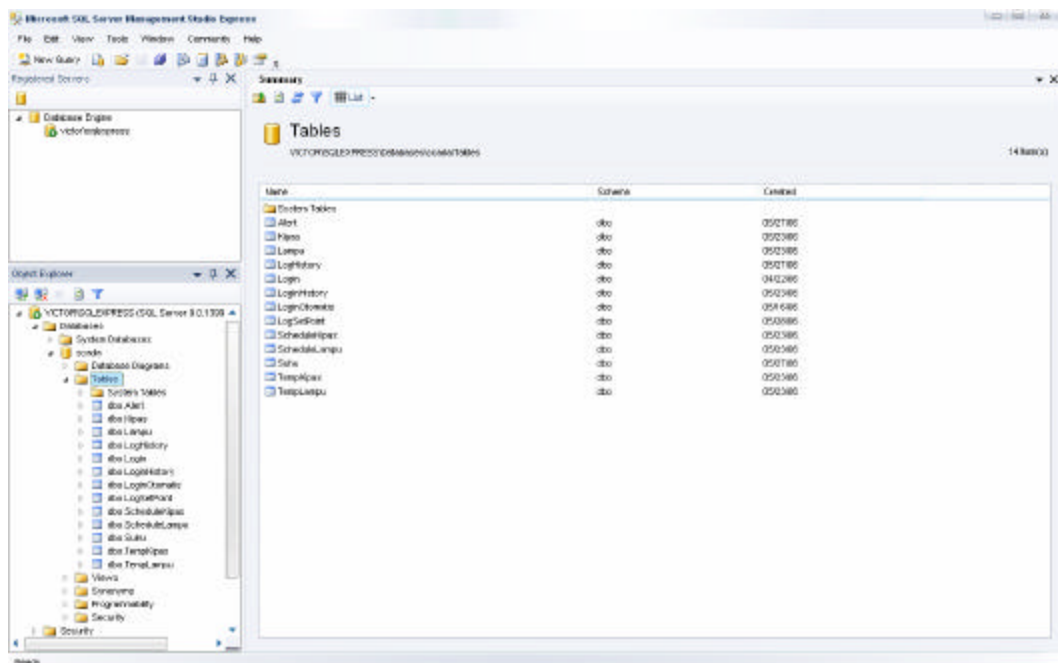


4. IMPLEMENTASI SISTEM

Pada bab ini dibahas mengenai implementasi sistem yang mengacu pada perancangan sistem yang telah dirancang pada bab sebelumnya. Implementasi sistem ini meliputi implementasi *database* dengan menggunakan Microsoft SQL Server 2005 Express Edition, serta implementasi *interface* dan implementasi program dengan menggunakan Visual Basic.NET.

4.1. Implementasi Database

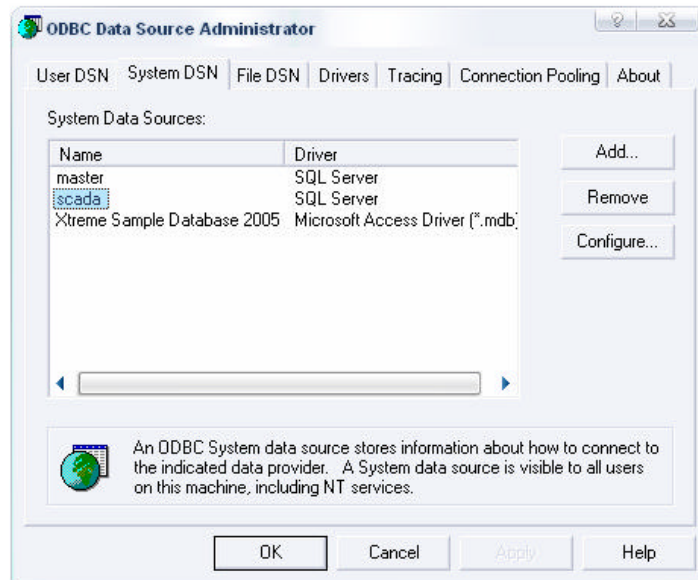
Database yang digunakan dalam membangun sistem *Mobile SCADA* ini adalah Microsoft SQL Server 2005 Express Edition. *Database* dibuat sesuai dengan perancangan struktur tabel yang telah dirancang pada Bab 3 yang kemudian pada Bab ini akan diimplementasikan ke dalam ODBC, seperti pada Gambar 4.1.



Gambar 4.1. Database dengan Microsoft SQL Server 2005 Express Edition

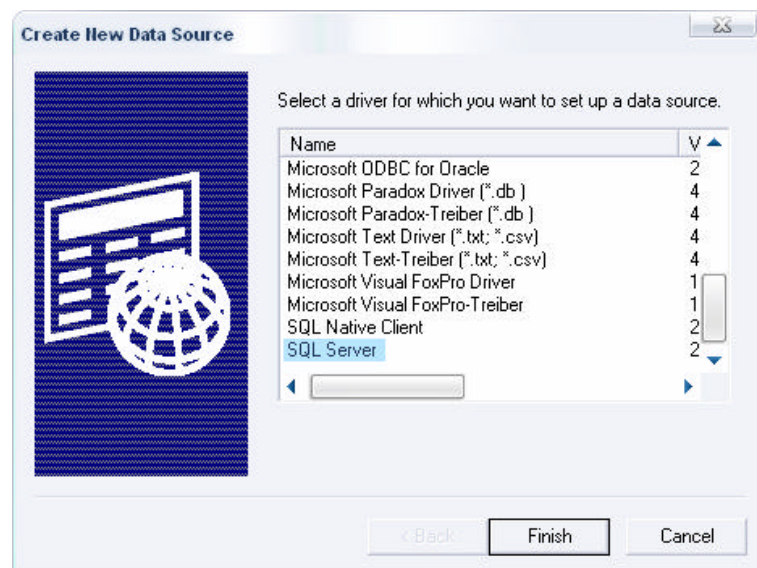
Koneksi aplikasi program PDA dengan *database* menggunakan komponen SQLlinkCE yaitu lewat ODBC di *Control Panel*. Setting database

dimulai dengan membuka ODBC dari *Control Panel*, kemudian pilih menu “*Administrative Tools*”, setelah itu pilihlah submenu “*Data Source (ODBC)*”, kemudian muncul gambar seperti pada Gambar 4.2.



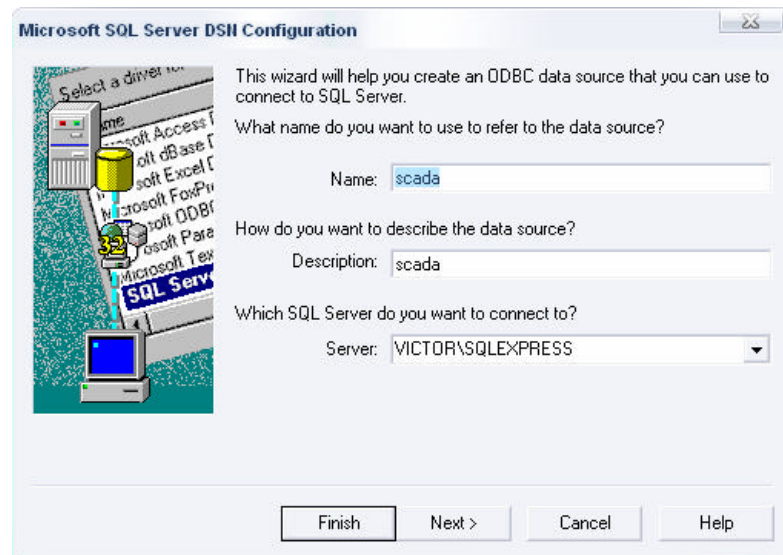
Gambar 4.2. ODBC *Data Source Administrator*

Dari ODBC *Data Source Administrator*, kemudian menambahkan *User Data Source* dengan menekan tombol *Add*, muncul *form* seperti Gambar 4.3. Kemudian memilih SQL Server dan tekan tombol *Finish*.



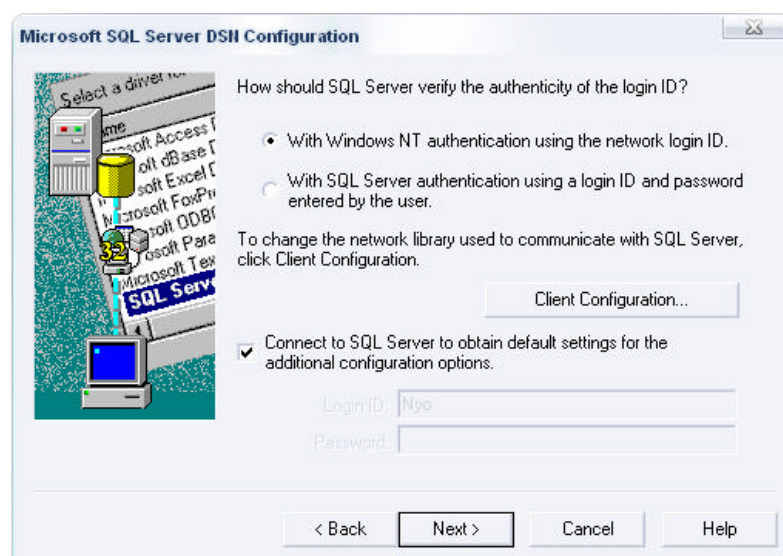
Gambar 4.3. Membuat *Data Source* baru

Setelah memilih *driver* SQL Server untuk membuat *data source* yang baru maka akan muncul *form* seperti Gambar 4.4. Kemudian, data nama *source* diisi *scada* dan *server* diisi dengan nama server SQL yang digunakan di masing-masing komputer. Setelah semua data diisi dengan lengkap, tekan tombol *Next*.



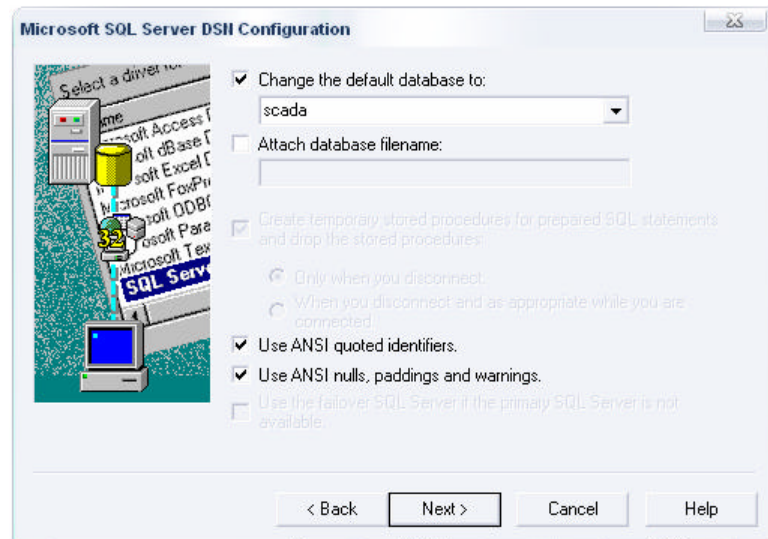
Gambar 4.4. Mengisi *Data Source* SQL Server

Setelah mengisi *data source*, kemudian pilih autentifikasi login dengan menggunakan Windows NT seperti yang terlihat pada Gambar 4.5, kemudian tekan tombol *Next*.



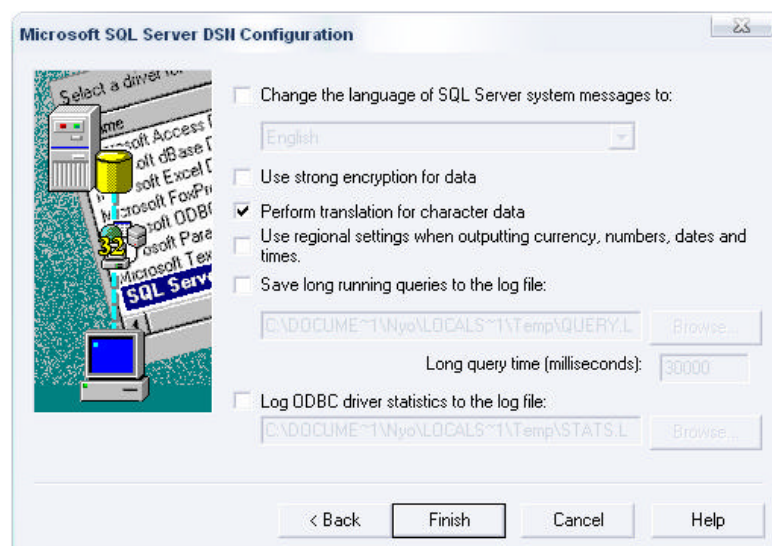
Gambar 4.5. Autentifikasi *Login* SQL Server

Setelah meng-autentifikasi login pada SQL Server, kemudian pilih database yang bernama “scada” yang sudah dibuat pada SQL Server 2005 Express Edition sebelumnya seperti yang terlihat pada Gambar 4.6, kemudian tekan tombol *Next*.



Gambar 4.6. Memilih *Database*

Setelah memilih *database*, maka muncul *form* seperti yang terlihat pada Gambar 4.7. Kemudian tekan tombol *Finish*.



Gambar 4.7. *Setting Database*

4.2. Implementasi *Interface*

Implementasi *interface* pada Tugas Akhir ini dibagi menjadi dua, yaitu *interface* pada *Server PC* dan *interface* pada program di *PDA*. Keduanya menggunakan tampilan Visual Basic.NET

4.2.1. *Interface* Pada *Server PC*

Desain *interface* yang dipergunakan dalam aplikasi program pada *server PC* terdiri dari 4 menu utama, yaitu menu *Login*, *Main Form*, *Setting* dan *About*. Penjelasan tentang *form-form* yang digunakan untuk setiap menu dapat dilihat pada Tabel 4.1, sedangkan penjelasan tiap *form* dan fungsinya dapat dilihat pada Tabel 4.2.

Tabel 4.1. Daftar Menu dan *Form Server PC* yang Digunakan

Nama Menu	Submenu	Nama <i>form</i> yang digunakan
<i>Login</i>		<i>Login</i>
<i>Main Form</i>	<i>Log User</i>	MainForm GroupBox1
	<i>Condition</i>	MainForm GroupBox3
	<i>Chart</i>	MainForm GroupBox2
<i>Setting</i>	<i>User</i>	User
	<i>Port</i>	Setting
<i>About</i>	<i>About</i>	AboutForm

Tabel 4.2. Tabel Daftar *Form Server PC*

<i>Form</i>	Fungsi
<i>Login</i>	Untuk validasi <i>user</i> (dikhhususkan <i>admin</i>)
<i>MainForm</i>	Sebagai menu utama menampilkan kondisi <i>plant</i> , log user, grafik, dan menjalankan program penghubung <i>PDA</i> ke <i>PLC</i>
<i>Log User</i>	Sebagai pusat informasi tentang aktifitas <i>user</i>
<i>Condition</i>	Sebagai informasi kondisi <i>plant</i> saat itu
<i>Chart</i>	Sebagai <i>trend</i> suhu
<i>User</i>	Untuk menambah <i>user</i> dan mengganti <i>password user</i>
<i>Setting</i>	Untuk menyeting <i>port</i> yang digunakan
<i>AboutForm</i>	Untuk mengetahui tentang program

4.2.2. *Interface* Pada *PDA*

Desain *interface* yang dipergunakan dalam aplikasi program pada *PDA* terdiri dari 7 menu utama, yaitu menu master, laporan, perhitungan, *user*, *admin*,

help dan *log out*. Penjelasan tentang *form-form* yang digunakan untuk setiap menu dapat dilihat pada Tabel 4.3, sedangkan penjelasan tiap *form* dan fungsinya dapat dilihat pada Tabel 4.4.

Tabel 4.3. Daftar Menu dan *Form* PDA yang Digunakan

Nama Menu	Submenu	Nama <i>form</i> yang digunakan
Start	Connect	
	Log In/Log Out	Login
	Change Password	ChangePassword
	Setting	TP_Setting
	Exit	
Main Form	Main	TP_Main
	Setting	TP_Setting
	Condition	TP_Condition
	Control Lamp	ControlLamp
	Control Fan	ControlFan
	Menu	TP_Menu
	Schedule Lamp	ScheduleLamp
	Schedule Fan	ScheduleFan
	Set Point Temperature	SetTemperature
	History	TP_History
Help	About	FrmAbout
	How To Use	FrmHelp

Tabel 4.4. Tabel Daftar *Form* PDA

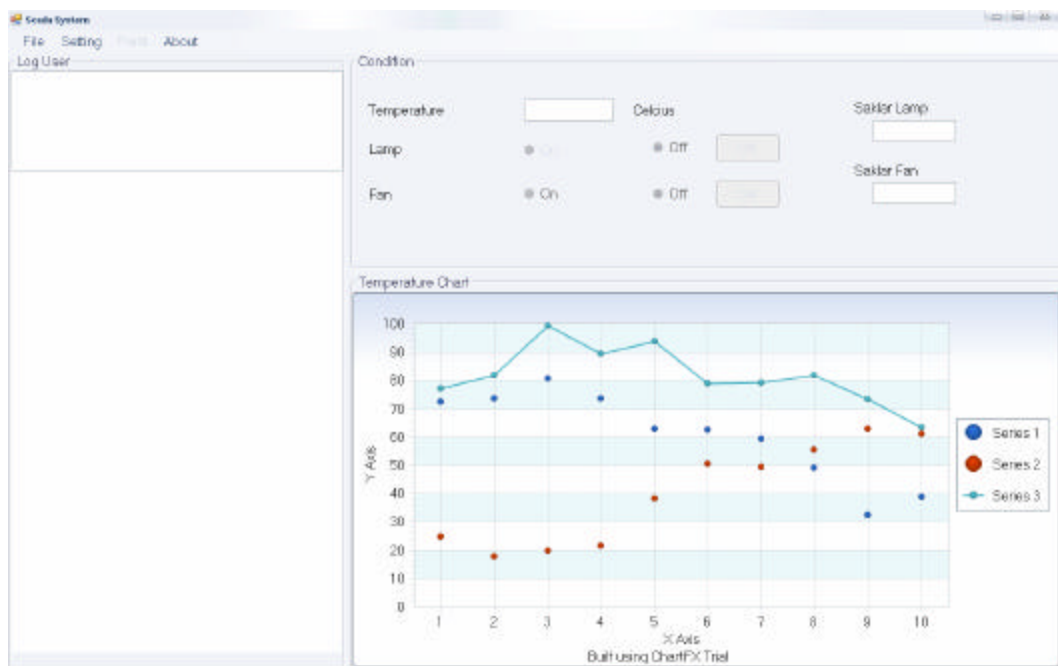
<i>Form</i>	Fungsi
Login	Untuk validasi user
ChangePassword	Fasilitas bagi user bila ingin mengganti password
TP_Setting	Untuk menyetting port, IP, dan datasource
TP_Main	Tampilan awal program.
TP_Condition	Untuk memberikan informasi tentang kondisi lampu, kipas, dan suhu
ControlLamp	Untuk memberikan perintah on/off lampu
ControlFan	Untuk memberikan perintah on/off kipas.
TP_Menu	Untuk menyediakan pilihan menu dalam pengontrolan
ScheduleLamp	Untuk membuat sebuah jadwal untuk mangaktifkan atau mematikan lampu di lain waktu
ScheduleFan	Untuk membuat sebuah jadwal untuk mangaktifkan atau mematikan kipas di lain waktu
SetTemperature	Memberikan batas atas dan bawah terhadap suhu dan pada akhirnya memberikan pemberitahuan.
TP_History	Sebagai wadah untuk melihat history data
FrmAbout	Berisi tentang informasi program
FrmHelp	Untuk membantu user tentang cara mengoperasikan.

4.3. Implementasi Program Server PC

Implementasi program bertujuan untuk menjelaskan fungsi-fungsi program dalam setiap form dan memberikan penjelasan singkat tentang perintah-perintah yang berkaitan dengan perencanaan sistem pada bab sebelumnya. Program Visual Basic.NET ini digunakan untuk melakukan koneksi atau hubungan dengan PLC atau bisa disebut PC Server. Seperti yang telah dijelaskan sebelumnya program pada Visual Basic.NET ini terdiri dari beberapa *form*, yaitu:

1. *MainForm*
2. *Login*
3. *User*
4. *Setting*

4.3.1. MainForm



Gambar 4.8. *MainForm Server*

Pada *MainForm* fungsi utamanya adalah sebagai penghubung antara PDA dan PLC atau bisa dibilang sebagai otak program. Program pada *Visual Basic.NET* ini terdiri dari beberapa bagian, yaitu:

1. Program untuk menghubungkan VB.NET dengan *database* SQL Server
2. Program untuk Awal *Load*

3. Program untuk menghasilkan 2 *digit* ASCII untuk FCS
4. Program pada *Timer1* dan 3(*Procedure* Lampu dan Kipas)
5. Program pada *Timer2* (Suhu)

4.3.1.1. Program Untuk Menghubungkan VB.NET Dengan *Database* SQL Server

Dalam pembuatan sistem untuk *server* VB.NET ini, dibutuhkan koneksi ke *database* SQL Server agar PC dapat berhubungan dengan PDA. Hal ini dibutuhkan karena untuk menjalankan sistem ini *server* akan mengambil nilai dari *database* SQL Server dan menghubungkannya ke PLC seperti yang telah dijelaskan pada bab sebelumnya. Untuk menghubungkan VB.NET dengan SQL Server maka pertama yang dilakukan adalah mengambil komponen dengan menggunakan perintah : `Imports System.Data.SqlClient`

Perintah ini berfungsi untuk pendeklarasian .NET Framework bahwa data yang diambil adalah *database* SQL Server. Untuk mengarahkan ke tabel *database* mana data yang diambil perlu adanya pendeklarasian *connection* yang sekaligus nantinya digunakan untuk membuka dan menutup *database*. Perintah pendeklarasiannya adalah :

```
Dim vconn As SqlConnection = New SqlConnection("Data
Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
Catalog = SCADA")
```

Kemudian yang kita lakukan hanyalah membuka dan menutup setiap kali kita ingin mengambil data dengan cara menuliskan perintah “`vconn.open()`” untuk membuka dan “`vconn.close()`” untuk menutup *database*. Contoh perintah lengkapnya dapat dilihat pada segmen program 1.1.

Segmen Program 4.1. Contoh Program *Save Database*

```
1. Dim vconn As SqlConnection = New SqlConnection("Data
   Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
   Catalog = SCADA")
2. vconn.Open()
3. Dim insertLamp2 As String = ("Insert into Lampu values
   ('FirstLoad', '" & CDate(Date.Now.ToShortDateString) & "', '" &
   Date.Now.ToLongTimeString & "', '" & sensor & "', 'ServerPC ',
   'Checking') ")
4. Dim msqku2 As SqlCommand = New SqlCommand(insertLamp2, vconn)
5. Dim myreader2 As SqlDataReader
6. myreader2 = msqku2.ExecuteReader
7. vconn.Close()
```

4.3.1.2. Program Untuk Awal *Load*

Ketika *MainForm* pertama kali *load*, maka hal yang pertama dijalankan adalah membuka *port* pada kabel serial (gambar 3.27).

Segmen Program 4.2. Program *Open Port*

```

1. Public miComPort As Integer
2. Public WithEvents moRS232 As Rs232
3. moRS232 = New Rs232()
4. If Rs232.IsPortAvailable(Int32.Parse(5)) Then
5.     MessageBox.Show("Port available", "Port test",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
6. Else
7.     MessageBox.Show("Port NOT available, go to Setting Page",
        "Port test", MessageBoxButtons.OK, MessageBoxIcon.Error)
8. Setting.ShowDialog()
9.     Exit Sub
10. End If
11. '// Setup parameters
12. With moRS232
13.     miComPort = 5
14.     .Port = miComPort
15.     .BaudRate = Int32.Parse(9600)
16.     .DataBit = 7
17.     .StopBit = Rs232.DataStopBit.StopBit_2
18.     .Parity = Rs232.DataParity.Parity_Even
19.     .Timeout = Int32.Parse(1500)
20. End With
21. '// Initializes port
22. moRS232.Open()
23. Timer1.Enabled = moRS232.IsOpen
24. Timer2.Enabled = moRS232.IsOpen
25. Timer3.Enabled = moRS232.IsOpen

```

Untuk membuka port dan melakukan koneksi dengan PLC, ada beberapa parameter-parameter pada VB.NET yang harus disesuaikan dengan parameter-parameter pada PLC. Dari segmen program di atas, perintah “*miComPort*” digunakan untuk menentukan *ComPort* mana yang akan digunakan untuk melakukan koneksi dengan PLC. Dalam implementasi sistem ini digunakan *ComPort* 5 untuk berkoneksi dengan PLC. Selain men-set *port* yang dipakai, harus juga di-set parameter-parameter yang lain seperti *BaudRate*, *DataBit*, *StopBit* dan *ParityBit*. Untuk *BaudRate* dalam sistem ini di-set sesuai dengan parameter dari PLC yaitu “9600” bps, *DataBit* di-set “7”, *StopBit* di-set “2” dan *ParityBit* di-set “even”. Selain itu juga di-set parameter *Timeout* yaitu sebesar “1500” ms. *Timeout* ini digunakan pada waktu dilakukan koneksi, dimana waktu timeout sebesar “1500” ms itu digunakan untuk memeriksa kembali apakah

koneksi sudah benar-benar tersambung (Koneksi PC-PLC menggunakan kabel RS232).

Kemudian ketika *port* telah terbuka maka program akan mengaktifkan semua *timer*. Selain itu program juga menuliskan perintah ke PLC untuk merubah ke kondisi *monitor* dengan memberikan perintah sebagai berikut,

Segmen Program 4.3. Program Mengaktifkan PLC

```
1. Dim data As String = "@00SC02"
2. moRS232.Write(GetFCS(data))
```

Setelah itu program akan membaca sensor dan menyimpannya ke dalam *database* untuk mengetahui keadaan terakhir dari *plant*.

Segmen Program 4.4. Program Membaca Kondisi Akhir *Plant*

```
1. ReadSensor()
2. Dim sensor, sensor1, hasil, final As String
3. hasil = GetBinary(moRS232.InputStreamString)
4. final = Mid(hasil, 1, 4)
5. sensor = Mid(final, 2, 1)
6. sensor1 = Mid(final, 1, 1)

7. Dim vconn As SqlConnection = New SqlConnection("Data
Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
Catalog = SCADA")
8. vconn.Open()
9. Dim insertLamp2 As String = ("Insert into Lampu values
('FirstLoad', '" & CDate(Date.Now.ToShortDateString) & "', '" &
Date.Now.ToLongTimeString & "', '" & sensor & "', 'ServerPC ',
'Checking') ")
10. Dim msqlku2 As SqlCommand = New SqlCommand(insertLamp2,
vconn)
11. Dim myreader2 As SqlDataReader
12. myreader2 = msqlku2.ExecuteReader
13. vconn.Close()

14. vconn.Open()
15. Dim insertFan2 As String = ("Insert into Kipas values
('FirstLoad', '" & CDate(Date.Now.ToShortDateString) & "', '" &
Date.Now.ToLongTimeString & "', '" & sensor1 & "', 'ServerPC ',
'Checking') ")
16. Dim msqlku As SqlCommand = New SqlCommand(insertFan2, vconn)
17. Dim myreader As SqlDataReader
18. myreader = msqlku.ExecuteReader
19. vconn.Close()
```

Di dalam segmen program di atas terdapat fungsi *ReadSensor()*, dimana fungsi tersebut berfungsi untuk mengirimkan perintah ke PLC untuk membaca nilai pada pin IR sepanjang satu word data dengan menggunakan perintah

“@00RR00000001”. Untuk lebih jelasnya dapat melihat segmen program di bawah ini.

Segmen Program 4.5. Fungsi *ReadSensor*

```

1. Public Sub ReadSensor()
2.     Dim dtx As String
3.     dtx = "@00RR00000001"
4.     moRS232.Write(GetFCS(dtx))
5.     moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
        Rs232.PurgeBuffers.RXClear)
6.     moRS232.RxBufferThreshold = Int32.Parse(15)
7.     moRS232.Read(Int32.Parse(15))
8. End Sub

```

Untuk menulis perintah ke PLC digunakan `moRS232.Write()`, sedangkan untuk membaca *response* dari PLC menggunakan `moRS232.Read(Int32.Parse(15))`. Maksud dari `Parse(15)` adalah program mengambil data sebanyak 15 bit sesuai dengan jumlah dari perintah PLC, sehingga *response* yang diterima oleh program menjadi utuh atau tidak terpecah-pecah.

4.3.1.3. Program Untuk Menghasilkan 2 *Digit* ASCII Untuk FCS

Seperti dijelaskan pada bab sebelumnya, untuk mengirimkan data atau memberi perintah ke PLC harus ada FCS(*Frame Check Sequence*) untuk mengecek kesalahan dalam memberi perintah kepada PLC. Untuk itu program juga harus menghasilkan ASCII untuk FCS. Program untuk menghasilkan ASCII untuk FCS ini terdapat di dalam prosedur `GetFCS`.

Segmen Program 4.6. Prosedur `GetFCS`

```

1. Private Function GetFCS(ByVal Masukan As String) As String
2. Dim Keluar As String
3. Dim y, z, d, i As Integer
4. If Masukan <> "" Then
5.     y = Len(Masukan)
6.     z = Asc(Mid(Masukan, 1))
7.     For i = 2 To y
8.         z = z Xor Asc(Mid(Masukan, i, 1))
9.     Next
10.     If Len(Hex(Str(z))) = 1 Then
11.         Keluar = Masukan + "0" + Hex(Str(z)) + "*" + Chr(13)
12.     Else
13.         Keluar = Masukan + Hex(Str(z)) + "*" + Chr(13)
14.     End If
15. Else

```

Segmen Program 4.6. Prosedur GetFCS (sambungan)

```

16.          d = MsgBox("Data belum diisi !", vbOKOnly +
vbExclamation, "Error")
17.          End If
18.          GetFCS = Keluar
19.  End Function

```

Perlu diadakan XOR pada setiap data, secara berurutan, sehingga akhirnya menghasilkan ASCII yang diinginkan sebagai FCS. Jika ternyata ASCII yang dihasilkan adalah 1 *digit*, maka perlu ditambahkan “0” di depan hasil dari XOR yang kedua. Lalu, kemudian data setelah digabung dengan FCS lalu dimasukkan ke dalam variabel *GetFCS*.

4.3.1.4. Program Pada *Timer1* dan 3 (*Procedure* Lampu dan Kipas)

Program pada *timer1* dan 3 akan membaca kondisi *output* dan *input* pada PLC dan *database* setiap 1000 ms menggunakan komponen *timer*. *Timer1* ditujukan untuk pengontrolan lampu sedangkan *timer2* ditujukan untuk pengontrolan lampu. Seperti yang telah dijelaskan pada perancangan sistem bahwa kedua *timer* ini memiliki logika yang sama hanya tabel *database* dan alamat pada PLC yang berbeda. Untuk itu penulis akan menjelaskan salah satu *timer* (*timer1*) sebagai contoh untuk menjelaskan secara lebih rinci. Terdapat tiga langkah (gambar 3.28 dan gambar 3.29) dalam program ini, yaitu pengontrolan melalui saklar (manual), pengontrolan melalui *schedule*, pengontrolan melalui langsung dari *device* (*direct*).

Pertama pengontrolan melalui saklar, yang dilakukan pertama kali adalah program mengambil data terakhir *plant*.

Segmen Program 4.7. Program Pengontrolan Saklar

```

2. vconn.Open()
3. Dim selectLlamp As String = ("select * from Lampu order by
tanggal desc, waktu desc")
4. Dim msq1 As SqlCommand = New SqlCommand(selectLlamp, vconn)
5. Dim reader As SqlDataReader
6. reader = msq1.ExecuteReader
7. If reader.Read Then
8.     Lkondisi = reader("Kondisi")
9.     Lampu(Lkondisi)
11.End If
12.vconn.Close()

```

Kemudian program mengambil nilai dari sensor, apabila nilai sensor tidak sama dengan nilai kondisi akhir maka nilai sensor akan disimpan ke dalam *database* dengan diberi keterangan bahwa *plant* dinyalakan melalui saklar.

Segmen Program 4.8. Program Pengecekan Kontrol Saklar

```

1. Public Sub Lampu(ByVal Lkondisi As String)
2. Dim sensor, hasil, final, saklar As String
3. ReadSensor()
4. hasil = GetBinary(moRS232.InputStreamString)
5. final = Mid(hasil, 1, 4)
6. saklar = Mid(final, 4, 1)
7. moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
   Rs232.PurgeBuffers.RXClear)
8. ReadSensor()
9. TextBox3.Text = saklar
10. sensor = Mid(final, 2, 1)
11. TextBox2.Text = sensor
12. If sensor <> Lkondisi Then
13.     Dim vconn As SqlConnection = New SqlConnection("Data
   Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
   Catalog = SCADA")
14.     vconn.Open()
15.     Dim insertLamp1 As String = ("Insert into Lampu values
   ('Unknown', '" & CDate(Date.Now.ToShortDateString) & "', '" &
   Date.Now.ToLongTimeString & "', '" & sensor & "', 'Saklar',
   'Manual' )")
16.     Dim msqlku1 As SqlCommand = New SqlCommand(insertLamp1,
   vconn)
17.     Dim myreader1 As SqlDataReader
18.     myreader1 = msqlku1.ExecuteReader
19.     vconn.Close()
20. Else
21. Exit Sub
22. End If
23. End Sub

```

Untuk mengambil nilai sensor pada segmen program di atas, digunakan perintah `mid (final, 2, 1)` yang artinya dari 4 bit data yang ada diambil data ke 2 sebanyak 1 bit. Sehingga dari pemecahan tersebut dapat dipilah antara saklar dan sensor.

Kedua adalah pengecekan pada *schedule* yang aktif, cara pengecekannya adalah dengan membandingkan tanggal *execute* dan waktu *execute* dari *database* terhadap tanggal dan waktu hari ini yang diambil dari sistem komputer. Apabila sama maka, kondisi tersebut di-*execute*. Untuk lebih jelasnya dapat dilihat pada segmen program 4.9.

Segmen Program 4.9. Program Cek *Schedule* Aktif

```

1. Public Sub scheduleLampu()
2.   Dim vconn As SqlConnection = New SqlConnection("Data
   Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
   Catalog = SCADA")
3.   vconn.Open()
4.   Dim Selectlamp As String = ("select * from ScheduleLampu where
   tanggalExecute = '" & CDate(Now.ToShortDateString) & "' and
   WaktuExecute = '" & Now.ToLongTimeString & "' ")
5.   Dim msqlku As SqlCommand = New SqlCommand(Selectlamp, vconn)
6.   Dim myreader As SqlDataReader
7.   myreader = msqlku.ExecuteReader
8.   If myreader.Read Then
9.     nama = myreader("Username")
10.    kondisi = myreader("Kondisi")
11.    device = myreader("device")
12.    vconn.Close()

```

Yang terakhir dari program ini adalah penulisan perintah secara langsung apabila PDA memberikan perintah maka akan langsung dikerjakan oleh *server*.

Segmen Program 4.10. Program *Execute* Langsung

```

1. ReadSensor()
2. hasil = GetBinary(moRS232.InputStreamString)
3. final = Mid(hasil, 1, 4)
4. saklar = Mid(final, 4, 1)
5.   If saklar = "0" Then
6.     If kondisi = "1" Then
7.       rTx = "@00WH00010001"
8.       moRS232.Write(GetFCS(rTx))
9.       moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
   Rs232.PurgeBuffers.RXClear)
10.    vconn.Open()
11.    ElseIf kondisi = "0" Then
12.      rTx = "@00WH00010000"
13.      moRS232.Write(GetFCS(rTx))
14.      moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
   Rs232.PurgeBuffers.RXClear)
15.    End If
16.    ElseIf saklar = "1" Then
17.      If kondisi = "1" Then
18.        rTx = "@00WH00010000"
19.        moRS232.Write(GetFCS(rTx))
20.        moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
   Rs232.PurgeBuffers.RXClear)
21.      ElseIf kondisi = "0" Then
22.        rTx = "@00WH00010001"
23.        moRS232.Write(GetFCS(rTx))
24.        moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or
   Rs232.PurgeBuffers.RXClear)
25.      End If
26.    End If

```

Sedangkan untuk memilah mana perintah yang telah dilaksanakan dan gagal dilaksanakan serta memasukkannya ke dalam *database alert* atau lampu, maka digunakan segmen program di bawah ini. Apabila Program gagal dilaksanakan maka akan disimpan ke *database alert* tetapi apabila berhasil maka disimpan ke *database lampu*. Cara membedakannya dengan membandingkan apabila sensor tidak sama dengan kondisi yang diminta maka artinya perintah gagal dilaksanakan dan sebaliknya.

Segmen Program 4.11. Program Apabila Perintah Gagal Dilaksanakan

```

1. ReadSensor()
2. hasil = GetBinary(moRS232.InputStreamString)
3. final = Mid(hasil, 1, 4)
4. sensor = Mid(final, 2, 1)
5. If sensor <> kondisi Then
6.   vconn.Open()
7.   Dim insertAlert1 As String = ("Insert into Alert values ( '" &
      CDate(Date.Now.ToShortDateString) & "', '" &
      Date.Now.ToLongTimeString & "', 'lampu', 'no_on')")
8.   Dim msqлку2 As SqlCommand = New SqlCommand(insertAlert1, vconn)
9.   Dim myreader2 As SqlDataReader
10.  myreader2 = msqлку2.ExecuteReader
11.  vconn.Close()
12. Else
13.  vconn.Open()
14.  Dim insertLamp2 As String = ("Insert into Lampu values ('" &
      nama & "', '" & CDate(Date.Now.ToShortDateString) & "', '" &
      Date.Now.ToLongTimeString & "', '" & sensor & "', '" & device &
      "', '" & tipe & "') ")
15.  Dim msqлку2 As SqlCommand = New SqlCommand(insertLamp2,
      vconn)
16.  Dim myreader2 As SqlDataReader
17.  myreader2 = msqлку2.ExecuteReader
18.  vconn.Close()
19. End If

```

4.3.1.5. Program pada Timer2 (Suhu)

Program pada *timer2* akan membaca kondisi *input* pada PLC setiap 999 ms menggunakan komponen *timer*. *Timer* akan membaca nilai dari sensor suhu lalu menyimpannya ke dalam *database*.

Segmen Program 4.12. Program Timer2

```

1. Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e
   As System.EventArgs) Handles Timer2.Tick
2.   ReadSensor()
3.   TextBox1.Text = ReadSuhu(moRS232.InputStreamString).ToString

```

Segmen Program 4.12. Program *Timer2* (sambungan)

```

4. Dim vconn As SqlConnection = New SqlConnection("Data
   Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
   Catalog = SCADA")
5. vconn.Open()
6. Dim insertSuhu As String = ("Insert into suhu values
   ('server', '" & CDate(Date.Now.ToShortDateString) & "', '" &
   Date.Now.ToLongTimeString & "', '" & TextBox1.Text & "') ")
7. Dim msqtku As SqlCommand = New SqlCommand(insertSuhu, vconn)
8. Dim myreader As SqlDataReader
9. myreader = msqtku.ExecuteReader
10. vconn.Close()
11. End Sub

```

Digunakan fungsi *ReadSuhu()* untuk memecah-mecah data dan mengkonversikannya ke dalam bentuk desimal. Sehingga data yang disimpan nantinya akan lebih mudah.

Segmen Program 4.13. Fungsi *Read Suhu*

```

1. Private Function ReadSuhu(ByVal inputan As String) As String
2.     Dim a, c, x As Integer
3.     Dim b As String
4.     If inputan <> "" Then
5.         a = Len(inputan)
6.         b = Mid(inputan, 10, 1)
7.         If b = "F" Then
8.             c = 15
9.         ElseIf b = "E" Then
10.            c = 14
11.        ElseIf b = "D" Then
12.            c = 13
13.        ElseIf b = "C" Then
14.            c = 12
15.        ElseIf b = "B" Then
16.            c = 11
17.        ElseIf b = "A" Then
18.            c = 10
19.        ElseIf b = "9" Then
20.            c = 9
21.        ElseIf b = "8" Then
22.            c = 8
23.        ElseIf b = "7" Then
24.            c = 7
25.        ElseIf b = "6" Then
26.            c = 6
27.        ElseIf b = "5" Then
28.            c = 5
29.        ElseIf b = "4" Then
30.            c = 4
31.        ElseIf b = "3" Then
32.            c = 3
33.        ElseIf b = "2" Then
34.            c = 2
35.        ElseIf b = "1" Then

```

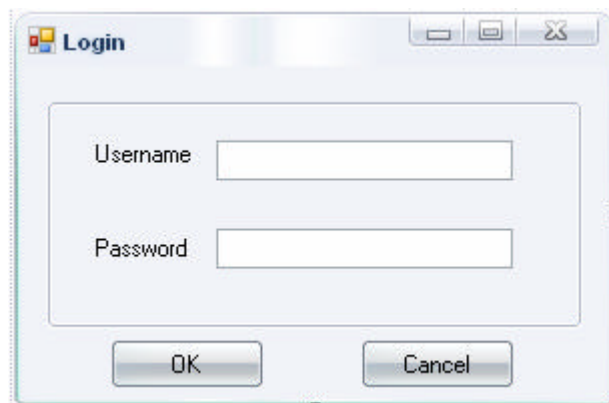
Segmen Program 4.13. Fungsi *Read Suhu* (sambungan)

```

36.         c = 1
37.     ElseIf b = "0" Then
38.         c = 0
39.     End If
40.     x = (100 / 15) * c
41. End If
42.     ReadSuhu = x
43. End Function

```

4.3.2. Login Server



Gambar 4.9. Login Server

Pada *form* ini *user (admin)* memasukkan *username* dan *password* sesuai dengan *field* yang tersedia, kemudian menekan tombol *OK* untuk masuk dalam sistem atau menekan tombol *Cancel* jika batal menggunakan aplikasi. Apabila *user* menekan tombol *OK* maka akan menjalankan *procedure Btn_Ok_Click*, sedangkan apabila menekan tombol *Cancel* maka akan menjalankan *procedure Btn_Cancel_Click*. Keterangan segmen program dapat dilihat pada Tabel 4.5.

Tabel 4.5. Daftar *Procedure Login Server*

<i>Procedure</i>	<i>Keterangan</i>
<i>procedure Btn_Ok_Click</i>	Mencocokkan <i>username</i> dan <i>password</i> sesuai dengan data yang terdapat dalam <i>database</i> .
<i>procedure Btn_Cancel_Click</i>	Menutup <i>Login</i>

Segmen Program 4.14. *Procedure Btn_Ok_Click*

```

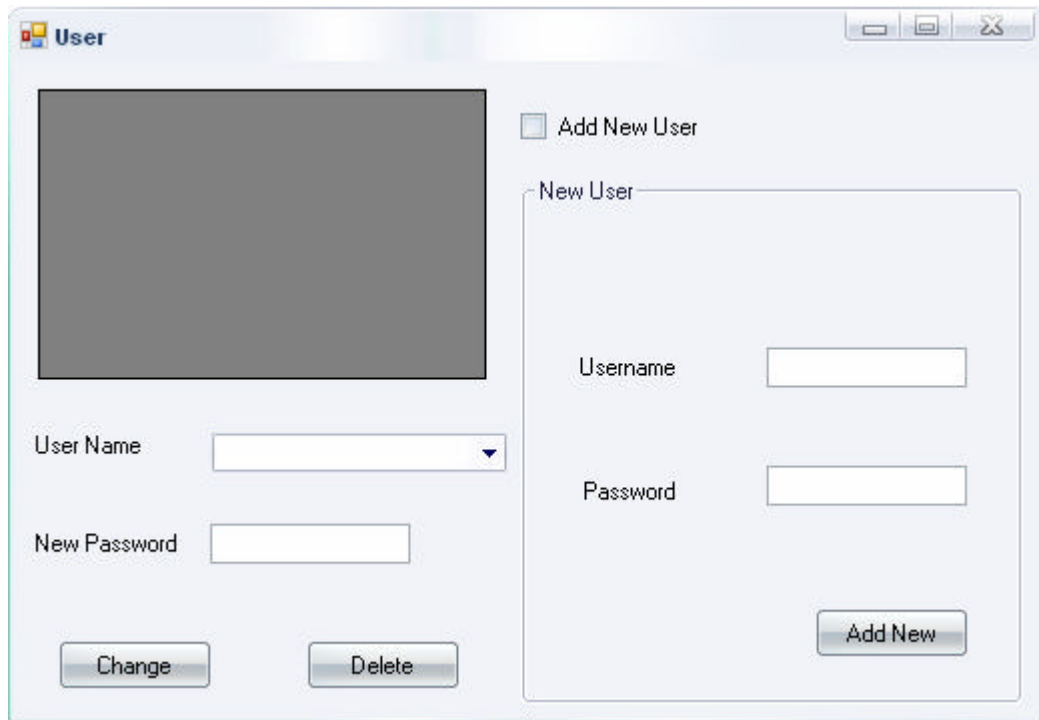
1. Private Sub Btn_Ok_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Btn_Ok.Click
2.     Dim vconn As SqlConnection = New SqlConnection("Data
   Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
   Catalog = SCADA")
3.     vconn.Open()
4.     Dim selex As String = ("Select * from Login where
   (username = 'admin' and password='" & TB_Password.Text & "'" and
   status='0') ")
5.     Dim msqlku As SqlCommand = New SqlCommand(selex, vconn)
6.     Dim myreader As SqlDataReader
7.     myreader = msqlku.ExecuteReader
8.     If myreader.Read Then
9.         vconn.Close()
10.        MessageBox.Show("Login Succeded")
11.        Uname = TB_Uname.Text
12.        vconn.Open()
13.        Dim update As String = ("Update Login set status ='1'
   where (username ='" & Uname & "'"")
14.        Dim msqlku2 As SqlCommand = New SqlCommand(update,
   vconn)
15.        Dim myreader2 As SqlDataReader
16.        myreader2 = msqlku2.ExecuteReader
17.        vconn.Close()
18.
19.    Else
20.        MessageBox.Show("Please check Username, Password, or it
   might be you haven't set logout")
21.    End If
22. End Sub

```

Pada *procedure Btn_Ok_Click* ini dilakukan pengecekan *user* yang diatur bahwa yang hanya bisa melakukan *login* adalah *admin*, apakah *username* serta *password* yang telah dimasukan sesuai dengan yang ada dalam tabel *Login* pada *database* (segmen program baris ke 3-12). Apabila *admin* telah melakukan *login* maka *admin* tidak dapat menggunakan di *device* lain karena pada status *database* login telah di-*update* menjadi bernilai “1”.

4.3.3. *User*

Dalam Program ini yang berhak untuk menambahkan, mengganti *password*, dan menghapus *user* adalah *admin*.



Gambar 4.10. User

Procedure yang digunakan adalah *Button1_Click*, *Button2_Click*, *Button3_Click*.

Segmen Program 4.15. *Procedure Button1_Click*

```

1. Private Sub Button1_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Button1.Click
2.     If TextBox3.Text = "" Then
3.         MsgBox("Please insert username")
4.     ElseIf TextBox4.Text = "" Then
5.         MsgBox("Please insert password")
6.     ElseIf TextBox3.Text <> "" And TextBox4.Text <> "" Then
7.         Dim vconn As SqlConnection = New SqlConnection("Data
Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
Catalog = SCADA")
8.         Dim insert As String = ("Insert into Login values ('" &
   TextBox3.Text & "', '" & TextBox4.Text & "') ")
9.         Dim msqlku2 As SqlCommand = New SqlCommand(insert, vconn)
10.        Dim myreader2 As SqlDataReader
11.        myreader2 = msqlku2.ExecuteReader
12.        vconn.Close()
13.    End If
14. End Sub

```

Program di atas bertujuan untuk menyimpan nama *user* dan *password* ke dalam *database Login*.

Segmen Program 4.16. *Procedure Button2_Click*

```

1. Private Sub Button2_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Button2.Click
2.     If ComboBox1.Text <> "" And TextBox1.Text <> "" Then
3.         Dim vconn As SqlConnection = New SqlConnection("Data
Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
Catalog = SCADA")
4.         vconn.Open()
5.         Dim update As String = ("Update ScheduleLampu set password
= '" & TextBox1.Text & "' where username = '" & ComboBox1.Text
& "'")
6.         Dim sql1 As SqlCommand = New SqlCommand(update, vconn)
7.         Dim reader As SqlDataReader
8.         reader = sql1.ExecuteReader
9.         vconn.Close()
10.    End If
11.    End Sub

```

Program di atas bertujuan untuk mengganti *password* lama ke baru lalu disimpan ke dalam *database Login*. Sedangkan *Procedure Button3_Click* dpada segmen program di bawah ini berfungsi untuk menghapus *user* dari data di *database Login*.

Segmen Program 4.17. *Procedure Button3_Click*

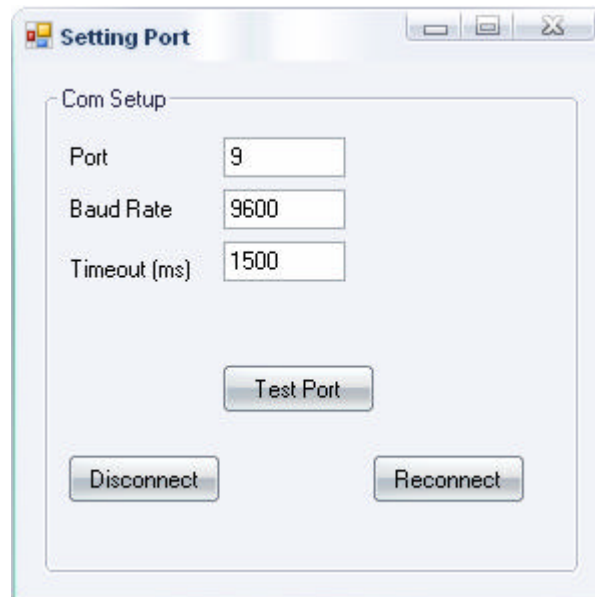
```

1. Private Sub Button3_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Button3.Click
2.     Dim vconn As SqlConnection = New SqlConnection("Data
Source=VICTOR\SQLEXPRESS; integrated security =SSPI; Initial
Catalog = SCADA")
3.     If ComboBox1.Text <> "" Then
4.         vconn.Open()
5.         Dim Delete As String = ("Delete from Login where username
= '" & ComboBox1.Text & "' ")
6.         Dim sql As SqlCommand = New SqlCommand(Delete, vconn)
7.         Dim baca As SqlDataReader
8.         baca = sql.ExecuteReader
9.         vconn.Close()
10.    Else
11.        MsgBox("Please chose username to delete")
12.    End If
13.    End Sub

```

4.3.4. *Setting*

Program ini berfungsi sebagai tempat pengesetan *port* apabila pada saat *connect* pertama gagal karena adanya perbedaan *setting* parameter.



Gambar 4.11. Setting

Berikut adalah Keterangan segmen program dapat dilihat pada Tabel 4.6.

Tabel 4.6. Daftar *Procedure Setting Port*

<i>Procedure</i>	<i>Keterangan</i>
<i>procedure BtnTest_Click</i>	Untuk mengecek kembali koneksi <i>port</i>
<i>procedure Button1_Click</i>	Untuk menyambungkan kembali koneksi
<i>procedure Button2_Click</i>	Untuk menutup koneksi

Segmen Program 4.18. *procedure BtnTest_Click*

```

1. Private Sub BtnTest_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles BtnTest.Click
2.     Try
3.         If Rs232.IsPortAvailable(Int32.Parse(TBPort.Text)) Then
4.             MessageBox.Show("Port available", "Port test",
               MessageBoxButtons.OK, MessageBoxIcon.Information)
5.         Else
6.             MessageBox.Show("Port NOT available", "Port test",
               MessageBoxButtons.OK, MessageBoxIcon.Error)
7.         End If
8.     Catch ex As Exception
9.         MessageBox.Show("Port test failed", "Port test",
               MessageBoxButtons.OK, MessageBoxIcon.Error)
10.    End Try
11. End Sub

```

Pada program di atas digunakan untuk mengetes apakah *port* yang telah dimasukkan itu ada atau tidak, digunakan perintah baris ke-3. *procedure*

Button1_Click harus dijalankan apabila pertama kali mencoba namun ingin mencoba kembali dengan *port* lain.

Segmen Program 4.19. *procedure Button1_Click*

```

1. Private Sub Button1_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Button1.Click
2.     MainForm.moRS232 = New Rs232()
3.     If Rs232.IsPortAvailable(Int32.Parse(TBPort.Text)) Then
4.         MessageBox.Show("Port available", "Port test",
           MessageBoxButtons.OK, MessageBoxIcon.Information)
5.     Else
6.         MessageBox.Show("Port NOT available, go to Setting Page",
           "Port test", MessageBoxButtons.OK, MessageBoxIcon.Error)
7.     End If
8. '/// Setup parameters
9.     With MainForm.moRS232
10.        MainForm.miComPort = TBPort.Text
11.        .Port = MainForm.miComPort
12.        .BaudRate = Int32.Parse(TBBaudRate.Text)
13.        .DataBit = 7
14.        .StopBit = Rs232.DataStopBit.StopBit_2
15.        .Parity = Rs232.DataParity.Parity_Even
16.        .Timeout = Int32.Parse(1500)
17.    End With
18. '/// Initializes port
19.    MainForm.moRS232.Open()
20. End Sub

```

Berikutnya adalah *Button2_Click* yang digunakan untuk memutus koneksi serial kabel.

Segmen Program 4.20. *procedure Button2_Click*

```

1. Private Sub Button2_Click(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles Button2.Click
2.     If Not MainForm.moRS232 Is Nothing Then
3.         '/// Disables Events if active
4.         MainForm.moRS232.DisableEvents()
5.         If MainForm.moRS232.IsOpen Then MainForm.moRS232.Close()
6.     End If
7. End Sub

```

4.4. Implementasi Program PDA

Implementasi program bertujuan untuk menjelaskan fungsi-fungsi program dalam setiap form dan memberikan penjelasan singkat tentang perintah-perintah yang berkaitan dengan perencanaan sistem pada bab sebelumnya. Dengan mengacu dari tabel 4.4, Jumlah *interface* yang akan diulas adalah sebanyak 14 tampilan.

4.4.1. Login PDA

Gambar 4.12. Login PDA

Pada *form* ini *user* memasukan *username* dan *password* sesuai dengan *field textbox* yang tersedia, kemudian menekan tombol *OK* untuk masuk dalam sistem atau menekan tombol *Cancel* jika batal menggunakan aplikasi. Apabila *user* menekan tombol *OK* maka akan menjalankan *procedure* *Btn_OK_Click*, sedangkan apabila menekan tombol *Cancel* maka akan menjalankan *procedure* *Btn_Cancel_Click*. Keterangan serta segmen program dapat dilihat pada Tabel 4.7. dan segmen program 4.21.

Tabel 4.7. Daftar *Procedure* Login PDA

<i>Procedure</i>	<i>Keterangan</i>
<i>procedure Btn_OK_Click</i>	Mencocokkan <i>username</i> dan <i>password</i> sesuai dengan data yang terdapat dalam <i>database Login</i> .
<i>procedure Btn_Cancel_Click</i>	Menutup <i>Login</i>

Segmen Program 4.21. *Procedure Btn_OK_Click*

1. If MI_inout.Text = "Login" Then
2. Dim frmLogin As New Login 'create form login
3. frmLogin.ShowDialog() 'tampilkan form login
4. User1 = frmLogin.TextBox1.Text
5. MsgBox(User1)

Segmen Program 4.21. *Procedure Btn_OK_Click* (sambungan)

```

6. Password1 = frmLogin.TextBox2.Text
7. MsgBox>Password1)
8. If frmLogin.DialogResult = Windows.Forms.DialogResult.OK Then
9.     sqlLink.Sql = "SELECT * FROM [scada].[dbo].[login] WHERE
[username] = '" & User1 & "' AND [password] = '" & Password1 &
"'AND [status]='0'"
10.    If sqlLink.Execute_Query() <> 0 Then
11.        MsgBox(sqlLink.Error_Message)
12.        Exit Sub
13.    Else
14.        MsgBox("Login Succeeded
15.    End If
16.    sqlLink.Sql = "INSERT INTO
[scada].[dbo].[LoginHistory]([Nama],[DateLogin],[TimeLogin],[Da
teLogout],[TimeLogout],[LamaLogin],[Device]) VALUES ('" & User1
& "', '" & CDate(Now.ToShortDateString) & "', '" &
Now.ToLongTimeString & "', '" & CDate(Now.ToShortDateString) &
"', 'Aktif', '00:00:00', 'PDA')"
17.    If sqlLink.Execute_Sql() <> 0 Then
18.        MsgBox("Failed to access datasource, repeat the
process", MsgBoxStyle.Information, "Error Message")
19.        Exit Sub
20.    End If
21.    sqlLink.Sql = "UPDATE [scada].[dbo].[Login] SET [Status] =
'" & 1 & "' WHERE [Username] = '" & User1 & "'"
22.    If sqlLink.Execute_Sql() <> 0 Then
23.        MsgBox("Failed to access datasource, repeat the
process", MsgBoxStyle.Information, "Error Message")
24.        Exit Sub
25.    Else
26.        MsgBox("Ok")
27.    End If
28.    If Not sqlLink.EOF Then
29.        User2 = sqlLink.Field_Value("Username").ToString()
30.        Exit Sub
31.    Else
32.        frmLogin.Dispose()
33.        MsgBox("Please retry to login", MsgBoxStyle.Information,
"Error Message")
34.        Exit Sub
35.    End If
36. End If

```

Pada program di PDA perintah yang digunakan untuk memanggil *database* adalah “sqllink.SQL”, karena komponen yang digunakan adalah SQLLinkCE. Pada awal segmen program di atas terdapat perintah “if MI_Inout.text = “Login” ”, perintah itu digunakan apabila *user* menekan tombol MI_Inout di *MenuItem1* dengan tulisan “Login” sebab apabila *user* telah melakukan proses *login* maka tulisan tersebut akan berubah menjadi “Logout”.

Perintah baris ke-9 pada segmen program 4.21, berfungsi untuk pengecekan *user*, apakah *username* serta *password* yang telah dimasukan sesuai

dengan yang ada dalam tabel *Login* pada *database*. Sedangkan perintah baris ke-16 digunakan untuk memasukkan informasi *user* tersebut ke dalam tabel *UserHistory* pada *database*. Setelah *user* melakukan proses tersebut, *user* tidak dapat melakukan *login* di tempat lain (program baris ke-21).

4.4.2. Change Password

Gambar 4.13. Change Password

Pada *form* ini *user* dapat mengganti *password* yang diberikan oleh *admin* pada waktu mereka mendaftar pertama kali. Pada *textbox* *username* dan *password* akan otomatis ter-*generate* apabila *user* masuk ke dalam *form* tersebut. *Button Confirm* digunakan ketika penggantian *password* ingin dilaksanakan, apabila *Button Confirm* ditekan maka akan mengaktifkan *procedure* *BT_Confirm_Click*. Sedangkan *procedure* *BT_Cancel_Click* aktif ketika *user* menekan *button cancel*.

Tabel 4.8. Daftar *Procedure Change Password*

Procedure	Keterangan
<i>procedure BT_Confirm_Click</i>	Mengganti <i>password</i> lama menjadi baru lalu menyimpannya di <i>database</i> .
<i>procedure BT_Cancel_Click</i>	Menutup <i>Form</i> .

Pada segmen program 4.22, berisi tentang perintah yang digunakan untuk mengambil *password* dari *database* dari *user* yang aktif.

Segmen Program 4.22. *Generate Otomatis Password Lama*

```

1. frmGantipswd.User = User1 'isi User dengan User
2. frmGantipswd.ok = False
3. sqlLink.Sql = "SELECT * FROM [scada].[dbo].[login] WHERE
  [username] = '" & User1 & "'"
4. If sqlLink.Execute_Query() <> 0 then
  MsgBox(sqlLink.Error_Message)
5.   Exit Sub
6. End If
7. If Not sqlLink.EOF Then
8.   password2 = sqlLink.Field_Value("Password").ToString()
9.   frmGantipswd.TB_OldP.Text = password2
10.  End If

```

Pada segmen program 4.23, berisi tentang *procedure BT_Confirm_Click*. Sebelum program ini dijalankan maka *user* harus mengisi setiap *field* yang tersedia. *Field New Password* dan *Confirmation* harus sama, apabila berbeda maka program tidak akan dijalankan.

Segmen Program 4.23. *Procedure BT_Confirm_Click*

```

1. Try
2.   Dim newpass As String = ""
3.   Dim frmGantipswd As New ChangePassword
4.   frmGantipswd.ShowDialog()
5.   newpass = frmGantipswd.TB_NewP.Text
6.   frmGantipswd.Dispose()
7.   If frmGantipswd.DialogResult = Windows.Forms.DialogResult.OK
   Then
8.     sqlLink.Sql = "UPDATE [scada].[dbo].[login] SET [password]
   = '" & newpass & "'" WHERE [username] = '" & User1 & "'"
9.     If sqlLink.Execute_Sql() <> 0 Then
       MsgBox("Failed to access datasource, repeat the process",
       MsgBoxStyle.Information, "Error Message")
10.    frmGantipswd.Dispose()
11.    Exit Sub
12.   Else
13.     MsgBox("Password has been changed")
14.     frmGantipswd.Dispose()
15.     Me.ShowDialog()
16.     frmGantipswd.Close()
17.   End If
18. End If
19. Catch ex As Exception
20.  MsgBox("Cannot create Form Change Password -> " +
  ex.ToString) 'bila proses try gagal, tampilkan errornya
21. End Try

```

4.4.3. *TP_Setting*

TP_Setting terletak pada *MainForm* hanya terbagi menjadi *tabpage* dan *TP_Setting* adalah salah satunya. Pada *TP_Setting* terdapat *textbox* yang harus diisi dalam rangka untuk melakukan koneksi dengan *database* pada *server PC*. Untuk lebih jelasnya dapat dilihat pada gambar 4.14.

Gambar 4.14 *TP_Setting*

Host dan *Port Number* diisi dengan *IP address* dari *server PC* dan *Port* yang digunakan oleh *SQLlinkCE*. Sedangkan *Data Source* diisi dengan nama *ODBC* yang telah dibuat. Setelah mengisi semua *field* maka tekan *START* lalu *Connect*. Berikut adalah segmen program yang digunakan untuk melakukan koneksi.

Segmen Program 4.24. Program Koneksi *Server PC*

```

1. Private Sub MI_Connect_Click(ByVal sender As System.Object,
   ByVal e As System.EventArgs) Handles MI_Connect.Click
2.     Notification1.Visible = True
3.     Notification1.Critical = True
4.     Notification1.Text = "Please Wait a Moment"
5.     Notification1.Caption = "Loading"
6.     If MI_Connect.Text = "Connect" Then
7.         Try
8.             'pengisian ip address dan host server
9.             sqlLink.Host = TB_host.Text.Trim
10.            sqlLink.Remote_Port = TB_port.Text.Trim

```

Segmen Program 4.24. Program Koneksi Server PC (sambungan)

```

11.         If Not sqlLink.Connect_To_Server() = 0 Then 'coba
             menghubungkan program dengan server
12.         MsgBox("Failed to connect, please check Host and Port
             Number", MsgBoxStyle.Information, "Error Connection")
13.         Awal_Load()
14.         Exit Sub
15.         End If
16.         sqlLink.Connection_String = "Provider=MSDASQL;DSN=" &
             TB_source.Text 'isi connection string dengan nama ODBC database
17.         sqlLink.Database_Password =
18.         sqlLink.Database_User = "sa"
19.         'coba membuka ODBC
20.         If sqlLink.Open_Database <> 0 Then
21.         MsgBox("There are no data source",
             MsgBoxStyle.Information, "Error Connection")
22.         MI_inout_Click(sender, e)
23.         Exit Sub 'keluar dari prosedur
24.         Else
25.         MI_Connect.Enabled = False
26.         End If

```

Di samping itu, terdapat *setting* untuk *user* apabila ingin mengaktifkan *automatic login*. Sehingga *user* tidak perlu melewati proses *login* dan proses koneksi lebih cepat. Karena PDA terkesan sebagai barang pribadi dan program ini telah terinstal di dalam PDA. Yang dilakukan program ketika *automatic user* diaktifkan yaitu menyimpannya ke dalam *database* tabel LoginOtomatis.

Segmen Program 4.25. Program Automatic Login

```

1. Private Sub CBAutoLogin_CheckStateChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    CBAutoLogin.CheckStateChanged
2.     If CBAutoLogin.Checked = True Then
3.         sqlLink.Sql = "INSERT INTO
             [scada].[dbo].[LoginOtomatis]([Username],[Password],[Date],[Tim
             e],[AutoLogin]) VALUES ('" & User1 & "'", '" & Password1 & "'",
             & CDate(Now.ToShortDateString) & "'", '" & Now.ToLongTimeString
             & "'", '1')'"
4.         If sqlLink.Execute_Sql() <> 0 Then
5.         MsgBox("Failed to access datasource, repeat the process",
             MsgBoxStyle.Information, "Error Message")
6.         Exit Sub
7.         End If
8.         End If
9.         If CBAutoLogin.Checked = False Then
10.         sqlLink.Sql = "INSERT INTO
             [scada].[dbo].[LoginOtomatis]([Username],[Password],[Date],[Tim
             e],[AutoLogin]) VALUES ('" & User1 & "'", '" & Password1 & "'",
             '" & CDate(Now.ToShortDateString) & "'", '" &
             Now.ToLongTimeString & "'", '0')'"
11.         If sqlLink.Execute_Sql() <> 0 Then

```

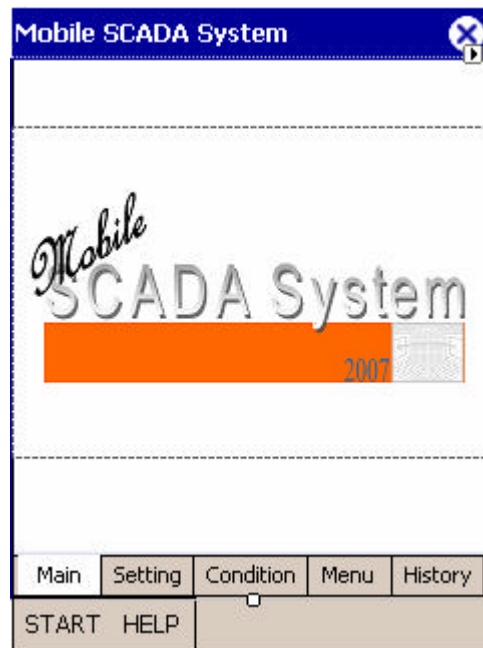
Segmen Program 4.25. Program *Automatic Login* (sambungan)

```

12.         MsgBox("Failed to access datasource, repeat the
           process", MsgBoxStyle.Information, "Error Message")
13.         Exit Sub
14.     End If
15. End If
16.     End Sub

```

4.4.4. *TP_Main*



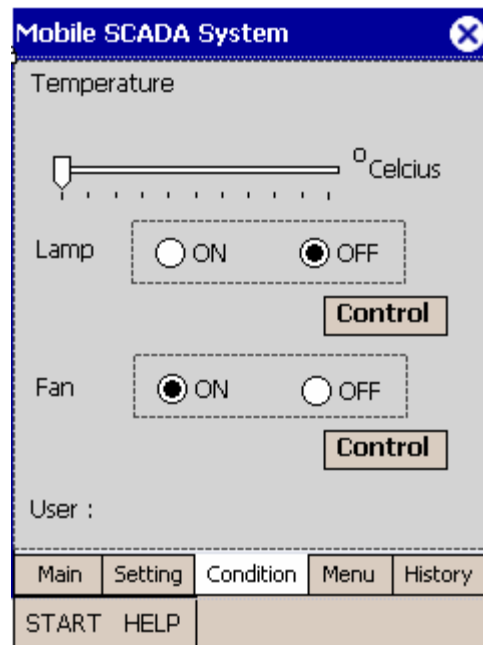
Gambar 4.15. *TP_Main*

TP_Main merupakan tampilan awal ketika program pertama kali dijalankan. Yang muncul pertama kali hanyalah logo dari program *Mobile SCADA System*. Tetapi ketika *user* telah melakukan *login* maka di atas logo akan muncul tulisan “hello” dengan nama *user* yang aktif. Sedangkan di bawah logo terdapat waktu dan tanggal sistem yang akan menjadi patokan bagi program untuk dimasukkan ke dalam *database*.

4.4.5. *TP_Condition*

TP_Condition berisi tentang informasi suhu saat ini dan kondisi *plant* yang terbaru. Di samping itu juga terdapat *button* dimana *user* dapat mengontrol *plant*

(lampu dan kipas). Apabila *button-button* tersebut ditekan maka akan tersambung ke *form ControlLamp* atau *form ControlFan*.



Gambar 4.16. *TP_Condition*

Untuk program digunakan komponen *timer2* dimana di-set 100ms dimana komponen tersebut harus mengambil data di *database*. Sehingga data yang didapat oleh *user* selalu *up-to-date*.

Segmen Program 4.26. Program *Timer2*

1. Private Sub *Timer2_Tick*(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles *Timer2.Tick*
2. *TP_condition_Suhu_click*()
3. *TP_condition_Lampu_click*()
4. *TP_condition_Kipas_click*()
5. End Sub

Pada segmen program 4.26. menggunakan fungsi suhu, lampu, dan kipas. Hal ini dilakukan untuk meringankan kerja dari komponen *timer*. Untuk lebih jelasnya dapat melihat pada segmen program 4.27. dimana berisi tentang penggabungan ketiga fungsi tersebut.

Segmen Program 4.27. Fungsi-Fungsi (Suhu, Lampu Kipas)

1. 'cek suhu
2. Public Sub *TP_condition_Suhu_click*()

Segmen Program 4.27. Fungsi-Fungsi (Suhu, Lampu Kipas) (sambungan)

```

3.  sqlLink.Sql = "select [Suhu] from [scada].[dbo].[Suhu] order
    by tanggal desc, waktu desc"
4.  If sqlLink.Execute_Query() <> 0 Then
5.  MsgBox(sqlLink.Error_Message)
6.  Exit Sub
7.  End If
8.  suhu1 = sqlLink.Field_Value("Suhu").ToString
9.  ProgressBar1.Value = CInt(suhu1)
10. lblTemp.Text = suhu1
11. End Sub

12. 'cek lampu
13. Public Sub TP_condition_Lampu_click()
14. sqlLink.Sql = "select [Kondisi] from [scada].[dbo].[Lampu]
    order by tanggal desc, waktu desc"
15. If sqlLink.Execute_Query() <> 0 Then
16. MsgBox(sqlLink.Error_Message)
17. Exit Sub
18. End If
19. kondisi = sqlLink.Field_Value("Kondisi").ToString
20. If kondisi = "0" Then
21. RBLampOFF.Checked = True
22. Else
23. RBLampON.Checked = True
24. End If
25. End Sub

26. 'cek kipas
27. Public Sub TP_condition_Kipas_click()
28. sqlLink.Sql = "select [Kondisi] from [scada].[dbo].[Kipas]
    order by tanggal desc, waktu desc"
29. If sqlLink.Execute_Query() <> 0 Then
30. MsgBox(sqlLink.Error_Message)
31. Exit Sub
32. End If
33. kondisi1 = sqlLink.Field_Value("Kondisi").ToString
34. If kondisi1 = "0" Then
35. RBFanOFF.Checked = True
36. Else
37. RBFanON.Checked = True
38. End If
39. End Sub

```

4.4.6. ControlLamp

Pada *control lamp* terdapat *radio button* dimana berfungsi untuk menyalakan atau mematikan lampu. Selain itu terdapat informasi tentang posisi saklar saat itu. Sehingga *user* dapat *visualisasi* tentang gambaran *plant*. Apabila *Apply* ditekan maka akan mengaktifkan *procedure ApplyLamp_Click* yang berfungsi untuk menyimpan kondisi yang diinginkan *user* ke dalam *database* tempLampu. Sedangkan apabila *cancel* ditekan maka *procedure CancelLamp_Click* akan aktif dan *form* akan menutup. Untuk lebih jelasnya lihat

pada tabel 4.9 dan segmen program 4.28. Saklar PLC dimaksudkan untuk melihat posisi saklar pada *plant* yang terhubung oleh PLC.

Gambar 4.17. *ControlLamp*

Tabel 4.9. Daftar *Procedure ControlLamp*

<i>Procedure</i>	<i>Keterangan</i>
<i>ApplyLamp_Click</i>	Menyimpan kondisi ke dalam <i>database</i>
<i>CancelLamp_Click</i>	Menutup <i>Form</i> .

Segmen Program 4.28. *Procedure ApplyLamp_Click*

```

1. frmcontrol.ShowDialog()
2. If frmcontrol.DialogResult = Windows.Forms.DialogResult.OK
   Then
3.     If frmcontrol.AL <> kondisi Then
4.         'insert ke temp
5.         sqlLink.Sql = "INSERT INTO [scada].[dbo].[TempLampu]
   ([username],[tanggal],[waktu],[kondisi],[Device],[tipe]) VALUES
   ('" & User1 & "', '" & CDate(Now.ToShortDateString) & "', '" &
   Now.ToLongTimeString & "', '" & frmcontrol.AL & "', 'PDA',
   'Direct')"
6.         If sqlLink.Execute_Sql <> 0 Then
7.             MsgBox(sqlLink.Error_Message)
8.             Exit Sub
9.         Else
10.            If frmcontrol.AL = "1" Then
11.                MsgBox(" Lamp ON")
12.            Else
13.                MsgBox(" Lamp OFF")
14.            End If
15.            TP_condition_Lampu_click()
16.        End If

```

Segmen Program 4.28. *Procedure ApplyLamp_Click* (sambungan)

```

17.      Else
18.          MsgBox("Please Check The Condition",
19.              MsgBoxStyle.Information, "Error Message")
19.      End If
20.      ElseIf frmcontrol.DialogResult =
21.          Windows.Forms.DialogResult.Cancel Then
21.          frmcontrol.Close()
22.      End If

```

4.4.7. *ControlFan*

Pada *controlFan* terdapat *radio button* dimana berfungsi untuk menyalakan atau mematikan kipas. Selain itu terdapat informasi tentang posisi saklar saat itu. Sehingga *user* dapat *visualisasi* tentang gambaran *plant*. Saklar PLC dimaksudkan untuk melihat posisi saklar pada *plant* yang terhubung oleh PLC.



Gambar 4.18. *ControlFan*

Apabila *Apply* ditekan maka akan mengaktifkan *procedure ApplyFan_Click* yang berfungsi untuk menyimpan kondisi yang diinginkan *user* ke dalam *database tempKipas*. Sedangkan apabila *cancel* ditekan maka *procedure CancelFan_Click* akan aktif dan *form* akan menutup. Untuk lebih jelasnya lihat pada tabel 4.10 dan segmen program 4.29.

Tabel 4.10. Daftar *Procedure ControlFan*

<i>Procedure</i>	<i>Keterangan</i>
<i>ApplyFan_Click</i>	Menyimpan kondisi ke dalam <i>database</i>
<i>CancelFan_Click</i>	Menutup <i>Form</i> .

Segmen Program 4.29. *Procedure ApplyFan_Click*

```

1. frmcontrolFan.ShowDialog()
2.   If frmcontrolFan.DialogResult =
   Windows.Forms.DialogResult.OK Then
3.       If frmcontrolFan.FL <> kondisi1 Then
4.           'insert ke temp
5.           sqlLink.Sql = "INSERT INTO [scada].[dbo].[TempKipas]
   ([username],[tanggal],[waktu],[kondisi],[device],[tipe]) VALUES
   ('" & User1 & "', '" & CDate(Now.ToShortDateString) & "', '" &
   Now.ToLongTimeString & "', '" & frmcontrolFan.FL & "', 'PDA',
   'Direct')"
6.           If sqlLink.Execute_Sql <> 0 Then
7.               MsgBox(sqlLink.Error_Message)
8.               Exit Sub
9.           Else
10.              If frmcontrolFan.FL = "1" Then
11.                  MsgBox("Fan ON")
12.              Else
13.                  MsgBox("Fan OFF")
14.              End If
15.              TP_condition_Kipas_click()
16.          End If
17.      Else
18.          MsgBox("Please Check The Condition",
   MsgBoxStyle.Information, "Error Message")
19.      End If
20.  ElseIf frmcontrolFan.DialogResult =
   Windows.Forms.DialogResult.Cancel Then
21.      frmcontrolFan.Close()
22.  End If

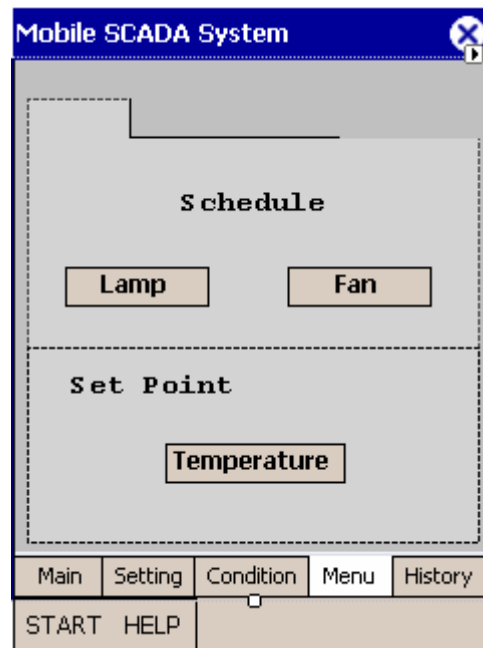
```

4.4.8. *TP_Menu*

TP_Menu berisi *button-button* yang masing-masing *button* terkoneksi pada *form* yang lain. *Button-button* tersebut antara lain:

1. *Button Schedule Lamp*, terhubung ke *ScheduleLamp*
2. *Button Schedule Fan*, Terhubung ke *ScheduleFan*
3. *Button Set Point Temperature*, Terhubung ke *SetTemperature*

Set Point Temperature disini dimaksud dengan memberikan batas bawah dan batas atas terhadap suhu saat itu. Sehingga *user* tidak perlu membuka program terus menerus untuk mengecek suhu saat itu, tetapi *user* dapat diberitahu apabila mencapai salah satu batas yang telah dimasukkan oleh *user*.



Gambar 4.19. TP_Menu

4.4.9. ScheduleLamp

Gambar 4.20. ScheduleLamp

Pada *form* ini, ketika *load* pertama kali maka akan muncul *history schedule* baik yang masih aktif ataupun yang sudah terlaksana. Apabila *user* ingin

memasukkan *schedule* baru, maka *user* harus memilih tanggal, waktu, dan kondisi yang diinginkan.

Segmen Program 4.30. Program Menampilkan *History ScheduleLamp*

```

1. sqlLink.Sql = "SELECT
   [Username],[Tanggal],[Waktu],[TanggalExecute],[WaktuExecute],[K
   ondisi],[device],[Status] FROM [scada].[dbo].[ScheduleLampu]
   order by tanggal desc, waktu desc"
2. If sqlLink.Execute_Query() <> 0 Then 'bila gagal execute query,
   tampilkan error
3. MsgBox("Failed to access Data Schedule")
4. Exit Sub
5. End If
6. frmLamp.LVSchedLamp.View = View.Details
7. frmLamp.LVSchedLamp.BeginUpdate()
8. frmLamp.LVSchedLamp.Items.Clear()
9. For number = frmLamp.LVSchedLamp.Columns.Count To 1 Step -1
10.  frmLamp.LVSchedLamp.Columns.Remove(frmLamp.LVSchedLamp.Column
   ns(number - 1))
11.  Next
12.  frmLamp.LVSchedLamp.Columns.Add("User", 100,
   HorizontalAlignment.Left)
13.  frmLamp.LVSchedLamp.Columns.Add("Date", 100,
   HorizontalAlignment.Left)
14.  frmLamp.LVSchedLamp.Columns.Add("Time", 100,
   HorizontalAlignment.Left)
15.  frmLamp.LVSchedLamp.Columns.Add("Date to Execute", 100,
   HorizontalAlignment.Left)
16.  frmLamp.LVSchedLamp.Columns.Add("Time to Execute", 100,
   HorizontalAlignment.Left)
17.  frmLamp.LVSchedLamp.Columns.Add("Condition", 100,
   HorizontalAlignment.Left)
18.  frmLamp.LVSchedLamp.Columns.Add("Device", 100,
   HorizontalAlignment.Left)
19.  frmLamp.LVSchedLamp.Columns.Add("Status", 100,
   HorizontalAlignment.Left)

20.  While Not sqlLink.EOF  'selama masih ada data, baca terus
21.  Dim lvi As New ListViewItem
22.  For number = 0 To sqlLink.Field_Value.Count
23.  lvi.Text = sqlLink.Field_Value("Username")
24.  lvi.SubItems.Add(sqlLink.Field_Value("Tanggal").ToString)
25.  lvi.SubItems.Add(sqlLink.Field_Value("Waktu").ToString)
26.  lvi.SubItems.Add(sqlLink.Field_Value("TanggalExecute").ToStr
   ing)
27.  lvi.SubItems.Add(sqlLink.Field_Value("WaktuExecute").ToStrin
   g)
28.  lvi.SubItems.Add(sqlLink.Field_Value("Kondisi").ToString)
29.  lvi.SubItems.Add(sqlLink.Field_Value("device").ToString)
30.  lvi.SubItems.Add(sqlLink.Field_Value("status ").ToString)
31.  Next
32.  frmLamp.LVSchedLamp.Items.Add(lvi)

```

Segmen Program 4.30. Program Menampilkan *History ScheduleLamp* (sambungan)

```

33.  sqlLink.MoveNext()
34.  End While
35.  frmLamp.LVSchedLamp.EndUpdate()
36.  Autosize_Listview(frmLamp.LVSchedLamp,
    frmLamp.LVSchedLamp.Columns.Count)
37.  Notification1.Visible = False
38.  frmLamp.ShowDialog()
39.  kondisi = frmLamp.KondisiLamp

```

Setelah semua di-set apabila *user* menekan *Apply* maka program akan mengaktifkan *procedure Btn_ApplyLamp_Click* sehingga program akan langsung mengirim data ke *database* tabel *ScheduleLampu*. Sebaliknya apabila *Cancel* ditekan maka *procedure Btn_CancelLamp_Click* akan aktif dan menutup *form*.

Tabel 4.11. Daftar *Procedure ScheduleLamp*

Procedure	Keterangan
<i>Btn_ApplyLamp_Click</i>	Menyimpan kondisi ke dalam <i>database</i>
<i>Btn_CancelLamp_Click</i>	Menutup <i>Form</i> .

Segmen Program 4.31. *Procedure Btn_ApplyLamp_Click*

```

1. If frmLamp.Btn_ApplyLamp.DialogResult =
  Windows.Forms.DialogResult.OK Then
2.   sqlLink.Sql = "INSERT INTO
  [scada].[dbo].[ScheduleLampu]([Username],[Tanggal],[Waktu],[Tan
  ggalExecute],[WaktuExecute],[Kondisi],[Device],[status]) VALUES
  ('' & User1 & '', '' & CDate(Now.ToShortDateString) & '', '' &
  Now.ToLongTimeString & '', '' & frmLamp.DTP_DateLamp.Value &
  '', '' & frmLamp.DTP_TimeLamp.Text & '', '' &
  frmLamp.KondisiLamp & '', 'PDA', 'Active')"
3.   If sqlLink.Execute_Sql <> 0 Then
4.     MsgBox(sqlLink.Error_Message)
5.     Exit Sub
6.   Else
7.     frmLamp.LVSchedLamp.Refresh()
8.     MsgBox(kondisi)
9.     MsgBox("Schedule Succeded")
10.  End If
11. ElseIf frmLamp.DialogResult =
  Windows.Forms.DialogResult.Cancel Then
12.  frmLamp.Close()
13.  End If

```

4.4.10. *ScheduleFan*

Pada *form* ini, ketika *load* pertama kali maka akan muncul *history schedule* baik yang masih aktif ataupun yang sudah terlaksana. Apabila *user* ingin

memasukkan *schedule* baru, maka *user* harus memilih tanggal, waktu, dan kondisi yang diinginkan. Untuk menampilkan *history schedule* kipas dapat melihat pada segmen program 4.30. yang berbeda hanyalah pemanggilan *database* yaitu tabel *ScheduleKipas*.

Gambar 4.21. *ScheduleFan*

Setelah semua di-set apabila *user* menekan *Apply* maka program akan mengaktifkan *procedure Btn_ApplyFan_Click* sehingga program akan langsung mengirim data ke *database* tabel *ScheduleKipas*. Sebaliknya apabila *Cancel* ditekan maka *procedure Btn_CancelFan_Click* akan aktif dan menutup *form*.

Tabel 4.12. Daftar *Procedure ScheduleFan*

<i>Procedure</i>	<i>Keterangan</i>
<i>Btn_ApplyFan_Click</i>	Menyimpan kondisi ke dalam <i>database</i>
<i>Btn_CancelFan_Click</i>	Menutup <i>Form</i> .

Segmen Program 4.32. *Procedure Btn_ApplyFan_Click*

1. If frmFan.Btn_ApplyFan.DialogResult = Windows.Forms.DialogResult.OK Then
2. sqlLink.Sql = "INSERT INTO
[scada].[dbo].[ScheduleKipas]([Username],[Tanggal],[Waktu],[TanggalExecute],[WaktuExecute],[Kondisi],[Device],[Status]) VALUES ('" & User1 & "', '" & CDate(Now.ToShortDateString) & "', '" & Now.ToLongTimeString & "', '" & frmFan.DTP_DateFan.Value & "',

Segmen Program 4.32. *Procedure Btn_ApplyFan_Click* (sambungan)

```

    '" & frmFan.DTP_TimeFan.Text & "', '" & kondisi &
    "','PDA','Active')"
```

3. If sqlLink.Execute_Sql <> 0 Then
4. MsgBox(sqlLink.Error_Message)
5. Exit Sub
6. Else
7. frmFan.LVSchedFan.Refresh()
8. MsgBox(kondisi)
9. MsgBox("Schedule Succeeded")
10. End If
11. ElseIf frmFan.DialogResult =
- Windows.Forms.DialogResult.Cancel Then
12. frmFan.Close()
13. End If

4.4.11. *SetTemperature*

Tabel 4.22. *SetTemperature*

SetTemperature berfungsi ketika *procedure Btn_ApplyTemp_Click* diaktifkan dengan menekan tombol *Apply*. Setelah *procedure* tersebut dijalankan maka program secara otomatis mengecek apakah suhu saat itu sama dengan nilai maksimum atau minimum yang telah disimpan (gambar 3.33). Tetapi apabila tombol *Cancel* ditekan maka *form* akan menutup.

Tabel 4.13. Daftar *Procedure SetTemperature*

Procedure	Keterangan
<i>Btn_ApplyTemp_Click</i>	Menjalankan <i>Timer 3</i>
<i>Btn_CancelTemp_Click</i>	Menutup <i>Form</i> .

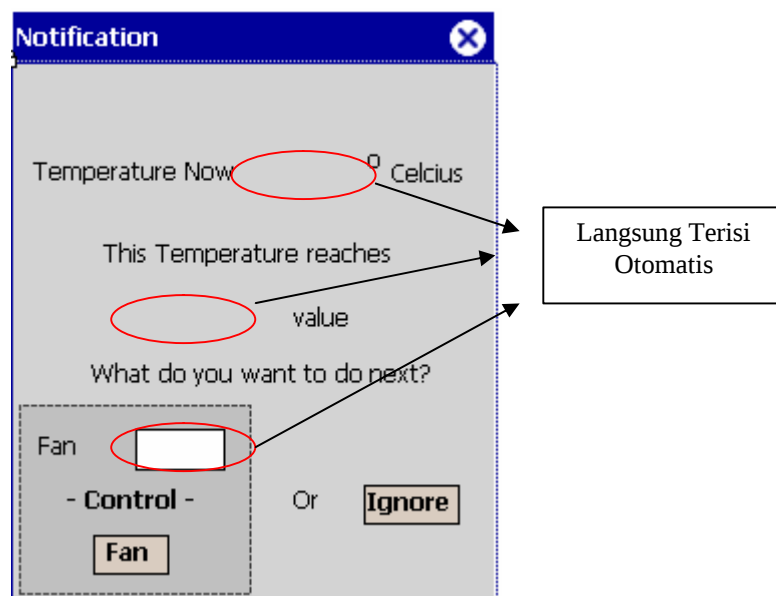
Segmen Program 4.33. *Procedure Btn_ApplyTemp_Click*

```

1. Dim frmtemp As New SetTemperature
2. frmtemp.applytemp = False
3. frmtemp.ShowDialog()
4.   If frmtemp.applytemp = True Then
5.     frmtemp.Close()
6.     maxtemp = frmtemp.maxval.ToString
7.     MsgBox(maxtemp)
8.     mintemp = frmtemp.minval.ToString
9.     MsgBox(mintemp)
10.    Timer3.Enabled = True
11.    frmtemp.Dispose()
12.  ElseIf frmtemp.DialogResult =
    Windows.Forms.DialogResult.Cancel Then
13.    frmtemp.Close()
14.  End If

```

Apabila salah satu nilai maksimum atau minimum terpenuhi, maka program menampilkan form *notification* untuk memberitahu dan menanyakan tindakan berikutnya.

Tabel 4.23. *Notification*

Pada bagian yang dilingkari pada gambar di atas akan otomatis terisi dengan suhu saat ini, kondisi yang terpenuhi (maksimum atau minimum), dan

kondisi Fan saat itu. Khusus untuk kondisi kipas, apabila yang terpenuhi kondisi maksimum dan kondisi kipas terakhir adalah sudah dalam keadaan menyala maka kipas tidak dapat dinyalakan kembali dan sebaliknya apabila kondisi kipas sudah mati dan suhu mencapai minimum maka kipas tidak dapat dimatikan.

Yang dapat dilakukan adalah hanya menekan tombol *Ignore* dan melihat ke *plant*. Tombol *Ignore* berfungsi untuk menutup *form* tersebut. Setelah *form* menutup maka nilai minimum dan maksimum akan di-reset menjadi nol.

4.4.12. *TP_History*

TP_History berisi tentang *history* setiap *plant*. Mulai dari lampu, kipas, suhu, serta *user*. Apabila *user* ingin mencari sesuai dengan tanggal dan waktu tertentu maka program telah menyediakan sebuah menu untuk melakukan pencarian tersebut. Sebab apabila tidak, maka program akan menampilkan *history* keseluruhan dari *history* yang dipilih pada saat *user* membuka program tersebut.

Gambar 4.24. *TP_History*

Field Select History digunakan untuk memilih *history* yang akan ditampilkan. Apabila *user* menekan tombol *OK* maka seperti yang telah dijelaskan sebelumnya, *history* akan keluar seluruhnya. Program yang dibuat dari semua jenis *history* serupa hanya tabel *database* yang berbeda. Sehingga penulis akan

memaparkan salah satu program (lampu) untuk mengambil dari *database* dan menampilkan ke dalam *listview*.

Segmen Program 4.34. Mengambil *Database* Apabila Tombol OK Ditekan

```

1. If CBHistory.Text = "Lamp" Then
2.     sqlLink.Sql = "SELECT
    [Username],[Tanggal],[Waktu],[Kondisi],[Device],[Tipe] FROM
    [scada].[dbo].[Lampu] order by tanggal desc, waktu desc where
    tanggal = '" & CDate(Now.ToShortDateString) & "'"
3.     If sqlLink.Execute_Query() <> 0 Then 'bila gagal
    execute query, tampilkan error
4.         MsgBox("There are no record available")
5.         Exit Sub
6.     End If

```

Berbeda dengan apabila pencarian dilakukan berdasarkan tanggal dan jam tertentu. Pada waktu penulisan perintah *database* maka di belakang diberi kondisi untuk mencari tanggal dan waktu yang diminta. Untuk lebih jelasnya lihat segmen program di bawah ini.

Segmen Program 4.35. Mengambil *Database* Apabila Diberi Kondisi

```

1. If CBHistory.Text = "Lamp" Then
2.     sqlLink.Sql = "SELECT
    [Username],[Tanggal],[Waktu],[Kondisi],[device],[tipe] FROM
    [scada].[dbo].[Lampu] where tanggal = '" & CDate(DTPDate.Value)
    & "'" and waktu between '" & DTPTimeFrom.Text & "'" and '" &
    DTPTimeTo.Text & "'"
3.     If sqlLink.Execute_Query() <> 0 Then 'bila gagal
    execute query, tampilkan error
4.         MsgBox("There are no record available")
5.         Exit Sub
6.     End If
7.     PanelAdvanced.Visible = False
8.     Btn_OKHistory.Enabled = True
9.     Button3.Enabled = False
10.    Button2.Text = "Advanced>>"

```

Yang terakhir adalah perintah untuk menampilkan di *Listview*. Pada *Listview* ini perlu dibuat tabel sendiri sehingga pemakaiannya terkesan sangat sulit karen harus mengatur kolom dan menaruh data sesuai dengan masing-masing kolom.

Segmen Program 4.36. Menampilkan *Database* di *Listview*

```

1. LVHistory.View = View.Details
2.     LVHistory.BeginUpdate()
3.     LVHistory.Items.Clear()

```

Segmen Program 4.36. Menampilkan *Database* di *Listview* (sambungan)

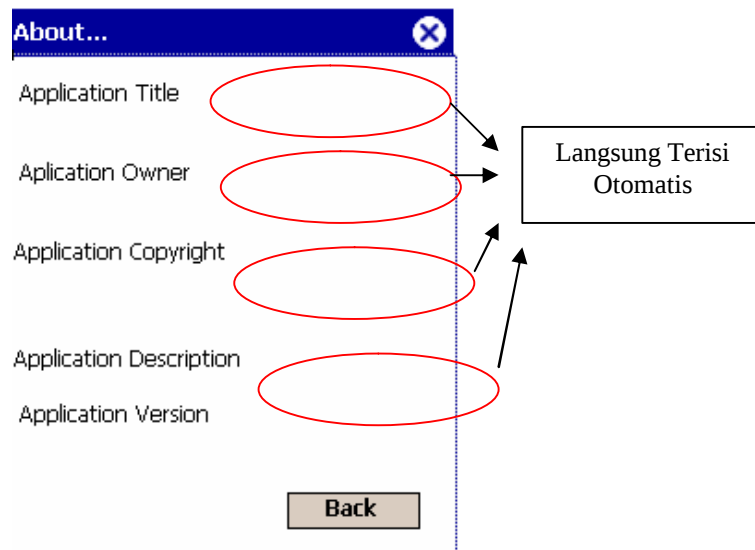
```

4.          For Ict = LVHistory.Columns.Count To 1 Step -1
5.              LVHistory.Columns.Remove(LVHistory.Columns(Ict
    - 1))
6.          Next
7.          LVHistory.Columns.Add("User", 100,
    HorizontalAlignment.Left)
8.          LVHistory.Columns.Add("Date", 100,
    HorizontalAlignment.Left)
9.          LVHistory.Columns.Add("Time", 100,
    HorizontalAlignment.Left)
10.         LVHistory.Columns.Add("Status", 100,
    HorizontalAlignment.Left)
11.         LVHistory.Columns.Add("Device", 100,
    HorizontalAlignment.Left)
12.         LVHistory.Columns.Add("Type", 100,
    HorizontalAlignment.Left)
13.
14.         While Not sqlLink.EOF 'selama masih ada data,
    baca terus
15.             Notification1.Text = "Loading, Wait a
    Moment"
16.             Dim lvi As New ListViewItem
17.             For Ict = 1 To sqlLink.Field_Value.Count
18.                 lvi.Text =
    sqlLink.Field_Value("Username")
19.                 lvi.SubItems.Add(sqlLink.Field_Value("Tanggal").ToString)
20.                 lvi.SubItems.Add(sqlLink.Field_Value("Waktu").ToString)
21.                 lvi.SubItems.Add(sqlLink.Field_Value("Kondisi").ToString)
22.                 lvi.SubItems.Add(sqlLink.Field_Value("device").ToString)
23.                 lvi.SubItems.Add(sqlLink.Field_Value("tipe").ToString)
24.             Next
25.             LVHistory.Items.Add(lvi)
26.             lvi = Nothing
27.             sqlLink.MoveNext()
28.         End While
29.         LVHistory.EndUpdate()
30.         Autosize_Listview(LVHistory,
    LVHistory.Columns.Count)

```

4.4.13. About

Form ini hanya sebagai inisialisasi program dan pemanis. Untuk mengambil setiap yang dilingkari merah pada gambar di atas maka program mengambil nilai pada bagian *assembly* pada keseluruhan program.



Gambar 4.25. About

Segmen Program 4.37. Menampilkan Inisialisasi Assembly

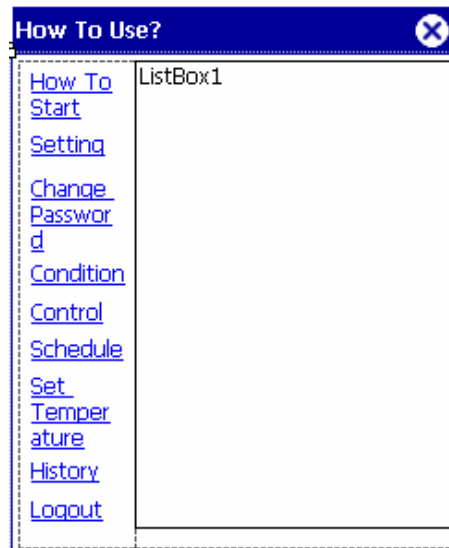
```

1. Private Sub FrmAbout_Load(ByVal sender As System.Object, ByVal
   e As System.EventArgs) Handles MyBase.Load
2.     Me.lblTitle.Text = "Application Title : " &
   ainfo.Product
3.     Me.lblOwner.Text = "Application Owner : " &
   myinfo.Trademark
4.     Me.lblVersion.Text = String.Format("Version {0}",
   dinfo.Version)
5.     Me.lblCopyright.Text = "Application Copyright : " &
   binfo.Copyright
6.     Me.lblDescription.Text = "Application Description : " &
   cinfo.Description
7. End Sub

```

4.4.14. Help

Form ini akan membantu *user* tentang bagaimana menjalankan program tersebut dengan memberikan langkah-langkah mulai dari menyambung ke *server* hingga cara menggunakan fitur-fitur yang ada di dalamnya. Apabila *user* ingin memilih tentang bagaimana menggunakan fitur yang diinginkan, maka *user* tinggal menekan tulisan di sebelah kiri program (*linklabel*). Ketika tulisan ditekan maka kotak di sebelah kanan akan muncul tulisan berisi petunjuk dari fitur yang diinginkan.



Gambar 4.26. *Help*