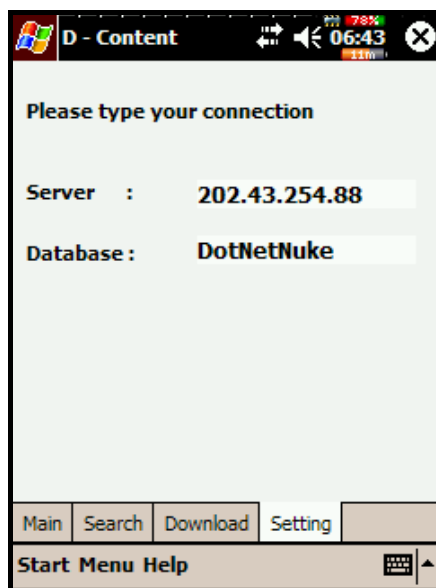


4. IMPLEMENTASI SISTEM

Pada bab ini dilakukan implementasi membahas mengenai *software* yang telah dibuat dari implementasi awal, jalan kerja sistem, dan akhir hingga keseluruhan kerja sistem, serta menjadikan aplikasi ini menjadi sebuah *installer*. Implementasi ini dilakukan untuk mengetahui apakah sistem ini sudah jalan untuk sebagaimana mestinya dan juga sesuai dengan *flowchart* yang telah dibuat. Dalam tahap implementasi dibagi menjadi beberapa bagian yaitu:

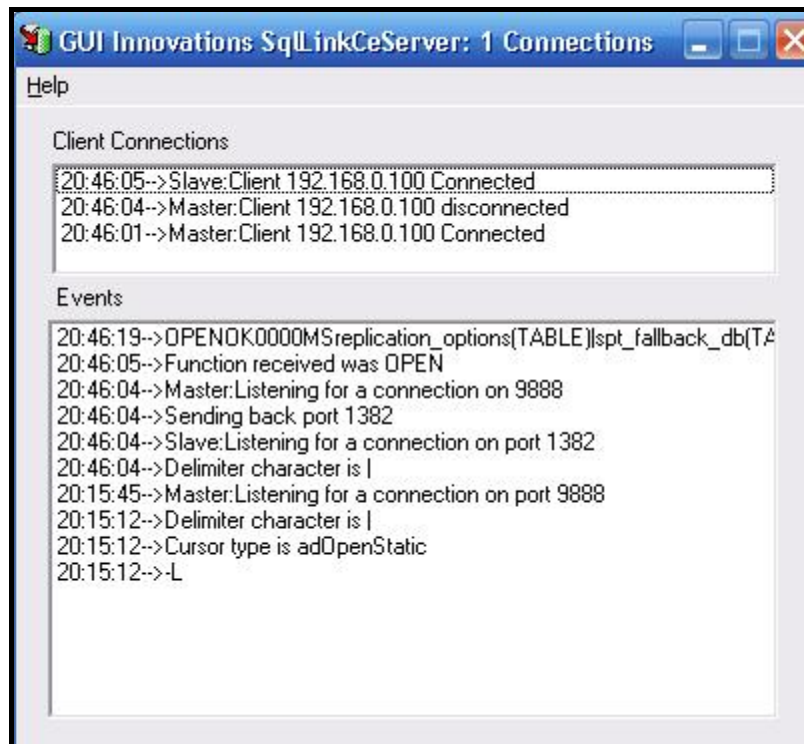
4.1. Implementasi Proses *Connect to Database*

Pertama kali *server* mengatur konfigurasi *database*, koneksi DSN (*Data Source Name*) dan "SQLLinkCEServer" sudah diaktifkan. Jika koneksi dari sisi *server* siap, maka *user* perlu mengatur koneksi dari *form setting* dan dikonfigurasi fungsi *connect*. *Setting form* ditunjukkan pada Gambar 4.1.



Gambar 4.1. *Setting Form*

Setting form mengetikkan pada 2 *textbox* yang disediakan, diantaranya *Server hostbox* diisi IP *server* dan *Database databox* diisi dengan nama *database* yang telah ditetapkan sebagai *data source*. Pada Gambar 4.2. diperlihatkan *log* dari proses *connect* dari "SQLLinkCE".



Gambar 4.2. Log Proses Connect

Pada Gambar 4.2. ditunjukkan bahwa *port master* yang ditetapkan oleh "SQLLinkCE" adalah 9888, sedangkan untuk *port slave* diberikan oleh "SQLLinkCE" secara acak yaitu 1382. Cara kerjanya yaitu *server* akan mendahului membangun koneksi dengan membuka *port* 9888 dan menunggu di alamat IP = "202.43.254.88". Jika ada *client* yang terhubung dengan "SQLLinkCE", maka "SQLLinkCE" akan memberikan *port slave*. Bila koneksi *server* dan *client* sudah berhasil, maka fungsi *open database* akan dieksekusi. Pada Program 4.1. diperlihatkan *listing* program dari fungsi *connect*.

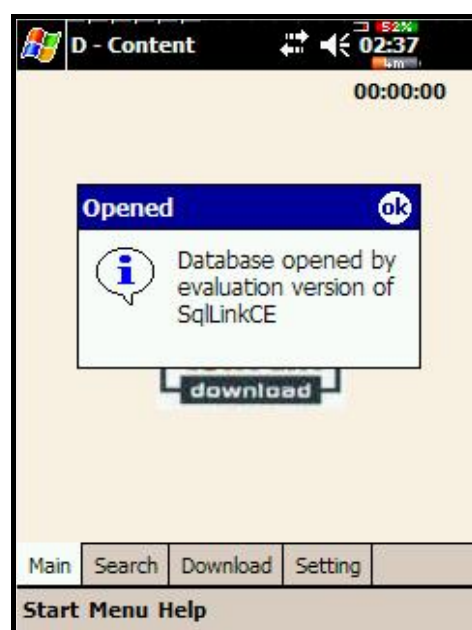
Program 4.1. Connect to Server

```

1. sqlLink.Host = hostbox.Text
2. sqlLink.Remote_Port = 9888
3. If Not sqlLink.Connect_To_Server() = 0 Then
4.     MsgBox(sqlLink.Error_Message)
5. End If
6. sqlLink.Connection_String = "Provider=MSDASQL;DSN=" & _
    databox.Text
7. If sqlLink.Open_Database <> 0 Then
8.     MsgBox("Data Source tidak ada")
9.     Exit Sub
10. End If

```

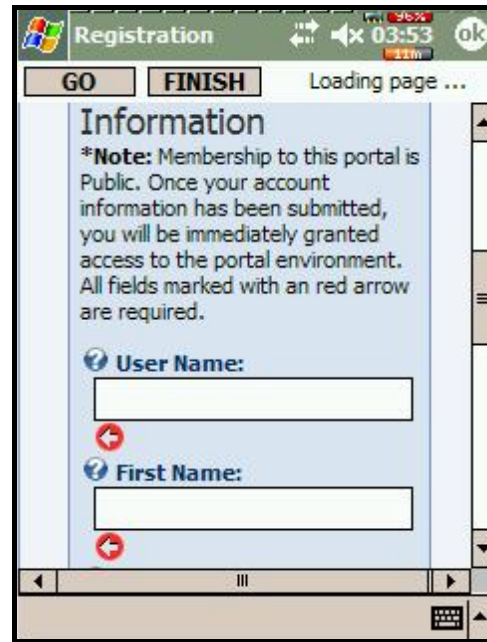
Pada Program 4.1. diperlihatkan bahwa *connection string* yang ditetapkan dengan *provider* MSDASQL, merupakan koneksi DSN yang disediakan oleh ODBC dengan menggunakan *Data Source Name*. Koneksi ini menyediakan fasilitas yang memudahkan *user*, karena deklarasi *connection string* tanpa harus menyertakan *username* ataupun *password server*. Jika proses koneksi berhasil, maka *client* dapat mengakses *database* dengan muncul pesan "Database opened by evaluation version of SQLLinkCE". Jika gagal maka muncul pesan "Data Source tidak ada". Proses *connect* berhasil ditunjukkan pada gambar 4.3.



Gambar 4.3. Proses *Connect* Berhasil

4.2. Implementasi Proses Registrasi

Registrasi ini dapat dilakukan secara *online* atau juga dapat dilakukan dari aplikasi dengan membuka menu *Registration*. Secara otomatis aplikasi akan membuka halaman yang berisi *webbrowser* yang sudah disediakan oleh aplikasi. Di dalam *webbrowser* ini sudah ditetapkan URL *registration* dari *webserver* untuk melakukan registrasi. Jika registrasi berhasil, maka *user* dapat melakukan proses *login* dengan data yang telah dimasukkan. Pada Gambar 4.4. diperlihatkan *layout* dari registrasi secara *online*. Proses membuka *registration form* secara *online* diperlihatkan pada Program 4.2.



Gambar 4.4. *Registration Online Layout*

Program 4.2. *Registration*

```

1. Try
2. 'tampilkan form registrasi
3. Dim ifrmRegistration As New Registration()
4. ifrmRegistration.ShowDialog() 'tampilkan form registration
5. ifrmRegistration.Dispose() 'destroy form registration
6. Catch ex As Exception
7. MsgBox("Cannot create Form Registration -> " + ex.ToString(),
   MsgBoxStyle.Critical) 'bila try gagal, tampilkan errornya
8. End Try
   'script di webbrowser
9. Label1.Visible = True
10. Dim main As New MainForm
11. host = main.hostbox.Text
12. Dim register As Uri = New Uri("http://" + host + _
   "/dotnetnuke/Home/tabid/37/ctl/Register/Default.aspx")
13. WebBrowser1.Navigate(register)

```

Program 4.2. memperlihatkan percobaan untuk menampilkan *form* registrasi dan diikuti dengan navigasi yang akan dilakukan oleh komponen *webbrowser* ke URL yang telah ditetapkan.

4.3. Implementasi Proses *Login*

Proses *login* dilakukan di *Login Form* diperlihatkan pada Gambar 4.5.

Gambar 4.5. Login Form

Proses *login* untuk pertama kali dibutuhkan *input* dari *user* berupa *user name*. Jika *user name* yang dimasukkan sesuai pada waktu *user* melakukan registrasi, maka aplikasi akan menyimpan *password* yang diambil dari *server*. Apabila *user* menghendaki *automatic login*, maka *user name* dan *password* akan otomatis tersimpan dalam *memory device user*. Jika *user* menginginkan *manual login*, maka *user name* dan *password* yang tersimpan sebelumnya akan terhapus dan proses pemeriksaan *user* akan berulang dengan memasukkan *username* yang baru. Proses *login* diperlihatkan dengan Program 4.3.

Program 4.3. Login

```
'konfigurasi save username dan password
1. Dim savename As String = "\Windows\a.txt"
2. Dim savepass As String = "\Windows\b.txt"
3. Dim savecek As String = "\Windows\c.txt"
4. Dim readname As String = "\Windows\a.txt"
5. Dim readpass As String = "\Windows\b.txt"
6. Dim readcek As String = "\Windows\c.txt"
7. 'select authorised (where username)
8. sqlLink.Sql = "SELECT UserPortals.Authorised FROM " + _
  "[DotNetNuke].[dbo].[UserPortals] INNER JOIN " + _
  "[DotNetNuke].[dbo].[Users] ON " + _
  "UserPortals.UserId = Users.UserID WHERE " + _
  "(Users.Username = '" & username.Text & "')"
9. sqlLink.Execute_Query() <> 0 Then
10. Exit Sub
11. End If
12. If Not sqlLink.EOF Then          'get user id
13. authorised = sqlLink.Field_Value("Authorised")
14. End If
```

```

        'delete user and password
15. If Not username.Text.Trim = "" Then
16. If (authorised = False) Then
17. If System.IO.File.Exists(readname) = True And
    System.IO.File.Exists(readpass) = True Then
18. System.IO.File.Delete(readname)
19. System.IO.File.Delete(readpass)
20. End If
21. End If
22. End If
23. If (Remember.Checked = True) Then
24. checked = True
    'file jika ada
25. If System.IO.File.Exists(readname) = True And
    System.IO.File.Exists(readpass) = True Then
26. Dim nameReader As New System.IO.StreamReader(savename)
27. Dim passReader As New System.IO.StreamReader(savepass)
28. username.Text = nameReader.ReadToEnd
29. myPass = passReader.ReadToEnd
30. nameReader.Close()
31. passReader.Close()
32. ElseIf authorised = False Then
33. If username.Text.Trim = "" Then 'bila username kosong
34. MsgBox("Please entry username for first automatic login",
    MsgBoxStyle.Information)
35. username.Focus() 'fokuskan ke username
36. End If
    'select pass
37. myUser = username.Text
38. sqlLink.Sql = "SELECT aspnet_Membership.Password FROM " + _
    "[DotNetNuke].[dbo].[aspnet_Membership] " + _
    "INNER JOIN [DotNetNuke].[dbo].[aspnet_Users] ON " + _
    "aspnet_Membership.UserId = aspnet_Users.UserId " + _
    "WHERE (aspnet_Users.UserName = '" & myUser & "')"
39. If sqlLink.Execute_Query() <> 0 Then
40. MsgBox(sqlLink.Error_Message)
41. Exit Sub
42. End If
43. If Not sqlLink.EOF Then 'get pass
44. myPass = sqlLink.Field_Value("Password").ToString
45. End If
    'file jika belum ada
46. Dim nameWriter As New System.IO.StreamWriter(savename)
47. Dim passWriter As New System.IO.StreamWriter(savepass)
48. nameWriter.Write(username.Text)
49. passWriter.Write(myPass)
50. nameWriter.Close()
51. passWriter.Close()
52. MsgBox("You've selected automatically login",
    MsgBoxStyle.Information)
53. ElseIf System.IO.File.Exists(readcek) = True Then 'check active
54. post = ""
55. Dim cekReader As New System.IO.StreamReader(savecek)
56. post = cekReader.ReadToEnd
57. cekReader.Close()
58. Else
59. MsgBox("Error, account used by another person",
    MsgBoxStyle.Information)
60. Exit Sub

```

```

61. End If
62. ElseIf (Remember.Checked = False) Then
    'checkbox false
63. checked = False
64. If System.IO.File.Exists(readname) = True And
    System.IO.File.Exists(readpass) = True And authorised = False
    Then
65. If username.Text.Trim = "" Then 'bila username kosong
66. MsgBox("Please entry username for first automatic login",
    MsgBoxStyle.Information)
67. username.Focus() 'fokuskan ke username
68. End If
69. MsgBox("You've selected manually login!!!",
    MsgBoxStyle.Information)
70. ElseIf (authorised = True) Then
71. If System.IO.File.Exists(readname) = True And
    System.IO.File.Exists(readpass) = True Then
72. Dim passReader As New System.IO.StreamReader(savepass)
73. myPass = passReader.ReadToEnd
74. passReader.Close()
75. MsgBox("You've selected manually login",
    MsgBoxStyle.Information)
76. Else
77. MsgBox("Error, account used by another person",
    MsgBoxStyle.Information)
78. Exit Sub
79. End If
80. ElseIf System.IO.File.Exists(readcek) = True Then 'check active
81. post = ""
82. Dim cekReader As New System.IO.StreamReader(savecek)
83. post = cekReader.ReadToEnd
84. cekReader.Close()
85. Else
86. MsgBox("Error, account used by another person",
    MsgBoxStyle.Information)
87. Exit Sub
88. End If
89. End If
90. If username.Text.Trim = "" Or myPass.Trim = "" Then
91. MsgBox("User name belum diisi atau account sudah dipakai",
    MsgBoxStyle.Information)
92. username.Focus() 'fokuskan ke username
93. Exit Sub
94. Else
95. myUser = username.Text
96. Mainform_load()
97. If MenuItem5.Text = "Login" Then
98. sqlLink.Sql = "SELECT Users.UserID, " + _
    "aspnet_Membership.Password, aspnet_Users.UserId " + _
    "AS nomor FROM [DotNetNuke].[dbo].[aspnet_Membership] " + _
    "INNER JOIN [DotNetNuke].[dbo].[aspnet_Users] " + _
    "ON aspnet_Membership.UserId = aspnet_Users.UserId " + _
    "CROSS JOIN [DotNetNuke].[dbo].[Users] " + _
    "WHERE (aspnet_Users.UserName = '" & myUser & "') AND " + _
    "(Users.Username = '" & myUser & "')"
99. If sqlLink.Execute_Query() <> 0 Then
100. Exit Sub
101. End If
102. If Not sqlLink.EOF Then 'get user id&pass

```

```

103. myID = sqlLink.Field_Value("UserID").ToString
104. myID1 = sqlLink.Field_Value("nomor").ToString
105. If (checked = False And authorised = False) Then
106. myPass = sqlLink.Field_Value("Password").ToString
107. End If
108. End If
    'disable get password
109. If (checked = True) Then
110. If Not (myPass.Trim = "") Then
111. sqlLink.Sql = "UPDATE [DotNetNuke].[dbo].[UserPortals] " + _
    "SET [Authorised] = 1 WHERE [UserId] = '" & myID & "'"
112. If sqlLink.Execute_Sql() <> 0 Then
113. Exit Sub
114. End If
115. End If
116. End If
    'record state
117. If Not (myPass.Trim = "") Then
118. Dim cekWriter As New System.IO.StreamWriter(savecek)
119. cekWriter.Write("active")
120. cekWriter.Close()
121. post = "active"
122. End If
    'select approve (where userid and pass)
123. sqlLink.Sql = "SELECT [IsApproved] FROM " + _
124. "[DotNetNuke].[dbo].[aspnet_Membership] WHERE [UserId] " + _
    "= '" & myID1 & "' AND [Password] = '" & myPass & "'"
125. If sqlLink.Execute_Query() <> 0 Then
126. Exit Sub
127. End If
128. If Not sqlLink.EOF Then          'get user id
129. approved = sqlLink.Field_Value("IsApproved")
130. End If
131. If (post = "active") Then
132. access = False
133. End If
134. If (approved = True) And (access = False) Then
    'select first name (where username)
135. sqlLink.Sql = "SELECT DisplayName FROM " + _
    "[DotNetNuke].[dbo].[Users] WHERE " + _
    "Users.Username = '" & myUser & "'"
136. If sqlLink.Execute_Query() <> 0 Then
137. Exit Sub
138. End If
139. If Not sqlLink.EOF Then 'welcome dan user_name
140. myName = sqlLink.Field_Value("DisplayName").ToString()
141. welcome.Text = "Welcome, " & myName
142. welcome1.Text = "Welcome, " & myName
143. welcome2.Text = "Welcome, " & myName
144. End If
    'select balance (where userid)
145. sqlLink.Sql = "SELECT TOP 1 Balance FROM " + _
    "[DotNetNuke].[dbo].[User_Balance] " + _
    "WHERE [UserId] = '" & myID & "' ORDER BY [Update_Id] DESC"
146. If sqlLink.Execute_Query() <> 0 Then
147. Exit Sub
148. End If
149. If Not sqlLink.EOF Then
150. mysaldo = sqlLink.Field_Value("Balance").ToString()

```

```

151. End If
152. MenuItem5.Text = "Logout"      'mengganti login --> logout
153. saldobl2.Visible = True
154. saldobl1.Text = "Saldo : Rp. " + mysaldo
155. saldobl1.Text = "Saldo : Rp. " + mysaldo
156. saldobl2.Text = "Saldo : Rp. " + mysaldo
157. Panel1.Enabled = True        'untuk mengakses Menu
158. Panel2.Enabled = True
159. MenuItem2.Enabled = False 'register disable
160. MenuItem16.Enabled = True 'login dulu agar dapat download
161. MenuItem17.Enabled = True 'login dulu agar dapat preview
162. MenuItem18.Enabled = True 'login dulu agar dapat search
163. MenuItem20.Enabled = True 'login dulu agar dapat add voucher
164. Timer1.Enabled = True
165. Else
166. MsgBox("You're not authorized!", MsgBoxStyle.Information)
167. End If
168. Else
169. MsgBox("Please retry to login")
170. MenuItem5_Click(sender, e)
171. Exit Sub
172. End If
173. End If

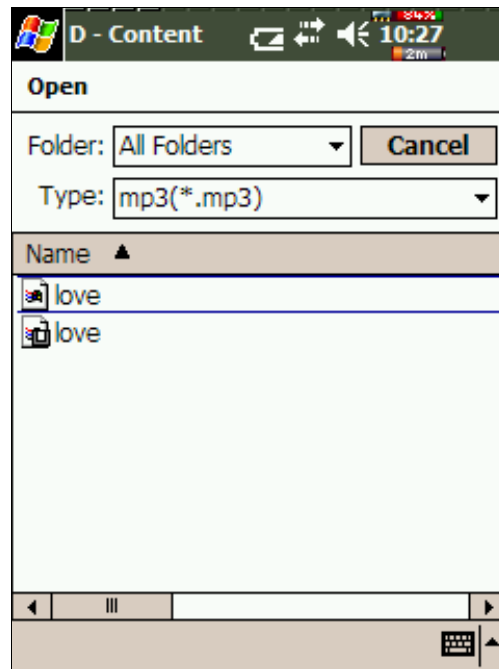
```

Dari Program 4.3. diperlihatkan pertama kali *user* memasukkan *user name* dan jika sesuai yang tersimpan di *database*, maka *password* akan tersimpan di variabel "myPass" (baris 45, 109). *User* dapat memilih untuk melakukan *login* secara otomatis dengan memeriksa *checkbox* dengan nama "remember" (baris 24 - 62). *User name* dan *password* akan secara otomatis tersimpan di *device user* dengan parameter "/windows/*.txt" (baris 1-4). Jika *user* memilih *manual login*, maka *user name* yang sudah pernah tersimpan akan terhapus, sehingga tiap kali *login* *user* harus memasukkan *user name* (baris 53-90).

Kemudian variabel "post" akan memeriksa jika *user* pernah mengalami *disconnect* karena *lost connection*. Jika pernah, maka "access" bernilai "False". Dan jika sebaliknya, maka program akan menyimpan variabel "post" di *device user* (baris 64-73, 83-85). Lalu proses akan berlanjut dengan memeriksa status akhir *user* yaitu "Authorised" dan "IsApproved". Jika keduanya bernilai "True", maka status "active" akan tersimpan di *device user* (baris 131 - 198). Proses ini akan mengantisipasi keadaan *disconnect* karena *lost connection* dan *account user* tidak dapat digunakan secara bersamaan. Bila *user* berhasil melakukan *login* maka akan tampil *display name* (baris 140 - 151) dan saldo akhir (baris 151 - 158), kemudian mengganti tulisan "Logout" menjadi "Login" pada pilihan sub menu (baris 159), mengaktifkan *timer* (baris 171) dan *member access* (baris 164 - 170).

4.4. Implementasi Proses *Open File*

Pada Gambar 4.6. diperlihatkan dialog untuk membuka *file*.



Gambar 4.6. *Open File Dialog*

Pada Gambar 4.6. diperlihatkan dialog yang menyediakan fasilitas untuk membuka *file* dari *device user* baik itu berupa file yang mempunyai *format audio* maupun *video*. Fasilitas ini membantu *user* untuk melihat dan memutar kembali *file* yang pernah di-*download* dan *file* yang sudah tersimpan di *memory device*. Pada Program 4.4. ditunjukkan *listing* program untuk membuka *file*.

Program 4.4. *Open file*

```

1. Dim abc As New OpenFileDialog
2. With abc
3. .FilterIndex = 1
4. .Filter = "All Files (*.*)"
5. End With
6. If abc.ShowDialog = Windows.Forms.DialogResult.OK Then
7. openfile = abc.FileName
8. Dim playfile As New System.Diagnostics.Process
9. With playfile 'open file
10. .StartInfo.FileName = openfile
11. .StartInfo.UseShellExecute = True
12. .Start()
13. End With
14. End If

```

Program 4.4. memperlihatkan proses membuka *file* menggunakan *shell execute* (baris 11), fungsi ini merupakan *library* dari *System.Diagnostic.Process* (baris 8). *File* akan secara otomatis terbuka dengan *Media Player* yang sudah ter-*install* di *device user*. Pada Gambar 4.7. diperlihatkan aplikasi pemutar *file* dengan TCPMP.



Gambar 4.7. Aplikasi Pemutar *File*

4.5. Implementasi Proses *Search*

Pada Gambar 4.8. diperlihatkan bahwa *form search* memiliki fitur pencari konten.



Gambar 4.8. *Search Finished*

User dapat memasukkan *file name*, *file type* dan atau *genre* dari konten yang dicari. *File name* juga menggunakan fasilitas "LIKE", user dapat mencari dengan huruf atau kata yang terkandung dalam nama konten tersebut. Jika konten berhasil dicari, maka di *listview* akan menampilkan hasil pencarian tersebut dengan tambahan deskripsi berupa *file ID* dan *price file*. Untuk lebih jelasnya, lihat Program 4.5.

Program 4.5. Search

```

1. If Not filenamebox.Text.Trim = "" AndAlso filetypebox.Text.Trim
   = "" AndAlso filenamebox.Text.Trim = "" Then
2. MsgBox("Insert your search item", MsgBoxStyle.Information)
   'input name file
3. ElseIf filetypebox.Text.Trim = "" AndAlso genrebox.Text.Trim =
   "" Then
4. searchname = filenamebox.Text
5. sqlLink.Sql = "SELECT File_TA.File_ID, File_TA.File_Name,
   File_TA_Category.Category_Name, File_TA.Genre,
   File_TA.File_Price " + _
   "FROM DotNetNuke.dbo.File_TA INNER JOIN
   DotNetNuke.dbo.File_TA_Category ON File_TA.Category_ID =
   File_TA_Category.Category_ID " + _
   "WHERE (File_TA.File_Name LIKE '%" & searchname & "%') AND
   (File_TA.Status = 'active')"
   'input genre file
6. ElseIf filenamebox.Text.Trim = "" AndAlso
   filetypebox.Text.Trim = "" Then
7. searchgenre = genrebox.Text
8. sqlLink.Sql = "SELECT File_TA.File_ID, File_TA.File_Name,
   File_TA_Category.Category_Name, File_TA.Genre,
   File_TA.File_Price " + _
   "FROM DotNetNuke.dbo.File_TA INNER JOIN
   DotNetNuke.dbo.File_TA_Category ON File_TA.Category_ID =
   File_TA_Category.Category_ID " + _
   "WHERE (File_TA.Genre = '" & searchgenre & "') AND
   (File_TA.Status = 'active')"
   'input type(category) file
9. ElseIf filenamebox.Text.Trim = "" AndAlso genrebox.Text.Trim =
   "" Then
10. searchtype = filetypebox.Text
11. sqlLink.Sql = "SELECT File_TA.File_ID, File_TA.File_Name,
   File_TA_Category.Category_Name, File_TA.Genre,
   File_TA.File_Price " + _
   "FROM DotNetNuke.dbo.File_TA INNER JOIN
   DotNetNuke.dbo.File_TA_Category ON File_TA.Category_ID =
   File_TA_Category.Category_ID " + _
   " WHERE (File_TA_Category.Category_Name = '" & searchtype & "')
   AND (File_TA.Status = 'active')"
   'input type & genre file
12. ElseIf filenamebox.Text.Trim = "" Then
13. searchtype = filetypebox.Text
14. searchgenre = genrebox.Text

```

```

15. sqlLink.Sql = "SELECT File_TA.File_ID, File_TA.File_Name,
File_TA_Category.Category_Name, File_TA.Genre,
File_TA.File_Price " + _
"FROM DotNetNuke.dbo.File_TA INNER JOIN
DotNetNuke.dbo.File_TA_Category ON File_TA.Category_ID =
File_TA_Category.Category_ID " + _
" WHERE (File_TA_Category.Category_Name LIKE '" & searchtype &
"' ) AND (File_TA.Genre = '" & searchgenre & "' ) AND
(File_TA.Status = 'active')"


```

```

36. If Not sqlLink.EOF Then
37. lv.BeginUpdate()      'ubah status listview untuk dapat diedit
38. numb = 1             'inisialisasi counter data
39. While Not sqlLink.EOF      'selama masih ada data, baca terus
40. If Not bLVInit Then
41. lv.Items.Clear()      'kosongkan listview
42. For Ict = lv.Columns.Count To 1 Step -1
43. lv.Columns.Remove(lv.Columns(Ict - 1))
44. Next 'buat judul kolom
45. lv.Columns.Add("No", 25, HorizontalAlignment.Left) 'data(numb)
46. lv.Columns.Add("ID", 40, HorizontalAlignment.Left)
47. lv.Columns.Add("Name", 100, HorizontalAlignment.Left)
48. lv.Columns.Add("Type", 45, HorizontalAlignment.Left)
49. lv.Columns.Add("Genre", 70, HorizontalAlignment.Left)
50. lv.Columns.Add("Price", 55, HorizontalAlignment.Left)
51. bLVInit = True
52. End If
    'buat kolom & isi kolom
53. lvi = New ListViewItem 'create item baris
54. lvi.Text = (numb.ToString) 'isi kolom pertama
55. For Ict = 1 To sqlLink.Field_Value.Count 'kolom selanjutnya
56. lvi.SubItems.Add(sqlLink.Field_Value(Ict).ToString)
57. Next
58. lv.Items.Add(lvi) 'tambahkan itembaris ke listview
59. lvi = Nothing 'destroy item baris
60. numb = numb + 1 'tambah counter data
61. sqlLink.MoveNext() 'pindah ke baris selanjutnya
62. End While
63. lv.EndUpdate() 'masukkan semua perubahan yang diupdate
64. MsgBox("Search finished", MsgBoxStyle.OkOnly)
    'tampung isi field search
65. Else
66. MsgBox("File not found !!!", MsgBoxStyle.Information)
67. End If

```

Pada Program 4.5. diperlihatkan proses *search content* yang meminta kunci pencarian berupa *file name*, *type*, atau dan *category* (baris 4-29). Untuk proses *file name* dapat dicari dengan menggunakan fungsi "LIKE", yaitu salah satu perintah SQL untuk mencari kemiripan huruf (baris 6, 9,12, 16, 20, 29). Jika *user* telah memasukkan kunci pencarian dan status konten aktif (baris 6, 9,12, 16, 20, 29), maka program menampilkan deskripsi *file* ditambah dengan *file ID* dan harga di *listview* (baris 38-65).

4.6. Implementasi Proses *Select File*

Proses *select file* merupakan proses pemilihan konten yang telah berhasil dicari dari proses *search content*. Bila *user* ingin melakukan proses *preview* atau *buy*, *user* harus memilih dari konten yang ditampilkan di *listview*. *User* dapat memilih salah satu konten yang tersedia dengan memilih nomor pada kolom pertama *listview*. Pada Gambar 4.9. diperlihatkan jika proses *select file* berhasil.



Gambar 4.9. Select File

Proses ini mengambil *file* ID sebagai parameter untuk mengambil variabel atau data yang lainnya seperti *path buy* dan *path preview*. Pada Program 4.6. ditunjukkan *listing* program *select file*.

Program 4.6. Select File

```

'select file to download
1. If Me.lv.SelectedIndices.Count <= 0 Then
2. Return
3. End If
4. Dim selNdx = Me.lv.SelectedIndices(0)
5. If selNdx >= 0 Then
6. getfileid = lv.Items(selNdx).SubItems(1).Text
7. End If
8. sqlLink.Sql = "SELECT File_TA.Genre, File_TA.File_Name,
File_TA.Category.Category_Name, File_TA.File_Price,
File_TA.Preview_Location, File_TA.File_Location " + _
"FROM DotNetNuke.dbo.File_TA INNER JOIN
DotNetNuke.dbo.File_TA_Category ON File_TA.Category_ID =
File_TA_Category.Category_ID " + _
"WHERE (File_TA.File_ID = '" & getfileid & "')"
9. If sqlLink.Execute_Query() <> 0 Then
10. MsgBox(sqlLink.Error_Message)
11. Exit Sub
12. End If
13. If Not sqlLink.EOF Then 'query path preview
14. txtFileName.Text = sqlLink.Field_Value("File_Name").ToString
15. txtFileType.Text = +_
sqlLink.Field_Value("Category_Name").ToString
16. pilih = server + _
sqlLink.Field_Value("Preview_Location").ToString

```

```

17. pathfile = server + _
    sqlLink.Field_Value("File_Location").ToString
18. harga = sqlLink.Field_Value("File_Price").ToString
19. hargafile.Text = "Price : Rp. " + harga
20. TabControll1.SelectedIndex = 2 'pindah ke halaman download
21. MsgBox("You've selected", MsgBoxStyle.OkOnly)
22. End If

```

Program 4.6. memperlihatkan proses pengambilan variabel untuk proses pembelian dan *preview* konten. Proses *select file* berawal dari pemilihan nomor konten yang ditampilkan dari *listview* (baris 4-7). Jika berhasil maka variabel "pilih" akan diisi dengan *preview location* dan variabel "pathfile" (baris 16) akan diisi dengan *file location* untuk pembelian (baris 17). Proses ini akan berakhir dengan berpindahnya *tab control* ke halaman *download* (baris 20).

4.7. Implementasi Proses Buy

Implementasi proses *buy* membutuhkan proses *search content* dan *select file*. Program 4.7. akan memperlihatkan cara kerja dari sistem untuk melakukan pembelian.

Program 4.7. Buy

```

1. If txtFileName.Text <> "" And txtFileType.Text <> "" Then
2. filename = txtFileName.Text
3. filetype = txtFileType.Text
4. Try
    'ambil harga dan path download
5. sqlLink.Sql = "INSERT INTO " + _
    "[DotNetNuke].[dbo].[User_Download] " + _
    "[UserId],[File_ID]) " + _
    "VALUES('" & myID & "','" & getfileid & "')"
6. If sqlLink.Execute_Sql() <> 0 Then
7. Exit Sub
8. End If
    'get last saldo
9. sqlLink.Sql = "SELECT TOP 1 Balance FROM " + _
    "[DotNetNuke].[dbo].[User_Balance] WHERE " + _
    "[UserId] = '" & myID & "' ORDER BY [Update_Id] DESC"
10. If sqlLink.Execute_Query() <> 0 Then
11. MsgBox(sqlLink.Error_Message)
12. Exit Sub
13. Else
14. mysaldo = ""
15. End If
16. If Not sqlLink.EOF Then
17. mysaldo = sqlLink.Field_Value("Balance").ToString
18. saldobl.Text = "Saldo : Rp. " + mysaldo
19. saldobl1.Text = "Saldo : Rp. " + mysaldo
20. End If
21. Catch ex As Exception
22. MsgBox("Uncompleted buy process!",MsgBoxStyle.Information)

```

```

23. End Try
24. Me.SaveFileDialog1.FileName = filename + "." + filetype
25. If Me.SaveFileDialog1.ShowDialog = + _
    Windows.Forms.DialogResult.OK Then
26. Me.whereToSave = Me.SaveFileDialog1.FileName
27. Me.SaveFileDialog1.FileName = ""
28. Labell1.Text = "Save to: " & whereToSave
29. Else
30. MsgBox("You've cancelled download", MsgBoxStyle.Information)
31. Exit Sub
32. End If
    'Creating the request and getting the response
33. Dim theResponse As System.Net.WebResponse
34. Dim theRequest As System.Net.WebRequest
35. Try 'Checks if the file exist
36. theRequest = System.Net.HttpWebRequest.Create(pathfile)
37. theResponse = theRequest.GetResponse
38. Catch ex As Exception
39. MsgBox("An error occurred while downloading file. Possibe
    causes:" & ControlChars.CrLf & _
    "1) File doesn't exist" & ControlChars.CrLf & _
    "2) Remote server error", MsgBoxStyle.Critical)
40. Exit Sub
41. End Try
    'Size of the response (in bytes)
42. Dim length As Long = theResponse.ContentLength
43. Dim writeStream As New IO.FileStream(Me.whereToSave,
    IO.FileMode.Create)
44. Dim nRead As Integer
45. Dim readings As Integer = 0
46. Do
47. Dim readBytes(4095) As Byte
48. Dim bytesread As Integer =
    theResponse.GetResponseStream.Read(readBytes, 0, 4096)
49. nRead += bytesread
50. Dim percent As Short = CShort((nRead * 100) / (length))
51. If bytesread = 0 Then
52. Exit Do
53. writeStream.Write(readBytes, 0, bytesread)
54. readings += 1
55. If readings >= 5 Then
56. readings = 0
57. End If
58. Loop
    'Close the streams
59. theResponse.GetResponseStream.Close()
60. writeStream.Close()
61. If stat = True Then
62. IO.File.Delete(Me.whereToSave)
63. End If
64. Else
65. MsgBox("Please insert valid filename and filetype for
    download", MsgBoxStyle.Critical)
66. End If

```

Dari Program 4.7. memperlihatkan pertama program akan memasukkan *file ID* dan *user name* ke *database* (baris 5-9). *Server* akan memeriksa jika nilai saldo lebih besar dari harga konten yang akan di *download*, maka saldo akhir akan muncul di layar *device* (baris 10-21). Kemudian akan tampil *save file dialog* (lihat Gambar 4.10.), yang meminta *user* untuk menentukan direktori dan nama *file* yang disimpan (baris 25-33). Proses *download* akan dimulai dengan *web request* (baris 34) kemudian menunggu *web response* (baris 35) dengan menggunakan *library System.Net*. Jika koneksi memungkinkan, maka pengiriman data akan dilakukan secara *stream* (baris 38-60). Proses terakhir dilakukan penutupan *stream* dan penghapusan, jika ada *file* yang sama (baris 60-64).



Gambar 4.10. *Save File Dialog*

4.1.8. Implementasi Proses *Preview*

Preview merupakan salah satu fitur *member access* yang memudahkan *user* untuk memutar konten tanpa memotong saldo. Proses ini pada prinsipnya hanya men-*download* konten dan memutar *file* konten di "Media Player". Proses *preview* ditunjukkan pada Program 4.8.

Program 4.8. Preview

```

1. Try
2. filename = txtFileName.Text
3. filetype = txtFileType.Text
4. Me.SaveFileDialog1.FileName = "\My Documents\" + filename
   'path preview has adjusted
5. Me.whereToSave = Me.SaveFileDialog1.FileName
6. Me.SaveFileDialog1.FileName = ""
   'Creating the request and getting the response
7. Dim theResponse As System.Net.WebResponse
8. Dim theRequest As System.Net.WebRequest
9. Try 'Checks if the file exist
10. theRequest = System.Net.HttpWebRequest.Create(pilih)
11. theResponse = theRequest.GetResponse
12. Catch ex As Exception
13. MsgBox("An error occurred while downloading file. Possibe
   causes:" & ControlChars.CrLf & _
   "1) File doesn't exist" & ControlChars.CrLf & _
   "2) Remote server error", MsgBoxStyle.Critical)
14. Exit Sub
15. End Try
16. Dim length As Long = theResponse.ContentLength
17. Dim writeStream As New IO.FileStream(Me.whereToSave,
   IO.FileMode.Create)
18. Dim nRead As Integer
19. Dim readings As Integer = 0
20. Do
21. Dim readBytes(4095) As Byte
22. Dim bytesread As Integer =
   theResponse.GetResponseStream.Read(readBytes, 0, 4096)
23. nRead += bytesread
24. Dim percent As Short = CShort((nRead * 100) / (length))
25. If bytesread = 0 Then
26. Exit Do
27. writeStream.Write(readBytes, 0, bytesread)
28. readings += 1
29. If readings >= 5 Then
30. readings = 0
31. If (percent = 50) Then Exit Do
32. End If
33. Loop
   'Close the streams
34. theResponse.GetResponseStream.Close()
35. writeStream.Close()
36. If Me.stat = True Then
37. IO.File.Delete(Me.whereToSave)
38. Exit Sub
39. End If
40. Catch ex As Exception
41. MsgBox("Error, cannot preview...", MsgBoxStyle.Critical)
42. Exit Sub
43. End Try
44. Dim playfile As New System.Diagnostics.Process
45. With playfile 'openfile
46. .StartInfo.FileName = "\My Documents\" + filename
47. .StartInfo.UseShellExecute = True
48. .Start()
49. End With

```

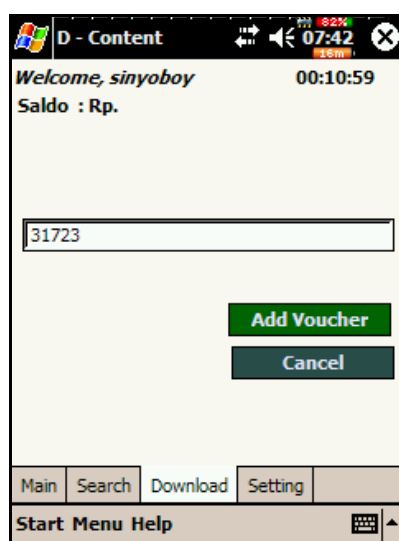
Program 4.7. memperlihatkan bahwa proses *preview* hampir sama dengan proses pembelian hanya saja untuk direktori dan *file name* sudah ditentukan (baris 4-6). Setelah itu program ini akan men-*download file* yang telah dipilih. Proses *download* akan dihentikan ketika variabel "percent" bernilai 50 (baris 31). *File* yang berhasil di-*download* akan langsung diputar dengan menggunakan aplikasi "Media Player" dalam hal ini menggunakan aplikasi "TCPMP" (baris 44 – 49), diperlihatkan pada Gambar 4.11.



Gambar 4.11. Aplikasi TCPMP

4.9. Implementasi Proses Add Voucher

Proses memasukkan ID *voucher* diperlihatkan pada Gambar 4.12.



Gambar 4.12. Proses Memasukkan ID *Voucher*

Proses implementasi ini membutuhkan pengisian ID *voucher* yang aktif. Proses ini akan gagal jika masa aktif berakhir, sudah terjual, atau ID *voucher* tidak sama dengan yang disediakan oleh *server*. Program 4.9. akan memperlihatkan cara kerja dari pemeriksaan ID *voucher* dan penambahan nilai saldo.

Program 4.9. Add voucher

```

1. voucherid = addvoucher.Text
2. sqlLink.Sql = "INSERT INTO " + _
   "[DotNetNuke].[dbo].[User_AddVoucher] " + _
   "([UserId], [Voucher_ID])
   "VALUES('" & myID & "',''" & voucherid & "')"
3. If sqlLink.Execute_Sql() <> 0 Then 'insert voucher id
4. Exit Sub
5. Else
6. MsgBox("You've got new balance")
7. End If
8. sqlLink.Sql = "SELECT TOP 1 Balance FROM " + _
   "[DotNetNuke].[dbo].[User_Balance] " + _
   "WHERE [UserId] = '" & myID & "' ORDER BY [Update_Id] DESC"
9. If sqlLink.Execute_Query() <> 0 Then
10. Exit Sub
11. End If
12. If Not sqlLink.EOF Then 'get last saldo
13. mysaldo = sqlLink.Field_Value("Balance").ToString
14. saldobl1.Text = "Saldo : Rp. " + mysaldo
15. saldobl11.Text = "Saldo : Rp. " + mysaldo
16. End If

```

Program 4.9. memperlihatkan penambahan saldo berawal dari pengisian ID *voucher* ke *database* (baris 2). Jika *voucher* yang dimasukkan aktif, maka server akan secara langsung menambah nilai saldo *user*. Langkah terakhir pengambilan nilai saldo akan ditampilkan di layar *device* (baris 9-18).

4.10. Implementasi Proses Logout

Program 4.10. memperlihatkan proses *user logout*.

Program 4.10. Logout

```

1. If MenuItem5.Text = "Logout" Then
2. If sqlLink.Close_Database = 0 Then
3. If sqlLink.Disconnect = 0 Then 'disconnect server
4. MsgBox("Disconnected from server", MsgBoxStyle.Information)
5. If System.IO.File.Exists(readcek) = True Then
6. System.IO.File.Delete(readcek)
7. End If
8. MenuItem5.Text = "Login" 'ubah kembali text menjadi login
9. Awal_Load()
10. File_Load()
11. Mainform_load()

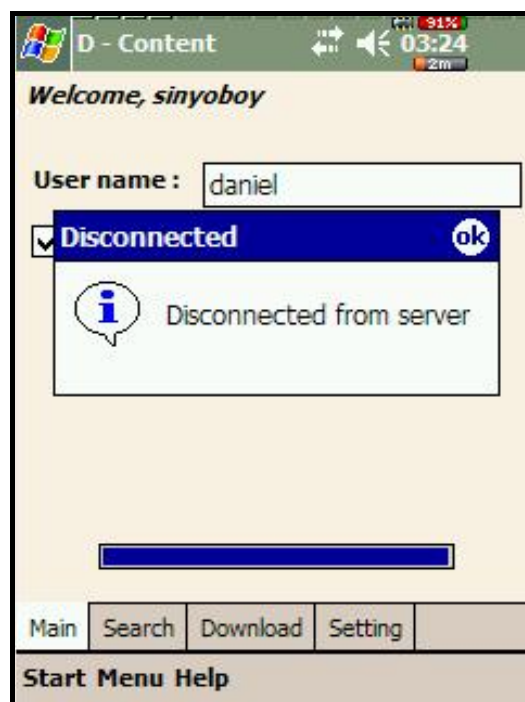
```

```

12. Else
13. MsgBox(sqlLink.Error_Message, MsgBoxStyle.Information)
14. End If
15. End If
16. End If

```

Jika Program 4.10. memperlihatkan bahwa aplikasi akan menghapus variabel "post", yang digunakan untuk mengantisipasi *lost connection*. Setelah itu aplikasi akan menutup *database* (baris 2) dan koneksi yang aktif (baris 3). Jika berhasil, maka aplikasi akan mengganti tulisan "Logout" menjadi "Login" (baris 8) dan memanggil variabel awal (baris 9 - 11). Seperti dieksekusi, maka koneksi *user* ke *database* dan ke *server* akan terputus dan tampilan *device* akan berubah, tampilan itu diperlihatkan pada Gambar 4.13.



Gambar 4.13. Sukses Logout

4.11. Implementasi Proses Exit

Proses ini akan menutup semua koneksi dan aplikasi *download content* yang sedang aktif. Proses itu akan dijalankan dengan Program 4.11.

Program 4.11. Exit

```

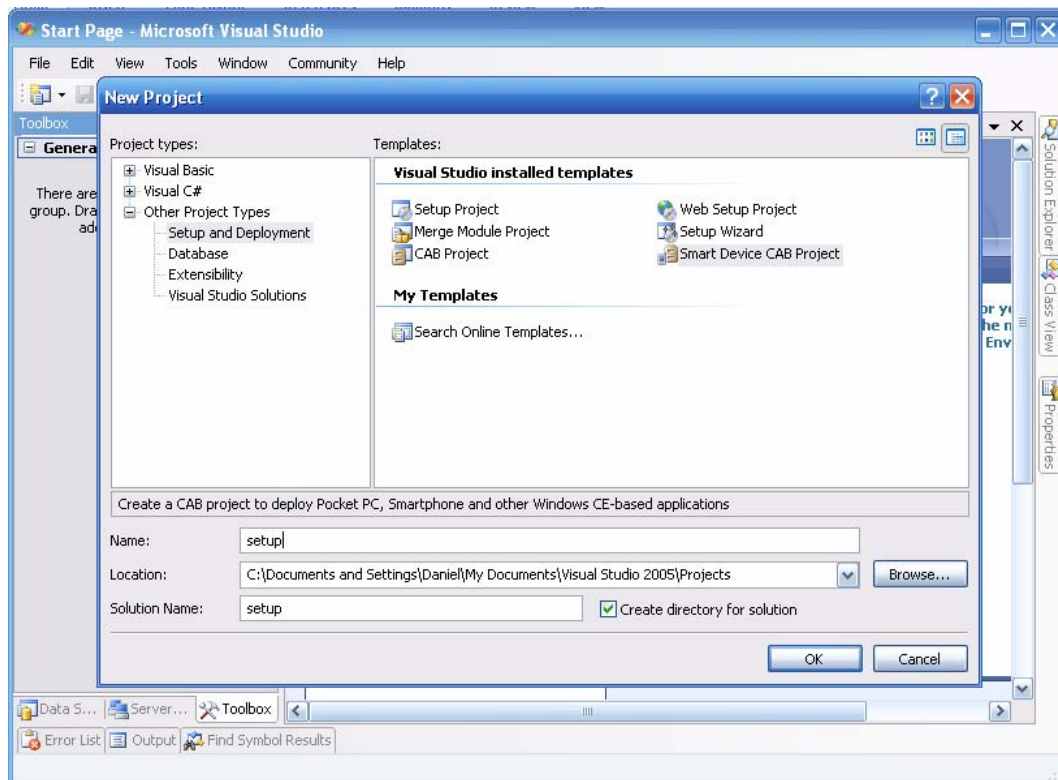
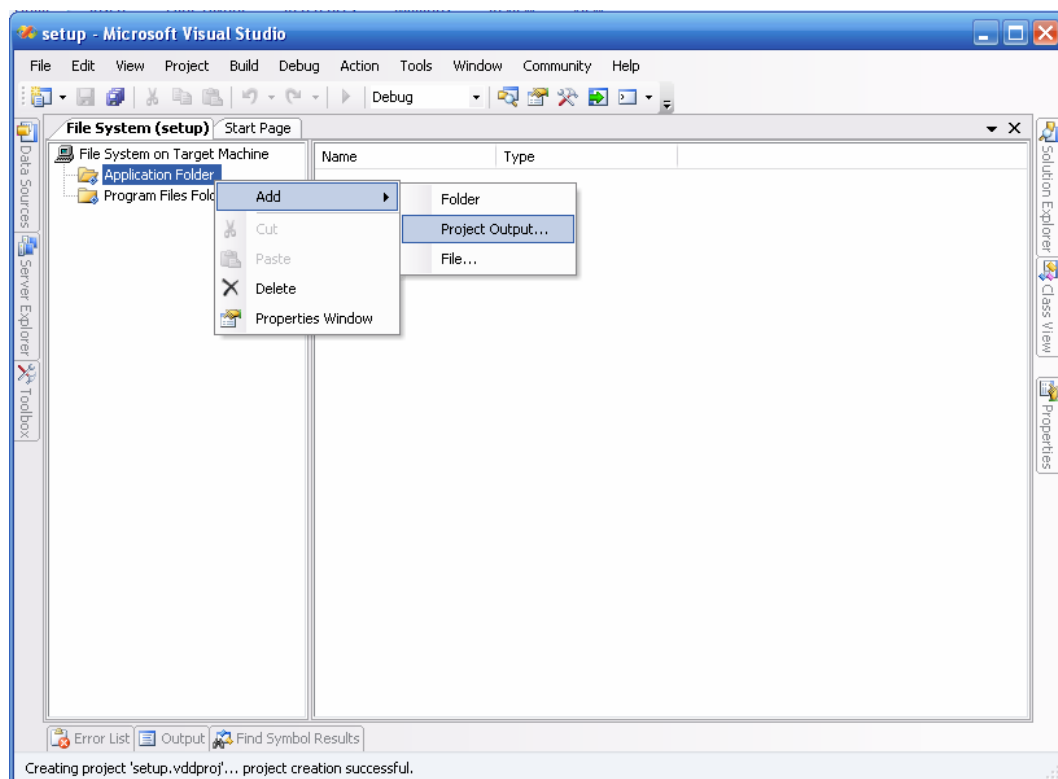
1. Condition()
2. Application.Exit()

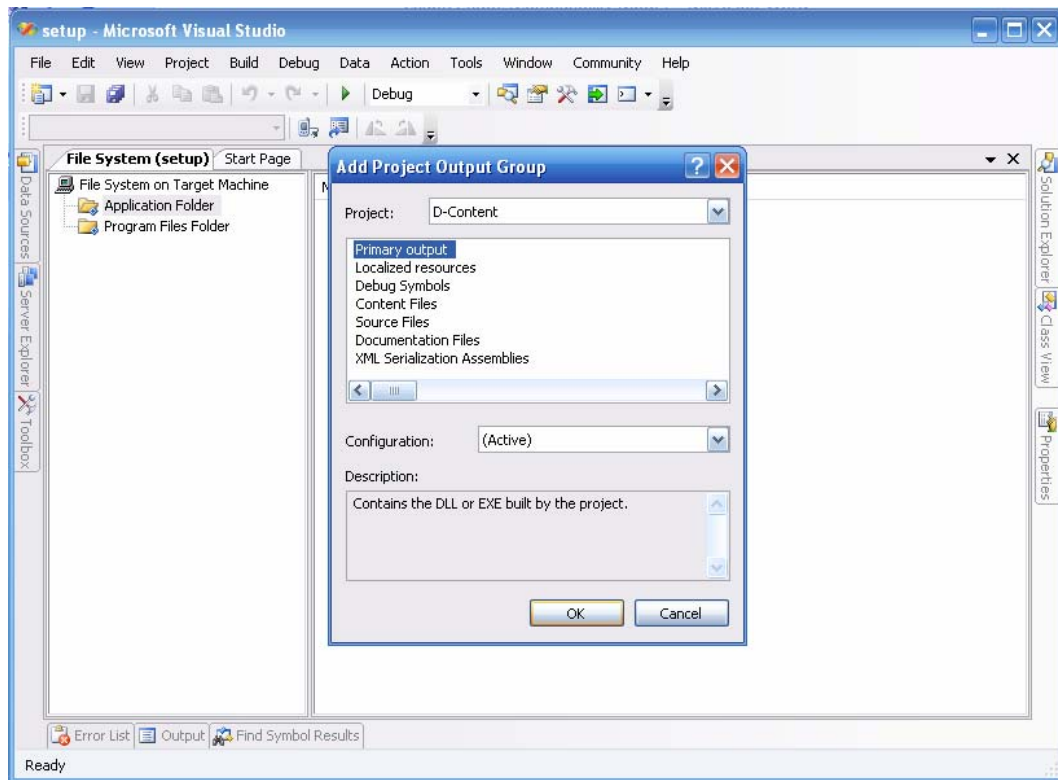
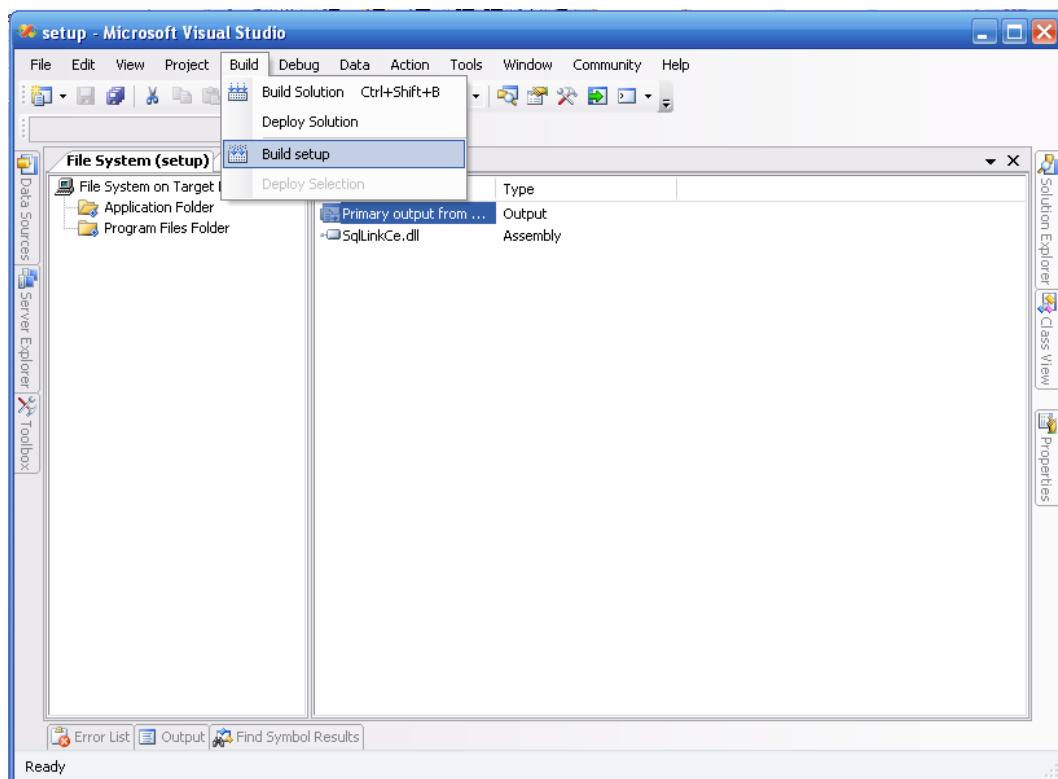
```

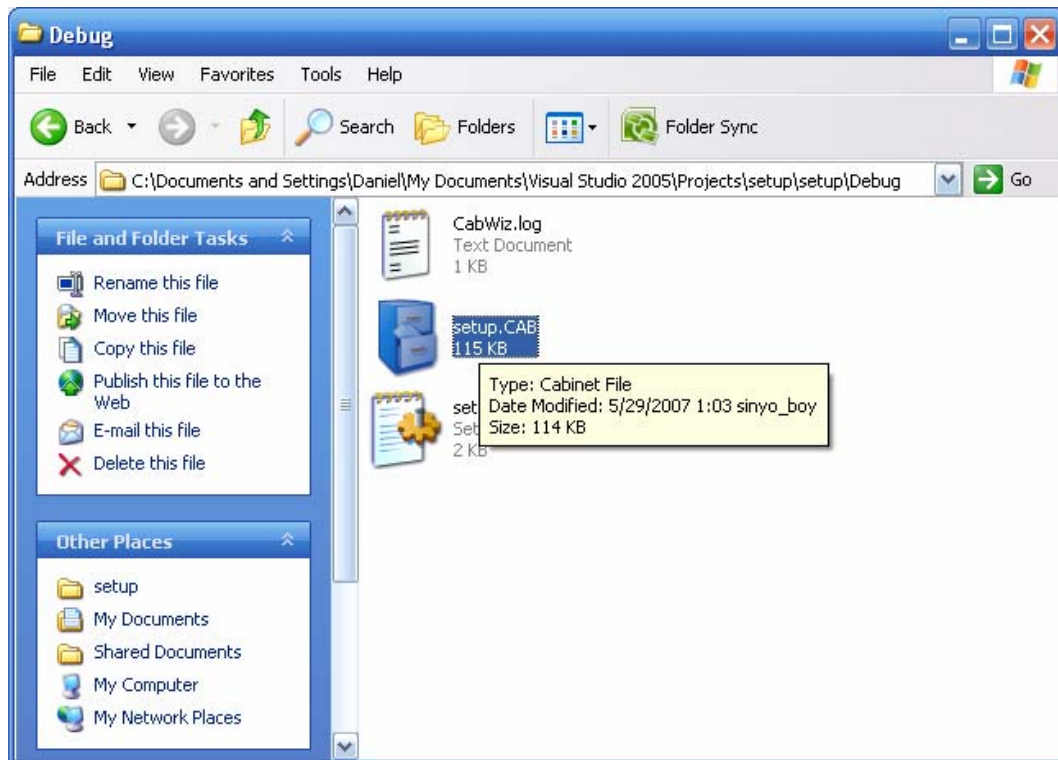
4.12. Pembuatan *Installer .CAB*

Proses instalasi yang dilakukan akan mengubah *project output* yaitu berupa ".EXE" menjadi sebuah *installer* yang mempunyai *format* ".CAB". Dalam *installer* ini terdapat *file project* hasil *build*, *assembly* dan komponen yang ditambahkan seperti "SQLLinkCE.dll". Format ".CAB" (*cabinet*) ini akan mendukung dalam proses instalasi di *mobile device*. "Microsoft Visual Studio 2005 .NET" menyediakan fasilitas untuk membentuk sebuah *installer*, langkah-langkahnya adalah sebagai berikut :

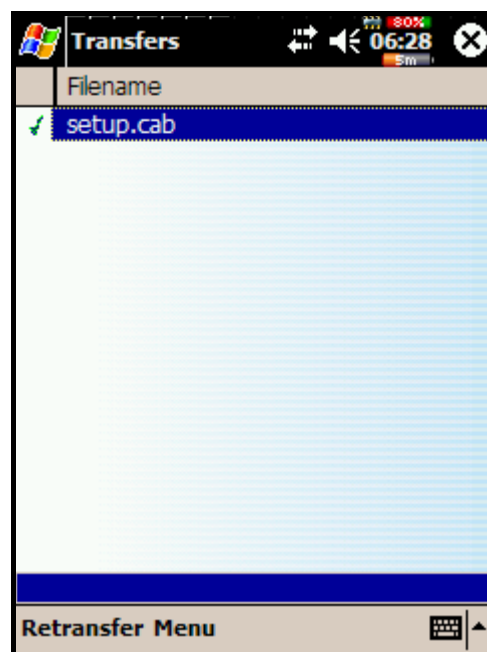
1. Klik New Project.
2. Pilih Setup and Deployment.
3. Pilih Template Smart Device CAB Project.
4. Masukkan nama *installer* dan lokasi penyimpanan, lebih jelasnya lihat Gambar 4.14.
5. Klik OK.
6. Klik kanan pada Application Folder. Agar lebih jelas, lihat Gambar 4.15.
7. Pilih Project Output, dengan catatan *project* utama sudah dibuka (dalam hal ini D-Content).
8. Pilih Primary Ouput dari *project* yang sedang aktif, lebih jelasnya lihat Gambar 4.16.
9. Klik OK.
10. Build setup, agar lebih jelas lihat Gambar 4.17.
11. Hasil dari *build project* dapat dilihat pada Gambar 4.18.
12. File tersebut dapat langsung di-*transfer* ke dalam *mobile device*. Dalam hal ini *transfer* "setup.CAB" dilakukan dengan men-*download* lewat *web*, dapat dilihat pada Gambar 4.19.

Gambar 4.14. Pilih *Template*Gambar 4.15. Pilih *Application Folder*

Gambar 4.16. Pilih *Primary Output*Gambar 4.17. *Build Project*



Gambar 4.18. Hasil *Build Project*



Gambar 4.19. *Transfer* "setup.CAB"