

# Desktop Component

Studi Kasus: VB.NET



# VB.NET Component

- VB.NET mendukung penciptaan **komponen** yang dibuat oleh programmer
- Dapat menciptakan **komponen baru** ataupun **meningkatkan** kemampuan komponen yang sudah ada
- Bahkan bisa berupa **kumpulan komponen** yang digabungkan
- Dapat digunakan lagi (**reuse**) pada program aplikasi

# VB.NET component

- There are 2 types:
  - Class Library
  - User Control:
    - Custom control
    - Windows Form User control
- Class Library is just a **class** that defines all functionality, data structure, methods, and attributes that can be used in another application
  - Non visual component
  - Must be imported to project

# Component & Programmer

- **User controls** – is an existing controls (visual/non visual). It's have properties.
- **Custom controls** – controls written from scratch
  - It's possible to inherit from existing controls and modify their behavior
- A **composite control** is made up of more than one individual control
- Programmers:
  - The **developer** uses the control
  - The **author** creates, tests, and compiles the control that then appears in the toolbox
  - **Authors** of controls must plan for the **design-time** and **run-time** behavior of controls

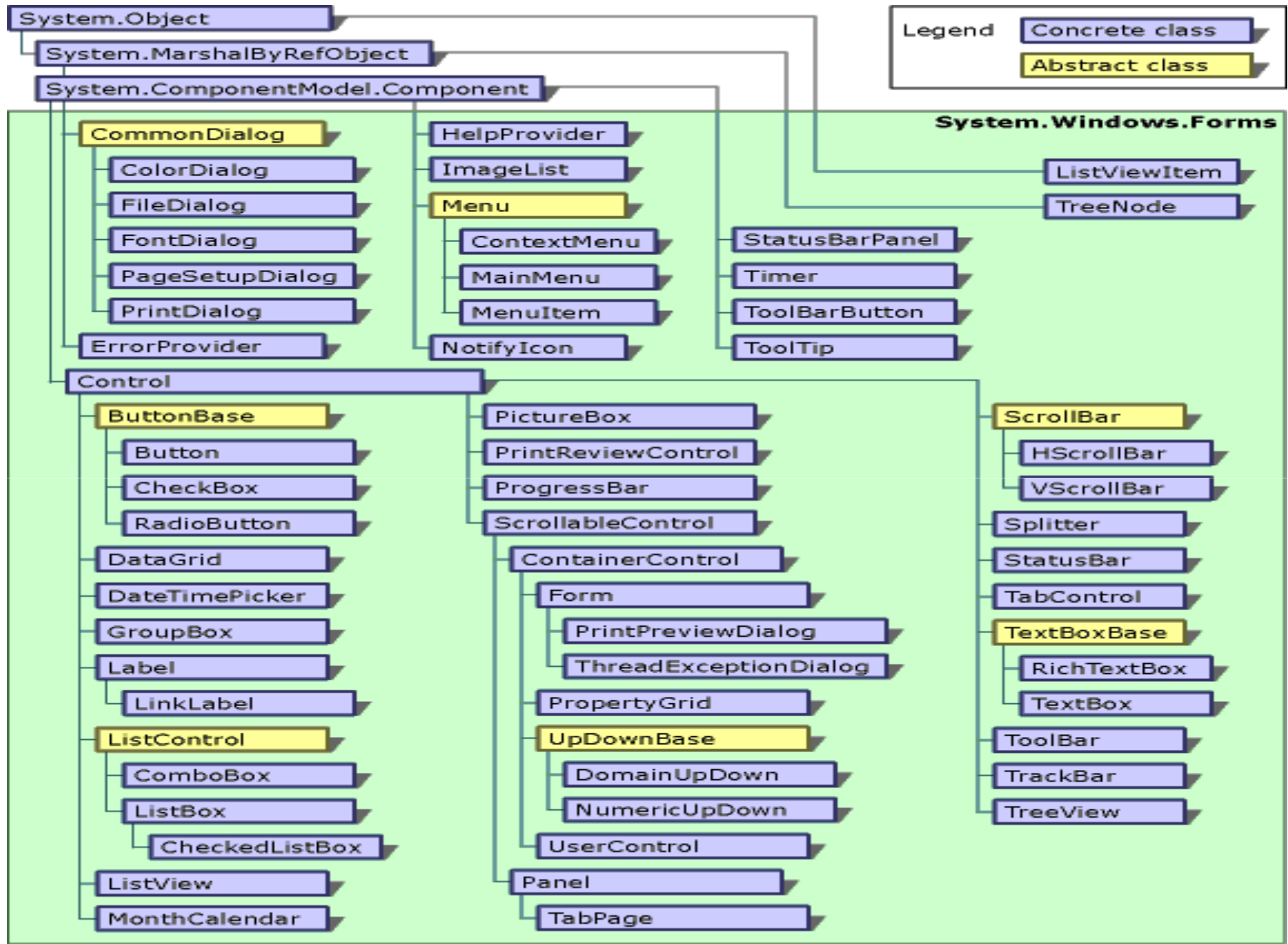
# .NET User Control

- A **user control** is just like the .NET supplied controls, such as **TextBox** or **Button**, but we can make **our own control** do whatever we like with our own code.
  - Even we can create **Web user control**
- A **user control** is a VB.NET **class**.
  - The class **Inherits** from the **UserControl** class
- The **UserControl** class gives your control the **base** functions it needs so it can be treated like the built-in controls

# .NET User control

- The original VB.NET UserControl Class is in **System.Windows.Forms**.
- The new WPF UserControl Class is in **System.Windows.Controls**.
- The ASP.NET UserControl Class is in **System.Web.UI**.



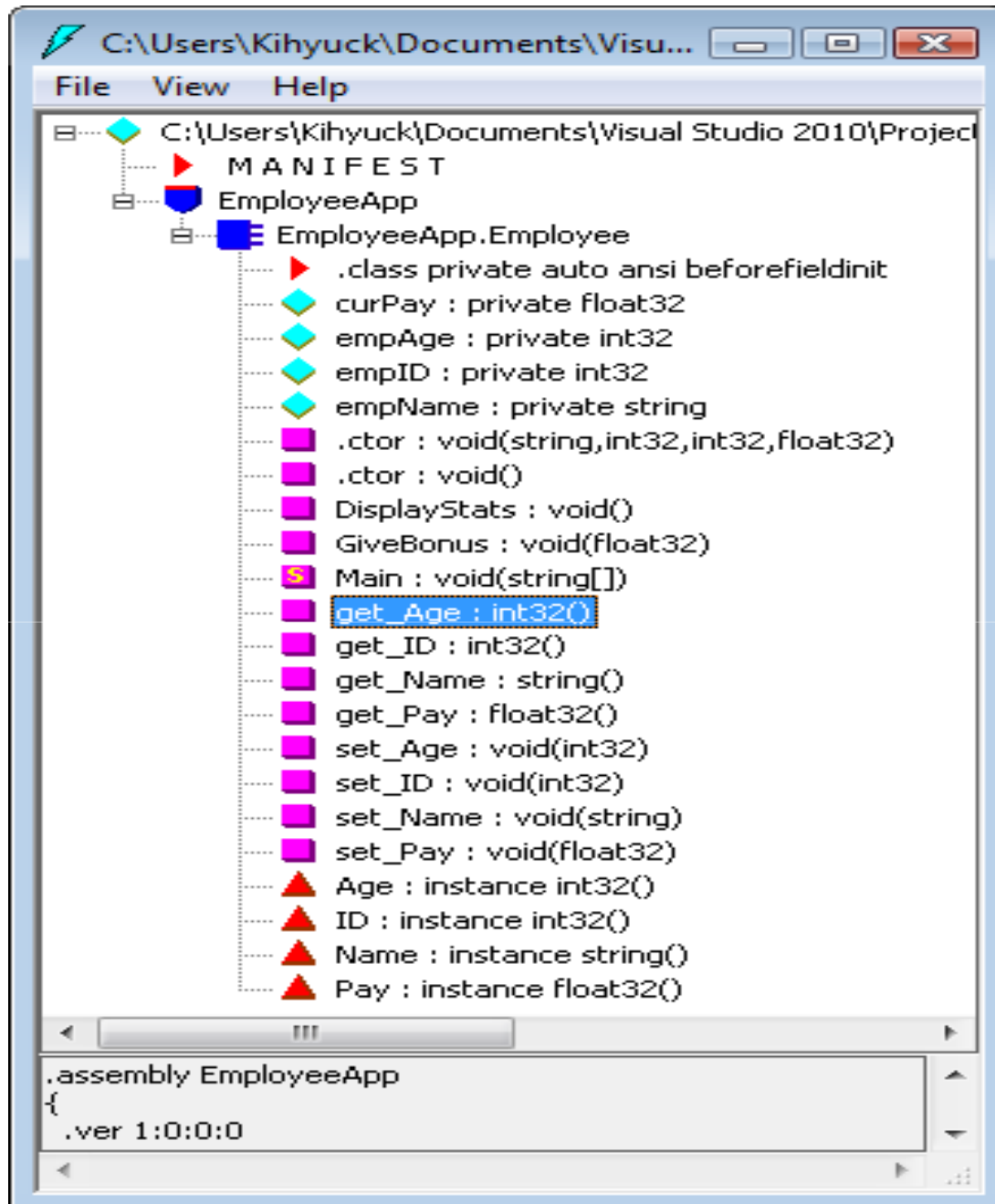


# .NET component's assembly

- A .NET component is *a single precompiled and self-described MSIL module built from **one or more classes or multiple modules** deployed in a DLL assembly file.*
- An assembly consists of up to four parts:
  - **Manifest** (table of info records): name of assembly, key info version, strong name, culture info, files that make up assembly, reference depended assemblies exported info.
  - **Metadata** of modules.
  - **IL code** of modules.
  - **Resources** such as image files.
- The manifest information embedded within an assembly can be viewed using **IL Disassembler** (ILDASM.exe)



# ILDASM in DLL file



# .NET component assembly

- An assembly can consist of **one or more module**
- An assembly is made up by **one or many classes** in a **module**.
  - Each module may be coded in **different languages** but finally in **same MSIL format**.
  - An assembly has a file extension **.DLL or .EXE** and is dynamically loadable.
- There are two kind of assemblies in .NET:
  - **private** : are simple and copied with each calling assemblies in the calling assemblies folder
  - **shared** : are copied to a single location (usually the **Global assembly cache**), not copied into private folder among the application

# Assembly

Version

Types

Name

Shared name

Manifest

Metadata

Computer > WIN7 (C:) > Windows > assembly

Include in library | Share with | Slide show | Burn | New folder

Assembly Name	Version	Cul...	Public Key Token	Proces...
Microsoft.SqlServer.Replication	9.0.242.0		89845dcd8080cc91	x86
Microsoft.SqlServer.Replication.BusinessLogicSupport	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.Rmo	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.ServiceBrokerEnum	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.Setup	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.Smo	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.SmoEnum	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.SqlEnum	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.SqlTdiagM	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.SString	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.WizardFrameworkLite	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.SqlServer.WmiEnum	9.0.242.0		89845dcd8080cc91	MSIL
Microsoft.StdFormat	7.0.330...		b03f5f7f11d50a3a	
Microsoft.Synchronization.Data	1.0.0.0		89845dcd8080cc91	MSIL
Microsoft.Synchronization.Data.Server	1.0.0.0		89845dcd8080cc91	MSIL
Microsoft.Synchronization.Data.SqlServerCe	3.5.0.0		89845dcd8080cc91	MSIL
Microsoft.Tpm	6.1.0.0		31bf3856ad364e35	MSIL
Microsoft.Tpm.Resources	6.1.0.0	en	31bf3856ad364e35	MSIL
Microsoft.Transactions.Bridge	3.0.0.0		b03f5f7f11d50a3a	MSIL

# Global Assembly Cache

- Microsoft provides two ways to interact with the GAC:
  - the Global Assembly Cache Tool (**gacutil.exe**)
  - the Assembly Cache Viewer (**shfusion.dll**)

One can check the availability of a shared assembly in GAC by using the command:

```
gacutil.exe /l <assemblyName>
```

One can register a shared assembly in the GAC by using the command:

```
gacutil.exe /i <assemblyName>
```

Or by copying an assembly file into the following location:

```
%windir%\assembly\
```

Other options for this utility will be briefly described if you use the `/?` flag, i.e.:

```
gacutil.exe /?
```

# gacutil

```
C:\Program Files\Microsoft Visual Studio 9.0\VC>gacutil -l Microsoft.VisualBasic

Microsoft (R) .NET Global Assembly Cache Utility. Version 3.5.21022.8
Copyright (c) Microsoft Corporation. All rights reserved.

The Global Assembly Cache contains the following assemblies:
  Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f
7f11d50a3a, processorArchitecture=MSIL

Number of items = 1
```

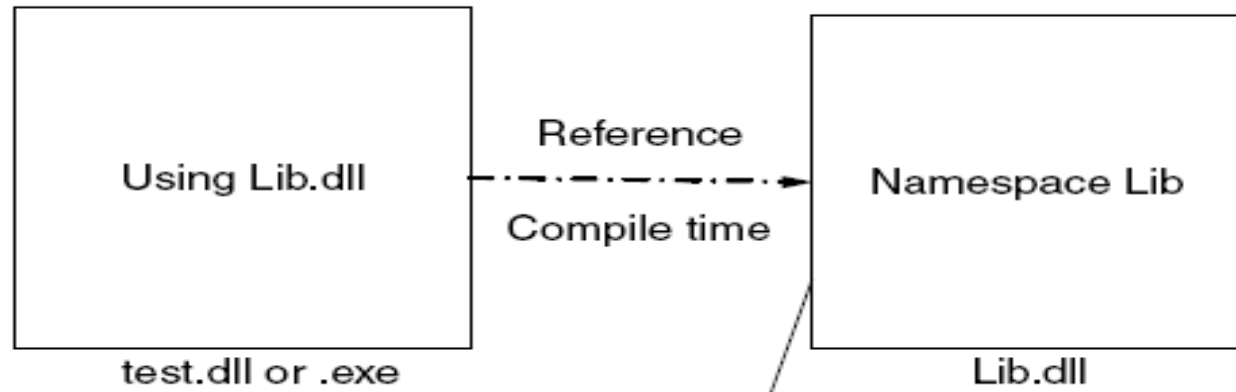
```
C:\Program Files\Microsoft Visual Studio 9.0\VC>gacutil -lr Microsoft.VisualBasic

Microsoft (R) .NET Global Assembly Cache Utility. Version 3.5.21022.8
Copyright (c) Microsoft Corporation. All rights reserved.

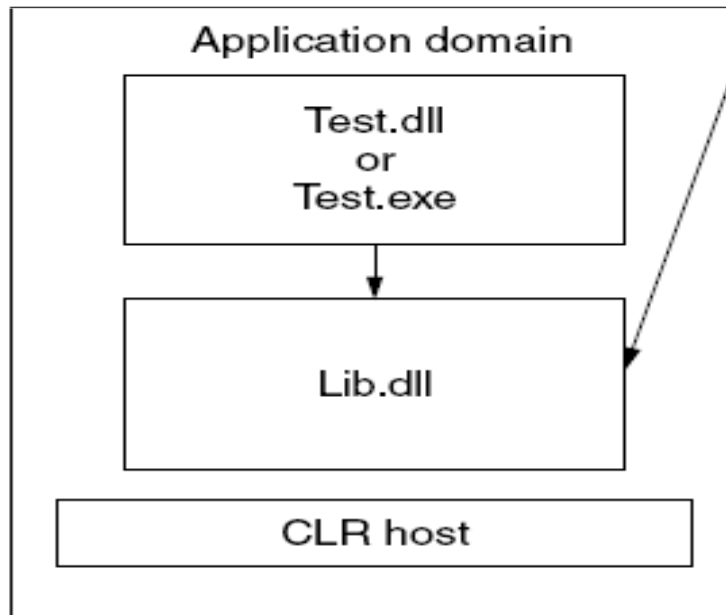
The Global Assembly Cache contains the following assemblies:
  Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f
7f11d50a3a, processorArchitecture=MSIL
  SCHEME: <OPAQUE> ID: <<71F8EFBF-09AF-418D-91F1-52707C DFA274>> DE
SCRIPTION : <.NET Framework Redist Setup>

Number of items = 1
```

# Compiling and runtime



(a) At compile time process



(b) At run time process

# Variabel Class

- Suatu variabel class sebaiknya private
- Pada VB.NET variabel class perlu mutator dan asesor (getter dan setter)
- Pada VB.NET sudah ada getter dan setter otomatis yang disebut PROPERTY
- Property digunakan dalam Properties komponen
- Method digunakan dalam Events komponen

# Property

**Public Property NamaProperty() As Integer**

**Get**

**Return \_variabel**

**End Get**

**Set(ByVal Value As Integer)**

**\_variabel = value**

**End Set**

**End Property**



# Methods

- **Sub NamaMethod(Parameters)**

**End Sub**

- **Function NamaFunction(Parameters) as TipeData**

**return \_variabel**

**end function**

# Build non visual component

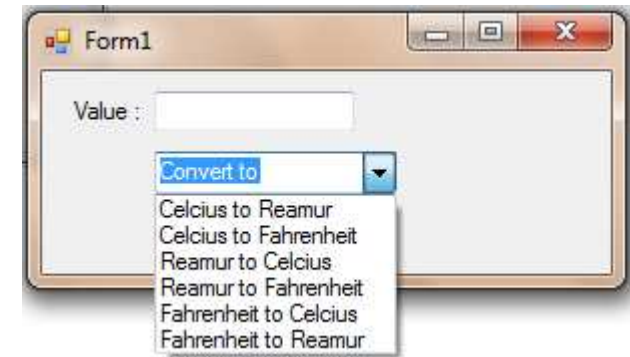
- Use **ClassLibrary** Project
- Name it : TempConv
- Convert Celcius, Reamur, Fahrenheit, and vice versa
- Build -> TempConv.dll

# TempConv class

```
Public Class TempConv
    Public Enum Options
        CtoR
        CtoF
        RtoC
        RtoF
        FtoC
        FtoR
        CtoK
    End Enum
    Public Function Convert(ByVal value As Double, ByVal opt As Options) As Double
        Select Case opt
            Case Options.CtoR
                Return 4 / 5 * value
            Case Options.CtoF
                Return 9 / 5 * value + 32
            Case Options.RtoC
                Return 5 / 4 * value
            Case Options.RtoF
                Return 9 / 4 * value + 32
            Case Options.FtoC
                Return (value - 32) / 1.8
            Case Options.FtoR
                Return (value - 32) / 0.8
            Case Options.CtoK
                Return value + 273
        End Select
    End Function
End Class
```

# Using it

- Make new Windows project
- Add project's references to **TempConv.dll**
- Use it in your application



```
Public Class Form1
    Dim conv As TempConv.TempConv = New TempConv.TempConv
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Select Case ComboBox1.SelectedIndex
        Case 0
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.CtoR))
        Case 1
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.CtoF))
        Case 2
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.RtoC))
        Case 3
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.RtoF))
        Case 4
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.FtoC))
        Case 5
            MsgBox(conv.Convert(Val(TextBox1.Text), TempConv.TempConv.Options.FtoR))
    End Select
    End Sub
End Class
```

# Building Library consumes Web Service

- You can build a class library that consumes a web service (SOAP/REST based)
- You just have to encapsulate all web service function in your class library
- You must have all setting dynamic by using class library properties and methods
- You have to set/get all the properties and make all the methods

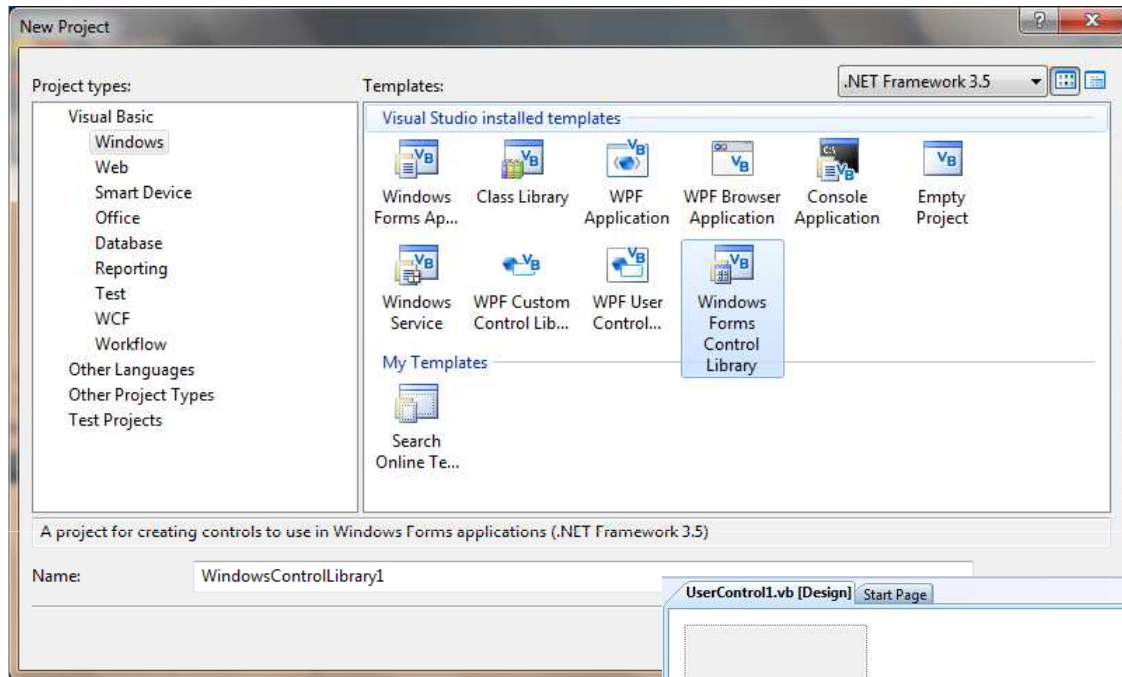
# Yuan's Quote Service

- Menggunakan RESTful Web Service
- Alamat:  
<http://lecturer.ukdw.ac.id/yuan/labs/quotes>
- Mendukung GET, POST, PUT, dan DELETE
- Kembalian data berupa XML string

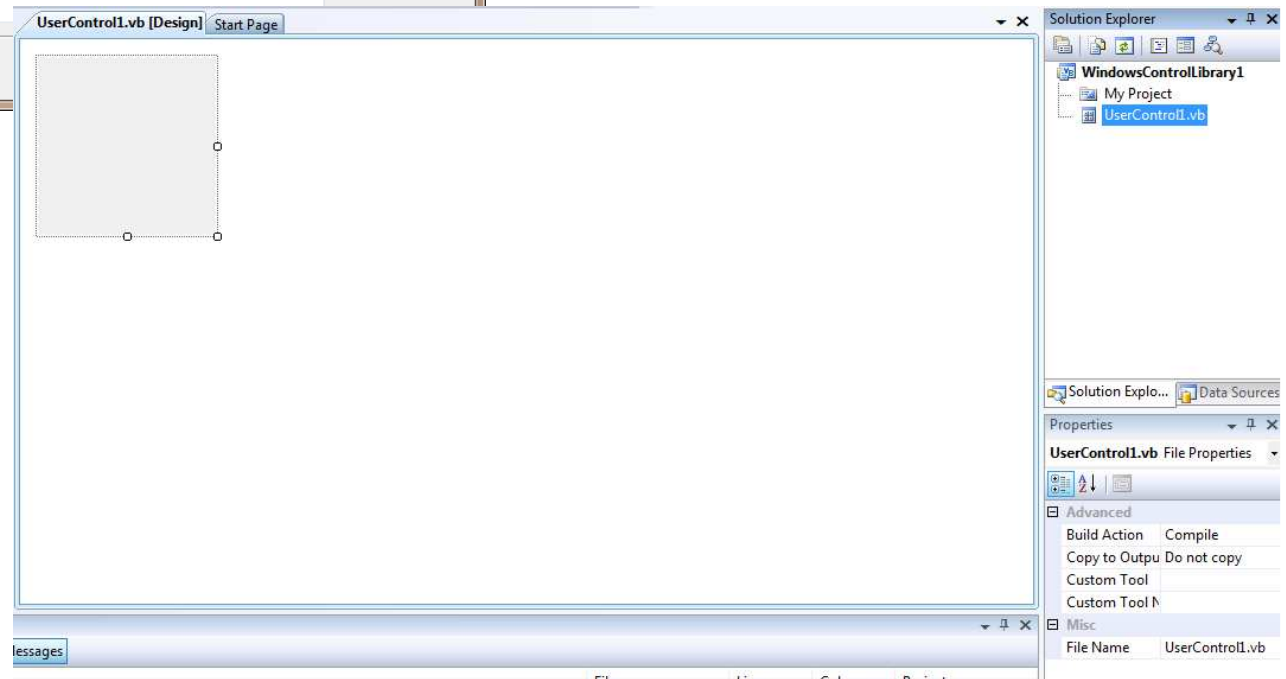
# Build QuoteControl

- Kita akan membuat control untuk menampilkan data Quote dari Web Service Quote dalam bentuk **label** yang menampilkan random quote per selang waktu tertentu
- Create New Projects | Windows Form Control Library, beri nama QuoteControl
- Tambahkan komponen **Label1** dan **Timer1**
- Set Timer1.Enabled = True
- Set Timer1.Interval = 3000

# Create Windows Control Library



Design **UserControl** from Project





# Definisi Property

- `TimeInterval()` as Integer
- `TimerEnabled()` as Boolean
- `StrQuote()` as String (readonly)
- `UrlRest()` as String

# Definisi Fungsi

- GetRandomQuote() -> GET
- FindQuote(cari) -> GET
- GetAllQuote() -> GET
- AddQuote(author,quote) -> POST
- DellQuote(id) -> DELETE
- EditQuote(id,author,quote) -> PUT
  
- Buat juga class Quote
  - Atribut Id
  - Atribut Content
  - Atribut Author

# QuoteControl.vb

```
<ToolboxBitmap("D:\quote.png")>
```

```
Public Class QuoteControl  
    Private pri_urlrest As String  
    Private str_quote As String
```

```
    ReadOnly Property StrQuote As String  
        Get  
            Return str_quote  
        End Get  
    End Property
```

```
    Property UrlRest As String  
        Get  
            Return pri_urlrest  
        End Get  
        Set(value As String)  
            pri_urlrest = value  
        End Set  
    End Property
```

```
Public Function GetAllQuote() As List(Of Quote)  
    Dim proxy As WebClient = New WebClient()  
  
    Dim hasilxml As String = proxy.DownloadString((New Uri(Me.UrlRest & "/all")))  
    Dim xmllinq = XDocument.Load(New StringReader(hasilxml))  
  
    Dim myquote = From elem In xmllinq.Root.Elements("quote")  
                  Select elem  
  
    Dim hslquote As List(Of Quote) = New List(Of Quote)  
  
    For Each x In myquote  
        Dim satuquote As Quote = New Quote()  
        satuquote.Id = x.Elements("id").Value  
        satuquote.Author = x.Elements("author").Value  
        satuquote.Content = x.Elements("content").Value  
        hslquote.Add(satuquote)  
    Next  
  
    Return hslquote  
End Function
```

```

Function FindQuote(ByVal cari As String) As List(Of Quote)
    Dim proxy As WebClient = New WebClient()

    Dim hasilxml As String = proxy.DownloadString((New Uri(Me.UrlRest & "/find/" & cari)))
    Dim xmllinq = XDocument.Load(New StringReader(hasilxml))

    Dim myquote = From elem In xmllinq.Root.Elements("quote")
                  Select elem

    Dim hslquote As List(Of Quote) = New List(Of Quote)

    For Each x In myquote
        Dim satuquote As Quote = New Quote()
        satuquote.Id = x.Elements("id").Value
        satuquote.Author = x.Elements("author").Value
        satuquote.Content = x.Elements("content").Value
        hslquote.Add(satuquote)
    Next

    Return hslquote
End Function

```

```

Function GetRandomQuote() As Quote
    Dim proxy As WebClient = New WebClient()

    Dim hasilxml As String = proxy.DownloadString((New Uri(Me.UrlRest & "/random")))
    Dim xmllinq = XDocument.Load(New StringReader(hasilxml))

    Dim myquote = From elem In xmllinq.Root.Elements("quote")
                  Select elem

    Dim satuquote As Quote = New Quote()

    satuquote.Id = myquote.Elements("id").Value
    satuquote.Author = myquote.Elements("author").Value
    satuquote.Content = myquote.Elements("content").Value

    Return satuquote
End Function

```

# AddQuote

```
Public Function AddQuote(ByVal author As String, ByVal strquote As String) As String
    Dim request As HttpWebRequest = HttpWebRequest.Create(Me.UrlRest & "/")
    request.Method = "POST"
    request.KeepAlive = False
    Dim postData As String = "author=" & author & "&quote=" & strquote
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postData)
    request.ContentType = "application/x-www-form-urlencoded"
    request.ContentLength = byteArray.Length
    Dim dataStream As Stream = request.GetRequestStream()
    dataStream.Write(byteArray, 0, byteArray.Length)
    dataStream.Close()
    Dim response As WebResponse = request.GetResponse()
    'Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
    dataStream = response.GetResponseStream()
    Dim reader As New StreamReader(dataStream)
    Dim responseFromServer As String = reader.ReadToEnd()
    'Console.WriteLine(responseFromServer)
    reader.Close()
    dataStream.Close()
    response.Close()
    Return responseFromServer
End Function
```

# DeleteQuote

```
Public Function DelQuote(ByVal id As Integer) As Boolean
    Dim request As HttpWebRequest = HttpWebRequest.Create(Me.UrlRest & "/" & id.ToString)
    request.Method = "DELETE"
    request.KeepAlive = False
    Dim response As HttpWebResponse = request.GetResponse()
    If response.StatusCode() = 200 Then
        Return True
    Else
        Return False
    End If
    response.Close()
End Function
```

```
Public Function EditQuote(ByVal id As Integer, ByVal author As String, ByVal strquote As String) As String
    Dim request As HttpWebRequest = HttpWebRequest.Create(Me.UrlRest & "/")
    request.Method = "PUT"
    request.KeepAlive = False
    Dim putData As String = "id=" & id.ToString & "&author=" & author & "&quote=" & strquote
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(putData)
    request.ContentType = "application/x-www-form-urlencoded"
```

# Class Quote

```
Public Class Quote
    Private pri_id As Integer
    Private pri_quote As String
    Private pri_author As String
Property Id As Integer
    Get
        Return pri_id
    End Get
    Set(value As Integer)
        pri_id = Id
    End Set
End Property

Property Content As String
    Get
        Return pri_quote
    End Get
    Set(value As String)
        pri_quote = value
    End Set
End Property
```

```
Property Author As String
    Get
        Return pri_author
    End Get
    Set(value As String)
        pri_author = value
    End Set
End Property
End Class
```

# Tambahkan Icon pada Komponen

- Pilih project Properties
- Bagian Resources > tambahkan Image Resources dari image yang sudah disiapkan (ukuran 16x16pixel, berformat bmp atau png)
- Load pada Awal Class QuoteControl.vb:  
<ToolboxBitmap("D:\quote.png")> atau  
<ToolboxBitmap(GetType(QuoteControl),"quote.png")>



# QuoteControlDesigner.vb

- Bagian Label dibuat menampilkan random quote yang berubah tiap satuan waktu

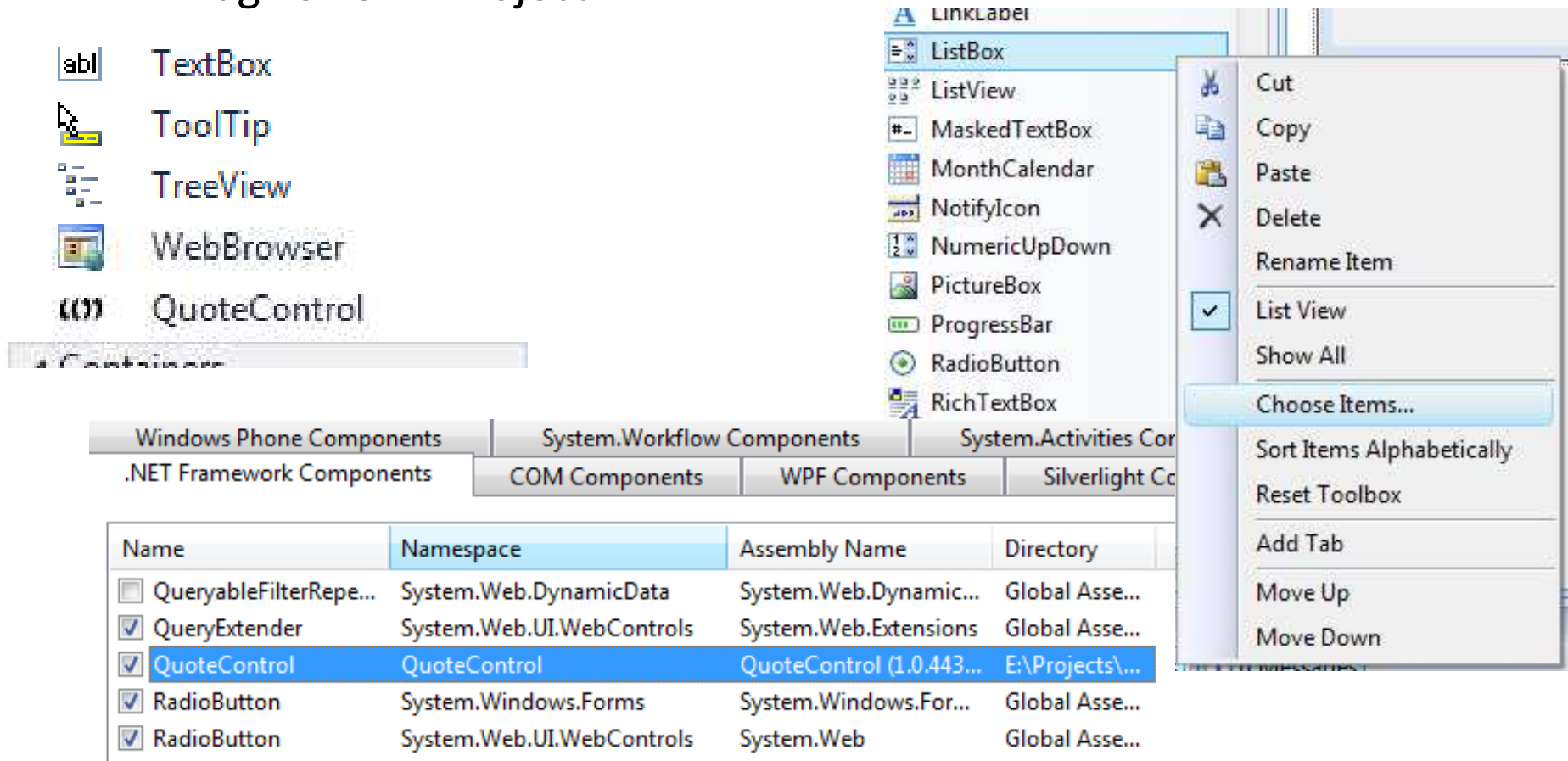
```
Me.Label1.AutoSize = True
Me.Label1.Location = New System.Drawing.Point(4, 14)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(0, 13)
Me.Label1.TabIndex = 0
UrlRest = "http://lecturer.ukdw.ac.id/yuan/labs/quotes"
Dim hasil As Quote = GetRandomQuote()
str_quote = hasil.Content
Me.Label1.Text = str_quote & " (" & hasil.Author & ")"
```

```
Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick
    Dim hasil As Quote = GetRandomQuote()
    str_quote = hasil.Content
    Me.Label1.Text = str_quote & " (" & hasil.Author & ")"
```

End Sub

# Gunakan pada Project nyata

- Klik kanan Toolbox, Choose Items
  - Pilih file DLL dari LoginControl yang sudah dibuat
- Drag ke Form Project



The screenshot shows the Visual Studio interface with the Toolbox on the left and the 'Choose Items' dialog on the right. The 'QuoteControl' item is selected in the Toolbox. The 'Choose Items' dialog is open, showing a list of controls and their properties. The 'QuoteControl' item is highlighted in the list.

Name	Namespace	Assembly Name	Directory
<input type="checkbox"/> QueryableFilterRepe...	System.Web.DynamicData	System.Web.Dynamic...	Global Asse...
<input checked="" type="checkbox"/> QueryExtender	System.Web.UI.WebControls	System.Web.Extensions	Global Asse...
<input checked="" type="checkbox"/> QuoteControl	QuoteControl	QuoteControl (1.0.443...	E:\Projects\...
<input checked="" type="checkbox"/> RadioButton	System.Windows.Forms	System.Windows.For...	Global Asse...
<input checked="" type="checkbox"/> RadioButton	System.Web.UI.WebControls	System.Web	Global Asse...

# QuoteClient

A screenshot of a Windows application window titled "Form1". The window displays a quote: "A little knowledge that acts is worth infinitely more than much knowledge that is idle (Khalil Gibran)". Below the quote is a "Get All" button. Underneath is a text input field and a "Get Quote" button. A list of quotes is shown in a scrollable area:

1. Stay foolish, stay hungry (Steve Jobs)
2. If you can dream it, you can do it (Walt Disney)
3. Friendship is always a sweet responsibility, never an opportunity (Khalil Gibran)
4. A little knowledge that acts is worth infinitely more than much knowledge that is idle (Khalil Gibran)
5. Age is an issue of mind over matter. If you don't mind, it doesn't matter (Mark Twain)
6. Darkness cannot drive out darkness; only light can do that. Hate cannot drive out hate; only love can do that (Martin Luther King, Jr.)
7. yayaya (yuanlukito)
8. algoritma (yuan)
9. algoritma pemrograman (antonius rc)

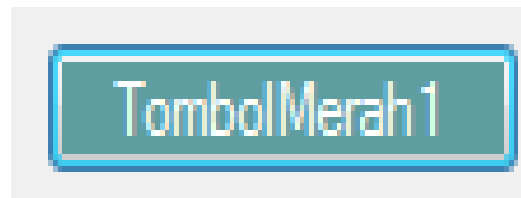
At the bottom, there are three input fields labeled "id", "author here", and "quote here", followed by "Delete" and "Edit" buttons.

A screenshot of a modal dialog box. It displays the same quote as the main window: "A little knowledge that acts is worth infinitely more than much knowledge that is idle (Khalil Gibran)". Below the quote is a "Get All" button. Underneath is a text input field containing the word "knowledge" and a "Get Quote" button. Below the input field is a scrollable list containing the quote: "4. A little knowledge that acts is worth infinitely more than much knowledge that is idle (Khalil Gibran)".

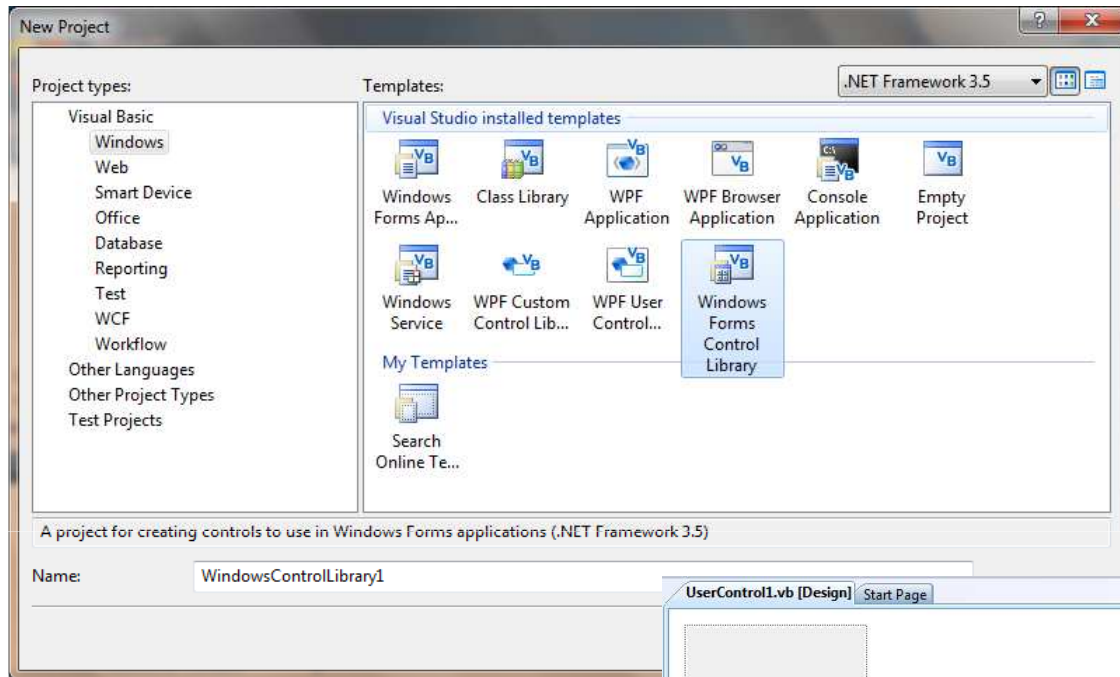
A screenshot of a form with three input fields: "id", "author here", and "quote here". Below the "id" and "author here" fields are "Delete" and "Edit" buttons. To the right of the "quote here" field is a "Post" button.

# Overrides ButtonControl

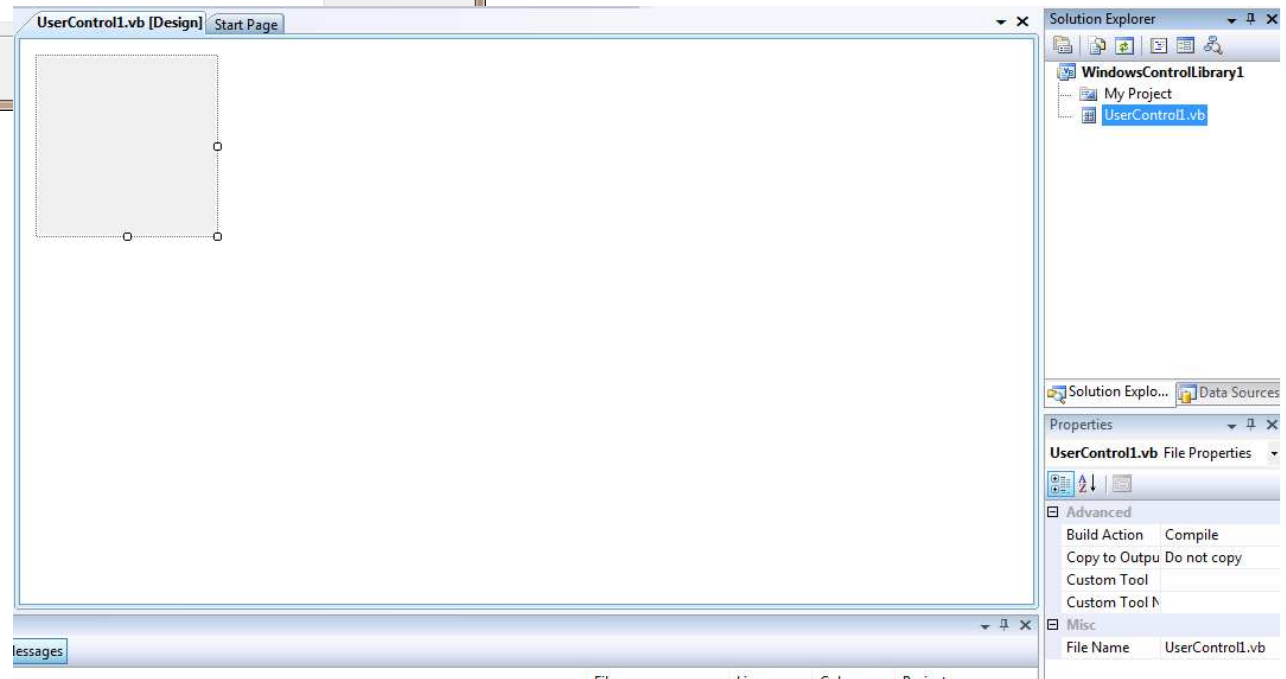
- Membuat button baru:
  - dengan teks berwarna **Merah Crimson**
  - Jika diclick tombol berubah warna
    - Background jadi **cadet blue**, font menjadi **white**
  - Jika diclick sekali lagi maka akan kembali seperti sedia kala
    - Color **control**



# Create Windows Control Library



Design **UserControl** from Project



# Langkah

- Membuat Windows User Control Library seperti biasa
  - Buka file TombolMerah.Designer.vb
- Ubah bagian class mejadi inherits TextBox

```
Partial Class TombolMerah
    Inherits System.Windows.Forms.Button
```

```
'Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
```

- Tambahkan method New dan Button

```
Public Sub New()
    ' This call is required by the Windows Form Designer.
    InitializeComponent()
    ' Add any initialization after the InitializeComponent() call.
    Me.ForeColor = Color.Crimson
End Sub
```

# Langkah selanjutnya

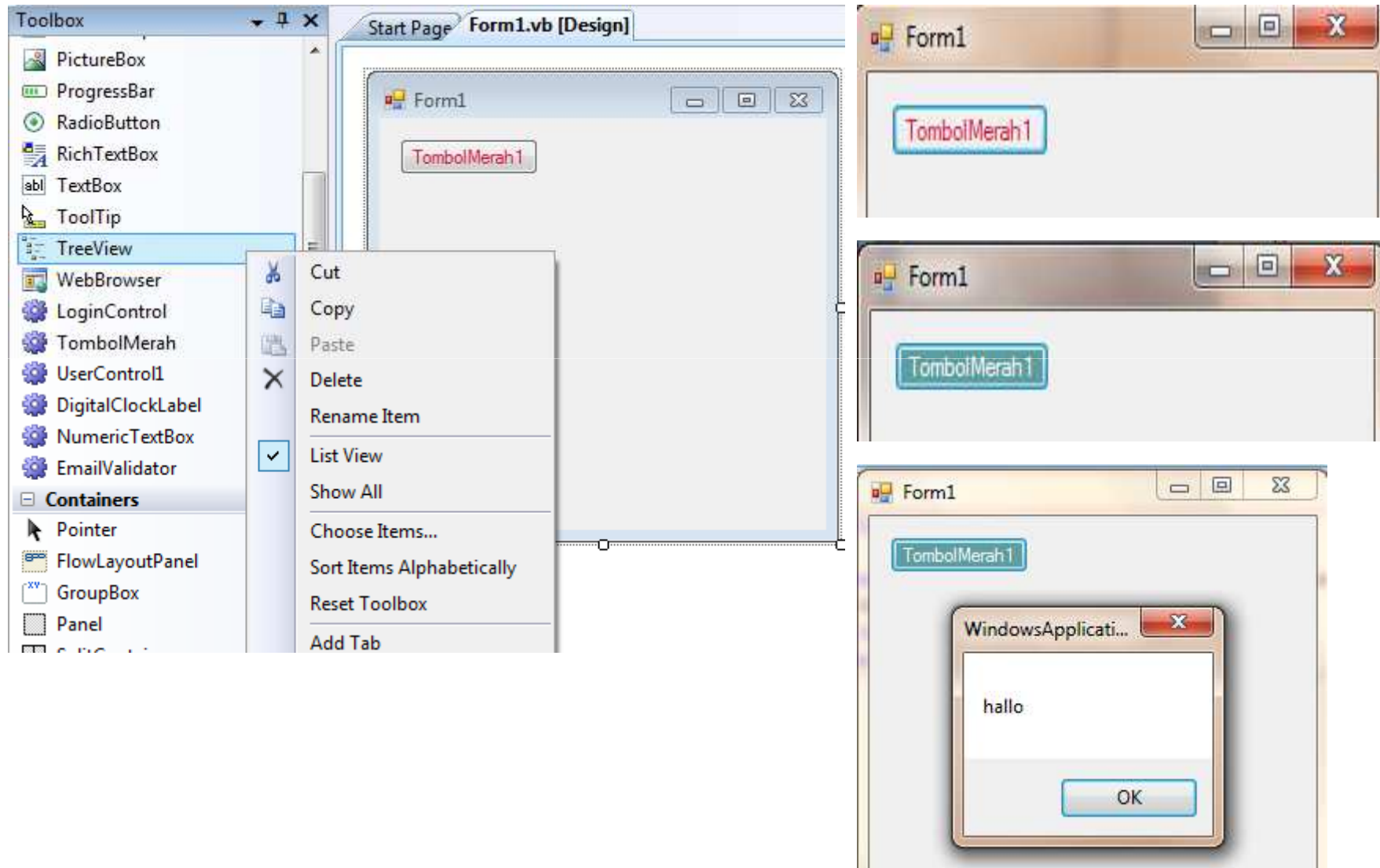
- Tambahkan Method Click pada button

```
private Sub TombolMerah_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
    If Me.ForeColor = Color.Crimson Then
        Me.ForeColor = Color.White
        Me.BackColor = Color.CadetBlue
    Else
        Me.ForeColor = Color.Crimson
        Me.BackColor = Color.FromKnownColor(KnownColor.Control)
    End If
End Sub
```

- Show All Files & Build!
- Gunakan pada aplikasi Anda!

To add components to your class, drag them from the [Toolbox](#) and use the Properties window to set their properties. To create methods and events for your class, [click here to switch to code view](#).

# Penggunaan pada Aplikasi





# Tombol Merah Web

- Buat Project > Web > ASP.NET Server Control

```
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Text
Imports System.Web
Imports System.Web.UI
Imports System.Web.UI.WebControls

Namespace Tombol
    Public Class TombolMerah
        Inherits Button

        Private Sub TombolMerah_Click(ByVal sender As Object, ByVal e As System.EventArgs)
            If Me.ForeColor = Drawing.Color.Crimson Then
                Me.ForeColor = Drawing.Color.White
                Me.BackColor = Drawing.Color.CadetBlue
            Else
                Me.ForeColor = Drawing.Color.Crimson
                Me.BackColor = Drawing.Color.FromKnownColor(Drawing.KnownColor.Control)
            End If
        End Sub

        Protected Overrides Sub OnInit(ByVal e As System.EventArgs)
            Me.ForeColor = Drawing.Color.Crimson
        End Sub

    End Class
End Namespace
```

# Hasil aplikasi web

```
Partial Class _Default  
    Inherits System.Web.UI.Page
```

---

```
    Protected Sub TombolMerah1_Click(ByVal sender As Object, ByVal e As System  
        If IsPostBack Then  
            If TombolMerah1.ForeColor <> Drawing.Color.Crimson Then  
                atas.InnerText &= " Anda sudah menurut!"  
                TombolMerah1.Text = "Klik balik!"  
            Else  
                atas.InnerText = "Hallo, silahkan tekan tombol dibawah!"  
                TombolMerah1.Text = "Tekan saya"  
            End If  
        End If  
    End Sub  
End Class
```

---

Hallo, silahkan tekan tombol dibawah!

Tekan saya

Hallo, silahkan tekan tombol dibawah! Anda sudah menurut!

Klik balik!

# CuteButton

- Buat Project Baru > Windows Form Control Library
- Beri nama: CuteButton
- Set inherits dari Button
- Buat 2 property warna:
  - cuteColor1
  - cuteColor2
- Buat 2 warna transparan:
  - cuteTransparant1
  - cuteTransparant2

# Kode

```
Private m_color1 As Color = Color.LightGreen
Private m_color2 As Color = Color.DarkBlue
private m_color1Transparent As Integer = 64
private m_color2Transparent As Integer = 64
```

```
Public Property cuteColor1 As Color
    Get
        Return m_color1
    End Get
    Set(ByVal value As Color)
        m_color1 = value
        Invalidate()
    End Set
End Property
```

```
Public Property cuteColor2 As Color
    Get
        Return m_color2
    End Get
    Set(ByVal value As Color)
        m_color2 = value
        Invalidate()
    End Set
End Property
```

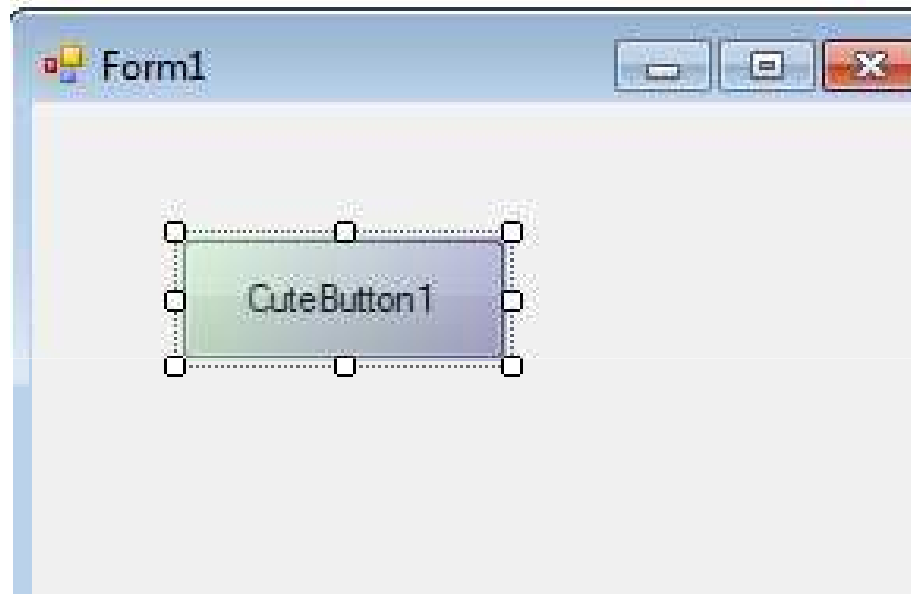
# Kode



```
Public Property cuteTransparant1 As Integer
    Get
        Return m_color1Transparent
    End Get
    Set(ByVal value As Integer)
        m_color1Transparent = value
        Invalidate()
    End Set
End Property
```

```
Public Property cuteTransparant2 As Integer
    Get
        Return m_color2Transparent
    End Get
    Set(ByVal value As Integer)
        m_color2Transparent = value
        Invalidate()
    End Set
End Property
```

```
Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
    MyBase.OnPaint(e)
    Dim c1 As Color = Color.FromArgb(m_color1Transparent, m_color1)
    Dim c2 As Color = Color.FromArgb(m_color2Transparent, m_color2)
    Dim b As Brush = New System.Drawing.Drawing2D.LinearGradientBrush(ClientRectangle, c1, c2, 10)
    e.Graphics.FillRectangle(b, ClientRectangle)
    b.Dispose()
    'Add your custom paint code here
End Sub
```

# Hasil



Properties (1/1)	
Cursor	Default
cuteColor1	 LightGreen
cuteColor2	 DarkBlue
cuteTranspar:	64
cuteTranspar:	64
DialogResult	None

# NumericTextBox

- Textbox yang hanya menerima angka, karakter titik (sebagai koma), dan karakter control (seperti backspace/tab)



- TextAlign = **HorizontalAlignment.Right**
- Ubah Windows User Control agar inherits **TextBox**

```
Partial Class NumericTextBox  
    Inherits System.Windows.Forms.TextBox
```

```
Private _AllowNeg As Boolean = True
Private _AllowDecimal As Boolean = True
Private _MaxValue As Double
Private _MinValue As Double
Private _intnilai As Integer = 0
Private _dnilai As Double = 0
```

```
Property AllowNeg() As Boolean
    Get
        Return _AllowNeg
    End Get
    Set(ByVal value As Boolean)
        _AllowNeg = value
    End Set
End Property
```

```
Property AllowDecimal() As Boolean
    Get
        Return _AllowDecimal
    End Get
    Set(ByVal value As Boolean)
        _AllowDecimal = value
    End Set
End Property
```

```
Property MaxValue() As Double
    Get
        Return _MaxValue
    End Get
    Set(ByVal value As Double)
        _MaxValue = value
        If _intnilai > _MaxValue Then
            _intnilai = _MaxValue
            Me.Text = Convert.ToString(_intnilai)
        End If
        If DNilai > _MaxValue Then
            DNilai = _MaxValue
            Me.Text = Convert.ToString(DNilai)
        End If
    End Set
End Property
```

```
Property MinValue() As Double
    Get
        Return _MinValue
    End Get
    Set(ByVal value As Double)
        _MinValue = value
        If _intnilai < _MinValue Then
            _intnilai = _MinValue
            Me.Text = Convert.ToString(_intnilai)
        End If
        If DNilai < _MinValue Then
            DNilai = _MinValue
            Me.Text = Convert.ToString(DNilai)
        End If
    End Set
End Property
```



```

Property INilai() As Integer
    Get
        Try
            Return Convert.ToInt16(Me.Text)
        Catch ex As Exception
            Return 0
        End Try
    End Get
    Set(ByVal value As Integer)
        Try
            If Not AllowDecimal Then
                Me.Text = Convert.ToString(value)
                _intnilai = value
            End If
        Catch ex As Exception
            Me.Text = "0"
            _intnilai = 0
        End Try
    End Set
End Property

```

```

Property DNilai() As Double
    Get
        Try
            Return Convert.ToDouble(Me.Text)
        Catch ex As Exception
            Return 0
        End Try
    End Get
    Set(ByVal value As Double)
        Try
            If Not AllowDecimal Then
                Me.Text = Convert.ToString(value)
            Else
                Me.Text = Convert.ToString(Math.Round(value))
            End If
            _dnilai = value
        Catch ex As Exception
            Me.Text = "0"
            dnilai = 0
        End Try
    End Set
End Property

```

```

Private Sub InitializeComponent()
    Text = "0"
    components = New System.ComponentModel.Container()
    'Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
End Sub

```

```

Private Function DecimalSeparator() As String
    DecimalSeparator = Mid$(1 / 2, 2, 1)
End Function

```

```

Overrides Dispose To Clean Up The Component List.
ics.DebuggerNonUserCode() > _
des Sub Dispose(ByVal disposing As Boolean)
sing AndAlso components IsNot Nothing Then
onents.Dispose()

```

```

Finally
    MyBase.Dispose(disposing)
End Try
End Sub

```

```

Protected Overrides Sub OnKeyPress(ByVal e As KeyPressEventArgs)
    If Char.IsNumber(e.KeyChar) = True Then
        |
    ElseIf e.KeyChar = "." Or e.KeyChar = "," Then
        If AllowDecimal = True And InStr(Me.Text, DecimalSeparator) = 0 Then
            e.KeyChar = DecimalSeparator()
        Else
            e.KeyChar = Chr(0)
        End If
    ElseIf Convert.ToDouble(Text) > MaxValue Then
        Text = Convert.ToString(MaxValue)
    ElseIf Convert.ToDouble(Text) < MinValue Then

```

```

        Text = Convert.To
    End If
Else
    End Sub

```

```

    Select Case e.Key
        Case Chr(45)
            Dim p As
            If Allow
                p =
                DNilai
            If D
                1
            Else
                1
            End
            e.Key
        Else
            e.Key
        End If
    Case vbBack
        Case Else
            e.KeyChar
    End Select

```

```

Protected Overloads Sub OnLeave(ByVal sender As System.Object, ByVal e As System.EventArgs)
    If _MaxValue < DNilai And _MaxValue > _MinValue Then
        DNilai = _MaxValue
        Text = Convert.ToString(_MaxValue)
    End If
    If _MinValue > DNilai And _MinValue < _MaxValue Then
        DNilai = _MinValue
        Text = Convert.ToString(_MinValue)
    End If
End Sub

```

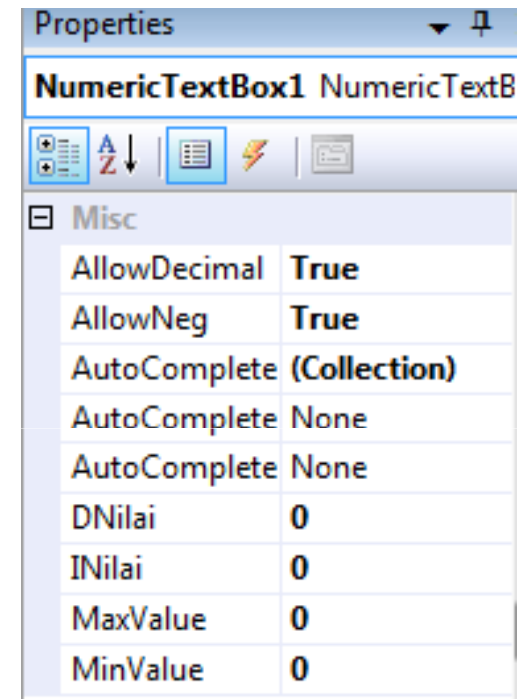
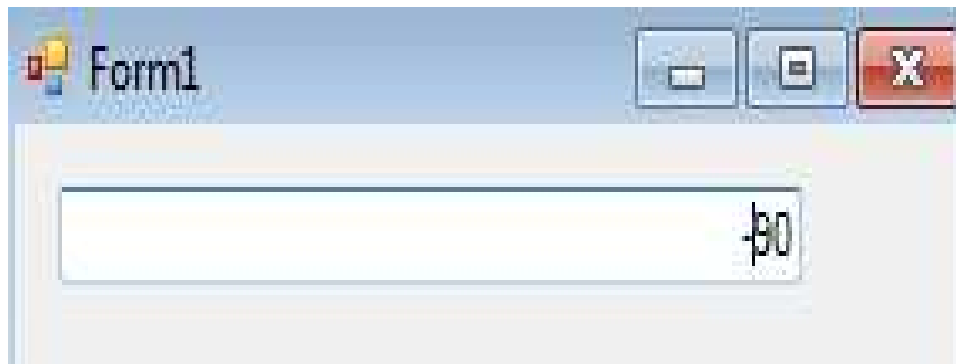
```

'UserControl1 overrides new to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
Public Sub New()
    ' This call is required by the Windows Form Designer.
    InitializeComponent()
    ' Add any initialization after the InitializeComponent() call.
    Me.Text = "0"
    INilai = 0
    DNilai = 0
    Me.TextAlign = HorizontalAlignment.Right
    Me.Refresh()
End Sub

```

# Gunakan NumericTextBox

- Build!
- Gunakan pada aplikasi Anda!



Properties

**NumericTextBox1** NumericTextB

Misc

AllowDecimal	<b>True</b>
AllowNeg	<b>True</b>
AutoComplete	<b>(Collection)</b>
AutoComplete	None
AutoComplete	None
DNilai	<b>0</b>
INilai	<b>0</b>
MaxValue	<b>0</b>
MinValue	<b>0</b>

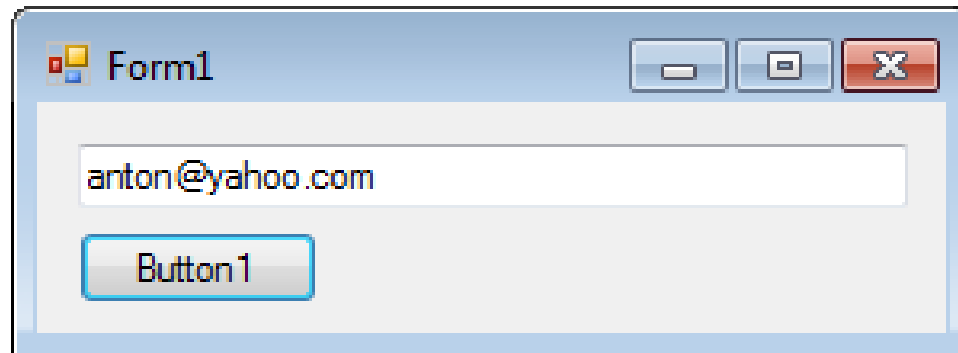
# EmailValidatorText

- Turunan TextBox untuk memvalidasi email address
- Property readonly **IsValid** as Boolean
- Private function **validasi**(byval eml as String)
- Gunakan **Regular Expression**

```
Protected Overrides Sub OnValidated(ByVal e As System.EventArgs)
    If validasi(Text) Then
        MsgBox("Benar")
    Else
        MsgBox("Salah")
    End If
End Sub
```

```
ReadOnly Property IsValid() As Boolean
    Get
        Return _isvalid
    End Get
End Property
```

# Hasil



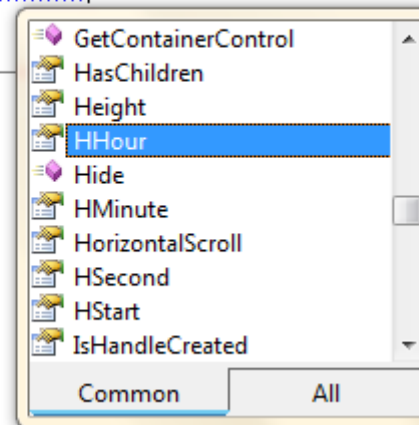
```
Private Function validasi(ByVal eml As String) As Boolean
    Dim pattern As String = "^[-a-zA-Z0-9][-a-zA-Z0-9]*@[-.a-zA-Z0-9]+(\.[-a-zA-Z0-9]+)*\." &
"(com|edu|info|gov|int|mil|net|org|biz|name|museum|coop|aero|pro|tv|[a-zA-Z]{2})$"
    Dim check As New Regex(pattern, RegexOptions.IgnorePatternWhitespace)
    Dim valid As Boolean = False
    If String.IsNullOrEmpty(eml) Then
        valid = False
    Else
        valid = check.IsMatch(eml)
    End If
    Return valid
End Function
```

# ClockLabel



- Suatu label control yang digabungkan dengan komponen timer, yang digunakan untuk menampilkan waktu sekarang
- Property: HHour, HMinute, HSecond, HStart
- Method: ClockEnable()

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.  
    DigitalClockLabel1.h  
End Sub  
Class
```



Public Property HHour() As Integer

# Kode

```
Public Class DigitalClockLabel
    Private hh As Integer
    Private mm As Integer
    Private ss As Integer
    Private start As Boolean = False

    Property HHour() As Integer
        Get
            Ret
            End Get
        Me.Label1.AutoSize = True
        Set (ByV Me.Label1.Font = New System.Drawing.Font("Arial Narrow", 15.75!, System.Drawing.FontStyle.R
            hh : Me.Label1.Location = New System.Drawing.Point(5, 4)
        End Set
    End Property
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(102, 25)

    Property HM
        Get
            Me.Label1.TabIndex = 0
            Ret Me.Label1.Text = CStr(Now.Hour) & " : " & CStr(Now.Minute) & " : " & CStr(Now.Second)
        End Get
        Set (ByVal value As Integer)
            mm = value
        End Set
    End Property

    Property HStart() As Boolean
        P
        Get
            Return start
        End Get
        Set (ByVal value As Boolean)
            start = value
            Timer1.Enabled = Not Timer1.Enabled
        End Set
    E
    End Property

    Public Sub ClockEnable (ByVal value As Boolean)
        HStart = value
        Timer1.Enabled = HStart
    End Sub

    Private Sub Timer1_Tick (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer
        Label1.Text = CStr(Now.Hour) & " : " & CStr(Now.Minute) & " : " & CStr(Now.Second)
    End Sub
End Class
```

# AdvancedEdit

- Tidak perlu property alignment karena VB.NET sudah memilikinya (rata kiri, tengah, atau kanan)
- Turunan dari TextBox

```
Partial Class AdvancedEdit
```

```
    Inherits System.Windows.Forms.TextBox
```

- Var private:
  - Private FOldBackColor As Color
  - Private FColorOnEnter As Color
  - Private FTabOnEnter As Boolean



# Property

```
Public Property ColorOnEnter() As Color
    Get
        Return FColorOnEnter
    End Get
    Set(ByVal value As Color)
        FColorOnEnter = value
    End Set
End Property
```

---

```
Public Property TabOnEnter() As Boolean
    Get
        Return FTabOnEnter
    End Get
    Set(ByVal value As Boolean)
        FTabOnEnter = value
    End Set
End Property
```

---

```
Protected Overrides Sub OnEnter(ByVal e As System.EventArgs)
    FOldBackColor = Me.BackColor
    Me.BackColor = ColorOnEnter
    MyBase.OnEnter(e)
End Sub
```

---

# Methods

```
Protected Overrides Sub OnEnter(ByVal e As System.EventArgs)
    FOldBackColor = Me.BackColor
    Me.BackColor = ColorOnEnter
    MyBase.OnEnter(e)
End Sub
```

---

```
Protected Overrides Sub OnLeave(ByVal e As System.EventArgs)
    BackColor = FOldBackColor
    MyBase.OnLeave(e)
End Sub
```

---

```
Protected Overrides Sub OnKeyDown(ByVal e As System.Windows.Forms.KeyEventArgs)
    MyBase.OnKeyDown(e)
    'MsgBox(TabOnEnter)
    If TabOnEnter Then
        If e.KeyCode =
            SendKeys.S
        End If
    End If
End Sub

Public Sub New()
    ' This call is required by the Windows Form Designer.
    InitializeComponent()

    ' Add any initialization after the InitializeComponent() call.
    ColorOnEnter = Me.BackColor
End Sub
```

# Contoh Aplikasi

A screenshot of a Windows application window titled "Form1". It contains three text boxes stacked vertically. The top text box contains the text "ab" and is highlighted with a blue background. The middle text box contains the text "ba". The bottom text box contains the text "cd".

A screenshot of a Windows application window titled "Form1". It contains three text boxes stacked vertically. The top text box contains the text "ab". The middle text box contains the text "ba". The bottom text box contains the text "cd" and is highlighted with a green background.

A screenshot of a Windows application window titled "Form1". It contains three text boxes stacked vertically. The top text box contains the text "ab". The middle text box contains the text "ba" and is highlighted with a teal background. The bottom text box contains the text "cd".

# NEXT

- Delphi Desktop Component