

VB 2008

untuk Berbagai Keperluan

PEMROGRAMAN

- Semua tentang VB 2008 ada di sini; pengenalan, perintah, module, table, semua dibahas dengan contoh aplikatif!
- Pengenalan berbagai program sederhana hingga pembuatan berbagai aplikasi pemrograman populer.
- Cocok untuk inspirasi penambahan fitur-fitur proyek pemrograman dan juga pembuatan tugas akhir.

Hendrayudi

VB 2008 untuk Berbagai KEPERLUAN PROGRAMMING

<http://pustaka-indo.blogspot.com>

Sanksi Pelanggaran Pasal 72:
Undang-Undang Nomor 19 Tahun 2002
Tentang Hak Cipta

1. Barangsiapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagai-mana dimaksud dalam Pasal 2 Ayat (1) atau Pasal 49 Ayat (1) dan Ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima milyar rupiah).
2. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait sebagai dimaksud pada Ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

VB 2008 untuk Berbagai KEPERLUAN PROGRAMMING

<http://pustaka-indu.blogspot.com>

PENERBIT PT ELEX MEDIA KOMPUTINDO



KOMPAS GRAMEDIA

VB 2008 untuk Berbagai Keperluan Programming

Hendrayudi

© 2009, PT Elex Media Komputindo, Jakarta

Hak cipta dilindungi undang-undang

Diterbitkan pertama kali oleh

Penerbit PT Elex Media Komputindo

Kelompok Gramedia, Anggota IKAPI, Jakarta 2009

Editor: Whindy Yoevestian

EMK121090520

ISBN: 978-979-27-4576-4

<http://pustaka-indo.blogspot.com>

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta

Isi di luar tanggung jawab percetakan

KATA PENGANTAR

Microsoft Visual Basic 2008 adalah bahasa pemrograman berbasis visual yang merupakan versi lanjutan dari Microsoft Visual Basic 6.0 yang tergabung dalam Microsoft Visual Studio bersama dengan pemrograman visual C, C++, dan Visual FoxPro. Buku ini berisi tentang pengenalan bahasa pemrograman Microsoft Visual Basic 2008, instalasi Visual Basic 2008, komponen-komponen yang terdapat pada Visual Basic 2008, serta teknik mengetik dan membuat perintah.

Setelah Anda mengenal cara mengetik perintah dasar, Anda akan diarahkan untuk membuat program yang sederhana dengan perintah-perintah Microsoft Visual Basic 2008. Kemudian pada bab berikutnya Anda akan diajak untuk mempelajari lebih rinci tentang perintah-perintah String Module, File System, Data Time Module, InterAction, dan Constant Module. Buku ini juga berisi tentang pemrograman database mengenai cara mengisikan data, menghapus, menyimpan dan mencari data yang terdapat di dalam tabel, serta teknik menggunakan perintah-perintah grafik.

Akhir kata, penulis mengucapkan selamat belajar dan salam sukses.

Salam sejahtera,

Hendrayudi

DAFTAR ISI

Kata Pengantar	v
Daftar Isi	vii
BAB 1 VISUAL BASIC 2008	1
1.1 Instalasi Visual Basic 2008	2
1.2 Menjalankan Aplikasi.....	6
1.3 Komponen Visual Basic 2008	11
1.4 Membuat Perintah.....	12
BAB 2 BERMAIN DENGAN PROGRAM SEDERHANA	17
2.1 Menghitung Luas Objek	17
2.2 Operasi Bilangan	20
2.3 Membuat Jam Digital	23
2.4 Mencari Bilangan Terbesar	25
2.5 Mencari Rata-Rata	27
2.6 Password	29
2.7 Cek Bilangan Ganjil Genap	32
BAB 3 BERMAIN DENGAN PERINTAH STRING MODULE	35
3.1 Menggabungkan String	36
3.2 Membuat Pesan	39
3.3 Teks Huruf Kecil dan Besar	42
3.4 Mengambil Character.....	44
3.5 Membuang Spasi Variabel	46
3.6 Membandingkan Dua Buah String	47
3.7 Mengambil Nilai ASCII dan Character	49

3.8	Format	51
3.9	Mengambil Posisi Character dan Split	56
3.10	Duplikat dan Mengganti String.....	59
BAB 4	BERMAIN DENGAN PERINTAH FILE SYSTEM	61
4.1	Create, Change, dan Delete Folder	61
4.2	File	64
4.2.1	FileDialogTime	65
4.2.2	FileLen.....	66
4.2.3	CopyFile	67
4.3	Atribut File dan Baca File Text	69
4.4	Menghapus File	71
4.5	Mengunci dan Membuka File	72
4.6	Mengganti Nama File.....	75
4.7	Mencetak Tulisan.....	75
BAB 5	BERMAIN DENGAN PERINTAH DATE AND TIME MODULE79	
5.1	Tanggal dan Waktu Sistem	80
5.1.1	Now	80
5.1.2	Today	80
5.1.3	TimeOfDay	81
5.2	Proses Tanggal	82
5.2.1	DateAdd	82
5.2.2	DateDiff	86
5.2.3	DatePart.....	90
5.3	Menampilkan Tanggal	93
5.3.1	DateSerial	93
5.3.2	DateValue	94
5.3.3	MonthName	95
5.3.4	WeekDayName.....	96
5.4	Mengatur Tanggal dan Waktu	98
5.4.1	TimeSerial	98
5.4.2	TimeValue	100
5.4.3	DateString.....	101
5.4.4	TimeString	101

5.5	Proses Waktu.....	103
5.5.1	Timer.....	103
5.5.2	IsDate	104
BAB 6	BERMAIN DENGAN INTERACTION DAN CONSTANTS MODULE	107
6.1	Interaction Module	107
6.1.1	AppActive dan Shell	107
6.1.2	Beep	108
6.1.3	Choose	108
6.1.4	IIF	110
6.1.5	InputBox	111
6.1.6	MsgBox.....	113
6.2	Constants Module	115
6.2.1	vbAbort.....	115
6.2.2	Constants Module lain.....	116
BAB 7	BERMAIN DENGAN TABLE	121
7.1	Membuat Tabel.....	121
7.1.1.	Menambahkan Data	124
7.1.2.	Menggunakan DataGridView	124
7.1.3.	Menghubungkan TextBox dengan Table	125
7.2	Button Navigator	127
7.3	Kontrol Format TextBox	130
7.4	Pencarian Data	135
7.5	Menyaring Data dengan Filter	138
BAB 8	MEMBUAT APLIKASI.....	143
8.1	Membuat Aplikasi File Text	143
8.2	Membuat Aplikasi Informasi Drive	146
8.3	Membuat Aplikasi UseFindInFiles	149
8.4	Print Form	151
8.5	Aplikasi UseOpenFileDialog	155
8.6	Aplikasi Factorial.....	158
8.7	Aplikasi Debug Windows	160
8.8	Aplikasi Use Dialogs	163

8.9	Aplikasi Make Button.....	172
8.10	Aplikasi Transparent Form	176
8.11	Aplikasi Validasi Integer.....	178
BAB 9	APLIKASI GRAPHIC.....	181
9.1	Aplikasi Draw Text Path.....	182
9.2	Aplikasi Draw Bezier.....	184
9.3	Aplikasi Arsir Garis.....	185
9.4	Aplikasi Gambar Rectangle	188
9.5	Aplikasi Save Restore	192
9.6	Aplikasi Graphic Texture	193
9.7	Aplikasi PathClip	195
9.8	Aplikasi Transform Text	196
9.9	Aplikasi LoadPicture	198
9.10	Aplikasi AutoRedraw	199
9.11	Aplikasi Transform Arrow	202
9.12	Aplikasi Closed Curve.....	204
9.13	Aplikasi LineCaps.....	206
9.14	Aplikasi List Installed Font.....	208
Daftar Pustaka	211	
Tentang Penulis	213	



VISUAL BASIC 2008

<http://pustaka-indo.blogspot.com>

Visual Basic berasal dari singkatan BASIC (Beginner's All-purpose Symbolic Instruction Code) yang dibuat oleh Profesor Jhon Kemeny dan Thomas Kurtz dari Darmont pada pertengahan tahun 1960. Perintah-perintah bahasa program yang digunakan adalah bahasa Inggris, dengan tujuan dapat mempermudah programmer yang menggunakan bahasa pemrograman ini.

Bahasa pemrograman BASIC dikembangkan dengan berbagai bentuk, diantaranya adalah Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, dan Apple BASIC. Apple BASIC dikembangkan oleh Steve Wozniak, seorang karyawan Hewlett-Packard yang pada akhirnya pada bulan April 1976 secara resmi membentuk perusahaan Apple Computer.

Kemudahan menggunakan bahasa pemrograman BASIC akhirnya mendorong Microsoft untuk mengembangkan bahasa BASIC dengan GUI-BASED. *Graphical User Interface* membuat pengguna bahasa Basic semakin senang dengan komponen yang disediakan oleh pembuatnya, mereka merasakan kemudahan dalam menggunakan dan membuat program dengan bahasa yang berbasis visual.

Sejak itu bahasa pemrograman Visual Basic berkembang dengan berbagai versi, dan sampai pada akhirnya muncul bahasa pemrograman Visual Basic 2008 atau Visual Basic 9. Visual Basic 2008 adalah salah satu kelompok bahasa pemrograman yang dibuat oleh Microsoft dan tergabung dalam satu paket bahasa pemrograman Microsoft Visual Studio 2008. Paket pemrograman tersebut terdiri dari Microsoft Visual C# 2008, Microsoft Visual Basic 2008, Microsoft C++ 2008, dan Microsoft Web Developer 2008.

<http://pustaka-indo.blogspot.com>

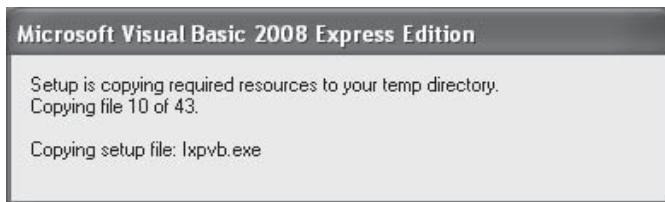
1.1 Instalasi Visual Basic 2008

Apabila Anda belum mempunyai aplikasi Visual Basic 2008, maka Anda dapat men-download aplikasi tersebut dari alamat web Microsoft Visual Basic di <http://www.Microsoft.com/express/vb/>. Apabila Anda sudah mempunyai DVD Visual Studio 2008, Anda tinggal memasukkan DVD tersebut hingga tampil seperti Gambar 1.1 lalu Anda pilih Visual Basic 2008.



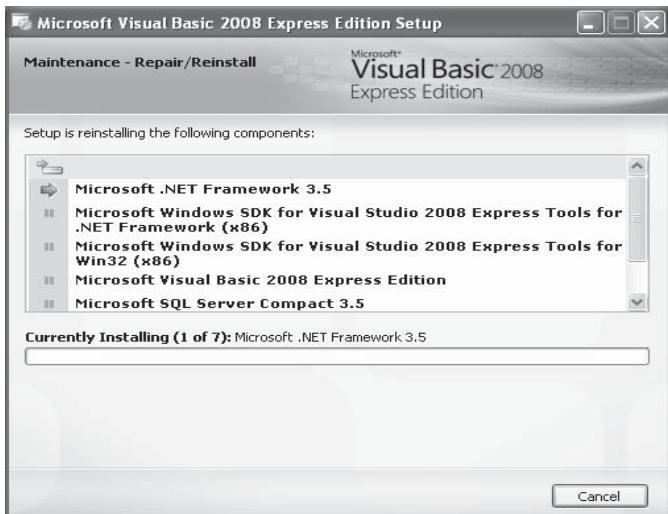
Gambar 1.1 Setup Visual Studio 2008

Lalu Anda klik Visual Basic 2008 untuk meng-install aplikasi Visual Basic, atau pilih aplikasi yang lain apabila Anda ingin meng-install aplikasi C atau Visual Web Developer, lalu akan muncul tampilan di bawah ini.



Gambar 1.2 Copy setup file Visual Basic 2008

Klik Next untuk melanjutkan setup aplikasi Visual Basic 2008 lalu pilih *I have read and accept the license term* kemudian klik Next. Klik button Install untuk melanjutkan instalasi bahasa pemrograman Visual Basic 2008.



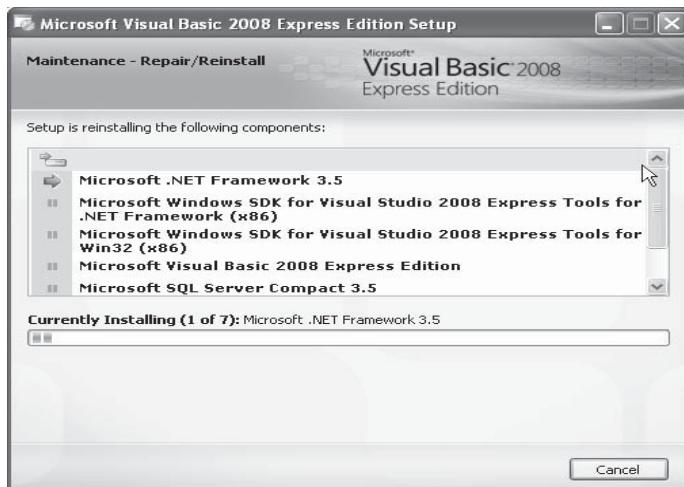
Gambar 1.3 Install Visual Basic 2008 pada folder

Klik Next untuk untuk melanjutkan proses instalasi bahasa pemrograman Visual Basic 2008.



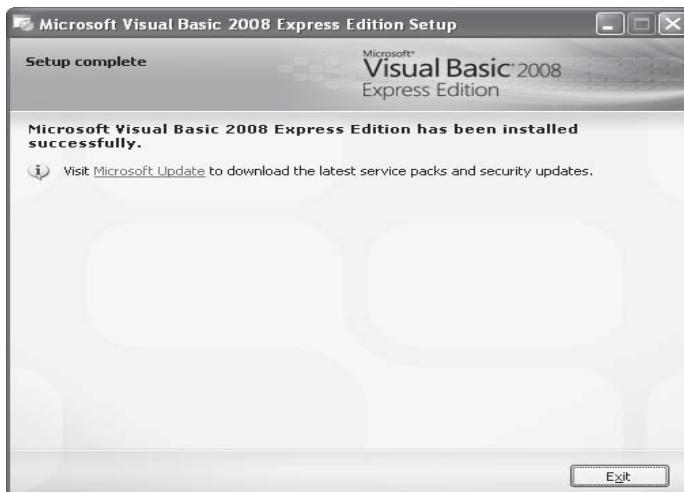
Gambar 1.4 Welcome to setup Visual Basic 2008

Tunggu proses berikut sampai dengan selesai.



Gambar 1.5 Intallation Progress

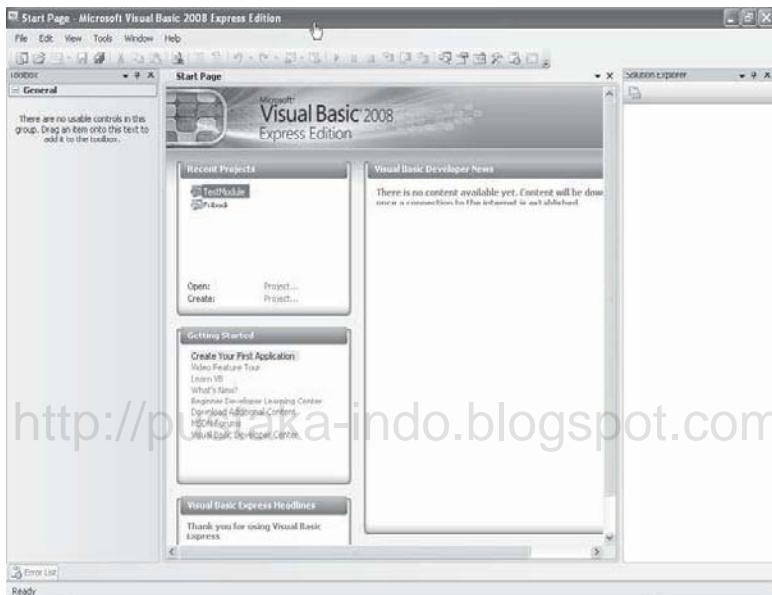
Apabila sudah selesai maka akan tampil gambar berikut ini:



Gambar 1.6 Setup Complete

1.2 Menjalankan Aplikasi

Untuk menjalankan aplikasi Visual Basic 2008, hampir sama dengan cara menjalankan aplikasi lain pada komputer Anda. Klik Start > Program > Microsoft Visual Basic 2008. Lalu akan tampil form berikut ini:

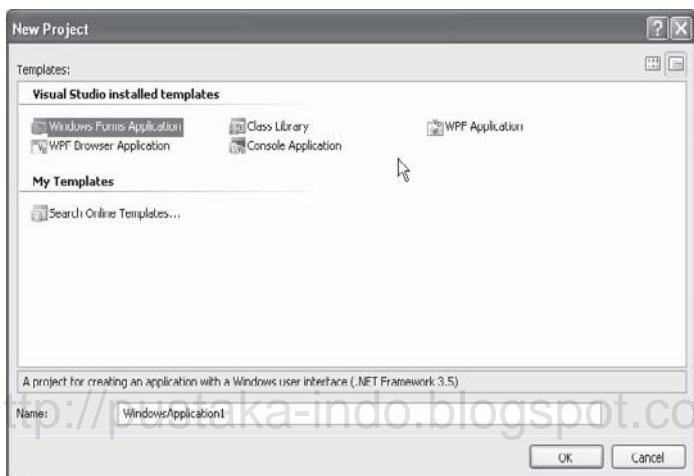


Gambar 1.7 Start Page Microsoft Visual Basic 2008

Untuk memulai membuat program, Anda dapat mengklik menu File lalu klik New Project. Kemudian klik **Windows Form Application** lalu klik tombol OK. Selain menggunakan menu File, Anda juga dapat mengklik Create Project yang terdapat pada bagian kiri tampilan form (Recent Project).

Anda juga dapat membuka aplikasi dengan perintah Open pada Recent Project lalu pilih nama project atau aplikasi yang akan Anda buka kemudian klik Open.

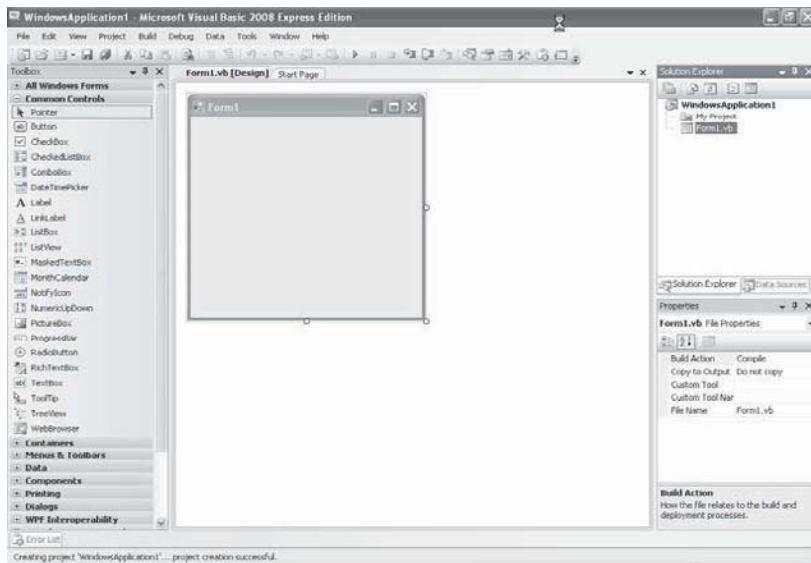
Atau Anda dapat mengklik perintah **Create Your First Application** untuk membuat aplikasi Anda yang pertama. Semua perintah tersebut digunakan untuk membuat aplikasi yang baru tetapi menggunakan cara-cara yang berbeda. Anda dapat memilih salah satu perintah saja untuk membuat aplikasi baru, tentu saja perintah yang Anda anggap mudah dan cepat.



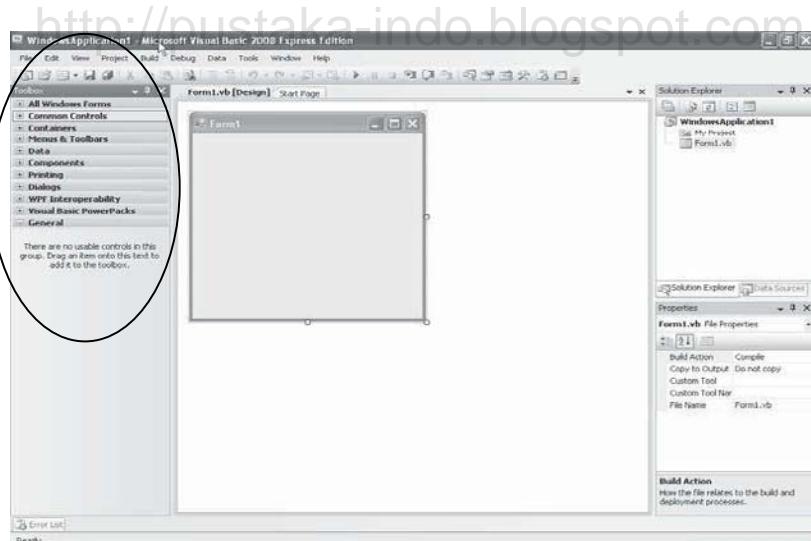
Gambar 1.8 New Project Visual Basic 2008

Untuk membuat nama aplikasi atau nama project, Anda dapat membuatnya pada Name lalu klik OK. Nama file project disesuaikan dengan nama aplikasi yang akan Anda buat. Tampilan awal seperti Gambar 1.9.

Untuk membuat aplikasi ada beberapa perlengkapan yang Anda butuhkan, yaitu ToolBox sebagai alat atau komponen untuk membuat program. ToolBox yang terdapat pada Visual Basic 2008 adalah All Windows Form yang menampilkan semua komponen, Common Control, Container, Menus & Toolbars, Data, Component, Printing, Dialog, dan General. Perhatikan Gambar 1.10.

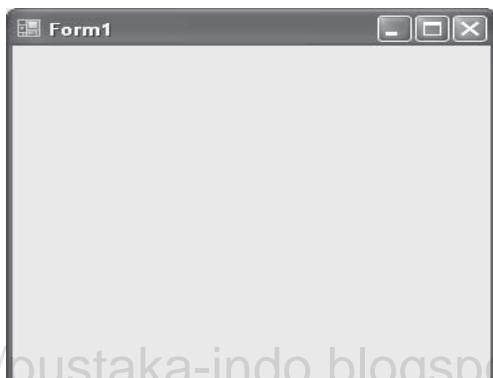


Gambar 1.9 Tempat membuat aplikasi



Gambar 1.10 ToolBox Visual Basic 2008

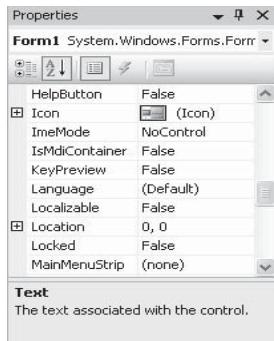
Tempat membuat aplikasi dan meletakkan komponen yang dibutuhkan dalam aplikasi disebut dengan form. Form juga digunakan untuk merancang tampilan program aplikasi yang akan Anda buat. Pada form terdapat ikon Minimize, Maximize, dan Close. Apabila Anda mengklik form, maka akan tampil titik *corner* yang dapat Anda gunakan untuk memperbesar dan memperkecil lebar form sesuai dengan keinginan Anda.



Gambar 1.11 Form

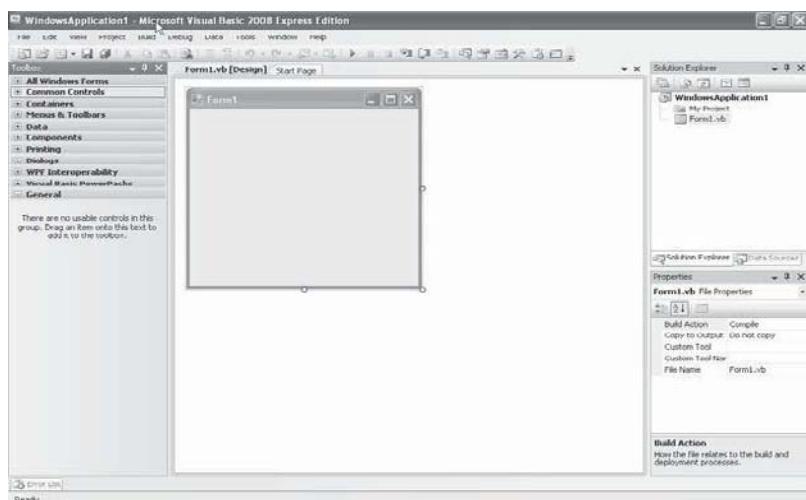
Properties digunakan untuk menuliskan/mengatur form dan komponen yang berisi perintah pelengkap dan pengatur aplikasi yang Anda buat. Dengan properties Anda dapat mengatur warna tulisan, membuat tulisan dengan tebal, miring, atau bentuk lain yang Anda inginkan. Properties juga digunakan sebagai petunjuk perintah yang akan Anda gunakan untuk membuat program dan untuk memanipulasi komponen yang terdapat di dalam form.

Selain Properties juga terdapat Solution Explorer yang digunakan untuk menampung informasi project, form, dan komponen yang aktif pada saat itu. Pada Solution Explorer juga terdapat data source yang digunakan untuk membuat dan mengolah data dengan berbagai jenis database.



Gambar 1.12 Properties

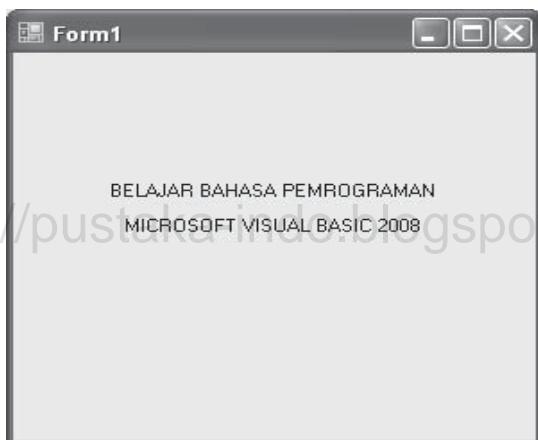
Jadi untuk membuat program menggunakan Visual Basic 2008 digunakan toolbox yang berisi komponen yang akan Anda gunakan untuk membuat program, kemudian komponen yang sudah Anda pilih tersebut diletakkan pada form untuk membentuk rancangan program aplikasi yang Anda inginkan. Setelah komponen diletakkan pada form, Anda akan mengatur properties dari komponen yang telah diletakkan pada form aplikasi.



Gambar 1.13 ToolBox, Form, dan Properties

Contoh di bawah ini menunjukkan cara membuat aplikasi dengan langkah yang dibahas. Pilih komponen (misalnya Anda pilih komponen label) lalu Anda klik mouse, pada form selanjutnya isi properties text dengan tulisan “BELAJAR BAHASA PEMROGRAMAN”. Masukkan kembali komponen label lalu klik di form dan isi kembali properties text dengan tulisan “MICROSOFT VISUAL BASIC 2008”.

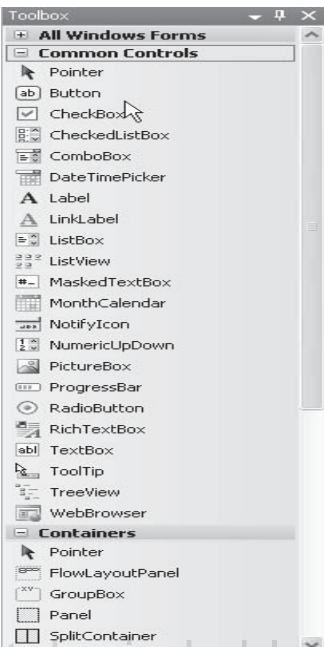
Apabila sudah selesai mem-buat teks, Anda dapat menjalankan program tersebut dengan menekan tombol F5 atau pilih ikon Start Debuging, maka akan tampak seperti gambar berikut:



Gambar 1.14 Perintah dasar pemrograman Visual Basic 2008

1.3 Komponen Visual Basic 2008

Komponen Visual Basic 2008 terdapat di dalam toolbox. Komponen ini digunakan sebagai alat-alat untuk membuat program. ToolBox tersebut diantaranya adalah pointer, button, checkBox, CheckedListBox, ComboBox, DateTimePicker, Label, ListBox, PictureBox, dan sebagainya.



<http://pusataindo.blogspot.com>

Gambar 1.15 ToolBox

1.4 Membuat Perintah

Untuk membuat perintah Microsoft Visual Basic 2008 yaitu dengan cara mengklik ganda komponen. Misalnya Anda ingin membuat perintah untuk komponen button, lakukan klik ganda komponen button, maka akan tampil editor dan terlihat perintah berikut:

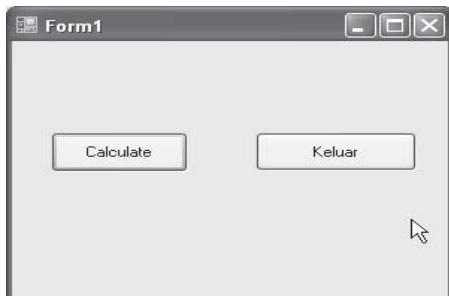
```
Public Class Form1  
  
    Private Sub Button1_Click(ByVal sender As System.Object,  
        ByVal e As System.EventArgs) Handles Button1.Click  
  
        '<Tempat membuat perintah>  
    End Sub  
  
End Class
```

Public class form adalah kumpulan prosedur perintah yang dikumpulkan dalam satu class. Private Sub Button1_Click adalah perintah yang harus dilakukan apabila button 1 (satu) diklik. Perintah Visual Basic 2008 dibuat terletak di antara *private sub* dan *end sub*. Tempat untuk membuat perintah tersebut disebut dengan editor.



Gambar 1.16 Editor perintah

Berikut contoh perintah dasar dan teknik menuliskan perintah pada komponen button.



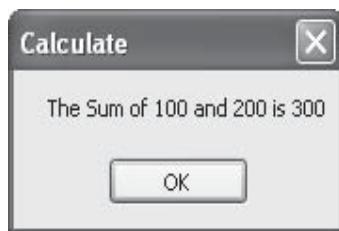
Gambar 1.17 Menulis perintah button

Masukkan dua komponen button dengan teks calculate dan teks keluar. Untuk memasukkan komponen button, klik toolbox Common Control lalu klik button, Selanjutnya klik ke form lalu isi properties teks dengan tulisan *calculate*.

Masukkan kembali komponen button yang kedua dengan cara yang sama dengan teks keluar. Button calculate akan berisi perintah untuk menghitung penjumlahan dua bilangan yang dibuat dengan variabel, sedangkan button keluar akan berisi perintah untuk keluar dari aplikasi. Untuk membuat perintah, klik ganda button calculate lalu ketikkan perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim num1, num2, sum As Single
    num1 = 100
    num2 = 200
    sum = num1 + num2
    MsgBox(" The Sum of " & num1 & " and " & num2 & " is " & sum)
End Sub
```

Perintah dim digunakan untuk menampung data dalam variabel num1, num2, sum dengan tipe data single. Kemudian num1 diisi dengan nilai 100, variabel num2 diisi dengan nilai 200, dan variabel sum diisi dengan nilai variabel num1 ditambah dengan nilai variabel num2. Perintah MsgBox digunakan untuk menampilkan kotak dialog yang akan menampilkan isi variabel setelah mengklik button.



Gambar 1.18 Gambar hasil aplikasi

Buat perintah untuk button keluar dengan cara mengklik ganda button keluar lalu buat perintah berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    End
End Sub
```

Perintah End akan diproses apabila Anda menekan button keluar. Jadi teknik atau cara untuk membuat perintah pada Visual Basic yaitu dengan cara mengklik ganda komponen, kemudian buat perintah yang akan digunakan. Dalam membuat perintah program terdapat dua cara, yaitu dengan Method dan Event. Method adalah perintah yang sudah tersedia di dalam suatu object yang terdapat di dalam sebuah komponen. Method dibuat dengan menggunakan pemisah tanda titik (.) pada saat membuat perintah. Misalnya perintah untuk menampilkan tulisan pada label dibuat dengan dengan perintah:

```
Label1.Text = "Belajar Visual Basic 2008"
```

Tulisan Text pada perintah label1 merupakan method dari komponen label. Biasanya method tersebut dapat Anda lihat pada properties Visual Basic. Selain dengan method perintah, juga bisa dibuat menggunakan event. Event adalah perintah yang dibuat berdasarkan kejadian pada saat mengklik atau menekan suatu tombol-tombol yang terdapat pada keyboard.



BERMAIN DENGAN PROGRAM SEDERHANA

<http://pustaka-indo.blogspot.com>

Untuk melatih dan mengenal perintah program Visual Basic 2008, Anda harus banyak bermain-main dengan komponen dan perintah yang akan digunakan dalam pembuatan aplikasi. Dengan berlatih membuat program, maka kita akan terbiasa dengan logika dan perintah-perintah yang sering digunakan dalam aplikasi.

2.1 Menghitung Luas Objek

Program untuk menghitung luas menggunakan rumus panjang dikalikan dengan lebar, berarti terdapat dua input, yaitu panjang dan lebar, serta satu output yaitu luas. Rancanglah form berikut untuk membuat aplikasi tersebut di atas.



Gambar 2.1 Desain program mencari luas

Untuk membuat program mencari luas di atas, klik menu **File > New Project > Windows News Application**. Selanjutnya klik komponen label lalu klik ke form dan isi properties text dengan *Panjang*. Masukkan textbox untuk *panjang* dengan cara klik komponen textbox dan klik komponen tersebut di sebelah label *panjang*.

Buat label *lebar* dengan cara mengklik komponen *lebar* dan klik ke form lalu isi properties text dengan *lebar*. Masukkan kembali textbox untuk *lebar* dengan cara mengklik komponen textbox lalu klik di sebelah label *lebar*. Buat label *luas* dengan cara mengklik komponen label kemudian klik ke form dan isi properties text dengan *luas*.

Masukkan komponen textbox untuk *luas* dan letakkan di sebelah label *luas*. Masukkan Button *Hitung luas* dengan cara mengklik komponen button lalu klik ke form dan isi properties text dengan tulisan *hitung luas*. Masukkan kembali komponen button untuk *Keluar* dengan cara yang sama dengan langkah di atas. Klik ganda button *hitung luas* lalu buat perintah program sebagai berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text)
End Sub
```

TextBox3.text adalah nama komponen untuk luas, TextBox1.text adalah nama komponen untuk input panjang, dan TextBox2.text adalah nama komponen untuk input lebar. Val adalah perintah untuk mengubah tipe data text menjadi angka, Val(TextBox1.Text) berarti mengubah data yang terdapat pada TextBox1 menjadi angka.

Apabila Anda sudah selesai membuat perintah untuk button hitung luas, maka Anda buat perintah untuk button keluar dengan cara mengklik ganda button keluar lalu buat perintah sebagai berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    End
End Sub
```

Perintah program lengkap menghitung luas objek tersebut adalah sebagai berikut:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text)
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub
End Class
```

Untuk menjalankan aplikasi yang sudah Anda buat, tekan tombol F5 lalu masukkan data untuk panjang, lebar, lalu klik tombol/button hitung luas.



Gambar 2.2 Hasil program menghitung luas

Apabila Anda ingin mengakhiri penggunaan aplikasi menghitung luas objek di atas, maka Anda klik button keluar. Apabila aplikasi sudah berjalan dengan baik, maka Anda dapat meng-compile program yang sudah Anda buat dengan cara mengklik menu Build, kemudian klik Build Desktop Windows dan tunggu proses compile selesai.

Ciri compile sudah selesai terdapat pada sudut kiri bawah form yang ber tuliskan *build succeeded* (compile sukses). Simpanlah program yang sudah Anda buat dengan cara mengklik menu file kemudian pilih Save all dan buatlah nama form, nama project, dan Anda tentukan folder lokasi penyimpanan file.

2.2 Operasi Bilangan

Program operasi bilangan adalah program aplikasi yang menggunakan operasi penjumlahan, perkalian, pembagian, dan pengurangan dua bilangan. Untuk membuat program operasi bilangan tersebut buatlah rancangan form berikut ini:



Gambar 2.3 Rancangan program operasi bilangan

Untuk membuat program operasi bilangan, klik menu File > New Project > Windows New Application. Klik Form lalu isi properties text dengan tulisan operasi bilangan. Klik ToolBox Container lalu pilih komponen GroupBox dan klik ke form. Isi properties text dengan penjumlahan. Masukkan komponen label Angka 1, label angka 2, dan hasil.

Selanjutnya masukkan juga komponen textbox sebanyak tiga buah untuk menampung data angka pertama, angka kedua, dan hasil. Buat kembali Groupbox untuk operasi pengurangan, perkalian dan pembagian dengan label dan textbox yang sama. Apabila sudah selesai, buat perintah untuk operasi penjumlahan dengan mengklik ganda textbox2 yang terletak di sebelah label angka ke-2 pada groupbox penjumlahan, kemudian ketik perintah program di bawah ini:

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged  
  
    TextBox3.Text = Val(TextBox1.Text) +  
    Val(TextBox2.Text)  
  
End Sub
```

Untuk membuat perintah pada groupbox pengurangan, hampir sama dengan cara yang telah kita lakukan terhadap groupbox penjumlahan, yaitu mengklik ganda textbox5 yang terletak di sebelah angka ke-3 pada groupbox pengurangan, kemudian ketik perintah berikut:

```
Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged  
    TextBox3.Text = Val(TextBox1.Text) -  
        Val(TextBox2.Text)  
End Sub
```

Sebenarnya perintah yang digunakan untuk operasi bilangan penjumlahan, pengurangan, perkalian dan pembagian adalah sama. Perbedaannya terletak pada tanda yang digunakan dalam melakukan operasi bilangan. Jalankan program yang Anda buat dengan menekan F5 lalu masukkan angka-angka yang akan dibuat operasi bilangan.



Gambar 2.4 Hasil program operasi bilangan

Listing lengkap program operasi bilangan:

```
Public Class Form1

    Private Sub GroupBox1_Enter(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles GroupBox1.Enter
    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    TextBox2.TextChanged
        TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
    End Sub

    Private Sub TextBox5_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    TextBox5.TextChanged
        TextBox4.Text = Val(TextBox6.Text) -
            Val(TextBox5.Text)
    End Sub

    Private Sub TextBox11_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    TextBox11.TextChanged
        TextBox10.Text = Val(TextBox11.Text) *
            Val(TextBox12.Text)
    End Sub

    Private Sub TextBox8_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    TextBox8.TextChanged
        TextBox7.Text = Val(TextBox9.Text) / Val(TextBox8.Text)
    End Sub

End Class
```

2.3 Membuat Jam Digital

Untuk membuat sebuah program jam digital, Anda buka dulu form baru dengan cara mengklik **File > New Project > Windows Form Application**. Buat nama project (Jam Digital) dan klik OK. Klik Form lalu isi Properties Text dengan tulisan “Jam Digital”, kemudian masukkan komponen label dan komponen timer ke dalam form. Klik komponen Timer lalu isi properties interval dengan angka 1000, pada bagian Enable dibuat true kemudian klik ganda komponen timer dan buat perintah berikut:

```
Public Class Form1
Dim TestStr As String

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick
    TestStr = Format(Now(), "Long Time")
    Label1.Text = TestStr
End Sub
End Class
```

Atur posisi label di tengah form dan buat dengan font yang lebih besar dengan cara mengklik label lalu klik properties font. Pilih jenis huruf dengan style bold dan size 18. Atur pula posisi form di tengah-tengah layar monitor dengan cara mengklik form kemudian isi properties Start-Position dengan CenterScreen. Jalankan program dengan menekan F5.



Gambar 2.5 Hasil program jam digital

Dim TestStr digunakan untuk menampung data dalam bentuk string ke dalam variabel memori. Perintah Format digunakan untuk membuat format tanggal, jam, dan bilangan. Untuk format tanggal menggunakan simbol G yang berarti format tanggal dan waktu secara umum dengan bentuk tampilan tanggal 4/3/93 05:34 PM. Bentuk format tanggal yang lain adalah Long Date, short Date, Long Time, short Time. Berikut ini contoh penggunaan format dalam bentuk perintah program.

```

Dim TestDateTime As Date = #1/27/2001 5:04:23 PM#
Dim TestStr As String
TestStr = Format(Now(), "Long Time")
TestStr = Format(Now(), "Long Date")
TestStr = Format(Now(), "D") ' Hasil "5:4:23".
TestStr = Format(TestDateTime, "h:m:s")' Hasil "05:04:23 PM".
TestStr = Format(TestDateTime, "hh:mm:ss tt")
' Hasil output "Saturday, Jan 27 2001".

TestStr = Format(TestDateTime, "dddd, MMM d yyyy")
' Hasil output "17:04:23".

TestStr = Format(TestDateTime, "HH:mm:ss")
' Hasil output "23".

TestStr = Format(23)

' Berikut ini contoh penggunaan format numeric / angka
' Hasil output "5,459.40".
TestStr = Format(5459.4, "##,##0.00")

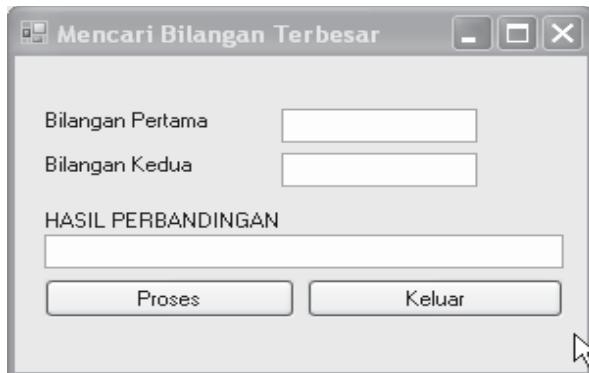
' Hasil output "334.90".
TestStr = Format(334.9, "###0.00")

' Hasil output "500.00%".
TestStr = Format(5, "0.00%")

```

2.4 Mencari Bilangan Terbesar

Program mencari bilangan terbesar digunakan untuk membandingkan dua (dua) nilai bilangan. Untuk membuat program ini Anda masukkan dulu label bilangan pertama, label bilangan kedua, dan label hasil perbandingan. Masukkan juga textBox untuk bilangan pertama, kedua, dan textbox hasil perbandingan bilangan. Masukkan dua buah button proses dan button keluar seperti tampak pada rancangan berikut ini:



Gambar 2.6 Rancangan program mencari bilangan terbesar

Buatlah judul form dengan cara mengklik form lalu isi properties text dengan tulisan *Mencari Bilangan Terbesar*. Lalu buatlah perintah untuk button keluar sebagai berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles Button2.Click  
End Sub
```

Untuk membuat perintah pada button proses yaitu dengan cara mengklik ganda button Proses dan buatlah perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles Button1.Click  
If Val(TextBox1.Text) > Val(TextBox2.Text) Then  
    TextBox3.Text = "Bilangan pertama lebih besar"  
End If  
  
If Val(TextBox1.Text) = Val(TextBox2.Text) Then  
    TextBox3.Text = "Bilangan pertama sama dengan  
    bilangan kedua"  
End If  
  
If Val(TextBox1.Text) < Val(TextBox2.Text) Then  
    TextBox3.Text = "Bilangan kedua lebih besar"  
End If  
End Sub
```

Perintah if digunakan untuk membandingkan antara dua variabel bilangan pertama dengan bilangan kedua. Ada tiga kemungkinan dari perbandingan antara dua bilangan tersebut, yaitu bilangan pertama lebih besar, bilangan pertama dan bilangan kedua sama besar, atau bilangan kedua lebih besar. Perintah val digunakan untuk mengkonversi tipe data string menjadi angka, sedangkan untuk membandingkan dua bilangan tersebut menggunakan tanda lebih besar ($>$), tanda lebih kecil ($<$), atau tanda sama dengan ($=$).



Gambar 2.7 Hasil program mencari bilangan terbesar

2.5 Mencari Rata-Rata

Program mencari nilai rata-rata didapat dari total data lalu dibagi dengan banyak data. Untuk membuat program mencari nilai rata-rata hampir sama dengan membuat program operasi bilangan, perbedaan program rata-rata hanya pada penggunaan dua tanda operasi bilangan, yaitu tambah dan bagi dalam satu operasi bilangan.

Buka aplikasi baru dengan cara klik menu File > New Project > Windows News Application. Buat nama project “Program Rata-Rata” dan klik button OK.

Masukkan empat komponen label, masing-masing untuk label bilangan ke satu, label bilangan ke dua, label bilangan ke tiga, dan label hasil rata-rata. Masukkan pula empat komponen TextBox masing-masing untuk textbox bilangan ke satu, bilangan ke dua, bilangan ke tiga, dan textbox untuk menampung rata-rata. Masukkan satu komponen button untuk keluar.



<http://pustaka.indo.blogspot.com>

Untuk membuat perintah program, klik ganda textbox3 lalu Anda buat perintah program berikut ini:

```
Public Class Form1
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged
    TextBox4.Text = (Val(TextBox1.Text) +
    Val(TextBox2.Text) + Val(TextBox3.Text)) / 3
End Sub
End Class
```

Klik ganda pada button Keluar untuk membuat perintah untuk menutup atau keluar dari form program rata-rata. Perintah untuk button keluar adalah sebagai berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
End
End Sub
```

Jalankan program yang sudah Anda buat dengan menekan tombol F5. Pada bagian terdahulu Anda membuat proses perhitungan menggunakan button, akan tetapi pada bagian ini Anda langsung dapat melihat hasil perhitungan rata-rata tanpa membuat tombol Proses.

Dengan klik ganda pada posisi terakhir input data angka, kita dapat membuat proses operasi perhitungan secara langsung. Private Sub TextBox3_TextChanged berarti perintah akan diproses apabila textbox3 terjadi perubahan. Itu berarti ketika Anda menginputkan data angka pada textbox3, maka perintah untuk mencari rata-rata akan segera diproses dan meletakkan hasil perhitungan rata-rata pada textbox4.

2.6 Password

Password digunakan untuk mencegah orang lain masuk ke dalam program yang kita buat, kecuali orang tersebut mempunyai hak akses berupa kata sandi yang benar untuk masuk dan menjalankan program. Berikut ini adalah dasar-dasar untuk membuat program password, rancanglah form program password berikut ini:



Gambar 2.9 Rancangan program password

Masukkan dua buah label masing-masing untuk label user dan label password, kemudian masukkan pula dua buah textbox, masing-masing untuk textbox user dan textbox password. Kemudian masukkan dua buah button, yaitu button masuk dan button keluar.

Klik textbox password lalu isi properties passwordChar dengan tanda asterisk (*), tanda asterisk tersebut digunakan untuk menampilkan tanda * pada saat Anda mengisi password. Klik form dan isi properties Start-Position dengan CenterScreen agar pada saat program dijalankan langsung pada posisi di tengah layar monitor. Klik ganda button Keluar lalu ketik perintah berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End
End Sub
```

Klik ganda button Masuk dan buat perintah untuk menyeleksi password yang diketik pemakai program sebagai berikut:

```
Public Class Form1

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub

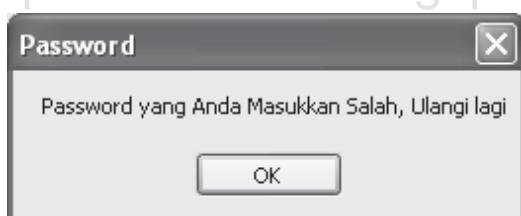
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Pass As String = "AKMI"
        If TextBox2.Text = Pass Then
            MsgBox("Password yang Anda Masukkan Benar")
        End
        Else
            MsgBox("Password yang Anda Masukkan Salah, Ulangi lagi")
            TextBox2.Focus()
        End If
    End Sub
End Class
```

Jalankan program dengan mengklik menu debug lalu pilih sub menu start debugging (F5) sehingga tampil output program password berikut:



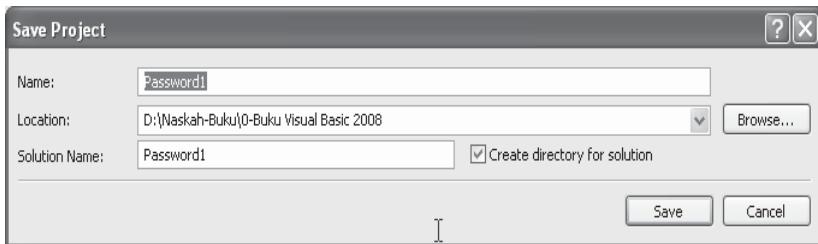
Gambar 2.10 Output program password

Password yang dibuat dalam perintah program adalah “AKMI”, terlihat dari perintah program Dim Pass As String = “AKMI”. Kemudian perintah akan dicek, apabila isi textbox2 sama dengan isi variabel pass, maka akan muncul pesan “Password yang Anda masukkan benar”. Kalau salah, maka akan muncul pesan “Password yang Anda masukkan salah, ulangi lagi”.



Gambar 2.11 Message box salah password

Simpanlah program yang Anda buat dengan cara mengklik ikon Save All lalu tentukan nama project dan lokasi folder penyimpanan. Klik button Save.



Gambar 2.12 Save Project

2.7 Cek Bilangan Ganjil Genap

Program untuk mengecek bilangan ganjil dan genap sebenarnya adalah program yang membagi angka bilangan dengan dua. Apabila bilangan tersebut habis dibagi dengan dua, maka bilangan yang kita masukkan adalah bilangan genap, dan sebaliknya adalah bilangan ganjil. Rancanglah form berikut ini untuk membuat program cek bilangan ganjil genap.



Gambar 2.13 Rancangan program cek bilangan genap ganjil

Klik ganda button proses untuk membuat perintah pengecekan input bilangan ganjil atau bilangan genap, kemudian ketiklah perintah program sebagai berikut:

```
Public Class Form1  
  
    Private Sub Button1_Click(ByVal sender As System.Object,  
        ByVal e As System.EventArgs) Handles Button1.Click  
        TextBox2.Text = IIf(Val(TextBox1.Text) Mod 2 = 1,  
            "Bilangan Ganjil", "Bilangan Genap")  
    End Sub  
End Class
```

Jalankan program dengan mengklik menu debug lalu pilih sub menu start debugging (F5) sehingga tampil output program password berikut:



<http://pustaka-indo.blogspot.com>
Gambar 2.14 Output program cek ganjil genap



BERMAIN DENGAN PERINTAH STRING MODULE

<http://pustaka-indo.blogspot.com>

Pengolahan string sangat penting dalam sebuah program, karena dengan string, program dapat dibuat dengan perintah yang singkat. Format digunakan untuk mengubah satu bentuk data ke bentuk data yang lain, misalnya dari data angka diubah ke dalam bentuk data string. Format juga digunakan untuk mengubah bentuk tampilan data satu ke bentuk tampilan data yang lain sesuai dengan simbol yang terdapat dalam perintah format tersebut.

3.1 Menggabungkan String

Untuk menggabungkan string dalam Microsoft Visual Basic 2008 menggunakan tanda & (dan), data yang digabungkan harus dalam bentuk data string. Apabila data tersebut dalam bentuk angka, tanggal atau jam, maka data tersebut harus diubah dulu dalam bentuk string. Masukkan dua buah label, yaitu label nama dan label hobi, kemudian masukkan pula dua buah textbox masing-masing untuk menampung data nama dan hobi. Masukkan satu button untuk proses hasil penggabungan string. Berikut ini contoh program menggabung string dengan string.

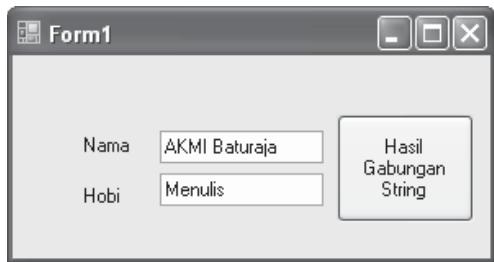


Gambar 3.1 Rancangan program penggabungan string

Klik ganda button hasil gabungan string lalu buat perintah program sebagai berikut:

```
Public Class Form1  
  
    Private Sub Button1_Click(ByVal sender As System.Object,  
        ByVal e As System.EventArgs) Handles Button1.Click  
        MsgBox("Hallo: " & TextBox1.Text & " Hobi anda adalah"  
        & CStr(TextBox2.Text) & ".")  
  
    End Sub  
End Class
```

Jalankan program yang sudah Anda buat perintahnya dengan menekan tombol F5.



Gambar 3.2 Output program penggabungan string

Setelah button hasil gabungan string diklik, maka akan muncul tampilan kotak pesan sebagai berikut:



Gambar 3.3 Kotak pesan hasil penggabungan string

Apabila program penggabungan string tersebut berupa data angka, maka kita harus mengubahnya ke dalam bentuk string dengan menggunakan perintah CStr seperti program berikut ini:



Gambar 3.4 Rancangan program penggabungan string dengan angka

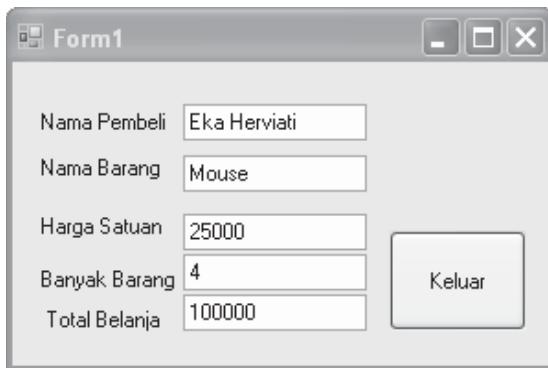
Untuk membuat program di atas, masukkan lima buah label untuk label nama pembeli, label nama barang, label harga satuan, label banyak barang, dan label total belanja serta satu button untuk keluar. Untuk membuat perintah program, klik ganda textbox Banyak Barang lalu buat perintah sebagai berikut:

```
Public Class Form1  
  
Private Sub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged  
    TextBox5.Text = Val(TextBox3.Text) *  
        Val(TextBox4.Text)  
    MsgBox("Hallo: " & TextBox1.Text & _  
        " Total Belanja Anda " & _  
        CStr(TextBox5.Text) & ".")  
End Sub  
End Class
```

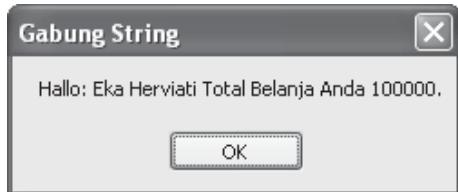
Kemudian klik ganda button Keluar dan buat perintah program berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
    End  
End Sub
```

Jalankan program dengan mengklik menu debug lalu pilih sub menu *start debugging* atau tekan tombol F5.



Gambar 3.5 Hasil program penggabungan string dengan angka



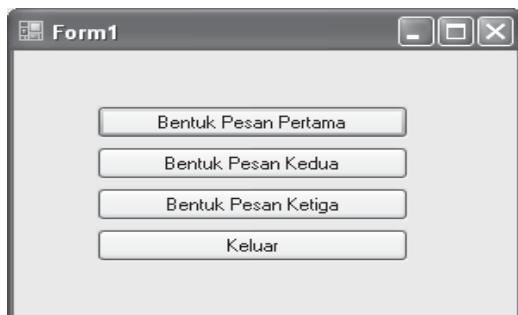
Gambar 3.6 Message box string dan angka

3.2 Membuat Pesan

Untuk membuat sebuah tampilan pesan dengan menggunakan Visual Basic 2008, kita menggunakan perintah MsgBox dengan format sebagai berikut:

```
Public Function MsgBox( _
    ByVal Prompt As Object, _
    Optional ByVal Buttons As MsgBoxStyle = MsgBoxStyle.OKOnly, _
    Optional ByVal Title As Object = Nothing _
) As MsgBoxResult
```

Prompt adalah tempat meletakkan pesan yang akan ditampilkan, Button adalah tombol-tombol yang akan digunakan, sedangkan title adalah judul dari pesan yang akan ditampilkan. Jadi secara garis besar perintah Msgbox tersebut adalah MsgBox (Peser, Tombol, Judul Pesan). Namun pesan tersebut dapat kita buat dengan perintah MsgBox(pesan) saja.



Gambar 3.7 Desain program pesan

Masukkan empat buah button, masing-masing untuk button bentuk pesan pertama, bentuk pesan kedua, bentuk pesan ketiga, dan button keluar. Klik ganda button bentuk pesan pertama untuk membuat perintah MsgBox yang sangat sederhana lalu ketik perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    MsgBox("Ini hanya berisi pesan saja (tanpa tombol dan title)")
End Sub
```

Klik ganda button bentuk pesan kedua untuk membuat pesan yang memiliki button dengan cara mengetikkan perintah berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim nPesan As MsgBoxResult
    nPesan = MsgBox("Ini hanya berisi pesan saja (tanpa tombol dan title)", MsgBoxStyle.OkCancel)
    'Apabila sudah menggunakan button maka harus pesan 'harus ditampung dengan variabel
    If nPesan = MsgBoxResult.Ok Then
        MsgBox("Anda memilih tombol OK")
    End If
    If nPesan = MsgBoxResult.Cancel Then
        MsgBox("Anda memilih tombol Cancel")
    End If
End Sub
```

Untuk membuat perintah button yang lengkap klik ganda button bentuk pesan ketiga lalu ketik perintah berikut:

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Dim msg As String
    Dim title As String
    Dim style As MsgBoxStyle
    Dim response As MsgBoxResult
    msg = "Apakah Anda ingin lanjut ?"
    style = MsgBoxStyle.DefaultButton2 Or _
        MsgBoxStyle.Critical Or MsgBoxStyle.YesNo
    title = "Demo Pesan MsgBox"

    'Bentuk MsgBoxnya sebagai berikut:
    response = MsgBox(msg, style, title)

    If response = MsgBoxResult.Yes Then
        ' User chose Yes.
        MsgBox("Anda memilih tombol Yes")
    Else
```

```
    MsgBox("Anda memilih tombol No")
End If
End Sub
```

Jalankan program dengan mengklik menu debug lalu pilih sub menu *start debugging* atau tekan tombol F5.

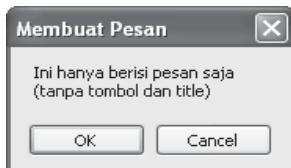


Gambar 3.8 Membuat pesan dengan MsgBox

Bentuk button dapat pula kita buat dengan menggunakan nilai angka. Nilai 0 untuk button OKOnly, Nilai 1 untuk button OKCancel, nilai 2 untuk button AbortRetryIgnore, nilai 3 untuk button YesNoCancel, nilai 4 untuk button YesNo, nilai 5 untuk button RetryCancel, nilai 16 untuk ikon Critical, nilai 32 untuk ikon Question, nilai 48 untuk ikon Exclamation, nilai 64 untuk ikon Information, nilai 0 untuk DefaultButton1, nilai 255 untuk DefaultButton2, nilai 512 untuk DefaultButton3, nilai 4096 untuk SystemModal, nilai 65536 untuk MsgBoxSetForeground, dan nilai 524588 untuk MsgBoxRight.

Perintah MsgBox juga dapat dikombinasikan dengan simbol chr(13) dan chr(10) untuk membuat baris pesan lebih dari satu baris. Berikut contoh pesan yang menggunakan line feed chr(10).

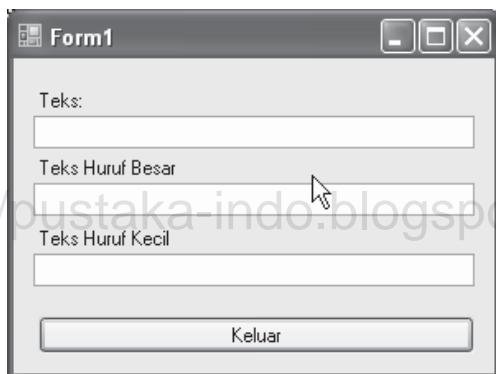
```
nPesan = MsgBox("Ini hanya berisi pesan saja" & Chr(10) &
"(tanpa tombol dan title)", MsgBoxStyle.OkCancel)
```



Gambar 3.9 Pesan dengan Line Feed

3.3 Teks Huruf Kecil dan Besar

Konversi huruf dalam suatu program sangat dibutuhkan, terutama pada saat kita ingin memasukkan kode yang dibuat harus dengan huruf besar atau kecil. Untuk membuat huruf besar tanpa menekan tombol Shift atau Caps Lock pada keyboard yaitu dengan menggunakan perintah Ucase dan Lcase. Perintah Ucase digunakan untuk mengubah teks ke dalam bentuk huruf besar, sedangkan Lcase digunakan untuk mengubah jenis teks ke dalam bentuk huruf kecil. Berikut ini contoh penggunaan perintah tersebut dalam suatu program.



Gambar 3.10 Desain program teks huruf kecil

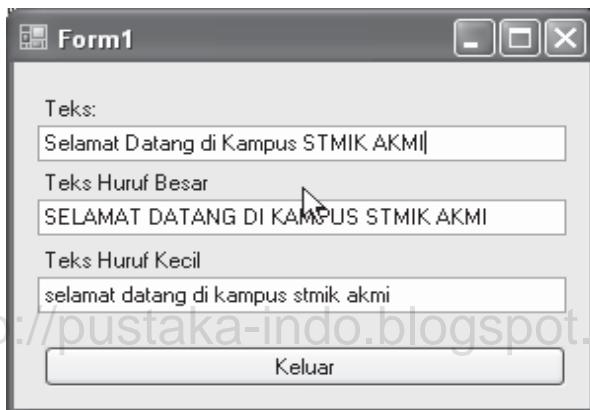
Masukkan tiga buah label, masing-masing untuk label teks, label teks huruf besar, dan label teks huruf kecil. Masukkan tiga buah textbox untuk menampung teks huruf besar dan kecil, serta satu buah komponen button keluar seperti terlihat pada gambar di atas. Klik ganda textbox teks kemudian buat perintah program berikut:

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    TextBox2.Text = UCASE(TextBox1.Text)
    TextBox3.Text = LCASE(TextBox1.Text)
End Sub
```

Kemudian klik ganda komponen button keluar lalu ketik perintah program berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End
End Sub
```

Jalankan program dengan menekan tombol F5 atau klik menu debug lalu pilih submenu *start debugging*.



Gambar 3.11 Hasil program teks huruf kecil

Berikut ini listing (source code) lengkap program di atas:

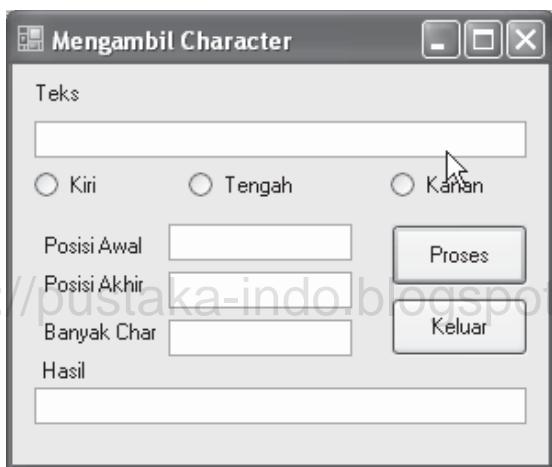
```
Public Class Form1

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
        TextBox2.Text = UCASE(TextBox1.Text)
        TextBox3.Text = LCASE(TextBox1.Text)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End
End Sub
End Class
```

3.4 Mengambil Character

Character yang akan diambil dalam bahasa pemrograman Visual Basic 2008 yaitu mulai dari posisi karakter dari kiri, kanan, tengah. Untuk mengambil karakter menggunakan perintah Left (kiri), Right (kanan), Mid (tengah). Untuk menghitung panjang karakter dalam suatu komponen atau variabel menggunakan perintah Len. Berikut ini adalah contoh program untuk mengambil character.



Gambar 3.12 Rancangan program mengambil character

Untuk merancang form di atas dibutuhkan komponen label teks, label posisi awal, label posisi akhir, label banyak character, dan label hasil. Kemudian masukkan komponen textbox untuk teks, textbox untuk input posisi awal, textbox untuk input posisi akhir, textbox untuk input banyak character, dan textbox untuk output hasil. Masukkan pula Radiobutton kiri, radiobutton kanan, dan radiobutton tengah. Kemudian masukkan button proses dan button keluar. Untuk membuat perintah, klik ganda button Proses kemudian buat perintah berikut ini:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
    If RadioButton1.Checked = True Then
        'Posisi Awal
        TextBox2.Text = "1"
        'Posisi Akhir sama dengan banyak character yang
        'diambil dari kiri
        TextBox3.Text = TextBox4.Text
        'Hasil yang ditampilkan pada textbox5
        TextBox5.Text =
        Microsoft.VisualBasic.Left(TextBox1.Text,
        Val(TextBox4.Text))
        ' Ambil character dari kiri pada textbox1
        sebanyak nilai yang ada pada textbox4
    End If
    If RadioButton2.Checked = True Then
        TextBox5.Text =
        Microsoft.VisualBasic.Mid(TextBox1.Text,
        Val(TextBox2.Text), Val(TextBox3.Text))
        'Ambil character di tengah pada textbox1 mulai
        'dari nilai yang ada pada textbox2, sebanyak
        'angka pada textbox3
    End If
    If RadioButton3.Checked = True Then
        TextBox2.Text = Len(TextBox1.Text) -
        Val(TextBox4.Text) + 1
        'Posisi awal teks diambil dari panjang teks yang
        'ditulis dikurangi dengan jumlah yang akan
        'diambil dari kanan dan ditambahkan dengan 1
        TextBox3.Text = Len(TextBox1.Text)
        'Posisi akhir teks adalah Posisi akhir teks yaitu
        'panjang character pada textbox1
        TextBox5.Text =
        Microsoft.VisualBasic.Right(TextBox1.Text,
        Val(TextBox4.Text))
        'Hasil diambil dari posisi paling kanan dari
        'textbox1 sebanyak angka pada textbox4
    End If
End Sub

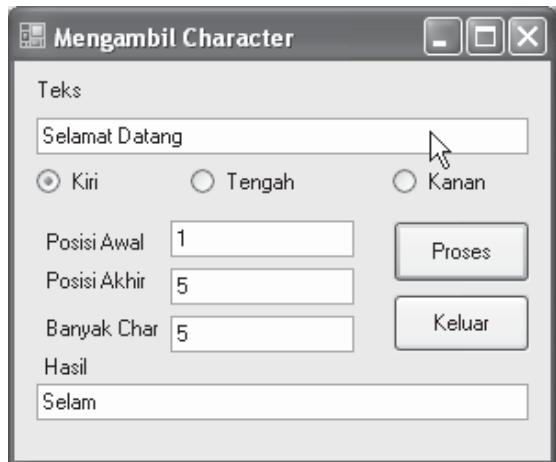
```

Kemudian buat perintah button keluar dengan cara mengklik ganda button Keluar kemudian ketik perintah berikut:

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
    'perintah untuk keluar
    End
End Sub

```



Gambar 3.13 Output program mengambil character

Langkah untuk input data pada program mengambil character yaitu tekan F5 untuk menjalankan program, kemudian inputkan teks yang akan diambil lalu klik radiobutton kiri. Isi banyak character yang akan diambil dari kiri dan klik button proses untuk menampilkan hasil pengambilan character.

3.5 Membuang Spasi Variabel

Untuk membuang spasi yang terdapat dalam variabel atau textbox digunakan perintah Trim, Ltrim, Rtrim. Perintah Trim digunakan untuk membuang spasi di bagian kiri dan kanan character, Ltrim digunakan untuk membuang spasi di kiri character, sedangkan Rtrim digunakan untuk membuang spasi di posisi kanan character.

```
' Membuang spasi di kiri, kanan teks
Dim TestString As String = " <-Trim-> "
Dim TrimString As String

' Output perintah menghasilkan "<-Trim-> ".
TrimString = LTrim(TestString)
' Output perintah menjadi " <-Trim-> ".
TrimString = RTrim(TestString)
```

```
' Output perintah menjadi "<-Trim->".
TrimString = LTrim(RTrim(TestString))

' Penggunaan perintah Ltrim dan Rtrim di atas sama dengan
' menggunakan satu perintah trim berikut ini
' Hasil "<-Trim->".
TrimString = Trim(TestString)
```

3.6 Membandingkan Dua Buah String

Untuk membandingkan dua buah string digunakan perintah StrComp. Metode yang digunakan dalam membandingkan string ada dua macam, yaitu metode binary dan metode text. Sedangkan hasil dari perbandingan adalah angka 0,1 dan -1. Nilai 0 (nol) merupakan hasil yang dimunculkan apabila kedua string sama. Apabila tidak, maka yang muncul angka 1 atau -1.

-1. Untuk lebih jelasnya perhatikan kode program berikut ini:

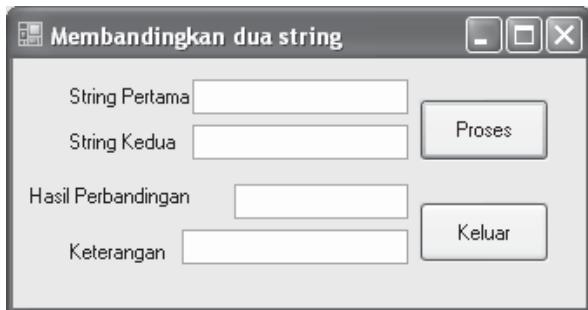
```
' Definisi variable.
Dim TestStr1 As String = "ABCD"
Dim TestStr2 As String = "abcd"
Dim TestComp As Integer

' Perbandingan 2 string dengan nilai 0.
TestComp = StrComp(TestStr1, TestStr2, CompareMethod.Text)

' Perbandingan menghasilkan nilai -1.
TestComp = StrComp(TestStr1, TestStr2, CompareMethod.Binary)

' Perbandingan string menghasilkan nilai 1.
TestComp = StrComp(TestStr2, TestStr1)
```

Untuk lebih jelas lagi tentang penggunaan perintah membandingkan kedua string, rancanglah form berikut ini dengan memasukkan komponen label string pertama, label string kedua, hasil perbandingan, dan string keterangan. Masukkan pula textbox untuk menampung input string pertama, textbox untuk input string kedua, textbox untuk menampung output hasil perbandingan, dan textbox untuk keterangan. Kemudian masukkan komponen button Proses dan komponen button Keluar.

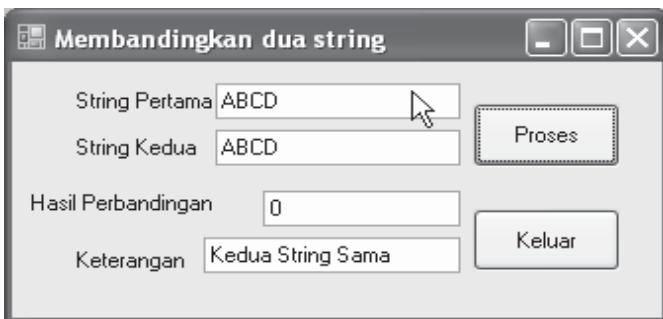


Gambar 3.14 Rancangan program membandingkan string

Setelah Anda merancang form di atas, klik ganda button Proses dan buatlah perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Nilai As Integer
    Nilai = StrComp(TextBox1.Text, TextBox2.Text,
                    CompareMethod.Text)
    TextBox3.Text = Nilai
    If Nilai = 0 Then
        TextBox4.Text = "Kedua String Sama"
    Else
        TextBox4.Text = "String Tidak Sama"
    End If
End Sub
```

Jalankan program dengan menekan tombol F5.



Gambar 3.15 Hasil program membandingkan dua string

3.7 Mengambil Nilai ASCII dan Character

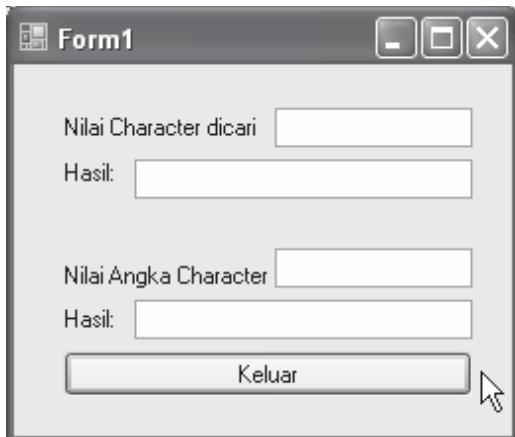
Untuk mengambil nilai ASCII menggunakan perintah Asc dan perintah AscW. Parameter yang terdapat dalam perintah Asc dan AscW adalah tipe data Char dan output dari perintah tersebut adalah Integer. Berikut ini adalah contoh penggunaan perintah asc dan ascW dalam variabel.

```
Dim codeInt As Integer  
' Nilai output adalah 65.  
codeInt = Asc("A")  
  
' Nilai ascii a kecil adalah 97.  
codeInt = Asc("a")  
  
' Nilai Ascii adalah 65, character yang diambil hanya  
'character yang paling depan.  
codeInt = Asc("Apple")
```

Nilai data ada dua macam, yaitu nilai berupa angka dan nilai berupa huruf atau character. Untuk mendapatkan nilai angka dari sebuah string menggunakan perintah Asc dan AscW. Sedangkan untuk mengambil nilai berupa huruf, Anda dapat menggunakan perintah Chr dan ChrW seperti contoh berikut:

```
Dim associatedChar As Char  
' Menghasilkan nilai "A".  
associatedChar = Chr(65)  
  
' menghasilkan nilai "a".  
associatedChar = Chr(97)  
  
' menghasilkan ">".  
associatedChar = Chr(62)  
  
' menghasilkan "%".  
associatedChar = Chr(37)
```

Untuk memperjelas penggunaan perintah-perintah di atas, mari kita buat program untuk mencari nilai ASCII dengan rancangan program sebagai berikut:



Gambar 3.16 Rancangan program nilai ASCII

Masukkan empat buah label masing-masing untuk nilai character yang dicari, label hasil nilai character, nilai angka character, dan hasil angka character. Masukkan pula textbox untuk nilai character yang dicari, textbox hasil character, textbox nilai angka character, dan textbox hasil angka character. Masukkan pula satu buah button Keluar.

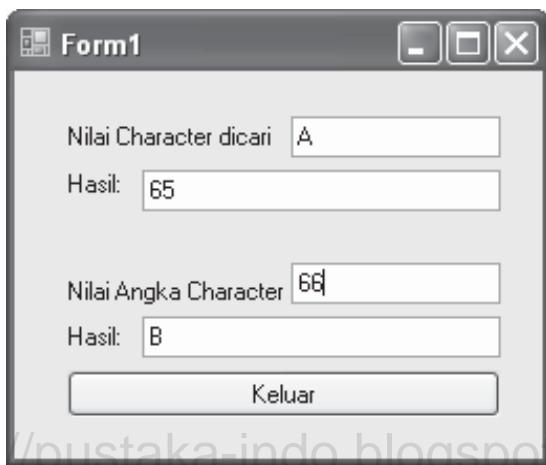
Untuk membuat perintah, klik ganda textbox nilai character yang dicari lalu ketik perintah berikut:

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    On Error Resume Next
    TextBox2.Text = Asc(TextBox1.Text)
End Sub
```

Klik ganda textbox nilai angka character kemudian ketik perintah berikut:

```
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged
    On Error Resume Next
    TextBox4.Text = Chr(Val(TextBox3.Text))
End Sub
```

Perintah On Error Resume Next digunakan untuk mencegah terjadinya error ketika Anda menjalankan program. Klik ganda button Keluar lalu buat perintah end. Apabila sudah selesai, jalankan program dengan menekan F5.

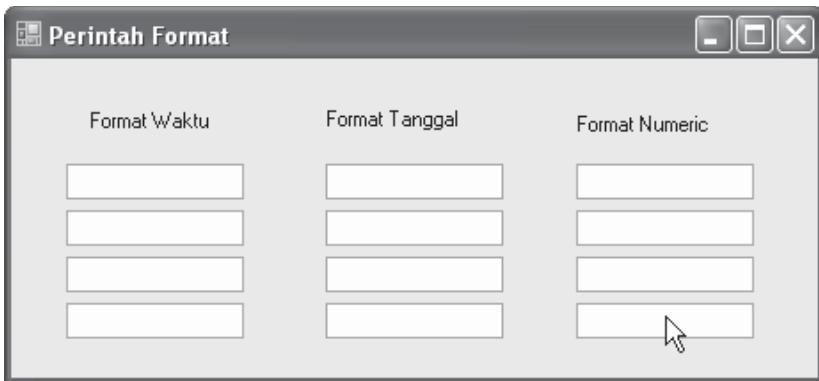


Gambar 3.17 Hasil program nilai ASCII

3.8 Format

Perintah format digunakan untuk mengatur bentuk tampilan angka, tanggal, dan waktu sesuai dengan standar yang ditetapkan oleh perintah tersebut. Rancanglah form di bawah ini untuk mencoba penggunaan perintah format. Masukkan label sebanyak tiga buah, masing-masing untuk label format waktu, label format tanggal, dan label format numeric.

Masukkan empat buah textbox untuk contoh perintah format waktu, empat buah textbox untuk contoh perintah format tanggal, dan empat buah textbox untuk contoh perintah format numeric.

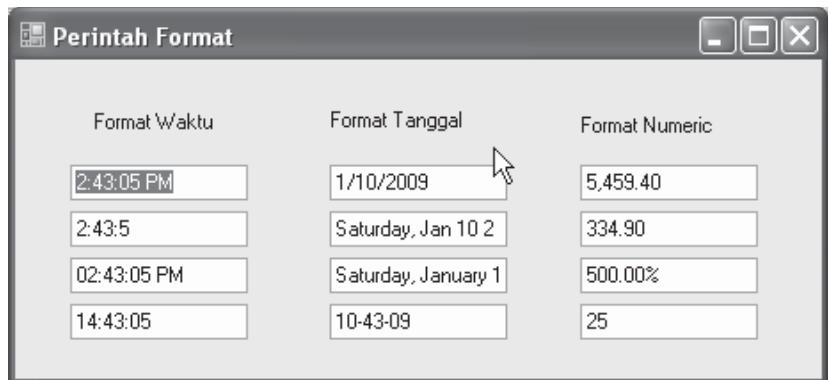


Gambar 3.18 Rancangan program format

Untuk membuat perintah format, klik ganda form lalu ketiklah perintah program berikut ini:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load  
    'Bentuk Perintah format Waktu  
    TestStr = Format(Now(), "Long Time")  
    TextBox1.Text = TestStr  
    TextBox2.Text = Format(Now(), "h:m:s")  
    TextBox3.Text = Format(Now(), "hh:mm:ss tt")  
    TextBox4.Text = Format(Now(), "HH:mm:ss")  
  
    'Bentuk perintah format tanggal  
    TextBox5.Text = Format(Now(), "Short Date")  
    TextBox6.Text = Format(Now(), "dddd, MMM d yyyy")  
    TextBox7.Text = Format(Now(), "D")  
    TextBox8.Text = Format(Now(), "dd-mm-yy")  
  
    'Contoh penggunaan perintah Format Numeric  
    TextBox9.Text = Format(5459.4, "##,##0.00")  
    TextBox10.Text = Format(334.9, "##0.00")  
    TextBox11.Text = Format(5, "0.00%")  
    TextBox12.Text = Format(25)  
End Sub
```

Jalankan program yang sudah dibuat dengan menekan tombol F5 atau klik menu debug lalu klik menu *start debugging*.



Gambar 3.19 Output program format

Selain menggunakan perintah di atas, format tanggal dapat juga dibuat dengan perintah `FormatDateTime`. Perintah `FormatDateTime` memiliki standar format `DateFormat.GeneralDate`, `DateFormat.LongDate`, `DateFormat.ShortDate`, `DateFormat.LongTime`, `DateFormat.ShortTime`.

Buatlah rancangan form format tanggal dan jam berikut untuk mencoba penggunaan perintah `FormatDateTime`. Masukkan lima buah label untuk Tanggal General Date, Label Tanggal Long Date, Label tanggal short Date, label Long Time, dan label untuk Short Time. Masukkan pula lima buah textbox untuk masing-masing label.



Gambar 3.20 Rancangan program format date time

Klik ganda form lalu kemudian ketiklah perintah program format date time berikut ini:

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load
        TextBox1.Text = FormatDateTime(Now,
        DateFormat.GeneralDate)
        TextBox2.Text = FormatDateTime(Now,
        DateFormat.LongDate)
        TextBox3.Text = FormatDateTime(Now,
        DateFormat.ShortDate)
        TextBox4.Text = FormatDateTime(Now,
        DateFormat.LongTime)
        TextBox5.Text = FormatDateTime(Now,
        DateFormat.ShortTime)
    End Sub
End Class
```



Gambar 3.21 Hasil program format date time

Untuk membuat format angka, ada perintah khusus untuk membuatnya, perintah khusus tersebut yaitu FormatCurrency; digunakan untuk format mata uang berdasarkan “TriState.UseDefault” atau berdasarkan pengaturan regional setting sistem operasi yang digunakan pada sistem komputer.

Contoh penggunaan perintah formatCurrency adalah sebagai berikut:

```
Dim TestDebt As Double = -4456.43
Dim TestString As String
' Hasil output perintah "($4,456.43)".
TestString = FormatCurrency(TestDebt, , , TriState.True,
TriState.True)
```

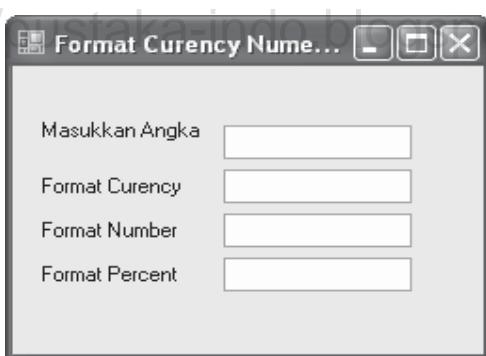
Format numeric yang lain adalah Perintah FormatNumber, penggunaan perintah ini mirip dengan perintah formatCurrency. Berikut ini contoh penggunaan perintah FormatNumber.

```
Dim TestNumber As Integer = 45600
' Hasil output perintah "45,600.00".
Dim TestString As String = FormatNumber(TestNumber, 2, , ,
TriState.True)
```

FormatPercent juga mempunyai format khusus. Jadi selain dengan perintah format, kita juga dapat menggunakan perintah lain seperti FormatCurrency, Format Number, FormatDateTime, dan FormatPercent.

```
Dim TestNumber As Single = 0.76
' Returns "76.00%".
Dim TestString As String = FormatPercent(TestNumber)
```

Berikut ini contoh penggunaan perintah formatCurrency, FormatNumber, dan FormatPercent dengan menggunakan textbox.



Gambar 3.22 Rancangan program Format Currency, Number, dan Persen

Masukkan empat buah label untuk angka, label format currency, label format number, dan label format persen. Masukkan pula textbox untuk menampung angka dan tiga buah textbox lainnya untuk menampung format currency, number, dan persen. Klik ganda textbox1 kemudian ketiklah perintah program berikut ini:

```

Public Class Form1

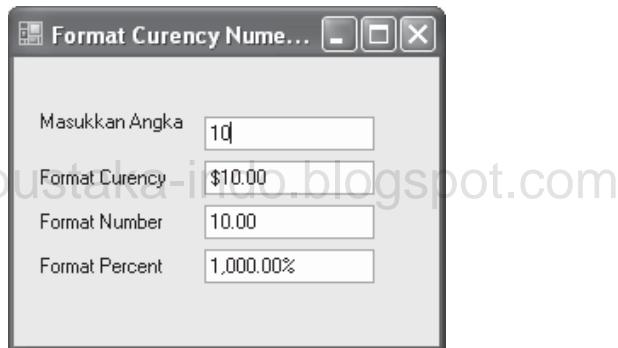
    Private Sub TextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged
        ' Contoh perintah formatcurrency
        TextBox2.Text = FormatCurrency(Val(TextBox1.Text), , ,
TriState.True, TriState.UseDefault)

        'Contoh perintah format number
        TextBox3.Text = FormatNumber(Val(TextBox1.Text), 2, ,
TriState.True)

        'Contoh perintah format persen
        TextBox4.Text = FormatPercent(Val(TextBox1.Text))
    End Sub
End Class

```

Jalankan program yang sudah Anda buat dengan menekan tombol F5.



Gambar 3.23 Output program Format Currency

3.9 Mengambil Posisi Character dan Split

Untuk mengambil Character atau huruf pada posisi tertentu, gunakan perintah GetChar. Lihat contoh berikut ini:

```

Dim TestString As String = "ABCDE"
Dim TestChar As Char
' Output "D"
TestChar = GetChar(TestString, 4)

```

Untuk mengambil character pada variabel TestString di posisi keempat akan menghasilkan keluaran/output “D”, karena posisi keempat pada baris “ABCDE” adalah huruf “D”.

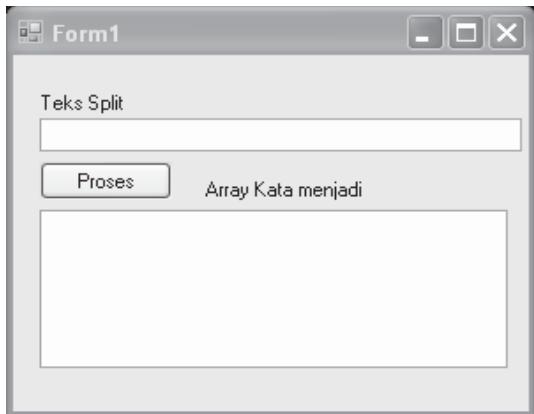
Perintah *split* digunakan untuk memotong kata yang terdapat dalam sebuah variabel, kemudian kata yang terdapat di dalam variabel tersebut akan dipisahkan per kata dan ditampung dalam variabel array. Berikut ini contoh perintah split:

```
Dim TestString As String = "Look at these!"  
  
' Hasil array menjadi "Look", "at", and "these!".  
Dim TestArray() As String = Split(TestString)
```

Contoh split dengan menggunakan For:

```
Dim TestString As String = "apple pear banana "  
Dim TestArray() As String = Split(TestString)  
' Array = {"apple", "", "", "", "pear", "banana", "", ""}  
  
Dim LastNonEmpty As Integer = -1  
For i As Integer = 0 To TestArray.Length - 1  
    If TestArray(i) <> "" Then  
        LastNonEmpty += 1  
        TestArray(LastNonEmpty) = TestArray(i)  
    End If  
Next  
ReDim Preserve TestArray(LastNonEmpty)  
' Array yang baru menjadi {"apple", "pear", "banana"}
```

Rancanglah program split berikut ini dengan dua label, masing-masing untuk label teks dan label untuk array. Masukkan satu buah textbox untuk menampung kalimat yang akan dibuat, kemudian masukkan komponen button Proses. Masukkan pula satu komponen listbox untuk menampung hasil array dari split (memotong kata).



Gambar 3.24 Rancangan program split

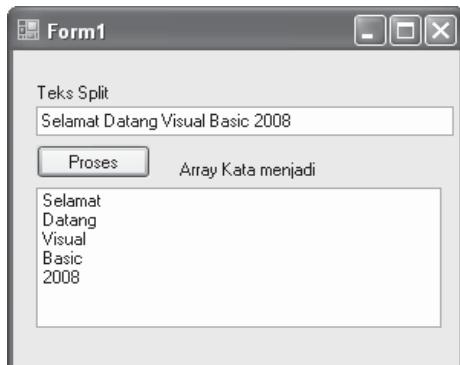
Klik ganda button Proses dan ketik perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
'Dim TestString As String = "Selamat Datang Visual Basic 2008"
Dim TestArray() As String
TestArray = Split(TextBox1.Text)

Dim LastNonEmpty As Integer = -1
For i As Integer = 0 To TestArray.Length - 1
    If TestArray(i) <> "" Then
        LastNonEmpty += 1
        'Ambil isi variabel TestArray
        TestArray(LastNonEmpty) = TestArray(i)

        'Kirim split ke listBox
        ListBox1.Items.Add(TestArray(i))
    End If
Next
ReDim Preserve TestArray(LastNonEmpty)
End Sub
```

Perintah variabel TestArray akan menampung data dari hasil split pada textbox1, kemudian hasil dari split (potongan kata) akan ditampung dalam variabel testarray sebanyak nilai variabel i. Setelah variabel testarray menampung data dari textbox, maka data akan dikirim ke listBox1.



Gambar 3.25 Hasil program split

3.10 Duplikat dan Mengganti String

Membuat duplikat string menggunakan perintah StrDup dengan format perintah dengan parameter jumlah duplikat dan string yang akan dibuat duplikat. Bentuk perintah StrDup adalah sebagai berikut:

```
Public Shared Function StrDup( _
    ByVal Number As Integer, _
    ByVal Character As { Char | String } _ 
) As String
' -or-
Public Shared Function StrDup( _
    ByVal Number As Integer, _
    ByVal Character As Object _ 
) As Object
```

Berikut contoh penggunaan perintah StrDup:

```
Dim aString As String = "Wow! What a string!"
Dim aObject As New Object
Dim TestString As String
aObject = "This is a String contained within an Object"
'Duplikatkan huruf "P" sebanyak lima kali
TestString = StrDup(5, "P") ' Hasil "PPPPP"

'Isi variabel pertama aString adalah huruf W
'Jadi Duplikatkan huruf W sebanyak 10 kali
TestString = StrDup(10, aString) ' Hasil "WWWWWWWWWW"

'Isi variabel pertama dari aObject adalah huruf T
TestString = CStr(StrDup(6, aObject)) ' Hasil "TTTTTT"
```

Replace adalah perintah yang digunakan untuk mengganti string dalam suatu variabel dengan string yang lain. Bentuk perintah Replace adalah sebagai berikut:

```
Public Function Replace(  
    ByVal Expression As String,  
    ByVal Find As String,  
    ByVal Replacement As String,  
    Optional ByVal Start As Integer = 1,  
    Optional ByVal Count As Integer = -1,  
    Optional ByVal Compare As CompareMethod =  
        CompareMethod.Binary  
) As String
```

Contoh penggunaan perintah:

```
Dim TestString As String = "Shopping List"  
'Ganti string "o" menjadi string "i" pada variabel TestString  
' Hasil "Shipping List".  
Dim aString As String = Replace(TestString, "o", "i")
```

<http://pustaka-indo.blogspot.com>



BERMAIN DENGAN PERINTAH FILE SYSTEM

<http://pustaka-indo.blogspot.com>

Perintah-perintah Microsoft Visual Basic 2008 yang berkaitan dengan file system adalah perintah-perintah DOS, seperti perintah untuk membuat folder, menghapus folder, menyalin file, dan sebagainya. Berikut ini pembahasan mengenai perintah-perintah file system tersebut.

4.1 Create, Change, dan Delete Folder

Untuk membuat sebuah folder dengan Visual Basic 2008 menggunakan perintah MKDIR (Make Directory) seperti terlihat pada contoh berikut:

```
' Membuat Folder atau directori.  
MkDir("C:\TESTFolder")
```

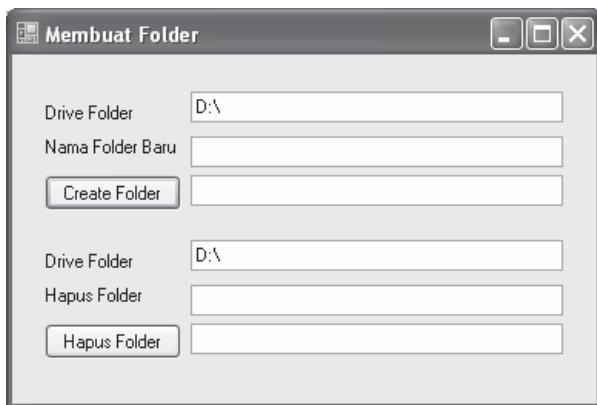
Untuk mengubah posisi folder ke folder yang lain, gunakan perintah CHDIR (Change Directory). Contoh penggunaan perintah change directory adalah sebagai berikut:

```
' Mengubah directory aktif ke folder "MYDIR".  
ChDir("MYDIR")  
  
' Kembali standar folder aktif  
ChDir("D:\WINDOWS\SYSTEM")  
  
'Pindah satu tingkat dari directory aktif  
ChDir("../")
```

Perintah untuk menghapus direktori yang aktif adalah RMDIR (Remove Directory). Contoh penggunaan perintah tersebut adalah sebagai berikut:

```
' Dengan Asumsi folder MYDIR dalam keadaan kosong.  
' Menghapus folder MYDIR.  
RmDir("MYDIR")
```

Syarat untuk menghapus directory yaitu tidak boleh berada pada folder yang akan dihapus, tidak boleh berada pada sub folder directory yang akan dihapus, dan juga folder dalam keadaan kosong atau tidak ada file yang di dalam folder tersebut.



Gambar 4.1 Rancangan program membuat folder

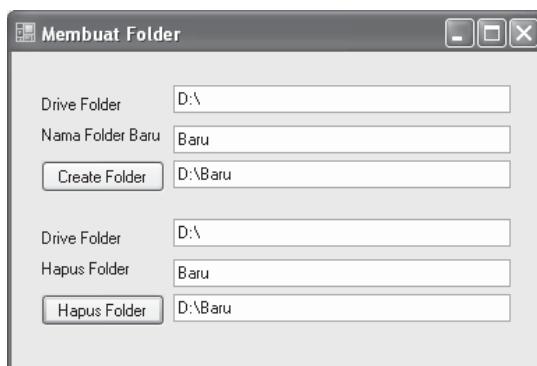
Rancanglah form Gambar 4.1 dengan membuat label Drive Folder dan label Nama Folder Baru. Masukkan komponen button Create Folder kemudian masukkan tiga buah textbox masing-masing untuk textbox *drive folder*, *nama folder*, dan *hasil create folder*. Untuk membuat perintah hapus folder hampir sama dengan create folder. Masukkan komponen yang dibutuhkan seperti tampak pada gambar tersebut. Klik ganda button Create Folder lalu buat perintah sebagai berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    On Error Resume Next
    ChDir(TextBox1.Text)
    MkDir(TextBox2.Text)
    TextBox3.Text = TextBox1.Text + TextBox2.Text
End Sub
```

Klik ganda button Hapus Folder dan buat perintah program berikut ini:

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    On Error Resume Next
    ChDir(TextBox6.Text)
    RmDir(TextBox5.Text)
    TextBox4.Text = TextBox6.Text + TextBox5.Text
End Sub
```

Jalankan program dengan menekan tombol F5.



Gambar 4.2 Output program membuat folder

Perintah CHDIR (Change Folder) mirip dengan perintah ChDrive, tetapi perintah ChDrive hanya dapat digunakan untuk pindah drive. Contoh penggunaan perintah ChDrive adalah sebagai berikut:

```
'Contoh perintah Pindah drive D  
ChDrive ("D")
```

Selain perintah ChDrive, terdapat satu perintah untuk menetapkan file, folder, atau directory aktif sesuai dengan atribut file, yaitu perintah DIR.

Contoh penggunaan perintah DIR adalah sebagai berikut:

```
Dim MyFile, MyPath, MyName As String  
' Aktif pada file "WIN.INI"  
MyFile = Dir("C:\WINDOWS\WIN.INI")  
  
' File aktif pada *.INI  
MyFile = Dir("C:\WINDOWS\*.INI")  
  
' File pada folder aktif  
MyFile = Dir()  
  
Menampilkan semua file TXT dengan attribut Hidden  
MyFile = Dir "*.TXT", vbHidden)  
  
MyPath = "c:\" ' mengatur path.  
MyName = Dir(MyPath, vbDirectory)  
Do While MyName <> "" ' Start the loop.  
    If (GetAttr(MyPath & MyName) And vbDirectory) =  
        vbDirectory Then  
            MsgBox(MyName)  
        End If  
    MyName = Dir() ' Menampilkan folder berikutnya  
Loop
```

4.2 File

File adalah data-data yang tersimpan di dalam media yang mempunyai informasi besar file, tanggal dan jam penyimpanan file, nama file, ciri file (ciri aplikasi yang membuat), dan attribut file.

4.2.1 FileDateTime

FileDateTime adalah perintah untuk menampilkan tanggal dan jam pembuatan file. File yang akan diambil tanggal dan jam pembuatannya harus sudah tersimpan di dalam disk.



Gambar 4.3 Desain program filedatetime

Untuk merancang form di atas, masukkan komponen label lokasi folder dan textbox. Masukkan lagi komponen label nama file beserta textboxnya. Masukkan juga komponen button Proses dan masukkan label dan textbox untuk output tanggal jam penyimpanan file. Klik ganda button Proses lalu ketik perintah berikut:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button1.Click
        On Error Resume Next
        Dim MyStamp As Date
        '
        MyStamp = FileDateTime(TextBox1.Text & TextBox2.Text)
        If Err.Number = 53 Then
            MsgBox("File Not Found (" & Err.Number & ")")
        End If
        If Err.Number = 52 Then
            MsgBox("PathName is invalid or contains
                    wildcards")
        End If
        TextBox3.Text = MyStamp
    End Sub
End Class
```

Jalankan program dengan menekan tombol F5, apabila nama file tidak ada, maka akan muncul pesan “File Not Found”. Tetapi apabila file ada di dalam folder yang sudah diinputkan, maka tanggal dan jam penyimpanan file akan muncul.



Gambar 4.4 Output program FileDateTime

4.2.2 FileLen

FileLen adalah perintah untuk menampilkan panjang file dalam bentuk byte. Output perintah FileLen adalah tipe data angka (long integer), jadi untuk menggunakan perintah FileLen, kita harus mendefinisikan variabel dengan tipe data long integer seperti contoh berikut ini:

```
MyFile As Long  
MyFile=FileLen("TestFile.txt")
```

Tambahkan desain program Tanggal dan Jam File di atas dengan label dan textbox Panjang file sehingga output program menjadi seperti yang tampak pada Gambar 4.5. Sehingga perintah akan ditambahkan menjadi beberapa baris program seperti terlihat di bawah ini:

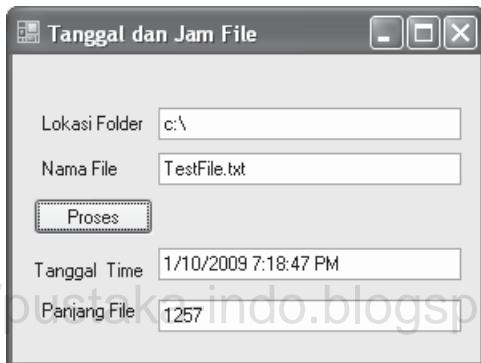
```
Public Class Form1  
    Private Sub Button1_Click(ByVal sender As System.Object,  
        ByVal e As System.EventArgs) Handles Button1.Click  
        On Error Resume Next  
        Dim MyStamp As Date  
        Dim MyFile As Long
```

```

MyStamp = FileDateTime(TextBox1.Text & TextBox2.Text)
If Err.Number = 53 Then
    MsgBox("File Not Found (" & Err.Number & ")")
End If
If Err.Number = 52 Then
    MsgBox("PathName is invalid or contains
wildcards")
End If
TextBox3.Text = MyStamp

'Perintah tambahan untuk cek panjang file
TextBox4.Text = FileLen(TextBox1.Text & TextBox2.Text)
End Sub
End Class

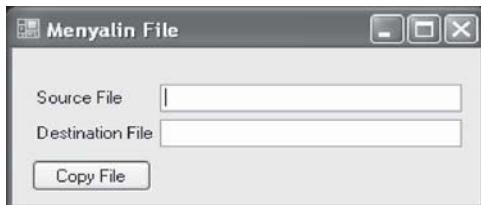
```



Gambar 4.5 FileLen

4.2.3 CopyFile

CopyFile adalah perintah untuk menyalin file. Untuk mencoba membuat program CopyFile, rancanglah form di bawah ini:

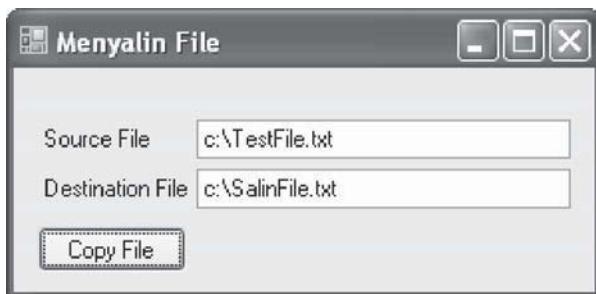


Gambar 4.6 Copy file

Masukkan Label dan TextBox untuk sumber atau source file, dan masukkan pula label dan textbox untuk Destination File. Tambahkan satu buah button dengan perintah sebagai berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    On Error Resume Next
    Dim SourceFile, DestinationFile As String
    SourceFile = TextBox1.Text
    DestinationFile = TextBox2.Text
    My.Computer.FileSystem.CopyFile(SourceFile,
        DestinationFile)
    If Err.Number = 53 Then
        MsgBox("File Not Found")
    End If
    If Err.Number = 55 Then
        MsgBox("File Already Open")
    End If
    If Err.Number = 52 Then
        MsgBox("Source or Destination is invalid or not
specified")
    End If
End Sub
```

Jalankan program dengan menekan tombol F5 lalu cek apakah file sudah tersalin ke dalam folder yang diinginkan.



Gambar 4.7 Output copy file

4.3 Atribut File dan Baca File Text

Untuk menampilkan atribut file dalam Microsoft Visual Basic 2008, gunakan perintah GetAttr. Parameter yang digunakan adalah nama file dan extention file. Untuk membaca isi file text, Anda gunakan perintah ReadAllText. Rancanglah form di bawah ini untuk mencoba perintah attribut file dan membaca file text.



Gambar 4.8 Desain program baca file text

Masukkan button untuk attribut file, masukkan pula button untuk Read File Text dan masukkan satu komponen richtextbox untuk menampilkan isi file text. Klik ganda button Attribute file lalu ketik perintah berikut ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    On Error Resume Next
    Dim MyAttr As FileAttribute
    ' Assume file TESTFILE is normal and readonly.
    MyAttr = GetAttr("C:\TESTFILE.txt") ' Returns
                                         vbNormal.

    ' Test for normal.
    If (MyAttr And FileAttribute.Normal) =
        FileAttribute.Normal Then
        MsgBox("This file is normal.")
    End If
```

```

' Test for normal and readonly.
Dim normalReadonly As FileAttribute
normalReadonly = FileAttribute.Normal Or
    FileAttribute.ReadOnly
If (MyAttr And normalReadonly) = normalReadonly Then
    MsgBox("This file is normal and readonly.")
End If

' Assume MYDIR is a directory or folder.
MyAttr = GetAttr("C:\MYDIR")
If (MyAttr And FileAttribute.Directory) =
    FileAttribute.Directory Then
    MsgBox("MYDIR is a directory")
End If

Dim information As System.IO.FileInfo
information = My.Computer.FileSystem.
    GetFileInfo("C:\MyLogFile.log")
MsgBox("The file's full name is " &
    information.FullName & ".")
MsgBox("Last access time is " &
    information.LastAccessTime & ".")
MsgBox("The length is " & information.Length & ".")
End Sub

```

Klik ganda button Read File Text kemudian ketiklah perintah berikut untuk menampilkan isi file text:

```

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    Dim fileReader As String
    fileReader = My.Computer.FileSystem.
        ReadAllText("C:\testfile.txt")
    MsgBox(fileReader)
    RichTextBox1.Text = fileReader
End Sub

```

Jalankan program dengan menekan tombol F5 atau klik menu debug lalu pilih sub menu *start debuging*. File yang akan ditampilkan isinya adalah file testfile.txt. Apabila file tersebut belum ada maka Anda harus membuat file tersebut dengan menggunakan aplikasi Notepad.

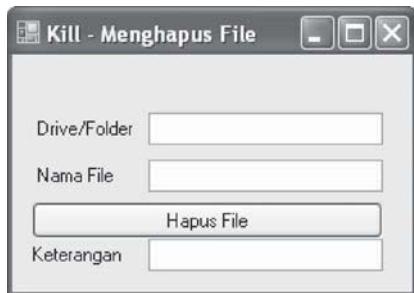


Gambar 4.9 Output program Atribut dan baca file text

4.4 Menghapus File

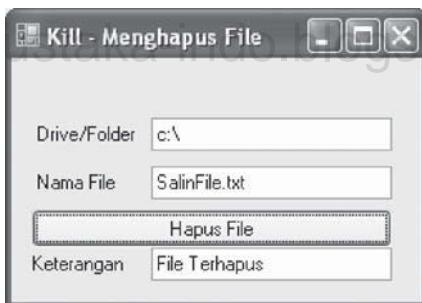
Untuk menghapus file yang ada di dalam suatu folder atau drive tertentu, gunakan perintah Kill. Perintah ini juga dapat dikombinasikan dengan tanda * (asterisk) yang berarti semua file atau semua extention. Juga dapat dikombinasikan dengan tanda ? (tanda tanya) yang berarti satu karakter bebas. Perintah kill dalam operating system windows sama dengan perintah delete atau del.

Rancanglah aplikasi berikut untuk mencoba perintah kill dalam sebuah program. Masukkan label drive/folder kemudian masukkan textbox untuk drive/folder. Masukkan label dan textbox nama file dan masukkan button menghapus file. Terakhir, masukkan label dan textbox keterangan seperti terlihat pada Gambar 4.10.



Gambar 4.10 Rancangan program Kill

Klik ganda button Hapus File dan ketiklah perintah program berikut. Apabila sudah selesai, jalankan program dengan menekan F5 lalu isilah folder dan nama file kemudian klik button Hapus File. Cek file yang ada dalam folder menggunakan Windows Explorer untuk memastikan file sudah terhapus atau belum.



Gambar 4.11 Hasil program Kill

4.5 Mengunci dan Membuka File

Perintah lock digunakan untuk mengunci file agar tidak dapat dibuka, sedangkan unlock digunakan untuk membuka kunci file agar dapat dibuka oleh suatu aplikasi. Rancanglah program lock dan unlock file seperti gambar berikut ini:



Gambar 4.12 Desain program Lock File

Masukkan label Folder, label nama file, dan label keterangan file.
Masukkan textbox folder, textbox nama file, dan textbox keterangan file.
Masukkan button lock, button unlock, button open file, dan masukkan komponen RichTextBox untuk menampilkan isi file yang akan dibuka.
Klik ganda button Lock lalu ketik perintah program untuk mengunci file:

```
Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    On Error Resume Next
    RichTextBox1.Text = " "
    FileOpen(1, TextBox1.Text & TextBox2.Text,
        OpenMode.Binary)
    Lock(1)
    TextBox3.Text = "Lock"
End Sub
```

Perintah on error resume next digunakan untuk mencegah kesalahan yang mungkin muncul pada saat program dijalankan. RichTextBox.text = “ ” digunakan untuk mengosongkan komponen RichTextBox. FileOpen digunakan untuk membuka file sesuai dengan lokasi folder yang dibuat pada textbox1 dan nama file yang tertulis pada textbox2.

Setelah file diaktifkan, kunci dengan perintah lock kemudian beri keterangan lock pada textbox3. Langkah selanjutnya Anda klik ganda button Unlock lalu ketik perintah program berikut ini:

```

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    On Error Resume Next
    RichTextBox1.Text = " "
    Unlock(1)
    FileClose(1)
    TextBox3.Text = "UnLock"
End Sub

```

Sedangkan maksud dari perintah di atas adalah untuk mengosongkan RichTextBox1. Buka kunci file yang aktif, tutup file yang aktif lalu beri keterangan UnLock pada textBox3. Selanjutnya klik ganda button Open File lalu ketik perintah program berikut:

```

Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    On Error Resume Next
    RichTextBox1.Text =
    My.Computer.FileSystem.ReadAllText(TextBox1.Text &
    TextBox2.Text)
End Sub

```

Maksud perintah di atas adalah membaca isi file yang terdapat pada textbox1 dan textbox2 lalu ditampilkan pada RichTextBox. Coba Anda jalankan program dengan menekan tombol F5.



Gambar 4.13 Output program Lock File

4.6 Mengganti Nama File

Untuk mengganti nama file, gunakan perintah rename dengan parameter nama file lama dan nama file baru. Bentuk perintah yang ada pada Microsoft Visual Basic 2008 ini mirip dengan perintah yang terdapat pada DOS. Berikut ini contoh penggunaan perintah rename dengan asumsi bahwa file sudah ada di dalam folder. Untuk mengganti nama file kita dapat menggunakan tanda * (asterisk) dan tanda ? (tanda tanya) untuk menggantikan banyak character dan satu character yang terdapat dalam nama file dan ciri file.

```
Dim OldName, NewName As String  
OldName = "OLDFILE" 'Nama File lama  
NewName = "NEWFILE" 'Nama File yang baru  
Rename(OldName, NewName)  
  
OldName = "C:\OLDDIR\OLDFILE"  
NewName = "C:\NEWDIR\NEWFILE"  
' Move and rename file.  
Rename(OldName, NewName)
```

<http://pustaka-indo.blogspot.com>

4.7 Mencetak Tulisan

Mencetak tulisan menggunakan perintah print dan print line, namun tulisan yang akan dicetak bukan dalam bentuk media kertas tetapi tulisan akan dicetak dalam bentuk file. Berikut contoh penggunaan perintah tersebut di atas.

```
FileOpen(1, "c:\trash.txt", OpenMode.Output)  
'Membuat file trash.txt dan diletakkan pada folder c:\  
' Membuka file dengan mode output.  
  
Print(1, "This is a test.")  
' Print text dalam file.  
  
PrintLine(1)  
' Print character blank atau baris kosong  
  
PrintLine(1, "Zone 1", TAB(), "Zone 2")  
' Pemisah spasi menggunakan perintah TAB.  
  
PrintLine(1, "Hello", "World")
```

```

PrintLine(1, SPC(5), "5 leading spaces ")
' Print spasi dengan perintah SPC.

PrintLine(1, TAB(10), "Hello")
' Print kata hello dengan 10 kolom dengan perintah tab.

Dim aBool As Boolean
Dim aDate As DateTime
aBool = False
aDate = DateTime.Parse("February 12, 2008")
'

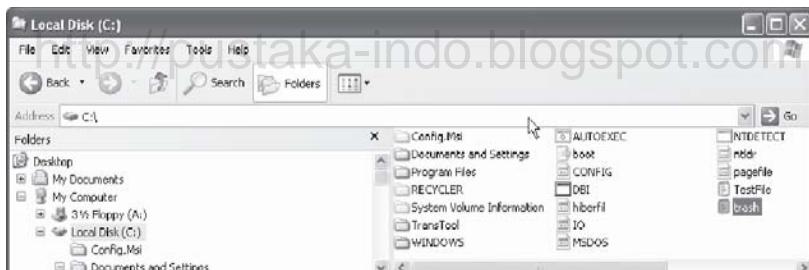
PrintLine(1, aBool, " is a Boolean value")
'Cetak isi variabel aBool

PrintLine(1, aDate, " is a date")
'Cetak isi variabel aDate

FileClose(1) ' Close file.
'Tutup File yang sedang aktif.

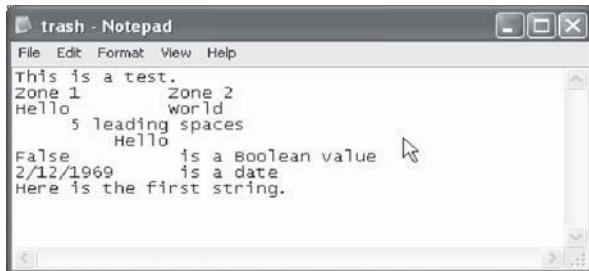
```

Untuk melihat hasil file text (trash.txt) yang sudah dibuat pada drive c:\ gunakan Windows Explorer.



Gambar 4.14 Explore File

Kemudian cari file trash.txt, apabila sudah Anda temukan, buka dengan cara mengklik ganda nama file tersebut sehingga akan terlihat isi file seperti pada Gambar 4.15.



Gambar 4.15 Isi file trash.txt

Selain menggunakan perintah print dan PrintLine, Anda dapat menggunakan perintah lain yaitu Write dan Writeline yang digunakan untuk menulis text pada file yang aktif. Penggunaan perintah write dan writeline mirip sekali dengan penggunaan perintah print dan println. Lihat listing program berikut adn bandingkan perintahnya dengan sebelumnya.

```
FileOpen(1, "TESTFILE", OpenMode.Output)
Write(1, "This is a test.")
WriteLine(1)
WriteLine(1, "Zone 1", TAB(), "Zone 2")
WriteLine(1, "Hello", " ", "World")
WriteLine(1, SPC(5), "5 leading spaces ")
WriteLine(1, TAB(10), "Hello")

Dim aBool As Boolean
Dim aDate As DateTime
aBool = False
aDate = DateTime.Parse("February 12, 1969")

WriteLine(1, aBool, " is a Boolean value")
WriteLine(1, aDate, " is a date")
FileClose(1) ' Close file.
```

Untuk membuka file text sudah yang sudah dibuat, gunakan perintah:

```
Private Sub Button2_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button2.Click
 Dim file As System.IO.StreamWriter
 file = My.Computer.FileSystem.
     OpenTextWriter("c:\trash.txt", True)
 file.WriteLine("Here is the first string.")
 file.Close()
 RichTextBox1.Text = My.Computer.
     FileSystem.ReadAllText("c:\Trash.txt")
End Sub
```




BERMAIN DENGAN PERINTAH DATE AND TIME MODULE

<http://pustaka-indo.blogspot.com>

DateAndTime module adalah perintah yang digunakan untuk mengatur dan mengoperasikan perintah tanggal dan waktu. Tanggal dan waktu tersebut dikalkulasi dengan tanda-tanda operasi matematika, ditampilkan dengan pengaturan tanggal dan durasi waktu. Perintah ini juga berkaitan dengan teknik dan format menampilkan nama hari, angka tanggal, angka dan character bulan, angka tahun, angka jam, juga angka menit dan detik.

5.1 Tanggal dan Waktu Sistem

5.1.1 Now

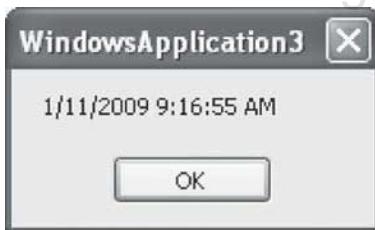
Fungsi perintah Now adalah untuk menampilkan tanggal dan waktu sistem komputer yang aktif sekarang. Tanggal yang ditampilkan adalah angka tanggal, angka bulan, dan angka tahun. Sedangkan untuk waktu, yang ditampilkan adalah jam, menit, dan detik. Berikut ini contoh penggunaan perintah now dalam sebuah program.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim ThisMoment As Date
    ' Buat Variabel ThisMoment dengan tipe data tanggal

    ThisMoment = Now
    'Variabel thisMoment diisi dengan perintah now

    MsgBox(ThisMoment)
    'Menampilkan tanggal dan waktu dengan message box
End Sub
```

<http://pustaka-indo.blogspot.com>



Gambar 5.1 Hasil program Now

5.1.2 Today

Fungsi perintah today adalah untuk menampilkan tanggal dari sistem komputer yang aktif sekarang. Tanggal akan dihitung mulai jam 00:00:00. Berikut ini adalah contoh penggunaan perintah today dalam program.

```

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim ThisMoment As Date
    ' Buat Variabel ThisMoment dengan tipe data tanggal
    ThisMoment = Today
    'Variabel thisMoment diisi dengan perintah TToday

    MsgBox(ThisMoment)
    'Menampilkan tanggal dengan message box

End Sub

```

Setalah program dijalankan maka akan tampil tanggal seperti terlihat pada gambar di bawah ini.



Gambar 5.2 Today

5.1.3 TimeOfDay

Fungsi perintah TimeOfDay adalah untuk menampilkan waktu sistem komputer yang aktif sekarang. Ketik kode program di bawah ini untuk mencoba perintah TimeOfDay.

```

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim ThisMoment As Date
    ' Buat Variabel ThisMoment dengan tipe data tanggal
    ThisMoment = TimeOfDay
    'Variabel thisMoment diisi dengan perintah TToday

    MsgBox(ThisMoment)
    'Menampilkan jam dengan message box
End Sub

```

Apabila program yang berisi perintah tersebut dijalankan, maka akan terlihat tampilan jam seperti pada gambar berikut:



Gambar 5.3 TimeOfDay

5.2 Proses Tanggal

5.2.1 DateAdd

Fungsi perintah DateAdd digunakan untuk mencari rentang/interval hari, bulan, tahun, jam, menit dan detik dengan menggunakan Type Interval, angka penambahan, dan tanggal/waktu awal penambahan. Type interval untuk menentukan penambahan.

Apabila kita ingin menambahkan hari maka menggunakan tipe interval “DateInterval.day”. Untuk membuat interval minggu menggunakan tipe interval “DateInterval.week”. Untuk membuat rentang/interval bulan menggunakan tipe interval “DateInterval.month”. Untuk membuat interval tahun menggunakan “DateInterval.Year”. Untuk membuat interval waktu jam, menit, dan detik menggunakan tipe interval hour, minute, dan second.

Berikut ini contoh penggunaan perintah masing type interval:



Gambar 5.4 Interval Tanggal

Masukkan komponen label tanggal dan jam. Masukkan komponen textbox tanggal dan jam. Masukkan komponen GroupBox Hari, label dan komponen interval, label dan textbox hasil. Salin GroupBox hari menjadi GroupBox minggu, bulan dan tahun, kemudian aturlah posisi GroupBox. Klik Form lalu isi properties text dengan judul form “Interval tanggal dan jam”. Klik ganda form kemudian ketiklah perintah berikut ini:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load  
    TextBox1.Text = Today  
    TextBox2.Text = TimeOfDay  
    TextBox3.Focus()  
End Sub
```

Klik ganda textbox interval hari, lalu ketiklah perintah untuk mencari tanggal dari nilai interval hari yang diinputkan.

```
Private Sub TextBox3_TextChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TextBox3.TextChanged  
    Dim sHari As String  
    sHari = DateAndTime.DateAdd(DateInterval.Day,  
        Val(TextBox3.Text), Today)  
    TextBox4.Text = sHari  
End Sub
```

Klik ganda textbox interval minggu untuk mencari tanggal dari nilai interval minggu yang diinputkan.

```
Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged
    Dim sMinggu As String
    sMinggu = DateAndTime.DateAdd(DateInterval.Weekday,
        Val(TextBox5.Text), Today)
    TextBox6.Text = sMinggu
End Sub
```

Klik ganda textbox interval bulan untuk membuat perintah mencari tanggal berdasarkan interval bulan yang diinputkan.

```
Private Sub TextBox8_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged
    Dim sBulan As String
    sBulan = DateAndTime.DateAdd(DateInterval.Month,
        Val(TextBox8.Text), Today)
    TextBox7.Text = sBulan
End Sub
```

Klik ganda textbox interval tahun untuk membuat perintah mencari tanggal berdasarkan interval tahun yang diinputkan.

```
Private Sub TextBox10_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox10.TextChanged
    Dim sTahun As String
    sTahun = DateAndTime.DateAdd(DateInterval.Year,
        Val(TextBox10.Text), Today)
    TextBox9.Text = sTahun
End Sub
```

Setelah perintah selesai diketik, jalankan program dengan menekan tombol F5 atau dengan menggunakan menu Debuging, lalu pilih submenu *start debugging*. Isilah interval hari, minggu, bulan dan tahun sesuai dengan nilai interval yang ingin Anda cari.

Gambar 5.5 Hasil program interval tanggal

Berikut ini source code lengkap program interval tanggal di atas:

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        TextBox1.Text = Today
        TextBox2.Text = TimeOfDay
        TextBox3.Focus()
    End Sub

    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged
        Dim sHari As String
        sHari = DateAndTime.DateAdd(DateInterval.Day, Val(TextBox3.Text), Today)
        TextBox4.Text = sHari
    End Sub

    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged
        Dim sMinggu As String
        sMinggu = DateAndTime.DateAdd(DateInterval.Weekday, Val(TextBox5.Text), Today)
        TextBox6.Text = sMinggu
    End Sub
```

```

Private Sub TextBox8_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox8.TextChanged
Dim sBulan As String
sBulan = DateAndTime.DateAdd(DateInterval.Month,
Val(TextBox8.Text), Today)
TextBox7.Text = sBulan
End Sub

Private Sub TextBox10_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox10.TextChanged
Dim sTahun As String
sTahun = DateAndTime.DateAdd(DateInterval.Year,
Val(TextBox10.Text), Today)
TextBox9.Text = sTahun
End Sub
End Class

```

Untuk membuat perintah mencari interval waktu, mirip dengan perintah mencari interval tanggal, bedanya hanya pada tipe interval yang digunakan.

5.2.2 DateDiff

Fungsi perintah DateDiff adalah untuk menentukan nilai interval dari dua tanggal atau waktu. Nilai yang diberikan oleh perintah datediff adalah hasil pengurangan tanggal kedua dengan tanggal pertama.

Type Interval yang digunakan pada perintah datediff mirip dengan tipe interval yang digunakan pada perintah DateAdd: “DateInterval.Day”, “DateInterval.DayOfYear”, “DateInterval.Hour”, “DateInterval.Minute”, “DateInterval.Month”, “DateInterval.Second”, “DateInterval.WeekDay”, “DateInterval.WeekOfYear”, “DateInterval.Year”. Rancanglah form berikut ini untuk mencoba perintah DateDiff.



Gambar 5.6 Rancangan program interval dua tanggal

Masukkan komponen tanggal DateTimePicker sebanyak dua buah untuk tanggal pertama dan untuk tanggal kedua. Masukkan label dan textbox untuk interval day. Masukkan label dan textbox untuk interval dayofyear (interval hari). Masukkan juga label dan textbox untuk week (interval minggu). Masukkan label dan textbox untuk month (bulan). Masukkan label dan textbox untuk year (interval tahun). Klik ganda DateTimePicker2 lalu ketik perintah berikut:

```
Private Sub DateTimePicker2_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DateTimePicker2.ValueChanged
    Dim Tanggal1 As Date
    Dim tanggal2 As Date
    Tanggal1 = DateTimePicker1.Value
    tanggal2 = DateTimePicker2.Value

    'Perintah untuk mengisi interval Hari
    TextBox1.Text = DateAndTime.DateDiff(DateInterval.Day,
                                         Tanggal1, tanggal2)
    TextBox2.Text =
    DateAndTime.DateDiff(DateInterval.DayOfYear,
                         Tanggal1, tanggal2)

    'Perintah untuk mengisi interval minggu
    TextBox3.Text = DateAndTime.DateDiff(
        DateInterval.Weekday, Tanggal1, tanggal2)
```

```

'Perintah untuk mengisi interval bulan
TextBox4.Text = DateAndTime.DateDiff(
    DateInterval.Month, Tanggal1, tanggal2)

'Perintah untuk mengisi interval tahun
TextBox5.Text = DateAndTime.DateDiff(
    DateInterval.Year, Tanggal1, tanggal2)
End Sub

```



<http://pustaka-indo.blogspot.com>

Gambar 5.7 Hasil program interval dua tanggal

Untuk menggunakan perintah interval waktu buatlah rancangan form berikut ini:



Gambar 5.8 Rancangan form interval waktu

Masukkan dua label untuk jam awal dan jam kedua. Masukkan komponen DateTimePicker lalu ubah properties format menjadi time. Masukkan komponen datetimetype untuk jam kedua dan ubah properties format menjadi time. Masukkan label dan textbox interval jam. Masukkan label dan textbox interval menit. Masukkan label dan textbox interval detik untuk menampung proses interval waktu. Masukkan pula button proses interval dan button keluar. Klik ganda button Keluar lalu buat perintah berikut ini:

```
Private Sub Button2_Click(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles Button2.Click  
    End  
End Sub
```

Klik ganda button Proses Interval Waktu, lalu ketik perintah untuk menampilkan interval jam, interval menit, dan interval detik.

```
Private Sub Button1_Click(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles Button1.Click  
    Dim Waktu1 As Date  
    Dim Waktu2 As Date  
    Waktu1 = DateTimePicker1.Value  
    Waktu2 = DateTimePicker2.Value  
  
    TextBox1.Text = DateAndTime.DateDiff(  
        DateInterval.Hour, Waktu1, Waktu2)  
    TextBox2.Text = DateAndTime.DateDiff(  
        DateInterval.Minute, Waktu1, Waktu2)  
    TextBox3.Text = DateAndTime.DateDiff(  
        DateInterval.Second, Waktu1, Waktu2)  
End Sub
```

Jalankan program dengan menekan F5 lalu isilah waktu awal dan waktu kedua, kemudian klik button Proses Interval Waktu sehingga tampil seperti Gambar 5.9.



Gambar 5.9 Hasil program interval waktu

5.2.3 DatePart

Fungsi perintah DatePart adalah untuk mengambil nilai dari sebagian tanggal dan waktu. Untuk mengambil nilai sebagian dari tanggal dan waktu tersebut menggunakan tipe interval sebagai berikut:

DatePart Function (Visual Basic)

FirstWeekOfYearValue, WeekOfYear
Optional. A value chosen from the **FirstWeekOfYear** enumeration that specifies the first week of the year. If not specified, **FirstWeekOfYear.Jan1** is used.

Settings

The *Interval* argument can have one of the following settings.

Enumeration value	String	Part of date/time value to return
DateInterval.Day	d	Day of month (1 through 31)
DateInterval.DayOfYear	y	Day of year (1 through 366)
DateInterval.Hour	h	Hour
DateInterval.Minute	n	Minute
DateInterval.Month	m	Month
DateInterval.Quarter	q	Quarter
DateInterval.Second	s	Second
DateInterval.Weekday	w	Day of week (1 through 7)
DateInterval.WeekOfYear	ww	Week of year (1 through 53)
DateInterval.Year	yyyy	Year

Gambar 5.10 Gambar tipe DateInterval

Rancanglah form di bawah ini dengan memasukkan komponen label tanggal dan jam. Masukkan komponen textbox untuk tanggal. Masukkan komponen textbox untuk jam. Masukkan button dan isi properties dengan “Proses DatePart”. Masukkan komponen GroupBox Tanggal dan Jam. Masukkan label dan textbox Day, Week, Month dan Year. Masukkan label Hour, minute, second. Masukkan komponen button Keluar dengan perintah sebagai berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button2.Click
    End
End Sub
```



Gambar 5.11 Rancangan form Date Part

Klik Form lalu isi properties text dengan “Date Part” kemudian klik ganda form lalu ketik perintah program untuk menampilkan tanggal dan waktu pada textbox berikut:

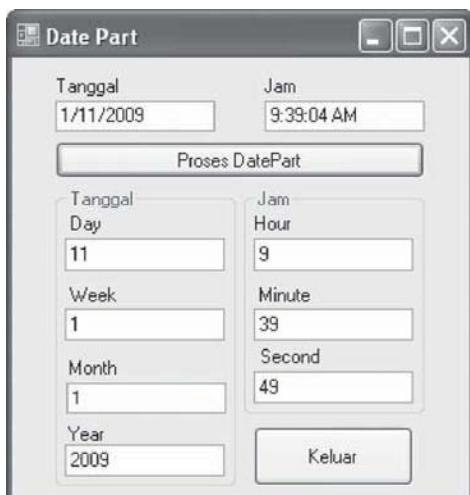
```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles MyBase.Load
    TextBox1.Text = Today
    TextBox2.Text = TimeOfDay
End Sub
```

Klik ganda button proses DatePart lalu ketik perintah untuk menampilkan bagian dari tanggal dan waktu berikut ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    'Mengambil isi/ bagian dari tanggal
    TextBox3.Text = DateAndTime.DatePart(
        DateInterval.Day, Today)
    TextBox4.Text = DateAndTime.DatePart(
        DateInterval.Weekday, Today)
    TextBox5.Text = DateAndTime.DatePart(
        DateInterval.Month, Today)
    TextBox6.Text = DateAndTime.DatePart(
        DateInterval.Year, Today)

    'Mengambil isi / bagian dari waktu
    TextBox7.Text = DateAndTime.DatePart(
        DateInterval.Hour, TimeOfDay)
    TextBox8.Text = DateAndTime.DatePart(
        DateInterval.Minute, TimeOfDay)
    TextBox9.Text = DateAndTime.DatePart(
        DateInterval.Second, TimeOfDay)
End Sub
```

Setelah perintah selesai Anda ketik, jalankan program dengan menekan tombol F5 untuk mengecek perintah yang sudah dibuat.



Gambar 5.12 Hasil program Date Part

5.3 Menampilkan Tanggal

5.3.1 DateSerial

Fungsi perintah DateSerial adalah untuk mengirimkan nilai angka tahun, angka bulan, dan angka tanggal ke dalam suatu variabel date. Nilai tahun yang dimasukkan antara 1 s.d. tahun 9999. Nilai bulan yang dimasukkan antara 1 s.d. 12, dan nilai tanggal yang dimasukkan antara 1 s.d. 31.



Gambar 5.13 Desain form DateSerial

Masukkan label dan textbox year, month, dan day. Masukkan button proses date serial. Masukkan label dan textbox hasil. Klik ganda button Proses Date Serial kemudian ketik perintah program berikut ini:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button1.Click
        'Variabel nTanggal dengan tipe data Date
        Dim nTanggal As Date

        'nTanggal diisi dengan tanggal yang diambil dari '
        'nilai pada textBox1, textBox2 dan textBox3
        nTanggal = DateAndTime.DateSerial(Val(textBox1.Text),
            Val(textBox2.Text), Val(textBox3.Text))

        'Isi textBox4 dengan nilai variabel nTanggal
        TextBox4.Text = CDate(nTanggal)
    End Sub
End Class
```

Jalankan program dengan menekan tombol F5 lalu isi program dengan input tahun, bulan, dan tanggal yang akan dimasukkan dalam variabel dengan perintah dateserial.



Gambar 5.14 Hasil program Date Serial

5.3.2 DateValue

Fungsi perintah DateValue adalah untuk mengubah sebuah string menjadi tipe data tanggal atau tipe data waktu. Parameter yang digunakan untuk membuat tanggal bertipe data string seperti pada contoh perintah berikut ini:

```
Dim oldDate As Date  
oldDate = DateValue("April 10, 2009")
```

Selain dengan menggunakan tipe data string tanggal yang akan dibuat dengan menggunakan DateValue, dapat juga dibuat menggunakan tanda # seperti terlihat pada contoh berikut:

```
Dim OldDate As Date  
oldDate = #23/11/2008#
```

5.3.3 MonthName

Fungsi perintah MonthName adalah untuk menampilkan character bulan dari nilai bilangan bulat yang dimasukkan. Nilai False akan menampilkan character bulan secara lengkap, sedangkan nilai True akan menampilkan nama bulan secara singkat atau pendek.



Gambar 5.15 Program Month Name

Masukkan label dan textbox tanggal. Masukkan komponen label dan textbox hari. Masukkan komponen label dan textbox bulan. Masukkan button proses dan button Keluar ke dalam form. Klik ganda form dan buatlah perintah berikut ini:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load  
    TextBox1.Text = Today  
End Sub
```

Klik ganda button Proses kemudian ketik perintah program berikut ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
    TextBox2.Text = Format(Now(), "ddddd")  
    TextBox3.Text = MonthName(Month(Today), False)  
End Sub
```

Perintah format(Now(), "dddd") digunakan untuk menampilkan nama hari dari tanggal sistem yang aktif, sedangkan perintah month (today) digunakan untuk mengambil angka bulan pada tanggal sistem komputer sekarang. Jalankan program dengan menekan tombol F5 kemudian klik tombol Proses.



Gambar 5.16 Hasil program MonthName

<http://pustaka-indo.blogspot.com>

5.3.4 WeekDayName

Fungsi perintah WeekDayName adalah untuk menampilkan nama hari dalam satu minggu. Parameter perintah WeekDayName adalah angka hari yang dibuat dengan tipe data integer. Angka yang dimasukkan dimulai dari angka 1 sebagai hari pertama dalam satu minggu sampai dengan angka 7 sebagai tanda hari terakhir dalam satu minggu tersebut.

Sedangkan perintah WeekDay untuk menampilkan angka hari dalam satu minggu dimulai dari angka 0, 1, sampai dengan 7. Angka 0 adalah untuk pengaturan sesuai dengan setting komputer, angka 1 untuk Sunday (Minggu).



Gambar 5.17 Form WeekDayName

Masukkan Label untuk tanggal lahir. Masukkan komponen DateTimePicker. Masukkan label dan textbox hari. Masukkan label dan textbox choose untuk mencari hari dengan perintah choose. Kemudian klik ganda komponen DateTimePicker lalu ketik perintah program berikut ini:

```
Private Sub DateTimePicker1_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DateTimePicker1.ValueChanged
    Dim IntHari As Double
    'Menampilkan Nama Hari dengan perintah WeekDayName
    TextBox1.Text = DateAndTime.WeekdayName(Weekday(
        DateTimePicker1.Value))
    'Mencari interval hari
    IntHari = DateAndTime.DateDiff(DateInterval.Day,
        DateTimePicker1.Value, Now)
    'Hari dengan perintah choose
    TextBox2.Text = Choose(Weekday(DateTimePicker1.Value),
        "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu",
        "Minggu")
End Sub
```

Mencari hari dengan menggunakan perintah WeekDayName sama jika menggunakan perintah Choose, hanya saja perintah choose lebih panjang, karena harus mengetikkan nama hari dalam suatu array hari. Anda dapat memilih salah satu perintah yang akan Anda gunakan untuk membuat perintah program mencari nama hari.

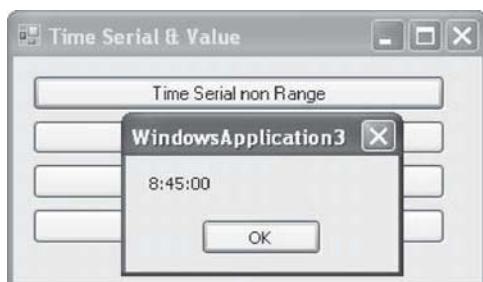


Gambar 5.18 Hasil program WeekDayName

5.4 Mengatur Tanggal dan Waktu

5.4.1 TimeSerial

Fungsi perintah TimeSerial adalah untuk mengatur waktu sistem sesuai dengan jam, menit, dan detik yang Anda inginkan. Untuk memasukkan jam, menit, detik dengan tipe data integer. Angka yang dapat dimasukkan untuk jam antara 00 sampai dengan 23. Angka untuk menit antara 00 sampai dengan 59. Angka untuk detik antara 0 sampai dengan 59. Sebenarnya angka di luar range tersebut juga masih dapat diterima oleh perintah TimeSerial, namun sebaiknya menggunakan angka-angka yang terdapat di dalam range.



Gambar 5.19 Time Serial Non Range

Masukkan button Time Serial non range ke dalam form untuk mencoba perintah yang tidak sesuai dengan range, baik angka negatif atau angka di luar range. Klik ganda button lalu ketik perintah berikut ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim alarmTime As Date = TimeSerial(12 - 3, -15, 0)
    MsgBox(alarmTime)

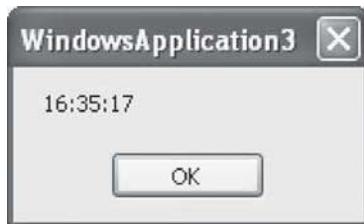
    Dim alarmTime1 As Date = TimeSerial(102 - 63, -65, 70)
    MsgBox(alarmTime1)
End Sub
```

Jalankan program dengan menekan tombol F5 lalu klik button Time Serial. Program tersebut masih berjalan secara normal, namun terjadi pengurangan terlebih dahulu di dalam program. Angka 12 (dua belas) akan dikurangi dengan angka 3 (tiga) sehingga hasilnya menjadi 8 (delapan). Ini menunjukkan bahwa jam yang diatur adalah jam 8 (delapan). Angka -15 (minus lima belas) menunjukkan bahwa total angka menit yaitu 60 (enam puluh) dikurangi dengan -15 sehingga menit yang muncul adalah 45.

```
MsgBox(TimeSerial(23, 75, 0))
' Hasil "1/2/0001 12:15:00 AM".
```

Apabila perintah di atas dibuat dalam program, maka jam yang akan tampil adalah jam 12:15:00.

```
Dim thisTime As Date
thisTime = TimeSerial(16, 35, 17)
```



Gambar 5.20 MsgBox Time Serial

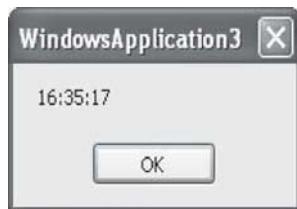
5.4.2 TimeValue

Fungsi perintah TimeValue adalah untuk mengubah nilai waktu string menjadi waktu dalam bentuk angka. Angka yang dapat dimasukkan hampir sama dengan range yang digunakan pada perintah TimeSerial, tetapi pada perintah TimeValue kita dapat memasukkan string dengan argumen yang lain namun menghasilkan nilai yang sama.

Sebagai contoh, jam “2:24 PM” dapat kita buat dengan angka string ”14:24”, kedua string tersebut diperbolehkan dalam perintah TimeValue. Namun apabila terdapat string yang meliputi bulan dalam bentuk string, maka akan terjadi kesalahan dalam pembuatan perintah program. Misalnya kita membuat “january 32” maka akan terjadi kesalahan. Berikut ini contoh penggunaan perintah TimeValue dalam sebuah program.

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    Dim thisTime As Date
    thisTime = TimeValue("4:35:17 PM")
    MsgBox(thisTime)
End Sub
```

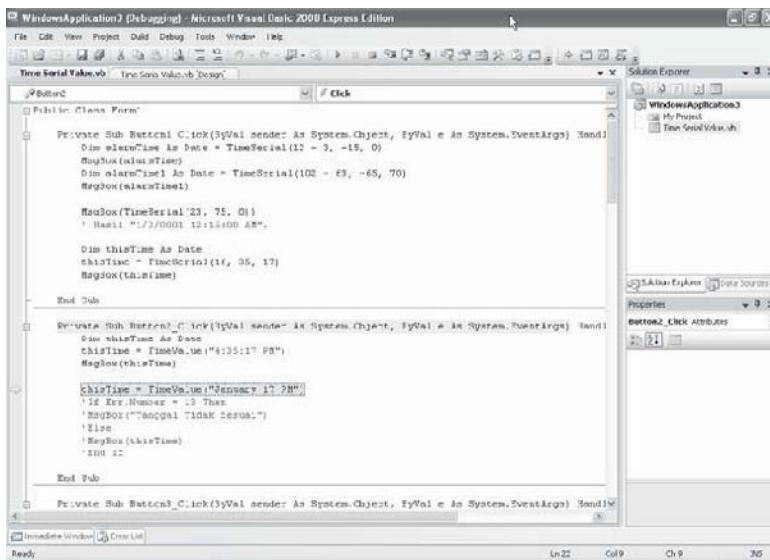
Hasil dari perintah di atas dapat dilihat pada gambar di bawah ini:



Gambar 5.21 Perintah TimeValue

Apabila kita salah dalam membuat perintah untuk time value seperti terlihat pada perintah berikut ini, maka akan muncul kesalahan dalam program.

```
thisTime = TimeValue("January 17 PM")
```



Gambar 5.22 Kesalahan dalam membuat perintah Time Value

<http://bustaka-indo.blogspot.com>

5.4.3 DateString

Fungsi perintah DateString adalah untuk menampilkan tanggal dalam bentuk string dari tanggal sistem komputer yang aktif sekarang.

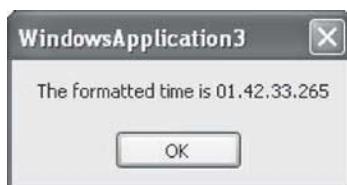
```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim dATEsTRING As String
    dATEsTRING = Format(Now, "dddd, d MMM yyyy")
    MsgBox("The current dateTanggal aktif sekarang adalah "
    & dATEsTRING)
End Sub
```

5.4.4 TimeString

Fungsi perintah TimeString adalah untuk menampilkan jam dalam bentuk string dari jam sistem komputer. Berikut ini contoh perintah untuk menampilkan tanggal sistem komputer dengan menggunakan format untuk waktu.

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    MsgBox("The formatted time is " & Format(Now,
        "hh.mm.ss.fff tt"))
End Sub
```

Format hh untuk jam, mm untuk menit, ss untuk detik, ff untuk seperseratus detik, fff untuk seperseribu detik, dan tt untuk format PM apabila jam antara jam 00 sampai dengan jam 11:59.



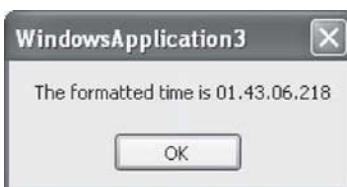
Gambar 5.23 Hasil perintah Format Jam

Berikut ini contoh penggunaan perintah TimeString dengan Message Box:

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    'Dengan Perintah Format
    MsgBox("The formatted time is " & Format(Now,
        "hh.mm.ss.fff tt"))

    'Time String
    MsgBox("The current time is " &TimeString)
End Sub
```

Sehingga akan muncul output program seperti berikut.



Gambar 5.24 Perintah Time String

Waktu yang diambil dengan perintah TimeString sesuai dengan waktu yang diatur pada Regional Setting dalam Control Panel.

5.5 Proses Waktu

5.5.1 Timer

Fungsi Perintah Timer digunakan untuk memberikan nilai banyaknya detik (seperseribu detik) mulai dari hitungan tengah malam.

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    If TimeOfDay >= #11:59:55 PM# Then
        MsgBox("Waktu tengah malam 5 detik lagi" & vbCrLf & "untuk sampai jam 0:0 Tengah malam")
        Return
    End If
    Dim start, finish, totalTime As Double
    If (MsgBox("Tekan Yes untuk berhenti 5 detik",
               MsgBoxStyle.YesNo)) =
        = MsgBoxResult.Yes Then
        start = Microsoft.VisualBasic.DateAndTime.Timer

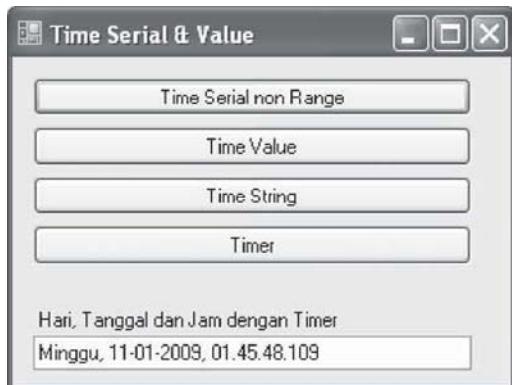
        ' Atur durasi
        finish = start + 5.0
        Do While Microsoft.VisualBasic.DateAndTime.Timer
            < finish
        'Proses Perulangan selama 5 detik
        Loop
        totalTime =
        Microsoft.VisualBasic.DateAndTime.Timer - start
        MsgBox("Berhenti Selama " & totalTime & " detik")
    End If
End Sub
```

Selain pengaturan dengan Timer terdapat komponen Timer yang dapat menampilkan gerak detik, menit, jam, dan tanggal. Untuk menggerakkan jam tersebut maka komponen timer harus diikutsertakan dalam form. Atur properties Enable menjadi TRUE, atur pula interval menjadi 1000, lalu klik ganda komponen timer dan ketik kode program berikut ini:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Timer1.Interval = 1000
    Timer1.Enabled = True

    'Format Menampilkan hari tanggal dan jam
    TextBox1.Text = Format(Now, "dddd, dd-MM-yyy,
                           hh.mm.ss.fff tt")
End Sub
```

Jalankan program yang menggunakan komponen timer dengan menekan tombol F5, sehingga akan terlihat detik dari jam akan bergerak mengikuti gerak pengaturan komponen timer.



Gambar 5.25 Jam dengan komponen Timer

5.5.2 IsDate

Fungsi IsDate adalah untuk memberikan nilai boolean (True/False) dari ekspresi tanggal dan waktu yang dimasukkan dalam suatu variabel data. Berikut ini contoh penggunaan perintah IsDate.

```
'Variabel data
Dim firstDate, secondDate As Date
Dim timeOnly, dateAndTime, noDate As String
Dim dateCheck As Boolean

'Isi variabel FirstDate
firstDate = CDate("February 12, 1969")

'isi Variabel secondDate
secondDate = #2/12/1969#
'Isi variabel timeOnly
timeOnly = "3:45 PM"

Isi Variabel dateAndTime
dateAndTime = "March 15, 1981 10:22 AM"

'isi Variabel noDate
noDate = "Hello"
```

```
'cek hasil variabel Data dengan IsDate
dateCheck = IsDate(firstDate)
dateCheck = IsDate(secondDate)
dateCheck = IsDate(timeOnly)
dateCheck = IsDate(dateAndTime)
dateCheck = IsDate(noDate)
MsgBox("FirstDate: " & IsDate(firstDate) & vbCrLf & _
"SecondDate: " & IsDate(secondDate) & vbCrLf & _
"timeOnly: " & IsDate(timeOnly) & vbCrLf & _
"dateAndTime: " & IsDate(dateAndTime) & vbCrLf & _
"noDate: " & IsDate(noDate))
```

Sehingga hasil dari perintah IsDate tersebut terlihat seperti gambar di bawah ini:



Gambar 5.26 IsDate dengan MsgBox



BERMAIN DENGAN INTERACTION DAN CONSTANTS MODULE

<http://pustaka-indo.blogspot.com>

Interaction module berisi perintah yang berkaitan dengan pesan, aplikasi, object, dan system. Constants Module berisi tentang perintah program yang tetap dan dapat digabungkan dengan perintah-perintah Microsoft Visual Basic 2008 yang lain.

6.1 Interaction Module

6.1.1 AppActive dan Shell

Fungsi perintah AppActive adalah untuk mengaktifkan aplikasi yang sudah dijalankan. Sedangkan fungsi perintah shell adalah untuk menjalankan aplikasi yang belum aktif. Untuk lebih jelasnya buatlah perintah program berikut ini dalam sebuah form menggunakan button.

```

Private Sub Button1_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button1.Click
 On Error Resume Next
 Dim notepadID As Integer
 ' Mengaktifkan Aplikasi NotePad.
 AppActivate("Untitled - Notepad")
 If Err.Number = 5 Then
 MsgBox("Title atau ProcessId Tidak ditemukan")
 End If

 ' Memanggil aplikasi notepad
 notepadID = Shell("C:\WINDOWS\System32\NOTEPAD.EXE",
 AppWinStyle.NormalFocus)

 ' Aktikan dengan integer notepadID
 AppActivate(notepadID)
End Sub

```

Buatlah perintah di atas dalam sebuah button kemudian jalankan program dengan menekan tombol F5, maka aplikasi Notepad akan diaktifkan.

6.1.2 Beep

Fungsi perintah beep adalah untuk memunculkan suara tone menggunakan speaker komputer. Berikut ini contoh penggunaan perintah beep dalam sebuah program.

```

Private Sub Button2_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button2.Click
 Dim I As Integer
 For I = 1 To 1000 ' Loop 1000 kali.
 Beep() ' Sound a tone.
 Next I
End Sub

```

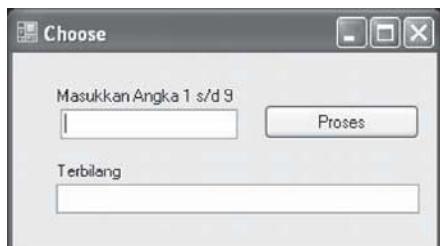
Bunyi beep akan diulangi sebanyak 1 s.d. 1000 kali dengan menggunakan perintah perulangan For dan Next.

6.1.3 Choose

Fungsi perintah Choose adalah untuk mengambil data dari array atau daftar argumen yang berurutan berdasarkan index yang bertipe double.

Berikut ini contoh penggunaan perintah choose yang dimasukkan dalam sebuah variabel string.

```
Dim GetChoice as String  
GetChoice = CStr(Choose(Ind, "Speedy", "United",  
"Federal"))
```



Gambar 6.1 Desain program Choose

Rancanglah form di atas dengan memasukkan komponen label dan textbox “masukkan angka 1 s/d 9”. Masukkan label dan textbox terbilang. Masukkan button proses. Klik ganda button Proses kemudian ketik perintah program berikut ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
    TextBox2.Text = Interaction.Choose(Val(TextBox1.Text),  
    "Satu", "Dua", "Tiga", "Empat", "Lima", "Enam", "Tujuh",  
    "Delapan", "Sembilan")  
End Sub
```

Jalankan program dengan menekan tombol F5 kemudian inputkan angka 1 s.d. 9.

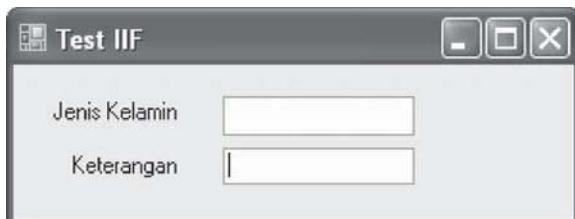


Gambar 6.2 Hasil program Choose

6.1.4 IIF

Fungsi perintah IIF adalah untuk membandingkan satu atau dua ekspresi dan memberikan nilai benar (TRUE) atau salah (False). Apabila ekspresi sesuai, maka akan memberikan nilai TRUE, atau nilai FALSE sebagai alternatif jawaban. Perintah ini mirip dengan penggunaan fungsi logika If ..EndIf, perhatikan contoh di bawah ini:

```
Function checkJenisKElamin(ByVal testMe As String) As String
    Return CStr(IIf(testMe = "L", "Laki-laki", "Perempuan"))
End Function
```



Gambar 6.3 Desain program IIF

Masukkan label dan textbox jenis kelamin. Masukkan pula label dan textbox keterangan. Klik ganda textbox jenis kelamin dan ketik perintah program berikut:

```
Public Class Form1
Function checkJenisKElamin(ByVal testMe As String) As String
    Return CStr(Interaction.IIf(testMe = "L", "Laki-laki",
        "Perempuan"))
End Function

Private Sub TextBox1_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    TextBox1.TextChanged
    TextBox2.Text = checkJenisKElamin(TextBox1.Text)
End Sub
End Class
```

Jalankan program dengan mengklik menu debugging dan pilih submenu *start debugging* kemudian isi textbox jenis kelamin dengan “L” atau “P”. Hasil yang terlihat seperti Gambar 6.4.



Gambar 6.4 Hasil program IIF

Perintah program di atas dapat pula dibuat dalam bentuk lain seperti contoh berikut:

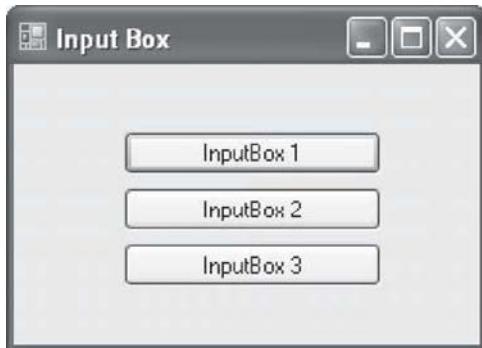
```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    'TextBox2.Text = checkJENisKElamin(TextBox1.Text)
    'Hasil sama dengan perintah berikut
    TextBox2.Text = Interaction.IIf( TextBox1.Text= "L",
                                    "Laki-laki", "Perempuan"))
End Sub
```

6.1.5 InputBox

Fungsi perintah InputBox adalah untuk menampilkan kotak input. Panjang text yang dapat diinputkan sebanyak 1024 character dan dapat dipisahkan dengan CHR(13) apabila baris terlalu panjang. Parameter perintah inputBox adalah textpesan, judul input box, dan posisi x dan y kotak dialog inputbox. Berikut ini contoh penggunaan perintah inputbox:

```
Dim message, title, defaultValue As String
Dim myValue As Object
message = "Enter a value between 1 and 3"
title="Demo"
defaultValue = "1" ' Set default value.

myValue = InputBox(message, title)
myValue = InputBox(message, title, defaultValue)
myValue = InputBox(message, title, defaultValue, 100, 100)
```



Gambar 6.5 Desain form InputBox

Masukkan tiga buah button untuk inputBox1, InputBox2, dan InputBox3 kemudian klik button InputBox1 dan ketik perintah di bawah ini:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim Ib1 As String
    Ib1 = Interaction.InputBox("Masukkan Nama Anda",
    "Input Box")
End Sub
```

Klik ganda button InputBox2 kemudian ketik perintah berikut ini:

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    Dim Ib2 As String
    Dim DefaultValue As String = "Hendrayudi"
    Ib2 = Interaction.InputBox("Masukkan Nama Anda",
    "Input Box", DefaultValue)
End Sub
```

Klik ganda button InputBox3 kemudian ketik perintah berikut ini:

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    Dim Ib3 As String
    Dim DefaultValue As String = "Hendrayudi"
    Ib3 = Interaction.InputBox("Masukkan Nama Anda" &
    Chr(10) & Chr(13) &
    "(Beserta gelar apabila ada)",
    "Input Box", DefaultValue, 100, 100)
End Sub
```



Gambar 6.6 Hasil program InputBox

Klik button Ok apabila input box sudah selesai diisi, dan klik button Cancel apabila Anda ingin membatalkan input data.

6.1.6 MsgBox

Fungsi perintah MsgBox adalah untuk menampilkan kotak dialog pesan dengan button-button yang mengirimkan nilai integer apabila diklik salah satu tombol. Parameter yang terdapat dalam perintah MsgBox diantaranya adalah pesan yang akan ditampilkan (dapat pula menggunakan chr(13) untuk pindah baris), button-button yang akan digunakan, dan judul kotak pesan.

Berikut ini contoh perintah menggunakan MsgBox. Button-button yang dapat ditampilkan dalam bentuk constanta OkOnly, OKCancel, AbortRetryIgnore, YesNoCancel, YesNo, RetryCancel, Critical, Question, Exclamation, Information, DefaultButton1, DefaultButton2, DefaultButton3, dan Application Modal.

Nilai Button OK adalah 1, Cancel adalah 2, Abort adalah 3, Retry adalah 4, Ignore adalah 5, Yes adalah 6, dan nilai button No adalah 7. Berikut ini contoh penggunaan perintah msgbox:

```
Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    MsgBox("Message Box pertama")
End Sub
```



Gambar 6.7 Desain form Pesan

Untuk lebih lengkapnya, desain form di atas dengan memasukkan komponen button Contoh Pesan 1, Contoh Pesan 2, dan Contoh Pesan 3. Klik ganda button Pesan 1 lalu ketiklah perintah di atas. Klik ganda button 2 kemudian ketik perintah berikut:

```
Private Sub Button2_Click(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles Button2.Click  
    Dim m As String = " ---- Contoh tampilan Pesan ---- "  
    Dim j As String = "judul Message"  
    MsgBox(m, , j)  
End Sub
```

Klik ganda button contoh pesan 3 lalu ketik perintah berikut ini:

```
Private Sub Button3_Click(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles Button3.Click  
    Dim m As String = " ---- Contoh tampilan Pesan ---- "  
    Dim j As String = "judul Message"  
    MsgBox(m, MsgBoxStyle.AbortRetryIgnore, j)  
End Sub
```

Jalankan program dengan menekan tombol F5 kemudian klik button contoh pesan 3 serta button lainnya.



Gambar 6.8 Hasil program pesan

Untuk memasukkan program untuk menyeleksi tombol yang ditekan, gunakan perintah IF..EndIf seperti contoh progam berikut ini:

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    Dim m As String = " === Contoh tampilan Pesan === "
    Dim j As String = "judul Message"
    Dim nilai As Integer
    nilai = MsgBox(m, MsgBoxStyle.AbortRetryIgnore, j)

    If nilai = vbAbort Then
        MsgBox("Anda Memilih tombol Abort")
    ElseIf nilai = vbRetry Then
        MsgBox("Anda Memilih tombol Retry")
    ElseIf nilai = vbIgnore Then
        MsgBox("Anda Memilih tombol Ignore")
    End If
End Sub
```

6.2 Constants Module

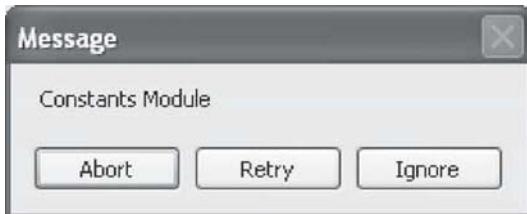
6.2.1 vbAbort

vbAbort adalah perintah yang terdapat pada constants module yang digunakan untuk mengambil nilai pembatalan (abort) yang sering digunakan bersamaan dengan perintah MsgBox.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim Pesan As String
    Dim Ket As String

    Pesan = MsgBox("Constants Module",
MsgBoxStyle.AbortRetryIgnore, "Message")

    'Cek tombol yang di pilih
    If Pesan = vbAbort Then
        Ket = "Anda memilih Abort"
        TextBox1.Text = Ket
    ElseIf Pesan = vbRetry Then
        Ket = "Anda memilih Retry"
        TextBox1.Text = Ket
    ElseIf Pesan = vbIgnore Then
        Ket = "Anda Memilih Ignore"
        TextBox1.Text = Ket
    End If
End Sub
```



Gambar 6.9 vbAbort, vbRetry, dan vbIgnore

6.2.2 Constants Module lain

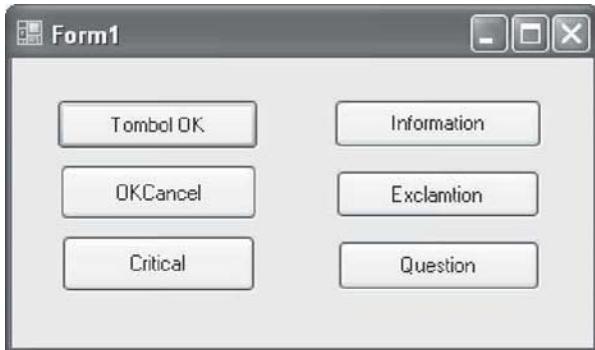
Constants Module yang lain penggunaannya sama dengan constants module vbAbort. Berikut ini constants module yang sering digunakan bersamaan dengan perintah messageBox:

Tabel 1. Message Box Style Constants Module

Member	Constants	Keterangan
OKOnly	vbOKOnly	Button OK
OKCancel	vbOKCancel	Button OK dan Cancel
AbortRetryIgnore	vbAbortRetryIgnore	Button Abort, Retry dan Ignore
YesNoCancel	vbYesNoCancel	Button Yes, No dan Cancel
RetryCancel	vbRetryCancel	Button Retry dan Cancel
Critical	vbCritical	Pesan
Question	vbQuestion	Pesan Peringatan
Exclamation	vbExclamation	Pesan Peringatan
Information	vbInformation	Pesan Informasi
DefaultButton1	vbDefaultButton1	Button Pertama standart
DefaultButton2	vbDefaultButton2	Button Kedua standart
DefaultButton3	vbDefaultButton3	Button Ketiga standart
ApplicationModal	vbApplicationModal	Kotak pesan Aplikasi

SystemModal	vbSystemModal	Message Box System Modal
MessageBoxRight	vbMessageBoxRight	Right Align Text

Contoh:

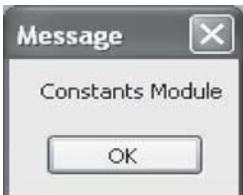


Gambar 6.10 Message box Style

Buatlah desain form seperti terlihat pada gambar di atas dengan cara memasukkan komponen button dan isilah text dengan tulisan Tombol OK, lalu klik ganda dan ketiklah perintah berikut:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim Pesan As String
    'Tombol OK
    Pesan = MsgBox("Constants Module", MsgBoxStyle.OkOnly,
    "Message")
End Sub
```

Jalankan program tersebut dengan menekan tombol F5, lalu klik button OK, maka akan tampak tombol yang dibuat pada kotak message dengan menggunakan style OKOnly.



Gambar 6.11 Constant Modul OkOnly

Klik button OK dan tutup aplikasi, kemudian tambahkan button OKCancel. Klik ganda button OkCancel dan ketiklah perintah berikut ini:

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    Dim Pesan As String
    'Tombol OKCancel
    Pesan = MsgBox("Constants Module",
    MsgBoxStyle.OkCancel, "Message")
End Sub
```

Jalankan kembali program dengan menekan tombol F5 lalu klik button OKCancel. Akan tampil message box seperti gambar di bawah ini:



Gambar 6.12 Style OKCancel

Tambahkan lagi button Critical, Information, Exclamation, dan Question untuk mencoba bentuk constants module dalam message box, seperti terlihat pada Gambar 6.10. Kemudian buatlah perintah program untuk masing-masing button tersebut.

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
    Dim Pesan As String
    'Tombol Critical
    Pesan = MsgBox("Constants Module",
    MsgBoxStyle.Critical, "Message")
End Sub
```



Gambar 6.13 Message box Critical

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click
    Dim Pesan As String
    'Tombol Information
    Pesan = MsgBox("Constants Module",
    MsgBoxStyle.Information, "Message")
End Sub
```



<http://pusataindo.blogspot.com>

Gambar 6.14 Message box Style Information

```
Private Sub Button5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button5.Click
    Dim Pesan As String
    'Tombol Exclamation
    Pesan = MsgBox("Constants Module",
    MsgBoxStyle.Exclamation, "Message")
End Sub
```



Gambar 6.15 Message box style Exclamation

```
Private Sub Button6_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button6.Click
    Dim Pesan As String
    'Tombol Question
    Pesan = MsgBox("Constants Module",
    MsgBoxStyle.Question, "Message")
End Sub
```



Gambar 6.16 Message box Style Question

<http://pustaka-indo.blogspot.com>



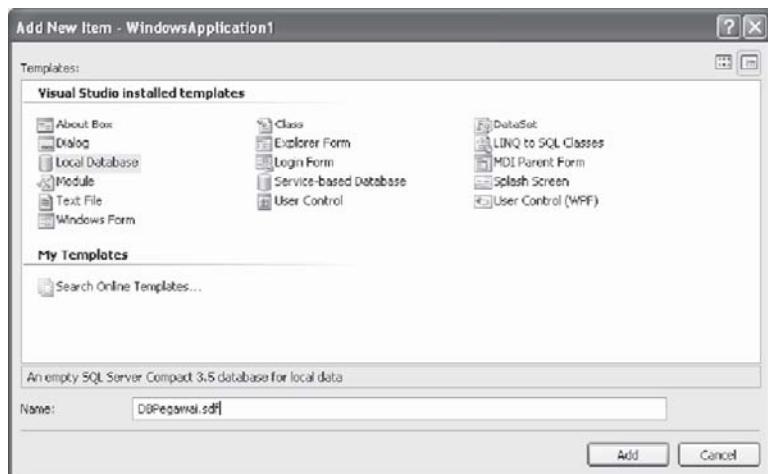
BERMAIN DENGAN TABLE

<http://pustaka-indo.blogspot.com>

Tabel adalah kumpulan field-field yang berisi nama, jenis, lebar data, dan kunci field. Untuk membuat nama field harus unik, maksudnya nama field tidak boleh sama dengan nama field lain yang terdapat di dalam tabel. Syarat dalam membuat nama field: nama field boleh disingkat misalnya kode barang menjadi kdbrg. Untuk field yang menggunakan spasi digantikan dengan tanda underscore (_), misalnya nama barang menjadi Nm_Brg.

7.1 Membuat Tabel

Untuk membuat tabel, Anda klik menu Project pilih Add New Items lalu Klik Local Database dan ketik nama database misalnya DB-Pegawai kemudian klik Add.



Gambar 7.1 Membuat nama database

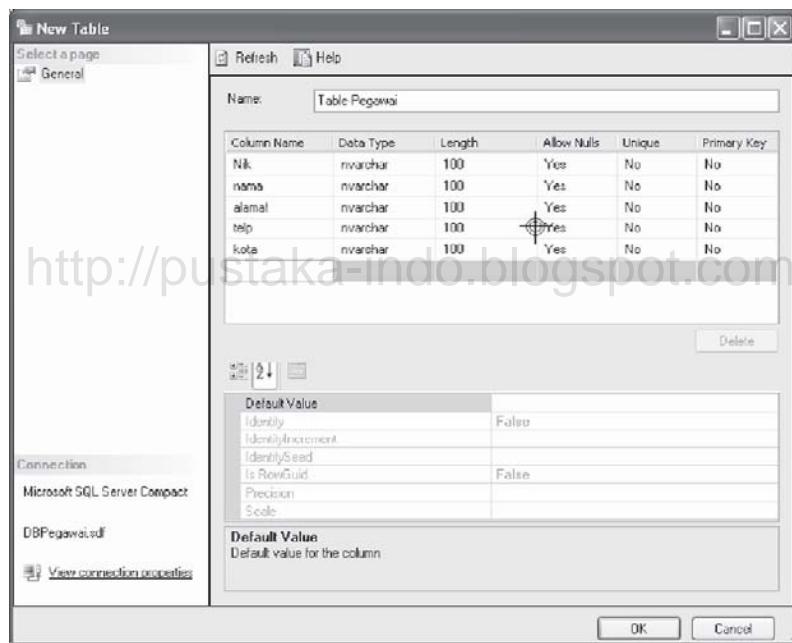
Setelah Anda membuat nama database, klik Finish untuk mengakhiri pembuatan nama database.



Gambar 7.2 Nama Dataset

Klik ganda nama database (DB-Pegawai.SDF) pada Solution Explorer yang terdapat di sebelah kanan layar untuk menampilkan database explorer (di bagian kanan layar). Pada database explorer klik plus (+), kemudian pada database, Anda klik kanan pada table lalu klik Create Table dan buat nama table pada kotak Name.

Kemudian buatlah nama-nama field (nik, nama, alamat, telp, dan kota) dan primary key seperti yang terlihat pada gambar di bawah ini. Apabila sudah selesai membuat nama field, klik OK untuk mengakhiri.



Gambar 7.3 Membuat tabel baru

7.1.1. Menambahkan Data

Untuk menambahkan data pada tabel, klik kanan nama tabel lalu klik **Show Table Data**, kemudian isi data pada tabel Tbl-Pegawai. Untuk menghapus data atau satu baris record data, klik bagian kiri table untuk memblok record yang akan dihapus, kemudian klik kanan dan pilih delete atau tekan tombol delete pada keyboard.

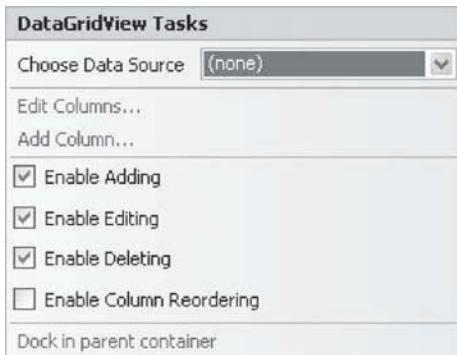
Nik	nama	alamat	telp
001	HEndrayudi	Jl. Hamka	0735 87464
002	Haris Saputra	Jl. Jogja	0735 747463
003	Eka Herviaty	Jl. Lintas Sumatera	0735 098787
004	Rusidi	Jl. Sumber Asri	0735 645321
NULL	NULL	NULL	NULL

<http://pustaka-indo.blogspot.com>

7.1.2. Menggunakan DataGridView

Data grid view adalah sebuah tabel yang memiliki sejumlah baris dan kolom untuk menampilkan data. Untuk memasukkan data grid, klik **Data Grid View** pada toolbox kemudian klik ke form seperti yang tampak pada Gambar 7.5.

Pilih Choose DataSource lalu klik **Add Project Data Source**, kemudian klik Next. Lanjutkan dengan mengklik Finish dan klik ikon Start untuk menjalankan program. Lihat Gambar 7.6.



Gambar 7.5 Data Grid View

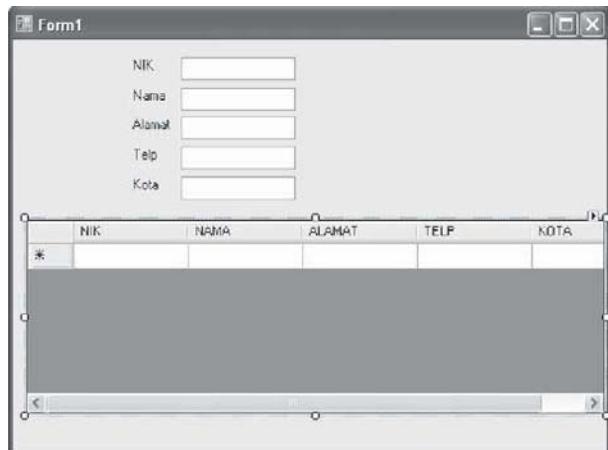
The screenshot shows a Windows application window titled 'Form1'. Inside the window is a DataGridView control displaying a table of data. The columns are labeled 'NIK', 'NAMA', 'ALAMAT', 'TELP', and 'KOTA'. The data rows are:

NIK	NAMA	ALAMAT	TELP	KOTA
0001	HARIS SAPUTRA	JL. AHMAD YANI 123	0735326789	BATURAJA
0002	HENDRAYUDI	JL. HAMKA 546	0735765432	BATURAJA
0003	RUSIDI	JL. BELITANG III	0735325654	BELITANG
0004	BUDI KURNIAWAN	JL. TALANG JAWA	0735532305	BELITANG
*				

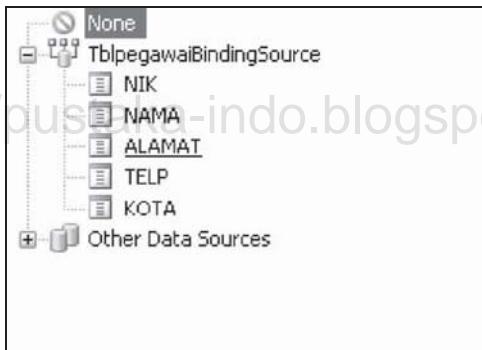
Gambar 7.6 Data Grid View

7.1.3. Menghubungkan Textbox dengan Table

Masukkan lima buah textbox, yaitu textbox nik, nama, alamat, telp, dan kota. Klik Textbox kemudian isi Properties Data Binding, pilih Advanced (...). Pilih Nama Table pada Binding, pilih nama field yang akan dihubungkan dengan textbox. Apabila sudah selesai klik OK.

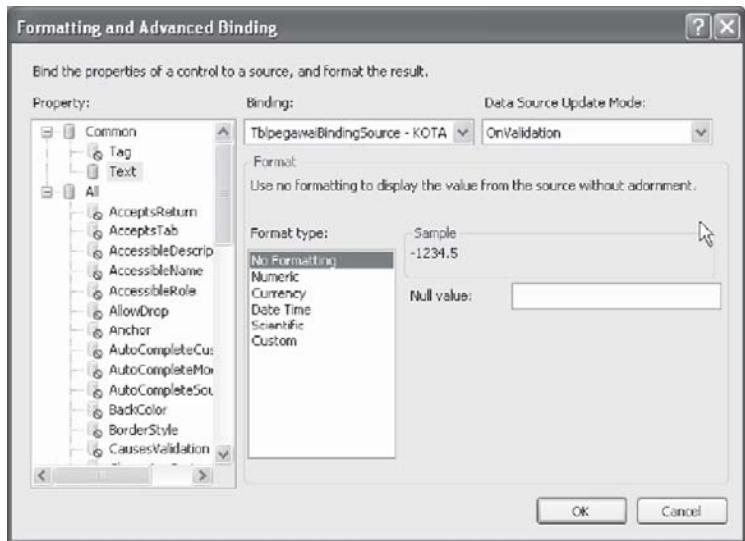


Gambar 7.7 Desain Textbox



Gambar 7.8 Data Binding Advanced

Pilih format type untuk membuat jenis format yang akan dimasukkan ke dalam textbox. Format Numeric untuk mengatur format angka, Currency untuk mengatur format mata uang, Date Time untuk mengatur format tanggal dan waktu, Scientific untuk mengatur format bilangan, dan juga Custom untuk mengatur format sesuai dengan keinginan kita sendiri. Apabila sudah selesai memilih jenis format data Anda, klik button OK.



Gambar 7.9 Format dan Advance Binding

Untuk melihat hasil dari pengaturan format angka, format mata uang, tanggal dan waktu, serta format lainnya, gunakan sample. Untuk memasukkan data untuk format standar, gunakan kotak input null value.

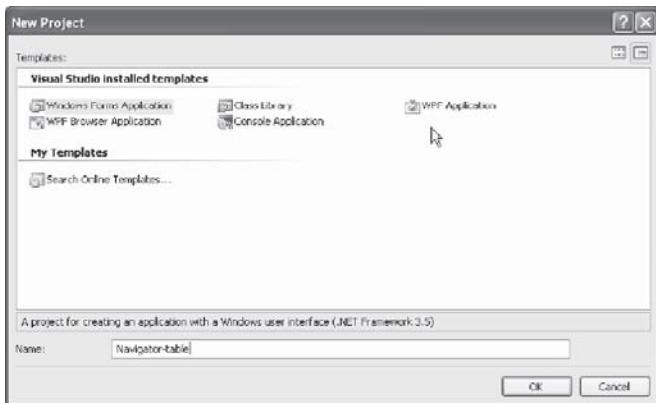
7.2 Button Navigator

Button navigator adalah button yang berisi penunjuk record awal, akhir, sebelum dan sesudah, button tambah, button hapus, dan button simpan.



Gambar 7.10 Button Navigator

Untuk memasukkan button navigator di atas, caranya buat file project baru (File > New Project). Kemudian pilih windows form application lalu buat nama project, misalnya “navigator tabel” dan klik OK.



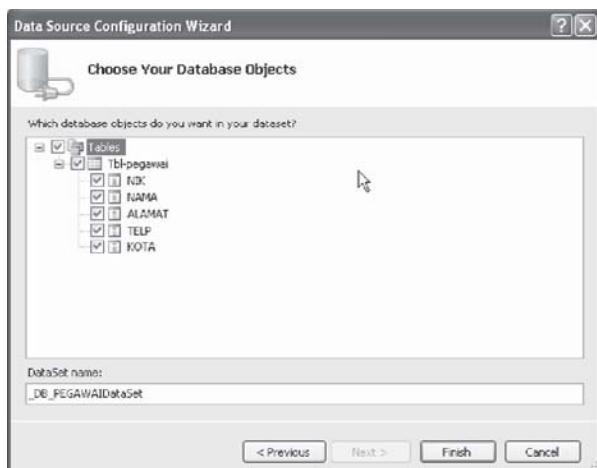
Gambar 7.11 New Project Navigator Tabel

Setelah form muncul, klik Data Source lalu klik Add New Data Source. Pilih database dan klik Next. Pilih database yang akan dihubungkan dengan form dengan cara mengklik New Connection. Pilih Browse pada textbox database dan pilih database yang sudah dibuat terdahulu kemudian klik OK.



Gambar 7.12 Add Connection

Klik Next lalu muncul pilihan Save Connection String, klik Next lagi. Pada pilihan *Choose Your Database Object* tandai pilihan table dengan tanda check kemudian klik Finish.



Gambar 7.13 Choose your database object

Perhatikan form sebelah kanan atas (pada Data Source) sudah terdapat tabel/database yang Anda pilih tadi. Untuk memasukkan navigator dan grid ke dalam form, klik nama tabel lalu drag ke form.



Gambar 7.14 Data Sources Table Pegawai

Setelah Anda drag ke form, klik DataGridView Task kemudian pilih Dock in Parent Container agar grid mengikuti lebar form. Setelah semua selesai Anda tekan tombol F5 untuk menjalankan aplikasi.



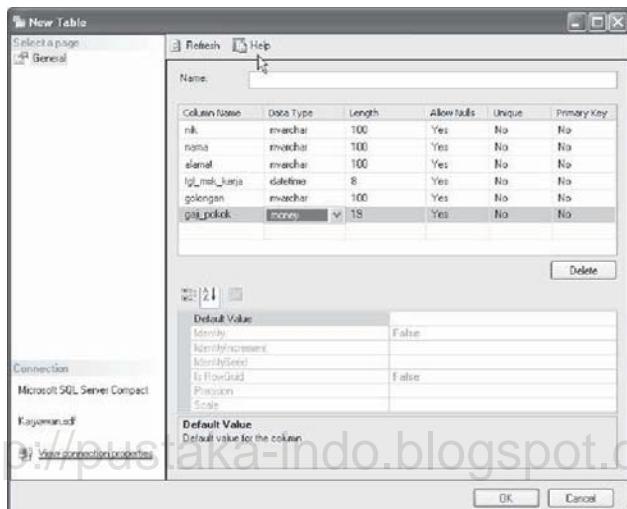
Gambar 7.15 Hasil membuat button navigator

7.3 Kontrol Format TextBox

Format textbox adalah pengaturan bentuk input data yang sesuai dengan yang Anda inginkan. Bentuk kesesuaian data yang diinginkan dapat berupa data numeric, currency, datetime, scientific, dan custom. Untuk membuat format tersebut, buatlah sebuah tabel karyawan yang terdiri dari field-field NIK, Nama, Alamat, Tgl_masuk_Kerja, Golongan, dan Gaji.

Untuk membuat tabel tersebut Anda klik menu Project > Add New Items lalu klik Local Database dan ketik nama database misalnya DB-Karyawan. Klik Add. Setelah Anda membuat nama database, klik Finish kemudian klik ganda nama database (DB-Karyawan.SDF) pada Solution Explorer yang terdapat di sebelah kanan layar.

Pada database explorer klik plus (+) kemudian pada database Anda klik kanan pada table, lalu klik New Table dan buat nama table pada kotak Name. Kemudian buatlah nama-nama field dan primary key. Klik OK apabila tabel Karyawan sudah selesai Anda buat.



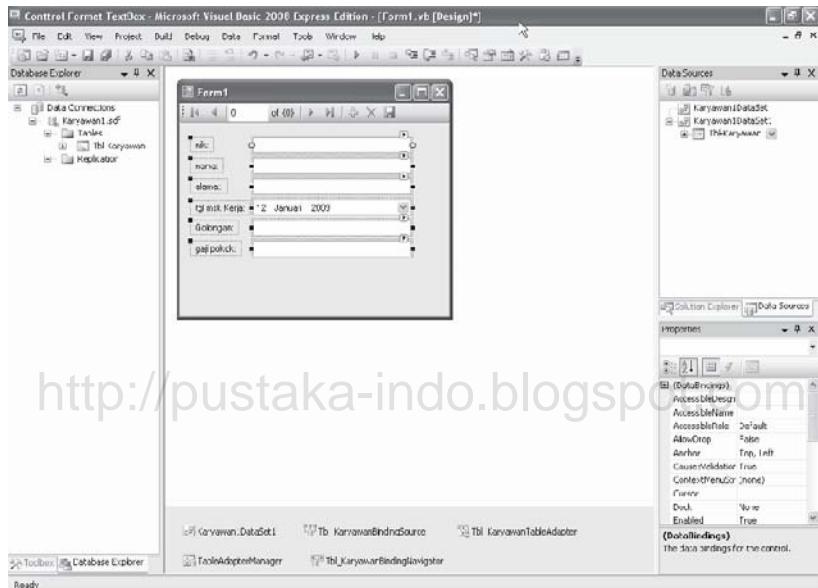
Gambar 7.16 Gambar desain tabel karyawan



Gambar 7.17 Dataset database karyawan

Klik Data Source kemudian Anda klik Add New Data Source. Pilih database dan klik Next. Pada kotak dialog *Choose New Connection* Anda klik Next dan beri tanda check pada tabel lalu klik Finish (Gambar 7.17).

Klik tabel karyawan lalu ubah option menjadi detail, kemudian Anda drag ke form sehingga akan tampil seperti gambar di bawah ini.



Gambar 7.18 Desain format

Untuk mengatur format textbox tanggal masuk kerja, klik properties dan pilih Format. Pada properties format terdapat tiga pilihan, yaitu *long*, *short*, dan *time*. Pilihan format long akan membuat tanggal menampilkan hari dan tanggal. Format short akan menampilkan tanggal, sedangkan time akan menampilkan waktu atau jam.

Untuk format tanggal masuk kerja Anda pilih format short. Untuk gaji pokok buatlah dengan format currency, klik textbox gaji pokok kemudian pilih Data Binding Advanced dan isi format type dengan Currency. Anda klik OK. Jalankan program dengan menekan tombol F5 kemudian isi data dan perhatikan format textbox untuk tanggal masuk kerja dan format textbox untuk gaji pokok.



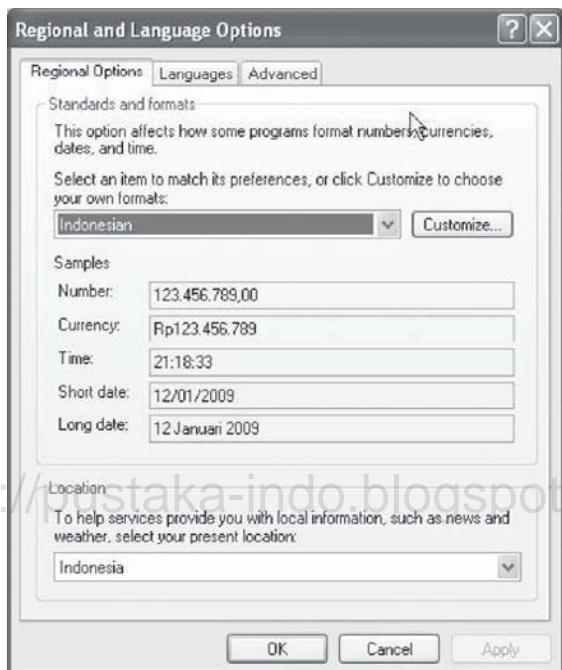
<http://pustaka-indo.blogspot.com>

Apabila Anda menginginkan hari pada tanggal masuk kerja dalam bentuk bahasa Indonesia, atur setting Regional di Control Panel Windows. Klik Start > Control Panel lalu pilih setting **Regional and Language Option**. Pilih Language menjadi Indonesia dan pilih Location menjadi Indonesia. Klik **Apply** dan klik **OK**.



Gambar 7.20 DateTimePicker

Begitu juga dengan gaji pokok, untuk mengganti mata uang menjadi Rp (Rupiah), secara otomatis akan ikut berubah sesuai dengan pengaturan Region and Language Setting.



Gambar 7.21 Regional and Language Options

Untuk melihat perintah yang dibuat oleh Microsoft Visual Basic 2008, Anda dapat mengklik kanan mouse kemudian Anda pilih View Code. Untuk kembali ke form, Anda klik kanan dan pilih View Designer.

```
Private Sub Tbl_KaryawanBindingNavigatorSaveItem_Click(ByVal  
    sender As System.Object, ByVal e As System.EventArgs) Handles  
    Tbl_KaryawanBindingNavigatorSaveItem.Click  
    Me.Validate()  
    Me.Tbl_KaryawanBindingSource.EndEdit()  
    Me.TableAdapterManager.UpdateAll(Me._DB_KaryawanDataSet)  
  
End Sub
```

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    'TODO: This line of code loads data into the
    '_DB_KaryawanDataSet._Tbl_Karyawan' table. You can move, or
    remove it, as needed.

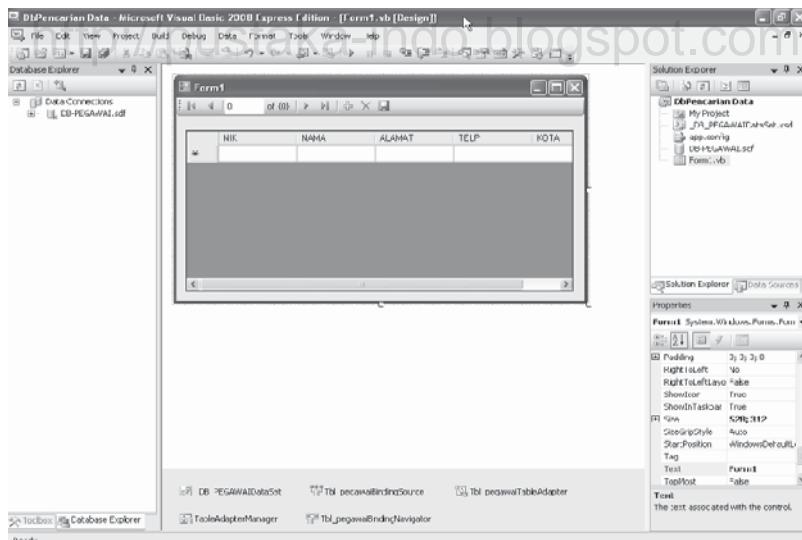
    Me.Tbl_KaryawanTableAdapter.Fill(Me._DB_KaryawanDataSet._Tbl_
    Karyawan)

End Sub
End Class

```

7.4 Pencarian Data

Pencarian data sangat diperlukan untuk mengetahui keberadaan informasi yang utuh dalam sebuah tabel. Untuk membuat pencarian data menggunakan DataSource tabel kita harus membuat form baru. Klik File > New Project, ketik nama project misalnya Pencarian Data kemudian klik button OK.

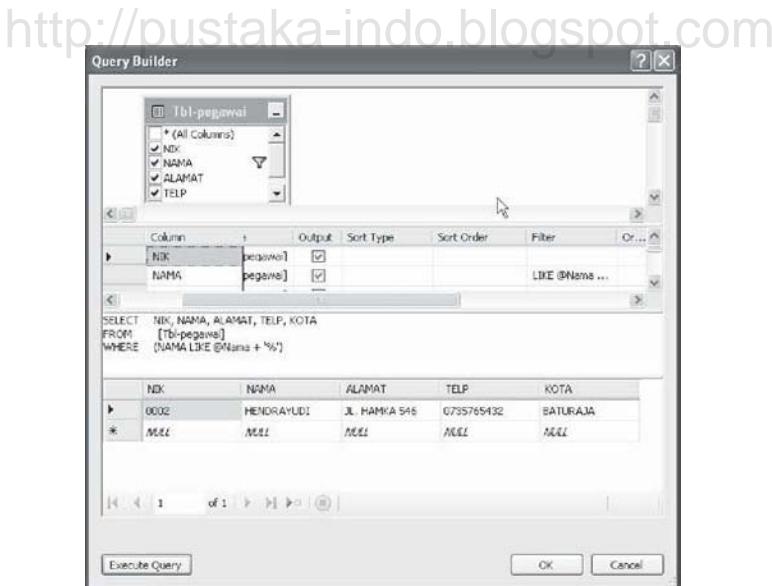


Gambar 7.22 Desain form Pencarian Data

Klik Data Sources kemudian konesikan database dalam bentuk ConnectionString untuk membuat grid. Setelah selesai, atur tabel Dataset dalam bentuk grid lalu drag ke form sehingga tampil seperti Gambar 7.22. Klik DataGridViewTask kemudian Anda pilih Add Query dan buat nama Query Pencarian. Klik Query Builder dan buatlah query pada kolom filter dengan perintah LIKE @Nama + '%'. Maksudnya adalah Nama akan dicari berdasarkan huruf awal yang dimasukkan dalam kriteria pencarian.

```
SELECT NIK, NAMA, ALAMAT, TELP, KOTA  
FROM [Tbl-pegawai]  
WHERE (NAMA LIKE @Nama + '%')
```

Kelompokkan NIK, NAMA, ALAMAT, TELP, KOTA dari tabel Pegawai berdasarkan Nama yang dicari. Untuk menjalankan query, klik Execute Query kemudian masukkan nama pada kolom null. Klik OK. Apabila pencarian nama ditemukan, maka tabel akan menampilkan nama tersebut seperti terlihat pada gambar di bawah ini.



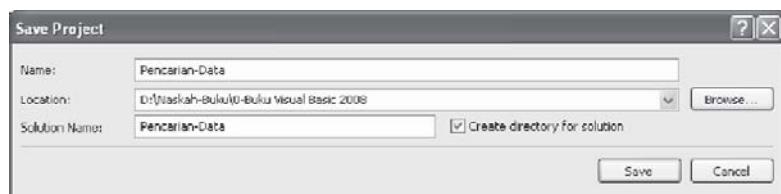
Gambar 7.23 Query Builder pencarian data

Apabila Query sudah selesai dibuat, klik button OK. Akan muncul tampilan seperti gambar di bawah ini. Apabila kriteria ingin disimpan dan ditampilkan ke dalam form, klik button OK.



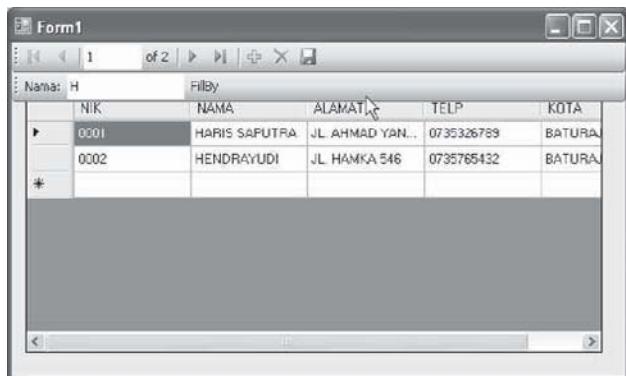
Gambar 7.24 Search Criteria Builder

Simpan form dengan mengklik ikon Save All. Atau mengklik menu File > Save All. Atau bisa juga dengan menekan tombol Ctrl+Shift+S pada keyboard lalu klik Save.



Gambar 7.25 Save Project pencarian data

Untuk menjalankan pencarian data pada form, tekan tombol F5 (run program) kemudian masukkan nama ada kolom pencarian dan klik tombol FillByCariNama. Apabila nama sesuai dengan kriteria, maka grid akan menampilkan nama-nama yang sesuai dengan kriteria tersebut.



Gambar 7.26 Form pencarian data

<http://pustaka-indo.blogspot.com>

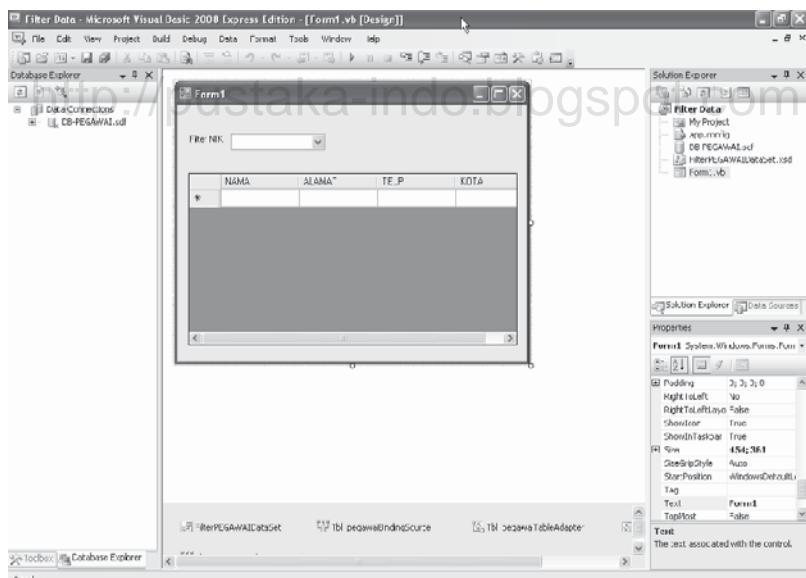
7.5 Menyaring Data dengan Filter

Menyaring data berarti mengelompokkan data sesuai dengan kriteria yang dipilih atau dimasukkan dalam suatu TextBox atau ComboBox. Berikut ini Anda akan membuat teknik menyaring data menggunakan ComboBox.

Buka Form baru dengan mengklik File > New Project lalu klik Windows Form Application, dan buat nama file misalnya “Filtering data” kemudian klik button OK. Buat koneksi dengan database dengan mengklik Add New Data Source kemudian pilih database lalu klik button Next. Klik New Connection kemudian pilih browse dan pilih database yang pernah Anda buat pada folder penyimpanan database.

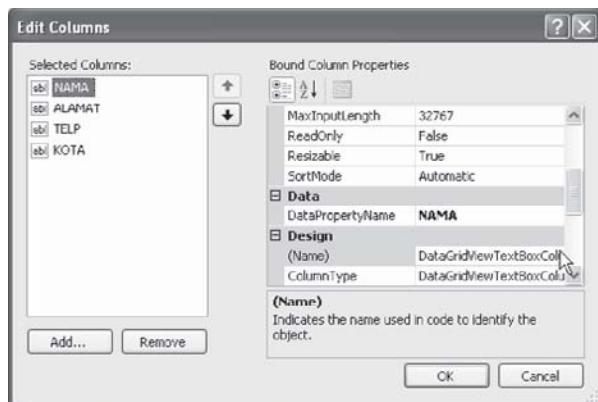
Klik Test Connection untuk memastikan database terhubung dengan baik kemudian klik button OK. Klik tombol Next dan klik Yes untuk memastikan penyalinan database ke folder penyimpanan project yang Anda buat lalu klik Next. Tandai tabel dengan tanda cek dan buat nama dataset misalnya FilterPegawaiDataset kemudian klik Finish.

Rancanglah form dengan memasukkan komponen Label dari tool dan isi properties text dengan “Filter NIK”. Masukkan komponen ComboBox dari tool dan isi properties DropDownStyle dengan DropDownList. Drag table pegawai ke ComboBox sehingga muncul FilterPegawaiDataSet, Binding Source untuk data pegawai, table adafter, dan table adafter manager. Kemudian drag tabel ke form untuk memasukkan data grid view seperti tampak pada gambar berikut:



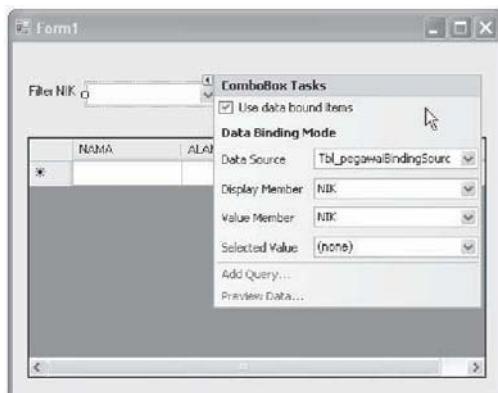
Gambar 7.27 Rancangan form filter database

Klik DatabaseView Task dan pilih Edit Cols. Kemudian klik nik dan klik button Remove untuk menghapus kolom nik pada data grid. Lalu klik button OK.



Gambar 7.28 Edit kolom Data Grid

Klik ComboBox kemudian klik ComboBox Tasks lalu pilih Display member menjadi NIK agar data yang difilter adalah nomor induk karyawan.



Gambar 7.29 ComboBox Tasks

Tekan tombol F5 untuk menjalankan aplikasi yang Anda buat, lalu klik combobox dan pilih salah satu NIK yang muncul. Setelah kita memilih data pada combobox, secara otomatis akan menyaring data yang terdapat di dalam data grid view.



Gambar 7.30 Hasil aplikasi filter database

Apabila Anda menggunakan dua tabel detail, maka tambahkan perintah berikut ini agar record dapat difilter oleh combobox.

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Me.Tbl_pegawaiBindingSource.Position =
    Me.ComboBox1.SelectedIndex
End Sub
```



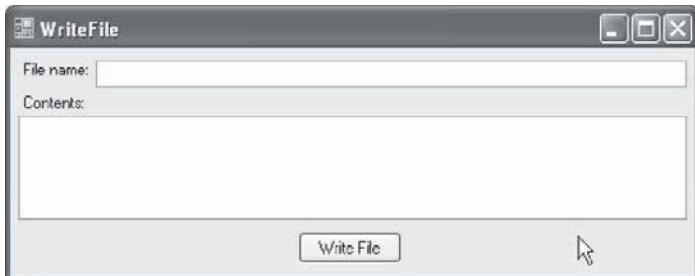

MEMBUAT APLIKASI

<http://pustaka-indo.blogspot.com>

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus). Berikut ini Anda akan membuat beberapa program aplikasi file text dan aplikasi lainnya yang bertujuan untuk memperlancar penggunaan perintah.

8.1 Membuat Aplikasi File Text

Aplikasi file text adalah sebuah program yang dapat menyimpan text atau tulisan ke dalam extention .txt. Untuk membuat aplikasi ini dibutuhkan komponen label, textbox, dan command.



Gambar 8.1 Rancangan aplikasi file text

Masukkan label dan isi properties text dengan File Name kemudian masukkan sebuah textbox di samping label File Name dan isi properties name-nya dengan txtFileName. Masukkan kembali textbox untuk menu- liskan text yang akan disimpan. Isi properties name dengan txtContents. Lalu masukkan sebuah button dan beri label Write File dan isi properties name dengan btnWriteFile. Klik ganda form kemudian ketik perintah berikut ini:

```
Private Sub Form1_Load() Handles MyBase.Load
    Dim file_name As String = Application.StartupPath
    If file_name.EndsWith("\bin\Debug") Then
        file_name.Replace("\bin\Debug", "")
        txtFileName.Text = file_name & "\test.txt"
    End Sub
```

Perintah Dim file_name as string merupakan sebuah variabel untuk menampung path atau lokasi folder. Buat folder “\bin\debug” untuk menyimpan file.txt. Klik ganda button write file dan ketik perintah berikut:

```
Private Sub btnWriteFile_Click() Handles btnWriteFile.Click
    System.IO.File.WriteAllText(txtFileName.Text,
        txtContents.Text)
    MessageBox.Show("Penyimpanan file sukses", "Pesan",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub
```

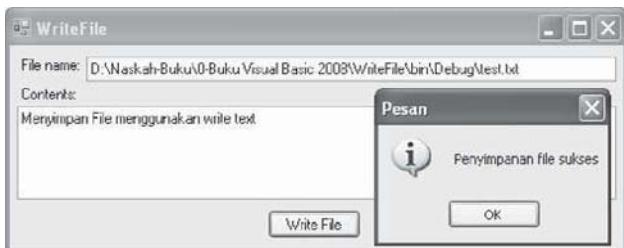
Perintah System.IO.File.WriteAllText digunakan untuk menyimpan file dengan nama “txtFileName.text” yang terdapat pada “txtContents.text”. Kemudian program akan menampilkan pesan “Penyimpanan file sukses”. Berikut ini perintah lengkap program:

```
Imports System.Security
Imports System.Security.Permissions

Public Class Form1
    Private Sub Form1_Load() Handles MyBase.Load
        Dim file_name As String = Application.StartupPath
        If file_name.EndsWith("\bin\Debug") Then
            file_name.Replace("\bin\Debug", "")
            txtFileName.Text = file_name & "\test.txt"
        End Sub

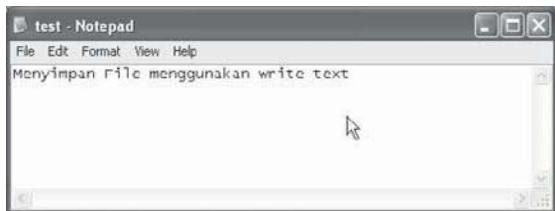
        Private Sub btnWriteFile_Click() Handles btnWriteFile.Click
            System.IO.File.WriteAllText(txtFileName.Text,
                txtContents.Text)
            MessageBox.Show("Penyimpanan file sukses", "Pesan",
                MessageBoxButtons.OK, MessageBoxIcon.Information)
        End Sub
    End Class
```

Tekan tombol F5 untuk menjalankan program dan ketik text atau tulisan yang akan disimpan, kemudian tekan button Write File.



Gambar 8.2 Hasil program Write Text

Untuk melihat file tersebut, bukalah explorer dan masuklah ke lokasi folder Write File\bin\debug dan cari file test.txt. Apabila Anda sudah menemukan file tersebut, lanjutkan dengan mengklik ganda file untuk melihat isi file text yang tersimpan.



Gambar 8.3 Isi file text yang tersimpan

8.2 Membuat Aplikasi Informasi Drive

Aplikasi informasi drive digunakan untuk menampilkan nama drive, jenis drive, root directory, volume label, drive format, free space, dan total free space dalam suatu drive. Rancanglah form berikut untuk menampilkan data-data tentang informasi drive.



Gambar 8.4 Rancangan aplikasi informasi drive

Masukkan komponen ListBox dan isi properties name dengan Is Drive, masukkan komponen label dan isi properties text dengan Name, masukkan komponen label dan isi properties text dengan is Ready, masukkan komponen label dan isi properties text dengan Drive Type, masukkan komponen label dan isi properties text dengan Root Directory, masukkan komponen label dan isi properties text dengan Volume Label,

masukkan komponen label dan isi properties text dengan Drive Format, masukkan komponen label dan isi properties text dengan Available Free Space, dan masukkan komponen label dan isi properties text dengan Total Free Space. Klik ganda form dan buat Function FormatBytes seperti terlihat pada listing program berikut ini:

```
' Perintah Format a disk size in bytes as KB, MB, GB, dll.
Private Function FormatBytes(ByVal num_bytes As Double,
    ByVal format_str As String) As String
    Dim postfixes() As String = {"Bytes", "KB", "MB",
        "GB", "TB", "PB", "EB", "ZB", "YB"}
    For i As Integer = postfixes.Length - 1 To 0 Step -1
        If num_bytes > 1024 ^ i Then
            num_bytes /= 1024 ^ i
            Return num_bytes.ToString(format_str) & " " &
                postfixes(i)
        End If
    Next i

    Return num_bytes.ToString(format_str) & " Bytes"
End Function
```

Setelah Anda membuat Function FormatBytes, klik ganda listbox Is Drive dan buatlah perintah berikut:

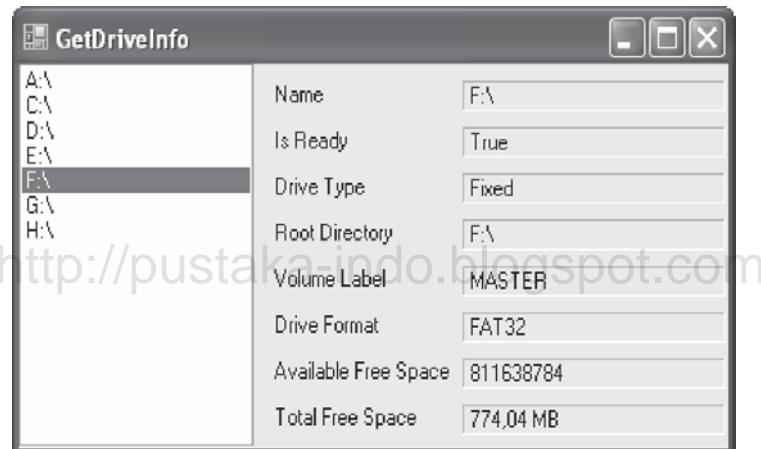
```
Private Sub lstDrives_SelectedIndexChanged() Handles
lstDrives.SelectedIndexChanged
    For Each ctl As Control In Me.Controls
        If (ctl.Name.StartsWith("lbl")) AndAlso (TypeOf
            ctl Is Label) Then
            Dim lbl As Label = DirectCast(ctl, Label)
            lbl.Text = ""
        End If
    Next ctl

    Dim drive_info As New DriveInfo(lstDrives.Text)
    lblName.Text = drive_info.Name()
    lblIsReady.Text = drive_info.IsReady().ToString()
    lblDriveType.Text = drive_info.DriveType().ToString()
    lblRootDirectory.Text =
        drive_info.RootDirectory.ToString()
    If drive_info.IsReady() Then
        lblAvailableFreeSpace.Text =
            drive_info.AvailableFreeSpace().ToString()
        lblDriveFormat.Text = drive_info.DriveFormat()
        lblTotalFreeSpace.Text =
            FormatBytes(drive_info.TotalFreeSpace, "0.00")
        lblVolumeLabel.Text = drive_info.VolumeLabel()
    End If
End Sub
```

Setelah membuat perintah untuk listbox lstDrive, klik ganda form dan buat juga perintah berikut ini:

```
Private Sub Form1_Load() Handles MyBase.Load
    For Each drive_info As DriveInfo In
        DriveInfo.GetDrives()
            lstDrives.Items.Add(drive_info.Name)
    Next drive_info
End Sub
```

Jalankan program yang Anda buat dengan menekan tombol F5, maka aplikasi akan tampak seperti gambar di bawah ini.



Gambar 8.5 Aplikasi informasi drive

Perintah lengkap program informasi drive sebagai berikut:

```
Imports System.IO

Public Class Form1
    Private Sub Form1_Load() Handles MyBase.Load
        For Each drive_info As DriveInfo In
            DriveInfo.GetDrives()
                lstDrives.Items.Add(drive_info.Name)
        Next drive_info
    End Sub

    Private Sub lstDrives_SelectedIndexChanged() Handles
        lstDrives.SelectedIndexChanged
        For Each ctl As Control In Me.Controls
```

```

If (ctl.Name.StartsWith("lbl")) AndAlso
    (TypeOf ctl Is Label) Then
    Dim lbl As Label = DirectCast(ctl, Label)
    lbl.Text = ""
End If
Next ctl

Dim drive_info As New DriveInfo(lstDrives.Text)
lblName.Text = drive_info.Name()
lblIsReady.Text = drive_info.IsReady().ToString()
lblDriveType.Text = drive_info.DriveType().ToString()
lblRootDirectory.Text =
    drive_info.RootDirectory.ToString()
If drive_info.IsReady() Then
    lblAvailableFreeSpace.Text =
        drive_info.AvailableFreeSpace().ToString()
    lblDriveFormat.Text = drive_info.DriveFormat()
    lblTotalFreeSpace.Text =
        FormatBytes(drive_info.TotalFreeSpace, "0.00")
    lblVolumeLabel.Text = drive_info.VolumeLabel()
End If
End Sub

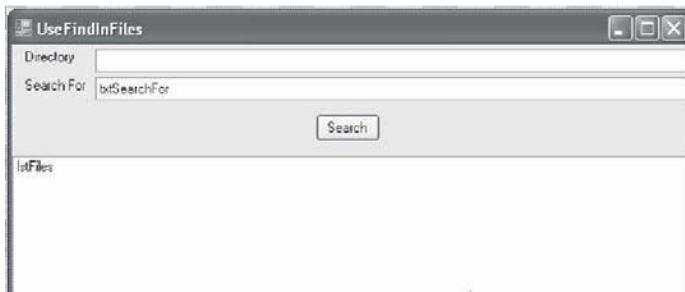
' Format a disk size in bytes as KB, MB, GB, etc.
Private Function FormatBytes(ByVal num_bytes As Double,
    ByVal format_str As String) As String
    Dim postfixes() As String = {"Bytes", "KB", "MB",
        "GB", "TB", "PB", "EB", "ZB", "YB"}
    For i As Integer = postfixes.Length - 1 To 0 Step -1
        If num_bytes > 1024 ^ i Then
            num_bytes /= 1024 ^ i
            Return num_bytes.ToString(format_str) & " " &
                postfixes(i)
        End If
    Next i

    Return num_bytes.ToString(format_str) & " Bytes"
End Function
End Class

```

8.3 Membuat Aplikasi UseFindInFiles

Aplikasi UseFindFiles adalah sebuah aplikasi yang digunakan untuk mencari teks dalam sebuah file yang terdapat dalam directory yang dipilih. Untuk membuat aplikasi tersebut rancanglah form berikut ini:



Gambar 8.6 Rancangan aplikasi UseFindInFiles

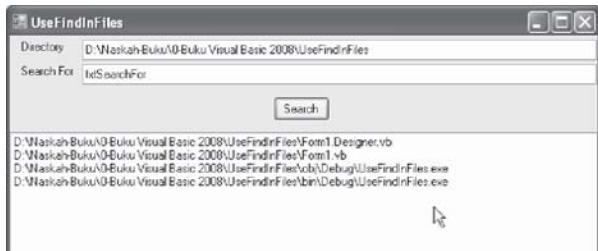
Masukkan label dan isi properties text dengan teks directory, masukkan komponen textbox dan isi properties name txtDirectory, masukkan komponen label dan isi properties text dengan Search For, masukkan komponen textbox dan isi properties name dengan txtSearchFor, masukkan komponen button dan isi properties text dengan search dan isi properties name button dengan btnSearch, masukkan komponen listbox dan isi properties name dengan lstFiles. Untuk membuat perintah, klik ganda button Search dan buat perintah berikut:

```
Private Sub btnSearch_Click() Handles btnSearch.Click
    Dim file_names As ReadOnlyCollection(Of String)
    file_names = My.Computer.FileSystem.FindInFiles( _
        txtDirectory.Text, _
        txtSearchFor.Text, _
        True, _
        FileIO.SearchOption.SearchAllSubDirectories)

    lstFiles.Items.Clear()
    For Each file_name As String In file_names
        lstFiles.Items.Add(file_name)
    Next file_name
End Sub
```

Klik ganda form dan buatlah perintah berikut:

```
Private Sub Form1_Load() Handles MyBase.Load
    Dim start_dir As String = Application.StartupPath
    start_dir =
    My.Computer.FileSystem.GetParentPath(start_dir)
    start_dir =
    My.Computer.FileSystem.GetParentPath(start_dir)
    txtDirectory.Text = start_dir
End Sub
```



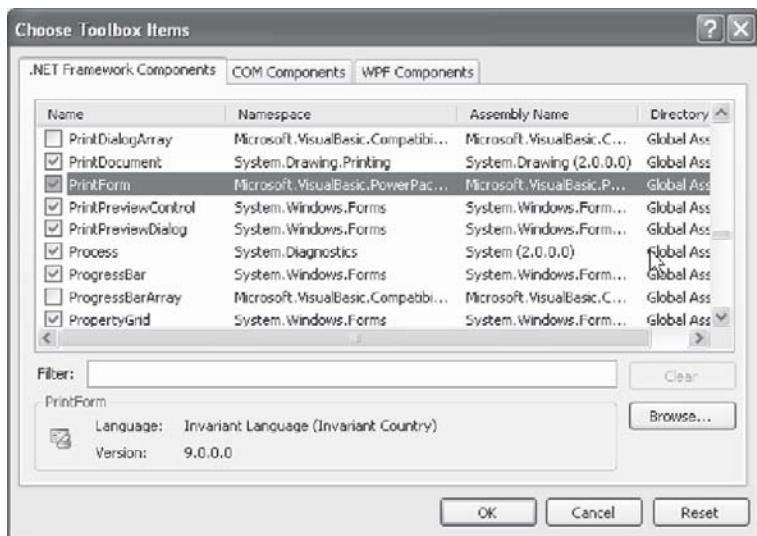
Gambar 8.7 Hasil aplikasi UseFindInFiles

Perintah lengkap aplikasi UseFindInFiles sebagai berikut:

```
Imports System.Collections.ObjectModel  
  
Public Class Form1  
    Private Sub Form1_Load() Handles MyBase.Load  
        Dim start_dir As String = Application.StartupPath  
        start_dir =  
        My.Computer.FileSystem.GetParentPath(start_dir)  
        start_dir =  
        My.Computer.FileSystem.GetParentPath(start_dir)  
        txtDirectory.Text = start_dir  
    End Sub  
  
    Private Sub btnSearch_Click() Handles btnSearch.Click  
        Dim file_names As ReadOnlyCollection(Of String)  
        file_names = My.Computer.FileSystem.FindInFiles( _  
            txtDirectory.Text, _  
            txtSearchFor.Text, _  
            True, _  
            FileIO.SearchOption.SearchAllSubDirectories)  
  
        lstFiles.Items.Clear()  
        For Each file_name As String In file_names  
            lstFiles.Items.Add(file_name)  
        Next file_name  
    End Sub  
End Class
```

8.4 Print Form

Printform adalah komponen untuk mencetak form ke file, preview, atau printer. Komponen tersebut terdapat pada ToolBox Printing, apabila Anda tidak melihat komponen tersebut, maka klik kanan mouse dan pilih Choose Item. Lalu pilih Print Form dan klik button OK pada Choose ToolBox Items.



Gambar 8.8 Choose ToolBox Items

Setelah Anda memilih PrintForm pada kotak Choose ToolBox Items, maka PrintForm akan tampil dalam kelompok toolbox Printing.



Gambar 8.9 PrintForm dalam komponen Printing

Buat koneksi database dengan mengklik Add New Data Source kemudian pilih database lalu klik button Next. Klik **New Connection** kemudian pilih browse dan pilih database yang pernah Anda buat pada folder penyimpanan database.

Klik **test Conection** untuk memastikan database terhubung dengan baik kemudian klik button OK. Klik tombol Next dan klik Yes untuk memastikan penyalinan database ke folder penyimpanan project yang Anda buat, lalu klik Next. Setelah tampil dataset ubahlah table menjadi detail lalu drag-lah ke form kemudian ubah kembali table menjadi gridview, lalu drag kembali ke tabel.

Klik **DataBindingTask**, ubah Dock menjadi bottom sehingga data binding berubah posisi dari bagian atas ke posisi bawah form. Masukkan komponen printForm ke form dengan cara mengklik toolbox printing kemudian klik komponen printform. Lalu drag-lah komponen printform ke dalam form. Kemudian klik ganda Data Binding Navigator dan pilih button. Setelah toolStrip masuk ke dalam data binding, Anda klik ganda data binding tersebut dan ketiklah kode program untuk mencetak form sebagai berikut:

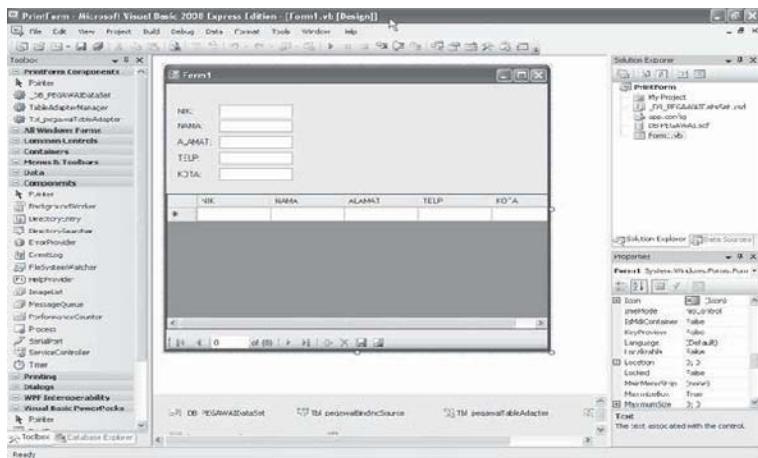
```
Private Sub ToolStripButton1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
ToolStripButton1.Click  
    Me.PrintForm1.PrintAction =  
        Printing.PrintAction.PrintToPreview  
    Me.PrintForm1.Print()  
End Sub
```

Perintah PrintAction digunakan untuk menentukan bentuk cetakan ke layar (preview), file, atau ke dalam bentuk cetakan printer. Tekan F5 untuk menjalankan aplikasi kemudian klik data binding print form yang terdapat pada data binding navigator sehingga tampil preview form sebagai berikut:



Gambar 8.10 Print Preview

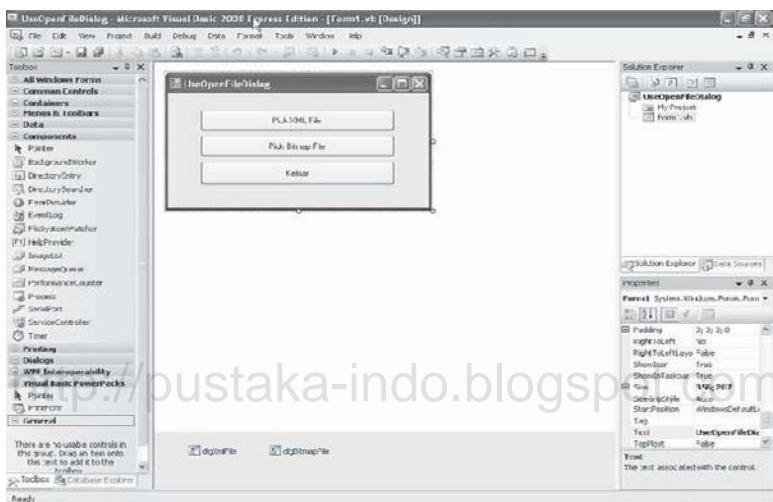
Dalam print preview terdapat ikon Printer untuk mencetak tampilan preview ke printer, ikon Zoom yang digunakan untuk mengatur skala tampilan, ikon One Page untuk menampilkan preview tiap satu halaman, ikon Two Page untuk menampilkan preview dalam bentuk dua halaman, dan button Close untuk keluar dari Print Preview.



Gambar 8.11 Rancangan form PrintForm

8.5 Aplikasi UseOpenFileDialog

Aplikasi UseOpenFileDialog adalah program yang menampilkan dialog untuk membuka file. Dalam aplikasi ini Anda akan membuat dialog untuk membuka file XML dan file BMP. Untuk membuat aplikasi tersebut rancanglah form berikut ini.



Gambar 8.12 Rancangan aplikasi Use OpenFileDialog

Klik ToolBox Dialog lalu drag OpenFileDialog ke Form, kemudian isi properties name dlgXmlFile. Pilih kembali OpenFileDialog kemudian isi properties name menjadi dlgBitmap file. **dlgXmlFile** adalah kotak dialog untuk membuka file XML, sedangkan **dlgBitmapFile** adalah kotak dialog untuk membuka file bitmap. Drag tiga buah button; button pertama diisi dengan text Pick XML File dan isi properties name **btnPickXmlFile**, button kedua diisi dengan text Pick Bitmap File dan isi properties name **btnPickBitmapFile**, dan button yang terakhir diisi dengan teks keluar. Untuk membuat printah, Anda klik ganda komponen OpenFileDialog **dlgBitmapFile** dan ketikkan:

```

Private Sub dlgBitmapFile_FileOk(ByVal sender As
System.Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles
dlgBitmapFile.FileOk
    For Each file_name As String In dlgBitmapFile.FileNames
        ' See if this file name ends with .bmp.
        If Not file_name.EndsWith(".bmp") Then
            MessageBox.Show("File '" & file_name & _
                "' is not a bitmap file", _
                "Invalid File Type", _
                MessageBoxButtons.OK, _
                MessageBoxIcon.Exclamation)
            e.Cancel = True
            Exit Sub
        End If
    Next file_name
End Sub

```

Kemudian klik ganda button Pick XML File dan buatlah perintah berikut ini:

```

Private Sub btnPickXmlFile_Click() Handles
btnPickXmlFile.Click
    If dlgXmlFile.ShowDialog() =
        Windows.Forms.DialogResult.OK Then
        MessageBox.Show(dlgXmlFile.FileName)
    End If
End Sub

```

Kemudian untuk membuat perintah pada button Pickt Bitmap File, Anda klik button tersebut kemudian buat perintah berikut:

```

Private Sub btnPickBitmapFile_Click() Handles
btnPickBitmapFile.Click
    If dlgBitmapFile.ShowDialog() =
        Windows.Forms.DialogResult.OK Then
        Dim file_names As String =
            Strings.Join(dlgBitmapFile.FileNames, " ")
        MessageBox.Show(file_names)
    End If
End Sub

```

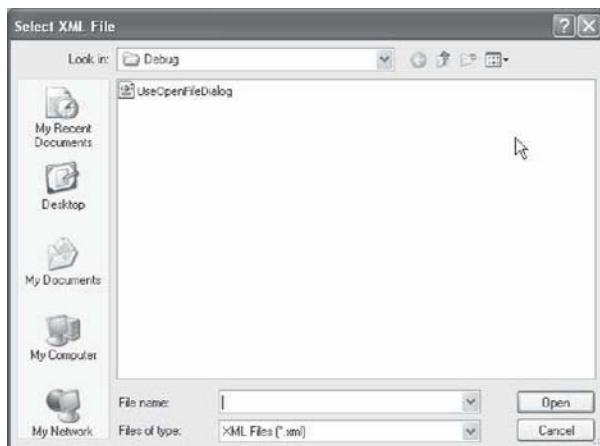
Kemudian buat perintah pada button keluar dengan cara mengklik ganda button.

```

Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    End
End Sub

```

Tekan F5 untuk menjalankan aplikasi dan klik button Pick XML File sehingga tampil kotak dialog sebagai berikut:



Gambar 8.13 Kotak dialog XML File

Untuk membuka file XML pilih file lalu klik button open. Begitu juga apabila Anda ingin membuka file Bitmap.

Berikut ini listing lengkap program UseFileDialog:

```
Public Class Form1
    Private Sub btnPickXmlFile_Click() Handles
        btnPickXmlFile.Click
        If dlgXmlFile.ShowDialog() =
            Windows.Forms.DialogResult.OK Then
            MessageBox.Show(dlgXmlFile.FileName)
        End If
    End Sub

    ' Ensure that the user only selects bitmap files.
    Private Sub dlgBitmapFile_FileOk(ByVal sender As
        System.Object, ByVal e As
        System.ComponentModel.CancelEventArgs) Handles
        dlgBitmapFile.FileOk
        For Each file_name As String In
            dlgBitmapFile.FileNames
            ' See if this file name ends with .bmp.
            If Not file_name.EndsWith(".bmp") Then
                MessageBox.Show("File '" & file_name & _
                    "' is not a bitmap file", _
                    "Invalid File Type", _
```

```

        MessageBoxButtons.OK, _
        MessageBoxIcon.Exclamation)
e.Cancel = True
Exit Sub
End If
Next file_name
End Sub

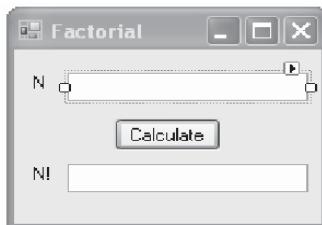
Private Sub btnPickBitmapFile_Click() Handles
btnPickBitmapFile.Click
If dlgBitmapFile.ShowDialog() =
Windows.Forms.DialogResult.OK Then
Dim file_names As String =
Strings.Join(dlgBitmapFile.FileNames, " ")
MessageBox.Show(file_names)
End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button1.Click
End
End Sub
End Class

```

8.6 Aplikasi Factorial

Aplikasi Factorial adalah program yang menghitung bilang factorial dengan menggunakan algoritma perhitungan berulang. Factorial dari 2 adalah 2, didapat dari nilai awal factorial dikalikan dengan 2 (nilai awal factorial adalah 1). Factorial dari 3 adalah $1 * 2 = 2$, $2 * 3 = 6$, Jadi factorial 3 adalah 6. Factorial dari 4 adalah $1 * 2 = 2$, $2 * 3 = 6$, $6 * 4 = 24$. Jadi Factorial 4 adalah 24. Untuk membuat program factorial rancanglah form sebagai berikut:



Gambar 8.14 Rancangan form aplikasi Factorial

Untuk membuat rancangan form tersebut, klik menu File > New Project dan ketik nama file (misalnya Factorial), kemudian Anda klik button OK. Masukkan komponen Label dan isilah properties Text dengan N. Kemudian masukkan textbox dan isi properties name dengan txtN. Masukkan lagi label dan isi properties text dengan N! (N Factorial). Masukkan kembali textbox dan isi properties name dengan txtResult. Masukkan Command Button dan isi properties text dengan Calculate yang digunakan untuk menghitung factorial. Untuk membuat perintah factorial klik ganda button calculate dan buat function factorial.

```
Private Function Factorial(ByVal N As Double) As Double
    Dim result As Double = 1
    For i As Integer = 2 To N
        result *= i
        MsgBox(i)
    Next i
    Return result
End Function
```

Dim Result as Double =1 adalah perintah untuk menampung data hasil factorial yang diberi nilai awal 1. Perintah for digunakan untuk mengulangi perhitungan factorial yang akan ditampung menggunakan variabel i. Nilai perulangan akan dimulai dari angka 2 sampai dengan angka factorial yang diinputkan ke dalam program. Kemudian hitung nilai factorial dengan perintah result = result * i sampai dengan nilai factorial yang diinginkan.

Setelah kita membuat function factorial, ketiklah perintah untuk memanggil function tersebut pada command button sebagai berikut:

```
Private Sub btnCalculate_Click() Handles btnCalculate.Click
    txtResult.Text = Factorial(Double.Parse(txtN.Text))
End Sub
```

txtResult akan menampung hasil dari function Factorial yang diambil dari TextBox txtN. Perintah Double.Parse digunakan untuk mengubah data string atau text menjadi type data angka double.



Gambar 8.15 Hasil Factorial

Untuk menguji program yang sudah kita buat, tekan tombol F5 dan masukkan nilai factorial pada textbox N lalu klik button Calculate. Hasil perhitungan factorial akan ditampilkan pada textbox N!. Perintah program lengkap aplikasi factorial adalah sebagai berikut:

```
Public Class Form1
    Private Sub btnCalculate_Click() Handles btnCalculate.Click
        txtResult.Text = Factorial(Double.Parse(txtN.Text))
    End Sub

    Private Function Factorial(ByVal N As Double) As Double
        Dim result As Double = 1
        For i As Integer = 2 To N
            result *= i
            MsgBox(i)
        Next i
        Return result
    End Function
End Class
```

8.7 Aplikasi Debug Windows

Aplikasi ini sebenarnya dibuat untuk mempelajari letak kutu (debug) atau kesalahan yang terjadi pada baris perintah program. Untuk melanjutkan jalannya program yang terjadi debug, dapat dilakukan dengan menekan tombol F5 yang berarti Continue. Untuk berhenti dari debug dapat menekan tombol Ctrl+Alt+Break atau klik menu Debug pilih *stop debugging*. Debug yang terjadi pada baris program biasanya ditandai dengan baris perintah yang diberi warna kuning. Warna tersebut menunjukkan bahwa di baris tersebut terjadi kesalahan debuging.

```

Public Class Form1
    ' Quick watch example.
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        txtDirectory.Text = Application.StartupPath
        Dim txt As String = txtDirectory.Text
    End Sub

    ' Locals window example.
    Private Sub btnLocals_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLocals.Click
        SearchDatabase()
    End Sub

    Private Sub SearchDatabase()
        FindEmployee()
    End Sub

    Private Sub FindEmployee()
        Dim values() As Integer = {1237, 4378, 901987, 3287, 17843, 328797}
        Dim i As Integer = 0
        Dim j As Integer = 207
        Dim base_rate(11) As Single
        Try
            j = i \ 1
            Stop
        Catch ex As Exception
            Stop
            Dim x As Double = 365.2135
            Dim y As Double = -1.3278E+16
        End Try
    End Sub
End Class

```

Gambar 8.16 Gambar debugging

Untuk mencoba aplikasi tersebut, buatlah project baru dengan nama debug window kemudian rancanglah form berikut ini.



Gambar 8.17 Desain aplikasi Debug Windows

Masukkan komponen label dan isi properties text dengan Directory, lalu masukkan komponen TextBox dan isi name dengan txtDirectory, serta masukkan satu komponen Button dan isi properties text dengan local dan name btnlocals. Klik ganda button Locals dan ketik perintah berikut:

```

Private Sub SearchDatabase()
    FindEmployee()
End Sub

Private Sub FindEmployee()
    Dim values() As Integer = {8237, 4378, 901987, 3287,
        17843, 328797}
    Dim i As Integer = 0
    Dim j As Integer = 287
    Dim base_rate(11) As Single
    Try
        j = 1 \ i
        Stop
    Catch ex As Exception
        Stop
        Dim x As Double = 365.2136
        Dim y As Double = -1.3278E+16
    End Try
End Sub
End Class

```

Kemudian klik ganda form sehingga tampil private sub dari form load sebagai berikut:

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
End Sub

```

Kemudian ketik perintah untuk mengisi textbox dengan lokasi folder aktif sekarang dengan perintah berikut ini:

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    txtDirectory.Text = Application.StartupPath
    Dim txt As String = txtDirectory.Text
End Sub

```

Setelah membuat perintah untuk form load, klik button Local dan ketik perintah berikut ini.

```

Private Sub btnLocals_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnLocals.Click
    SearchDatabase()
End Sub

```

Setelah selesai membuat perintah, Anda tekan tombol F5, maka akan terdapat kesalahan (debug) dengan warna kuning pada perintah stop.

Mengapa terjadi debug? Apakah karena sudah terdapat dua perintah stop pada baris di atasnya? Untuk menghilangkan kesalahan tersebut, beri tanda note (‘) di depan perintah stop, seperti terlihat di bawah ini:

```
Private Sub FindEmployee()
    Dim values() As Integer = {8237, 4378, 901987, 3287,
        17843, 328797}
    Dim i As Integer = 0
    Dim j As Integer = 287
    Dim base_rate(11) As Single
    Try
        j = 1 \ i
        Stop
    Catch ex As Exception
        'Stop
        Dim x As Double = 365.2136
        Dim y As Double = -1.3278E+16
    End Try
End Sub
```

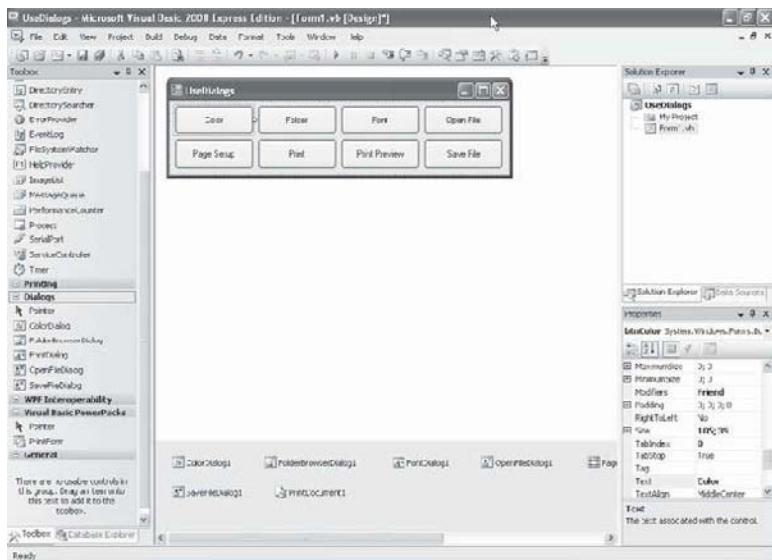
Setelah program diberi tanda note (‘) berarti perintah tersebut tidak akan diproses. Tekan tombol Ctrl+Alt+Break untuk menghentikan debug kemudian jalankan kembali program aplikasi dengan menekan tombol F5.

<http://pustaka-indo.blogspot.com>

8.8 Aplikasi Use Dialogs

Aplikasi use dialog adalah program yang dibuat dengan komponen-komponen yang terdapat pada toolbox dialog dan printing, seperti ColorDialog, FolderBrowserDialog, FontDialog, OpenFileDialog, PageSetupDialog, PrintPrepiewDialog, SaveFileDialog, dan PrintFileDialog.

ColorDialog adalah kotak dialog yang menampilkan warna yang dapat dipilih oleh pengguna. FontDialog adalah dialog untuk memilih jenis font, style, dan size font. OpenFileDialog adalah dialog yang digunakan untuk memilih file yang akan dibuka oleh user. PageSetupDialog adalah kotak dialog yang digunakan untuk mengatur kertas. SaveFileDialog adalah kotak dialog yang digunakan untuk menyimpan file, dan PrintFileDialog adalah kotak dialog yang digunakan untuk mencetak file.

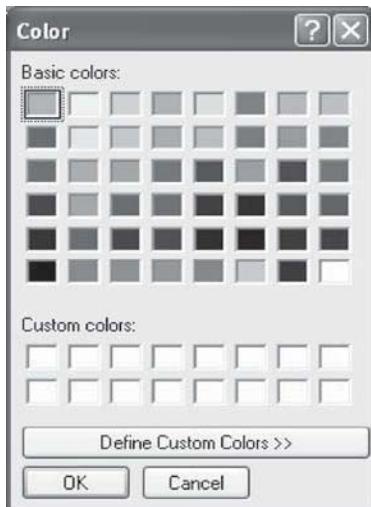


Gambar 8.18 Desain aplikasi UseDialogs

Masukkan command button masing-masing untuk button Color, Folder, Open File, Page Setup, Print, Print Preview, dan Save File. Masukkan komponen Color Dialog dengan cara mengklik ganda komponen Color Dialog, kemudian klik ganda button Color lalu ketik perintah program berikut:

```
Private Sub btnColor_Click() Handles btnColor.Click
    ColorDialog1.Color = Me.BackColor
    If ColorDialog1.ShowDialog() =
        Windows.Forms.DialogResult.OK Then
            Me.BackColor = ColorDialog1.Color
    End If
End Sub
```

Tekan tombol F5 untuk menjalankan program sehingga akan tampil kotak dialog warna. Anda pilih salah satu warna kemudian klik tombol OK, sehingga warna yang Anda pilih akan menggantikan warna form yang aktif.



Gambar 8.19 Kotak ColorDialog

Klik ganda komponen FolderBrowserDialog kemudian klik ganda button folder lalu ketik perintah berikut:

```
Private m_Folder As String = ""  
Private Sub btnFolder_Click(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles btnFolder.Click  
    If m_Folder.Length > 0 Then  
        FolderBrowserDialog1.SelectedPath = m_Folder  
        If FolderBrowserDialog1.ShowDialog =  
            Windows.Forms.DialogResult.OK Then  
                m_Folder = FolderBrowserDialog1.SelectedPath  
                MessageBox.Show("Folder: " & m_Folder, "Ok",  
                    MessageBoxButtons.OK, MessageBoxIcon.Information)  
            End If  
    End Sub
```

Tekan tombol F5 kemudian klik button folder sehingga akan muncul kotak dialog folder.

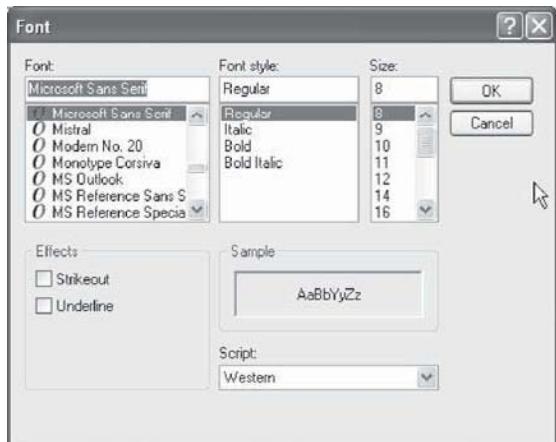


Gambar 8.20 Kotak dialog Folder

Klik ganda komponen FontDialog lalu klik ganda button font dan buat perintah untuk menampilkan kotak dialog font.

```
Private Sub btnFont_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnFont.Click
    FontDialog1.Font = Me.Font
    If FontDialog1.ShowDialog =
        Windows.Forms.DialogResult.OK Then
            Me.Font = FontDialog1.Font
    End If
End Sub
```

Tekan tombol F5 kemudian klik button Font sehingga akan tampil kotak dialog font. Apabila Font dipilih dan Anda menekan tombol OK, maka Font yang dipilih tersebut akan mengubah semua ukuran font yang terdapat pada form aplikasi Use Dialogs.

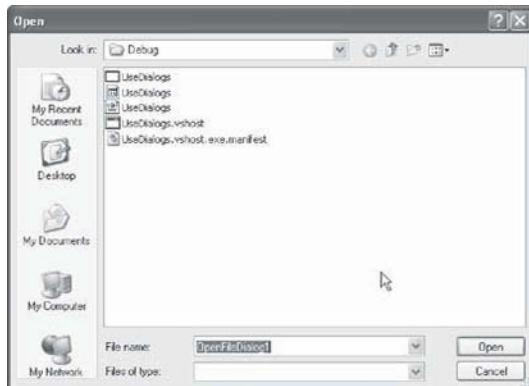


Gambar 8.21 Kotak dialog Font

Klik ganda komponen OpenFileDialog kemudian klik ganda button Open File lalu ketiklah perintah program berikut ini.

```
Private m_OpenFile As String
Private Sub btnOpenfile_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnOpenfile.Click
    If m_OpenFile.Length > 0 Then
        OpenFileDialog1.FileName = m_OpenFile
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        m_OpenFile = OpenFileDialog1.FileName
        MessageBox.Show("Open File: " & m_OpenFile, "Ok",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End Sub
```

Setelah program selesai diketik, tekan tombol F5 untuk menjalankan program dan klik button Open File. Apabila Anda ingin membuka file lain pada saat kotak dialog Open File, maka Anda dapat memilih file tersebut melalui Look In. Pilih lokasi folder lalu pilih file yang akan dibuka.

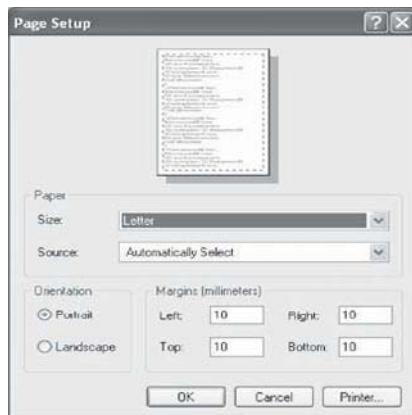


Gambar 8.22 Kotak dialog Open File

Klik ToolBox Printing lalu klik ganda komponen pagesetupdialog. Untuk membuat perintah menampilkan page setup dialog, klik ganda button Page Setup kemudian ketik perintah.

```
Private Sub btnPageSetup_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnPageSetup.Click
    PageSetupDialog1.ShowDialog()
End Sub
```

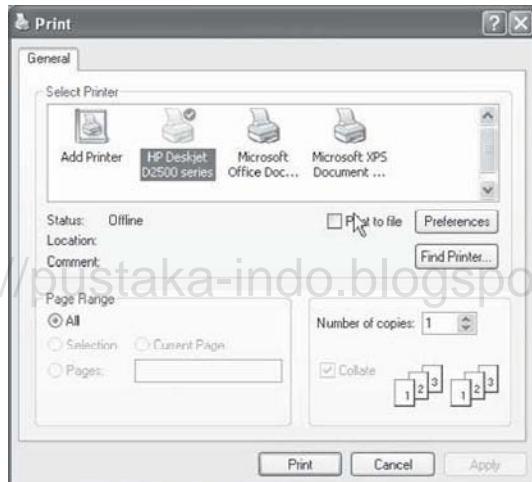
Tekan tombol F5, akan tampil kotak dialog page setup seperti berikut:



Gambar 8.23 Page setup dialog

Klik ToolBox Printing kemudian klik ganda komponen PrintDialog lalu klik ganda button Print dan ketik perintah untuk button Print Dialog sebagai berikut:

```
Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Using the_pen As New Pen(Color.Red, 10)
        e.Graphics.DrawEllipse(the_pen, e.MarginBounds)
    End Using
End Sub
```



Gambar 8.24 Print Dialog

Untuk membuat perintah pada button Print Preview, klik ganda lalu ketik perintah:

```
Private Sub btnPrintPreview_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrintPreview.Click
    PrintPreviewDialog1.ShowDialog()
End Sub
```

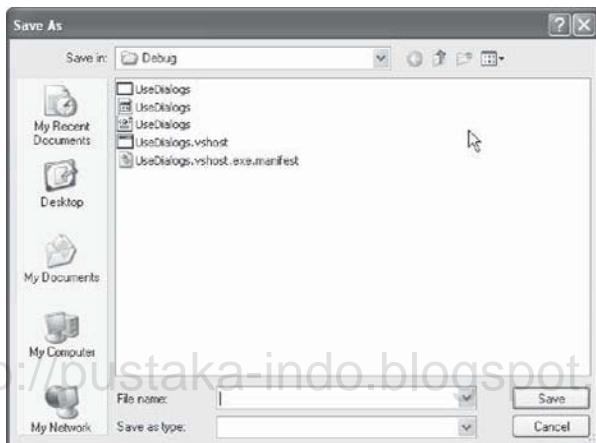
Klik ganda button Save File dan ketiklah perintah berikut:

```
Private m_SaveFile As String = ""
```

```

Private Sub btnSaveFile_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSaveFile.Click
    If m_SaveFile.Length > 0 Then
        SaveFileDialog1.FileName = m_SaveFile
        If SaveFileDialog1.ShowDialog() =
            Windows.Forms.DialogResult.OK Then
            m_SaveFile = SaveFileDialog1.FileName
            MessageBox.Show("Open File: " & m_SaveFile, "Ok",
                MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End Sub
End Class

```



Gambar 8.25 Kotak dialog Save As

Berikut ini perintah lengkap dari aplikasi UseDialogs:

```

Public Class Form1
    Private Sub btnColor_Click() Handles btnColor.Click
        ColorDialog1.Color = Me.BackColor
        If ColorDialog1.ShowDialog() =
            Windows.Forms.DialogResult.OK Then
            Me.BackColor = ColorDialog1.Color
        End If
    End Sub

    Private m_Folder As String = ""
    Private Sub btnFolder_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnFolder.Click
        If m_Folder.Length > 0 Then
            FolderBrowserDialog1.SelectedPath = m_Folder
            If FolderBrowserDialog1.ShowDialog() =
                Windows.Forms.DialogResult.OK Then
                m_Folder = FolderBrowserDialog1.SelectedPath
                MessageBox.Show("Folder: " & m_Folder, "Ok",

```

```

        MessageBoxButtons.OK, MessageBoxIcon.Information)
End If
End Sub

Private Sub btnFont_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnFont.Click
    FontDialog1.Font = Me.Font
    If FontDialog1.ShowDialog =
        Windows.Forms.DialogResult.OK Then
            Me.Font = FontDialog1.Font
    End If
End Sub

Private m_OpenFile As String = ""
Private Sub btnOpenFile_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnOpenFile.Click
    If m_OpenFile.Length > 0 Then
        OpenFileDialog1.FileName = m_OpenFile
    If OpenFileDialog1.ShowDialog =
        Windows.Forms.DialogResult.OK Then
            m_OpenFile = OpenFileDialog1.FileName
            MessageBox.Show("Open File: " & m_OpenFile, "Ok",
                MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End Sub

Private Sub btnPageSetup_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnPageSetup.Click
    PageSetupDialog1.ShowDialog()
End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As
 System.Object, ByVal e As
 System.Drawing.Printing.PrintPageEventArgs) Handles
 PrintDocument1.PrintPage
    Using the_pen As New Pen(Color.Red, 10)
        e.Graphics.DrawEllipse(the_pen, e.MarginBounds)
    End Using
End Sub

Private Sub btnPrint_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnPrint.Click
    PrintDialog1.ShowDialog()
End Sub

Private Sub btnPrintPreview_Click(ByVal sender As
 System.Object, ByVal e As System.EventArgs) Handles
 btnPrintPreview.Click
    PrintPreviewDialog1.ShowDialog()
End Sub

Private m_SaveFile As String = ""
Private Sub btnSaveFile_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles btnSaveFile.Click
    If m_SaveFile.Length > 0 Then SaveFileDialog1.FileName =
m_SaveFile
    If SaveFileDialog1.ShowDialog =
        Windows.Forms.DialogResult.OK Then

```

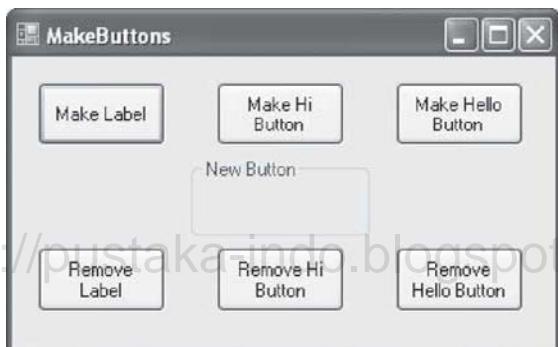
```

    m_SaveFile = SaveFileDialog1.FileName
    MessageBox.Show("Open File: " & m_SaveFile, "Ok",
    MessageBoxButtons.OK, MessageBoxIcon.Information)
End If
End Sub
End Class

```

8.9 Aplikasi Make Button

Aplikasi Make Button adalah program yang berisi perintah cara menambahkan button, dan perintah cara menghapus button dari form. Untuk membuat aplikasi, rancanglah form berikut ini.



Gambar 8.26 Desain aplikasi Make Button

Masukkan enam button, masing-masing dengan text make label, make Hi button, make hello button, Remove label, Remove Hi Button, Remove Hello Button, dan sebuah groupBox. Klik button Make Label dan isi properties name dengan btnNewLabel. Klik ganda button tersebut dan buat perintah berikut:

```

' Membuat sebuah label.
Private Sub btnNewLabel_Click() Handles btnNewLabel.Click
    Dim lbl As New Label
    lbl.SetBounds(16, 64, 80, 23)
    lbl.Text = "Hello World!"
    lbl.Name = "lblHelloWorld"
    Me.Controls.Add(lbl)
End Sub

```

Dim lbl as new label adalah variabel untuk menampung komponen label. Kemudian label tersebut dibuat text “Hello World!” dan diberi nama lblHelloWorld. Setelah perintah untuk button make label selesai dibuat, Klik button Make Hi Button dan isi properties name dengan btnMakeHi-Button. Klik ganda button tersebut lalu ketik perintah program berikut:

```
' Deklarasi btnHi button dengan WithEvents.  
Private WithEvents btnHi As Button  
  
' membuat btnHi button yang baru  
Private Sub btnMakeHiButton_Click() Handles  
    btnMakeHiButton.Click  
    btnHi = New Button  
    btnHi.SetBounds(16, 16, 80, 23)  
    btnHi.Text = "Say Hi"  
    grpNewButton.Controls.Add(btnHi)  
End Sub
```

Klik ganda button make hello button kemudian ketik perintah berikut:

```
' Membuat Hello button  
Private Sub btnMakeHelloButton_Click() Handles  
    btnMakeHelloButton.Click  
    Dim btnHello As New Button  
    btnHello.SetBounds(240, 64, 80, 23)  
    btnHello.Text = "Say Hello"  
    btnHello.Name = "btnHello"  
    Me.Controls.Add(btnHello)  
  
    AddHandler btnHello.Click, AddressOf Hello_Click  
End Sub
```

Setelah selesai membuat perintah untuk button Make Hello Button, klik ganda button Remove Label dan ketik perintah berikut:

```
Private Sub btnRemoveLabel_Click() Handles  
    btnRemoveLabel.Click  
    Me.Controls.Remove(Me.Controls("lblHelloWorld"))  
End Sub
```

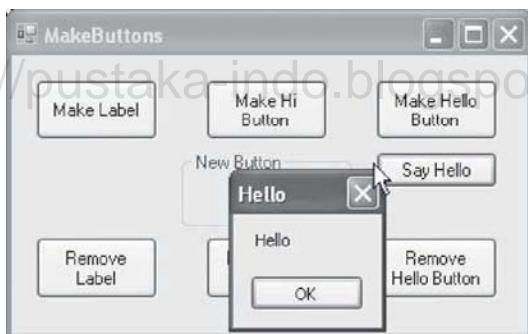
Button Remove Label digunakan untuk menghapus label yang sudah dibuat menggunakan perintah remove. Klik ganda button Remove Hi Button dan ketik perintah berikut:

```
Private Sub btnbRemoveHiButton_Click() Handles  
btnbRemoveHiButton.Click  
    grpNewButton.Controls.Remove(btnHi)  
    btnHi = Nothing  
End Sub
```

Seletah Anda membuat perintah untuk menghapus atau membuat button Hi, klik ganda button Remove Hello Button kemudian ketik perintah programnya:

```
Private Sub btnRemoveHelloButton_Click() Handles  
btnRemoveHelloButton.Click  
    Me.Controls.Remove(Me.Controls("btnHello"))  
End Sub
```

Kemudian setelah perintah sudah selesai dibuat, tekan tombol F5 untuk menjalankan program sehingga akan tampil hasil dari aplikasi yang kita buat.



Gambar 8.27 Hasil dari aplikasi Make Button

Berikut ini perintah lengkap dari aplikasi make button:

```
Public Class Form1  
    ' Membuat sebuah label.  
    Private Sub btnMakeLabel_Click() Handles  
        btnMakeLabel.Click  
        Dim lbl As New Label  
        lbl.SetBounds(16, 64, 80, 23)  
        lbl.Text = "Hello World!"  
        lbl.Name = "lblHelloWorld"  
        Me.Controls.Add(lbl)  
    End Sub
```

```

' DeKlarasi btnHi button dengan WithEvents.
Private WithEvents btnHi As Button

' membuat btnHi button yang baru
Private Sub btnMakeHiButton_Click() Handles
btnMakeHiButton.Click
    btnHi = New Button
    btnHi.SetBounds(16, 16, 80, 23)
    btnHi.Text = "Say Hi"
    grpNewButton.Controls.Add(btnHi)
End Sub

Private Sub btnHi_Click() Handles btnHi.Click
    MessageBox.Show("Hi", "Hi", MessageBoxButtons.OK)
End Sub

' Membuat Hello button
Private Sub btnMakeHelloButton_Click() Handles
btnMakeHelloButton.Click
    Dim btnHello As New Button
    btnHello.SetBounds(240, 64, 80, 23)
    btnHello.Text = "Say Hello"
    btnHello.Name = "btnHello"
    Me.Controls.Add(btnHello)

    AddHandler btnHello.Click, AddressOf Hello_Click
End Sub

' Say Hello.
Private Sub Hello_Click()
    MessageBox.Show("Hello", "Hello", MessageBoxButtons.OK)
End Sub

Private Sub btnbRemoveHiButton_Click() Handles
btnbRemoveHiButton.Click
    grpNewButton.Controls.Remove(btnHi)
    btnHi = Nothing
End Sub

Private Sub btnRemoveHelloButton_Click() Handles
btnRemoveHelloButton.Click
    Me.Controls.Remove(Me.Controls("btnHello"))
End Sub

Private Sub btnRemoveLabel_Click() Handles
btnRemoveLabel.Click
    Me.Controls.Remove(Me.Controls("lblHelloWorld"))
End Sub
End Class

```

http://ristaka-indo.blogspot.com

8.10 Aplikasi Transparent Form

Aplikasi transparent form adalah program untuk membuat form seolah-olah tembus pandang (transparent). Istilah transparent sebenarnya adalah memperkecil tingkat pewarnaan terhadap objek, dalam hal ini form. Istilah yang digunakan untuk menambah dan mengurangi tingkat pewarnaan tersebut adalah opacity. Untuk membuat aplikasi transparent form rancanglah form berikut ini.



Gambar 8.28 Desain form Transparent

Masukkan tiga komponen button ke dalam form, kemudian masukkan label dan isi text dengan First Name. Masukkan textbox untuk first name lalu masukkan kembali label dan isi text dengan last name. Masukkan textbox lastname kemudian masukkan label street dan masukkan textbox street. Masukan sebuah button Close kemudian klik ganda button close tersebut lalu ketik perintah berikut:

```
Private Sub btnClose_Click() Handles btnClose.Click  
    Me.Close()  
End Sub
```

Klik ganda button1 kemudian ketik perintah untuk menampilkan pesan apabila Anda mengklik button1:

```
Private Sub Button1_Click() Handles Button1.Click
    MessageBox.Show("Anda mengklik Button1")
End Sub
```

Ketik juga perintah untuk button2 dan button3 dengan perintah untuk menampilkan pesan apabila button2 dan button3 diklik. Perintah lengkap untuk aplikasi transparan dapat Anda lihat di bawah ini.

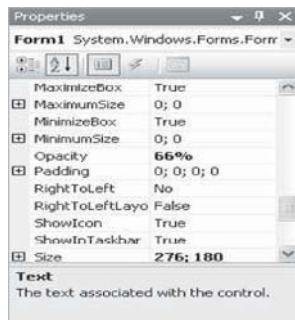
```
Public Class Form1
    Private Sub Button1_Click() Handles Button1.Click
        MessageBox.Show("Anda mengklik Button1")
    End Sub

    Private Sub Button2_Click() Handles Button2.Click
        MessageBox.Show("Anda mengklik Button2")
    End Sub

    Private Sub Button3_Click() Handles Button3.Click
        MessageBox.Show("Anda mengklik Button3")
    End Sub

    Private Sub btnClose_Click() Handles btnClose.Click
        Me.Close()
    End Sub
```

Tekan tombol F5 untuk menjalankan program, perhatikan tampilan form apakah sudah terlihat transparan? Apabila belum, Anda klik Form dan isi properties Opacity dengan angka 66% dan tekan kembali tombol F5.

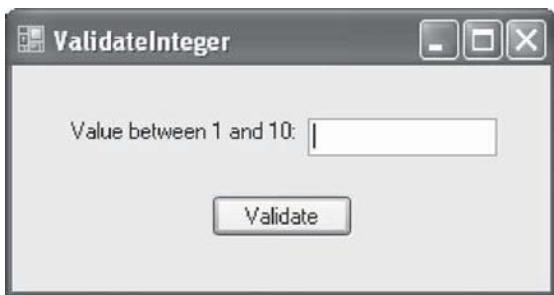


Gambar 8.29 Properties opacity

Setelah kita mengatur properties opacity form, maka form akan terlihat transparan (tembus pandang).

8.11 Aplikasi Validasi Integer

Validasi integer adalah program yang digunakan untuk mengontrol angka integer yang dimasukkan ke dalam sebuah textbox. Dalam aplikasi ini Anda akan membuat range angka mulai dari 1 sampai dengan angka 10 yang dapat dimasukkan ke dalam textbox. Rancanglah form aplikasi validasi integer berikut ini.



Gambar 8.30 Desain aplikasi validasi integer

Masukkan label dan isi properties text dengan *value between 1 and 10* (masukkan angka 1 sampai dengan 10) kemudian masukkan textbox dan isi properties name dengan txtValue, serta masukkan sebuah komponen button Validate dan isi properties name btnValidate. Klik ganda button Validate dan ketik function berikut ini:

```
Private Function IsValidInteger(ByRef result As Integer,  
    ByVal txt As TextBox, ByVal field_name As String, Optional  
    ByVal min_value As Integer = Integer.MinValue, Optional ByVal  
    max_value As Integer = Integer.MaxValue) As Boolean  
    ' Cek apabila entri kosong  
    Dim num_items_txt As String = txt.Text  
    If num_items_txt.Length < 1 Then  
        MessageBox.Show("Please enter " & field_name & ".")  
        txt.Focus()  
        Return False  
    End If  
  
    ' Apakah data numerik  
    If Not IsNumeric(num_items_txt) Then  
        MessageBox.Show(field_name & " must be a number.")  
        txt.Select(0, num_items_txt.Length)
```

```

        txt.Focus()
        Return False
    End If

    ' cek data pada text
    Try
        result = Integer.Parse(txt.Text)
    Catch ex As Exception
        MessageBox.Show("Error in " & field_name & ". " &
            vbCrLf & ex.Message)
        txt.Select(0, num_items_txt.Length)
        txt.Focus()
        Return False
    End Try

    ' Cek apakah range minimal dan maksimal sudah sesuai
    If result < min_value Or result > max_value Then
        MessageBox.Show(field_name & " must be between " &
            & _min_value.ToString & " and " &
            max_value.ToString & ".")
        txt.Select(0, num_items_txt.Length)
        txt.Focus()
        Return False
    End If
    ' Data sesuai (OK)
    Return True
End Function

```

Apabila angka yang dimasukkan lebih kecil dari angka 1, maka akan muncul pesan yang meminta Anda untuk memasukkan kembali value (nilai integer) ke dalam textbox. Kemudian function validate akan mengecek apakah yang Anda masukkan adalah angka. Apabila Anda tidak memasukkan angka, maka akan muncul pesan yang meminta Anda untuk memasukkan nilai integer. Apabila Anda memasukkan angka di luar dari angka 1 s.d. 10, maka function akan mendeteksi dan akan menampilkan pesan yang meminta Anda untuk memasukkan ulang.

Setelah Anda membuat function untuk mengecek bilangan integer, ketik function tersebut pada button validate dengan perintah sebagai berikut:

```

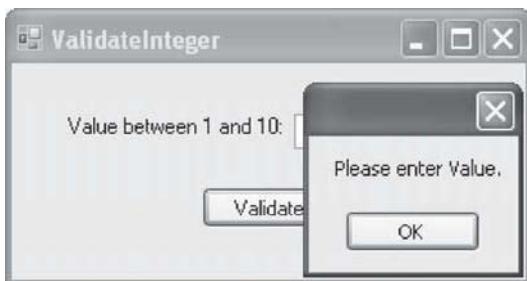
Private Sub btnValidate_Click() Handles btnValidate.Click
    Dim the_value As Integer

    ' Cek apakah validasi sesuai
    If Not IsValidInteger(the_value, txtValue, "Value", 1,
        10) Then Exit Sub

```

```
' Tempat meletakkan perintah proses selanjutnya  
' ...  
    MessageBox.Show("OK")  
End Sub
```

Setelah selesai membuat perintah program, Anda tekan tombol F5 untuk menjalankan aplikasi, sehingga tampil seperti gambar di bawah ini.



Gambar 8.31 Hasil aplikasi ValidateInteger

<http://pustaka-indo.blogspot.com>



APLIKASI GRAPHIC

<http://pustaka-indo.blogspot.com>

Graphic adalah kumpulan titik-titik yang tergabung menjadi satu sehingga membentuk sebuah garis dan gambar. Garis-garis terbentuk pada sebuah layar dengan menggunakan koordinat x dan y. Koordinat inilah yang akan menggabungkan titik-titik tersebut sehingga membentuk sebuah garis. Selain garis, Anda dapat juga membuat shape (rectangle dan ellipse), draw text, dan image. Buatlah project baru lalu klik ganda form, kemudian pilih event paint dan ketik perintah untuk membuat sebuah garis dengan Drawline.

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Me.Paint  
'Menggambar garis dengan 400 pixel berwarna hitam 25 pixels  
'dari atas form.  
e.Graphics.DrawLine(Pens.Black, 0, 25, 400, 25)
```

```
e.Graphics.DrawLine(Pens.Red, 100, 0, 100, 500)
e.Graphics.DrawLine(Pens.Blue, 0, 0, Me.Width, Me.Height)
End Sub
```

Setelah program kita jalankan, maka akan muncul garis mendatar dengan warna hitam, garis tegak lurus dengan warna merah, dan garis diagonal dengan warna biru.

9.1 Aplikasi Draw Text Path

Aplikasi draw text path adalah sebuah program yang menampilkan teks dalam modus grafik pada sebuah form. Untuk membuat aplikasi ini, buat file atau project baru dengan mengklik File > New Project. Buat nama project (misalnya DrawTextPath) lalu klik button OK.



Gambar 9.1 Desain form aplikasi DrawTextPath

Klik ganda form sehingga methode name load muncul, kemudian gantilah load tersebut dengan memilih event paint lalu ketik perintah:



Gambar 9.2 Method name

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Create a GraphicsPath.
    Using graphics_path As New Drawing2D.GraphicsPath
        ' Add some text to the path.
        graphics_path.AddString("AKMI Baturaja", _
            New FontFamily("Times New Roman"), _
            CInt(FontStyle.Bold), _
```

```

80, New Point(10, 10), _
StringFormat.GenericTypographic)

' Draw the path.
e.Graphics.SmoothingMode = SmoothingMode.AntiAlias
e.Graphics.FillPath(Brushes.White, graphics_path)
e.Graphics.DrawPath(New Pen(Color.Black, 3),
    graphics_path)
End Using
End Sub

```



Gambar 9.3 Hasil aplikasi DrawTextPath

Tambahkan perintah Imports System.Drawing.Drawing2D di atas public class seperti terlihat pada listing program berikut:

```

Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Create a GraphicsPath.
    Using graphics_path As New Drawing2D.GraphicsPath
        ' Add some text to the path.
        graphics_path.AddString("AKMI Baturaja", _
            New FontFamily("Times New Roman"), _
            CInt(FontStyle.Bold), _
            80, New Point(10, 10), _
            StringFormat.GenericTypographic)

        ' Draw the path.
        e.Graphics.SmoothingMode = SmoothingMode.AntiAlias
        e.Graphics.FillPath(Brushes.White, graphics_path)
        e.Graphics.DrawPath(New Pen(Color.Black, 3),
            graphics_path)
    End Using
End Sub

```

9.2 Aplikasi Draw Bezier

Aplikasi Bezier adalah aplikasi yang menggunakan komponen grafik (graphics), dan untuk menggambar garis tersebut digunakan perintah DrawBezier. Untuk membuat aplikasi tersebut, klik File > New Project dan buat nama project (misalnya bezier) lalu klik button OK. Klik ganda form dan ganti methode load menjadi paint dengan cara memilih paint dengan mouse lalu ketik perintah untuk menggambar bezier.

```
Public Class Form1
    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.EventArgs)
        ' Define the Bezier curve's control points.
        Dim pts() As Point = {
            New Point(10, 10),
            New Point(200, 10),
            New Point(50, 200),
            New Point(200, 150)
        }

        ' Connect the points with dashed lines.
        Using dashed_pen As New Pen(Color.Black, 0)
            dashed_pen.DashStyle = Drawing2D.DashStyle.Dash
            For i As Integer = 0 To 2
                e.Graphics.DrawLine(dashed_pen, pts(i), pts(i + 1))
            Next i
        End Using

        ' Draw the Bezier curve.
        e.Graphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Using bez_pen As New Pen(Color.Black, 3)
            e.Graphics.DrawBezier(bez_pen, pts(0), pts(1), pts(2), pts(3))
        End Using
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End Sub
End Class
```

Gambar 9.4 Perintah Bezier dan Method paint

Untuk menggambar Bezier, tentukan titik-titik lokasi menggunakan perintah:

```
Dim pts() As Point = {
    New Point(10, 10),
    New Point(200, 10),
    New Point(50, 200),
    New Point(200, 150)
}
```

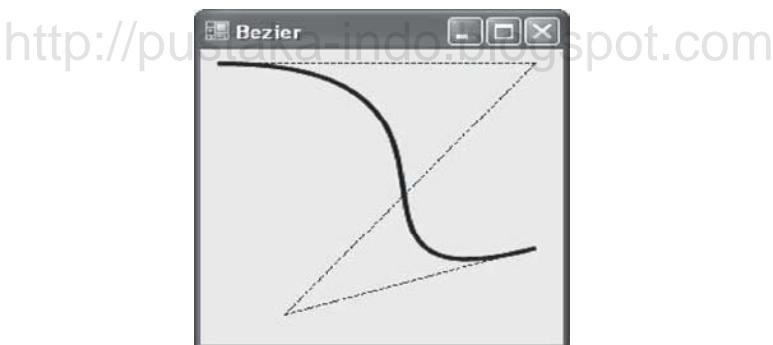
Setelah point tersebut ditentukan, selanjutnya adalah menentukan warna dan garis yang digunakan.

```
Using dashed_pen As New Pen(Color.Black, 0)
    dashed_pen.DashStyle = Drawing2D.DashStyle.Dash
    For i As Integer = 0 To 2
        e.Graphics.DrawLine(dashed_pen, pts(i), pts(i + 1))
    Next i
End Using
```

Kemudian kita akan memasukkan perintah untuk menggambar bezier tersebut pada baris berikutnya.

```
e.Graphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
Using bez_pen As New Pen(Color.Black, 3)
e.Graphics.DrawBezier(bez_pen, pts(0), pts(1), pts(2), pts(3))
End Using
```

Apabila Anda sudah selesai mengetik perintah tersebut, tekan tombol F5 untuk menjalankan program.



Gambar 9.5 Bezier

9.3 Aplikasi Arsir Garis

Untuk membuat aplikasi arsir, Anda klik menu File > New Project, lalu beri nama project dan klik button OK. Klik ganda form kemudian ketik perintah berikut:

```
Private Sub Form1_Load() Handles MyBase.Load
    Me.SetStyle(ControlStyles.AllPaintingInWmPaint Or
    ControlStyles.ResizeRedraw, True)
    Me.UpdateStyles()
End Sub
```

Setelah membuat perintah pada form load, Anda buat pula perintah dalam sebuah procedure dengan nama drawSamples.

```
Private Sub DrawSamples(ByVal gr As Graphics)
Dim styles As Array =
Drawing2D.HatchStyle.GetValues(GetType(Drawing2D.HatchStyle))

Dim x As Integer = 10
Dim y As Integer = 20
gr.Clear(Me.BackColor)
For Each hatch_style As Drawing2D.HatchStyle In styles
    DrawSample(gr, x, y, hatch_style)
Next hatch_style
End Sub
```

Kemudian buat pula procedure dengan nama DrawSample berikut ini:

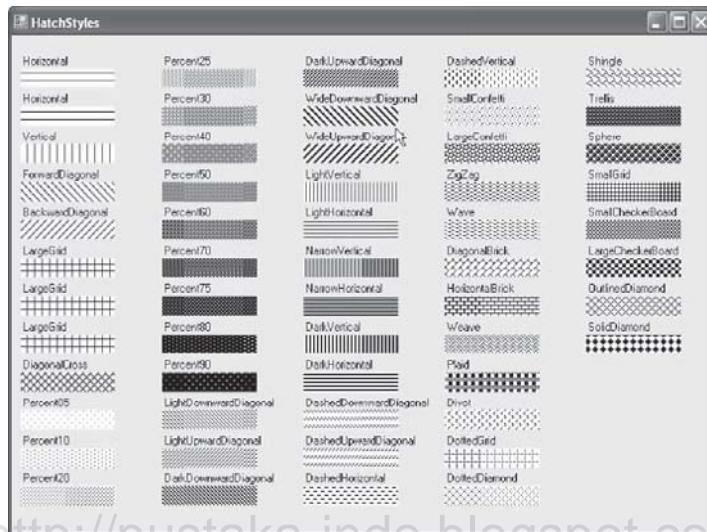
```
Private Sub DrawSample(ByVal gr As Graphics, ByRef x As
Integer, ByRef y As Integer, ByVal hatch_style As
Drawing2D.HatchStyle)
    gr.DrawString(hatch_style.ToString, Me.Font,
Brushes.Black, x, y)
    Using the_brush As New
        Drawing2D.HatchBrush(hatch_style, Color.Black,
Color.White)
            gr.FillRectangle(the_brush, x, y + 15, 100, 20)
        End Using

    y += 40
    If y + 40 > Me.ClientRectangle.Height Then
        x += 150
        y = 20
    End If
End Sub
```

Setelah procedure dibuat, Anda klik ganda form dan pilih paint kemudian buat perintah untuk menampilkan gambar menggunakan procedure DrawSamples dan DrawSample.

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles
MyBase.Paint
    DrawSamples(e.Graphics)
End Sub
```

Tekan tombol F5 untuk menjalankan aplikasi yang sudah Anda buat perintahnya tadi sehingga muncul gambar sebagai berikut:



<http://pustaka-indo.blogspot.com>

Gambar 9.6 Arsir

Perintah lengkap aplikasi arsir garis tersebut adalah sebagai berikut:

```
Public Class Form1
    Private Sub Form1_Load() Handles MyBase.Load
        Me.SetStyle(ControlStyles.AllPaintingInWmPaint Or
                    ControlStyles.ResizeRedraw, True)
        Me.UpdateStyles()
    End Sub

    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
        System.Windows.Forms.PaintEventArgs) Handles
        MyBase.Paint
        DrawSamples(e.Graphics)
    End Sub

    Private Sub DrawSamples(ByVal gr As Graphics)
        ' Get an array of all HatchStyle values.
        Dim styles As Array =
            Drawing2D.HatchStyle.GetValues(GetType(Drawing2D.HatchStyle))

        ' Draw samples of all the styles.
        Dim x As Integer = 10
        Dim y As Integer = 20
        gr.Clear(Me.BackColor)
```

```

For Each hatch_style As Drawing2D.HatchStyle In styles
    DrawSample(gr, x, y, hatch_style)
Next hatch_style
End Sub

Private Sub DrawSample(ByVal gr As Graphics, ByRef x As
    Integer, ByRef y As Integer, ByVal hatch_style As
    Drawing2D.HatchStyle)
    gr.DrawString(hatch_style.ToString, Me.Font,
        Brushes.Black, x, y)
    Using the_brush As New
        Drawing2D.HatchBrush(hatch_style, Color.Black,
            Color.White)
            gr.FillRectangle(the_brush, x, y + 15, 100, 20)
    End Using

    y += 40
    If y + 40 > Me.ClientRectangle.Height Then
        x += 150
        y = 20
    End If
End Sub
End Class

```

9.4 Aplikasi Gambar Rectangle

Aplikasi ini berisi tentang perintah untuk menggambar menggunakan graphic rectangle. Untuk membuat aplikasi ini, Anda buat project baru kemudian masukkan tiga buah radio button, masing-masing untuk radio button large, radio button narrow, dan radio button Flipped seperti tampak pada desain form berikut:



Gambar 9.7 Desain aplikasi Rectangle

Klik ganda form dan buatlah procedure smiley dengan perintah berikut:

```
Private Sub DrawSmiley(ByVal gr As Graphics)
    Using the_pen As New Pen(Color.Black, 0)
        gr.FillEllipse(Brushes.Yellow, 0, 0, 1, 1)
        gr.DrawEllipse(the_pen, 0, 0, 1, 1)
        gr.DrawArc(the_pen, 0.2!, 0.2!, 0.6!, 0.6!, 0.0!, 80.0!)
        ' Smile.
        gr.FillEllipse(Brushes.Black, 0.4!, 0.4!, 0.2!, 0.25!)
        ' Nose
        gr.FillEllipse(Brushes.White, 0.25!, 0.15!, 0.2!, 0.25!)
        ' Left eye.
        gr.DrawEllipse(the_pen, 0.25!, 0.15!, 0.2!, 0.25!)
        gr.FillEllipse(Brushes.Black, 0.35!, 0.2!, 0.1!, 0.15!)
        gr.FillEllipse(Brushes.White, 0.55!, 0.15!, 0.2!, 0.25!)
        ' Right eye.
        gr.DrawEllipse(the_pen, 0.55!, 0.15!, 0.2!, 0.25!)
        gr.FillEllipse(Brushes.Black, 0.65!, 0.2!, 0.1!, 0.15!)
    End Using
End Sub
```

Setelah Anda membuat procedure DrawSmiley, ketik perintah procedure:

```
Private Sub MapRectangles(ByVal gr As Graphics, ByVal
world_rect As Rectangle, ByVal device_rect As Rectangle)
' Reset the transformation.
gr.ResetTransform()
' Translate to center the world coordinate
' rectangle at the origin.
gr.TranslateTransform(
    CSng(-(world_rect.X + world_rect.Width / 2)), -
    CSng(-(world_rect.Y + world_rect.Height / 2)), -
    MatrixOrder.Append)

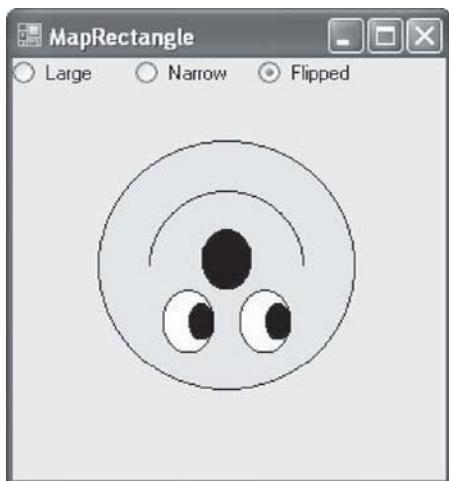
' Scale.
gr.ScaleTransform(
    CSng(device_rect.Width / world_rect.Width), -
    CSng(device_rect.Height / world_rect.Height), -
    MatrixOrder.Append)

' Translate to move the origin to the center
' of the device coordinate rectangle.
gr.TranslateTransform(
    CSng(device_rect.X + device_rect.Width / 2), -
    CSng(device_rect.Y + device_rect.Height / 2), -
    MatrixOrder.Append)
End Sub
```

Klik ganda Radio Button Large dan buatlah perintah berikut ini:

```
Private Sub rad_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles radLarge.CheckedChanged,
radNarrow.CheckedChanged, radFlipped.CheckedChanged
    Me.Invalidate()
End Sub
```

Tekan Tombol F5 untuk menjalankan program sehingga akan muncul tampilan berikut:



Gambar 9.8 Hasil aplikasi Rectangle

<http://pustaka-indo.blogspot.com>

Berikut ini perintah lengkap dari aplikasi:

```
Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Draw the target rectangle.
    Dim device_rect As Rectangle
    If radLarge.Checked Then
        device_rect = New Rectangle(50, 50, 150, 150)
    ElseIf radNarrow.Checked Then
        device_rect = New Rectangle(50, 50, 50, 150)
    Else
        device_rect = New Rectangle(50, 200, 150, -150)
    End If

    e.Graphics.DrawRectangle(Pens.Black, device_rect)

    ' Map between world and device coordinate rectangles.
    Dim world_rect As New Rectangle(0, 0, 1, 1)
    MapRectangles(e.Graphics, world_rect, device_rect)

    ' Draw the smiley face.
    DrawSmiley(e.Graphics)
End Sub
```

```

' Draw a smiley face in the rectangle
' 0 <= X <= 1, 0 <= Y <= 1.
Private Sub DrawSmiley(ByVal gr As Graphics)
    Using the_pen As New Pen(Color.Black, 0)
        gr.FillEllipse(Brushes.Yellow, 0, 0, 1, 1)
        gr.DrawEllipse(the_pen, 0, 0, 1, 1)
        gr.DrawArc(the_pen, 0.2!, 0.2!, 0.6!, 0.6!, 0.0!, 80.0!)
        ' Smile.
        gr.FillEllipse(Brushes.Black, 0.4!, 0.4!, 0.2!, 0.25!)
        ' Nose
        gr.FillEllipse(Brushes.White, 0.25!, 0.15!, 0.2!, 0.25!)
        ' Left eye.
        gr.DrawEllipse(the_pen, 0.25!, 0.15!, 0.2!, 0.25!)
        gr.FillEllipse(Brushes.Black, 0.35!, 0.2!, 0.1!, 0.15!)
        gr.FillEllipse(Brushes.White, 0.55!, 0.15!, 0.2!, 0.25!)
        ' Right eye.
        gr.DrawEllipse(the_pen, 0.55!, 0.15!, 0.2!, 0.25!)
        gr.FillEllipse(Brushes.Black, 0.65!, 0.2!, 0.1!, 0.15!)
    End Using
End Sub

Private Sub MapRectangles(ByVal gr As Graphics, ByVal
world_rect As Rectangle, ByVal device_rect As Rectangle)
' Reset the transformation.
    gr.ResetTransform()

    ' Translate to center the world coordinate
    ' rectangle at the origin.
    gr.TranslateTransform(
        CSng(-(world_rect.X + world_rect.Width / 2)), -
        CSng(-(world_rect.Y + world_rect.Height / 2)), -
        MatrixOrder.Append)

    ' Scale.
    gr.ScaleTransform(
        CSng(device_rect.Width / world_rect.Width), -
        CSng(device_rect.Height / world_rect.Height), -
        MatrixOrder.Append)

    ' Translate to move the origin to the center
    ' of the device coordinate rectangle.
    gr.TranslateTransform(
        CSng(device_rect.X + device_rect.Width / 2), -
        CSng(device_rect.Y + device_rect.Height / 2), -
        MatrixOrder.Append)
End Sub

Private Sub rad_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles radLarge.CheckedChanged,
radNarrow.CheckedChanged, radFlipped.CheckedChanged
    Me.Invalidate()
End Sub

```

9.5 Aplikasi Save Restore

Aplikasi Save dan Restore merupakan program dengan modus grafis yang menggunakan perintah graphics yang dapat menyimpan gambar dan menampilkan kembali grafik yang sudah tersimpan tersebut. Untuk membuat aplikasi ini, Anda buat project baru kemudian klik ganda form, lalu ganti event load dengan event paint dengan cara memilih event tersebut menggunakan mouse. Ketiklah perintah progam berikut ini:

```
Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    Using tghe_pen As New Pen(Color.Blue, 0)
        ' Scale by a factor of 90.
        e.Graphics.ScaleTransform(90, 90, MatrixOrder.Append)

        ' Translate to center on the form.
        e.Graphics.TranslateTransform(
            Me.ClientRectangle.Width \ 2, -
            Me.ClientRectangle.Height \ 2, -
            MatrixOrder.Append)

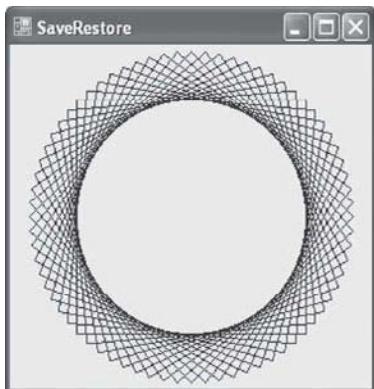
        For i As Integer = 5 To 90 Step 5
            ' Save the state.
            Dim graphics_state As GraphicsState = e.Graphics.Save()

            ' Rotate i degrees.
            e.Graphics.RotateTransform(i, MatrixOrder.Prepend)

            ' Draw a rectangle.
            e.Graphics.DrawRectangle(tghe_pen, -1, -1, 2, 2)

            ' Restore the saved state.
            e.Graphics.Restore(graphics_state)
        Next i
    End Using
End Sub
End Class
```

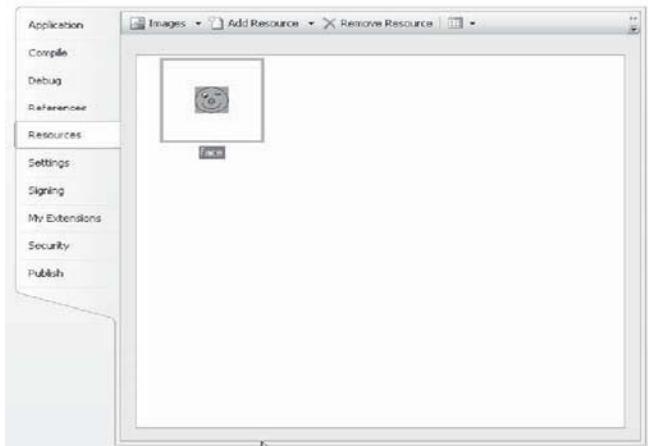
Apabila Anda sudah selesai membuat perintah, tekan tombol F5 untuk menampilkan hasil dari perintah yang sudah Anda ketik.



Gambar 9.9 Aplikasi SaveRestore

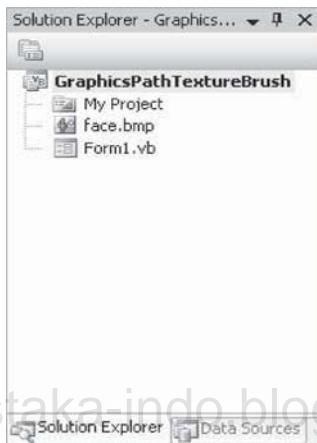
9.6 Aplikasi Graphic Texture

Graphic texture adalah gambar yang dipadukan dengan tulisan sehingga membentuk graphic texture. Untuk membuat aplikasi ini Anda klik File > New Project lalu dan buat nama project (GraphicPath TextureBrush) dan klik OK.



Gambar 9.10 Resources Image

Masukkan sebuah gambar bitmap dengan extention bmp ke dalam project dengan cara mengklik ganda my project sehingga tampil seperti Gambar 9.10. Klik Resources lalu pilih images, kemudian klik list add Resource dan pilih file face.bmp. Setelah memilih file face.bmp tersebut maka gambar akan masuk pada bagian project.

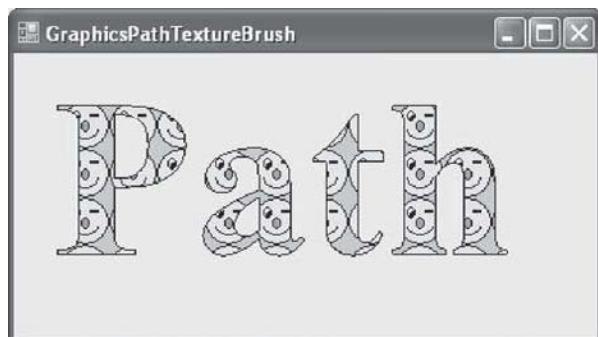


Gambar 9.11 My Project

Klik ganda form kemudian pilih event paint dan ketiklah perintah program berikut ini:

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    Dim txt As String = "Path"
    Using graphics_path As New GraphicsPath
        graphics_path.AddString(txt, _
            New FontFamily("Times New Roman"), _
            FontStyle.Bold, 150, _
            New Point(0, 0), _
            New StringFormat())
        Using smiley_brush As New TextureBrush(My.Resources.face)
            e.Graphics.FillPath(smiley_brush,
                graphics_path)
        End Using
        e.Graphics.DrawPath(Pens.Black, graphics_path)
    End Using
End Sub
```

Setelah Anda selesai membuat perintah program, tekan tombol F5 untuk menjalankan aplikasi yang sudah dibuat.



Gambar 9.12 Aplikasi GrapicsTexture

9.7 Aplikasi PathClip

Aplikasi pathclip dibuat dengan graphic line secara random (acak) yang memenuhi tiap bagian dari tulisan. Untuk membuat aplikasi ini Anda buat project baru dan beri nama GrapicsPathClip. Klik ganda form kemudian Anda pilih event paint dan ketikkan perintah imports System.Drawing.Drawing2D, lalu ketik perintah untuk membuat graphic sebagai berikut:

```
Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Make a GraphicsPath containing text.
    Dim txt As String = "Path"
    Using graphics_path As New GraphicsPath
        graphics_path.AddString(txt, _
            New FontFamily("Times New Roman"), _
            FontStyle.Bold, 150, _
            New Point(0, 0), _
            New StringFormat)
        e.Graphics.SetClip(graphics_path)
    End Using
```

```

' Fill the ClientRectangle with white.
e.Graphics.FillRectangle(Brushes.White,
Me.ClientRectangle)

' Draw a bunch of random lines on the form.
Dim rnd As New Random
Dim x1, y1, x2, y2 As Integer
For i As Integer = 1 To 200
    x1 = rnd.Next(0, Me.ClientSize.Width - 1)
    y1 = rnd.Next(0, Me.ClientSize.Height - 1)
    x2 = rnd.Next(0, Me.ClientSize.Width - 1)
    y2 = rnd.Next(0, Me.ClientSize.Height - 1)
    e.Graphics.DrawLine(Pens.Black, x1, y1, x2, y2)
Next i
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
End Sub
End Class

```



Gambar 9.13 Aplikasi GraphicPathClip

Tekan tombol F5 atau klik menu debug, kemudian pilih submenu *start debugging* untuk menjalankan program aplikasi.

9.8 Aplikasi Transform Text

Aplikasi ini Anda buat untuk memutar tulisan yang sudah didefinisikan dalam sebuah variabel string. Untuk membuat aplikasi ini sama caranya dengan membuat aplikasi graphic sebelumnya, dan masih tetap menggunakan event paint dengan perintah sebagai berikut:

```

Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Load() Handles MyBase.Load
Me.SetStyle(ControlStyles.AllPaintingInWmPaint Or
ControlStyles.ResizeRedraw, True)
    Me.UpdateStyles()
End Sub

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    Dim rect_wid As Integer = Me.ClientSize.Height - 20
    Dim rect_hgt As Integer = Me.ClientSize.Width - 20
    Dim layout_rect As New RectangleF(
        (Me.ClientSize.Width - rect_wid) \ 2, _
        (Me.ClientSize.Height - rect_hgt) \ 2, _
        rect_wid, rect_hgt)

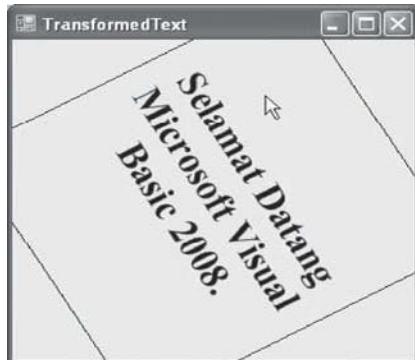
    ' Setting jenis huruf.
Dim txt As String="Selamat Datang Microsoft Visual Basic 2008"
Using the_font As New Font("Times New Roman", 30,
FontStyle.Bold, GraphicsUnit.Pixel)

    ' Mengatur tulisan di tengah
Using string_format As New StringFormat
    string_format.Alignment = StringAlignment.Center
    string_format.LineAlignment = StringAlignment.Center

    ' Perintah untuk memutar tulisan.
    e.Graphics.TranslateTransform(-Me.ClientSize.Width \
2, -Me.ClientSize.Height \ 2, MatrixOrder.Append)
    e.Graphics.RotateTransform(60, MatrixOrder.Append)
        e.Graphics.TranslateTransform(Me.ClientSize.Width \
2, Me.ClientSize.Height \ 2, MatrixOrder.Append)

    ' Draw the text and layout rectangle.
    e.Graphics.TextRenderingHint =
        System.Drawing.Text.TextRenderingHint.AntiAliasGridFit
        e.Graphics.DrawString(txt, the_font,
        Brushes.Black, layout_rect, string_format)
        e.Graphics.DrawRectangle(Pens.Black,
        Rectangle.Round(layout_rect))
End Using ' string_format
End Using ' the_font
End Sub
End Class

```



Gambar 9.14 Hasil aplikasi Transform Text

Jalankan aplikasi dengan menekan tombol F5 sehingga aplikasi akan terlihat seperti gambar di atas.

9.9 Aplikasi LoadPicture

Aplikasi loadPicture adalah program graphic yang menampilkan gambar melalui komponen picture image. Untuk membuat aplikasi ini, Anda klik File > New Project kemudian buat nama project (loadpicture) dan klik button OK. Setelah tampil form, masukkan komponen picture image ke dalam form, lalu Anda klik ganda form dan ketiklah perintah program berikut ini.

```
Public Class Form1
    'Menggunakan event load bukan event paint
    Private Sub Form1_Load() Handles MyBase.Load
        Dim file_name As String = Application.StartupPath
        file_name = file_name.Substring(0,
            file_name.LastIndexOf("\"))
        file_name = file_name.Substring(0,
            file_name.LastIndexOf("\"))
        file_name &= "\HalfJack2.bmp"

        ' Load the bitmap file.
        Dim bm As New Bitmap(file_name)

        ' Make a copy.
        Dim new_bm As New Bitmap(bm)
```

```

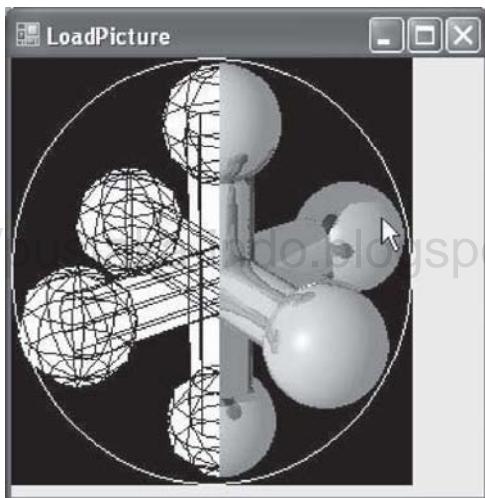
' Dispose of the original Bitmap.
bm.Dispose()

' Draw on the new Bitmap and display the result.
Using gr As Graphics = Graphics.FromImage(new_bm)
    gr.DrawEllipse(Pens.White, 0, 0, new_bm.Width - 1,
    new_bm.Height - 1)
End Using

picImage.Image = new_bm
End Sub
End Class

```

Setelah perintah program selesai diketik, tekan tombol F5 untuk menjalankan aplikasi.



Gambar 9.15 Aplikasi LoadPicture

9.10 Aplikasi AutoRedraw

Aplikasi autoRedraw ini dibuat menggunakan perintah FillPolygon dan DrawPoligon. Buka sebuah file project baru dan beri nama autoredraw. Klik ganda form dan buat procedure drawdiamond dengan perintah berikut ini:

```

Private Sub DrawDiamond()
    ' Get the drawing surface's size.
    Dim wid As Integer = picCanvas.ClientSize.Width
    Dim hgt As Integer = picCanvas.ClientSize.Height

    If wid < 1 Or hgt < 1 Then Exit Sub

    ' Membuat sebuah gambar Bitmap Dan Graphics
    Dim bm As New Bitmap(wid, hgt)
    Using gr As Graphics = Graphics.FromImage(bm)
        ' Fill the drawing area with a hatch pattern.
        Using bg_brush As New HatchBrush(HatchStyle.HorizontalBrick, _
        Color.Blue, Color.Aqua)
            gr.FillRectangle(bg_brush, picCanvas.ClientRectangle)
        End Using ' bg_brush

        ' Gambar kotak dimond.
        Dim pts() As Point = { _
        New Point(wid \ 2, 0), _
        New Point(wid, hgt \ 2), _
        New Point(wid \ 2, hgt), _
        New Point(0, hgt \ 2) _
        }
        gr.FillPolygon(Brushes.White, pts)
        gr.DrawPolygon(Pens.Black, pts)

        ' Display the result.
        picCanvas.Image = bm
    End Using ' gr
End Sub

```

<http://pustaka-indo.blogspot.com>

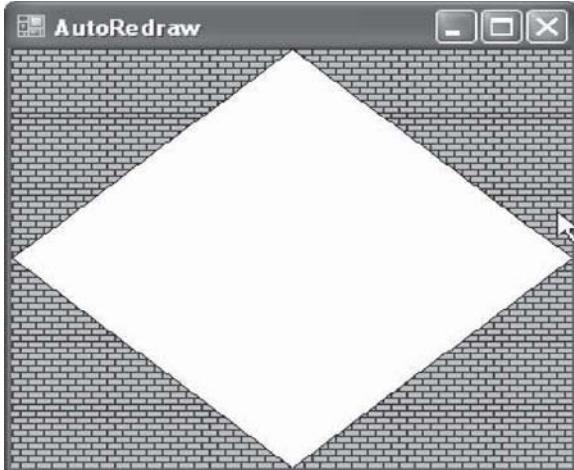
Buat perintah untuk memanggil procedure drawDiamont pada event load. Perintah untuk memanggil procedure tersebut adalah sebagai berikut:

```

Private Sub Form1_Load() Handles MyBase.Load
    DrawDiamond()
End Sub

```

Jalankan aplikasi dengan menekan tombol F5 sehingga akan muncul Gambar 9.16.



Gambar 9.16 Aplikasi AutoRedraw

Perintah lengkap aplikasi autodraw adalah sebagai berikut:

```
Imports System.Drawing.Imaging
Imports System.Drawing.Drawing2D
Public Class Form1
    ' Draw the initial image.
    Private Sub Form1_Load() Handles MyBase.Load
        DrawDiamond()
    End Sub

    ' Draw the an image with the new size.
    Private Sub Form1_Resize() Handles MyBase.Resize
        DrawDiamond()
    End Sub

    Private Sub DrawDiamond()
        ' Get the drawing surface's size.
        Dim wid As Integer = picCanvas.ClientSize.Width
        Dim hgt As Integer = picCanvas.ClientSize.Height

        If wid < 1 Or hgt < 1 Then Exit Sub

        ' Membuat sebuah gambar Bitmap Dan Graphics
        Dim bm As New Bitmap(wid, hgt)
        Using gr As Graphics = Graphics.FromImage(bm)
            ' Fill the drawing area with a hatch pattern.
            Using bg_brush As New HatchBrush(HatchStyle.HorizontalBrick,
                Color.Blue, Color.Aqua)
                gr.FillRectangle(bg_brush, picCanvas.ClientRectangle)
            End Using
        End Using
    End Sub

```

```

' Gambar kotak dimond.
Dim pts() As Point = { _
    New Point(wid \ 2, 0), _
    New Point(wid, hgt \ 2), _
    New Point(wid \ 2, hgt), _
    New Point(0, hgt \ 2) _
}
gr.FillPolygon(Brushes.White, pts)
gr.DrawPolygon(Pens.Black, pts)

' Display the result.
picCanvas.Image = bm
End Using ' gr
End Sub
End Class

```

9.11 Aplikasi Transform Arrow

Aplikasi transform arrow adalah program graphic yang dibuat dengan perintah fillPolygon yang berbentuk panah. Untuk membuat program graphic ini Anda buat sebuah file project baru, dan Anda beri nama transformArrow. Klik ganda form dan ketik procedure:

```

Private Sub DrawArrow(ByVal gr As Graphics, ByVal hatch_style As HatchStyle)
    Dim pts() As Point = { _
        New Point(0, 1), _
        New Point(2, 1), _
        New Point(2, 0), _
        New Point(4, 2), _
        New Point(2, 4), _
        New Point(2, 3), _
        New Point(0, 3) _
    }
    Using hatch_brush As New HatchBrush(hatch_style, Color.Black, Color.White)
        gr.FillPolygon(hatch_brush, pts)
    End Using

    Using black_pen As New Pen(Color.Black, 0)
        gr.DrawPolygon(black_pen, pts)
    End Using
End Sub

```

Setelah membuat procedure DrawArrow ketik perintah untuk event paint dengan perintah berikut:

```

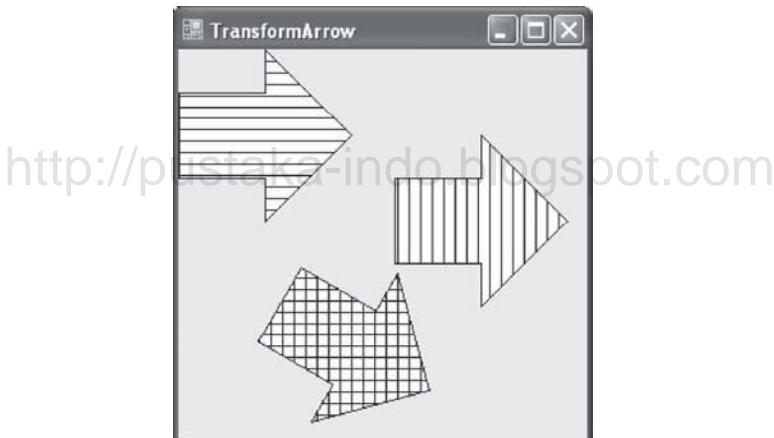
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Scale by a factor of 30.
    e.Graphics.ScaleTransform(30, 30, MatrixOrder.Append)
    DrawArrow(e.Graphics, HatchStyle.Horizontal)

    ' Translate 150 horizontally and 60 vertically.
    e.Graphics.TranslateTransform(150, 60,
        MatrixOrder.Append)
    DrawArrow(e.Graphics, HatchStyle.Vertical)

    ' Rotate 30 degrees.
    e.Graphics.RotateTransform(30, MatrixOrder.Append)
    DrawArrow(e.Graphics, HatchStyle.Cross)
End Sub

```

Tekan tombol F5 pada keyboard untuk menjalankan aplikasi yang sudah Anda buat hingga tampil gambar berikut:



Gambar 9.17 Aplikasi TranformArrow

Perintah lengkap aplikasi transformArrow:

```

Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Scale by a factor of 30.
    e.Graphics.ScaleTransform(30, 30, MatrixOrder.Append)
    DrawArrow(e.Graphics, HatchStyle.Horizontal)

```

```

' Translate 150 horizontally and 60 vertically.
e.Graphics.TranslateTransform(150, 60,
MatrixOrder.Append)
DrawArrow(e.Graphics, HatchStyle.Vertical)

' Rotate 30 degrees.
e.Graphics.RotateTransform(30, MatrixOrder.Append)
DrawArrow(e.Graphics, HatchStyle.Cross)
End Sub

Private Sub DrawArrow(ByVal gr As Graphics, ByVal hatch_style
As HatchStyle)
Dim pts() As Point = {
    New Point(0, 1), _
    New Point(2, 1), _
    New Point(2, 0), _
    New Point(4, 2), _
    New Point(2, 4), _
    New Point(2, 3), _
    New Point(0, 3) _
}
Using hatch_brush As New HatchBrush(hatch_style,
    Color.Black, Color.White)
    gr.FillPolygon(hatch_brush, pts)
End Using

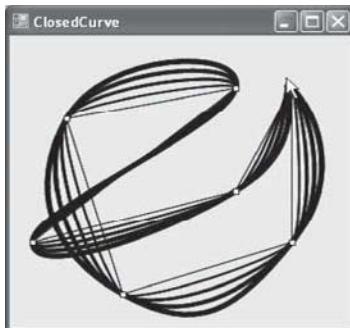
Using black_pen As New Pen(Color.Black, 0)
    gr.DrawPolygon(black_pen, pts)
End Using
End Sub
End Class

```

<http://pustaka-indo.blogspot.com>

9.12 Aplikasi Closed Curve

Aplikasi closed curve adalah aplikasi yang dibuat dengan perintah graphic Fill dan drawRectangle.



Gambar 9.18 Aplikasi Closed Curve

Untuk membuat aplikasi tersebut, buatlah file project baru dan beri nama closed project. Untuk membuat perintah, klik ganda form dan buatlah perintah berikut ini:

```
Public Class Form1
    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
        ' Point untuk membuat curva
        Dim pts() As Point = {
            New Point(50, 80), _
            New Point(200, 50), _
            New Point(20, 200), _
            New Point(200, 150), _
            New Point(250, 50), _
            New Point(250, 200), _
            New Point(100, 250) _
        }

        ' Draw the closed curve.
        e.Graphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        If False Then
            ' Draw just one curve.
            Using curve_pen As New Pen(Color.Black, 3)
                e.Graphics.DrawClosedCurve(curve_pen, pts)
            End Using
        Else
            ' Draw a series of curves with different tensions.
            For tension As Single = 0 To 1 Step 0.25
                Using curve_pen As New Pen(Color.Black, tension * 4 + 1)
                    e.Graphics.DrawClosedCurve(curve_pen, pts,
                        tension,
                        Drawing2D.FillMode.Alternate)
                End Using
            Next tension
        End If

        ' Draw rectangles on the control points.
        For i As Integer = 0 To pts.Length - 1
            e.Graphics.FillRectangle(Brushes.White, pts(i).X - 2,
                pts(i).Y - 2, 5, 5)
            e.Graphics.DrawRectangle(Pens.Black, pts(i).X - 2,
                pts(i).Y - 2, 5, 5)
        Next i
    End Sub
End Class
```

9.13 Aplikasi LineCaps

Aplikasi line caps adalah program yang dibuat dengan perintah graphic drawline. Untuk membuat aplikasi ini, caranya hampir sama dengan aplikasi lain, yaitu dengan mengklik ganda form dan ketik perintah program dalam sebuah procedure DrawSample.

```
Private Sub DrawSample(ByVal gr As Graphics, ByRef x As Integer, ByRef y As Integer, ByVal style As Drawing2D.LineCap)
    Using the_pen As New Pen(Color.Black, 10)
        the_pen.EndCap = style
        gr.DrawLine(the_pen, x + 100, y, x + 150, y)
    End Using

    gr.DrawString(style.ToString, Me.Font, Brushes.Black,
        x, y - 5)

    y += 30
    If y > Me.ClientRectangle.Height Then
        x += 200
        y = 17
    End If
End Sub
```

Setelah membuat procedure Drawline, Anda buat perintah program untuk event paint sebagai berikut:

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Get an array of the available LineCap values.
    Dim line_caps As Array =
    Drawing2D.LineCap.GetValues( GetType(Drawing2D.LineCap) )
    ' Display the values.
    Dim x As Integer = 5
    Dim y As Integer = 17
    For Each line_cap As Drawing2D.LineCap In line_caps
        DrawSample(e.Graphics, x, y, line_cap)
    Next line_cap
End Sub
```

Jalankan program dengan menekan tombol F5 hingga tampil gambar berikut ini:



Gambar 9.19 Aplikasi LineCaps

Untuk lebih jelasnya lihat perintah program berikut ini:

```
Imports System.Drawing.Drawing2D

Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
    ' Get an array of the available LineCap values.
    Dim line_caps As Array =
        Drawing2D.LineCap.GetValues(
        GetType(Drawing2D.LineCap))

    ' Display the values.
    Dim x As Integer = 5
    Dim y As Integer = 17
    For Each line_cap As Drawing2D.LineCap In line_caps
        DrawSample(e.Graphics, x, y, line_cap)
    Next line_cap
End Sub

Private Sub DrawSample(ByVal gr As Graphics, ByRef x As Integer, ByRef y As Integer, ByVal style As Drawing2D.LineCap)
    Using the_pen As New Pen(Color.Black, 10)
        the_pen.EndCap = style
        gr.DrawLine(the_pen, x + 100, y, x + 150, y)
    End Using

    gr.DrawString(style.ToString, Me.Font, Brushes.Black, x - 5)
    y += 30
    If y > Me.ClientRectangle.Height Then
        x += 200
        y = 17
    End If
End Sub
End Class
```

9.14 Aplikasi List Installed Font

Aplikasi list installed Font adalah aplikasi yang menampilkan daftar jenis huruf yang sudah ter-install di dalam komputer Anda. Untuk membuat aplikasi ini, buat sebuah file project baru dan beri nama listinstalledfont. Masukkan sebuah komponen ListBox dan isi properties name dengan lstfonts kemudian masukkan pula sebuah komponen picture dan isi properties dengan nama picSample. Komponen listbox digunakan untuk meletakkan jenis huruf yang terdapat di dalam komputer, sedangkan komponen picture digunakan untuk menampilkan bentuk font. Untuk lebih jelasnya perhatikan gambar desain aplikasi berikut ini.



Gambar 9.20 Desain aplikasi List Installed Font

Klik ganda form dan ketik perintah program berikut ini pada event load:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load  
    ' Get the installed fonts collection.  
    Dim installed_fonts As New InstalledFontCollection  
  
    ' Get an array of the system's font families.  
    Dim font_families() As FontFamily =  
        installed_fonts.Families()  
  
    ' Display the font families.  
    For Each font_family As FontFamily In font_families  
        lstFonts.Items.Add(font_family.Name)
```

```
Next font_family  
    lstFonts.SelectedIndex = 0  
End Sub
```

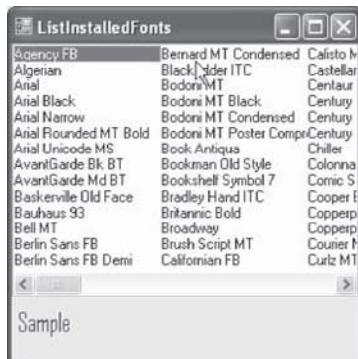
Kemudian klik ganda komponen listbox dan ketik perintah program pada event SelectedIndexChanged:

```
Private Sub lstFonts_SelectedIndexChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
lstFonts.SelectedIndexChanged  
    m_Font = lstFonts.SelectedItem  
    picSample.Invalidate()  
End Sub
```

Selanjutnya klik ganda komponen picture dan ketik perintah tersebut pada event paint.

```
Private Sub picSample_Paint(ByVal sender As Object, ByVal e As  
System.Windows.Forms.PaintEventArgs) Handles picSample.Paint  
    e.Graphics.Clear(picSample.BackColor)  
    If m_Font.Length < 1 Then Exit Sub  
  
    On Error Resume Next  
    Using the_font As New Font(m_Font, 16,  
        GraphicsUnit.Point)  
        e.Graphics.DrawString("Sample", the_font,  
            Brushes.Black, 0, 0)  
    End Using  
End Sub
```

Tekan tombol F5 untuk menjalankan aplikasi.



Gambar 9.21 Aplikasi List Install Font

Source code lengkap program adalah sebagai berikut:

```
Imports System.Drawing.Text

Public Class Form1
    Private m_Font As String = ""

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ' Get the installed fonts collection.
    Dim installed_fonts As New InstalledFontCollection

    ' Get an array of the system's font families.
    Dim font_families() As FontFamily =
    installed_fonts.Families()

    ' Display the font families.
    For Each font_family As FontFamily In font_families
        lstFonts.Items.Add(font_family.Name)
    Next font_family
    lstFonts.SelectedIndex = 0
End Sub

    ' Display a sample.
Private Sub lstFonts_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lstFonts.SelectedIndexChanged
    m_Font = lstFonts.SelectedItem
    picSample.Invalidate()
End Sub

    ' Draw a sample.
Private Sub picSample_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles picSample.Paint
    e.Graphics.Clear(picSample.BackColor)
    If m_Font.Length < 1 Then Exit Sub

    On Error Resume Next
    Using the_font As New Font(m_Font, 16, GraphicsUnit.Point)
        e.Graphics.DrawString("Sample", the_font, Brushes.Black, 0, 0)
    End Using
End Sub
End Class
```

DAFTAR PUSTAKA

<http://code.msdn.microsoft.com/vbsamples/>

http://en.wikipedia.org/wiki/Visual_Basic_.NET

<http://msdn.microsoft.com/en-ca/ms789075.aspx>

<http://searchwindevelopment.techtarget.com/>

<http://visualbasic.about.com/od/learnvbnet>

<http://www.gotmyidea.com/2007/09>

<http://www.programminglearn.com/>

<http://www.truveo.com/>

<http://www.tutorialized.com/tutorials/Visual-Basic/1>

<http://www.vbtutor.net/vb2008/vb2008tutor.html>

TENTANG PENULIS

Hendrayudi lahir di Baturaja Sumatera Selatan pada tanggal 27 Juli 1974.



Sekolah Dasar sampai dengan Sekolah Menengah Umum diselesaikan di Baturaja. Setelah tamat SMU, penulis melanjutkan kuliah di Universitas Bina Darma Palembang, Program Studi Sistem Informasi. Setelah menyelesaikan S1, penulis bekerja di STMIK Pramulih dan sekarang aktif di kampus "ping" AMIK AKMI Baturaja Sumatera Selatan sebagai Pudir III bidang Kemahasiswaan dan Marketing.

Penulis adalah staf ahli di AMIK AKMI Baturaja dan pernah sebagai kepala laboratorium hardware dan jaringan, kepala laboratorium software, dan kepala laboratorium internet. Penulis juga sebagai staf peneliti di LPPPM AKMI Sumatera Selatan, dan akan terus meneliti serta menulis buku-buku komputer berikutnya.

Buku-buku komputer terbaru Elex Media Komputindo:

ID	JUDUL	PENULIS	HARGA
121090370	Meraup Dollar Hanya dengan Klik!	Widianto	31,800
121090180	YUWIE: Berteman sambil Meraup Uang!	Stefan & Adhicipta R. Wirawan	32,800
121090215	Aplikasi Game dan Multimedia dengan VB + CD	Fajrillah Hasballah	49,800
121090259	Membangun Aplikasi Retail dan Toko Modern dengan Access 2007 + CD	Ahmad Muhlis	57,800
121090268	Teknik Makeover Blogspot	Dominikus Juju & MataMaya Studio	32,800
121090123	Step by Step Facebook	Sartika Kurniali	35,800
121090050	Desain Portal Perusahaan dengan SharePoint Designer 2007	V. Djong	21,800
121090098	Jurus Jitu Webmaster Freelance + CD	Dominikus Juju & Muhammad Syukrie	60,800
121082474	Membangun Blog WordPress	Oya Suryana & MataMaya Studio	28,800
121082499	Jurus-Jurus Tersembunyi Access 2007 + CD	Ahmad Muhlis	37,800
121082467	Pemrograman Handphone dengan J2ME + CD	Ir. Yuniar Supardi	64,800
121082444	Step by Step CMS Mambo + CD	Saropi, S.Kom	47,800
121082430	101 Tip & Trik Nero 9	Jubilee Enterprise	42,800
121082346	Pertolongan Pertama pada Insiden Hacking	Rahmat Rafiudin & Ganesha Progress	36,800
121082393	101 Tip & Trik Myspace	Jubilee Enterprise	39,800
121082268	Cari Duit Modal Dengkul Cara Blogger	Budi Kurniawan, S.Kom	29,800
121082269	Teknik Membobol Data dan Password	Muzammil Sanusi	44,800
121082210	Enterprise Project Management dengan Project Server 2007	Ronald Rajagukguk	19,800
121082195	RPG Studio + DVD	Frank Albert Pangajow	79,800
121082187	Rahasia Besar Photoshop + CD	Friza Reihan	36,800

Catatan:

- Untuk melakukan pemesanan, hubungi Layanan Langsung PT Elex Media Komputindo, telp. (021) 53650110, 111 ext 3901 & 3902.
- Harga di atas dapat berubah sewaktu-waktu tanpa pemberitahuan terlebih dahulu.

VB 2008

untuk Berbagai Keperluan

PEMROGRAMAN

Microsoft Visual Basic 2008 adalah bahasa pemrograman berbasis visual yang merupakan versi lanjutan dari Microsoft Visual Basic 6.0 yang tergabung dalam Microsoft Visual Studio bersama dengan pemrograman Visual C, C++, dan Visual FoxPro. Buku ini mengajak Anda mempelajari Visual Basic 2008 dalam berbagai bentuk perintah yang mudah dipahami untuk segala keperluan pemrograman Anda.

Buku ini juga dilengkapi dengan pembahasan database dan tabel yang membahas tentang cara mengisikan (input), menghapus (delete), menyimpan (save), dan mencari (search) data yang terdapat di dalam tabel. Juga membahas pengenalan perintah dalam membuat berbagai aplikasi teks dan grafik.

Pembahasan lengkap isi buku:

- Dasar bahasa pemrograman Microsoft Visual Basic 2008
- Instalasi Visual Basic 2008
- Komponen Visual Basic 2008
- Teknik mengetik dan membuat perintah
- Membuat program perintah-perintah Microsoft Visual Basic 2008
- Perintah String Module
- Perintah File System
- Perintah Data Time Module
- Perintah InterAction dan Constant Module
- Membuat aplikasi berbasis teks
- Membuat aplikasi grafik
- Berbagai aplikasi lainnya

Penerbit PT Elex Media Komputindo
Jl Palmerah Barat 29-37
Jakarta 10270
Telp. (021) 53650110 - 53650111
ext. 3201-3202
Web Page: <http://www.elexmedia.co.id>

Kelompok
Pemrograman
Kelembagaan
Tingkat Pemula
Tingkat Menengah
Tingkat Mahir
Jenis Buku
Referensi
Tutorial
Latihan

ISBN 978-979-27-4576-4



9 78979 2745764

EMK121090520