

VB 9 With MySQL

Junindar

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

E-Mail : xx_lead@yahoo.com

junindar@gmail.com

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.

I. Pengantar

MySQL merupakan salah satu database open source yang paling populer di dunia dikarenakan kecepatannya tinggi, high reliability dan user friendly. Untuk mengetahui detail dari MySQL bisa menuju ke link berikut <http://en.wikipedia.org/wiki/MySQL> dan www.mysql.com

Setelah membaca dan mengetahui apa saja kelebihan dan fitur yang terdapat pada MySQL akan membawa berkeinginan langsung berinteraksi dengan MySQL.

Tapi bagaimana jika kita merupakan programmer Vb.Net, apakah bisa melakukan koneksi ke MySQL. Pertanyaan ini sering sekali datang ke email saya, jadi dengan adanya artikel ini mudah-mudahan bisa menjawab semua pertanyaan mengenai ini. Dan didalam artikel ini selain menjelaskan bagaimana melakukan koneksi dengan MySQL juga terdapat beberapa tip dan trick seputar pengolahan data.

II. Koneksi ke MySQL

Untuk melakukan koneksi ke MySQL dengan .Net sebenarnya ada dua cara yaitu dengan menggunakan ODBC dan kedua menggunakan file MySql.Data.dll yang

kita import terlebih dahulu. Tapi dalam artikel ini saya akan menjelaskan dengan menggunakan metode ke dua.

Note : Di asumsikan kita semua sudah menginstall MySQL di komputer. Untuk yang belum memiliki MySQL bisa mendownload di sini (<http://dev.mysql.com/downloads/>), Dan juga di asumsikan sudah mengerti bagaimana membuat Database dan Table pada MySQL, untuk yang belum mengerti sama sekali bisa belajar disini <http://dev.mysql.com/doc/>

Terlebih dahulu kita buat database dengan nama “JMB“ dengan dua buah table seperti dibawah.

| No | Field Name | Field Type | Field Length |
|----|---------------|------------|--------------|
| 1 | UserName (PK) | Varchar | 15 |
| 2 | Password | Varchar | 10 |

Table Users

| No | Field Name | Field Type | Field Length |
|----|-------------|------------|--------------|
| 1 | B_ISBN (PK) | Varchar | 15 |
| 2 | B_Title | Varchar | 50 |
| 3 | B_Authors | Varchar | 50 |
| 4 | B_Publisher | Varchar | 50 |
| 5 | B_Price | Float | 50 |

Table tb buku

Isikan data pada table users.

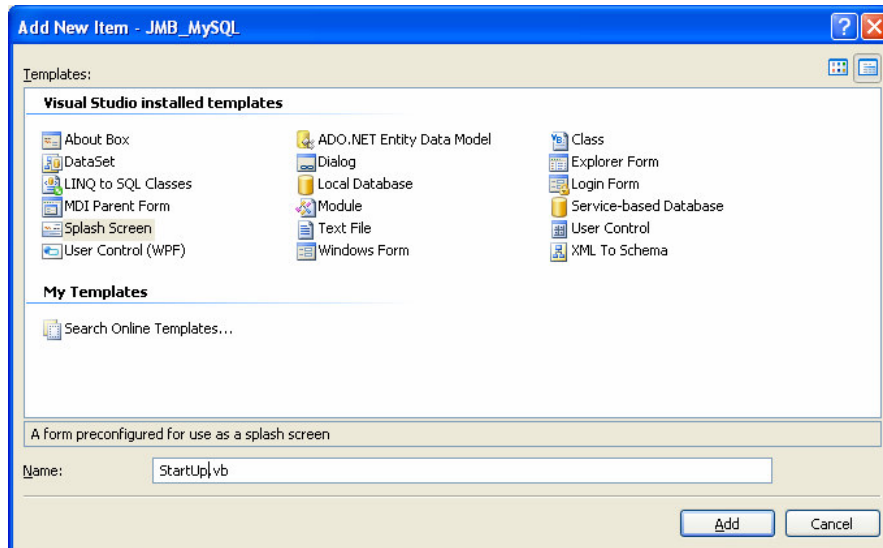
Setelah selesai membuat 2 buah table seperti diatas, sekarang saat nya kita melakukan koneksi dengan MySQL.

Pertama-tama adalah kita harus mendonwload MySQL Connector/Net, ini bisa di download disini (<http://dev.mysql.com/downloads/connector/net/1.0.html> untuk vb.net 2003), tapi bagi yang menggunakan versi 2005 keatas (2.0 keatas) bisa download disini

<http://dev.mysql.com/downloads/connector/net/5.2.html> . Download Windows Installer (.exe) lalu install ke PC.

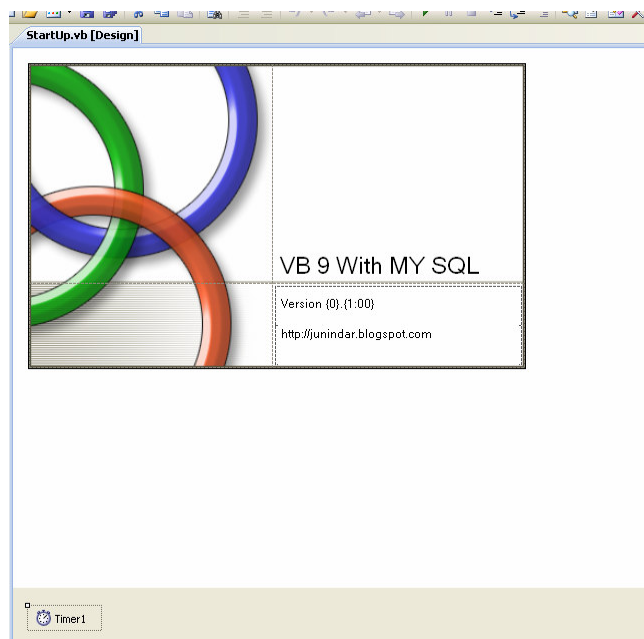
Buat project baru dengan nama JMB_MySQL (untuk nama project tidak di haruskan sama). Delete Form1 pada project.

Lalu Add>New Item, lalu pilih Splash Screen ganti nama nya menjadi **StartUp.vb**

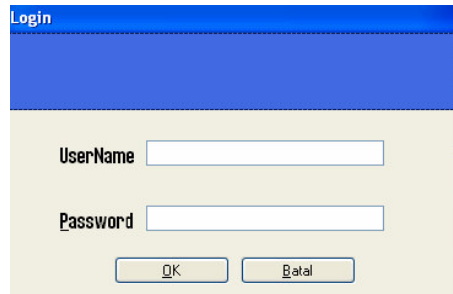


Setelah itu Add>New Form dan ubah nama form menjadi **F_Login.vb**

Pada form StartUp tambahkan control Timer dan ubah Properties Enabled = **True**



Sedangkan untuk form Login design seperti gambar dibawah ini.



Add Reference (Project>Add Reference). Pada jendela Add Reference, klik tab “Projects” dan Browse. Lalu cari file **MySql.Data.dll** di *C:\Program Files\MySQL\MySQL Connector Net n.n.n\bin\NET N.N* (N.N tergantung dari versi .netframework) dan OK.

Dengan melakukan hal seperti diatas kita sudah siap untuk melakukan proses pada MySQL. Tetap ikuti langkah-langkah pada artikel ini dengan benar.

Tambahkan folder pada project kita dengan cara Add>New Folder, ubah namanya menjadi AccessData. Pada folder AccessData tambahkan sebuah Class dan ubah namanya menjadi DataBaseConnection.vb. Ketikkan sintaks dibawah pada Class DataBaseConnection :

```
Imports MySql.Data.MySqlClient

Namespace AccessData
    Public Class DataBaseConnection
        Dim conn As New MySqlConnection("server=pcbt0112;user id=root;
        database=JMB; pooling=false;Connection Timeout=5")

        Public Function open() As MySqlConnection

            If conn.State <> ConnectionState.Open Then
                conn.Open()
            End If
            Return conn

        End Function

        Public Function close() As MySqlConnection
            conn.Close()
            Return conn

        End Function

    End Class
End Namespace
```

Pada Class diatas, kita telah membuat Class khusus untuk melakukan koneksi dengan MySQL. Class ini di gunakan oleh banyak Class lain yang melakukan koneksi ke database. Oleh karena itu kita telah melakukan suatu penghematan waktu karena kita tidak perlu lagi mengetikkan berulang-ulang koneksi ke database. Yang harus di perhatikan adalah lakukan Imports Mysql

```
Imports MySql.Data.MySqlClient
```

Ganti Nilai server sesuai dengan nama dimana MySQL di install.

Buat Folder baru dan ganti nama nya menjadi “**Entity**“. Tambahkan Class baru dalam folder Entity dan ganti namanya menjadi User.vb. Ketikkan sintaks dibawah pada class User

```
Public Class User

    Private m_username As String
    Private m_password As String

    Public Property UserName() As String
        Get
            'Get : hanya ijinan akses data
            Return m_username
        End Get
        Set(ByVal value As String)
            If value = "" Then
                Throw New Exception("Silahkan masukkan username")
            End If
            'Set : mengijinkan ubah data
            m_username = value
        End Set
    End Property

    Public Property Password() As String
        Get
            Return m_password
        End Get
        Set(ByVal value As String)
            If value = "" Then
                Throw New Exception("Silahkan masukkan password")
            End If
            m_password = value
        End Set
    End Property


```

```
Throw New Exception("Silahkan masukkan username")
```

```
Throw New Exception("Silahkan masukkan password")
```

Merupakan Handling Exeption yang kita buat, jika nilai dari pada value kosong, maka Pesan error akan muncul. Hal ini sangat membantu sekali jika value diatas

sering di gunakan. Maka kita juga telah melakukan penghematan waktu dari pengetikan yang berulang-ulang.

Tambahkan Class baru pada folder AccessData dan ubah namanya menjadi AccessUser. Ketikkan sintaks dibawah ini pada class AccessUser.

```
Imports MySql.Data.MySqlClient

Namespace AccessData

    Public Class AccessUser
        Private myconnection As New AccessData.DataBaseConnection

        Public Function FindByUsername (ByVal Username As String) As List (Of User)
            Dim User As New List (Of User) ()
            Try
                'String SQL untuk mencari username dari table users
                Dim strSQL As String = "SELECT * FROM Users where Username=@Username"
                Dim myCommand As New MySqlCommand (strSQL, myconnection.open)
                myCommand.Parameters.Add ("@Username", MySqlDbType.VarChar).Value = Username
                Dim rdr As MySqlDataReader = myCommand.ExecuteReader ()

                If rdr.Read = False Then 'jika user yang dicari tidak ada
                    Throw New Exception ("Username salah atau tidak dikenal")
                Else
                    Dim Users As New User ()
                    Users.UserName = rdr ("Username").ToString ()
                    Users.Password = rdr ("Password").ToString ()
                    User.Add (Users)
                End If
            Catch sqllex As MySqlException
                Throw New Exception (sqllex.Message.ToString ())
            End Try
            Return User
        End Function

    End Class

End Namespace
```

Class ini berfungsi untuk memvalidasi UserName pada saat login, apakah user yang di masukkan ada pada table. Sebelum kita masuk ke form login, buka form Startup double click control timer dan ketikkan sintaks dibawah ini

```
val += 4
If val = 100 Then
    Timer1.Dispose()
    Me.Visible = False
    Dim Login As New F_Login
    Login.Show()
End If
```

Untuk detail penggunaan Class ini sekarang Double Click button OK pada form Login, dan ketikkan sintak di bawah ini.

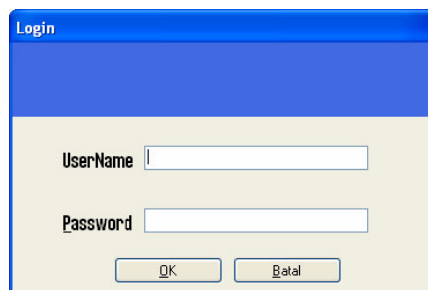
```
Try
    Dim AccessUser As New AccessData.AccessUser
    Dim User As New User

    User.UserName = txtUsername.Text.Trim
    User.Password = txtPassword.Text.Trim

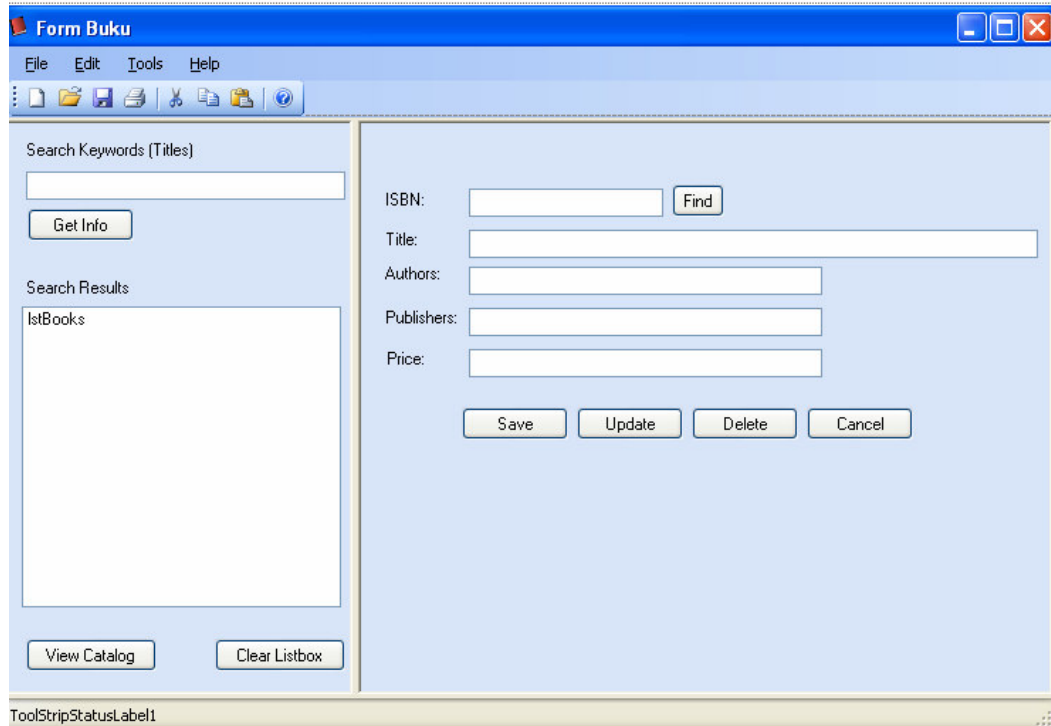
    Dim list As List(Of User) =
AccessUser.FindByUsername(txtUsername.Text.Trim)

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

Sebelum dijalankan Add>New Form dan ubah nama nya menjadi F_Buku. Dan pada Menu Project > JMB_MySql Properties, pada Startup form ubah menjadi StartUp. Dan jalan kan program. Masukkan Username dan Password. Apakah sudah berjalan dengan baik??



Jika sudah berhasil mari kita lanjutkan ke proses berikut nya. Aktifkan form buku dan design formnya seperti dibawah ini.



Setelah selesai membuat design seperti diatas, kita akan melanjutkan coding kita. Pada folder AccessData tambahkan Class baru dan ubah namanya menjadi DataControl.vb. Dan ketikkan sintaks dibawah ini pada class tersebut.

```
Imports MySql.Data.MySqlClient

Namespace AccessData

    Public Class DataControl
        Private myconnection As New AccessData.DataBaseConnection
        Public Function GetDataSet(ByVal SQL As String) As DataSet

            Dim adapter As New MySqlDataAdapter(SQL, myconnection.open)
            Dim myData As New DataSet

            adapter.Fill(myData, "Data")

            Return myData
        End Function
    End Class

End Namespace
```


Class diatas akan sangat bermanfaat sekali bila aplikasi yang kita buat berskala besar. Dengan menggunakan class diatas kita bisa menampilkan seluruh table pada database hanya dengan mengetikkan Sql nya saja. Selanjutnya data akan di isi ke Dataset.

Selanjutnya pada Folder entity tambahkan Class baru dan ubah namanya menjadi Buku.vb ketikkan sintak dibawah pada class buku.

```
Public Class Buku
    Private m_ISBN As String
    Private m_Title As String
    Private m_Authors As String
    Private m_Publisher As String
    Private m_Price As String

    Public Property ISBN() As String
        Get
            Return m_ISBN
        End Get
        Set(ByVal value As String)
            If value = "" Then
                Throw New Exception("Silahkan masukkan ISBN")
            End If
            m_ISBN = value
        End Set
    End Property

    Public Property Title() As String
        Get
            Return m_Title
        End Get
        Set(ByVal value As String)
            m_Title = value
        End Set
    End Property

    Public Property Authors() As String
        Get
            Return m_Authors
        End Get
        Set(ByVal value As String)
            m_Authors = value
        End Set
    End Property

    Public Property Publisher() As String
        Get
            Return m_Publisher
        End Get
        Set(ByVal value As String)
            m_Publisher = value
        End Set
    End Property
End Class
```

```
Public Property Price() As Double
    Get
        Return m_Price
    End Get
    Set(ByVal value As Double)
        m_Price = value
    End Set
End Property

End Class
```

Dan buat Class baru pada Folder AccessData dan ubah namanya menjadi AccessBuku. Ketikkan sintaks dibawah pada class AccessBuku

```
Imports MySql.Data.MySqlClient

Namespace AccessData

    Public Class AccessBuku
        Private myconnection As New AccessData.DataBaseConnection

        Public Function BukuInsert (ByVal Buku As Buku) As MySqlDataReader
            Try

                Dim StrSQL As String = "INSERT INTO tbBuku " & _
                    " (B_ISBN,B_Title,B_Authors,B_Publisher,B_Price) " & _
                    " VALUES (@B_ISBN,@B_Title,@B_Authors,@B_Publisher,@B_Price);"

                Dim myCommand As MySqlCommand = New MySqlCommand(StrSQL, myconnection.open)
                myCommand.CommandType = CommandType.Text

                Dim parameterB_ISBN As MySqlParameter = New MySqlParameter("@B_ISBN",
                    MySqlDbType.VarChar, 15)
                parameterB_ISBN.Value = Buku.ISBN

                Dim parameterB_Title As MySqlParameter = New MySqlParameter("@B_Title",
                    MySqlDbType.VarChar, 50)
                parameterB_Title.Value = Buku.Title

                Dim parameterB_Authors As MySqlParameter = New MySqlParameter("@B_Authors",
                    MySqlDbType.VarChar, 50)
                parameterB_Authors.Value = Buku.Authors

                Dim parameterB_Publisher As MySqlParameter = New
                    MySqlParameter("@B_Publisher", MySqlDbType.VarChar, 50)
                parameterB_Publisher.Value = Buku.Publisher

                Dim parameterB_Price As MySqlParameter = New MySqlParameter("@B_Price",
                    MySqlDbType.Float)
                parameterB_Price.Value = Buku.Price
                With myCommand.Parameters
                    .Add(parameterB_ISBN)
                    .Add(parameterB_Title)
                    .Add(parameterB_Authors)
                    .Add(parameterB_Publisher)
                    .Add(parameterB_Price)
                End With

                Dim result As MySqlDataReader = myCommand.ExecuteReader(CommandBehavior.CloseConnection)
                Return result
            End Try
        End Function
    End Class
End Namespace
```

```
        Catch SqlEx As MySqlException
            Throw New Exception (SqlEx.Message.ToString())
        End Try
End Function

Public Function BukuUpdate (ByVal Buku As Buku) As MySqlDataReader
    Try

        Dim StrSQL As String = "Update tbBuku set " & _
"B_Title=@B_Title,B_Authors=@B_Authors,B_Publisher=@B_Publisher,B_Price=@B_Price " & _
" where B_ISBN=@B_ISBN "

        Dim myCommand As MySqlCommand = New MySqlCommand (StrSQL, myconnection.open)
myCommand.CommandType = CommandType.Text

        Dim parameterB_Title As MySqlParameter = New MySqlParameter ("@B_Title",
MySqlDbType.VarChar, 50)
parameterB_Title.Value = Buku.Title

        Dim parameterB_Authors As MySqlParameter = New MySqlParameter ("@B_Authors",
MySqlDbType.VarChar, 50)
parameterB_Authors.Value = Buku.Authors

        Dim parameterB_Publisher As MySqlParameter = New
MySqlParameter ("@B_Publisher", MySqlDbType.VarChar, 50)
parameterB_Publisher.Value = Buku.Publisher
Dim parameterB_Price As MySqlParameter = New MySqlParameter ("@B_Price", MySqlDbType.Float)
parameterB_Price.Value = Buku.Price

        Dim parameterB_ISBN As MySqlParameter = New MySqlParameter ("@B_ISBN",
MySqlDbType.VarChar, 15)
parameterB_ISBN.Value = Buku.ISBN

        With myCommand.Parameters
            .Add (parameterB_Title)
            .Add (parameterB_Authors)
            .Add (parameterB_Publisher)
            .Add (parameterB_Price)
            .Add (parameterB_ISBN)
        End With
        Dim result As MySqlDataReader =
myCommand.ExecuteReader (CommandBehavior.CloseConnection)
        Return result

        Catch SqlEx As MySqlException
            Throw New Exception (SqlEx.Message.ToString())
        End Try
End Function
```

```
Public Function BukuDelete(ByVal Buku As Buku) As MySqlDataReader
    Try

        Dim strSQL As String = "Delete From tbBuku Where B_ISBN=@B_ISBN "

        Dim myCommand As MySqlCommand = New MySqlCommand(strSQL, myconnection.open)
        myCommand.CommandType = CommandType.Text

        Dim parameterB_ISBN As MySqlParameter = New MySqlParameter("@B_ISBN",
MySqlDbType.VarChar, 15)
        parameterB_ISBN.Value = Buku.ISBN

        myCommand.Parameters.Add(parameterB_ISBN)

        Dim result As MySqlDataReader =
myCommand.ExecuteReader(CommandBehavior.CloseConnection)
        Return result

    Catch SqlEx As MySqlException
        Throw New Exception(SqlEx.Message.ToString())
    End Try

End Function

Public Function FindBukuByISBN(ByVal ISBN As String, ByVal ProcessName As String) As List(Of
Buku)

    Dim buku As New List(Of Buku) ()
    Try
        Dim strSQL As String = "SELECT * FROM tbbuku where B_ISBN=@B_ISBN"
        Dim myCommand As New MySqlCommand(strSQL, myconnection.open)
        myCommand.Parameters.Add("@B_ISBN", MySqlDbType.VarChar).Value = ISBN
        Dim rdr As MySqlDataReader = myCommand.ExecuteReader

        If rdr.Read = False Then
            If ProcessName = "Find" Then
                Throw New Exception("ISBN salah atau buku yang di cari tidak ada")
            End If
        Else
            If ProcessName = "Save" Then
                Throw New Exception("ISBN sudah ada di dalam tabel")
            End If
            Dim bukus As New Buku()
            bukus.ISBN = rdr("B_ISBN").ToString()
            bukus.Title = rdr("B_Title").ToString()
            bukus.Authors = rdr("B_Authors").ToString()
            bukus.Publisher = rdr("B_Publisher").ToString()
            bukus.Price = rdr("B_Price").ToString()
            buku.Add(bukus)
        End If
        rdr.Close()
    Catch sqlEx As MySqlException
        Throw New Exception(sqlEx.Message.ToString())
    End Try
    Return buku
End Function
```

```
Public Function FindBukuByTitle(ByVal Title As String) As List(Of Buku)
    Dim buku As New List(Of Buku)()
    Try
        Dim strSQL As String = "SELECT * FROM tbbuku where B_Title like
Concat(@B_Title, '%') "
        Dim myCommand As New MySqlCommand(strSQL, myconnection.open)
        myCommand.Parameters.Add("@B_Title", MySqlDbType.VarChar).Value = Title
        Dim rdr As MySqlDataReader = myCommand.ExecuteReader

        While rdr.Read()
            Dim bukus As New Buku()
            bukus.ISBN = rdr("B_ISBN").ToString()
            bukus.Title = rdr("B_Title").ToString()
            bukus.Authors = rdr("B_Authors").ToString()
            bukus.Publisher = rdr("B_Publisher").ToString()
            bukus.Price = rdr("B_Price").ToString()
            buku.Add(bukus)
        End While

        Catch sqllex As MySqlException
            Throw New Exception(sqllex.Message.ToString())
        End Try
        Return buku
    End Function

End Class
End Namespace
```

Setelah selesai mengetikkan sintaks panjang diatas sekarang saya akan menjelaskan tujuan dari sintaks diatas. Pada sintaks diatas kita telah membuat 5 buah Function dimana fungsinya berbeda-beda pertama Insert, Update, Delete, Mencari data buku berdasarkan ISBN dan mencari data buku berdasarkan Title (dengan menggunakan fungsi Like).

Mari buka form buku, ketikkan sintaks dibawah ini dibawah `Public Class F_Buku`

```
#Region "PrivateSub"
    Private Sub DisabledTextBox(ByVal nilai As Boolean)
        txtISBN.Enabled = nilai
        txtTitle.Enabled = nilai
        txtPrice.Enabled = nilai
        txtAuthors.Enabled = nilai
        txtPublisher.Enabled = nilai
    End Sub
    Private Sub ClearTextBox()
        txtISBN.Clear()
        txtTitle.Clear()
        txtPrice.Clear()
        txtAuthors.Clear()
        txtPublisher.Clear()
    End Sub
#End Region
```

Setelah selesai double klik button Save. Dan ketikkan sintaks dibawah ini.

```
Try
    Dim buku As New Buku
    Dim AccessBuku As New AccessData.AccessBuku
    Dim list As List(Of Buku) =
AccessBuku.FindBukuByISBN(txtISBN.Text.Trim, "Save")

    Dim mYes_No As String = MsgBox("Do you want save the data ?",
vbYesNo, "Buku")
    If mYes_No = vbYes Then
        buku.ISBN = txtISBN.Text
        Buku.Title = txtTitle.Text
        Buku.Authors = txtAuthors.Text
        Buku.Publisher = txtPublisher.Text
        Buku.Price = txtPrice.Text
        AccessBuku.BukuInsert (Buku)
        ClearTextBox ()
    End If

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

Masih ingat `Public Function FindBukuByISBN` yang kita buat pada Class AccessBuku ? disana kita menggunakan 2 buah variable (ISBN dan ProcessName), mungkin teman-teman semua mengerti jika ISBN digunakan untuk pencarian data berdasarkan ISBN buku. Tapi mungkin ada yang bertanya apa fungsi dari ProcessName? Processname merupakan tanda yang kita buat kepada program kita jika ProcessName nya Save, maka ISBN nya tidak boleh sama, oleh karena itu jika ISBN yang di masukkan sudah ada pada table maka pesan akan muncul, dan jika processname nya Find data table berdasarkan ISBN yang di masukkan akan ditampilkan. Berarti dengan Function tersebut kita bisa menggunakan untuk berbagai macam proses, tidak hanya untuk menampilkan data tapi bisa juga menjadi error handling.

Coba jalankan program dan masukkan data, apakah berjalan dengan baik?

Lalu double click button Update dan Delete ketikkan sintaks dibawah

Button Update

```
Try
    Dim mYes_No As String = MsgBox("Do you want update the data ?",
vbYesNo, "Buku")
    If mYes_No = vbYes Then
        Dim buku As New Buku
        Dim AccessBuku As New AccessData.AccessBuku
        buku.ISBN = txtISBN.Text
        buku.Title = txtTitle.Text
        buku.Authors = txtAuthors.Text
        buku.Publisher = txtPublisher.Text
        buku.Price = txtPrice.Text
        AccessBuku.BukuUpdate(buku)
        ClearTextBox()
    End If

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

Button Delete

```
Try
    Dim mYes_No As String = MsgBox("Do you want delete the data ?",
vbYesNo, "Buku")
    If mYes_No = vbYes Then
        Dim buku As New Buku
        Dim AccessBuku As New AccessData.AccessBuku
        buku.ISBN = txtISBN.Text
        AccessBuku.BukuDelete(buku)
        ClearTextBox()
    End If

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

Coba jalankan program dan lakukan proses update dan delete.

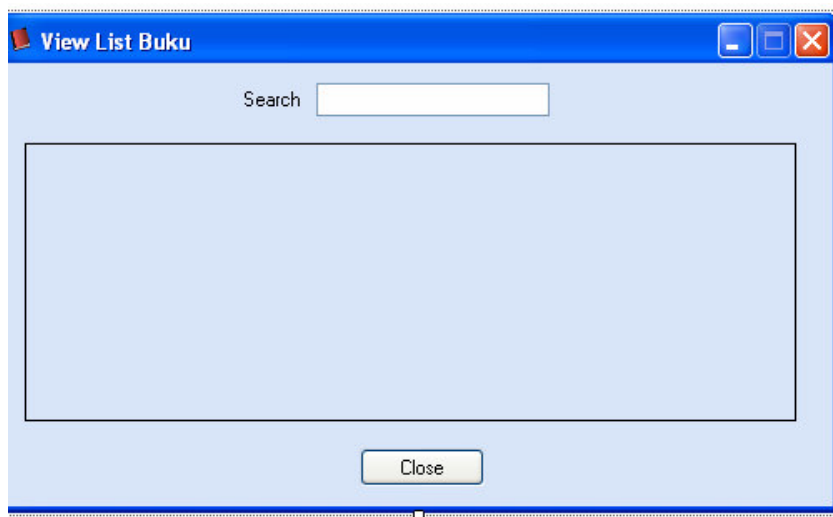
Untuk proses Insert, delete dan update telah selesai kita lakukan sekarang kita lanjutkan untuk menampilkan data buku pada textbox. Untuk proses ini kita menggunakan 2 buah cara, pertama jika ISBN di ketik pada textbox ISBN lalu user meng klik button find, jika ISBN yang di masukkan benar atau ada pada table buku maka akan tampil datanya. Kedua dengan menggunakan event Keyup pada textbox ISBN. Jika user menekan tombol F8 maka form help akan muncul, lalu user memilih ISBN yang dicari dan dengan melakukan double click maka form help akan otomatis tertutup dan data akan di tampilkan pada form buku. Sekarang kita

akan melakukan cara pertama. Double click button Find, dan ketikkan sintaks dibawah ini.

```
Try
    Dim AccessBuku As New AccessData.AccessBuku
    Dim list As List(Of Buku) =
    AccessBuku.FindBukuByISBN(txtISBN.Text.Trim, "Find")
    txtISBN.Text = list.First.ISBN
    txtTitle.Text = list.First.Title
    txtAuthors.Text = list.First.Authors
    txtPublisher.Text = list.First.Publisher
    txtPrice.Text = list.First.Price
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

Fungsi diatas juga kita gunakan pada proses insert. Coba jalan proses pencarian, masukkan ISBN yang telah ada pada table maupun yang tidak ada. Apakah sudah berjalan dengan benar?

Sekarang akan kita coba cara kedua. Tambahkan form baru pada project dengan nama H_Buku.Vb design seperti gambar dibawah.



Ketikkan sintaks dibawah ini pada form H_Buku (dibawah `Public Class H_Buku`)


```
Dim DtDataview As New DataView
Dim objDataTable As New DataTable

Private Sub FormatGridWithBothTableAndColumnStyles()
    Me.DgBuku.DefaultCellStyle.ForeColor = Color.Coral
    Me.DgBuku.RowsDefaultCellStyle.BackColor = Color.AliceBlue
    Me.DgBuku.GridColor = Color.Blue
    Me.DgBuku.BorderStyle = BorderStyle.Fixed3D
    Me.DgBuku.RowHeadersBorderStyle = BorderStyle.Fixed3D

    DgBuku.Columns(0).HeaderText = "ISBN"
    DgBuku.Columns(1).HeaderText = "Title"
    DgBuku.Columns(2).HeaderText = "Author"
    DgBuku.Columns(3).HeaderText = "Publisher"
    DgBuku.Columns(4).HeaderText = "Price"
End Sub
```

Double click form buku dan ketikkan sintaks dibawah ini :

```
Try
    Dim DataControl As New AccessData.DataControl
    Dim myData As DataSet = DataControl.GetDataSet("Select * from
    tbbuku")
    objDataTable = myData.Tables("data")
    DtDataview.Table = objDataTable
    DgBuku.DataSource = DtDataview
    FormatGridWithBothTableAndColumnStyles()

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

Masih ingat dengan Class Data control ? sekarang bayangkan jika kita memiliki banyak form untuk menampilkan data pada DataGrid pasti nya hal ini akan sangat membantu sekali. Kita tinggal memanggil Class DataControl dan mengganti sintaks sql sesuai kebutuhan.

Sekarang pada event textbox cari KeyUp, ketikkan sintaks dibawah ini :

```
Try
    Dim TheFilter As String = "B_ISBN like '" & _
    txtcari.Text.Replace("'", "''").Replace("''", "''''") & "%'"
    DtDataview.RowFilter = TheFilter
    DgBuku.DataSource = DtDataview

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

Pada event keyup ini kita menggunakan DataView, untuk memfilter data. Jadi data pada Datatable sudah kita kirim semua ke DataView.

Lalu buat event baru , yaitu `Private Sub DgBuku_CellDoubleClick`

Ketikkan sintaks dibawah pada event `CellDoubleClick`

```
Dim row As DataGridViewRow = DgBuku.Rows(e.RowIndex)
txtcari.Text = row.Cells(0).Value
Me.Close()
```

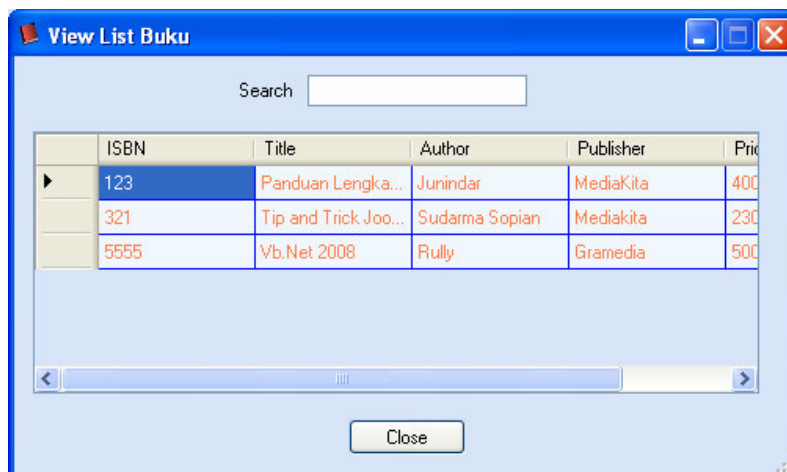
Setelah user melakukan double click maka data pada baris yang di click akan disimpan di textbox cari, selanjutnya form akan tutup.

Selanjutnya aktifkan kembali form Buku, dan buat event Keyup pada textbox ISBN, lalu ketikkan sintaks dibawah ini.

```
If e.KeyCode = Keys.Enter Then
    If txtISBN.Text = "" Then Exit Sub
    btnFind.PerformClick()
ElseIf e.KeyCode = Keys.F8 Then
    Dim H_Buku As New H_Buku
    H_Buku.ShowDialog()
    txtISBN.Text = H_Buku.txtcari.Text
    If txtISBN.Text <> "" Then btnFind.PerformClick()
End If
```

Jika pada textbox user menekan tombol enter maka, program akan menjalankan proses pada button Find (melakukan pencarian). Dan jika user menekan tombol F8 maka form Help buku akan tampil.

Coba jalan kan program dan lakukan proses pencariannya.

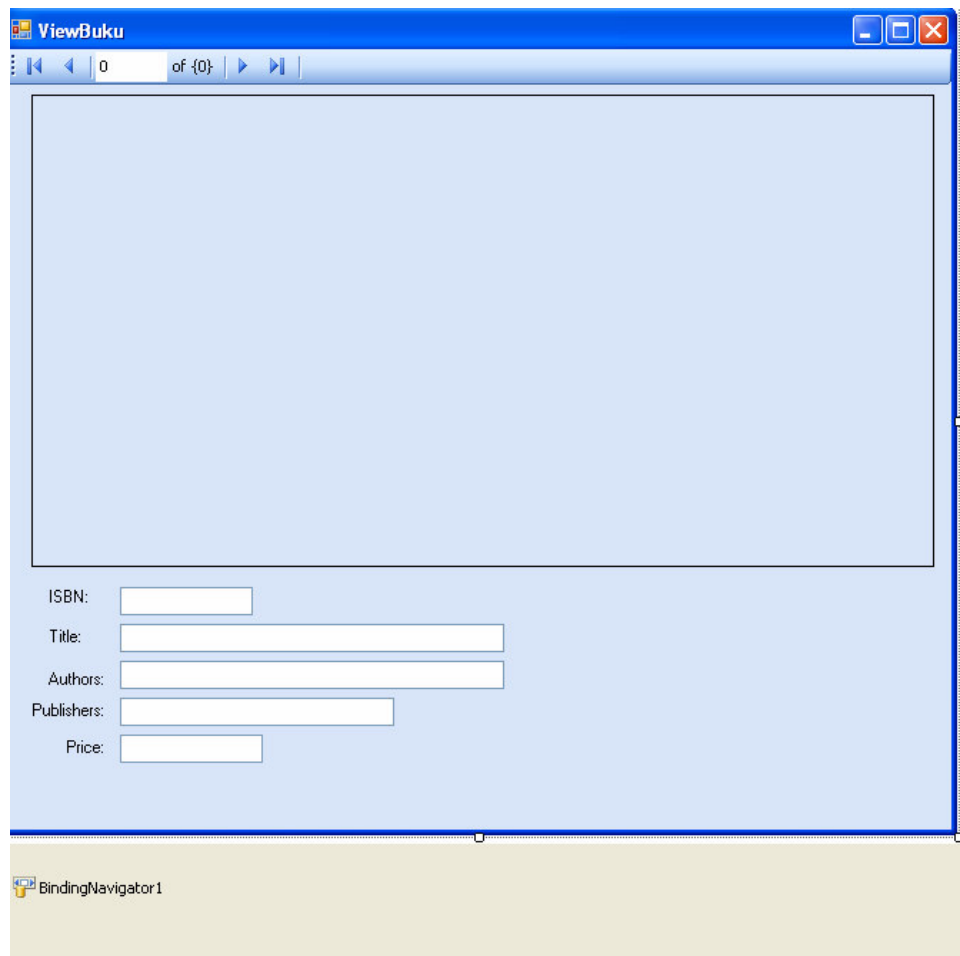


Double click button Get Info pada form Buku, dan ketikkan sintaks dibawah ini :

```
Try
    lstBooks.Items.Clear()
    Dim AccessBuku As New AccessData.AccessBuku
    Dim Buku As New Buku
    Dim list As List(Of Buku) =
AccessBuku.FindBukuByTitle(txtSearch.Text)
    For Each Bukus As Buku In list
        lstBooks.Items.Add(Bukus.Title)
    Next
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

Fungsi ini adalah untuk menampilkan Title buku yang dimasukkan, pada fungsi kita menggunakan fungsi like.

Tambahkan Form baru dengan nama ViewBuku.vb dan buat seperti gambar dibawah.



Berikut sintaks lengkap untuk form diatas

```
Public Class ViewBuku
    Dim DtDataview As New DataView
    Dim objDataTable As New DataTable
    Dim bMgr As BindingManagerBase
    Dim BukuBindSource As New BindingSource

    Private Sub FormatGridWithBothTableAndColumnStyles()
        Me.DgBuku.DefaultCellStyle.ForeColor = Color.Coral
        Me.DgBuku.RowsDefaultCellStyle.BackColor = Color.AliceBlue
        Me.DgBuku.GridColor = Color.Blue
        Me.DgBuku.BorderStyle = BorderStyle.Fixed3D
        Me.DgBuku.RowHeadersBorderStyle = BorderStyle.Fixed3D
        Me.DgBuku.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.AutoSize
        DgBuku.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells
        DgBuku.Columns(0).HeaderText = "ISBN"
        DgBuku.Columns(1).HeaderText = "Title"
        DgBuku.Columns(2).HeaderText = "Author"
        DgBuku.Columns(3).HeaderText = "Publisher"
        DgBuku.Columns(4).HeaderText = "Price"
    End Sub

    Private Sub ViewBuku_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        Try
            Dim DataControl As New AccessData.DataControl
            Dim myData As DataSet = DataControl.GetDataSet("Select * from tbbuku")

            BukuBindSource.DataSource = myData.Tables("data")
            BindingNavigator1.BindingSource = BukuBindSource
            DgBuku.DataSource = BukuBindSource

            txtISBN.DataBindings.Add("text", BukuBindSource, "B_ISBN")
            txtTitle.DataBindings.Add("text", BukuBindSource, "B_Title")
            txtAuthors.DataBindings.Add("text", BukuBindSource, "B_Authors")
            txtPublisher.DataBindings.Add("text", BukuBindSource, "B_Publisher")
            txtPrice.DataBindings.Add("text", BukuBindSource, "B_Price")

            FormatGridWithBothTableAndColumnStyles()

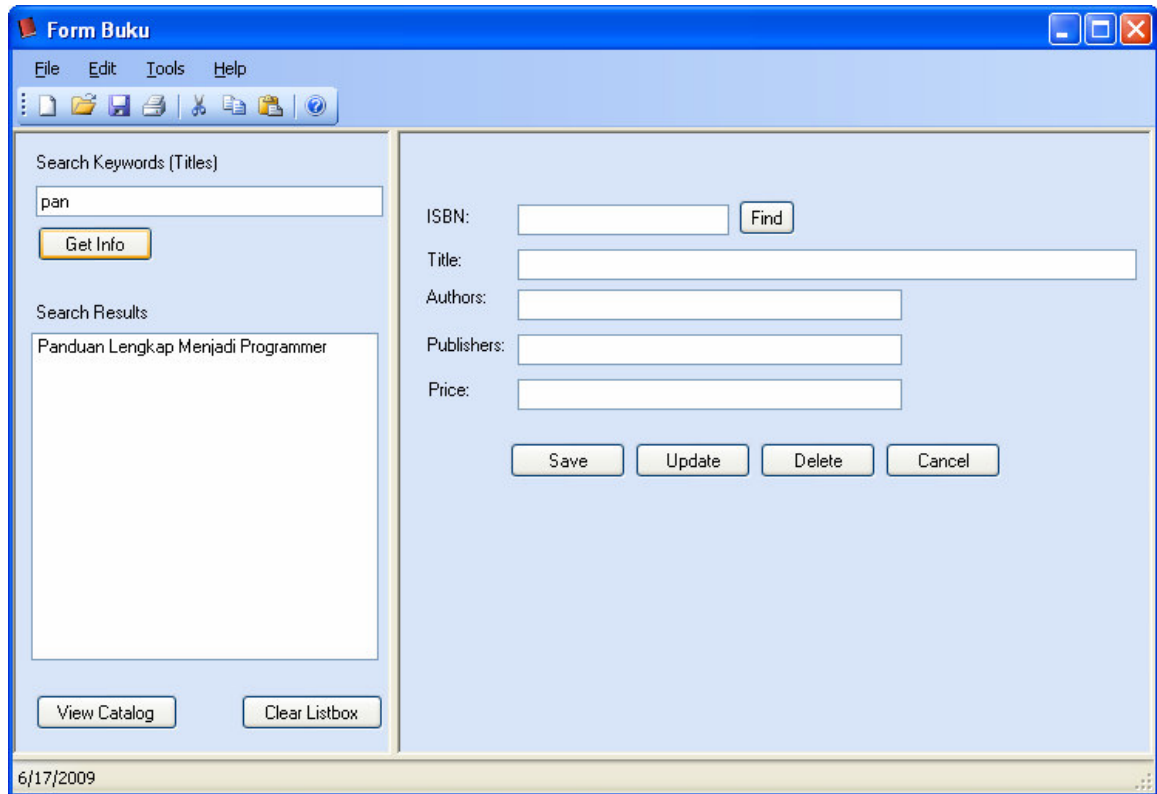
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

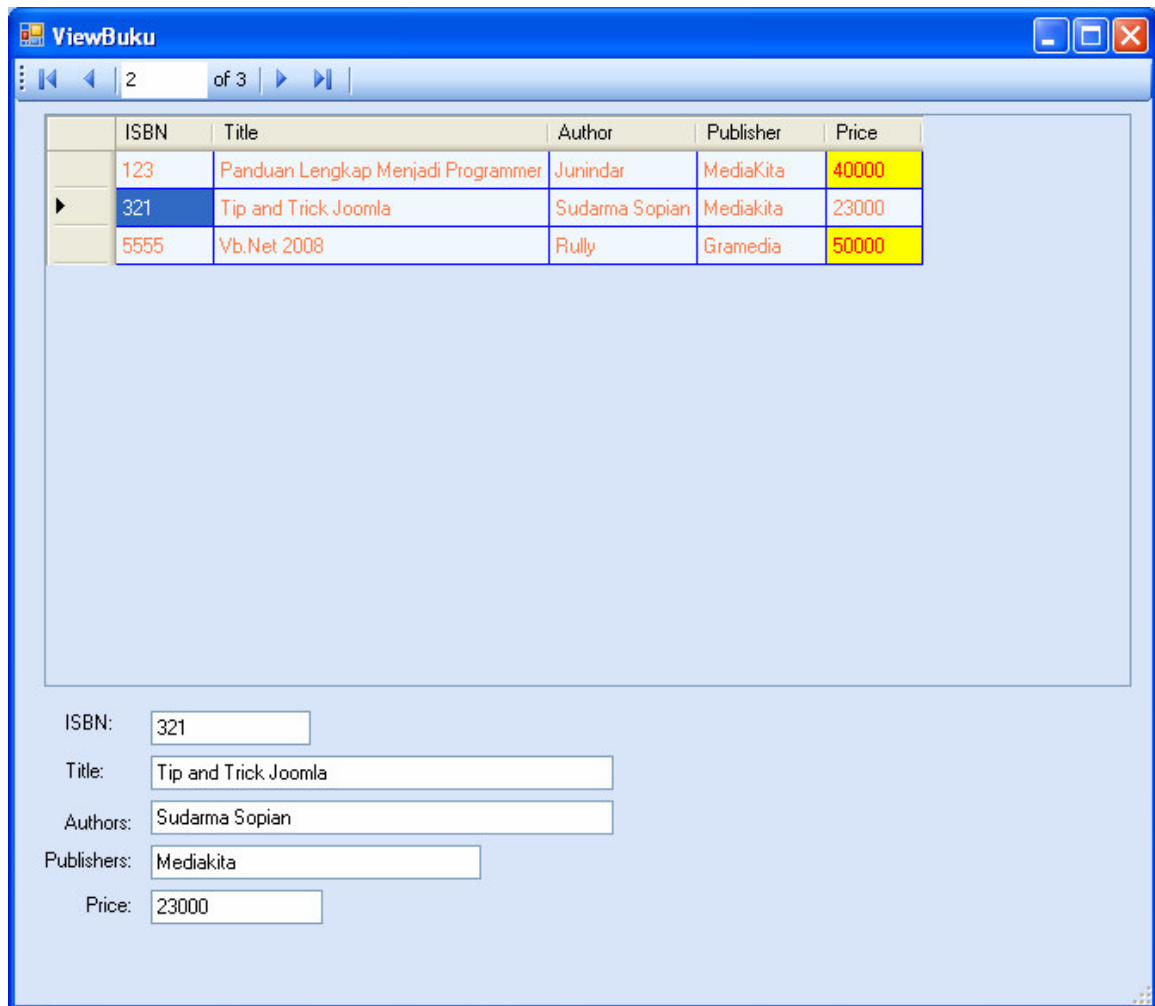
    Private Sub DgBuku_CellFormatting(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellFormattingEventArgs) Handles DgBuku.CellFormatting
        If DgBuku.Columns(e.ColumnIndex).Name = "B_Price" Then
            If Convert.ToInt32(e.Value) > 25000 Then
                e.CellStyle.ForeColor = Color.Red
                e.CellStyle.BackColor = Color.Yellow
            End If
        End If
    End Sub

End Class
```

Sekarang buka kembali form buku dan double click button view catalog, ketikkan sintaks dibawah ini :

```
Dim V_Buku As New ViewBuku  
V_Buku.ShowDialog()
```





The screenshot shows a web application window titled "ViewBuku". At the top, there is a navigation bar with "2 of 3" and navigation arrows. Below this is a table with the following data:

| ISBN | Title | Author | Publisher | Price |
|------|------------------------------------|----------------|-----------|-------|
| 123 | Panduan Lengkap Menjadi Programmer | Junindar | MediaKita | 40000 |
| 321 | Tip and Trick Joomla | Sudarma Sopian | Mediakita | 23000 |
| 5555 | Vb.Net 2008 | Rully | Gramedia | 50000 |

Below the table, there is a form with the following fields:

ISBN:

Title:

Authors:

Publishers:

Price:

Penutup

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

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 mengantar 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.

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). 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 : xx_lead@yahoo.com
junindar@gmail.com*