yang digunakan untuk menghapus salah satu data non-spasial yaitu data simpang dan diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.3.

```

Sub hapus ()
1   If Mode = -1 Then
2       If MsgBox("Anda Yakin Menghapus Seluruh Data ?" & txtIdSimpang & " pada Database?", vbQuestion + vbYesNo, "Konfirmasi") = vbYes Then
3           Set rsPencarian = frmPeta.Map1.Layers("traffic light")._
                SearchExpression("simpang='" & Trim(txtIdSimpang) & "'")
4           If rsPencarian.EOF = False Then
5               rsPencarian.Edit
6               rsPencarian.Delete
7               rsPencarian.MoveNext
8               rsPencarian.StopEditing
9               frmPeta.Map1.Refresh
10          End If
11          Set rsPencarian = Nothing
12          Screen.MousePointer = vbNormal
13          frmPeta.IsiListView
14          Katasql = "DELETE from simpang where ucase(nama_simpang)='" & Trim(UCase(txtIdSimpang)) & "'"
15          Conn.Execute Katasql
16          LoadData
17              txtIdSimpang = ""
18              txtJmlhKaki = ""
19              txtKondisi = ""
20              txtKtrngn = ""
21          End If
22      End If
End Sub
    
```

Algoritma 5.3 Implementasi algoritma Hapus ()
Sumber: Implementasi

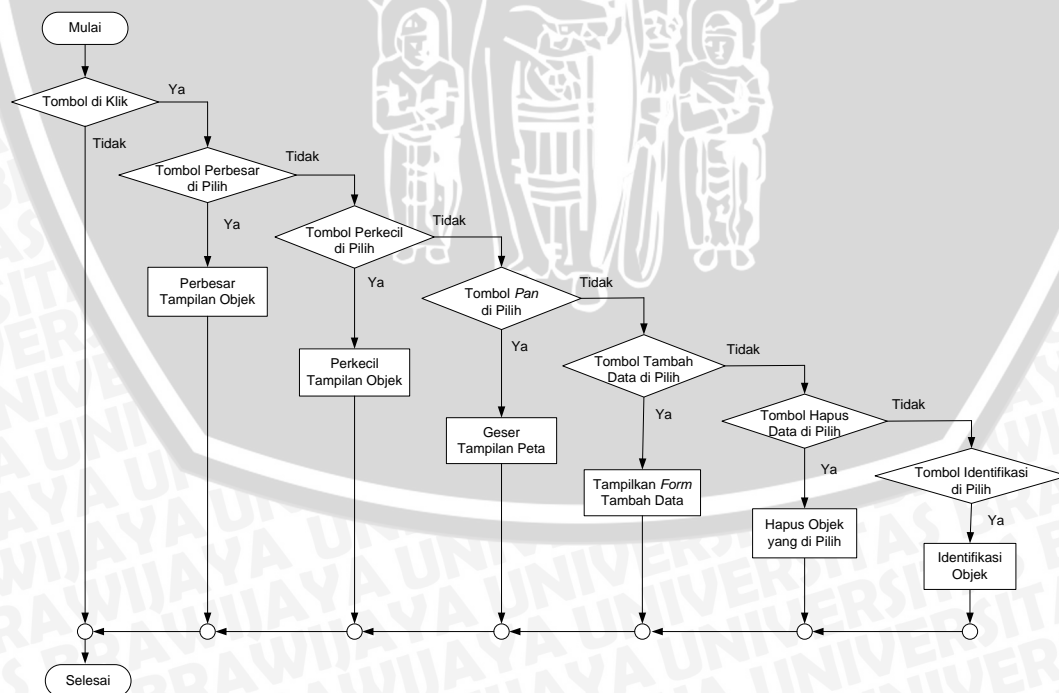


Penjelasan implementasi Algoritma 5.3 yaitu:

1. Baris 1, jika nilai mode -1 maka jalankan proses selanjutnya, jika tidak maka keluar prosedur.
2. Baris 2-3, jika pada `MsgBox` yang berisi konfirmasi untuk menghapus data pengguna memilih *Yes* maka program akan melakukan pencarian data simpang yang telah dipilih pada *database* spasial.
3. Baris 4-10, jika data simpang yang telah dipilih untuk dihapus ada pada *database* spasial maka *record* simpang tersebut dihapus.
4. Baris 11-13, *recordset* hasil pencarian di set kosong dan tampilan tabel data disesuaikan dengan perubahan yang telah terjadi.
5. Baris 14-16, hapus data simpang yang telah dipilih pada *database* non-spasial.
6. Baris 17-20, kolom untuk memasukkan data simpang menjadi kosong.

5.3.4 Implementasi Algoritma Mengolah Tampilan Peta

Mengolah tampilan peta dilakukan dengan memilih tombol pada *toolbar* spasial yang telah disediakan. Algoritma untuk mengolah tampilan peta diimplementasikan pada prosedur `Map1_MouseDown`. Prosedur ini akan aktif ketika *layer* peta di klik. Diagram alir prosedur `Map1_MouseDown` ditunjukkan pada gambar 5.16.



Gambar 5.16 Flowchart prosedur `Map1_MouseDown`

Sumber: Implementasi

Algoritma prosedur `Map1_MouseDown` yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.4.

```
Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
1   If Button = 1 Then
2       If tandaPerbesar = True Then
3           Set Map1.Extent = Map1.TrackRectangle
4       ElseIf tandaPerkecil = True Then
5           Dim rect As MapObjects2.Rectangle
6           Set rect = Map1.Extent
7           rect.ScaleRectangle 2
8           Map1.Extent = rect
9       ElseIf tandaPan = True Then
10          Map1.Pan
11      ElseIf tandaTitik = True Then
12          frmTambahSpasial.Show 1
13      ElseIf tandaHapusSpasial = True Then
14          IdentifyHapus X, Y, Map1
15      ElseIf tandaIdentifikasi = True Then
16          Identify X, Y, Map1
17      End If
18  End If
End Sub
```

Algoritma 5.4 Implementasi algoritma `Map1_MouseDown`

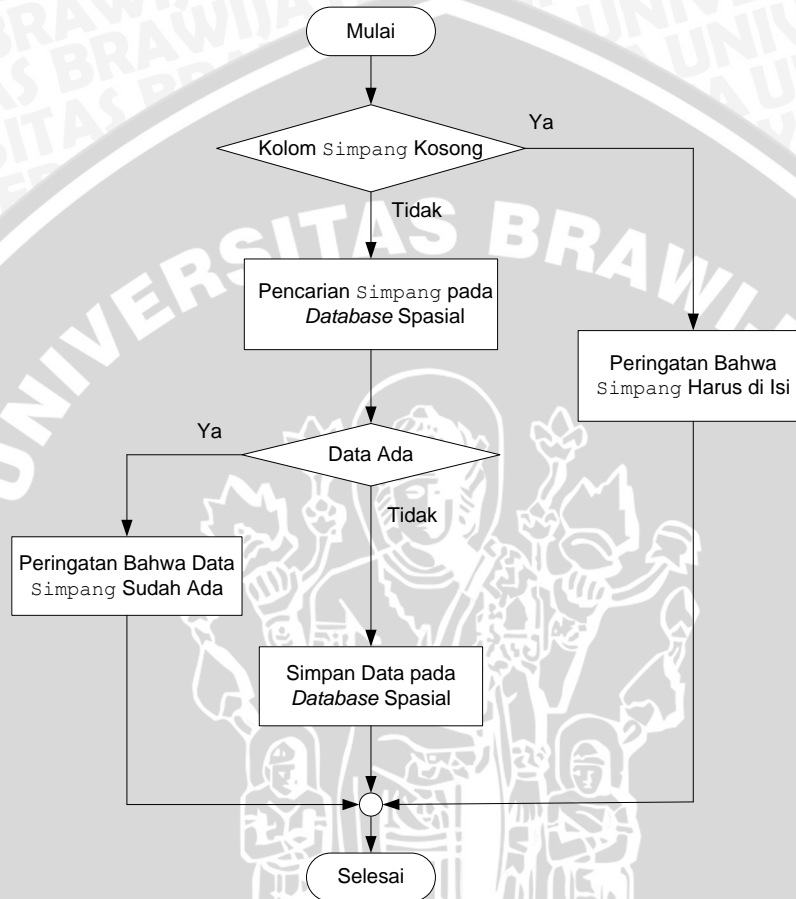
Sumber: Implementasi

Penjelasan implementasi Algoritma 5.4 yaitu:

1. Baris 1, jika tombol (*button*) di klik maka jalankan proses selanjutnya, jika tidak maka keluar prosedur.
2. Baris 2-3, jika tombol `tandaPerbesar` dipilih maka tampilan peta akan diperbesar sesuai dengan luas *mouse* yang di *drag* pada peta.
3. Baris 4-8, jika tombol `tandaPerkecil` dipilih maka setiap *mouse* di klik pada peta, tampilan peta akan diperkecil dengan nilai faktor 2.
4. Baris 9-10, jika tombol `tandaPan` dipilih maka tampilan peta bisa digeser dengan *mouse*.
5. Baris 11-12, jika tombol `tandaTitik` dipilih maka akan muncul tampilan *form* `frmTambahSpasial`. *Form* ini digunakan untuk menyimpan data spasial berupa data simpang.
6. Baris 13-14, jika tombol `tandaHapusSpasial` di pilih maka prosedur `IdentifyHapus` dipanggil. Prosedur ini digunakan untuk menghapus data spasial berupa data simpang yang telah di pilih.
7. Baris 15-16, jika tombol `tandaIdentifikasi` di pilih maka prosedur `Identify` dipanggil. Prosedur ini digunakan untuk mengidentifikasi data spasial berupa data simpang yang telah di pilih.

5.3.5 Implementasi Algoritma Simpan Data Spasial

Algoritma untuk menyimpan data spasial pada *database* diimplementasikan pada prosedur `cmdSimpan_Click()`. Prosedur ini dijalankan jika tombol untuk menyimpan data spasial di klik. Diagram alir prosedur `cmdSimpan_Click()` ditunjukkan pada gambar 5.17.



Gambar 5.17 Flowchart prosedur `cmdSimpan_Click()`

Sumber: Implementasi

Algoritma prosedur `cmdSimpan_Click()` yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.5.

```

Private Sub cmdsimpan_Click()
    Dim rsBaru As MapObjects2.Recordset
    Dim lyr As New MapObjects2.MapLayer
    Dim i As Long
    Dim ppt As New MapObjects2.Point
    Dim layer As New MapObjects2.MapLayer
    1   If Trim(flxDData.TextMatrix(1, indeksID)) = "" Then
    2       MsgBox "Nama Simpang Harus Diisi Terlebih Dahulu!", vbExclamation,
           "Peringatan"
    3       Exit Sub
    4   End If
    5   Set lyr.GeoDataset = Koneksi.FindGeoDataset("traffic light.shp")
    6   Set rsBaru = lyr.SearchExpression("Simpang =" & flxDData.TextMatrix(1,
           indeksID) & "")
  
```



```

7   If rsBaru.EOF <> True Then
8     MsgBox "Nama Simpang " & flxData.TextMatrix(1, indeksID) & " Sudah Ada !",
          vbExclamation, "Peringatan"
9     Exit Sub
10  End If
11  Set rsBaru = Nothing
12  Set lyr = Nothing
13  Set rsMap = Nothing
14  Set layer = frmPeta.Map1.Layers("traffic light")
15  Set rsMap = layer.Records
16  pPt.X = txtX
17  pPt.Y = txtY
18  pPt.z = 0
19  rsMap.Fields("Shape").Value = pPt
20  rsMap.AddNew
21  For i = 0 To UBound(ArrayNama)
22    rsMap.Fields(ArrayNama(i)).Value = (flxData.TextMatrix(1, i))
23  Next
24  rsMap.Update
25  frmPeta.Map1.Refresh
26  tandaExit = True
27  frmPeta.IsiListView
28  Set rsMap = Nothing
29  Set rsBaru = Nothing
30  Set lyr = Nothing
31  Unload Me
End Sub

```

Algoritma 5.5 Implementasi algoritma cmdSimpan_Click()

Sumber: Implementasi

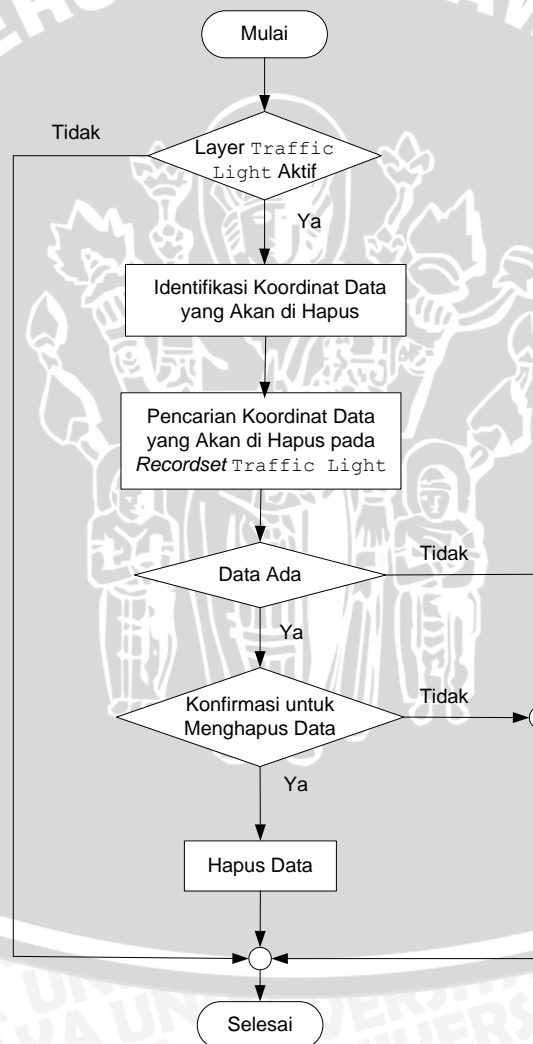
Penjelasan implementasi Algoritma 5.5 yaitu:

1. Baris 1-4, jika pada kolom untuk mengisi data simpang belum diisi maka akan muncul peringatan bahwa data simpang harus diisi terlebih dahulu.
2. Baris 5-6, koneksi pada layer *traffic light* kemudian dilakukan pencarian nama simpang yang telah diisi pada *recordset* layer *traffic light* tersebut.
3. Baris 7-10, jika data simpang ada maka akan muncul peringatan bahwa data simpang tersebut sudah ada kemudian keluar dari prosedur.
4. Baris 11-15, deklarasi variabel. Layer *traffic light* dinamakan dengan *layer* sedangkan *recordset* dari layer *traffic light* dinamakan dengan *rsMap*.
5. Baris 16-18, nilai koordinat dari data yang akan di simpan dimasukkan ke variabel *pPt* X, Y dan Z.
6. Baris 19, nilai variabel *pPt* disimpan pada *field* *Shape* yang dimiliki oleh *rsMap* (*recordset* dari layer *traffic light*).
7. Baris 20-23, semua data yang telah di isi oleh pengguna disimpan pada masing-masing *field* *rsMap* yang sesuai dengan nama *field* tabel isian.

8. Baris 24-27, `rsMap` di *update* dan tampilan peta di *refresh*. Setelah itu tampilan tabel data disesuaikan dengan perubahan yang telah terjadi.
9. Baris 28-31, sebelum tampilan *form* untuk menambah data spasial di tutup *recordset* di set kosong.

5.3.6 Implementasi Algoritma Hapus Data Spasial

Algoritma untuk menghapus data spasial pada *database* diimplementasikan pada prosedur `IdentifyHapus`. Prosedur ini dijalankan jika pengguna memilih tombol `tandaHapusSpasial` yang ada pada *toolbar* spasial data simpang yang berupa tanda titik pada peta di klik. Diagram alir prosedur `IdentifyHapus` ditunjukkan pada gambar 5.18.



Gambar 5.18 Flowchart prosedur `IdentifyHapus`
Sumber: Implementasi

Algoritma prosedur IdentifyHapus yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.6.

```

Sub IdentifyHapus(X As Single, Y As Single, Map As MapObjects2.Map)
    Dim tipeLayer As Long
    tipeLayer = Map1.Layers("traffic light").ShapeType
    Const SEARCHTOLPIXELS = 3
    Dim Loc As New MapObjects2.Point
    Dim theTol As Double
    Dim aLayer As Object
    Dim recs As MapObjects2.Recordset
    Dim xStr As String, yStr As String
1   Set aLayer = Map.Layers("traffic light")
2   If aLayer.Visible And aLayer.LayerType = moMapLayer Then
3       Set Loc = Map.ToMapPoint(X, Y)
4       theTol = Map.ToMapDistance(SEARCHTOLPIXELS * Screen.TwipsPerPixelX)
5       xStr = Int(Loc.X) : yStr = Int(Loc.Y)
6       Set recs = aLayer.SearchByDistance(Loc, theTol, "")
7       If recs.EOF = False Then
8           If MsgBox("Anda Yakin Menghapus Data Tersebut ? ", vbQuestion +
                vbYesNo, "Konfirmasi") = vbNo Then
9               Screen.MousePointer = vbNormal
10              Exit Sub
11              End If
12              recs.Edit
13              recs.Delete
14              recs.MoveNext
15              recs.StopEditing
16              Map1.Refresh
17              IsiListView
18              End If
19          End If
20          Set recs = Nothing
21          Screen.MousePointer = vbNormal
End Sub

```

Algoritma 5.6 Implementasi algoritma IdentifyHapus

Sumber: Implementasi

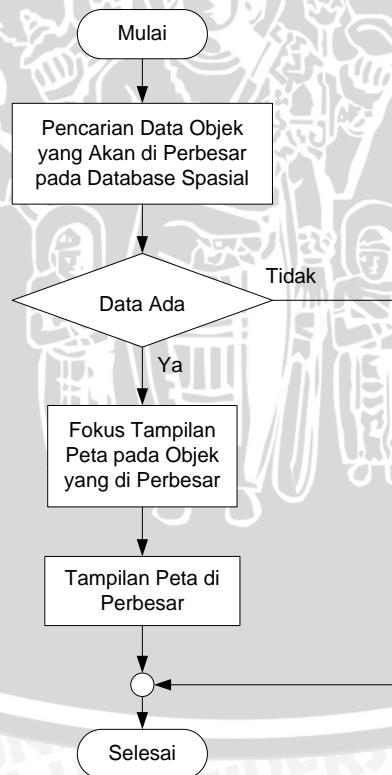
Penjelasan implementasi Algoritma 5.6 yaitu:

1. Baris 1, definisi variabel aLayer dimana aLayer merupakan layer *traffic light*.
2. Baris 2, jika aLayer terlihat dan tipe layer aLayer adalah moMapLayer maka jalankan proses selanjutnya, jika tidak maka langsung menuju ke baris 20.
3. Baris 3, definisi variabel Loc dimana nilai Loc didapatkan melalui konversi nilai koordinat *screen* ke koordinat peta.
4. Baris 4, definisi variabel theTol dimana nilai theTol merupakan hasil pengesetan toleransi peta.
5. Baris 5, nilai koordinat X dan Y yang telah didapat melalui baris 3 dikonversi sehingga mempunyai tipe *integer*.
6. Baris 6, variabel recs di set dengan hasil pencarian *recordset*. Hasil pencarian ini merupakan nilai koordinat dari data simpang yang telah dipilih.

7. Baris 7-11, jika hasil pencarian ada maka akan muncul konfirmasi untuk menghapus data. Jika pada `MsgBox` yang berisi konfirmasi tersebut pengguna memilih *No* maka keluar dari prosedur, jika tidak maka langsung menuju ke baris 12.
8. Baris 12-17, data simpang yang telah dipilih akan dihapus dan tampilan peta akan di *refresh*. Selanjutnya tampilan tabel data disesuaikan dengan perubahan yang terjadi.
9. Baris 20, sebelum keluar dari prosedur, *record* hasil pencarian (`recs`) di set kosong.

5.3.7 Implementasi Algoritma Perbesar Hasil Pencarian

Algoritma untuk memperbesar (*zoom*) tampilan hasil pencarian data simpang diimplementasikan pada prosedur `cmdZoom_Click()`. Lokasi data simpang yang dicari akan diperbesar tiap tombol *Zoom* di klik. Diagram alir prosedur `cmdZoom_Click()` ditunjukkan pada gambar 5.19. berikut.



Gambar 5.19 Flowchart prosedur `cmdZoom_Click()`

Sumber: Implementasi

Algoritma prosedur `cmdZoom_Click()` yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.7.

```
Private Sub cmdZoom_Click()
    Dim RECT2 As Object
    Dim shapex As Double
    Dim shapex As Double
    Dim deltax As Double
    Dim deltay As Double
    1   Katasql = Pencarian & "=" & KataCari & ""
    2   Set rsPencarian = Map1.Layers("traffic light").SearchExpression(Katasql)
    3   If rsPencarian.EOF = False Then
    4       Set RECT2 = Map1.Extent
    5       shapex = rsPencarian("shape").Value.X
    6       shapex = rsPencarian("shape").Value.Y
    7       deltax = shapex - RECT2.Center.X
    8       deltay = shapex - RECT2.Center.Y
    9       RECT2.Offset deltax, deltay
    10      RECT2.ScaleRectangle 0.5
    11      Map1.Extent = RECT2
    12  End If
End Sub
```

Algoritma 5.7 Implementasi algoritma `cmdZoom_Click()`

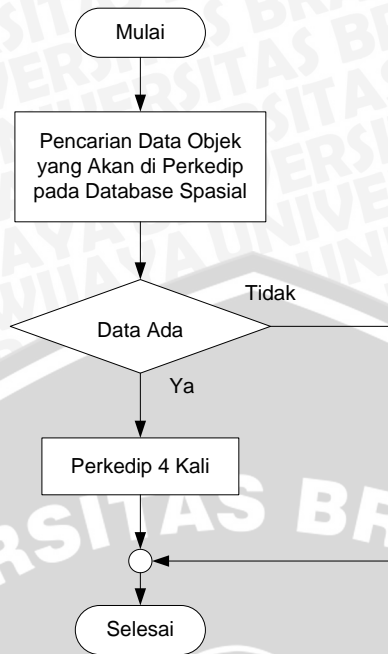
Sumber: Implementasi

Penjelasan implementasi Algoritma 5.7 yaitu:

1. Baris 1, pemberian nilai pada variabel `Katasql`. Nilai ini merupakan nama simpang hasil pencarian sistem.
2. Baris 2, sistem melakukan pencarian nama simpang tersebut pada pada *recordset* layer *traffic light*. Hasil pencarian nama simpang dimasukkan ke variabel `rsPencarian`.
3. Baris 3-9 jika nama simpang ada pada *recordset* layer *traffic light* maka fokus tampilan peta adalah titik simpang yang di cari.
4. Baris 10-12, tampilan lokasi hasil pencarian akan diperkecil dengan faktor 0.5 yang juga berarti bahwa tampilan peta akan diperbesar dengan faktor 2.

5.3.8 Implementasi Algoritma Kedip Hasil Pencarian

Algoritma untuk memperkedip (*flash*) hasil pencarian data simpang diimplementasikan pada prosedur `cmdKedip_Click()`. Data simpang hasil pencarian yang direpresentasikan dengan simbol titik pada peta akan berkedip tiap tombol Kedip di klik. Diagram alir prosedur `cmdKedip_Click()` ditunjukkan pada gambar 5.20.



Gambar 5.20 Flowchart prosedur cmdKedip_Click()

Sumber: Implementasi

Algoritma prosedur cmdKedip_Click() yang diimplementasikan menggunakan bahasa pemrograman Visual Basic ditunjukkan pada Algoritma 5.8.

```

Private Sub cmdKedip_Click()
    Dim shp As Object
    1   Katasql = Pencarian & "=" & KataCari & ""
    2   Set rsPencarian = Map1.Layers("traffic light"). _
        SearchExpression(Katasql)
    3   If rsPencarian.EOF = False Then
    4       Set shp = rsPencarian.Fields("Shape").Value
    5       Map1.FlashShape shp, 4
    6   End If
End Sub
  
```

Algoritma 5.8 Implementasi algoritma cmdZoom_Click()

Sumber: Implementasi

Penjelasan implementasi Algoritma 5.8 yaitu:

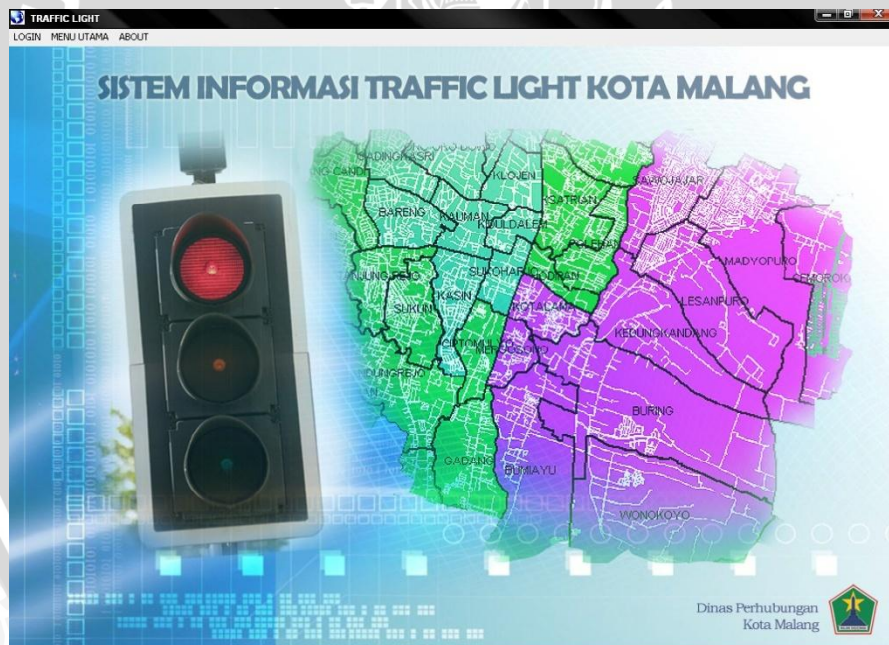
1. Baris 1, pemberian nilai pada variabel *Katasql*. Nilai ini merupakan nama simpang hasil pencarian sistem.
2. Baris 2, sistem melakukan pencarian nama simpang tersebut pada *recordset* layer *traffic light* dan hasilnya dimasukkan ke variabel *rsPencarian*.
3. Baris 3-4, jika nama simpang ada pada *recordset* layer *traffic light* maka nilai pada *field* *shape* simpang tersebut dimasukkan ke variabel *shp*.
4. Baris 5-6, titik simpang hasil pencarian akan berkedip sebanyak 4 kali.

5.4 Implementasi Antarmuka Aplikasi

Antarmuka aplikasi ini dibuat sesuai dengan kebutuhan fungsional yang harus disediakan oleh sistem. Tujuan utama pengembangan antarmuka adalah untuk memberikan kemudahan bagi pengguna (*user*) dalam menggunakan aplikasi yang telah dibangun. Implementasi antarmuka aplikasi ini terdiri atas implementasi antarmuka Administrator dan Tamu.

5.4.1 Implementasi Antarmuka Tampilan Awal

Antarmuka ini merupakan tampilan awal ketika program pertama kali dibuka. Menu yang dapat dipilih pada antarmuka ini terdiri atas 3 macam yaitu *Login*, Menu Utama dan *About*. *Login* digunakan Administrator untuk memasuki sistem, Menu Utama digunakan untuk membuka aplikasi yang ditujukan untuk Tamu sedangkan *About* berisi informasi tentang hak cipta pembuatan sistem. Gambar 5.21 merupakan gambar hasil implementasi antarmuka Tampilan Awal.



Gambar 5.21 Antarmuka Tampilan Awal
Sumber: Implementasi

5.4.2 Implementasi Antarmuka *Login*

Antarmuka *login* ini akan ditampilkan ketika pengguna ingin menggunakan aplikasi sebagai Administrator. Antarmuka ini mempunyai dua buah masukan data yaitu "*Username*" dan "*Password*". Gambar 5.22 merupakan gambar hasil implementasi antarmuka *Login*.



Gambar 5.22 Antarmuka *Login*
Sumber: Implementasi

Penjelasan bagian dari antarmuka ini diuraikan pada Tabel 5.3.

Tabel 5.3 Penjelasan tombol pada antarmuka *Login*

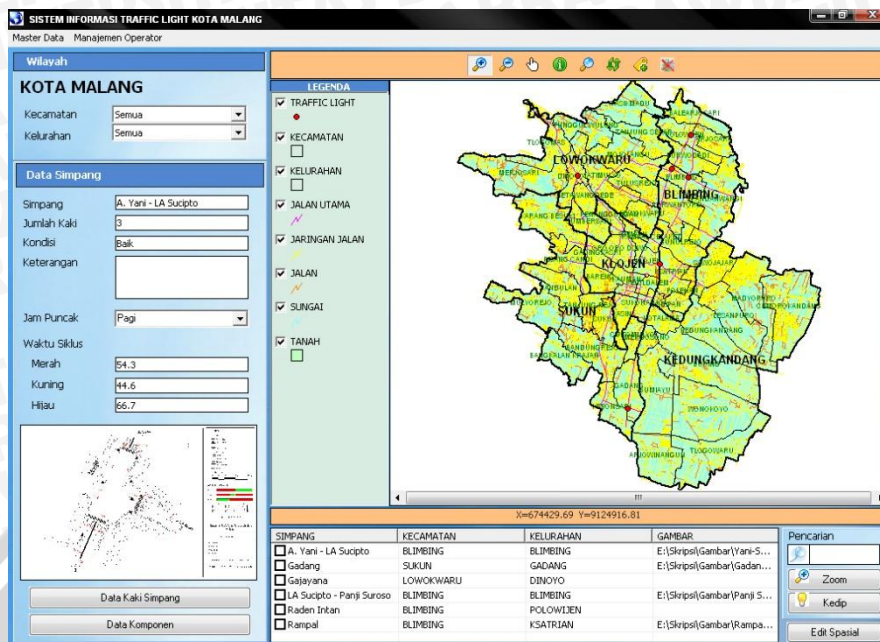
No	Label Tombol	Penjelasan
1	OK	Tombol ini digunakan jika pengguna ingin melakukan validasi akun yang telah dimasukkan.
2	Batal	Tombol ini digunakan jika pengguna ingin membatalkan <i>Login</i> dan menutup <i>form</i> ini.

Sumber: Implementasi

5.4.3 Implementasi Antarmuka Aplikasi Administrator

Antarmuka Aplikasi Administrator merupakan aplikasi yang ditujukan untuk Administrator. Antarmuka ini akan ditampilkan jika data yang dimasukkan oleh pengguna berupa *username* dan *password* pada *form Login* benar. Aplikasi Administrator mempunyai beberapa fungsi yaitu menu yang digunakan untuk mengolah data non-spasial, tombol yang digunakan untuk menampilkan data non spasial serta tombol yang digunakan untuk mengolah data spasial.

Menu yang dapat dipilih pada antarmuka ini terbagi menjadi 2 macam yaitu Master Data dan Manajemen Operator. Menu-menu tersebut digunakan untuk mengolah data non-spasial. Tombol Data Kaki Simpang dan tombol Data Komponen berfungsi untuk menampilkan data non-spasial. Sedangkan tombol yang digunakan untuk mengolah data spasial atau mengubah tampilan peta terdapat pada *toolbar* spasial dan tombol Edit Spasial. Gambar 5.23 merupakan gambar hasil implementasi antarmuka aplikasi untuk Administrator.



Gambar 5.23 Antarmuka Aplikasi Administrator
Sumber: Implementasi

Penjelasan masing-masing tombol dari antarmuka ini diuraikan pada Tabel 5.4.

Tabel 5.4 Penjelasan tombol pada antarmuka Aplikasi Administrator

No	Label Tombol	Penjelasan
1	Data Kaki Simpang	Tombol ini digunakan untuk menampilkan data kaki simpang dari data simpang yang telah dipilih melalui tabel data.
2	Data Komponen	Tombol ini digunakan untuk menampilkan data komponen dari data simpang yang telah dipilih melalui tabel data.
3	Zoom	Tombol ini digunakan untuk memperbesar tampilan objek berupa data simpang hasil pencarian.
4	Kedip	Tombol ini digunakan untuk menjadikan hasil pencarian berupa data simpang yang bersimbol titik berkedip.
5	Edit Spasial	Tombol ini digunakan untuk merubah data simpang yang telah dipilih melalui tabel data.

Sumber: Implementasi

5.4.3.1 Menu Master Data

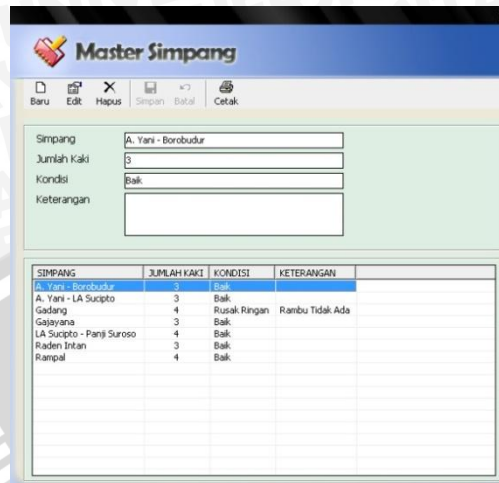
Menu Master Data terbagi menjadi 3 submenu yaitu Master Data Simpang, Master Data Kaki Simpang dan Master Data Komponen.

5.4 3.1.1 Menu Master Data Simpang

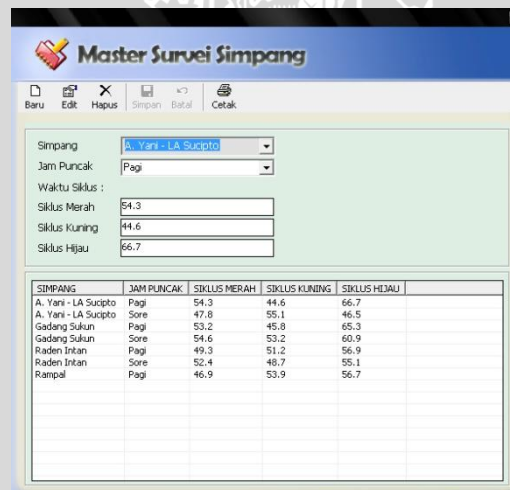
Menu Master Data Simpang terbagi menjadi 2 submenu yaitu Data Simpang dan Data Survei. Data Simpang digunakan untuk menyimpan data simpang sedangkan Data Survei digunakan untuk menyimpan data hasil survei pada simpang tersebut. Gambar 5.24 merupakan gambar hasil implementasi



antarmuka menu Data Simping sedangkan gambar 5.25 merupakan gambar hasil implementasi antarmuka menu Data Survei Simping.

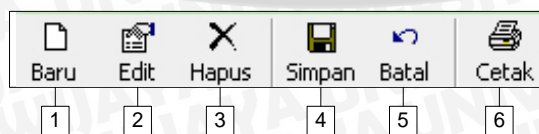


Gambar 5.24 Antarmuka Menu Data Simping
Sumber: Implementasi



Gambar 5.25 Antarmuka Menu Data Survei Simping
Sumber: Implementasi

Menu yang digunakan untuk mengolah data-data non spasial mempunyai *toolbar button*. *Toolbar button* non-spasial ini mempunyai beberapa tombol dengan fungsi berbeda-beda. Tampilan *toolbar* ditunjukkan pada gambar 5.26 berikut.



Gambar 5.26 Antarmuka *Toolbar* Non-Spasial
Sumber: Implementasi

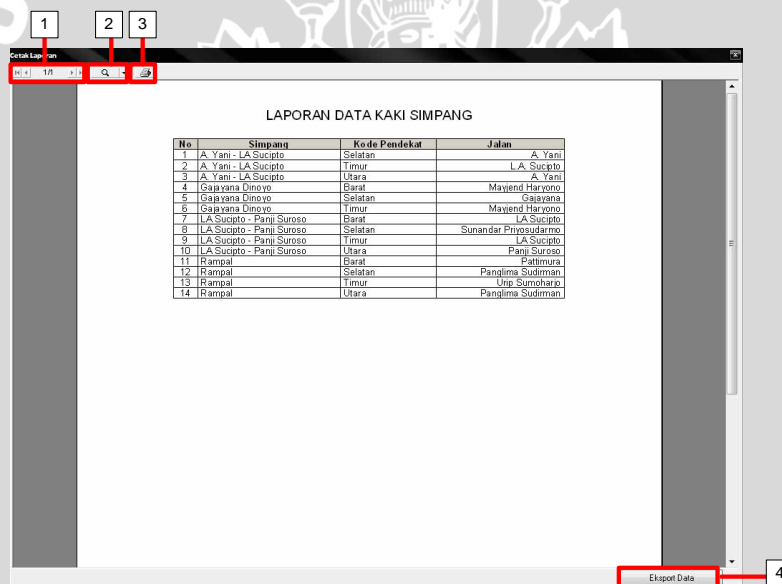
Penjelasan masing-masing tombol pada *toolbar* diuraikan pada Tabel 5.5.

Tabel 5.5 Penjelasan tombol pada *toolbar* non-spasial

No	Label Tombol	Penjelasan
1	Baru	Tombol ini digunakan untuk menambah data baru.
2	Edit	Tombol ini digunakan untuk mengedit data yang telah dipilih melalui tabel data masing-masing <i>form</i> .
3	Hapus	Tombol ini digunakan untuk menghapus data yang telah dipilih melalui tabel data masing-masing <i>form</i> .
4	Simpan	Tombol ini digunakan untuk menyimpan data.
5	Batal	Tombol ini digunakan untuk membatalkan proses menyimpan data.
6	Cetak	Tombol ini digunakan untuk mencetak data.

Sumber: Implementasi

Data yang akan dicetak adalah data yang terdapat pada tabel data masing-masing *form*. Contoh tampilan antarmuka Cetak Data dimana data yang akan dicetak adalah Data Kaki Sempang ditunjukkan pada gambar 5.27 berikut.



Gambar 5.27 Antarmuka Cetak Data

Sumber: Implementasi

Penjelasan bagian dari antarmuka ini diuraikan pada Tabel 5.6.

Tabel 5.6 Penjelasan tombol pada antarmuka Cetak Data

No	Label Tombol	Penjelasan
1	Halaman	<i>Label</i> ini menunjukkan jumlah halaman yang akan dicetak.
2	Tampilan	Tombol ini digunakan untuk mengatur tampilan (skala perbesaran) halaman.
3	Print	Tombol ini digunakan untuk mencetak data.
4	Eksport Data	Tombol ini digunakan untuk mengekspor data ke Microsoft Word.

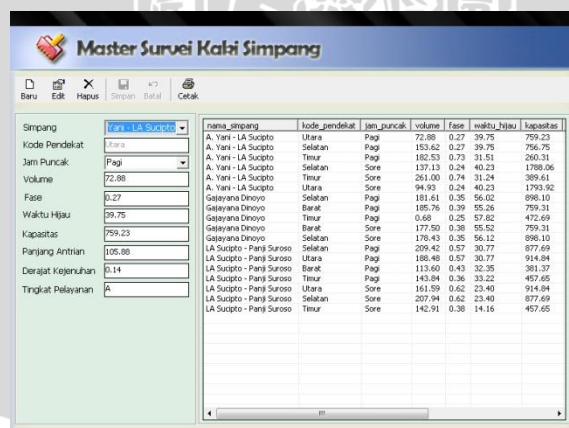
Sumber: Implementasi

5.4.3.1.2 Menu Master Data Kaki Simpang

Menu Master Data Kaki Simpang terbagi menjadi 2 submenu yaitu Data Kaki Simpang dan Data Survei. Data Kaki Simpang digunakan untuk menyimpan data kaki simpang sedangkan Data Survei digunakan untuk menyimpan data hasil survei pada masing-masing kaki simpang. Gambar 5.28 merupakan gambar hasil implementasi antarmuka menu Data Kaki Simpang sedangkan gambar 5.29 merupakan gambar hasil implementasi antarmuka menu Data Survei Kaki Simpang.



Gambar 5.28 Antarmuka Menu Data Kaki Simpang
Sumber: Implementasi

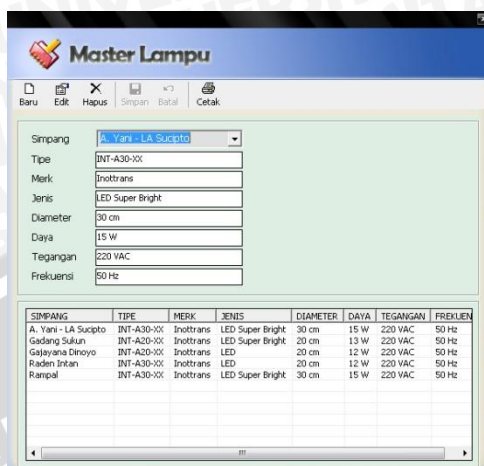


Gambar 5.29 Antarmuka Menu Data Survei Kaki Simpang
Sumber: Implementasi

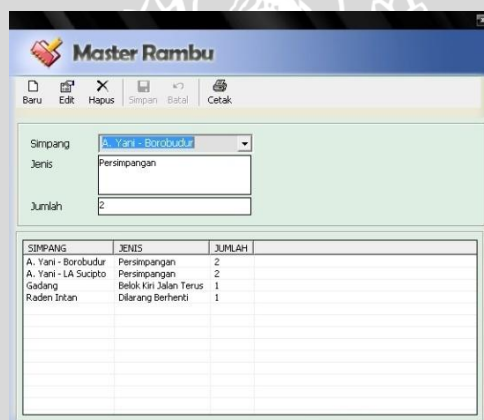
5.4.3.1.3 Menu Master Data Komponen

Menu Master Data Komponen terbagi menjadi 5 submenu yaitu menu Master Lampu, Master Rambu, Master Marka, Master Tiang dan Master

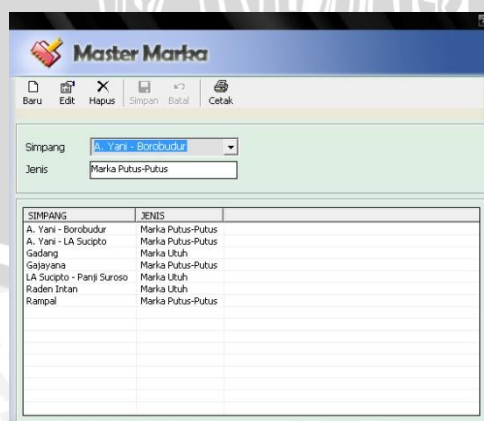
Perkebalan. Berikut merupakan gambar hasil implementasi antarmuka menu-menu Master Data Komponen.



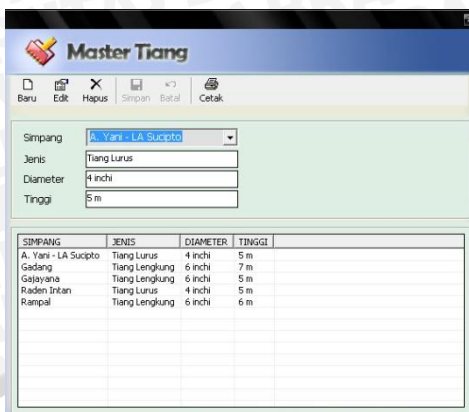
Gambar 5.30 Antarmuka Menu Data Lampu
Sumber: Implementasi



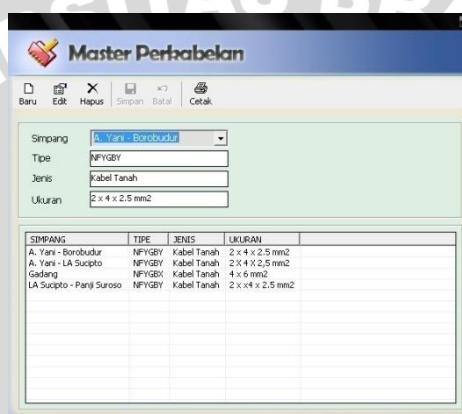
Gambar 5.31 Antarmuka Menu Data Rambu
Sumber: Implementasi



Gambar 5.32 Antarmuka Menu Data Marka
Sumber: Implementasi



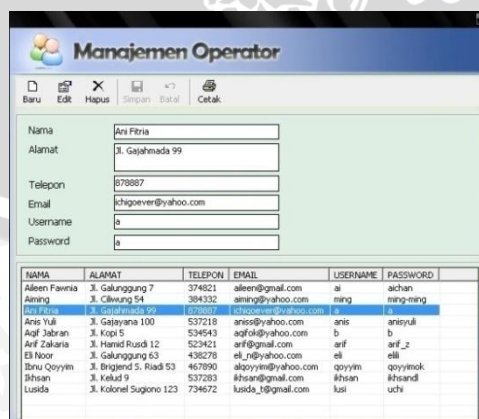
Gambar 5.33 Antarmuka Menu Data Tiang
Sumber: Implementasi



Gambar 5.34 Antarmuka Menu Data Perkabelan
Sumber: Implementasi

5.4.3.2 Menu Manajemen Operator

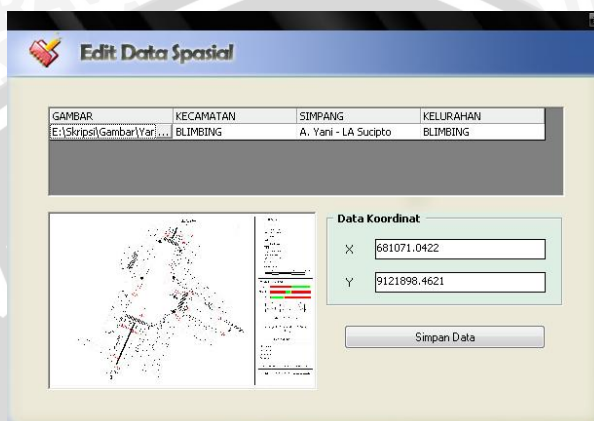
Menu Manajemen Operator digunakan untuk menyimpan data identitas dan akun Administrator. Gambar 5.35 merupakan gambar hasil implementasi antarmuka menu Manajemen Operator.



Gambar 5.35 Antarmuka Menu Manajemen Operator
Sumber: Implementasi

5.4.3.3 Menu Edit Data Spasial

Menu Edit Data Spasial akan ditampilkan jika tombol Edit Spasial di klik. Data yang akan di edit adalah data spasial simpang yang dipilih pengguna melalui tabel data. Contoh tampilan Menu Edit Data Spasial dimana data yang akan di *edit* adalah data spasial simpang A.Yani – L.A Sucipto ditunjukkan pada gambar 5.36 berikut.



Gambar 5.36 Antarmuka Menu Edit Data Spasial

Sumber: Implementasi

Penjelasan bagian dari antarmuka ini diuraikan pada Tabel 5.6.

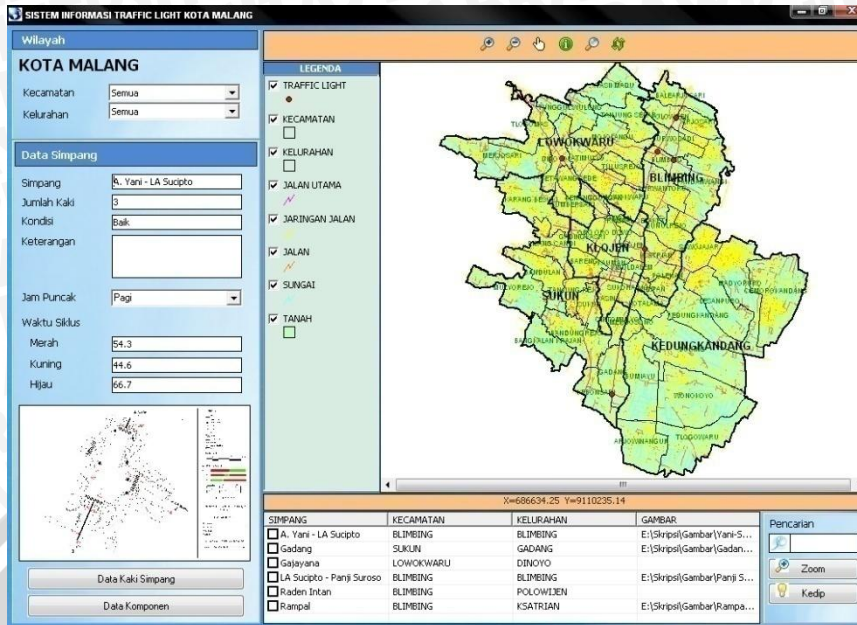
Tabel 5.7 Penjelasan tombol pada antarmuka Edit Data Spasial

No	Label Tombol	Penjelasan
1	Simpan	Tombol ini digunakan jika pengguna ingin menyimpan data-data yang telah diedit pada masing-masing kolom.

Sumber: Implementasi

5.4.4 Implementasi Antarmuka Aplikasi Tamu

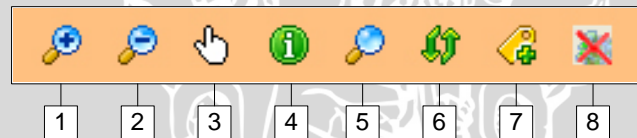
Antarmuka Aplikasi Tamu merupakan aplikasi yang ditujukan untuk Tamu. Antarmuka ini akan ditampilkan jika Menu Utama yang terdapat pada Tampilan Awal dipilih oleh pengguna. Perbedaan antara Aplikasi Tamu dengan Aplikasi Administrator yaitu bahwa pada Aplikasi Tamu tidak terdapat menu Master Data, Manajemen Operator dan tombol Edit Spasial. Selain itu perbedaan juga terletak pada *toolbar spasial*. *Toolbar spasial* pada Aplikasi Tamu tidak terdapat tombol untuk menambah dan menghapus data spasial. Gambar 5.37 merupakan gambar hasil implementasi antarmuka aplikasi untuk Tamu.



Gambar 5.37 Antarmuka Aplikasi Tamu
Sumber: Implementasi

5.4.5 Implementasi Antarmuka *Toolbar* Spasial

Toolbar spasial berisi tombol-tombol yang digunakan untuk mengubah atau mengolah tampilan peta. Tampilan *toolbar* ini ditunjukkan pada gambar 5.38.



Gambar 5.38 Antarmuka *Toolbar* Spasial
Sumber: Implementasi

Penjelasan masing-masing tombol pada *toolbar* diuraikan pada Tabel 5.7.

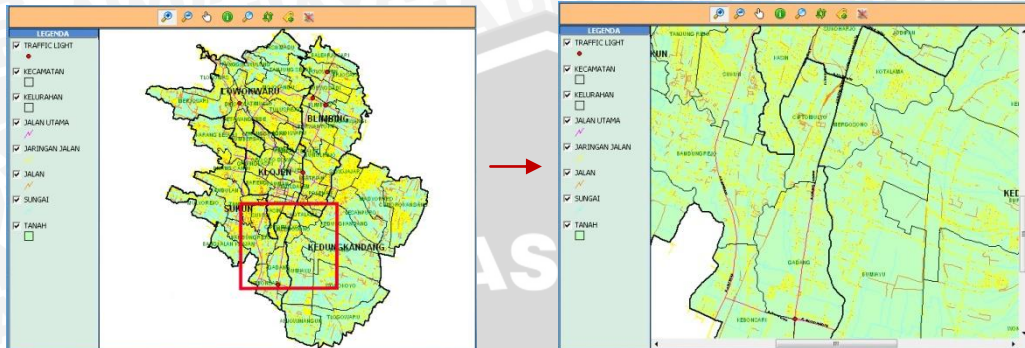
Tabel 5.8 Penjelasan tombol pada *toolbar* spasial

No	Label Tombol	Penjelasan
1	Zoom In	Tombol ini digunakan untuk memperbesar tampilan objek.
2	Zoom Out	Tombol ini digunakan untuk memperkecil tampilan objek.
3	Pan	Tombol ini digunakan untuk menggeser tampilan objek.
4	Identifikasi	Tombol ini digunakan untuk memberikan informasi dari suatu objek (titik simpang) yang telah dipilih.
5	Zoom Fit	Tombol ini digunakan untuk menampilkan keseluruhan objek sehingga semua peta tampak.
6	Refresh	Tombol ini digunakan untuk <i>refresh</i> tampilan peta.
7	Tambah Data	Tombol ini digunakan untuk menambah objek berupa titik simpang.
8	Hapus Data	Tombol ini digunakan untuk menghapus objek (titik simpang) yang telah dipilih.

Sumber: Implementasi

5.4.5.1 Zoom In

Tombol *Zoom In* digunakan untuk memperbesar tampilan objek yang dipilih. Tombol ini dengan cara melakukan *drag* pada objek yang dituju. Contoh tampilan objek yang telah diperbesar dapat dilihat pada gambar 5.39 berikut.

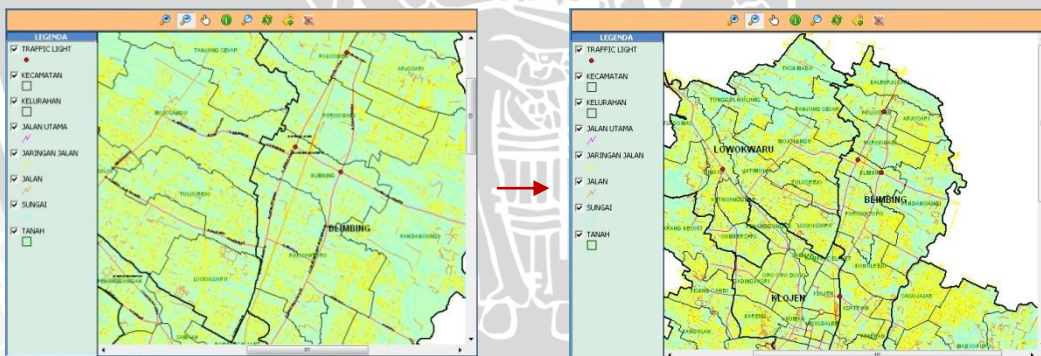


Gambar 5.39 Antarmuka peta sebelum dan sesudah diperbesar

Sumber: Implementasi

5.4.5.2 Zoom Out

Tombol *Zoom Out* digunakan untuk memperkecil tampilan objek yang dipilih. Tombol *Zoom Out* digunakan dengan cara mengklik objek yang akan diperkecil. Contoh tampilan objek yang telah diperkecil dapat dilihat pada gambar 5.40 berikut.

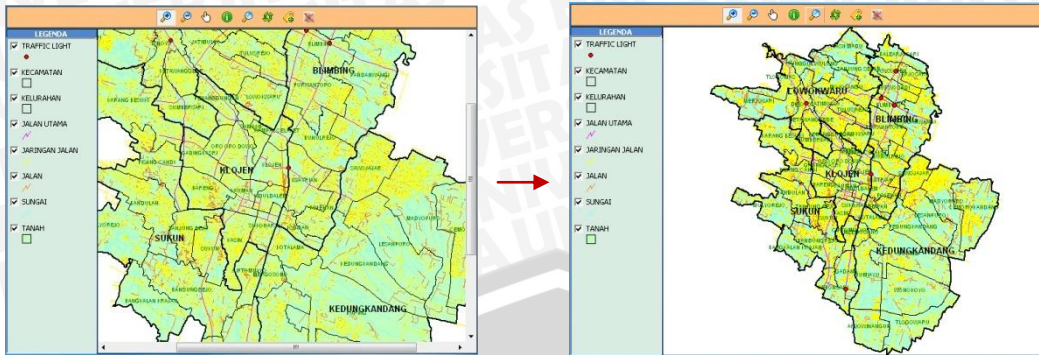


Gambar 5.40 Antarmuka peta sebelum dan sesudah diperkecil

Sumber: Implementasi

5.4.5.3 Zoom Fit

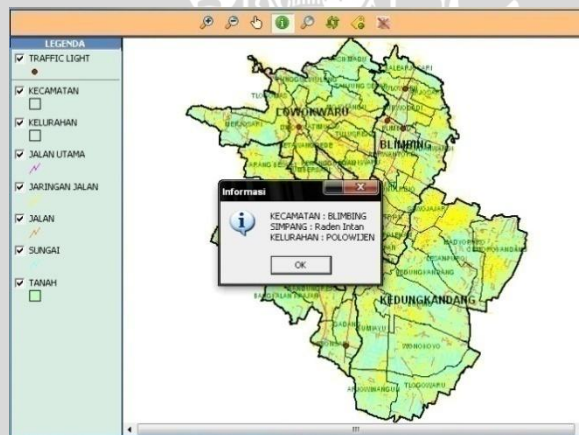
Tombol *Zoom Fit* digunakan untuk menampilkan keseluruhan objek. Tombol ini digunakan dengan cara mengklik objek yang akan di *Zoom Fit*. Contoh tampilan objek yang telah di *zoom fit* dapat dilihat pada gambar 5.41 berikut.



Gambar 5.41 Antarmuka peta sebelum dan sesudah di *Zoom Fit*
Sumber: Implementasi

5.4.5.4 Identifikasi

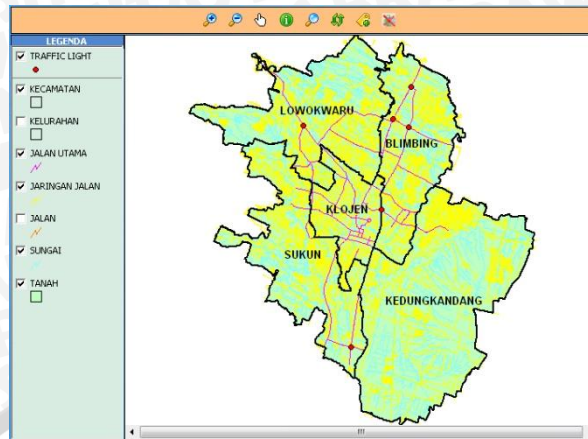
Tombol Identifikasi digunakan untuk memberikan informasi dari suatu objek (titik simpang) yang telah dipilih. Tombol ini digunakan dengan cara mengklik objek yang akan diidentifikasi. Contoh tampilan objek yang telah diidentifikasi dapat dilihat pada gambar 5.42 berikut.



Gambar 5.42 Antarmuka identifikasi simpang
Sumber: Implementasi

5.4.6 Implementasi Antarmuka Legenda

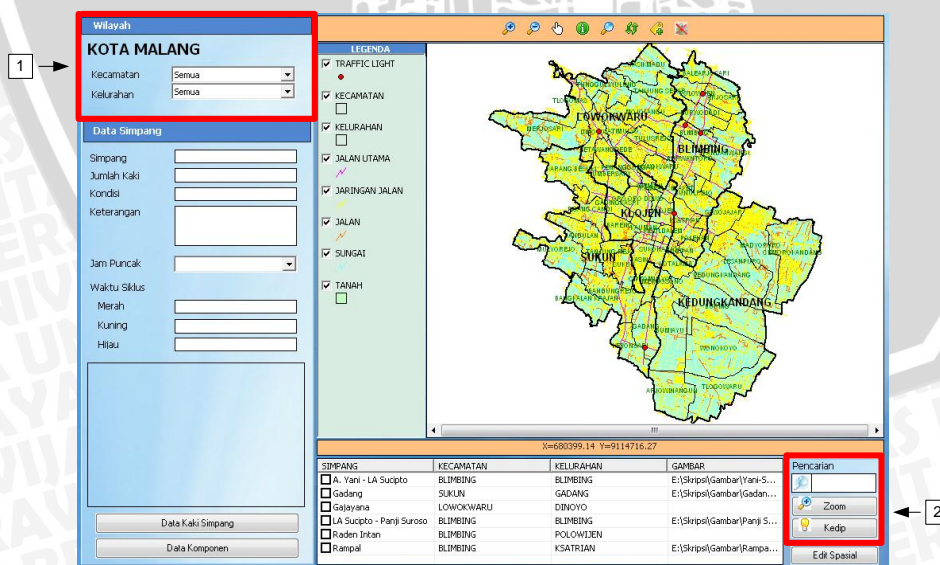
Legenda yang ditampilkan pada peta merepresentasikan tiap *layer* yang ada pada *database* spasial. Legenda ini terdiri atas 8 macam, yaitu *Traffic Light*, Kecamatan, Kelurahan, Jalan Utama, Jaringan Jalan, Jalan, Sungai dan Tanah. Pengguna dapat memilih legenda yang akan ditampilkan pada peta dengan cara mengklik *checkbox* pada tiap legenda tersebut. Contoh tampilan peta dimana legenda Kelurahan dan Jalan tidak dipilih ditunjukkan pada gambar 5.43 berikut.



Gambar 5.43 Antarmuka peta dengan legenda yang dapat dipilih
Sumber: Implementasi

5.4.7 Implementasi Antarmuka Pencarian

Pencarian data spasial yang ada pada sistem informasi ini terbagi menjadi 2 macam, yaitu pencarian berdasarkan wilayah dan pencarian lokasi simpang. Pencarian berdasarkan wilayah meliputi pencarian wilayah kecamatan dan kelurahan. Ketika pengguna memilih kecamatan maupun kelurahan maka secara otomatis sistem akan memperbesar dan memperkedip wilayah kecamatan atau kelurahan yang telah dipilih tersebut. Sedangkan pada pencarian lokasi simpang, tampilan hasil pencarian berupa titik simpang diperbesar dan diperkedip menggunakan tombol *Zoom* dan *Kedip*. Gambar 5.44 merupakan gambar hasil implementasi antarmuka pencarian.



Gambar 5.44 Antarmuka pencarian
Sumber: Implementasi



Penjelasan bagian dari antarmuka ini diuraikan pada Tabel 5.8.

Tabel 5.9 Penjelasan *frame* pada antarmuka Pencarian

No	Frame	Penjelasan
1	Wilayah	Frame ini menyediakan kolom pencarian wilayah kecamatan dan kelurahan.
2	Pencarian	Frame ini menyediakan kolom pencarian lokasi simpang.

Sumber: Implementasi

Hasil pencarian lokasi simpang ditunjukkan dengan adanya tanda cek pada *record* tabel data. Data yang akan diperbesar dengan tombol *Zoom* atau Data yang akan diperkedip dengan tombol *Kedip* adalah *record* yang mempunyai tanda cek.

5.4.8 Implementasi Antarmuka Tampilan Data Non-Spasial

Data Non-Spasial yang ditampilkan pada tampilan utama sistem terdiri atas 3 macam, yaitu data Simpang, data Kaki Simpang dan data Komponen. *Frame* Data Simpang ditampilkan secara *default* pada tampilan utama. Sedangkan Data Kaki Simpang dan Data Komponen ditampilkan dengan cara menekan tombol Data Kaki Simpang dan Data Komponen. Contoh tampilan data non-spasial simpang, kaki simpang dan komponen dengan nama simpang A.Yani – L.A Sucipto ditunjukkan pada gambar 5.45 berikut.

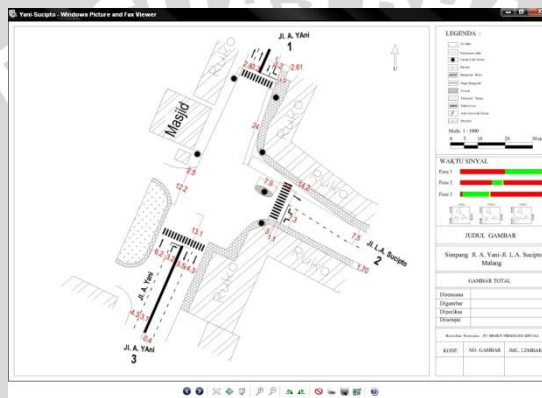
The image shows three side-by-side data entry forms for a non-spatial data system. Each form is titled with its respective data type: 'Data Simpang', 'Data Kaki Simpang', and 'Data Komponen'. The 'Data Simpang' form includes fields for 'Simpang' (A. Yani - LA Sucipto), 'Jumlah Kaki' (3), 'Kondisi' (Bak), 'Keterangan', 'Jam Puncak' (Pagi), 'Waktu Siklus' (Merah: 54.3, Kuning: 44.6, Hijau: 56.7), and a map area with a 'Double Klik untuk Buka Gambar' instruction. The 'Data Kaki Simpang' form includes fields for 'Simpang' (A. Yani - LA Sucipto), 'Kode Pendekat' (Timur), 'Jalan' (L.A. Sucipto), 'Jam Puncak' (Pagi), 'Tampilkan' button, 'Volume' (182.53), 'Fase' (0.73), 'Waktu Hijau' (31.51), 'Kapasitas' (260.31), 'Panjang Antrian' (80.00), 'Derajat Kejuhuan' (0.81), 'Tingkat Pelayanan' (A), and a 'Tutup' button. The 'Data Komponen' form includes fields for 'Simpang' (A. Yani - LA Sucipto), 'Lampu' (Tipe: INT-A30-XX, Merk: Inotrans, Jenis: LED Super Bright, Diameter: 30 cm, Daya: 15 W, Tegangan: 220 VAC, Frekuensi: 50 Hz), 'Rambu' (Jenis: Persimpangan, Jumlah: 2), 'Marka' (Jenis: Marka Putus-Putus), 'Tiang' (Jenis: Tiang Lurus, Diameter: 4 inch, Tinggi: 5 m), and 'Perkabelan' (Tipe: NFVGBY, Jenis: Kabel Tanah, Ukuran: 2 X 4 X 2,5 mm2). Each form has a 'Tutup' button at the bottom.

Gambar 5.45 Antarmuka data non-spasial
Sumber: Implementasi

Data hasil survei simpang terdiri atas jumlah kaki simpang, kondisi, waktu siklus pada tiap jam puncak dan gambar kondisi geometrik pada simpang tersebut. Sedangkan data hasil survei kaki simpang terdiri atas volume, fase, waktu hijau,

kapasitas, panjang antrian, derajat kejenuhan dan tingkat pelayanan pada tiap jam puncak di masing-masing kaki simpang tersebut. Untuk data komponennya terdiri atas data lampu, rambu, marka, tiang dan perkabelan.

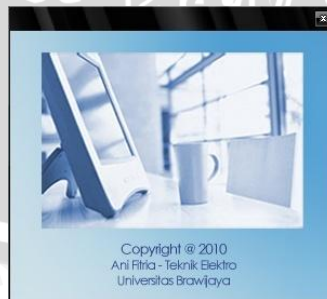
Gambar yang terdapat pada *frame* Data Simpang merupakan gambar kondisi geometrik dari tiap simpang. *File* gambar tersebut dapat dibuka dengan cara melakukan *double click* pada gambar yang bersangkutan. Jika gambar kondisi geometrik pada *frame* Data Simpang tersebut di *double* klik maka akan muncul tampilan aplikasi gambar tersebut seperti yang ditunjukkan pada gambar 5.46 berikut.



Gambar 5.46 Antarmuka kondisi geometrik
Sumber: Implementasi

5.4.9 Implementasi Antarmuka *About*

Antarmuka *About* ditampilkan jika menu *About* pada Tampilan Awal di klik. Antarmuka ini berisi informasi tentang hak cipta pembuatan sistem. Gambar 5.47 merupakan gambar hasil implementasi antarmuka *About*.



Gambar 5.47 Antarmuka *About*
Sumber: Implementasi