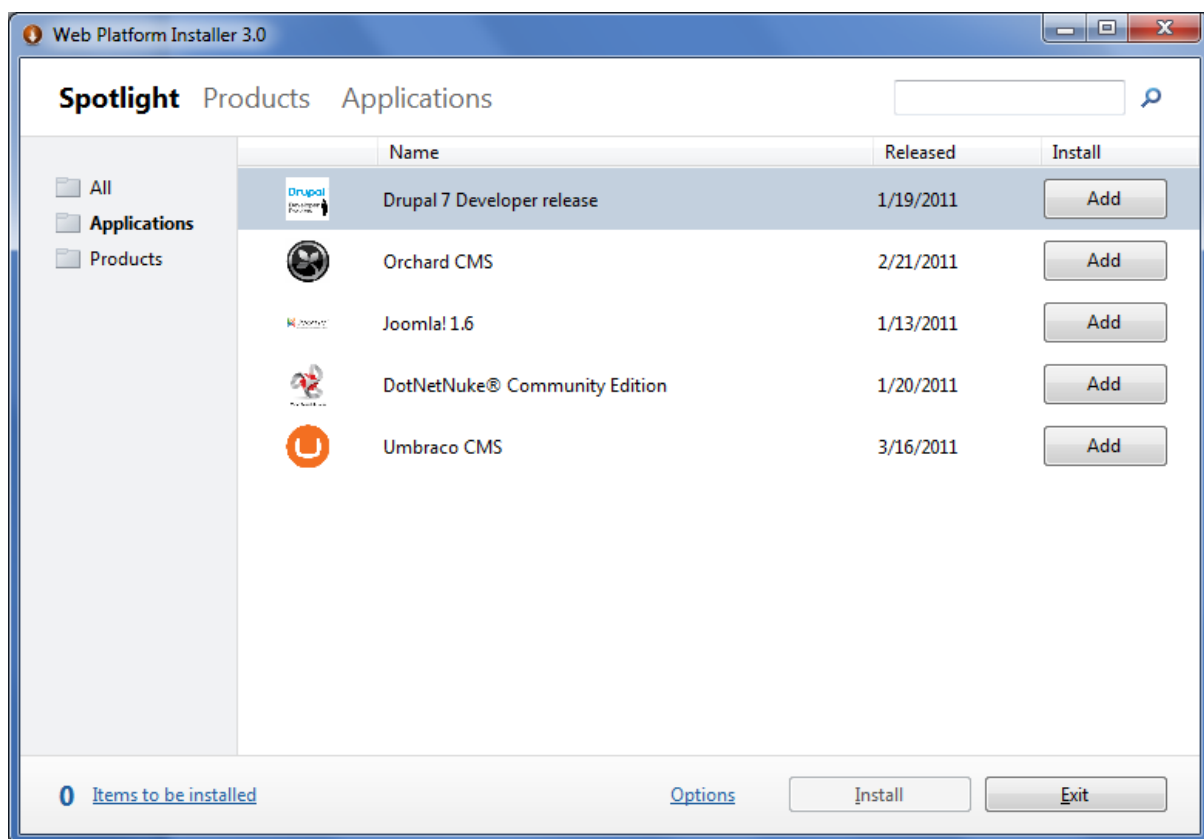


## Develop Simple Blog Using Microsoft WebMatrix

Membangun Blog Sederhana.....	2
Membangun desain database.....	3
Pembuatan Tabel Pengguna .....	5
Pembuatan Tabel Kategori.....	5
Pembuatan Tabel Artikel.....	5
Membuat Relasi Antar Tabel .....	5
Memilih dan Mengelola Template Website .....	7
Membuat Halaman Signup Member .....	11
Membuat Halaman Login.....	13
Membuat Halaman Tambah Kategori.....	14
Membuat Halaman Tulis Blog.....	15
Membuat Halaman Profile.....	17
Perbaiki Halaman Artikel .....	18
Membuat Halaman Hapus Blogpost .....	22
Melengkapi Halaman Index Beserta Tambahan – Tambahan Lain.....	22
Membuat Halaman Lain yang Diperlukan .....	25

# Develop Simple Blog Using Microsoft WebMatrix

WebMatrix merupakan salah satu tool terbaru yang dikembangkan oleh Microsoft yang dapat digunakan untuk membangun aplikasi berbasis web. Seperti yang sudah kita kenal sebelumnya, framework yang paling diandalkan oleh Microsoft adalah ASP.NET dan biasanya untuk membangun aplikasi yang menggunakan framework ini kita hanya mengandalkan Microsoft Visual Studio yang paling tidak kita harus memiliki spek komputer yang cukup tinggi dengan minimal ram 2gb. Munculnya tool WebMatrix ini menjadi salah satu opsi pilihan bagi para programmer yang menginginkan tool yang ringan, akan tetapi mampu mengakomodasikan framework ASP.NET dengan handal. Selain hal itu, banyak sekali platform yang dapat kita gunakan dengan menggunakan WebMatrix ini.



## Membangun Blog Sederhana

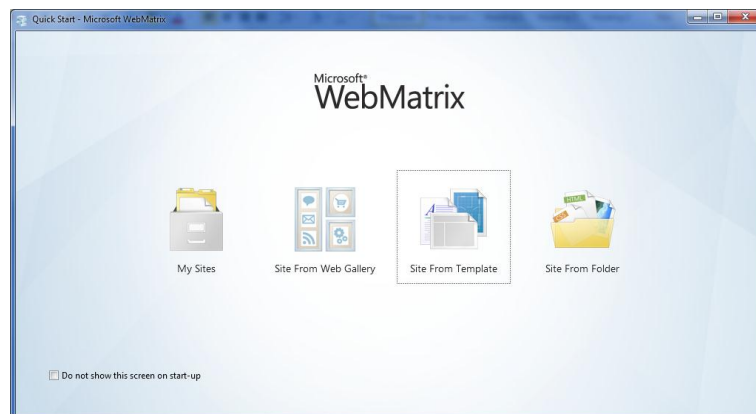
Blog merupakan salah satu website yang cukup kompleks apabila kita mau melihatnya sebagai satu aplikasi besar. Akan tetapi pada workshop kali ini, kita hanya akan membahas mengenai menampilkan blog post, mengedit, dan menghapus saja. Selebihnya bisa kita kembangkan sendiri untuk keperluan anda.

Tahapan yang akan kita lewati dalam membangun blog sederhana itu antara lain :

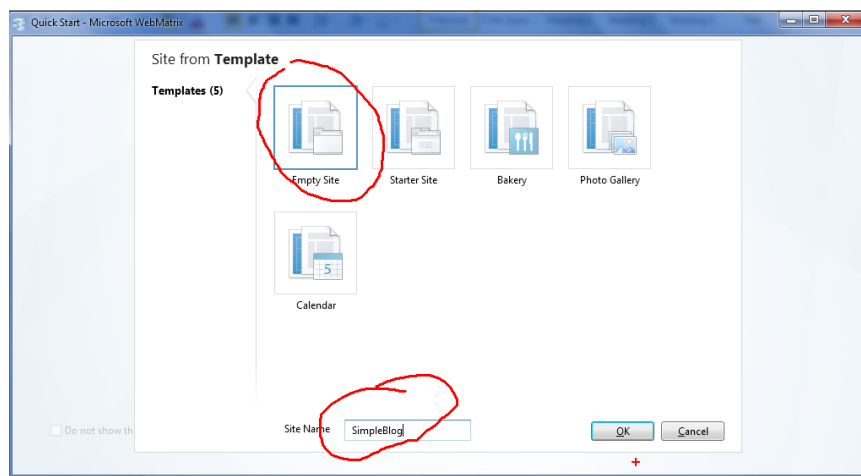
1. Membangun desain database
2. Memilih dan Mengelola Template Website
3. Membuat Halaman Signup Member
4. Membuat Halaman Login
5. Membuat Halaman Tambah Kategori
6. Membuat Halaman Tulis Blog
7. Membuat Halaman Profile
8. Perbaiki Halaman Artikel
9. Membuat Halaman Hapus Blogpost
10. Melengkapi Halaman Index Beserta Tambahan – Tambahan Lain
11. Membuat Halaman Lain yang Diperlukan

## Membangun desain database

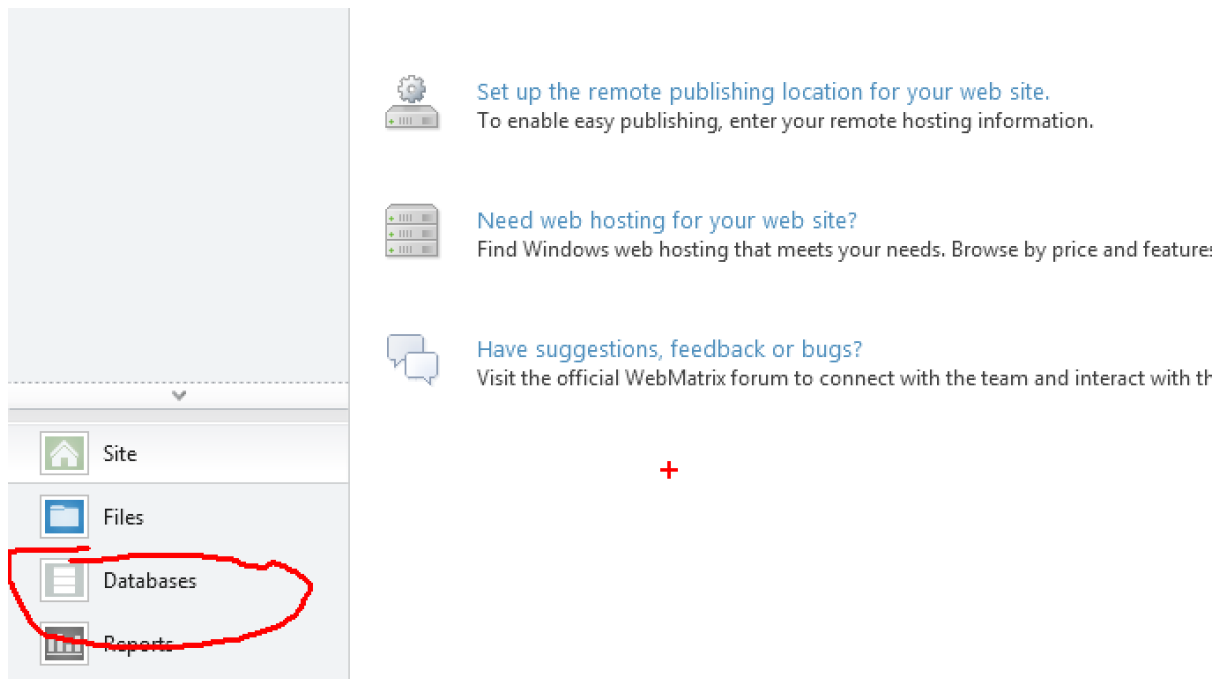
Hal pertama yang kita perlu lakukan adalah membuka aplikasi WebMatrix sehingga muncul tampilan dibawah ini :



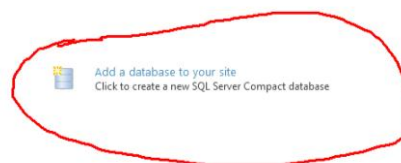
Selanjutnya karena kita akan membuat website yang kosong, maka kita bisa membuatnya dengan memilih Site From Template → Empty Site. Pada bagian Site Name beri nama SimpleBlog.



Setelah itu kita dapat langsung membuat database kita dengan memilih tab pada gambar dibawah ini :

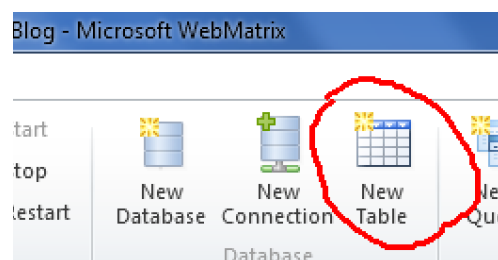


Selanjutnya kita bisa menambahkan database dengan mengklik gambar dibawah ini :



Lalu setelah kita mengklik gambar tersebut pada tab bagian kiri akan muncul konfirmasi adanya database baru yang telah siap kita gunakan. Hanya saja kita perlu memberikan nama terhadap database yang nantinya kita akan gunakan. Beri nama database kita dengan nama DbBlog.

Database tersebut masih kosong dan belum memiliki objek – objek database yang akan kita gunakan. Nah sekarang kita tinggal menambahkan table dengan cara menekan gambar dibawah ini :



Setelah kita mengklik gambar tersebut, maka kita akan diperhadapkan dengan antarmuka yang bisa kita gunakan untuk mengatur banyak kolom yang kita butuhkan pada tabel tersebut beserta dengan tipe data dan properti – properti lainnya yang bisa set pada bagian bawah. Hal yang terpenting dalam membangun tabel adalah kita mendefinisikan Primary Key terlebih dahulu dan usahakan

apabila kita mau menggunakan auto increment yang telah disediakan oleh Sql CE, tabel tersebut memang belum pernah dilakukan proses pengeditan tabel. Apabila sudah terlanjur menyimpan dan kita membutuhkan fungsi auto increment, maka solusi yang paling tepat adalah dengan mengulang pembuatan tabel tersebut.

### Pembuatan Tabel Pengguna

Set properti tabel ini menjadi seperti gambar dibawah ini. Selanjutnya tekan ctrl+s untuk menyimpan tabel tersebut. Beri nama Pengguna

Name	Allow Nulls	Properti			
		Data Type	Default Value	Is Primary Key?	Length
Username	False	Nchar	Null	True	10
Password	True	Nchar	Null	False	

### Pembuatan Tabel Kategori

Buat tabel baru dan berikanlah properti kolom seperti tabel dibawah ini dan simpan dengan nama Kategori.

Name	Allow Nulls	Data Type	Properti			
			Default Value	Is Identify	Is Primary Key?	Length
Id_Kat	False	Bigint	Null	True	True	-
Kategori	True	Nvarchar	Null	-	False	30

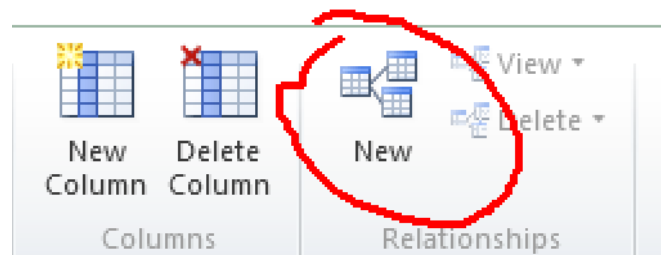
### Pembuatan Tabel Artikel

Buat tabel baru dan berikanlah properti kolom seperti tabel dibawah ini dan simpan dengan nama Artikel.

Name	Allow Nulls	Data Type	Properti			
			Default Value	Is Identify	Is Primary Key?	Length
Id_Art	False	Bigint	Null	True	True	-
Username	False	Nchar	Null	-	False	10
Id_Kat	False	Bigint	Null	False	False	-
Waktu	True	Datetime	Null	-	False	-
Isi	True	Nvarchar	Null	-	False	4000
Judul	True	Nvarchar	Null	-	False	50

### Membuat Relasi Antar Tabel

Untuk membuat relasi antar tabel, kita harus melihat definisi struktur tabel Artikel terlebih dahulu. Setelah kita melihat definisi tabel Artikel, maka kita akan melihat gambar dibawah ini pada bagian atas.



Setelah itu kita bisa memberikan nilai seperti dibawah ini :

**New Relationship**

**Foreign Key Table:** Artikel

**Primary Key Table:** Pengguna

☐ Id\_Art bigint  
☒ Username nchar (10)  
☐ Id\_Kat bigint  
☐ Waktu datetime  
☐ Isi nvarchar (4000)  
☐ Judul nvarchar (50)

☒ Username nchar (10)  
☐ Password nchar (15)

There is already a relationship with the name 'FK\_Artikel\_Pengguna' in this table. Please choose a different name.

Name: FK\_Artikel\_Pengguna

OK Cancel

Demikian juga kita menambahkan relasi pada tabel Kategori :

**New Relationship**

**Foreign Key Table:** Artikel

**Primary Key Table:** Kategori

☐ Id\_Art bigint  
☐ Username nchar (10)  
☒ Id\_Kat bigint  
☐ Waktu datetime  
☐ Isi nvarchar (4000)  
☐ Judul nvarchar (50)

☒ Id\_Kat bigint  
☐ Kategori nvarchar (30)

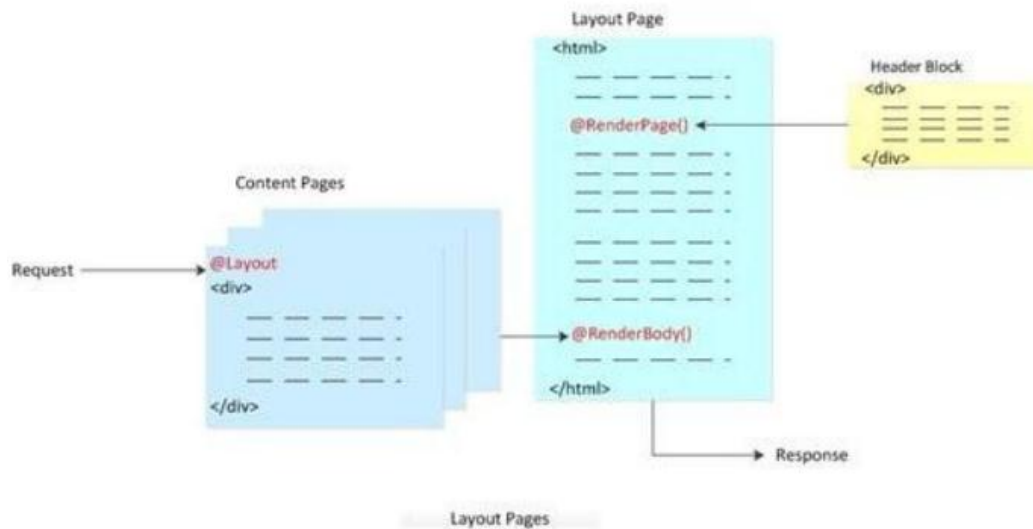
There is already a relationship with the name 'FK\_Artikel\_Kategori' in this table. Please choose a different name.

Name: FK\_Artikel\_Kategori

OK Cancel

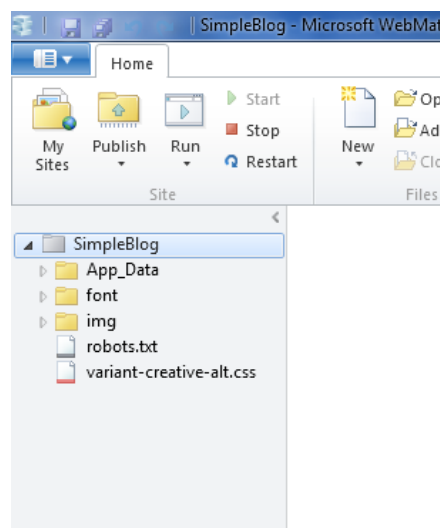
## Memilih dan Mengelolah Template Website

Konsisten merupakan salah satu unsur yang patut kita jaga dalam membangun sebuah web. Apabila kita harus melakukan penulisan tag – tag HTML ataupun pemberian import css yang kita buat, itu akan menjadi pekerjaan yang memusingkan dan cukup melelahkan. Oleh karena itu Microsoft menjembatani hal itu dengan memberikan metode –metode yang cukup memudahkan programmer untuk membuat tampilan yang konsisten.

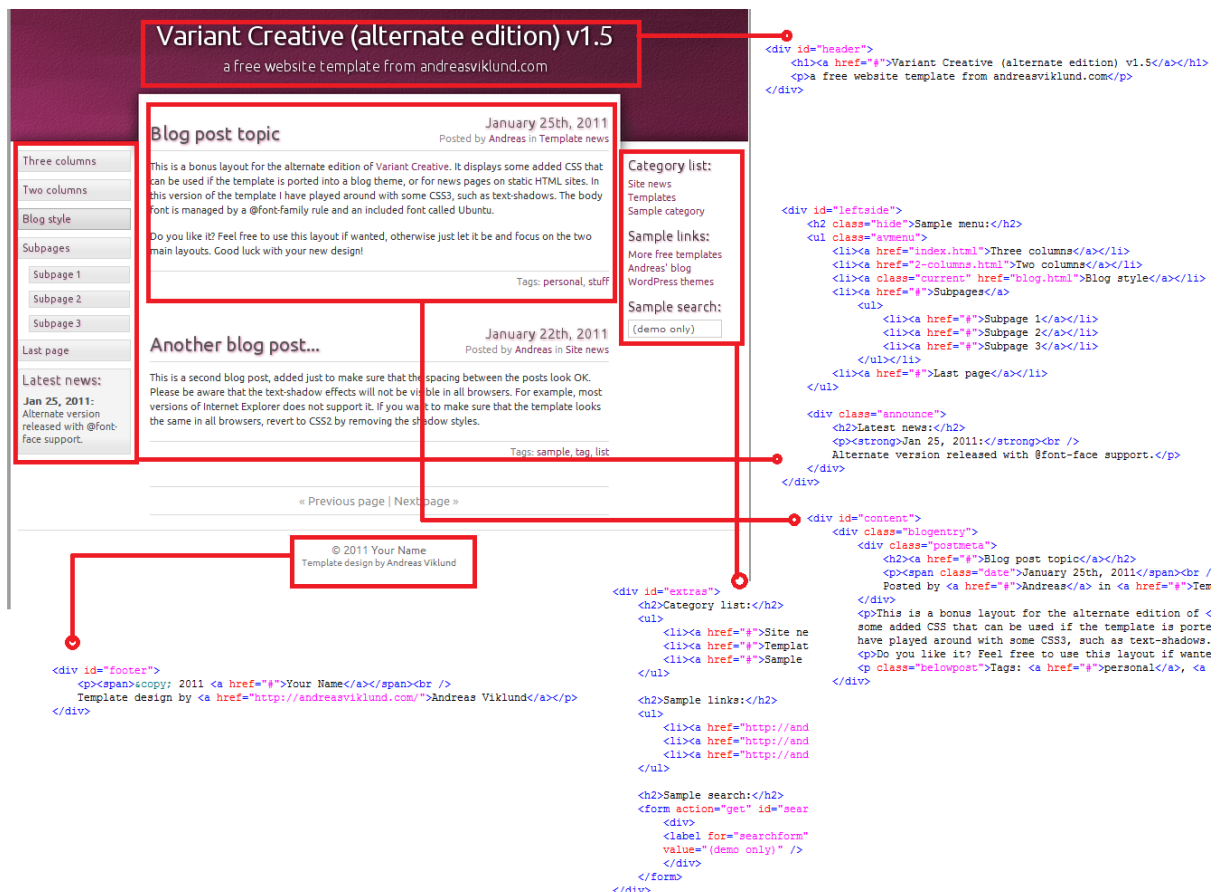


Bagan diatas menggambarkan bagaimana kita membuat tampilan yang konsisten. Biasanya dalam mendesain layout sebuah halaman kita akan membaginya menjadi 3 bagian. Bagian header, footer dan content atau isi. Serupa dengan pembagian itu, maka kita memerlukan sebuah halaman yang hanya mendefinisikan bagian header, footer. Lalu kita mencampur adukan pada halaman Layout. Bagian isi sendiri diisi pada halaman yang berbeda dan kita hanya tinggal mendefinisikan variable Layout yang telah disediakan oleh ASP.NET Razor.

Terlebih dahulu untuk belajar tentang membuat tampilan yang konsisten, bukalah template yang sediakan, copykan pada folder kita folder font,img, dan file variant-creative-alt.css. sehingga tampilan folder kita saat ini seperti gambar dibawah :

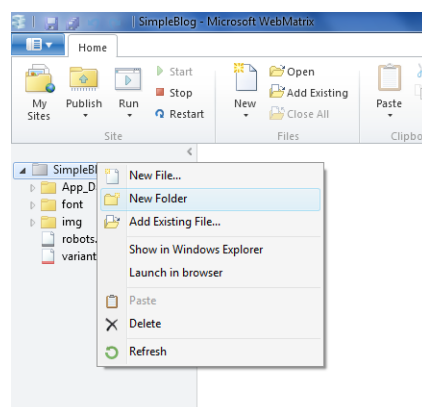


Pada file template yang sudah ada, buka halaman blog.html dengan menggunakan notepad, editplus, ataupun editor – editor lainnya untuk melihat bagaimana struktur tag yang diperlukan untuk menjadikan halaman yang kita buat sesuai dengan style yang dimiliki oleh template. Untuk lebih jelasnya, lihatlah gambar dibawah :



Gambar diatas menunjukan pembagian desain yang ada pada file blog.html. Langkah selanjutnya adalah membuat bagian – bagian tersebut sesuai dengan konsep ASP.NET Razor.

Untuk lebih jelasnya, pada project yang telah memiliki database, kita menambahkan folder baru dengan cara klik kanan dan pilih new folder. Berilah nama folder tersebut Shared.





Tambahkanlah file baru dengan nama `_Header.vbhtml` pada folder tersebut. File dengan ekstensi `.vbhtml` merupakan file dengan format yang didukung oleh ASP.NET Razor dengan menggunakan bahasa Visual Basic. Sedangkan apabila kita menginginkan menggunakan bahasa C#, maka kita bisa memilih file dengan ekstensi `.cshtml`.

Hapuslah seluruh isi pada halaman baru kita (`_Header.vbhtml`) dan isilah dengan tag HTML sebagai berikut :

```
<div id="header">
    <h1><a href="#">Variant Creative (alternate edition) v1.5</a></h1>
    <p>a free website template from andreasviklund.com</p>
</div>

<div id="leftside">
    <h2 class="hide">Sample menu:</h2>
    <ul class="avmenu">
        <li><a href="index.html">Three columns</a></li>
        <li><a href="2-columns.html">Two columns</a></li>
        <li><a class="current" href="blog.html">Blog style</a></li>
        <li><a href="#">Subpages</a>
            <ul>
                <li><a href="#">Subpage 1</a></li>
                <li><a href="#">Subpage 2</a></li>
                <li><a href="#">Subpage 3</a></li>
            </ul>
        </li>
        <li><a href="#">Last page</a></li>
    </ul>

    <div class="announce">
        <h2>Latest news:</h2>
        <p><strong>Jan 25, 2011:</strong><br />
        Alternate version released with font-face support.</p>
    </div>
</div>
```

Buat halaman baru lagi bernama `_Footer.vbhtml` pada folder yang tadi kita buat. Kemudian ganti isinya dengan tag HTML berikut :

```
<div id="footer">
    <p><span>&copy; 2011 <a href="#">Your Name</a></span><br />
    Template design by <a href="http://andreasviklund.com/">Andreas
    Viklund</a></p>
</div>
```

Lalu untuk mengintegrasikan kedua bagian tersebut buatlah halaman baru dengan nama `_Layout.vbhtml`. Isikan tag HTML dibawah ini :

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <link rel="stylesheet" type="text/css" href="@Href("~/variant-creative-
        alt.css")" media="all" />
        <title>Variant Creative Alt v1.5</title>
    </head>
    <body>
```

```

<div id="wrap">
    @RenderPage("~/Shared/_Header.vbhtml")

    <div id="content">
        @RenderBody()
    </div>

    @RenderPage("~/Shared/_Footer.vbhtml")
</div>
</body>
</html>

```

Pada ASP.NET Razor ini kita mulai mengenal cara pemrograman seperti yang php gunakan. Tanda '@' merupakan penanda bahwa sesuatu yang ditulis setelah tanda '@' merupakan baris program yang akan dijalankan. Pada bagian <head> kita mulai mengenal metode Href() yang dapat mengenal tanda ~. Tanda ~ merupakan current directory yang akan diubah otomatis menjadi file path secara fisik. Selanjutnya kita juga mulai dikenalkan dengan metode RenderPage() yang artinya bagian tersebut diimport dari halaman lain yang sudah kita definisikan pada parameternya. Sedangkan bagian RenderBody() sendiri akan digantikan pada halaman yang akan kita buat dengan cara memanggil variabel Layout. Lebih jelasnya pada current directory project kita, kita tambahkan file baru bernama index.vbhtml dan isi dengan :

```

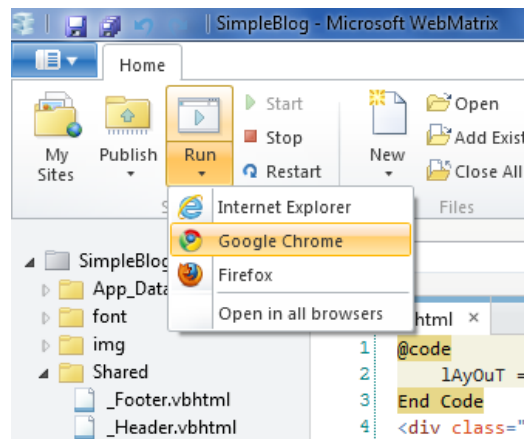
@code
    Layout = "~/Shared/_Layout.vbhtml"
End Code
<div class="blogentry">
    <div class="postmeta">
        <h2><a href="#">Blog post topic</a></h2>
        <p><span class="date">January 25th, 2011</span><br />
        Posted by <a href="#">Andreas</a> in <a href="#">Template
news</a></p>
    </div>
    <p>Do you like it? Feel free to use this layout if wanted, otherwise just
let it be and focus on the two main layouts. Good luck with your new design!</p>
    <p class="belowpost">Tags: <a href="#">personal</a>, <a
href="#">stuff</a></p>
</div>

<div class="blogentry">
    <div class="postmeta">
        <h2><a href="#">Another blog post...</a></h2>
        <p><span class="date">January 22th, 2011</span><br />
        Posted by <a href="#">Andreas</a> in <a href="#">Site news</a></p>
    </div>
    <p>This is a second blog post, added just to make sure that the spacing
between the posts look OK. Please be aware that the text-shadow effects will not
be visible in all browsers. For example, most versions of Internet Explorer does
not support it. If you want to make sure that the template looks the same in all
browsers, revert to CSS2 by removing the shadow styles.</p>
    <p class="belowpost">Tags: <a href="#">sample</a>, <a href="#">tag</a>, <a
href="#">list</a></p>
</div>

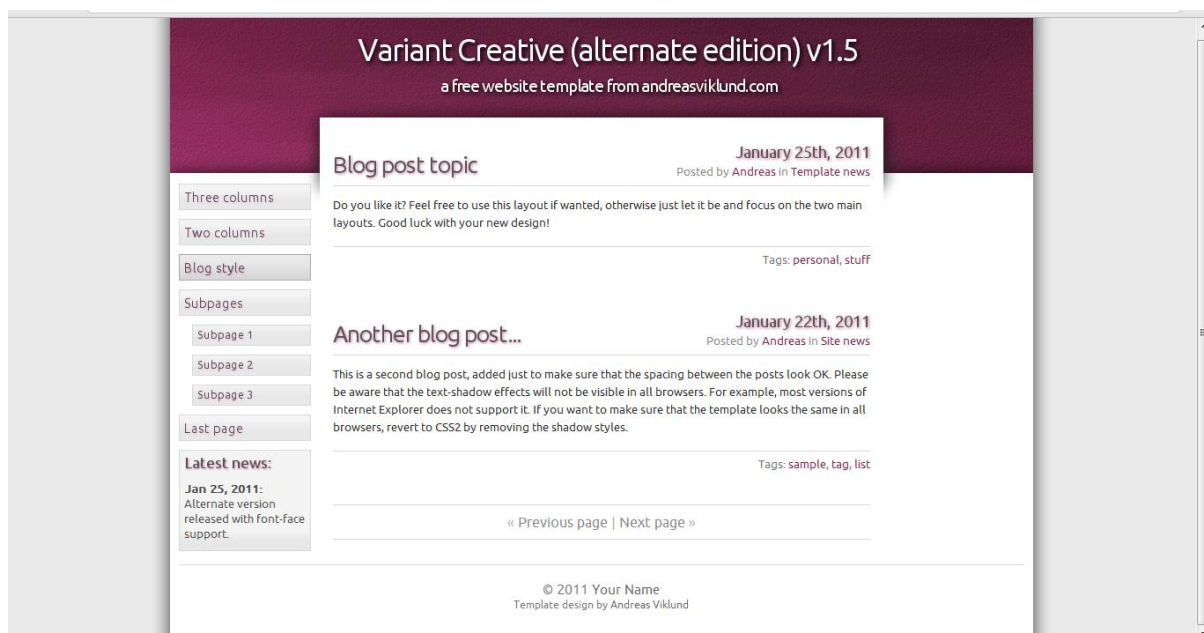
<div class="archivelinks">&laquo; <a href="#">Previous page</a> | <a href="#">Next
page</a> &raquo;</div>

```

Jalankan halaman Index.vbhtml tersebut dengan menggunakan tombol seperti dibawah ini. Amati hasilnya, apakah sudah sesuai dengan template yang sudah ada.



Hasil dari pekerjaan kita dibrowser seperti gambar dibawah :



## Membuat Halaman Signup Member

Buatlah sebuah halaman baru dengan nama Signup.vbhtml. Kita memiliki tabel Pengguna pada database kita. Kolom yang terdapat pada tabel tersebut adalah Username dan Password. Untuk keperluan itu kita bisa membuat form sebagai berikut :

```
@code
  Layout = "~/Shared/_Layout.vbhtml"
End Code
<form method="post" action="">
  <div class="blogentry" align="center">
    <h2>Signup your membership</h2>
    <label for="Username">Username</label><br/>
    <input type="text" name="Username"/><br/><br/>
    <label for="Password">Password</label><br/>
```

```



```

Dengan demikian kita sudah mendapatkan antarmuka yang kita inginkan. Akan tetapi kita belum memberikan penambahan coding apapun untuk membuatnya bisa berjalan sesuai dengan yang kita inginkan. Untuk itu karena form ini digunakan untuk keperluan penambahan data pengguna, maka kita memerlukan koneksi ke database seperti aplikasi – aplikasi lain dan melakukan perintah DML bagian Insert. Permasalahan yang muncul adalah terkadang kita membutuhkan validasi agar data – data yang memang masuk kedalam database kita sesuai dengan apa yang dibutuhkan oleh sistem. Oleh karena itu kita akan mempelajari bagaimana melakukan insert data sekaligus belajar melakukan validasi dengan metode yang telah disediakan oleh ASP.NET Razor. Pada bagian bawah Layout, tambahkan code berikut :

```

Dim db = Database.Open("DbBlog")

If IsPost Then
    Dim name = Request("Username")
    If name.IsEmpty Then
        ModelState.AddModelError("Username", "Username harus diisi!")
    End If

    Dim pass = Request("Password")
    If pass.IsEmpty Then
        ModelState.AddModelError("Password", "Password harus diisi!")
    End If

    Dim repass = Request("Repassword")
    If repass.IsEmpty Then
        ModelState.AddModelError("Repassword", "Repassword harus diisi!")
    End If

    If repass <> pass Then
        ModelState.AddModelError("Repassword", "Password dan Repassword harus
sama!")
    End If

    If ModelState.IsValid Then
        Dim strSql = "INSERT INTO Pengguna VALUES(@0,@1)"
        db.Execute(strSql, name, pass)

        Response.Redirect("~/Index?Ket=Sukses")
    End If

```

Selanjutnya setelah perintah End Code, tambahkan code berikut :  
@Html.ValidationSummary("Terdapat kesalahan : ")

Variabel IsPost merupakan variabel bertipe boolean yang akan bernilai true ketika melakukan post atau pada contoh kasus diatas kita menekan tombol Submit. Ketika kita menekan tombol Submitlah kita mulai mengecek data – data dengan memanggil metode Request() sebagai variabel yang menampung data – data pada form. Jika ternyata kosong, maka kita bisa menggunakan kelas ModelState yang telah disediakan untuk menambahkan pesan kesalahan.

Jika ternyata tidak terjadi kesalahan kita mulai melakukan proses input data langsung ke database dengan menggunakan metode Execute() yang menerima inputan string beserta dengan parameter – parameter terkait. Untuk tambahannya apabila user sukses dalam menambah data pada tabel pengguna, maka diarahkan ke halaman lain bernama Index dengan menambahkan querystring bernama Ket dengan nilai Sukses. Ini akan terlihat gunanya setelah kita membuka halaman Index. Bukalah halaman Index dan tambahkan kode program berikut dibawah Layout :

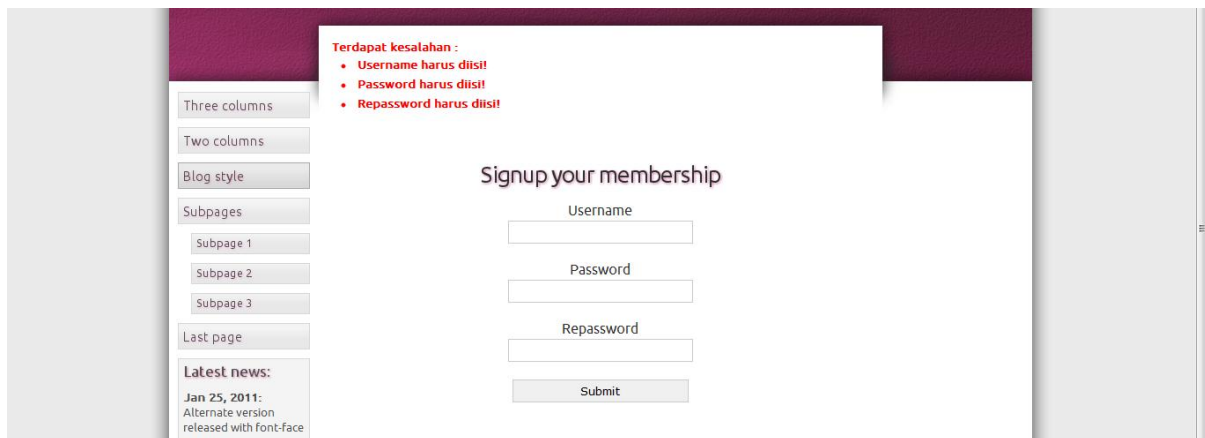
```
Dim konfirmasi = ""
If Request.QueryString("Ket") = "Sukses" Then
    konfirmasi = "Selamat anda telah sukses mendaftar!"
End If
```

Pada bagian bawah End Code tambahkan baris program berikut :

```
<h3 style="color:Red">@konfirmasi</h3>
```

Selanjutnya buka halaman \_Layout, kita tambahkan sedikit CSS agar penggunaan HTML.ValidationSummary() kita lebih sempurna :

```
<style type="text/css">
    .validation-summary-errors{font-weight:bold; color:Red;}
</style>
```



## Membuat Halaman Login

Buatlah folder baru pada current directory anda, dan berikanlah nama User. Selanjutnya tambahkan file baru dengan nama Login.vbhtml. Lalu isikan kode dibawah ini :

@code

```
Layout = "~/Shared/_Layout.vbhtml"
```

```
Dim ket = Request.QueryString("Ket")
```

```
Dim pesan = ""
```

```
If ket = "Err" Then
```

```
    ModelState.AddModelError("Login", "Username atau Password salah!")
```

```
End If
```

```
If ket = "Logout" Then
```

```
    Session.Clear()
```

```

        pesan = "Anda telah berhasil logout!"
    End If

    If IsPost Then
        Dim name = Request("Username")
        Dim pass = Request("Password")

        Dim db = Database.Open("DbBlog")
        Dim strSql = "SELECT * FROM Pengguna WHERE Username=@0 AND Password=@1"

        Try
            Dim row = db.QuerySingle(strSql, name, pass)
            Session.Add("Username", row.Username)
            Session.Timeout = 60
            Response.Redirect("~/User/Profile")
        Catch ex As NullReferenceException
            Response.Redirect("~/User/Login?Ket=Err")
        End Try

    End If

End Code
@pesan
@Html.ValidationSummary("Terdapat kesalahan : ")
<form method="post" action="">
    <div class="blogentry" align="center">
        <h2>Form Login</h2>
        <fieldset>
            <label for="Username">Username : </label> <br/>
            <input type="text" name="Username" value="@Request("Username")" />
        <br/><br/>
            <label for="Password">Password : </label> <br/>
            <input type="password" name="Password" /> <br/> <br/>
            <input type="submit" value="Submit" />
        </fieldset>
    </div>
</form>

```

Sebenarnya ada berbagai macam cara atau algoritma yang bisa diterapkan untuk membuat login. Akan tetapi pada kasus ini saya menggunakan cara select karena menurut saya cara itu adalah cara paling mudah. Oleh karena bisa saja user salah memasukan data, maka kita memerlukan try catch agar apabila memang isi data dari hasil select kita null, kita bisa memberikan konfirmasi bahwa username kita itu salah. Apabila ternyata masukan kita benar, maka kita bisa menambahkan sesi kedalam server. Hal inipun kondisional, kita bisa menggunakan cookies ataupun session. Tetapi contoh penggunaannya saat ini adalah dengan menggunakan session.

## Membuat Halaman Tambah Kategori

Buat halaman TambahKategori.vbhtml pada folder User. Gantikan tag yang sudah ada dengan code dan tag HTML berikut :

```

@code
Layout = "/Shared/_Layout.vbhtml"

```

```

If Session.Count = 0 Then
    Response.Redirect("~/User/Login")
End If

Dim db = Database.Open("DbBlog")
Dim judul, isi, kat
Dim kategori = ""
If IsPost Then
    kategori = Request("Kategori")
    If kategori.IsEmpty Then
        ModelState.AddModelError("Kategori", "Kategori harus diisi!")
    End If

    If ModelState.IsValid Then
        Dim strSql = "INSERT INTO Kategori(kategori) VALUES(@0)"

        db.Execute(strSql, kategori)
        Response.Redirect("~/User/Artikel")
    Else
        Dim strSql = "INSERT INTO Artikel(Username,Id_Kat,Waktu,Isi)
VALUES(@0,@1,@2,@3)"

        db.Execute(strSql)
    End If
End If

End Code
@Html.ValidationSummary("Terdapat kesalahan : ")
<div class="blogentry" align="center">
    <h2>Tambah Kategori</h2>
    <form method="post" action="">
        <label for="Kategori">Kategori</label><br/>
        <input type="text" name="Kategori"/><br/><br/>
        <input type="submit" value="Tambah"/>
    </form>
</div>

```

## Membuat Halaman Tulis Blog

Buatlah halaman dan beri nama Artikel.vbhtml pada folder User. Selanjutnya gantikan tag yang sudah ada dengan :

```

@code
    Layout = "~/Shared/_Layout.vbhtml"
    Dim db = Database.Open("DbBlog")
    Dim Ket = Request.QueryString("Ket")
    Dim pesan = ""
    If Ket = "Sukses" Then
        pesan = "Blogpost anda telah dipublish!"
    End If

    Dim judul, kat, isi As String

    If Session.Count = 0 Then
        Response.Redirect("~/User/Login")
    End If

```

```

Dim data

If Not IsPost Then
    Dim strSql = "SELECT * FROM Kategori ORDER BY Id_Kat"
    data = db.Query(strSql)
Else
    judul = Request("Title")
    If judul.IsEmpty Then
        ModelState.AddModelError("Title", "Judul harus diisi!")
    End If
    isi = Request("Isi")
    If isi.IsEmpty Then
        ModelState.AddModelError("Isi", "Harus terdapat isi!")
    End If
    kat = Request("Kategori")
    If kat.IsEmpty Then
        ModelState.AddModelError("Kategori", "Harus memilih kategori!")
    End If

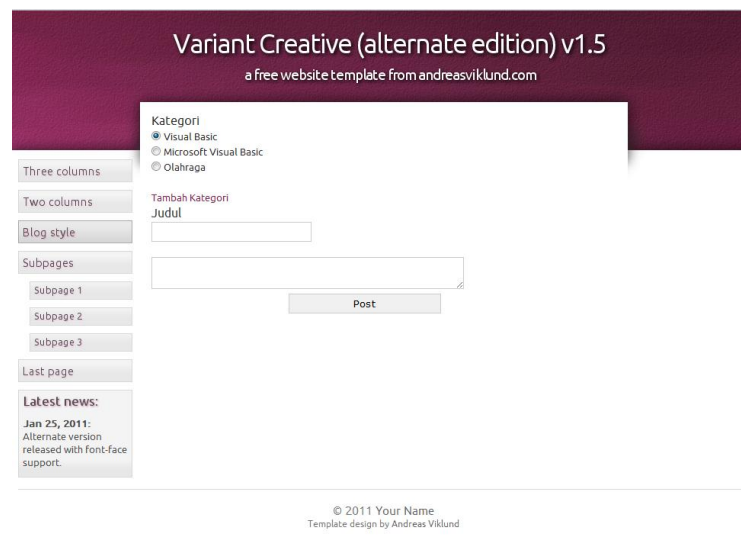
    If ModelState.IsValid Then
        Dim strSql = "INSERT INTO Artikel(Username,Id_Kat,Waktu,Isi,Judul)
VALUES(@0,@1,@2,@3,@4)"
        db.Execute(strSql, Session("Username"), kat, Date.Now, isi, judul)
        Response.Redirect("~/User/Artikel?Ket=Sukses")
    End If
End If

End Code
<h3 style="color:Red">@pesan</h3>
<div id="blogentry">
    <form method="post" action="">
        <label for="Kategori">Kategori</label><br/>
        @For Each row In data
            @:<input style="width:auto" type="radio" name="Kategori"
value="@row.Id_Kat" /> @row.Kategori <br/>
        Next
        <br/>
        <a href="@Href("~/User/TambahKategori")">Tambah Kategori</a><br/>
        <label for="Title">Judul</label><br/>
        <input type="text" name="Title" />
        <br/><br/>
        <textarea id="Isi" name="Isi" rows="0" cols="0"></textarea>
        <br/>
        <input style="margin-left:180px" type="submit" value="Post"/>
    </form>
</div>

```



Hasilnya adalah :



Akan tetapi apabila kita menambahkan isi yang memiliki nilai berupa tag HTML dan kemudian mencoba menyimpannya, maka terjadi error dibawah ini :

#### Server Error in '/' Application.

*A potentially dangerous Request.Form value was detected from the client (Isi="<p>asdadas</p> <i>sa...").*

**Description:** Request Validation has detected a potentially dangerous client input value, and processing of the request has been aborted. This value may indicate an attempt to compromise the security of your application, such as a cross-site scripting attack. To allow pages to override application request validation settings, set the requestValidationMode attribute in the httpRuntime configuration section to requestValidationMode="2.0". Example: <httpRuntime requestValidationMode="2.0" />. After setting this value, you can then disable request validation by setting validateRequest="false" in the <pages> configuration section. However, it is strongly recommended that your application explicitly check all inputs in this case. For more information, see <http://go.microsoft.com/fwlink/?Linkid=153133>.

**Exception Details:** System.Web.HttpRequestValidationException: A potentially dangerous Request.Form value was detected from the client (Isi="<p>asdadas</p> <i>sa...").

#### Source Error:

```
Line 25:         ModelState.AddModelError("Title", "Judul harus diisi!")
Line 26:     End If
Line 27:     Isi = Request("Isi")
Line 28:     If Isi.IsEmpty Then
Line 29:         ModelState.AddModelError("Isi", "Harus terdapat isi!")
```

**Source File:** C:\Users\Chrysan\Documents\My Web Sites\SimpleBlog\User\Artikel.vbhtml **Line:** 27

#### Stack Trace:

```
[HttpRequestValidationException (0x80004005): A potentially dangerous Request.Form value was detected from the client (Isi="<p>asdadas</p> <i>sa...").]
System.Web.HttpRequest.ValidateString(String value, String collectionKey, RequestValidationSource requestCollection) +8730676
Microsoft.Web.Infrastructure.DynamicValidationHelper.<.>c__DisplayClass12.<ReplaceCollectionb_d(String value, String key)>+79
Microsoft.Web.Infrastructure.DynamicValidationHelper.LazilyEvaluatedNameObjectEntry.ValidateObject() +89
Microsoft.Web.Infrastructure.DynamicValidationHelper.LazilyValidatingHashTable.GetItem(Object key) +54
System.Collections.Specialized.NameValueCollectionBase.FindEntry(String key) +20
System.Collections.Specialized.NameValueCollection.Get(String name) +8
System.Web.HttpRequest.GetItem(String key) +52
System.Web.HttpRequestWrapper.GetItem(String key) +11
ASP._Page_User_Artikel_vbhtml.Execute() in C:\Users\Chrysan\Documents\My Web Sites\SimpleBlog\User\Artikel.vbhtml:27
System.Web.WebPages.WebPageBase.ExecutePageHierarchy() +207
System.Web.WebPages.WebPage.ExecutePageHierarchy(IEnumerable`1 executors) +68
System.Web.WebPages.WebPage.ExecutePageHierarchy() +156
System.Web.WebPages.WebPageBase.ExecutePageHierarchy(WebPageContext pageContext, TextWriter writer, WebPageRenderingBase startPage) +76
System.Web.WebPages.WebPageHttpHandler.ProcessRequestInternal(HttpContext context) +249
```

**Version Information:** Microsoft .NET Framework Version 4.0.30319; ASP.NET Version 4.0.30319.1

Hal ini terjadi karena fungsi Request Validation sendiri melarang adanya penambahan tag HTML pada isi karena validation secara otomatis menilai bahwa nilai yang diinputkan bisa membahayakan sistem. Untuk mengatasi hal tersebut maka kita perlu menambahkan fungsi berikut pada baris 27 :

```
Isi = Request.Unvalidated("Isi")
```

## Membuat Halaman Profile

Halaman profile ini berguna sebagai halaman dashboard yang kita miliki. Sehingga dengan adanya halaman ini kita bisa melakukan manajemen terhadap blog kita. Misalnya saja kita bisa menulis blogpost kita melalui halaman ini, melihat seluruh blogpost yang kita buat dan lalu bisa melakukan manajemen seperti hapus blogpost ataupun melakukan edit. Akan tetapi halaman Artikel yang tadi

kita buat belum mampu melakukan edit data. Nantinya akan ada penambahan baris program kedalamnya.

Buatlah halaman Profile.vbhtml pada folder User, kemudian gantikan isi halaman tersebut dengan :

```
@code
    Layout = "~/Shared/_Layout.vbhtml"

    If Session.Count = 0 Then
        Response.Redirect("~/User/Login")
    End If

    Dim ket = Request.QueryString("Ket")
    Dim pesan = ""
    If ket = "HapusSukses" Then
        pesan = "Anda berhasil menghapus blogpost"
    End If

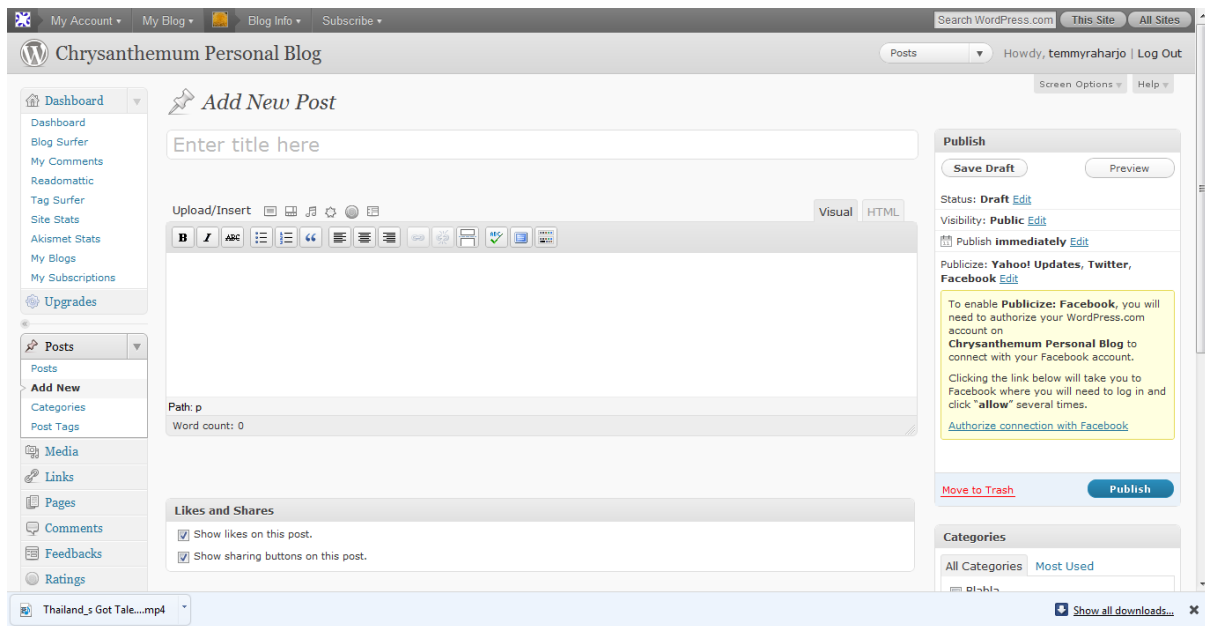
    Dim strSql = "SELECT Id_Art, Judul, Kategori, Waktu FROM Artikel A, Kategori K
WHERE A.Id_Kat = K.Id_Kat AND username=@0 ORDER BY Id_Art DESC"
    Dim Db = Database.Open("DbBlog")
    Dim data = Db.Query(strSql, Session("Username"))

End Code
<h3 style="color:Red">@pesan</h3>
<div class="blogentry">
    <h2>Data Blog yang telah dibuat</h2>
    <table>
        <thead>
            <tr>
                <td>&nbsp;</td>
                <td>Judul</td>
                <td>Kategori</td>
                <td>Waktu</td>
            </tr>
        </thead>
        <tbody>
            @For Each row In data
                @:<tr><td><a
href="@Href(String.Format("~/User/Artikel?Ket=Edit&Art={0}",
row.Id_Art))">Edit</a> &nbsp;&nbsp;&nbsp;<a href="@Href("~/User/Delete",
row.Id_Art)">Delete</a></td>
                @:<td>@row.Judul</td><td>@row.Kategori</td><td>@row.Waktu</td></tr>
            Next
        </tbody>
    </table>
    <a href="@Href("~/User/Artikel")">Create blogpost</a>
</div>
```

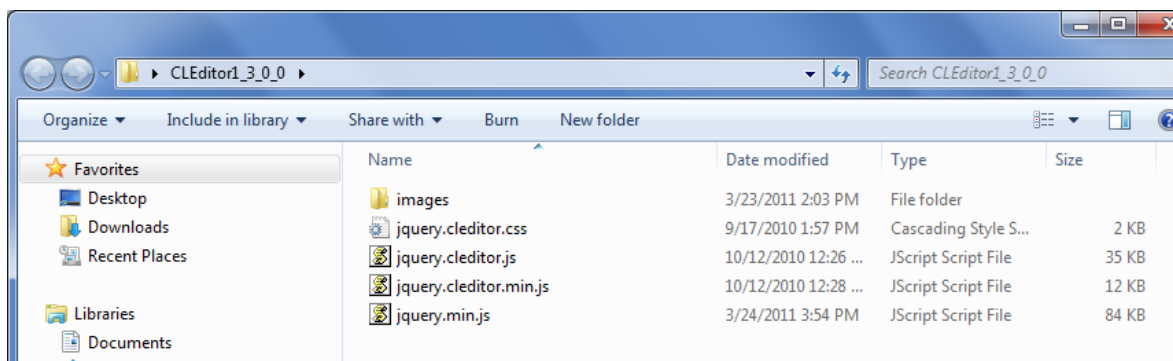
## Perbaiki Halaman Artikel

Halaman Artikel yang sudah kita buat agaknya kurang sesuai atau terlampau biasa – biasa saja. Apalagi editor merupakan salah satu faktor penting yang membuat kita nyaman dalam menulis. Oleh sebab itu diperlukan tampilan yang benar – benar menarik dan membuat kita sedikit mengetahui tag HTML akan tetapi dapat menggunakannya dengan baik. Dalam jQuery, kita biasa mengenal editor

yang demikian adalah WYSIWYG editor. WYSIWYG adalah kependekan dari What You See Is What You Get yang artinya apa yang kita ketikkan, kita lihat, itulah yang akan kita dapatkan.



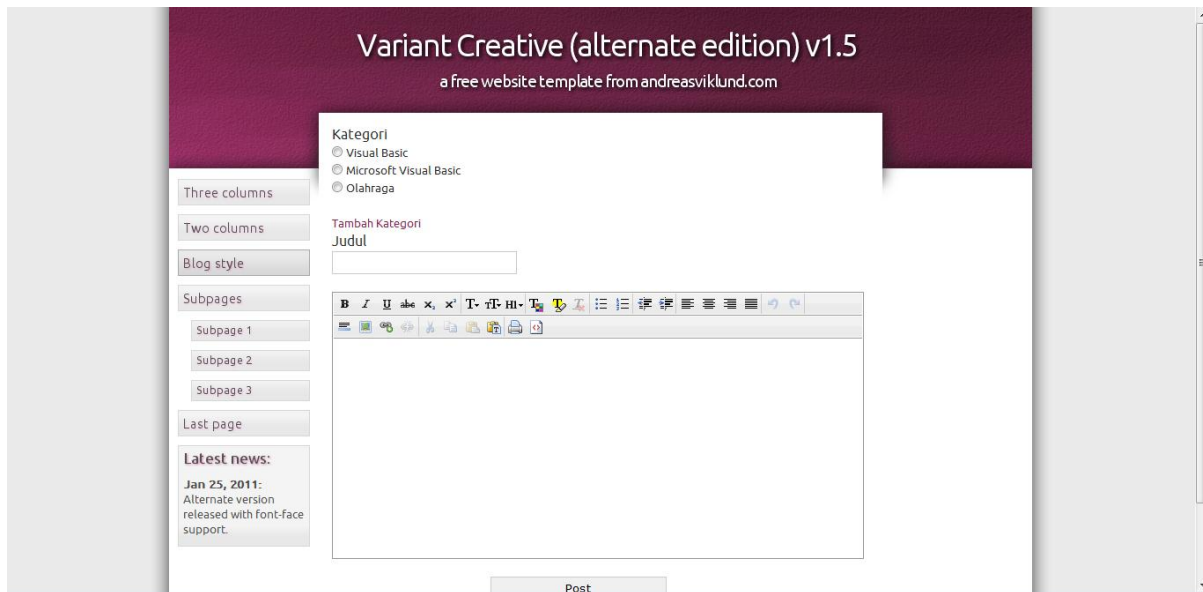
Oleh karena itu copykan WYSIWYG editor yang sudah disediakan, copy file pada gambar berikut ke directory project anda :



Bukalah halaman \_Layout.vbhtml dan tambahkan tag berikut :

```
<link rel="stylesheet" type="text/css" href="@Href("~/jquery.cleditor.css")" />
<script type="text/javascript" src="@Href("~/jquery.min.js")"></script>
<script type="text/javascript"
src="@Href("~/jquery.cleditor.min.js")"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $("#Isi").cleditor({width:600,height:300});
    });
</script>
```

Cobalah Run kembali halaman Artikel anda dan viola :



Sekarang kita akan membereskan halaman Artikel supaya dapat mampu mengedit blogpost kita. Sebenarnya ASP.NET sendiri sudah memiliki fungsi tersendiri untuk mengirim nilai yang sangat mirip dengan lemparan nilai versi Code Igniter. Akan tetapi pada kasus ini saya tetap menggunakan querystring agar lebih mudah. Ubahlah halaman tersebut hingga menjadi :

```
@code
Layout = "~/Shared/_Layout.vbhtml"
Dim db = Database.Open("DbBlog")
Dim Ket = Request.QueryString("Ket")
Dim pesan = ""
If Ket = "Sukses" Then
    pesan = "Blogpost anda telah dipublish!"
End If

'tambahan
If Ket = "EditSukses" Then
    pesan = "Blogpost anda telah berhasil diupdate!"
End If

Dim judul, kat, isi As String

If Ket = "Edit" And Not Request.QueryString("Art").IsEmpty Then
    Dim strSql = "SELECT * FROM Artikel WHERE Id_Art=@0"
    Dim datanya = db.QuerySingle(strSql, Request.QueryString("Art"))
    judul = datanya.Judul
    kat = datanya.Id_Kat
    isi = datanya.Isi
End If

If Session.Count = 0 Then
    Response.Redirect("~/User/Login")
End If

Dim data

'tambahan
```

```

If Not IsPost Then
    Dim strSql = "SELECT * FROM Kategori ORDER BY Id_Kat"
    data = db.Query(strSql)
Else
    judul = Request("Title")
    If judul.IsEmpty Then
        ModelState.AddModelError("Title", "Judul harus diisi!")
    End If
    isi = Request.Unvalidated("Isi")
    If isi.IsEmpty Then
        ModelState.AddModelError("Isi", "Harus terdapat isi!")
    End If
    kat = Request("Kategori")
    If kat.IsEmpty Then
        ModelState.AddModelError("Kategori", "Harus memilih kategori!")
    End If

    'tambahan
    If Not Ket = "Edit" Then
        If ModelState.IsValid Then
            Dim strSql = "INSERT INTO
Artikel(Username,Id_Kat,Waktu,Isi,Judul) VALUES(@0,@1,@2,@3,@4)"
            db.Execute(strSql, Session("Username"), kat, Date.Now, isi,
judul)
            Response.Redirect("~/User/Artikel?Ket=Sukses")
        End If
    Else
        If ModelState.IsValid Then
            Dim strSql = "UPDATE Artikel SET
Id_Kat=@0,Waktu=@1,Isi=@2,Judul=@3 WHERE Id_Art=@4"
            db.Execute(strSql, kat, Date.Now, isi,
judul,Request.QueryString("Art"))
            Response.Redirect("~/User/Artikel?Ket=EditSukses")
        End If
    End If
End If
End Code
<h3 style="color:Red">@pesan</h3>
<div id="blogentry">
    <form method="post" action="">
        <label for="Kategori">Kategori</label><br/>
        @For Each row In data
            If row.Id_Kat = kat Then
                @:<input style="width:auto" type="radio" name="Kategori"
value="@row.Id_Kat" checked="checked" /> @row.Kategori <br/>
            Else
                @:<input style="width:auto" type="radio" name="Kategori"
value="@row.Id_Kat" /> @row.Kategori <br/>
            End If
        Next
    <br/>
    <a href="@Href("~/User/TambahKategori")">Tambah Kategori</a><br/>
    <label for="Title">Judul</label><br/>
    <input type="text" name="Title" value="@judul"/>
    <br/><br/>
    <textarea id="Isi" name="Isi" rows="0" cols="0">@isi</textarea>
    <br/>
    <input style="margin-left:180px" type="submit" value="Post"/>
</form>

```

```
</div>
```

## Membuat Halaman Hapus Blogpost

Untuk keperluan lain seperti yang telah dilihat pada halaman profile, kita membutuhkan satu halaman yang dapat menghapus blogpost kita. Maka pada folder User, buatlah halaman berikut :

@code

```
Layout = "~/Shared/_Layout.vbhtml"

If Session.Count = 0 Then
    Response.Redirect("~/User/Login")
End If

Dim db = Database.Open("DbBlog")
Dim strSql = "SELECT judul FROM Artikel WHERE Id_Art=@0"
Dim Art = UrlData(0)

Dim row = db.QuerySingle(strSql, Art)

If IsPost Then
    strSql = "DELETE FROM Artikel WHERE Id_Art=@0"
    db.Execute(strSql, Art)
    Response.Redirect("~/User/Profile?Ket=HapusSukses")
End If
End Code

<div class="blogentry" align="center">
    <h2>Hapus Artikel</h2>
    <form method="post" action="">
        <p>Apakah anda yakin menghapus artikel dengan judul @row.Judul ?</p>
        <input type="submit" value="Hapus" /><br /><br />
        <a href="@Href("~/User/Profile")">Back to Profile</a>
    </form>
</div>
```

## Melengkapi Halaman Index Beserta Tambahan – Tambahan Lain

Untuk memperlihatkan blogpost yang kita buat, maka kita memerlukan halaman Index yang biasanya akan ditampilkan pertama kali dibandingkan dengan halaman – halaman lain. Oleh karena itu, ubahlah halaman Index.vbhtml pada current directory anda. Gantilah kodenya dengan :

@code

```
Layout = "~/Shared/_Layout.vbhtml"

Dim konfirmasi = ""
If Request.QueryString("Ket") = "Sukses" Then
    konfirmasi = "Selamat anda telah sukses mendaftar!"
End If

Dim db = Database.Open("DbBlog")
Dim strSql = "SELECT Id_Art,Username, K.Id_Kat, K.Kategori, Waktu, Isi, Judul
FROM Artikel A, Kategori K WHERE " +
    "A.Id_Kat = K.Id_Kat ORDER BY Id_Art DESC"
```

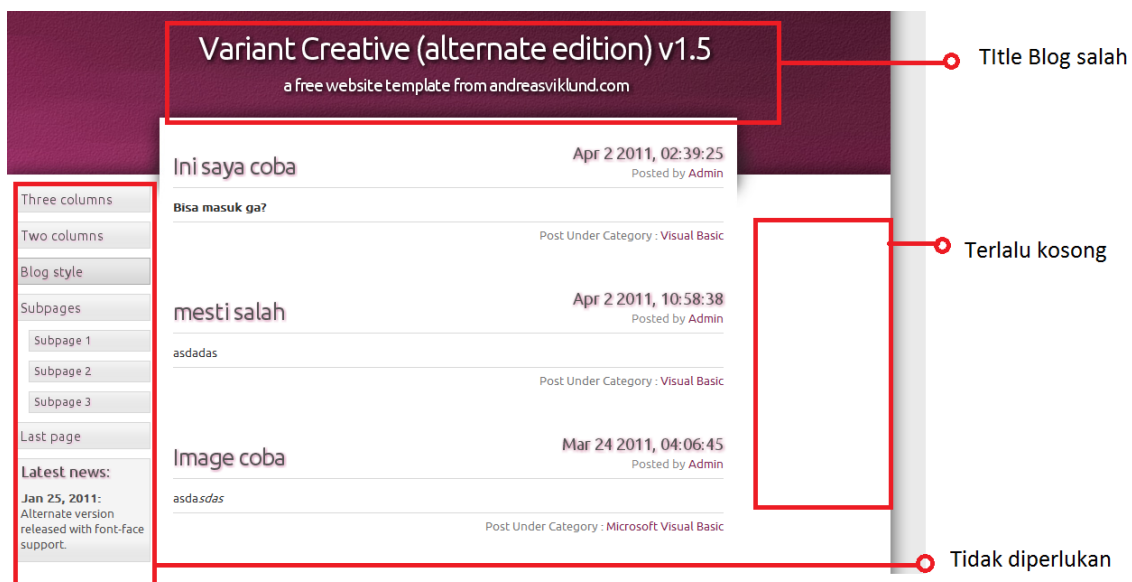
```

If Not Request.QueryString("User").IsEmpty Then
    strSql = String.Format("SELECT Id_Art,Username, K.Id_Kat, K.Kategori,
Waktu, Isi, Judul FROM Artikel A, Kategori K WHERE " +
    "A.Id_Kat = K.Id_Kat AND Username='{0}' ORDER BY Id_Art
DESC",Request.QueryString("User"))
End If

If Not Request.QueryString("Kat").IsEmpty Then
    strSql = "SELECT Id_Art,Username, K.Id_Kat, K.Kategori, Waktu, Isi, Judul
FROM Artikel A, Kategori K WHERE " +
    "A.Id_Kat = K.Id_Kat AND K.Id_Kat=" + Request.QueryString("Kat") + "
ORDER BY Id_Art DESC"
End If

Dim data = db.Query(strSql)
End Code
<h3 style="color:Red">@konfirmasi</h3>
@For Each row In data
    @:<div class="blogentry">
        @:<div class="postmeta">
            @:<h2><a href="@Href("~/LihatArtikel",
row.Id_Art)">@row.Judul</a></h2>
            @:<p><span class="date">@Format(row.Waktu,"MMM d yyyy,
hh:mm:ss")</span><br />Posted by <a href="@Href("~/Index?User="+row.Username)">
@:@row.Username</a></p></div>
            @Html.Raw(row.Isi)
            @:<p class="belowpost">Post Under Category :<a
href="@Href("~/Index?Kat="+row.Id_Kat.ToString())"> @row.Kategori</a></p>
        @:</div>
Next

```



Lihatlah gambar diatas. Itulah hasil apabila Index kita dijalankan. Terdapat banyak kekurangan. Oleh karena itu kita mulai memperbaiki template kita. Selain itu seharusnya kita memperlihatkan link login ataupun signup member pada kiri atas. Oleh karena itu pada halaman \_Header kita ubah @code

```

Dim db = Database.Open("DbBlog")
Dim strSql = "SELECT * FROM Kategori"

```

```

Dim data = db.Query(strSql)
End Code
<div id="User">
    @If Session.Count = 0 Then
        @:<p><a href="@Href("~/User/Login")">Login</a> |
        @:<a href="@Href("~/Signup")">Signup</a></p>
    Else
        @:<p><a href="@Href("~/User/Profile")">My Profile</a> |
        @:<a href="@Href("~/User/Login?Ket=Logout")">Logout</a></p>
    End If
</div>
<div id="header">
    <h1><a href="#">Mugi</a></h1>
    <p>simple blog using ASP.NET Razor</p>
</div>

<div id="leftside">
    <h2 class="hide">Sample menu:</h2>
    <ul class="avmenu">
        <li><a href="@Href("~/Index")">Home</a></li>
        <li><a href="#">Kategori</a>
        <ul>
            @For Each baris In data
                @:<li><a href="@Href(String.Format("~/Index?Kat={0}",
baris.Id_Kat))">@baris.Kategori</a></li>
            Next
        </ul>
        </li>
    </ul>
</div>

```

Dengan adanya penambahan tag div dengan Id="User", maka kita perlu merubah sedikit css kita pada halaman \_Layout. Tambahkan style ini :

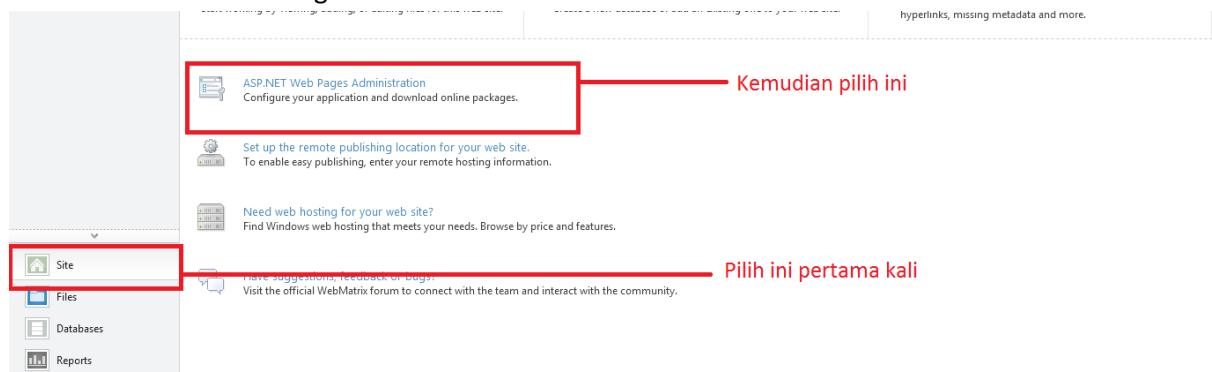
```

#User {clear:right;float:right;font-weight:bold;padding-right:30px;}
#User a, #User p{color:White}

```

Langkah selanjutnya adalah kita mengisi kekosongan pada bagian kiri. Pada ASP.NET Razor ini kita dapat memasukan widgets dengan sangat mudah. Pada contoh kali ini saya menginginkan adanya penambahan widget Twitter pada blog saya. Maka hal yang perlu dilakukan adalah sebagai berikut :

- Bukalah bagian Site pada WebMatrix anda
- Pilih ASP.NET Web Pages Administration

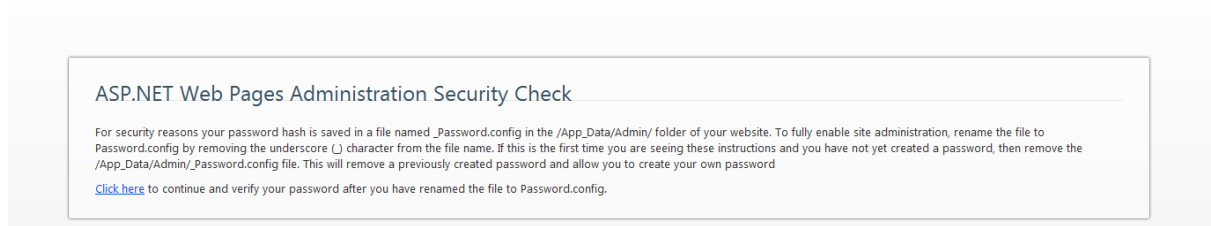


- Selanjutnya anda akan dibawa ke halaman website. Isilah password anda sesuai dengan keinginan anda.

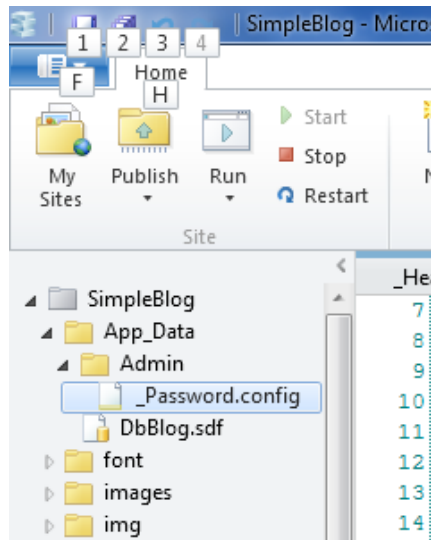


- Setelah sukses anda akan dibawa ke sebuah halaman yang berisikan informasi dibawah :

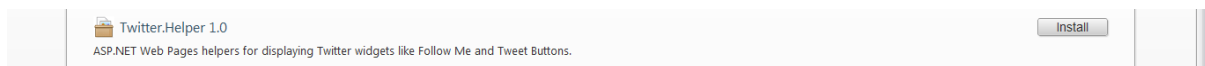
 ASP.NET Web Pages Administration



- Lihatlah WebMatrix anda, buka bagian files dan lihatlah pada App\_Data apakah anda memiliki data dibawah ini :



- Hilangkanlah underscore pada nama file tersebut dan rerun lagi ASP.NET Web Pages Administration. Masukkan password anda dan carilah Twitter goodies.



- Instal-lah helper tersebut dan bila selesai, tambahkanlah baris tag HTML dan program ini pada halaman \_Header yang berada pada folder Shared :

```
<div id="extras">
    <h2>My Twitter Profile</h2>
    @TwitterGoodies.Profile("temmy_raharjo")
</div>
```

## Membuat Halaman Lain yang Diperlukan

Terkadang kita membutuhkan untuk melihat artikel secara satuan. Nah untuk melengkapi hal tersebut maka kita akan membuat halaman yang dapat melihat artikel secara satuan. Buatlah halaman baru bernama LihatArtikel.vbhtml. Kemudian isikan dengan kode berikut :

```
@code
PageData("title") = "Lihat Artikel"
Layout = "~/Shared/_Layout.vbhtml"

Dim art = UrlData(0)
```

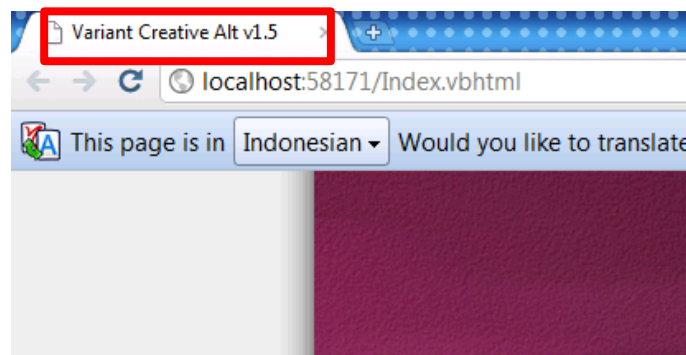
```

if art.isEmpty
    Response.Redirect("~/Index")
end if
Dim db = Database.Open("DbBlog")
Dim strSql = "SELECT Id_Art,Username, K.Id_Kat, K.Kategori, Waktu, Isi, Judul
FROM Artikel A, Kategori K WHERE " +
    "A.Id_Kat = K.Id_Kat AND Id_Art=@0"
Dim row = db.QuerySingle(strSql,art)
End Code

<div class="blogentry">
    <div class="postmeta">
        <h2><a href="@Href("~/Index", row.Id_Art)">@row.Judul</a></h2>
        <p><span class="date">@Format(row.Waktu,"MMM d yyyy, hh:mm:ss")</span><br
    />Posted by <a href="@Href("~/Index?User="+row.Username)">
        @row.Username</a></p></div>
        @Html.Raw(row.Isi)
        <p class="belowpost">Post Under Category :<a
href="@Href("~/Index?Kat="+row.Id_Kat.ToString())"> @row.Kategori</a></p>
    </div>

```

Lihatlah gambar dibawah :



Apabila kita mencoba menjalankan halaman web kita, kita bisa mengetahui bahwa judul halaman kita tidaklah dinamis. Hal ini dikarenakan pada halaman layout kita memberikan tag HTML <title></title> dan halaman ini kita pergunakan ke semua halaman yang kita buat.

Meskipun demikian, kita bisa membuat judul halaman tersebut menjadi dinamis. Untuk membuat halaman judul kita bisa berubah – ubah, bukalah halaman \_Layout.vbhtml dan ubahlah bagian dibawah ini :

```

4 <html lang="en">
3 <head>
4 <link rel="stylesheet" type="text/css" href="@Href("~/variant-creative-alt.css")" media="all" />
5 <title>@PageData("title")</title>
6 <style type="text/css">

```

Metode PageData() inilah yang akan membuat kita bisa memasukan judul halaman secara dinamis. Metode ini seakan – akan menjadi sebuah variabel yang dapat kita isi sesuka hati dan pada halaman – halaman lainnya kita perlu menambahkan metode ini dan mengisinya seperti contoh berikut :

```

@code
PageData("title") = "Halaman Muka"
Layout = "/Shared/_Layout.vbhtml"

```

-End Tutor-