

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

Blog : <http://junindar.blogspot.com>

<https://mvp.support.microsoft.com/default.aspx/profile/junindar>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

E-Mail : junindar@gmail.com

I. Pengantar

LINQ merupakan fitur baru pada Vb 9.0, dan fitur ini sangat membantu para programmer dalam membangun suatu aplikasi. Lalu apakah LINQ itu dan mahluk seperti apakah LINQ?? Mungkin masih banyak sekali di antara pembaca yang belum tau apa itu LINQ dan fungsi-fungsinya. . Ada beberapa macam LINQ Provider seperti LINQ to Object, LINQ to XML, LINQ to SQL, LINQ to Datasets, tapi pada artikel ini kita akan membuat aplikasi dengan menggunakan LINQ to SQL.

Sesuai dengan judul diatas, dengan LINQ akan memudahkan kita dalam membuat aplikasi. Artikel ini akan menjelaskan step by step cara pembuatan aplikasi, dari pembuatan database dengan SQL Server 2005 Express, bekerja dengan Object

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

Relational Designer, sampai dengan penggunaan fungsi-fungsi LINQ untuk aplikasi.

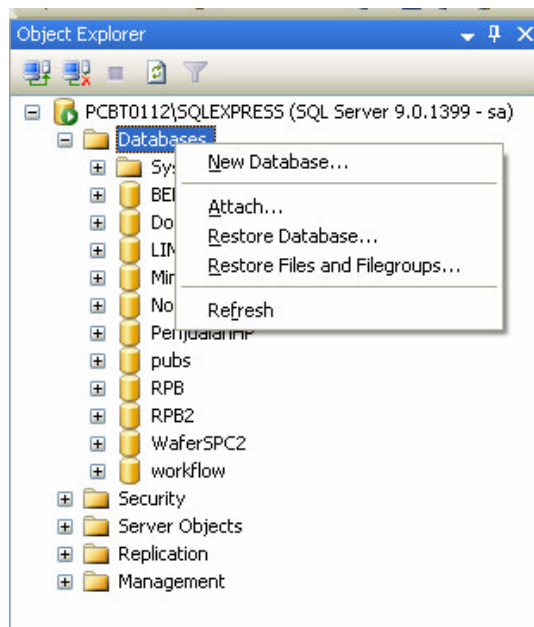
Untuk memudahkan pengerjaan latihan download terlebih dahulu project untuk artikel ini disini.

http://geeks.netindonesia.net/files/folders/source_codes_vb/default.aspx

II. Pembuatan Database

Sebelum masuk cara penggunaan LINQ, terlebih dahulu kita persiapkan database dengan menggunakan Management Studio Express. Ikuti langkah dibawah :

1. Buka Management Studion
2. Buat sebuah database baru, dengan cara klik kanan Root **Database** dan pilih **New Database**, dengan nama “**BELAJARLINQ**”



3. Lalu expand database “BELAJARLINQ”, pada root table klik kanan New Table. Dan ketikkan field-field dan skema table seperti gambar dibawah, dan simpan dengan nama **T_User**.

Table - dbo.T_User		Summary
Column Name	Data Type	Allow Nulls
UserAccount	varchar(20)	<input type="checkbox"/>
Password	varchar(15)	<input type="checkbox"/>
		<input type="checkbox"/>

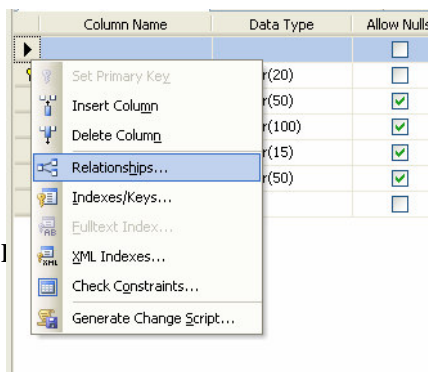
MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

4. Buka table T_User dengan cara klik kanan > **Open Table**, lalu masukkan data sebagai berikut :UserAccount :Jun, Password :12345
5. Lalu ikuti langkah-langkah diatas untuk membuat table-table seperti dibawah, simpan nama table-table berikut dengan nama T_Supplier dan T_Material.

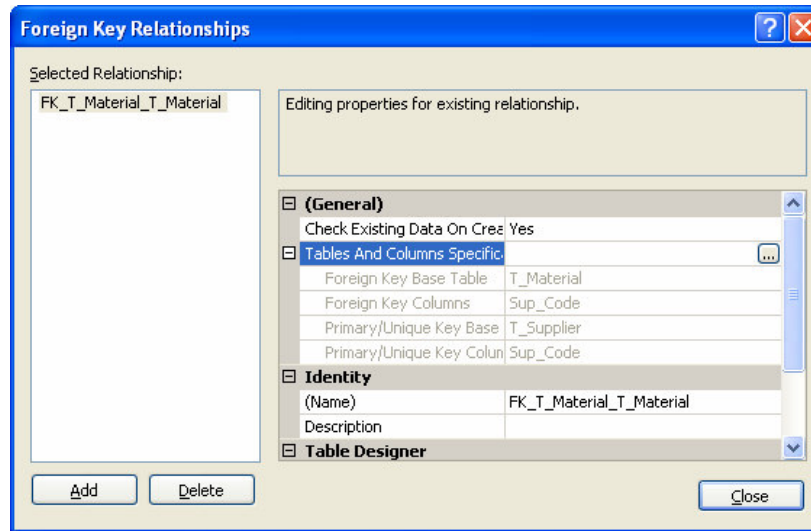
Column Name	Data Type	Allow Nulls
Sup_Code	varchar(20)	<input type="checkbox"/>
Sup_Nama	varchar(50)	<input checked="" type="checkbox"/>
Sup_Alnmt	varchar(100)	<input checked="" type="checkbox"/>
Sup_Telp	varchar(15)	<input checked="" type="checkbox"/>
Sup_Person	varchar(50)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
Mat_Code	varchar(20)	<input type="checkbox"/>
Mat_Nama	varchar(50)	<input checked="" type="checkbox"/>
Mat_Desc	varchar(255)	<input checked="" type="checkbox"/>
Mat_Ukur	varchar(50)	<input checked="" type="checkbox"/>
Mat_Hrg	money	<input checked="" type="checkbox"/>
Mat_Qty	int	<input checked="" type="checkbox"/>
Mat_Min	int	<input checked="" type="checkbox"/>
Sup_Code	varchar(20)	<input type="checkbox"/>

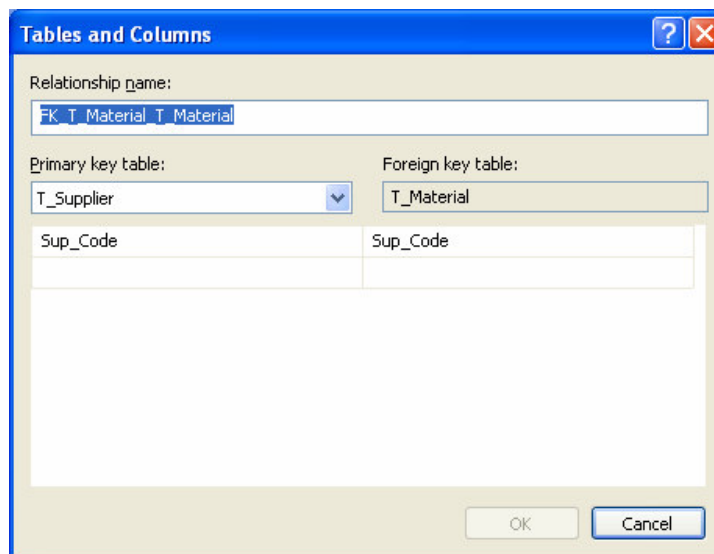
6. Setelah selesai membuat table seperti diatas (telah disimpan), sekarang kita akan membuat Relationships antara Table Material ke Supplier. Klik kanan pada field Sup_Code (T_Supplier), lalu pilih **Relationships**



7. Lalu jendela “Foreign Key Relationships” akan tampil.



8. Dan tekan tombol ellipsis (...), pada Bagian “Tables And Columns Spesification” dan jendela seperti dibawah akan tampil



MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

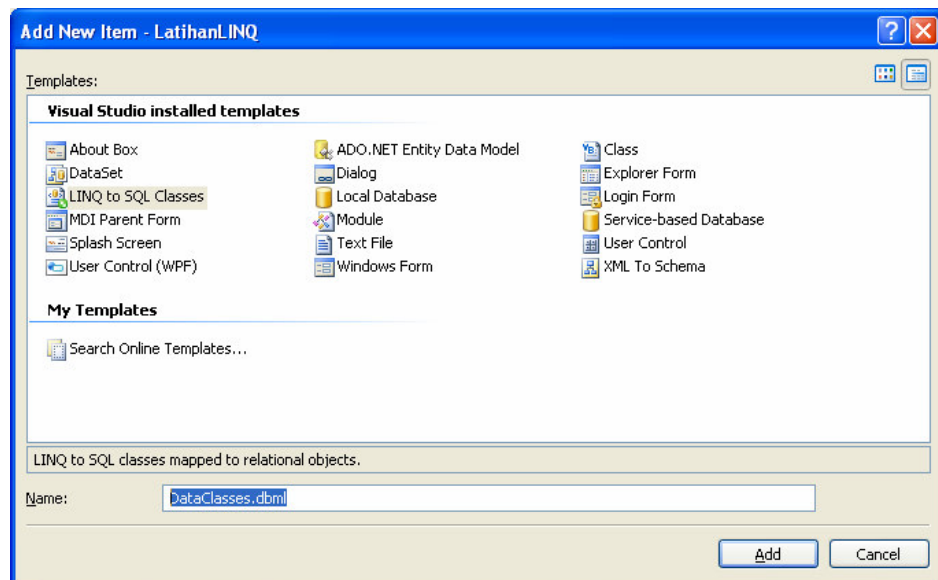
9. Pada primary key table pilih T_Supplier, dengan field “Sup_Code”, untuk foreign key table T_Material dengan field “Sup_Code”
10. Jika telah selesai membuat database dan table seperti langkah-langkah diatas, tutup Management Studio. Dan sekarang kita telah memili Database yang akan kita gunakan pada program kita selanjutnya.

III. Membuat Aplikasi dengan LINQ

Untuk tahap ini kita akan membuat Project baru pada Visual Studio 2008 express edition. Buka VS 2008 ed, Lalu pilih File > New Project, dan pada jendela “New Project”, pilih “Windows Forms Application”, dan Ubah Name menjadi “LatihanLINQ” dan klik OK.

Selanjutnya kita akan melakukan Mapping Table dengan Object Relational Designer.

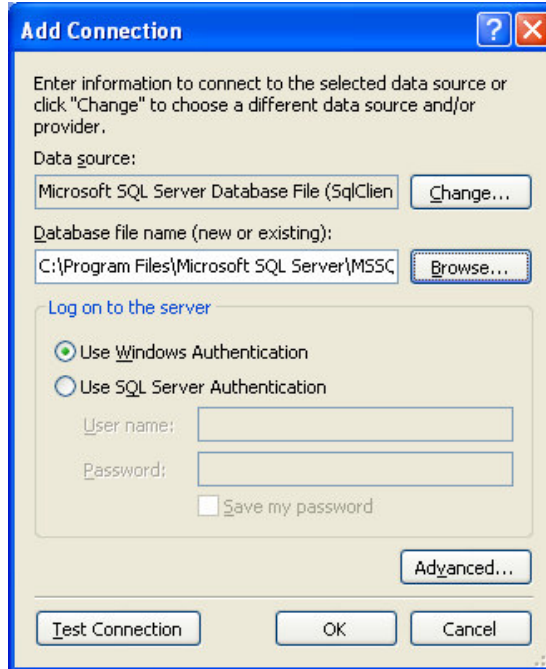
1. Tekan Ctrl+Shift+A maka jendela Add New Item akan muncul, pilih “LINQ to SQL Classes”, dan ubah Name menjadi “DataClasses.dbml”, dan tekan tombol Add



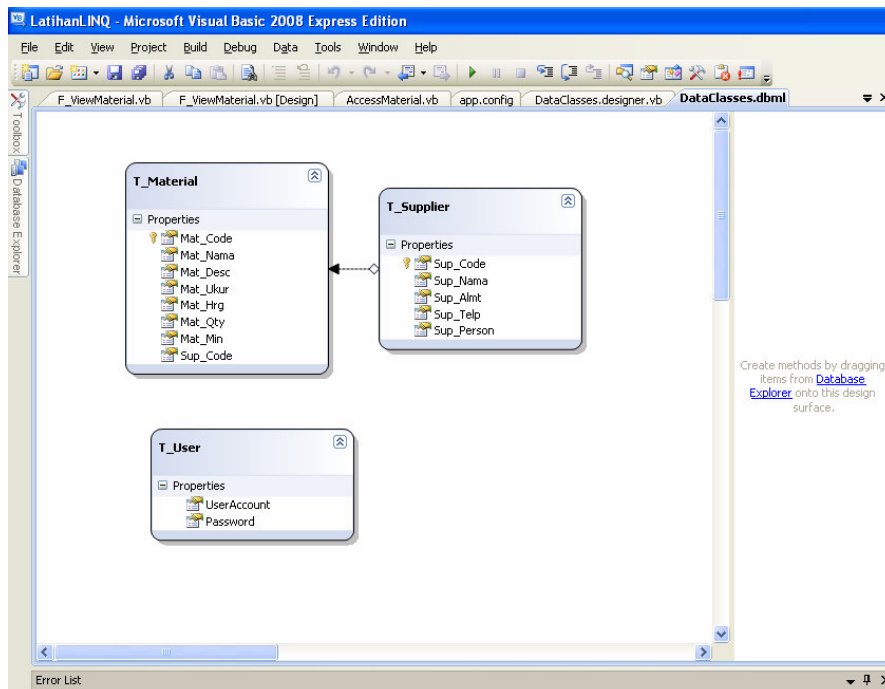
2. Lalu jendela Object Relational Designer akan tampil .
3. Buka jendela Database Explorer (Ctrl+Alt+S), lalu klik kanan pada “Data Connection” lalu pilih “Add Connection”.

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

4. Untuk Data Source ubah menjadi “Microsoft SQL Server Database File (SqlClient)”, dan browse database name (C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data) dan pilih “BELAJARLINQ.mdf”



5. Lalu Drag seluruh table kedalam jendela “Object Relational Designer”

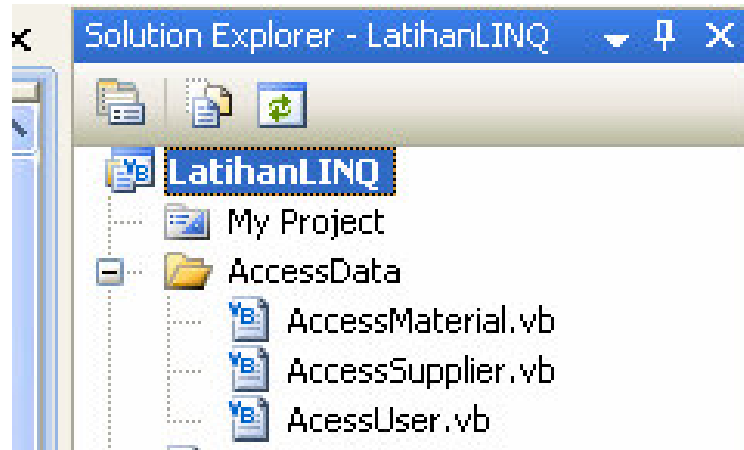


MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

6. Pada Solution Explorer buka file app.config, dan ganti Connection String seperti dibawah

```
<connectionStrings>  
  <add  
    name="LatihanLINQ.My.MySettings.BelajarLINQConnectionString"  
    connectionString="Data  
Source=localhost\SQLEXPRESS;Initial  
Catalog=BelajarLINQ;Integrated Security=True"  
    providerName="System.Data.SqlClient" />  
</connectionStrings>
```

7. Add sebuah Folder dengan nama AccessData, dengan cara klik kanan pada Solution Explorer LatihanLINQ > Add > New Folder. Dan tambahkan 3 buah class (Add > Class) dengan masing-masing nama **AccessMaterial,AccessSupplier dan AccessUser**



8. Ketikkan sintaks dibawah pada Class AccessUser.vb

```
Namespace AccessData
    Public Class AccessUser

Public Function FindByUsername(ByVal UserAccount As String) As List(Of T_User)
    Try
        'Pembuatan Object List dengan User dari class T_USer
        Dim User As New List(Of T_User) ()
        'Pembuatan DataContext
        Dim dc As New DataClassesDataContext
        'Melakukan query pada Table T_User dengan criteria nya adalah
        UserAccount
        Dim MatchUser = (From c In dc.GetTable(Of T_User) () _
            Where c.UserAccount = UserAccount Select c).SingleOrDefault

        If (MatchUser Is Nothing) Then 'jika MatchUser = kosong
            Throw New Exception("Username salah atau tidak di kenal")
        Else
            'Jika ada, maka nilainya akan disimpan kedalam List
            Dim Users As New T_User()
            Users.UserAccount = MatchUser.UserAccount
            Users.Password = MatchUser.Password
            User.Add(Users)
        End If
        Return User
    Catch ex As Exception
        Throw New Exception(ex.Message.ToString())
    End Try
End Function
End Class
End Namespace
```

Sintaks diatas di gunakan pada saat kita melakukan login. Pada saat user memasukkan username, maka program akan mencari username tersebut, apakah ada atau tidak didalam table. Jika tidak ada maka ada pesan yang akan muncul seperti diatas. Jika ada maka program akan menyimpan nilai username dan password dari hasil pencarian. Untuk proses selanjutnya (mengecek kebenaran password).

9. Selanjutnya pada Class AccessSupplier.vb ketikkan sintaks-sintaks dibawah ini.

```
'menampilkan seluruh isi dari table T_Supplier
Public Function GetSupplierTable() As System.Data.Linq.Table(Of T_Supplier)
    Dim dc As New DataClassesDataContext
    Return dc.GetTable(Of T_Supplier) ()
End Function
```

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

```
Public Function GetSupplierTableAsc() As List(Of T_Supplier)
    Dim dc As New DataClassesDataContext
    'menampilkan seluruh isi dari table T_Supplier dengan sorting Sup_Nama Ascending
    Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        Order By sup.Sup_Nama Select sup).ToList()
    Return hasil
End Function
```

```
'Menampilkan data dengan menggunakan fungsi Like
Public Function GetSupplierLike() As List(Of T_Supplier)
    Dim dc As New DataClassesDataContext
    Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        Where sup.Sup_Nama.StartsWith("Ja") Select sup).ToList() '=like %Ja

    'Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        'Where sup.Sup_Nama.EndsWith("os") Select sup).ToList() '=like os%

    'Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        'Where sup.Sup_Nama.Contains("a") Select sup).ToList() '=like %a%

    'untuk menggunakan SqlMethods, import terlebih dahulu Imports
    System.Data.Linq.SqlClient
    'Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        'Where SqlMethods.Like(sup.Sup_Nama, "%a%") Select sup).ToList() '=like %a%

    Return hasil
End Function
```

```
'Menampilkan data sejumlah 3 baris dan di mulai dari baris ke 2
Public Function GetSupplierRow() As List(Of T_Supplier)
    Dim dc As New DataClassesDataContext
    Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
        Order By sup.Sup_Code Select sup).Skip(2).Take(3).ToList()
    Return hasil
End Function
```

```
Public Function FindSupplierCode(ByVal Sup_Code As String, ByVal ProcessName As String) As List(Of T_Supplier)

    Try
        Dim Supplier As New List(Of T_Supplier) ()
        Dim dc As New DataClassesDataContext
        'Melakukan query pada Table T_Supplier dengan criteria nya adalah Sup_Code
        Dim MatchCode = (From c In dc.GetTable(Of T_Supplier) () _
            Where c.Sup_Code = Sup_Code Select c).SingleOrDefault

        If (MatchCode Is Nothing) Then 'jika MatchUser = kosong
            If ProcessName = "Find" Then
                Throw New Exception("Supplier Code Salah atau code yang di cari tidak ada")
            End If
        Else
            If ProcessName = "Save" Then
                Throw New Exception("Supplier Code sudah ada di dalam tabel")
            End If
            Dim suppliers As New T_Supplier
            suppliers.Sup_Code = MatchCode.Sup_Code
            suppliers.Sup_Nama = MatchCode.Sup_Nama
            suppliers.Sup_Almt = MatchCode.Sup_Almt
            suppliers.Sup_Telp = MatchCode.Sup_Telp
            suppliers.Sup_Person = MatchCode.Sup_Person
            'nilainya akan di masukkan kedalam List
            Supplier.Add(suppliers)
        End If

        Return Supplier
    Catch ex As Exception
        Throw New Exception(ex.Message.ToString())
    End Try
End Function
```

```
Public Shared Sub SupplierInsert(ByVal Supplier As T_Supplier)
    Try
        Dim dc As New DataClassesDataContext ()

        Dim Suppliers As New T_Supplier
        Suppliers.Sup_Code = Supplier.Sup_Code
        Suppliers.Sup_Nama = Supplier.Sup_Nama
        Suppliers.Sup_Almt = Supplier.Sup_Almt
        Suppliers.Sup_Telp = Supplier.Sup_Telp
        Suppliers.Sup_Person = Supplier.Sup_Person

        dc.T_Suppliers.InsertOnSubmit (Suppliers)
        dc.SubmitChanges ()

    Catch Ex As Exception
        Throw New Exception (Ex.Message.ToString ())
    End Try
End Sub
```

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

```
Public Shared Sub SupplierUpdate(ByVal Supplier As T_Supplier)
    Try
        Dim dc As New DataClassesDataContext
        Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
            Where sup.Sup_Code.Equals(Supplier.Sup_Code) Select sup).First

        hasil.Sup_Nama = Supplier.Sup_Nama
        hasil.Sup_Almt = Supplier.Sup_Almt
        hasil.Sup_Telp = Supplier.Sup_Telp
        hasil.Sup_Person = Supplier.Sup_Person

        dc.SubmitChanges()
    Catch Ex As Exception
        Throw New Exception(Ex.Message.ToString())
    End Try
End Sub
```

```
Public Shared Sub SupplierDelete(ByVal Supplier As T_Supplier)
    Try
        Dim dc As New DataClassesDataContext
        Dim hasil = (From sup In dc.GetTable(Of T_Supplier)() _
            Where sup.Sup_Code.Equals(Supplier.Sup_Code) Select sup).First

        dc.T_Suppliers.DeleteOnSubmit(hasil)
        dc.SubmitChanges()
    Catch Ex As Exception
        Throw New Exception(Ex.Message.ToString())
    End Try
End Sub
```

10. Untuk Class T_Material dapat dilihat pada project, yang dapat di download.

IV. Bekerja dengan windows form

1. Pada project yang kita buat, kita memiliki Form1, ubah properties dan tambahkan komponen-komponen sebagai berikut. Dan design form seperti pada gambar dibawah.

Text = Login

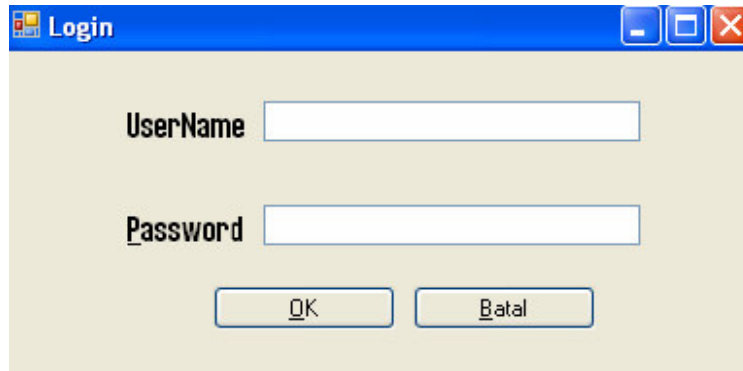
StartPosition = CenterScreen

Name = F_Login

Dan tambahkan 2 buah button, 2 buah label dan 2 buah textbox.

Atur posisi seperti pada gambar dibawah.

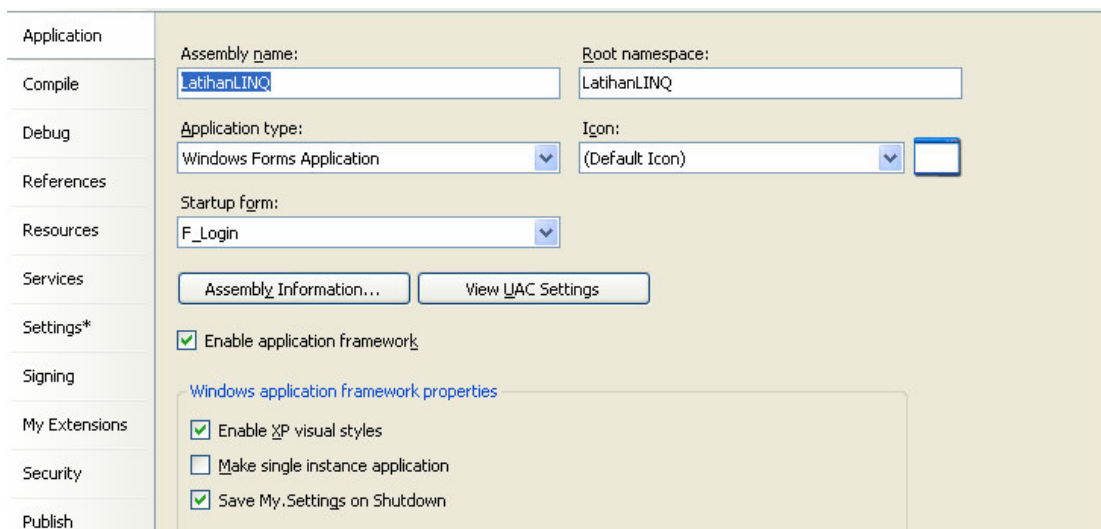
MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ



Ubah properties button.

Button1	Button2
Name = btnLogin	Name = btnCancel
Text = &OK	Text = &Batal
TextBox1	TextBox2
Name = txtUserName	Name = txtPassword
Text=kosong	Text =kosong
Label1	Label2
Text = UserName	Text = Password

Lalu pada menu Project > LATIHANLINQ properties, pilih **F_Login** sebagai Startup form.



MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

2. Tambahkan form baru (Add > Windows Form), dan tambahkan komponen MenuStrip. Lalu tambahkan menu item seperti dibawah.

File

> Exit

Inputan

- Supplier
- Material

View

- Supplier
- Material

Dan Ganti properties Name untuk form menjadi **F_Utama** dan Text = **Menu Utama**.



3. Kembali ke form Login, dan double click button login, dan ketik sintaks seperti dibawah.

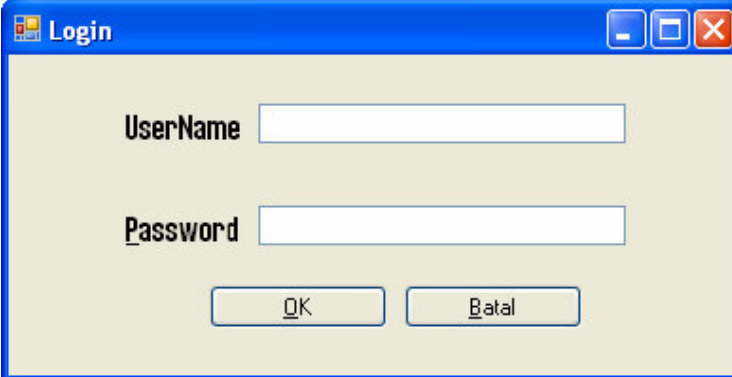
```
Try
Dim AccessUser As New AccessData.AcessUser
'Mengecek username apakah ada pada table (Class AccessUser)
'jika ada maka nilai akan di simpan didalam list
Dim list As List(Of T_User) = AccessUser.FindByUsername(txtUsername.Text.Trim)

If list.First.Password <> txtPassword.Text.Trim Then
    MsgBox("Password salah", MsgBoxStyle.Exclamation, "LINQ")
    Exit Sub
Else
    Me.ShowInTaskbar = False
    Me.Visible = False
    Dim F_utama As New F_Utama
    F_utama.Show()
    txtUsername.Clear()
    txtPassword.Clear()
End If
Catch ex As Exception
    MsgBox(ex.Message, MsgBoxStyle.Exclamation, "LINQ")
End Try
```

4. Dan double click button batal, dan ketikkan sintaks berikut `Me.Close()`
5. Jalankan program dan masukkan username dan password sesuai dengan diatas.

UserAccount :jun, Password :12345

Apakah program berjalan dengan sempurna, jika username dan password dimasukkan dengan benar sesuai pada table, maka form utama akan tampil.



MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ



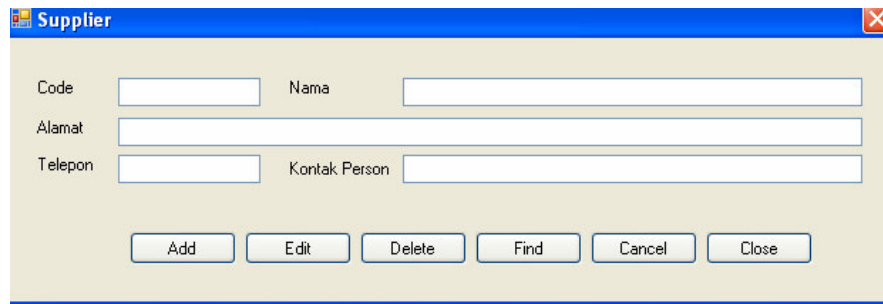
6. Jika sudah berhasil maka kita akan melanjutkan ke tahap selanjutnya, bagi yang belum, di lihat terlebih dahulu error yang muncul, apakah karena username / password salah, atau error karena kesalahan coding.
7. Tambahkan form baru, dan tambahkan 5 TextBox, 6 Button dan 5 Label, dan ubah properties Form dan komponen seperti table dibawah.

Form	
Name = F_Supplier Text = Supplier Start Position = CenterScreen	
TextBox1	TextBox2
Name = txtCode Text=kosong	Name = txtNama Text =kosong
TextBox3	TextBox4
Name = txtAlamat Text=kosong	Name = txttelepon Text =kosong
TextBox5	
Name = txtkontak Text=kosong	

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Label1	Label2
Text = Code	Text = Nama
Label3	Label4
Text = Alamat	Text = Telepon
Label5	
Text = Kontak Person	
Button1	Button2
Name = btnAdd Text = Add	Name = btnEdit Text = Edit
Button3	Button4
Name = btnDelete Text = Delete	Name = btnFind Text = Find
Button5	Button6
Name = btnCancel Text = Cancel	Name = btnClose Text = Close

Dan atur posisi komponen-komponen diatas seperti gambar dibawah.



8. Tekan F7 untuk menampilkan jendela Code, dan ketikkan sintaks berikut dibawah

```
Public Class F_Supplier
```

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

```
'Mendklarasikan Class AccessSuplier
Dim AccessSupplier As New AccessData.AccessSupplier

Private Sub ClearTextBox()
    'membuat procedure untuk menghapus isi dari textbox
    txtCode.Clear()
    txtNama.Clear()
    txtAlamat.Clear()
    txtTelepon.Clear()
    txtKontak.Clear()
End Sub
```

9. Double Click seluruh button, dan ketikkan sintaks-sintaks berikut sesuai dengan proses.

a. Button Add

```
Try
    Dim Supplier As New T_Supplier
    'untuk mengecek apakah code supplier sudah ada didalam table..
    'jika ada maka akan tampil pesan error
    Dim list As List(Of T_Supplier) = AccessSupplier.FindSupplierCode(txtCode.Text, "Save")

    Dim mYes_No As String = MsgBox("Do you want save the data ?", vbYesNo, "LINQ")
    If mYes_No = vbYes Then
        Supplier.Sup_Code = txtCode.Text
        Supplier.Sup_Nama = txtNama.Text
        Supplier.Sup_Almt = txtAlamat.Text
        Supplier.Sup_Telp = txtTelepon.Text
        Supplier.Sup_Person = txtKontak.Text
        'Melakukan proses simpan
        AccessData.AccessSupplier.SupplierInsert(Supplier)
        ClearTextBox()
    End If

Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

b. Button Update

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

```
Try
  Dim mYes_No As String = MsgBox("Do you want update the data ?", vbYesNo, "LINQ")

  If mYes_No = vbYes Then
    Dim Supplier As New T_Supplier

    Supplier.Sup_Code = txtCode.Text
    Supplier.Sup_Nama = txtNama.Text
    Supplier.Sup_Almt = txtAlamat.Text
    Supplier.Sup_Telp = txtTelepon.Text
    Supplier.Sup_Person = txtKontak.Text
    'Melakukan Proses Update
    AccessData.AccessSupplier.SupplierUpdate(Supplier)
    ClearTextBox()
  End If

Catch ex As Exception
  MsgBox(ex.Message)
End Try
```

c. Button Delete

```
Try
  Dim mYes_No As String = MsgBox("Do you want delete the data ?", vbYesNo, "LINQ")
  If mYes_No = vbYes Then
    Dim supplier As New T_Supplier
    supplier.Sup_Code = txtCode.Text
    AccessData.AccessSupplier.SupplierDelete(supplier)
    ClearTextBox()
  End If

  Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

d. Button Find

```
Try

  Dim list As List(Of T_Supplier) = AccessSupplier.FindSupplierCode(txtCode.Text.Trim, "Find")

  txtCode.Text = list.First.Sup_Code
  txtNama.Text = list.First.Sup_Nama
  txtAlamat.Text = list.First.Sup_Almt
  txtTelepon.Text = list.First.Sup_Telp
  txtKontak.Text = list.First.Sup_Person

Catch ex As Exception
  MsgBox(ex.Message)
End Try
```

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

e. Button Cancel

```
ClearTextBox()
```

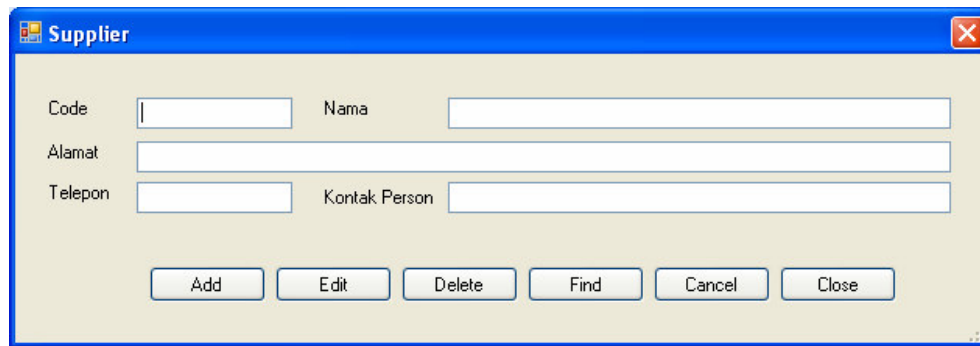
f. Button Close

```
Me.Close()
```

10. Buka Form Utama, dan Double click menu item Inputan > Supplier pada MenuStrip. Ketik sintaks berikut.

```
Dim F_Supplier As New F_Supplier  
F_Supplier.ShowDialog()
```

11. Jalankan program, setelah login lakukan seluruh proses pada form supplier (Add, Edit, Delete, Find)



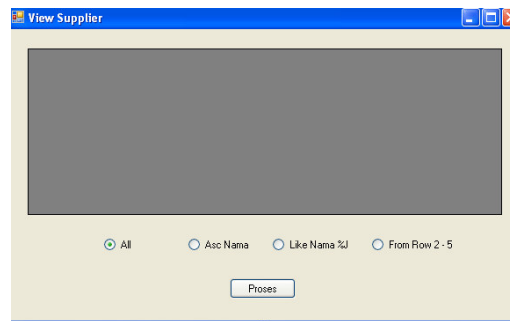
12. Jika sudah selesai dan benar-benar jalan kita lanjutkan ke Form View, dimana form untuk menampilkan data dalam datagridview. Untuk form material bisa melihat contoh pada project

13. Add sebuah form dan tambahkan 1 buah DataGridView dan 4 RadioButton, 1 button dan ubah properties sebagai berikut.

Form	
Name = F_ViewSupplier	
Text = View Supplier	
Start Position = CenterScreen	
RadioButton1	RadioButton2
Name = RbAll	Name = RbAsc
Text=All	Text= Asc Nama
Checked = True	Checked = False

RadioButton3	RadioButton4
Name = RbLike Text= Like Nama %J Checked = False	Name = RbRow Text= From Row 2 - 5 Checked = False
Button1	
Name = BtnProses Text= Proses	

Dan atur posisi nya seperti gambar dibawah.



14. Ketikkan sintaks berikut dibawah `Public Class F_ViewSupplier`

```
Dim AccessSupplier As New AccessData.AccessSupplier

Private Sub FormatGridWithBothTableAndColumnStyles ()
    Me.DataGridView1.DefaultCellStyle.ForeColor = Color.Coral
    Me.DataGridView1.RowsDefaultCellStyle.BackColor = Color.AliceBlue
    Me.DataGridView1.GridColor = Color.Blue
    Me.DataGridView1.BorderStyle = BorderStyle.Fixed3D
    Me.DataGridView1.RowHeadersBorderStyle = BorderStyle.Fixed3D
    Me.DataGridView1.ColumnHeaderHeightSizeMode =
        DataGridViewColumnHeadersHeightSizeMode.AutoSize
    DataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells
    DataGridView1.Columns(0).HeaderText = "Code"
    DataGridView1.Columns(1).HeaderText = "Nama"
    DataGridView1.Columns(2).HeaderText = "Alamat"
    DataGridView1.Columns(3).HeaderText = "Telp"
    DataGridView1.Columns(4).HeaderText = "Contact Person"
End Sub
```

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

15. Double click form, lalu ketikkan sintaks berikut :

```
Try
DataGridView1.DataSource = AccessSupplier.GetSupplierTable
    FormatGridWithBothTableAndColumnStyles()
    Catch ex As Exception
        MsgBox(ex.Message)
End Try
```

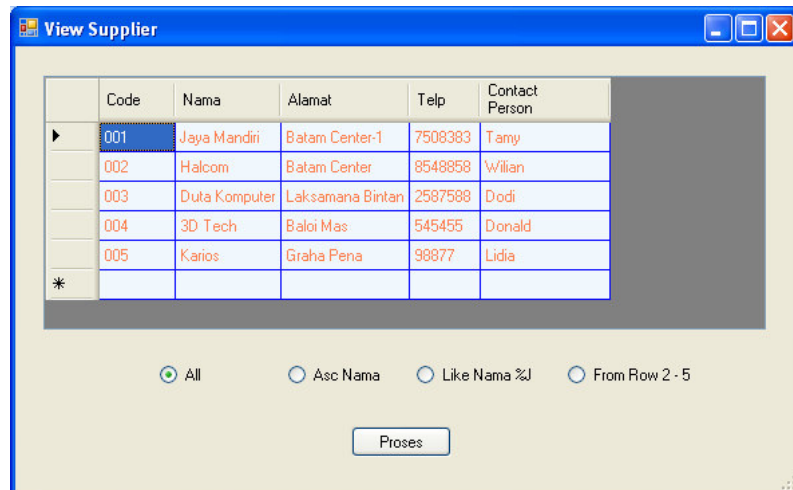
16. Double click button proses.

```
'memilih radio button yang aktif (Checked =True)
Select Case True
    Case RbAll.Checked = True
        DataGridView1.DataSource = AccessSupplier.GetSupplierTable
    Case RbAsc.Checked = True
        DataGridView1.DataSource = AccessSupplier.GetSupplierTableAsc
    Case RbLike.Checked = True
        DataGridView1.DataSource = AccessSupplier.GetSupplierLike
    Case RbRow.Checked = True
        DataGridView1.DataSource = AccessSupplier.GetSupplierRow
End Select
```

17. Buka Form Utama, dan Double click menu item View > Supplier pada MenuStrip. Ketik sintaks berikut.

```
Dim F_Supplier As New F_Supplier
F_Supplier.ShowDialog()
```

18. Jalankan program dan coba seluruh proses



19. Untuk form View Material dapat dilihat pada project.

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Penutup

Mudah-mudahan artikel ini membuat kita semua menjadi bertambah pengetahuanya.

Referensi

1. www.msdn.microsoft.com
2. www.planetsourcecode.com
3. www.codeproject.com
4. www.aspnet.com

Masih banyak lagi referensi yang ada di Intenet. Anda tinggal cari di www.Google.com. Dengan kata kunci “**tutorial VB.Net**”



Sinopsis buku “Panduan Lengkap Menjadi Programmer”

Setelah mengetahui apa itu Microsoft.NET Framework, Microsoft Visual Basic.NET, ADO.NET, Database dan Microsoft Access, dan fungsi-fungsi serta dasar penggunaannya, ternyata menjadi programmer VB.NET itu mudah! Sebab, dari dasar pengetahuan tersebut, kita sudah bisa membuat suatu program yang kita inginkan. Misalnya, membuat aplikasi toko, seperti form penjualan dan pembelian. Visual Basic.NET (VB.NET) merupakan program generasi selanjutnya dari Visual Basic. VB.NET memungkinkan kita untuk membangun aplikasi database client atau server dengan performa tinggi.

Umumnya, para programmer banyak menggunakan VB.NET untuk membuat aplikasi atau program yang mereka butuhkan. Pasalnya, program ini memiliki kelebihan dan kemudahan tertentu dibandingkan dengan program-program lainnya yang sejenis.

Selanjutnya, langkah apa saja yang mesti diketahui dan dilakukan agar kita bisa menjadi programmer VB.NET? Mediakita menerbitkan buku *Panduan Lengkap Menjadi Programmer* yang ditulis oleh **Junindar**. Buku ini berisi tutorial pembuatan program, pembuatan laporan, sampai dengan melakukan *deployment* program. Melalui buku ini, Anda bisa menjadi seorang programmer VB.NET yang handal. Sebab, buku ini merupakan buku yang komplit untuk mengantarkan seseorang dari tidak tahu apa-apa tentang VB.NET menjadi seorang programmer.

Memang, saat ini, banyak sekali buku-buku tentang pemrograman VB.NET, tetapi sangat jarang yang menjelaskan secara detail tentang fungsi-fungsi dan pemrograman di dalam VB.NET. Di buku ini, secara terperinci Anda akan dibimbing bagaimana membuat database, membuat program (*create program*), tampilan login, menu, form, laporan, sampai dengan membuat file setup (*installer*).

Selain itu, buku ini juga dilengkapi CD berisikan *source code* pembuatan program sehingga lebih memudahkan Anda dalam berlatih.

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar

Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Informatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Microsoft MVP VB pertanggal 1 oktober 2009. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, LINQ, VBA. Reporting : Crystal Report (Under Program). Database : Access, paradox dan SQL Server 00/05. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Swasta di BATAM sebagai Software Development. Mempunyai moto hidup : **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.



Kritik dan saran kirim ke : junindar@gmail.com

MEMBUAT APLIKASI MENJADI LEBIH MUDAH DENGAN LINQ

Junindar