

ST-RK-1.16-082-007/R-



Modul Praktikum

PEMGROGRAMAN VISUAL I

Praktikan mengerti tentang control dan penggunaannya, tipe data, serta error handling pada pemrograman Visual Basic .Net



Join Us

**THE
CYBER
CAMPUS**
<http://www.stikom.edu>

Laboratorium Komputer

DAFTAR ISI

Daftar Isi	1
Pengantar Umum.....	4
MODUL 1	5
1.1. Tipe Data	6
1.2. Variable	7
1.3. Array.....	8
Menciptakan Array	9
Reinitializing Arrays	10
Menggunakan Variabel Array.....	11
Variabel Array Multi Dimensi.....	12
Multidimensional Arrays	12
Rectangular arrays.....	12
Jagged Arrays	13
1.4. Acces Modifier.....	14
1.5. Konversi Tipe Data.....	14
Secara Explicit.....	15
Ctype	15
Latihan	17
MODUL 2	23
2.1. Selection	24
If...Then...Else Statement.....	24
Pilih Antara Dua.....	24
Pilih Antara Banyak	25
Pilihan dalam pilihan.....	26
Select Case.....	26
Select case dengan Case Is.....	27
Select case dengan ..To.....	27
2.2. Looping	28
Do... Loop.....	28
Looping Counter	29
WhileLoop.....	30
For Each..... next.....	30
Struktur Kendali Nested.....	31
Meninggalkan suatu Struktur Kendali.....	32
Latihan	34
MODUL 3	41
3.1. Procedure	42
3.2. Function.....	43
3.3. Pemanggilan Prosedure dan Function	44

Latihan	46
MODUL 4	49
Latihan	50
MODUL 5	53
5.1. Exception Handling	54
Structured Exception Handling	54
Menangkap Sebuah Exception.....	55
Throw Exception	57
5.2. Error Provider.....	58
Latihan Modul 1	61
MODUL 6	67
6.1. Pengenalan Konsep OOP (Object Oriented Programming)	68
Class.....	68
Data Member dan Properties	69
Method	69
Constructor vs Destructor	69
Object adalah Instance of Class.....	71
Deklarasi dan Inisialisasi Object	72
Properties	72
Methods	72
6.2. Mengapa Harus OOP ?.....	72
Latihan	74
MODUL 7	81
7.1. Control dan Properties.....	82
Method dan Event.....	82
MenuStrip	83
ToolBar / ToolStrip.....	85
7.2. MDI Form.....	86
StatusBar / StatusStrip	88
Mengedit item status pada StatusStrip :.....	89
Melalui properties window :.....	89
Latihan	91
MODUL 8	93
8.1. Visual Studio .NET Deployment Project Templates	94
Setup Project Template	94
Web Setup Project Template	94
Merge Module Project Template.....	95
Setup Wizard Template	95
CAB Project Template.....	95
Smart Device CAB Project Template	95

8.2. Membuat Installer Package	95
Konfigurasi Project Properties	96
Jenis-Jenis Setup Editors	97
Menambah Additional Custom Dialogs	102
Membangun Installer Package	103
Latihan	104

PENGANTAR UMUM

Pada praktikum Pemrograman Visual 1, kita akan menggunakan Visual Basic.Net 2005 sebagai tools developer.

Dalam setiap latihan praktikum, diberikan alternatif penyelesaian, jika anda belum berhasil menyelesaikan, anda dapat mencoba dengan penyelesaian yang diberikan.

Latihan praktikum pada modul dapat membutuhkan waktu lebih dari 40 menit saat praktikum sehingga cobalah dirumah dan tanyakan kesulitan anda saat praktikum.

Tugas Praktikum akan diambil dari modul dan latihan praktikum yang anda, sehingga cobalah dan pelajari dengan benar.

Pada saat pelaksanaan praktikum sebaiknya anda telah mencoba latihan praktikum sehingga 40 menit yang disediakan dapat digunakan untuk mendiskusikan error handling terhadap kesalahan dan kesulitan anda saat mencoba.

Dalam setiap latihan akan diberikan alur atau logika untuk mengerjakan. Cobalah ikuti hal tersebut.

Username dan Password Windows

Username : nimpanjang

Password : nimpanjang

Simpan aplikasi pada komputer lokal terlebih dahulu. Setelah semua selesai baru dipindahkan ke server.

Untuk mapping server :

Buka menu Tools - Map Network Drive. Drive biarkan default, untuk Folder isi dengan \\data-server\.... ..\$. Klik Finish.

MODUL 1

VISUAL BASIC.NET BASIC PROGRAMMING

*Ciri Orang Sukses adalah Ketika tidak tahu dia
mencari, ketika tidak bisa dia mencoba, dan
ketika tidak mengerti dia bertanya
---Anonymous---*



Tujuan

Mahasiswa mampu memahami pemrograman dasar pada Visual Basic.Net meliputi Tipe Data, Variabel, Array dan Acces Modifier

Materi

Tipe Data (String, Decimal, Double, Integer, Single), Variabel, Array, Acces Modifier

Referensi

<http://www.publicjoe.f9.co.uk/vbnet/vb03b.html>
<http://visualbasic.about.com/od/usingvbnet/a/blinheritancea.htm>
<http://indoprog.tripod.com/>
<http://startvbdotnet.com/language/datatypes.aspx>

1.1. Tipe Data

Sebagaimana bahasa pemrograman yang lainnya, Visual Basic.Net juga menyediakan berbagai tipe data yang dapat digunakan oleh pemrogram sesuai dengan konteks pemakaiannya. Berikut adalah tabel beberapa tipe data yang tersedia dalam visual Basic. Net yang sering kita gunakan.

Data Type	Size in Bytes	Return Value	Type
Byte	1	8-bit unsigned integer	System.Byte
Char	2	16-bit Unicode characters	System.Char
Integer	3	32-bit signed integer	System.Int32
Double	8	64-bit floating point varabel	System.Double
Long	8	64-bit signed integer	System.Int64
Short	2	16-bit signed integer	System.Int16
Single	4	32-bit floating point varabel	System.Single
String	Varies	Non-Numeric Type	System.String
Date	8	0:00:00 on Januari 1, 0001 through 11:59:59	System.Date

PM on December 31, 9999			
Boolean	2	True or False	System.Boolean
Object	4	Any tipe can be stored in variable of type object	System.Object
Decimal	16	128-bit floating point variabel	System.Decimal

1.2. Variable

Variable merupakan tempat penyimpanan data. Variable memiliki sebuah nama disertai dengan tipe data yang akan ditampung oleh variable tersebut.

Berbeda dengan VB 6, variable dalam VB.NET harus dideklarasikan terlebih dahulu jika tidak akan terjadi kesalahan kompilasi, hal ini dikarenakan secara default setting eksplisit pada VB.NET adalah Option Explicit ON.

Dim merupakan keyword dalam pendeklarasian variable.

```
Dim (namavariabel) as (tipe data)
```

Atau pendeklarasian variable disertai pemberian nilai awal :

```
Dim (namavariabel) as (tipe data) = nilai awal
```

Berikut contoh pendeklarasian variable

```
Sub main ()
```

```
    Dim a,c as integer ' pendeklarasian variable tanpa nilai awal
```

```
    Dim b as integer = 10 ' pendeklarasian variable dengan nilai awal
```

```
a = 10 'pemberian nilai variable
```

```
c = a + b
```

```
End Sub
```

Jika kita tidak memberikan nilai awal pada sebuah variable maka nilai berikut akan diberikan sebagai nilai awal variable tersebut :

0 untuk tipe data numerik termasuk byte

Binary 0 untuk tipe data char

Nothing untuk tipe data reference (String, objek dan array)

False untuk boolean

12:00 AM pada januari 1 untuk date

Hal yang harus anda perhatikan dalam pembuatan variable adalah daya tampung value yang dimiliki, karena jika anda memasukkan nilai melebihi daya tampung yang dimiliki variable tersebut maka overflow exception akan terjadi.

1.3. Array

Sebuah variable hanya dapat menyimpan sebuah nilai dari data, jika kita ingin menyimpan beberapa data yang memiliki tipe data yang sama dengan menggunakan variable yang berbeda - beda akan memakan banyak memory dan merepotkan. Bagaimana cara mengatasi hal tersebut????

Array merupakan salah satu solusi untuk masalah diatas, mengapa demikian ????

Array dapat diartikan sebagai sekumpulan data yang menggunakan nama variable yang sama dengan tipe data yang sama.

Tiap - tiap nilai atau data yang tersimpan dalam array disebut sebagai elemen dan memiliki index (atau urutan), dimana index tersebut akan dimulai dari 0.

masih bingung dengan array???? Atau kita analogikan seperti ini : misalkan kita ingin mendata jenis anak anjing yang baru lahir berjumlah 6 ekor . Asumsi untuk tipe datanya adalah anak anjing.

Jika menggunakan variable, maka kita akan mendeklarasikan 6 buah variable dengan tipe anak anjing,

Dim anjing1 as anak anjing = pudel

Dim anjing2 as anak anjing = jerman... dst sampai dengan 6 variable.

Jika menggunakan sebuah array maka kita hanya mendeklarasikan sekali yaitu sebuah array dengan daya tampung atau panjang array sesuai dengan yang kita inginkan, dalam hal ini panjang dari array adalah 6.

Dim anjing (5) as anak anjing ---- > (karena dimulai dari 0)

Anjing(0)= pudel

Anjing(1) = doberman

:

Anjing (5) = bulldog

Jika kita ingin memasukkan tipe data yang tidak sesuai akan error

Anjing(2) = persia (asumsi persia adalah jenis kucing)

Bagaimana sudah lebih mengerti??? Atau tambah bingung ????

Menciptakan Array

Untuk menciptakan variabel array dapat menggunakan perintah :

```
DIM varabel[(panjang array)] [AS tipe]
```

Contoh :

DIM A(6) As Integer

A(0)=0	A(1)=0	A(2)=0	A(3)=0	A(4)=0	A(5)=0	A(6)=0
--------	--------	--------	--------	--------	--------	--------

Akan menciptakan variabel array yang bernama A yang memiliki 7 elemen (mulai dari index 0 s/d 6), yang masing-masing dapat menampung data integer, dan memiliki nilai awal 0 (nol), atau dengan kata lain secara default setiap nilai untuk masing-masing index diberi nilai 0 (nol).

Kita juga bisa membuat suatu array dengan langsung memberikan nilainya, dengan cara seperti contoh berikut.

```
DIM A() As Integer = {1, 2, 3, 4, 5, 6, 7}
```

A(0)=1	A(1)=2	A(2)=3	A(3)=4	A(4)=5	A(5)=6	A(6)=7
--------	--------	--------	--------	--------	--------	--------

Variable array A akan memiliki indek maksimum 6 mulai dari 0, dengan jumlah elemen 7.

Reinitializing Arrays

Kita juga bisa melakukan inisialisasi ulang terhadap suatu varabel array, misalnya saja awalnya kita memiliki sebuah array dengan panjang 7, kemudian saat program berlansung, kita akan memasukkan data baru dalam array tersebut sehingga panjang array yang kita miliki harus dirubah, musalnya 2 maka untuk hal tersebut kita harus meng inisialisasi ulang array yang kita milik,. Yaitu dengan menggunakan kata kunci **ReDim**. Namum penggunaan redim ini akan menghapus seluruh nilai yang telah ada sebelumnya, nah agar nilai dalam array tidak hilang kita perlu menambahkan keyword lainnya yaitu **Preserve**.

Berikut adalah inisialisai ulang sebuah array dimana nantinya Semua nilai yang sudah diberikan kepada array yang lama akan dikembalikan ke nilai default, yaitu 0.

Contoh :

```
Dim A(4) as integer.
```

A(0)=1	A(1)=2	A(2)=3	A(3)=4	A(4)=5
--------	--------	--------	--------	--------

```
ReDim A(5)
```

A(0)=0	A(1)=0	A(2)=0	A(3)=0	A(4)=0	A(5)=0
--------	--------	--------	--------	--------	--------

Nah, Jika kita ingin membuat nilai yang telah ada dalam array sebelumnya tidak hilang, kita bisa menggunakan kata kunci "Preserve" , contoh :

```
ReDim Preserve A(7)
```

A(0)=1	A(1)=2	A(2)=3	A(3)=4	A(4)=5	A(5)=6	A(6)=0	A(7)=0
--------	--------	--------	--------	--------	--------	--------	--------

Maka nilai dari array mulai index ke 0 sampai 5 tidak akan hilang nilainya, sehingga jika kita ingin mengambil nilai dari index 0 sampai 5 maka nilai dari array sebelum di redim preserve akan dipanggil, sedangkan nilai dari index setelahnya bernilai default. lihat pada gambar diatas.

Menggunakan Variabel Array

Penggunaan Variabel Array sama saja dengan variabel biasa, baik untuk **memberi nilai** maupun **mengambil nilai** dalam array, hanya saja bedanya dengan variable, untuk array kita perlu menyebutkan nomor index elemen yang diinginkan.

Contoh :

A(0) = 100

A(1) = 200

Textbox1.text = A(2) -> me

A(0)=100	A(1)=200	A(2)=300	A(3)=0	A(4)=0	A(5)=0
----------	----------	----------	--------	--------	--------

Akan mengisi nilai 100 ke elemen pertama, 200 ke elemen kedua, dan 300 ke elemen ketiga. Catatan : Elemen pertama dari array dimulai dengan nomor index 0.

Static Array dan Dynamic Array

Pada Visual Basic, semua array yang dideklarasikan dengan numerik adalah Static Array, dan array yang dideklarasikan dengan variabel adalah Dynamic Array.

Contoh :

DIM D(30) ; D adalah variabel Static Array,

E = 30

DIM F(E); **F adalah variabel Dynamic Array**, karena jumlah elemennya ditentukan dengan menggunakan variabel E, dalam hal ini 30.

Variabel Array Multi Dimensi

Anda dapat membentuk variabel array 2 atau 3 dimensi dengan perintah dim.

Contoh :

```
DIM A(10,10)
```

Akan membentuk array 11 baris (0 s/d 10), 11 kolom (0 s/d 10) (121 elemen)

Cara pemakaian array Multi Dimensi sama saja dengan array 1 dimensi yaitu perlu disebutkan nomor indexnya masing-masing.

Contoh :

```
A(2,2) = 5
```

Akan mengisi nilai 5 ke elemen baris 3 kolom 3 dari Array dua dimensi A. (ingat index dimulai dari 0, kecuali dengan pernyataan OPTION BASE 1)

Multidimensional Arrays

Semua array yang dijelaskan diatas merupakan array linear. Ada dua macam array multidimensional, yaitu : Rectangular arrays dan Jagged arrays.

Rectangular arrays

Merupakan sebuah array yang setiap anggota dimensinya akan terdiri dari beberapa dimensi lain, dengan ukuran yang sama.

Contoh pendeklarasian :

```
Dim A (4, 2) As Integer
```

Pendeklarasian array dengan jumlah 5 dengan masing-masing anggota berjumlah 3, jadi jumlah keseluruhan menjadi 15.

A (0,0)=0	A(0,1)=0	A(0,2)=0
A(1,0)=0	A(1,1)=0	A(1,2)=0
A(2,0)=0	A(2,1)=0	A(2,2)=0

A(3,0)=0	A(3,1)=0	A(3,2)=0
A(4,0)=0	A(4,1)=0	A(4,2)=0

```
Dim A(, ) As Integer = {{1, 2, 3}, {12, 13, 14}, {11, 10, 9}}
```

Pendeklarasian dengan sekaligus inisialisasinya.

A (0,0)=1	A(0,1)=2	A(0,2)=3
A(1,0)=12	A(1,1)=13	A(1,2)=14
A(2,0)=11	A(2,1)=10	A(2,2)=9

Jagged Arrays

Merupakan array multidimensional, dimana array tersebut bersifat array off array, artinya array yang berasal dari array lainnya, dimana panjangnya bisa berbeda.

Perhatikan contoh berikut ini :

```
Dim colors(2)() as String
colors(0)=NewString() {"Red", "blue", "Green"}
colors(1)=NewString() {"Yellow", "Purple", "Green", "Violet"}
colors(2)=NewString() {"Red", "Black", "White", "Grey", "Aqua"}
```

"Red"	"Blue"	"Green"		
"Yellow"	"Purple"	"Green"	"Violet"	
"Red"	"Blak"	"White"	"Grey"	"Aqua"

Pada contoh diatas kita akan mendeklarasikan sebuah array dengan nama colors yang panjangnya 3. untuk index yang pertama yaitu index ke 0 panjangnya 3, index nomor 1 panjangnya 4, index nomor 2 panjangnya 5.

1.4. Acces Modifier

Sebuah Acces modifier mendefinisikan dimanakah sebuah method, properties, variable, class dalam sebuah kode program dapat digunakan. Acces Modifier yang tersedia dalam Visual Basic.Net adalah :

Public: memberikan acces public pada sebuah variable berarti tidak ada batasan apapun dalam pengaksesan variable.

Private: memberikan acces private pada sebuah variable berarti variable tersebut hanya dapat diakses pada method atau class tempat variable tersebut dideklarasikan.

Protected: memberikan acces protected pada sebuah variable berarti variable tersebut dapat diakses oleh class dimana dia di deklarasikan dan juga class lain yang merupakan turunan dari class tersebut.

Friend: memberikan acces friend pada variable berarti variable tersebut dapat diakses oleh class lain yang berada pada satu project dengan class dimana variable tersebut dideklarasikan.

Protected Friend: memberikan acces pada variable gabungan antara protected dan friend.

Static: mendeklarasikan variable static berarti variable tersebut akan memiliki nilai sampai prosedur atau tempat dia dideklarasikan berakhir.

Shared: Variable dapat di gunakan oleh kode program yang lain

ReadOnly: Membuat variable hanya dapat dibaca tanpa dapat dilakukan pemberian nilai baru.

1.5. Konversi Tipe Data

Didalam Visual Basic .Net kita bisa melakukan konversi terhadap suatu tipe data dengan dua cara yaitu, secara implisit dan secara eksplisit.

Secara Implisit

Cara ini dilakukan dengan memberikan nilai kepada suatu variabel yang bertipe data tertentu dengan nilai dari variabel lain dengan tipe data yang berbeda.

Untuk lebih jelasnya perhatikan contoh berikut ini :


```
Sub Main()  
Dim d=132.31223 as Double  
Dim i as Integer  
i=d  
MsgBox("Integer value is" & i)  
End Sub
```

Nilai i pada message box adalah 132.

Secara Eksplisit

Jika kita tidak bisa melakukan konversi data dengan cara implisit, kita harus melakukannya dengan cara eksplisit. Cara ini disebut juga dengan cara Cast (Casting). Function yang digunakan adalah function CType.

Ctype

Terdiri dari dua argumen, yaitu :

- Argumen pertama merupakan data yang akan kita konversikan
- Argumen kedua merupakan tipe data yang akan digunakan

Perhatikan contoh berikut ini :

```
Private Sub Button1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs)  
Handles Button1.Click  
Dim d As Double  
d = 132.31223  
Dim i As Integer  
i = CType(d, Integer)  
MsgBox("Integer value is" & i)  
End Sub
```

Selain dengan menggunakan function CType kita juga bisa menggunakan function seperti berikut ini :

- CBool - Digunakan untuk mengkonversikan data ke tipe data Boolean
- CByte - Digunakan untuk mengkonversikan data ke tipe data Byte
- CChar - Digunakan untuk mengkonversikan data ke tipe data Char
- CDate - Digunakan untuk mengkonversikan data ke tipe data Date
- CDbl - Digunakan untuk mengkonversikan data ke tipe data Double

CDec - Digunakan untuk mengkonversikan data ke tipe data Decimal

CInt - Digunakan untuk mengkonversikan data ke tipe data Integer

CLng - Digunakan untuk mengkonversikan data ke tipe data Long

CObj - Digunakan untuk mengkonversikan data ke tipe data Object

Cshort - Digunakan untuk mengkonversikan data ke tipe data Short

CSng - Digunakan untuk mengkonversikan data ke tipe data Single

Cstring - Digunakan untuk mengkonversikan data ke tipe data String

Syntax penulisan berbeda dengan CType untuk casting dengan menggunakan function diatas hanya menggunakan satu parameter saja yaitu nilai yang akan kita konversi.

Untuk contoh diatas kita bisa tuliskan dengan cara :

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim d As Double
    d = 132.31223

    Dim i As Integer
    i = CInt(d)
    MsgBox("Integer value is" & i)
End Sub
```

Nilai i yang akan muncul pada message box adalah 132.

Latihan

1. Buatlah sebuah windows application baru.
2. Buatlah design form seperti berikut:

The image shows a Windows application window titled "Form1" with a subtitle "Data Penilaian Mahasiswa PV1". The window contains the following elements:

- A "ListBox1" at the top.
- A "Data ke-" label followed by a text input field and a "Tampilkan Data" button.
- Four labels with corresponding text input fields: "Nama", "Nilai UTS", "Nilai UAS", and "Nilai Quiz dan Tugas".
- A large button labeled "Hitung dan Tampilkan Nilai Akhir" below the input fields.
- A "ListBox2" at the bottom.

3. Alur program yang akan kita buat adalah pertama kali dijalankan, pada listbox1 akan tercetak isi dari sebuah array 2 dimensi.

Jika kita menginputkan urutan array ke berapa pada textbox data ke , kemudian kita tekan tombol tampilkan data , maka data yang bersesuaian akan tercetak pada masing - masing textbox yang bersesuaian.

Kemudian jika kita tekan tombol hitung dan tampilkan nilai akhir maka akan menghitung nilai akhir kemudian

memasukkan nilainya pada array nakhir untuk kemudian ditampilkan di listbox 2. (untuk mengerjakan ikuti urutan no berikutnya sebagai bimbingan).

- 4. Definisikan variable untuk menyimpan nama dengan type string, variable nilai uts, nilai uas, nilai quis dan tugas dengan type integer, dan nilai akhir dengan type double.

Definisikan sebuah array dua dimensi dengan banyak baris 3 dan banyak kolom 4 beri nama asiswa, dan type data dari array adalah string. (Array ini merupakan array data siswa).

Definisikan array satu dimensi untuk menyimpan nilai akhir dengan nama aAkhir dengan panjang 0, type dari array adalah double.

Definisikan juga sebuah array satu dimensi untuk menyimpan nama siswa yang akan kita kombinasikan dengan array aAkhir dalam penggunaannya. Beri nama array ini aNama dengan type String dan panjang 0.

Maksudnya, jika kita memasukkan nilai pada index ke- n pada aNama, kita juga akan memasukkan nilai perhitungan akhir ke dalam array aAkhir sesuai dengan index ke n juga.

```
Public Class Form1
    Dim nama As String
    Dim uts, uas, tugas As Integer
    Dim akhir As Double
    Dim asiswa(2, 3) As String
    Dim aAkhir(0) As Double
    Dim aNama(0) As String

```

- 5. Pada event form load isikan data siswa pada array aSiswa.Data tersebut sebagai berikut :

Nama	UTS	UAS	Quis dan Tugas
Cahyo	75	67	80
Adi	55	95	70
Saori	40	90	77

Masih pada form load event, masukkan data array aSiswa tersebut dalam listbox1. (Gunakan method items.add pada listbox1).

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    aSiswa(0, 0) = "Cahyo"
    aSiswa(0, 1) = "75"
    aSiswa(0, 2) = "67"
    aSiswa(0, 3) = "80"

    aSiswa(1, 0) = "Adi"
    aSiswa(1, 1) = "55"
    aSiswa(1, 2) = "95"
    aSiswa(1, 3) = "70"

    aSiswa(2, 0) = "Saori"
    aSiswa(2, 1) = "60"
    aSiswa(2, 2) = "90"
    aSiswa(2, 3) = "77"

    ListBox1.Items.Add(aSiswa(0, 0) & vbTab & aSiswa(0, 1) & vbTab & aSiswa(0, 2) & vbTab & aSiswa(0, 3))
    ListBox1.Items.Add(aSiswa(1, 0) & vbTab & aSiswa(1, 1) & vbTab & aSiswa(1, 2) & vbTab & aSiswa(1, 3))
    ListBox1.Items.Add(aSiswa(2, 0) & vbTab & aSiswa(2, 1) & vbTab & aSiswa(2, 2) & vbTab & aSiswa(2, 3))
End Sub
```

- 6 Pada event click di button tampilkan data, panggilah data pada array aSiswa dengan index baris sesuai dengan inputan pada textbox Data ke dikurangi 1(karena index mulai 0) kedalam masing - masing variable yang bersesuaian yang telah anda buat. Kemudian set text dari masing-masing textbox sesuai dengan nilai pada variable yang bersesuaian(misal textbox nama di isi dengan nilai dari variable nama.

```
Private Sub btn_Tampilkan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_
    nama = aSiswa(TextBox1.Text - 1, 0)
    uts = aSiswa(TextBox1.Text - 1, 1)
    uas = aSiswa(TextBox1.Text - 1, 2)
    tugas = aSiswa(TextBox1.Text - 1, 3)

    TextBox2.Text = nama
    TextBox3.Text = uts
    TextBox4.Text = uas
    TextBox5.Text = tugas
End Sub
```

- 7 Pada event click pada button hitung dan tampilkan nilai akhir. Lakukan perhitungan nilai akhir dengan rumus 30% uts, 40 % uas, 30% tugas dan quis. Kemudian hasilnya masukkan dalam variable akhir.

Lakukan inisialisasi ulang dengan nilai array yang tidak hilang setelahnya (lihat keyword pada pendahuluan) pada array aNama dengan panjang sesuai dengan panjang array saat ini (length array) dan isilah aNama pada baris ke panjang array dikurangi 1 dengan nilai dari variable nama.

Lakukan hal sama untuk array aAkhir.

Isikan data tersebut pada listbox dua dengan memanggil aNama dan aAkhir pada baris ke panjang array -1

Untuk memberi jarak tab pada listbox gunakan vbTab dan untuk menggabungkan string gunakan &

```

Private Sub btn_Hitung_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_
    akhir = (Cdbl(uts * 0.3) + Cdbl(uas * 0.4) + Cdbl(tugas * 0.3))

    ReDim Preserve aNama(aNama.Length)
    aNama(aNama.Length - 1) = nama

    ReDim Preserve aakhir(aakhir.Length)
    aakhir(aakhir.Length - 1) = akhir

    ListBox2.Items.Add(aNama(aNama.Length - 1) & vbTab & aakhir(aakhir.Length - 1))

End Sub

```

- 8 Output pada program yang akan anda buat akan tampak seperti

The screenshot shows a Windows application window titled "Form1" with a title bar containing standard Windows window controls. The main content area has a title "Data Penilaian Mahasiswa PV1" and contains the following elements:

- A table displaying student data:

Cahyo	75	67	80
Adi	55	95	70
Saori	60	90	77
- A "Data ke-" label followed by a text box containing the number "1".
- A "Tampilkan Data" button.
- Input fields for student details:
 - Nama: Cahyo
 - Nilai UTS: 75
 - Nilai UAS: 67
 - Nilai Quiz dan Tugas: 80
- A "Hitung dan Tampilkan Nilai Akhir" button.
- A text box at the bottom displaying the result: "Cahyo 73.3".

MODUL 2

LOGIKA PEMROGRAMAN 1

practices make perfect
--Anonymous--



Tujuan
Mahasiswa memahami tentang logika dalam pemrograman visual
basic .Net
Mahasiswa mengerti tentang pemakaian selection dan looping
dalam pemrograman

Materi
Selection (If...then....Else...End If, Select case.....End Select)
Nested Selection
Looping (For...Next, Do While....Loop)
Nested Loop

Referensi
<http://www.startvbdotnet.com/language/loops.aspx>
<http://indoprog.tripod.com/>
Deitel, 2002, *Visual Basic .Net How To Program* , Prentice Hall, New
Jersey.

2.1. Selection

If....Then....Else Statement

Merupakan ekspresi berkondisi pada blok pernyataan khusus, fungsinya mengendalikan apakah pernyataan lain pada program dijalankan dan bagaimana urutan dijalankan. Dalam bentuk yang paling sederhana, struktur keputusan If... Then ditulis dalam satu baris tunggal tanpa menggunakan End If:

```
If condition Then Statement
```

Dimana *condition* adalah ekspresi berkondisi dan *statement* adalah pernyataan program Visual Basic yang valid.

Sebagai contoh :

```
If jumlah >= 20 Then Label1.Caption = "Anda  
Menang!"
```

Maksudnya adalah sebuah struktur keputusan If... Then... yang menggunakan ekspresi berkondisi jumlah ≥ 20 .

Visual Basic juga mendukung struktur keputusan If... Then dimana bisa menyertakan beberapa ekspresi berkondisi. Blok pernyataan ini panjangnya bisa beberapa baris dan mengandung kata-kata kunci yang penting seperti ElseIf, Else, dan End If.

Sintax penulisannya :

```
If ekspresi logika Then  
    Statement  
    ...  
End If
```

Berfungsi menentukan apakah kondisi (ekspresi logika) memenuhi syarat untuk melaksanakan statement-statement yang di samping IF...Then ataupun berada dalam Blok IF... Then...End If.

Pilih Antara Dua

```
IF ekspresi logika Then  
    Statement1  
    ...
```

```
Else
  Statement2
  ...
End If
```

Berfungsi menentukan Blok statement mana yang akan dilaksanakan komputer, jika ekspresi logika menghasilkan nilai Benar maka Statement1 akan dilaksanakan dan sebaliknya jika nilai yang dihasilkan salah maka Statement2 akan dijalankan.

Pilih Antara Banyak

```
IF ekspresi logika 1 Then
  Statement1
  ...
ElseIF ekspresi logika 2 Then
  Statement2
  ...
Else
  Statement2
  ...
Endif
```

Berfungsi menentukan Blok statement yang mana akan dijalankan oleh komputer sesuai dengan ekspresi logika yang menghasilkan nilai benar.

Contoh :

```
Sub Main()
  Dim i As Integer
  i= Val(textbox1.text)
  if i= 1 Then
    msgbox("One")
  ElseIf i = 2 Then
    msgbox("Two")
  ElseIf i = 3 Then
    msgbox("Three")
  Else
    msgbox("Number not 1,2,3")
  End If
End Sub
```

Pilihan dalam pilihan

```
IF ekspresi logika 1 Then
  IF ekspresi logika 2 Then
    Statement2
    .....
  End If
Else
  IF ekspresi logika 3 Then
    Statement3
    .....
  End If
Endif
```

Berguna untuk pengecekan kondisi lain jika suatu kondisi sudah terpenuhi. Dengan syntax diatas setelah ekspresi logika pertama selesai dijalankan dan menghasilkan true maka program akan mengeksekusi ekspresi logika 2 namun jika menghasilkan false maka ekspresi logika 3 yang akan dijalankan.

Select Case

Selain menggunakan perintah IF, untuk pilihan Antara Banyak dapat menggunakan perintah Select...Case.

Jika dalam perintah *IF* pengecekan dilakukan berdasarkan kondisi, namun dalam *select case* pengecekan dilakukan berdasarkan nilai ekspresi.

Syntax-nya :

```
Select Case Variabel
  Case ekspresi 1
    Statement1
    ...
  Case ekspresi 2
    Statement2
    ...
  Case Else
    ...
End Select
```

Contoh :

```
Imports System.Console
Module Module1
  Sub Main()
```

```

Dim keyIn As Integer
WriteLine("Enter a number between 1 and 4")
keyIn = Val(ReadLine())
Select Case keyIn
    Case 1
        WriteLine("You entered 1")
    Case 2
        WriteLine("You entered 2")
    Case 3
        WriteLine("You entered 3")
    Case 4
        WriteLine("You entered 4")
End Select
End Sub
End Module

```

Select case dengan Case Is...

Kita juga bisa menggunakan perbandingan dalam mengeksekusi syarat dalam case, seperti pada contoh berikut ini :

```

Select Case AdjustedIncome
    Case Is <= 27050
        ' 15% tax bracket
        TaxDue = AdjustedIncome * 0.15
    Case Is <= 65550
        ' 28% tax bracket TaxDue = 4057.5 +
        ((AdjustedIncome - 27050) *0.28)
    Case Is <= 136750
        ' 31% tax bracket TaxDue = 4057.5 +
        ((AdjustedIncome - 65550) *0.31)
    Case Is <= 297350
        ' 36% tax bracket TaxDue = 4057.5 +
        ((AdjustedIncome-136750)*0.36)
    Case Else
        ' 39.6% tax bracket TaxDue
        =4057.5+((AdjustedIncome-297350)*0.396)
End Select

```

Dari sintax diatas, jika nilai adjustedIncome kurang dari 136750 maka 31% akan dilakukan.

Select case dengan ..To...

Untuk syarat yang berupa *range* angka kita bisa menggunakan kata kunci "To"

```
Select Case AdjustedIncome
  Case 1000 To 27050
    ' 15% tax bracket
    TaxDue = AdjustedIncome * 0.15
  Case 27051 To 65550
    ' 28% tax bracket TaxDue = 4057.5 +
((AdjustedIncome - 27050) *0.28)
  Case 65551 To 136750
    ' 31% tax bracket TaxDue = 4057.5 +
((AdjustedIncome - 65550) *0.31)
  Case 136751 To 297350
    ' 36% tax bracket TaxDue = 4057.5 +
((AdjustedIncome - 136750)*0.36)
  Case Else
    ' 39.6% tax bracket TaxDue = 4057.5
+((AdjustedIncome-297350)*0.396)
End Select
```

2.2. Looping

Untuk melakukan perulangan Anda dapat menggunakan beberapa sintax yang sesuai dengan kebutuhan Anda, yaitu Do...Loops, For...Ne xt, While...end While, For...Each.

Do... loops, While...End While bekerja dengan baik, ketika anda tidak tahu berapa kali program akan menjalankan iterasi.

Ketika anda telah mengetahui harus menjalankan statement sejumlah n kali, For...Next adalah pilihan yang lebih baik.

Do... Loop

Gunakan Doloop untuk mengeksekusi suatu blok statement dengan jumlah perulangan yang tak terhingga dan anda tidak mengetahui dengan pasti jumlah iterasi yang digunakan, ada beberapa Do...Loop statement, tetapi masing-masing mengevaluasi suatu kondisi untuk menentukan apakah melanjutkan eksekusi atau tidak, yaitu :

Pemeriksaan diawal

```
DO [{WHILE | UNTIL} ekspresi logika]
  Statement
  ...
LOOP
```

Pemeriksaan diakhir

```
DO
    [statementblock]
    .....
LOOP [{WHILE | UNTIL} ekspresi logika]
```

Perbedaan dari kedua syntax diatas adalah pada pemeriksaan struktur kondisi, pada syntax pertama pemeriksaan kondisi dilakukan di awal sebelum statement dijalankan. Sedangkan pada syntax kedua pemeriksaan dilakukan di akhir setelah statement dijalankan.

Dalam syntax tersebut, **statements akan dijalankan selama kondisi benar jika kondisi salah maka akan keluar dari looping.**

```
Do While condition
    statements
Loop
```

Contoh

```
Dim i as integer = 0
Do While i < 5
    i = i + 1
    msgbox(i)
Loop
```

Dalam contoh diatas, selama i masih kurang dari 5 maka looping akan dilakukan.

Looping Counter

Digunakan untuk proses berulang yang jumlah perulangannya ditentukan berdasarkan perhitungan (sistem counter).

```
FOR counter = mulai TO selesai [STEP selisih]
    Statement
    ...
NEXT [counter]
```

Catatan :

Jika Step tidak ditulis maka secara default step akan bernilai 1.

Step tidak harus bilangan bulat tetapi dapat juga floating point (contoh : 0.1)

Contoh

```
For i = 0 To 5
  ..
Next
```

Akan mencetak 0 , 1 , 2 , 3 , 4.

```
For i = 1 To 10 step 0.1
  ..
Next
```

Akan mencetak 1 , 1.1 , 1.2 , 1.3 , ..., 10.

```
For i = 10 To 1 step -1
  ..
Next
```

Akan mencetak 10 , 9 , 8 , ..., 1.

WhileLoop

While....loop statement tidak jauh berbeda dengan Do...Loop statement, yaitu untuk perulangan dimana kita tidak tahu berapa jumlah counter yang akan dilakukan, dan statement akan dijalankan jika kondisi yang akan dicek bernilai benar jika tidak maka keluar dari looping.

Sintax :

```
While condition
  Statements
End While
```

For Each..... next

Digunakan untuk melakukan iterasi perulangan pada collection tertentu, misalnya jika kita ingin mencetak seluruh isi dari sebuah array atau jika kita ingin menghapus seluruh text pada textbox yang ada dalam form kita.

Sintaxnya :


```
For Each element [As data type] In group
    Statement
Next
```

Contoh :

```
Dim control As Control
For Each control In Me.Controls
    If TypeOf (control) Is TextBox Then
        control.Text = ""
    End If
Next
```

Sintax diatas digunakan jika kita ingin menghapus text pada textbox - textbox yang ada pada form kita.

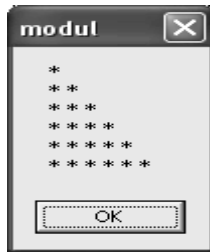
Struktur Kendali Nested

Anda dapat meletakkan suatu struktur kendali didalam struktur kendali lainnya, yang dikenal dengan istilah nested. Struktur kendali pada Visual Basic dapat nested sebanyak level yang anda inginkan. Atau istilah mudahnya looping dalam looping sebanyak yang anda inginkan.

Contoh :

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim i, j As Integer
    Dim output As String
    For i = 0 To 5
        For j = 0 To i
            output = output & " " & "*"
        Next
        output = output & vbCrLf
    Next
    MsgBox(output)
End Sub
```

Kode program diatas merupakan contoh nested loop dimana output yang kita inginkan seperti pada gambar berikut :



Meninggalkan suatu Struktur Kendali

Statement Exit memperbolehkan anda untuk keluar langsung dari suatu For loop maupun Do Loop, syntax untuk statement Exit adalah sederhana : Exit For dapat muncul sebanyak mungkin dalam suatu For Loop, demikian juga Exit Do untuk Do loop.

```
For counter = start To end [Step increment]
    [statementblock]
    [Exit For]
    [statementblock]
Next [counter[, counter] [, ...]]
Do [{While | Until} condition]
    [statementblock]
    [Exit Do]
    [statementblock]
Loop
```

Statement Exit Do bekerja pada semua versi syntax Do loop.

Exit For dan Exit Do dilakukan jika kita perlu untuk keluar dari suatu loop dengan segera, tanpa melakukan iterasi yang berikutnya. jadi jika terdapat perintah exit For maka compiler akan menjalankan kode program dibawah next,

jika menjalankan perintah Exit Do maka compiler akan menjalankan kode program dibawah statement Loop.

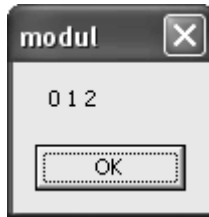
Berikut adalah contoh penggunaan exit for, misalkan kita akan melakukan perulangan mulai dari 0 sampai 10 untuk kemudian kita tampilkan melalui messagebox, namun jika kita telah melakukan perulangan sebanyak 3 maka kita akan keluar dari for.

Syntax :

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

```
Dim i As Integer
Dim output As String
For i = 0 To 10
    If i = 3 Then
        Exit For
    End If
    output = output & " " & i
Next
MsgBox(output)
End Sub
```

Output dari kode program diatas :



Pada saat $i = 3$ maka exit for akan dijalankan sehingga compiler akan melanjutkan kode program setelah perintah next sehingga hanya 0,1,2 yang tercetak tapi 3 tidak, karena saat 3 kode program $output = output \& \text{" " \& } i$ tidak dijalankan.

Latihan

1. Buatlah sebuah windows application baru. Buatlah design form seperti berikut :



2. Untuk mengisi tanggal Lahir pada combo box, gunakan perulangan for dengan melakukan iterasi sebanyak 31 dimulai dari 1, kemudian masukkan dalam item combo box dengan menggunakan method items add dengan parameter nilai dari iterasi tersebut.

Untuk mengisi Bulan lahir lakukan hal yang sama dengan pengisian tanggal lahir dengan iterasi sebanyak 12 mulai dari 1, masukkan dalam combo box dengan nilai string bulan yang didapatkan dengan menggunakan method monthname(iterasi perulangan)

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    For i As Integer = 1 To 31
        ComboBox1.Items.Add(i)
    Next

    For i As Integer = 1 To 12
        ComboBox2.Items.Add(MonthName(i))
    Next
End Sub
```

3. Pada saat tombol cari ditekan, cek nilai yang ada dalam combo box bulan dan tanggal untuk menemukan zodiak apa yang dimiliki kemudian akan tampil pada textbox .

Berikut adalah nilai dari data yang akan dicocokkan dengan inputan.

Jika tanggal lahir antara Des 22 – Jan 19 maka zodiak Capricorn

Jika tanggal lahir antara Jan 20 – Feb 17 maka zodiak Aquarius

Jika tanggal lahir antara Feb 18 – Mar 19 maka zodiak Pisces

Jika tanggal lahir antara Mar 20 – Apr 19 maka zodiak Aries

Jika tanggal lahir antara Apr 20 – Mei 19 maka zodiak Taurus

Jika tanggal lahir antara Mei 20 – Jun 20 maka zodiak Gemini

Jika tanggal lahir antara Juni 21 – Juli 21 maka zodiak Cancer

Jika tanggal lahir antara Juli 22 – Aug 22 maka zodiak Leo

Jika tanggal lahir antara Aug 23 – Sep 21 maka zodiak Virgo

Jika tanggal lahir antara Sep 22 – Okt 22 maka zodiak Libra

Jika tanggal lahir antara Okt 23 – Nop 21 maka zodiak Scorpio

Jika tanggal lahir antara Nop 22 – Des 21 maka zodiak Sagitarius

Kemudian setelah dilakukan pengecekan terhadap zodiak, textbox di bawahnya akan menampilkan pesan seperti berikut: (isi dari textbox nama) , anda lahir pada (isi dari combo box tanggal) (bulan berdasar isi combo box bulan) zodiak anda adalah (zodiak berdasar nilai yang sesuai).

```
Dim zodiak As String

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    If ComboBox1.SelectedIndex < 20 And ComboBox2.SelectedIndex = 0 Or ComboBox1.SelectedIndex > 21 And ComboBox2.SelectedIndex = 11 Then
        zodiak = "Capricorn"
    ElseIf ComboBox1.SelectedIndex < 18 And ComboBox2.SelectedIndex = 1 Or ComboBox1.SelectedIndex > 19 And ComboBox2.SelectedIndex = 0 Then
        zodiak = "Aquarius"
    ElseIf ComboBox1.SelectedIndex < 20 And ComboBox2.SelectedIndex = 2 Or ComboBox1.SelectedIndex > 17 And ComboBox2.SelectedIndex = 1 Then
        zodiak = "Pisces"
    ElseIf ComboBox1.SelectedIndex < 20 And ComboBox2.SelectedIndex = 3 Or ComboBox1.SelectedIndex > 19 And ComboBox2.SelectedIndex = 2 Then
        zodiak = "Aries"
    ElseIf ComboBox1.SelectedIndex < 20 And ComboBox2.SelectedIndex = 4 Or ComboBox1.SelectedIndex > 19 And ComboBox2.SelectedIndex = 3 Then
        zodiak = "Taurus"
    ElseIf ComboBox1.SelectedIndex < 21 And ComboBox2.SelectedIndex = 5 Or ComboBox1.SelectedIndex > 19 And ComboBox2.SelectedIndex = 4 Then
        zodiak = "Gemini"
    ElseIf ComboBox1.SelectedIndex < 22 And ComboBox2.SelectedIndex = 6 Or ComboBox1.SelectedIndex > 20 And ComboBox2.SelectedIndex = 5 Then
        zodiak = "Cancer"
    ElseIf ComboBox1.SelectedIndex < 23 And ComboBox2.SelectedIndex = 7 Or ComboBox1.SelectedIndex > 21 And ComboBox2.SelectedIndex = 6 Then
        zodiak = "Leo"
    ElseIf ComboBox1.SelectedIndex < 22 And ComboBox2.SelectedIndex = 8 Or ComboBox1.SelectedIndex > 22 And ComboBox2.SelectedIndex = 7 Then
        zodiak = "Virgo"
    ElseIf ComboBox1.SelectedIndex < 23 And ComboBox2.SelectedIndex = 9 Or ComboBox1.SelectedIndex > 21 And ComboBox2.SelectedIndex = 8 Then
        zodiak = "Libra"
    ElseIf ComboBox1.SelectedIndex < 22 And ComboBox2.SelectedIndex = 10 Or ComboBox1.SelectedIndex > 22 And ComboBox2.SelectedIndex = 9 Then
        zodiak = "Scorpio"
    ElseIf ComboBox1.SelectedIndex < 22 And ComboBox2.SelectedIndex = 11 Or ComboBox1.SelectedIndex > 20 And ComboBox2.SelectedIndex = 10 Then
        zodiak = "Sagittarius"
    End If

    TextBox2.Text = TextBox1.Text & ", anda lahir pada " & ComboBox1.Text & " " & ComboBox2.Text & " zodiak anda adalah " & zodiak
End Sub
```

- 4 Tombol bersih untuk membersihkan seluruh kontrol di form (gunakan for each looping untuk hal ini).

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    For Each a As Control In Me.Controls
        If TypeOf (a) Is TextBox Or TypeOf (a) Is ComboBox Then
            a.Text = ""
        End If
    Next
End Sub
```

- 5 Tombol keluar untuk keluar dari aplikasi.(gunakan method exit dari application)

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Application.Exit()
End Sub
```

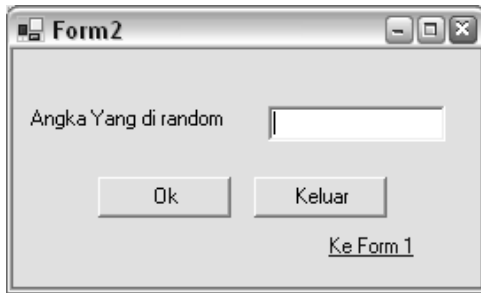
- 6 Link Label ke form 2 untuk menghubungkan form 1 dan form dua yang nanti akan kita buat. (Buatlah sebuah object dengan type form2 kemudian hide form me dan show object dari form 2).

```
Private Sub f2_Link(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
    Dim form2 As New Form2
    Me.Hide()
    form2.Show()
End Sub
```

- 7 Contoh output yang akan ditampilkan :

The screenshot shows a window titled "Form1" with the main heading "Zodiak Anda". It features three input fields: "Nama" with the value "saori", "Tanggal Lahir" with the value "19", and "Bulan Lahir" with the value "December". Below these fields are two buttons: "Cari" (highlighted with a dashed border) and "Bersih". A text area below the buttons displays the output: "saori , anda lahir pada 19 December zodiak anda adalah Sagitarius". At the bottom of the window are two more buttons: "Keluar" and "Ke Form 2".

- 8 Buatlah design form 2 sebagai berikut:



- 9 Kita coba dengan melakukan randomize Kita akan melakukan random terhadap angka 1 sampai 5. Jika hasil random sesuai dengan inputan textbox yang anda masukkan (antara 0 sampai 5) ,misal anda masukkan 3 jika randomize menghasilkan 3 , maka keluar dari looping dan memberikan message 3 dan message pada perulangan ke berapa. Lakukan exit looping jika data yang dicari telah ditemukan.
- 10 Untuk melakukan randomize kode programnya adalah :

```
Dim MyValue As Integer
Randomize ' Initialize random-number generator
          harus ditulis
MyValue = CInt(Int((6 * Rnd()) + 0)) '
Generate random value between 0 and 6.
```

- 11 lakukan perulangan dengan while atau do..loop untuk hal diatas contoh outputnya adalah:



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim MyValue As Integer
    Dim lop As Integer = 0
    Randomize() ' Initialize random-number generator harus ditulis
    Do
        lop += 1
```



```
    MyValue = CInt(Int((6 * Rnd()) + 0)) ' Generate random value
between 0 and 6.
    If MyValue = TextBox1.Text Then
        Exit Do
    End If
Loop
MsgBox("value " & MyValue & " looping ke " & lop)
End Sub
```

MODUL 3

PROCEDURE DAN FUNCTION

*Hanya orang yang berani mengambil resiko
besar
yang dapat mencapai kesuksesan besar.
---Robert R. Kennedy---*



Tujuan
Mahasiswa mampu menggunakan procedure dan function dalam pemrograman visual basic.Net

Materi
Procedure
Function

Referensi
Deitel, 2002, Visual Basic .Net How To Program , Prentice Hall, New Jersey

Dalam programming sering kali kita menggunakan sebuah kode program yang sama yang akan digunakan pada bagian yang berbeda - beda dan sering kali kita gunakan berulang ulang. Jika kita menyetik ulang kode program tersebut saat kita membutuhkannya, tentunya hal itu akan menjadikan program kita tidak efisien, tidak enak dilihat dan memakan resource dari program kita.

Lalu Bagaimana cara mengatasi hal diatas??? Salah satu cara yang dapat dilakukan adalah dengan memecah program kita menjadi kesatuan logika yang lebih kecil dimana proses yang sama dikelompokkan menjadi sub-sub program dan kita beri nama untuk kemudian dapat dipanggil oleh kode program lain yang membutuhkan.

Sub - sub program tersebut dapat berupa prosedur dan function.

3.1. Procedure

Prosedure tidak mengembalikan nilai balik terhadap pemanggilnya, maksudnya ketika kita memanggil sebuah prosedur maka statement di dalam prosedur akan dijalankan sampai prosedur itu berakhir.

Prosedure ada yang menggunakan parameter dan ada pula yang tidak.

Prosedure dideklarasikan dengan menggunakan **sub** keyword.

Syntax pembuatan sebuah prosedur :

Tanpa Parameter :

```
[Acces Modifier ] Sub NamaProcedure()  
    Isi_Procedure  
End Sub
```

Dengan Parameter :

```
[Acces Modifier] Sub NamaProcedure (Param As  
TIPE_DATA, Param2 As TIPE_DATA)  
    Isi_Procedure  
End Sub
```

Contoh :

```
Private Sub TampilPesan()  
    MsgBox("Selamat Pagi Dunia",  
    MsgBoxStyle.Information +  
    MsgBoxStyle.OKOnly, "Pesan")  
End Sub
```

Atau dengan menggunakan parameter

```
Private Sub TampilPesan(ByVal pesan As String)  
    MsgBox(pesan, MsgBoxStyle.Information +  
    MsgBoxStyle.OKOnly, "Pesan")  
End Sub
```

3.2. Function

Sama halnya dengan prosedur, function merupakan sub program untuk mempermudah maintenance program yang kita buat, perbedaannya dengan prosedur adalah sebuah function akan mengembalikan sebuah nilai balik pada pemanggilnya, sehingga pemanggil sebuah function harus memiliki penampung dari hasil function, penampung tersebut dapat berupa sebuah variable atau lainnya, misal menampilkan pada message box.

Biasanya function dibuat untuk kode program perhitungan, atau kode program lain yang akan melakukan pengembalian nilai data.

Sama halnya dengan prosedur, function juga ada yang menggunakan parameter dan ada pula yang tidak.

Function dideklarasikan dengan keyword **function**.

Karena function akan mengembalikan sebuah return value atau nilai balik maka harus ada keyword return yang berisi nilai dari yang akan dikembalikan atau dapat juga menggunakan nama dari function tersebut.

Syntaxnya :

Tanpa Parameter

```
[AccessModifier] Function NamaFunction() as  
TipeDtNilaiBalik  
    Isi_Function  
    Return nilaiBalik  
End Function
```

Dengan Parameter :

```
[Acces Modifier] Function NamaFunction (Param
As TIPE_DATA, Param2 As TIPE_DATA)
    Isi_Function
    Return nilaibalik
End Function
```

Contoh :

```
Private Function CekGenap(ByVal angka As
Integer) As Boolean
    If (angka Mod 2 = 0) Then
        'Return False ( menggunakan return)
        atau
        CekGenap = True ( menggunakan nama function
    )
    End If
End Function
```

3.3. Pemanggilan Prosedure dan Function

Seperti telah dijelaskan sebelumnya, sebuah prosedure dan function pasti memiliki sebuah nama. Jika kita ingin menggunakan prosedure dan function tentu kita harus memanggilnya terlebih dahulu dengan menggunakan nama yang dimiliki dan parameter jika ada.

Untuk Pemanggilan sebuah Prosedure :

Misalkan kita ingin memanggil prosedure pada contoh diatas:

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    TampilPesan()           ` Tanpa parameter
    TampilPesan('Halo")    ` Menggunakan
parameter
End Sub
```

Untuk pemanggilan sebuah function :

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Dim a as integer
    A = cekGenap(2) ` menampung nilai
    balik pada variable
    Atau
```

```
    MessageBox.show(cekGenap(3))  
`meampilkan nilai balik pada sebuah message  
End Sub
```

Kode diatas jika kita ingin memanggil function pada contoh diatas :

Latihan

1. Buatlah design form seperti berikut, dimana saat pertama kali load data pada kode member telah terisi:

The screenshot shows a Windows form titled "Form1" with the following elements:

- Four text boxes labeled "Kode Member", "Nama Member", "Tinggi Member", and "Berat Member". The "Kode Member" box contains the text "KM1".
- Two buttons labeled "Simpan" and "Keluar".
- A listview control with a header row containing five columns: "Kode", "Nama", "Tinggi", "Berat", and "Berat Ideal". The listview is currently empty.

2. Deklarasikan variable - variable untuk menyimpan nilai dari inputan nama, tinggi, berat
3. Buatlah sebuah procedure untuk membuat kode member dengan format "KMx" dimana x adalah urutan data, jika merupakan data pertama maka x bernilai 1 jika urutan ke 2 bernilai 2 dst. (gunakan sebuah variable pembantu untuk melakukan counter, variable ini akan bertambah setelah ditekan simpan).
4. Buatlah sebuah procedure untuk menyimpan data - data yang ada dalam control ke dalam variable - variable yang bersesuaian
5. Buatlah sebuah function untuk menghitung nilai berat ideal yaitu $\text{tinggi} + \text{berat} - 110$
6. Buatlah sebuah procedure untuk memasukkan data dalam variable ke dalam listview. (data dimasukkan menggunakan method `items.add` untuk data pada kolom pertama, dan untuk kolom selanjutnya menggunakan method `items parameter berupa index count listview -1 .subitem s.add`).

7. Tombol simpan digunakan untuk menyimpan data pada variable, kemudian menghitung nilai berat ideal dan menampilkan data nya pada list view.
Untuk melakukan hal di atas, Panggilah prosedur dan function yang telah anda buat dengan urutan sesuai permintaan simpan.
8. Berikut adalah contoh output program yang dibuat:

Kode	Nama	Tinggi	Berat	Berat Ideal
KM1	saori	160	45	50
KM2	surya	156	56	46

```

Public Class Form1
    Inherits System.Windows.Forms.Form

    Dim tinggi, berat As Integer
    Dim nama As String

    Dim i As Integer 'variable pembantu penambahan
    counter

    Function kode() As String
        kode = "KM" & i
        TextBox1.Text = kode

    End Function

    Private Sub Form1_Load(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    MyBase.Load
        i = 1
        TextBox1.Text = kode()
    End Sub

```

```
Sub isiList()  
    ListView1.Items.Add(kode)  
    ListView1.Items(ListView1.Items.Count -  
1).SubItems.Add(nama)  
    ListView1.Items(ListView1.Items.Count -  
1).SubItems.Add(tinggi)  
    ListView1.Items(ListView1.Items.Count -  
1).SubItems.Add(berat)  
    ListView1.Items(ListView1.Items.Count -  
1).SubItems.Add(hitungIdeal())  
  
End Sub  
  
Sub isiVariable()  
    nama = TextBox2.Text  
    tinggi = CInt(TextBox3.Text)  
    berat = CInt(TextBox4.Text)  
End Sub  
  
Function hitungIdeal() As Integer  
    hitungIdeal = tinggi - 110  
End Function  
  
Dim status As String  
  
Private Sub Button1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button1.Click  
    isiVariable()  
    isiList()  
  
    TextBox2.Text = ""  
    TextBox3.Text = ""  
    TextBox4.Text = ""  
  
    i += 1  
    kode()  
End Sub  
End Class
```

MODUL 4

LOGIKA PEMROGRAMAN II

*Takut akan kegagalan seharusnya tidak
menjadi alasan untuk tidak mencoba sesuatu
-frederict Smith-*



TUJUAN

Mahasiswa mampu menggunakan logika yang meliputi selection dan looping yang diterapkan dalam sebuah prosedur dan function.

Materi

Penggunaan Selection, looping
Penggunaan dan Pemanggilan Prosedur dan Function

Referensi :

<http://www.startvbdotnet.com/language/loops.aspx>

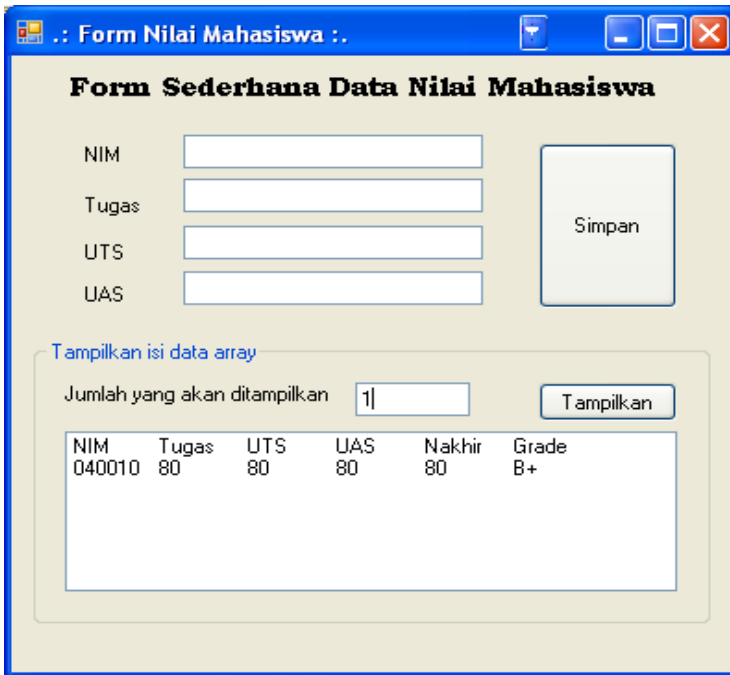
<http://indoprog.tripod.com/>

Deitel, 2002, Visual Basic .Net How To Program , Prentice Hall, New Jersey.

Pada Modul Praktikum ini adalah gabungan dari modul 2 dan modul 3 sehingga anda diminta untuk mempelajari ulang materi selection dan Looping pada modul 2 dan materi Prosedure dan Function pada modul 3.

Latihan

- 1 Kita akan menggunakan array multi dimensi dinamis, procedure, function dan logika pemrograman 1.
- 2 Bukalah design form dan buatlah tampilan seperti berikut ini :



- 3 Skenario dari program diatas adalah kita akan menginputkan nim, tugas, uts, uas kemudian kita simpan, dalam proses menyimpan kita hitung nilai akhir, konversi ke grade dan menyimpannya dalam sebuah array. Jika kita inputkan jumlah yang akan ditampilkan kemudian menekan tombol tampilkan maka textbox yang ada dibawahnya akan menampilkan data dari array sebanyak jumlah yang diinputkan dalam textbox.

- 4 Deklarasikan sebuah jagged array 2 dimensi yang dinamis untuk menyimpan data diatas.
- 5 Buatlah sebuah prosedur untuk menyimpan data yang diinputkan dalam textbox - textbox ke dalam array dengan jumlah kolom 4 yaitu nim, tugas, uts,uas. Prosedur ini akan dipanggil saat kita menekan tombol simpan dan Array yang dibuat ini akan bersifat dinamis dalam panjang barisnya saat kita menekan tombol simpan.
- 6 Buatlah sebuah function untuk menghitung nilai akhir dengan rumus 30% tugas, 30%uts, 40%uas.
- 7 Buatlah prosedur untuk mengkonversi nilai akhir ke dalam huruf yaitu ≥ 95 A+, ≥ 90 dan < 95 A, ≥ 85 dan < 90 A-, ≥ 80 dan < 85 B+, ≥ 75 dan < 80 B, ≥ 70 dan < 75 B- , ≥ 65 dan < 70 C+, ≥ 60 dan < 65 C, ≥ 55 dan < 60 C-, < 55 D.
- 8 Buatlah sebuah prosedur untuk menyimpan nilai akhir dan grade ke dalam sebuah array.
- 9 Buatlah sebuah prosedur untuk menampilkan data yang tersimpan dalam array dengan ketentuan sebelum menampilkan data anda menghitung nilai akhir dan grade dari setiap baris record dalam array dan menyimpan hasil tersebut dalam array anda kemudian baru menampilkan pada textbox.
- 10 Berikut alur logika yang dapat anda ikuti :
 - Buat masing -masing prosedur dan function yang diminta
 - Untuk prosedur menyimpan array buatlah prosedur dengan sebuah parameter dimana parameter tersebut untuk menambahkan panjang baris array yang akan ditambahkan.
 - Panggil prosedur ini pada button simpan click.
 - Untuk prosedur menghitung maupun mengkonversi nilai buatlah memiliki parameter index dari array mana yang akan dihitung.
 - Pada button tampilkan pertama lakukan looping hingga panjang baris array gunakan syntax :
Namaarray.GetLength(0) \rightarrow 0 adalah dimensi karena kita akan ambil panjang baris maka dimensi 0, jika untuk kolom maka 0 kita ganti 1.
 - Dalam looping tersebut panggil prosedur untuk menghitung nilai akhir dan mengkonversi ke dalam grade.
 - Setelah itu panggil prosedur untuk menyimpan array untuk nilai akhir dan grade.

- Kedua Lakukan looping sepanjang array kemudian panggil masing -masing nilai array tersebut dan masukkan nilainya dalam textbox.

MODUL 5

VALIDATING DAN ERROR HANDLING

*"There is nothing either good or bad, but
thinking makes it so".
-Shakespeare-*



TUJUAN

Mahasiswa mampu menangani kesalahan input dari user melalui validasi
Mahasiswa mampu menangani kesalahan logika yang terjadi pada sebuah kode program melalui error handling

Materi

Error Provider
Exception Handling (Try..... CatchFinallyEndTry)

Referensi

<http://www.informit.com/articles/article.asp?p=30936&rl=1>
<http://www.vbdotnetheaven.com/Code/Jun2003/2009.asp>
<http://support.microsoft.com/default.aspx?scid=kb;en-us;315965>
<http://www.informit.com/guides/content.asp?g=dotnet&seqNum=90&rl=1>

5.1. Exception Handling

Dalam pembuatan sebuah aplikasi tentunya kita sering sekali mengalami error atau kesalahan.

Jika kita melakukan kesalahan koding terhadap program yang dibuat, akan mudah sekali kita ketahui, yaitu dalam VB.Net akan secara otomatis compiler akan memberikan tanda berupa garis melengkung mirip cacing pada kode program kita yang salah, namun bagaimana jika error yang terjadi adalah kesalahan dari logika yang kita buat yang hanya akan terdeteksi saat runtime, hal itu kan membuat program kita mengalami break, atau tidak jarang membuat komputer kita hang karena overflow.

Untuk mengatasi hal tersebut, kita akan mempelajari bagaimana cara menangkap error tersebut untuk kita atasi agar tidak terjadi break.

Biasanya kita secara langsung mencegah kemungkinan error yang terjadi dengan menggunakan statement IF.

Seperti contoh berikut :

```
Result = SomeFunction(parameters)
If (Result < 0) Then
  ' Handle an error
Else
  ' Function was successful
End If
```

Structured Exception Handling

VB.Net memberikan tiga keyword yaitu *try*, *catch*, *finally* dalam melakukan exception handling.

Syntax :

Try

Statement yang berpotensi dapat menyebabkan error

Catch variable As jenisexception

Statement untuk meng-handle error yang mungkin terjadi

Finally

Statement final untuk membersihkan error

End try

Try dan *End Try* adalah statement yang harus ada,

jika tidak, error syntax akan muncul,

sedangkan *catch* dan *finally* bersifat optional atau boleh ada boleh tidak, meskipun secara logika kita harus memasukkan paling tidak satu *catch exception* untuk melakukan *handle* terhadap error yang mungkin terjadi.

Kita dapat pula menggunakan multiple *catch* pada satu blok *try* dimana masing-masing *catch* akan menangani *exception* yang berbeda-beda.

Saat error terjadi pada blok *try* maka compiler akan memeriksa pada blok *catch* untuk melakukan *handle* terhadap error,

ada ataupun tidak *exception* yang cocok untuk menangani kesalahan pada blok *try*, compiler akan menjalankan code program pada *finally* block atau mudahnya *finally* block pasti akan dijalankan.

Menangkap Sebuah Exception

Seperti telah dijelaskan sebelumnya, pada blok *catch* segala error yang terjadi pada blok *try* akan dihandle sesuai dengan *catch* yang kita letakkan pada blok *catch*.

Untuk menangkap error secara umum yang dihasilkan oleh suatu proses. Tanpa memperhatikan tipe errornya. Kita dapat menggunakan *catch* pada class *Exception* seperti di bawah ini :

```
Catch Ex As Exception
```

Contoh :

```
Dim a As Integer = 0
Dim b As Integer = 0
Dim c As Integer = 0
Try
    a = b \ c
Catch exc As Exception
    Console.WriteLine("A run-time error occurred")
Finally
    Console.ReadLine()
End Try
```

Dari kode program diatas, saat kita jalankan maka akan di layar akan muncul tulisan "A run-time error occured" karena terjadi error pada block try yaitu error yang terjadi karena pembagian 0 dan block catch dapat menangkap error tersebut kemudian menjalankan statement yang terdapat pada block catch exc.

Kita juga dapat menggunakan multiple catch dengan masing-masing catch menghandle error yang spesifik.

Contoh :

```
Sub Main()
    Dim a As Integer = 2147483647
    Dim b As Integer = 0
    Dim c As Integer = 0
    Try
        a += 1
    Catch exc As DivideByZeroException
        Console.WriteLine("Error: Divide by zero")
    Catch exc As OverflowException
        Console.WriteLine("Error: Overflow")
    Catch exc As Exception
        Console.WriteLine("Error: " & exc.Message)
```

```
Finally  
    Console.ReadLine()  
End Try  
End Sub
```

Pada kode program diatas, terdapat 3 catch exception yang berbeda yaitu `divideByzeroExeption` (Exception yang muncul karena pembagian 0), `overflowexceprion` (Exception yang muncul karena overflow terhadap tipe data), `Exception` (exception untuk keseluruhan exception yang mungkin terjadi).

Jalannya kode program diatas adalah, setelah masuk pada block try kemudian statement yang berpotensi menyebabkan error dijalankan, pada contoh diatas nilai a menjadi lebih dari nilai maksimum yang dapat ditampung oleh sebuah nilai integer, akibatnya error terjadi. Kemudian error tersebut dilempar pada block catch, pertama block catch memeriksa error apakah `DivideByZeroExeption`? Jika ya maka akan menjalankan statement dibawahnya, jika tidak maka akan memeriksa pada catch berikutnya apakah `Overflow exception` ? demikian seterusnya hingga ditemukan catch yang sesuai. Setelah menemukan yang sesuai dalam kasus ini `overflow exception` maka compiler akan menjalankan statement pada catch `overflow exception` kemudian setelah itu menjalankan `finally` statement

Throw Exception

Selain menangkap error yang mungkin terjadi dengan menggunakan `catch` kita juga dapat `handle` error yang terjadi dengan `melempar /throw exception`. Keyword yang digunakan adalah **throw** .

Berikut contoh dari penggunaan `throw exeption`. Misal kita akan melakukan validasi terhadap data yang akan dimasukkan dalam sebuah property jika nilai yang dimasukkan kurang dari 0 maka akan `melempar exception` tidak boleh dibawah 0.

Kode program :

```
Public Class clsPerson
    Private mintAge As Integer
    Public Property Age() As Integer
        Get
            Age = mintAge
        End Get

        Set(ByVal Value As Integer)
            If Value >= 0 Then
                mintAge = Value
            Else
                Throw New ApplicationException("Age cannot be
negative")
            End If
        End Set
    End Property
End Class
```

5.2. Error Provider

Error provider adalah salah satu komponen yang tersedia pada VB.Net untuk memudahkan kita dalam melakukan validasi terhadap kemungkinan error yang terjadi khususnya untuk melakukan validasi pada inputan yang berhubungan dengan user interface.

Error provider akan memberikan notify berupa message yang akan ditampilkan pada tooltip dan juga icon untuk memberitahu pada user jika terjadi error.

Methods yang sering digunakan untuk menampilkan pesan ke user adalah method `SetError`, seperti pada contoh :

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e_
As System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "" Then
        ErrorProvider1.SetError(TextBox1,
"Cannot leave textbox blank")
    Else
        ErrorProvider1.SetError(TextBox1, "")
    End If
End Sub
```

Contoh:

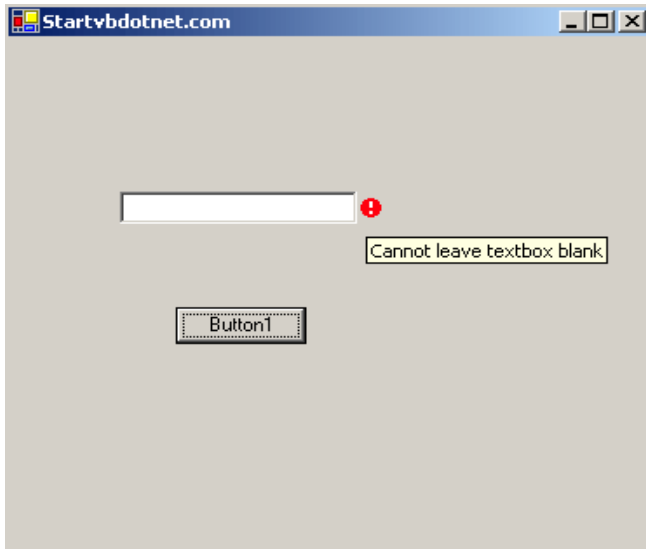
Misalkan kita ingin membuat sebuah textbox telepon yang hanya dapat diisi oleh inputan berupa angka dan tidak boleh kosong dimana error divalidasi saat menekan button1. Langkah yang akan kita lakukan :

- Masukkan sebuah textbox dan button dari toolbox pada sebuah form
- Masukkan komponen error provider dari toolbox pada form (dalam sebuah form, walaupun kita akan melakukan validasi terhadap banyak control dalam 1 form misalnya textbox ada 3 buah, button, combo box, dsb... error provider yang dimasukkan dalam form cukup satu buah saja.)
- Buka code editor
- Pada Button 1 klik tulis kode program

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e_
As System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "" or
isnumeric(textbox1.text) = false Then
        ErrorProvider1.SetError(TextBox1,
"Cannot leave textbox blank and must be
numeric")
    Else
        ErrorProvider1.SetError(TextBox1, "")
    End If
End Sub
```

```
End If  
End Sub
```

- Run Program tersebut kemudian tekan button1 tanpa mengetik apapun pada textbox, maka akan tampil seperti gambar berikut



Penggunaan error provider dalam program dapat dilakukan dengan dua cara, yaitu dengan memvalidasi per control atau control validated dimana pengecekan terhadap error akan dilakukan setelah kita meninggalkan control tersebut satu persatu,

atau dengan form validated yaitu pengecekan dilakukan secara bersama sama untuk semua control sekaligus saat sebuah event dilakukan, misalnya saja saat button simpan untuk mengecek apakah semua inputan dalam masing- masing control di form telah sesuai.

Untuk melakukan control validated, kita akan menggunakan dua event yang dimiliki oleh control yaitu validated dan validating.

Validating merupakan event yang digunakan untuk mengecek apakah inputan telah benar

Validated merupakan event yang digunakan untuk membersihkan error yang terjadi

Berikut adalah contoh dari code program untuk pengecekan textbox apakah angka, jika bukan angka maka error provider akan muncul

menampilkan messagebox kesalahan, dan jika telah terisi benar maka error provider akan hilang.

```
Private Sub TextBox1_Validated(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
TextBox1.Validated

    Me.ErrorProvider1.SetError(TextBox1, "")
End Sub

Private Sub TextBox1_Validating(ByVal sender As
Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles
TextBox1.Validating

    Try

        If Not IsNumeric(TextBox1.Text) Then

            Me.ErrorProvider1.SetError(Me.TextBox1,
            "Please Enter a Numeric Value!")

            Throw New Exception("Please enter a
            numeric ")

        End If

    Catch ex As Exception

        MsgBox(ex.Message)

    End Try

End Sub
```

Latihan Modul 1

- 1 Kita akan mencoba menggunakan komponen error provider untuk validasi terhadap inputan yang kita inginkan. Message

singkat yang tampil pada tool tip saat icon kesalahan muncul kita buat sesuai error yang terjadi.

- 2 Untuk error provider kita akan mencoba dua macam penggunaannya yaitu control validated dan form validated.
- 3 Pertama kita akan coba menggunakan control validated. Buatlah design form seperti berikut :

The screenshot shows a Windows application window titled "Form Control Validasi". Inside the window, there is a group box labeled "Control Validasi". Within this group box, there are three input controls: a text box for "Nama", a text box for "Nilai", and a date picker for "Tanggal Masuk" which is currently set to "2/25/2008". Below the group box, there are two buttons: "Go to Form Validated" and "Go To Try Catch Form".

- 4 Untuk textbox nama validasi yang dilakukan tidak boleh kosong.(cek lah property text yang dimiliki oleh textbox apakah kosong atau tidak atau anda cek berdasarkan length yang dimiliki, jika =0 maka set lah error providernya)

```
Private Sub TextBox1_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox1.Validating
    If TextBox1.Text = "" Then
        ErrorProvider1.SetError(TextBox1, "tidak boleh kosong")
    End If
End Sub

Private Sub TextBox1_Validated(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.Validated
    If TextBox1.Text <> "" Then
        ErrorProvider1.SetError(TextBox1, "")
    End If
End Sub
```

- 5 Untuk textbox nilai, validasi yang dilakukan isian harus angka.


```
Private Sub TextBox2_Validated(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox2.Validated
    If IsNumeric(TextBox2.Text) = True Then
        ErrorProvider1.SetError(TextBox2, "")
    End If
End Sub
```

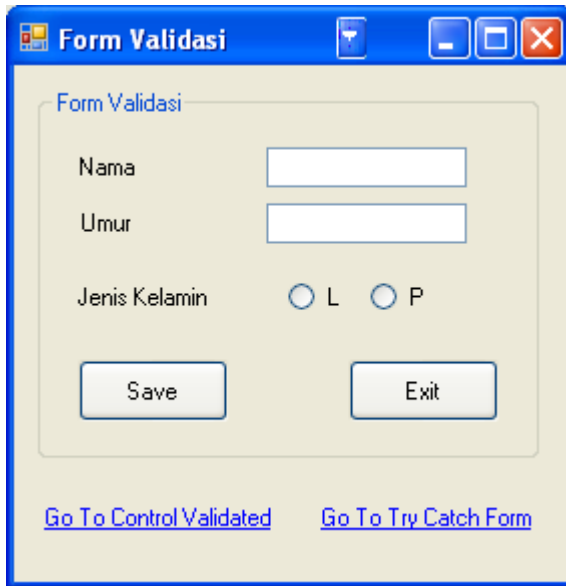
```
Private Sub TextBox2_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox2.Validating
    If IsNumeric(TextBox2.Text) = False Then
        ErrorProvider1.SetError(TextBox2, "harus angka")
    End If
End Sub
```

- 6 Untuk date time picker adalah data yang dipilih tidak boleh lebih 3 minggu dari sekarang.(Cek berdasarkan property value)

```
Private Sub DateTimePicker1_Validated(ByVal sender As Object, ByVal e As System.EventArgs) Handles DateTimePicker1.Validated
    If DateDiff(DateInterval.Day, Now, DateTimePicker1.Value) <= 21 Then
        ErrorProvider1.SetError(DateTimePicker1, "")
    End If
End Sub
```

```
Private Sub Dtp_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles DateTimePicker1.Validating
    If DateDiff(DateInterval.Day, Now.Date, DateTimePicker1.Value) > 21 Then
        ErrorProvider1.SetError(DateTimePicker1, "maksimal 3 minggu")
    End If
End Sub
```

- 7 Untuk melakukan validasi per komponen gunakan 2 event validasi yang dimiliki setiap komponen tersebut yaitu validating dan validated.
- 8 Kita telah mencoba menggunakan error provider dengan pengecekan per komponen, sekarang kita akan coba melakukan validasi pengecekan seluruh komponen sekaligus dengan pada saat event tertentu.
- 9 Buatlah design form seperti berikut :



10 Pada saat tombol save maka seluruh control dilakukan pengecekan yaitu untuk textbox nama tidak boleh kosong, untuk textbox umur harus angka, dan jenis kelamin harus dipilih salah satu. (letakkan kode pengecekan dan set error provider dalam button click)

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "" Then
        ErrorProvider1.SetError(TextBox1, "tidak boleh kosong")
    Else
        ErrorProvider1.SetError(TextBox1, "")
    End If

    If IsNumeric(TextBox2.Text) = False Then
        ErrorProvider1.SetError(TextBox2, "harus angka")
    Else
        ErrorProvider1.SetError(TextBox2, "")
    End If

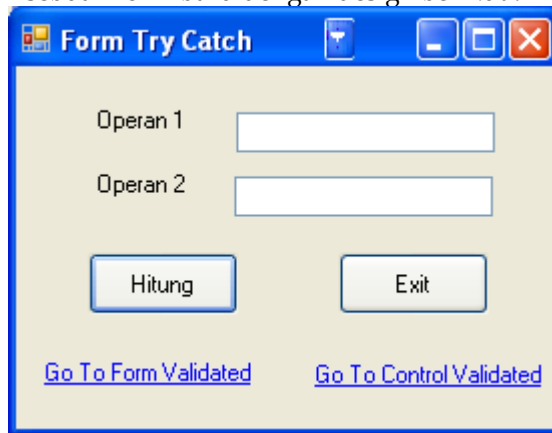
    If RadioButton1.Checked = False And RadioButton2.Checked = False Then
        ErrorProvider1.SetError(RadioButton1, "harus pilih salah satu")
    Else
        ErrorProvider1.SetError(RadioButton1, "")
    End If
End Sub
```

11 Tombol exit untuk keluar dari aplikasi, link label go to validated untuk memanggil form control dan go to try cath form untuk memanggil form 3 yang akan kita buat selanjutnya.

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

```
Private Sub LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabel.LinkClickedEventArgs) Handles LinkLabel2.LinkClicked
    Dim form2 As New Form2
    Me.Hide()
    form2.Show()
End Sub
```

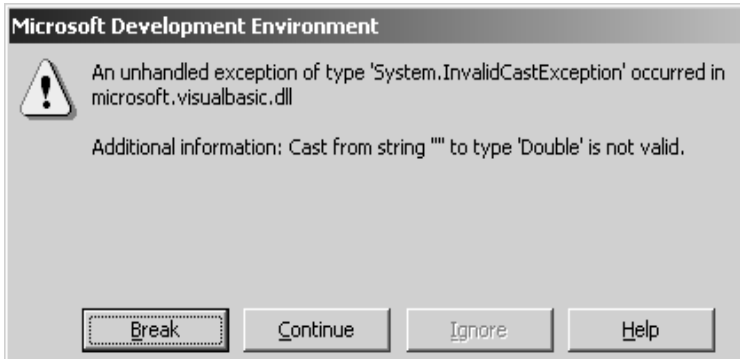
- 12 Anda sekarang telah menguasai penggunaan error provider, sekarang kita akan mencoba mempelajari lebih dalam mengenai exception.
- 13 buatlah sebuah form baru dengan design berikut :



- 14 buka kode editor untuk event button hitung click, tulislah kode program seperti di bawah ini :

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

- 15 Sekarang jalankan program anda, kemudian tanpa memasukkan nilai apapun dalam kedua textbox, tekan button1. Apa yang terjadi ????? break bukan? Terdapat pesan kesalahan :



- 16 Bagaimana agar exception diatas tidak tampil ? disinilah kita gunakan exception handling.
- 17 Dari kode program diatas maka kode yang mungkin menyebabkan error adalah kode perhitungan dengan inputan dari textbox sehingga ubahlah kode program pada button1 dengan menambah exception handling try..catch.. untuk menghindari error diatas.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Try
        Dim x As Double
        x = TextBox1.Text * TextBox2.Text
        MsgBox(x)
    Catch ex As Exception
    End Try
End Sub
```

MODUL 6

PENGENALAN KONSEP OOP

*"Kegagalan adalah awal dari Keberhasilan
Jangan Pernah Menyerah pada Kegagalan"*
-- Anonymous--



TUJUAN

Mahasiswa mampu memahami konsep OOP dan class pada VB.NET

Materi

Pengenalan OOP, class, class member, konstruktor dan destructor, Properties dan Data Member, deklarasi dan inialisasi object

Referensi

Visual Basic .NET How To Program Second Edition, Deitel (Chapter 8)

6.1. Pengenalan Konsep OOP (Object Oriented Programming)

Kita sudah sering mendengar istilah OOP atau Object Oriented Programming. Jika diterjemahkan dalam bahasa Indonesia menjadi Pemrograman Berorientasi Objek alias PBO. Tapi apa sebenarnya OOP/PBO itu? OOP adalah bahasa pemrograman yang memiliki konsep objek seperti halnya objek di dunia nyata. Objek di dunia nyata misalnya saja “Burung Articuno” (salah satu burung legendaris di anime Pokèmon). Burung tadi kita anggap sebagai objek, dia memiliki paruh, sayap, bulu, kaki, dan bagian tubuh lainnya. Burung juga mempunyai kemampuan untuk terbang. Jika dijadikan dalam konsep OOP maka jadinya adalah sebagai berikut:

Dunia Nyata

Klas : Aves
Objek : Burung Articuno
Memiliki : Paruh, Sayap, Bulu, dll
Warna Bulu : Biru
Kemampuan : Terbang

Dunia OOP

Class : Aves
Object : Burung Articuno
Data Member : Paruh, Sayap, Bulu, dll
Properties Bulu : Biru
Method : Terbang

Dilihat dari contoh “Burung Articuno” di atas dapat dijelaskan secara singkat jika:

Class

Class merupakan cetak biru (*blue print*) dari objek atau juga bisa disebut cetakan (*template*) untuk membuat objek. Class sendiri merupakan generalisasi dari beberapa objek yang memiliki kesamaan ciri dan sifat. Seperti kita lihat, class “Aves” merupakan generalisasi dari klas burung yang memiliki kesamaan ciri dan sifat.

Syntax pembuatan class :

```
[Acces Modifier] Class NamaClass
----- Data Member
-----Methods
-----Properties
-----Events
End Class
```

Data Member dan Properties

Biasa juga disebut dengan atribut. Secara gampang dapat disebut jika data member itu “apa-apa yang dimiliki class”. Biasanya data member ini bersifat private sehingga tidak bisa langsung diakses dari luar class. Oleh karena itu butuh yang namanya properties untuk mengubah ataupun mengambil nilai dari data member. Setiap data member pasti punya properties masing-masing yang nantinya bisa diatur. Dengan properties inilah kita bisa membuat suatu objek berbeda antara yang satu dengan objek lainnya.

Syntax pembuatan sebuah data member :

```
[Acces modifier] namaDataMemeber as typeDataMember
```

Method

Biasa juga disebut dengan operation (operasi). Secara gampang juga bisa disebut “apa-apa yang bisa dilakukan oleh class”. Sebenarnya method di VB.Net sendiri dibagi menjadi 2 bagian, yaitu Procedure dan Function. Yang membedakan hanyalah nilai baliknya saja. Procedure tidak punya nilai balik, Function punya nilai balik.

Constructor vs Destructor

Constructor adalah suatu method yang pertama kali dijalankan pada saat suatu object terbentuk (diinstantkan dari sebuah class). Dapat di **overload** tanpa menggunakan perintah **overloads**.

Dalam VB.Net pembuatan constructor menggunakan keyword *New*

Contoh :

```
Sub New ()
    Me.paruh = "Runcing"
    Me.sayap = "Lebar"
```

```
Me.bulu = Color.Blue  
End Sub
```

```
Sub New(ByVal xParuh As String, ByVal xBulu As Color)  
Me.paruh = xParuh  
Me.bulu = xBulu  
End Sub
```

Destructor adalah method yang dijalankan sebelum suatu obyek tidak digunakan lagi atau pada saat object dihancurkan.

Contoh :

```
Protected Overrides Sub Finalize()  
' Menutup aplikasi  
Application.Exit()  
End Sub
```

Berikut adalah contoh lengkap dari pembuatan sebuah class Aves serta data member , properties dan method yang dimiliki :

```
Public Class Aves  
Private paruh As String  
Private sayap As String  
Private bulu As Color  
  
Sub New()  
Me.paruh = ""  
Me.sayap = ""  
Me.bulu = Color.Blue  
End Sub  
  
Public Property BentukParuh() As String  
Get  
Return Me.paruh  
End Get  
Set(ByVal Value As String)
```



```
        Me.paruh = Value
    End Set
End Property

Public Property BentukSayap() As String
    Get
        Return Me.sayap
    End Get
    Set(ByVal Value As String)
        Me.sayap = Value
    End Set
End Property

Public Property WarnaBulu() As Color
    Get
        Return Me.bulu
    End Get
    Set(ByVal Value As Color)
        Me.bulu = Value
    End Set
End Property

Public Sub Terbang()
    Dim mampuTerbang As Boolean
    mampuTerbang = False
End Sub

Public Function BisaTerbang() As Boolean
    Return True
End Function
End Class
```

Object adalah Instance of Class

Object adalah bentuk instant dari sebuah class, dalam contoh di atas “burung Articulo” adalah bentuk instant dari class “Aves”. Bisa juga disebut objek “burung Articulo” itu dibentuk dari class “Aves”. Untuk membuat object dari sebuah class diperlukan syntax seperti berikut:

```
Dim namaObject As New namaClass
```

Misal :

```
Dim burungArticuno As New Aves
```

Deklarasi dan Inisialisasi Object

Berikut ini contoh untuk mendeklarasikan dan inisialisasi instant class yang membentuk suatu object:

Constructors

```
Dim burungElang As New Aves("Runcing",  
"Lebar", Color.Brown)
```

Atau

```
Dim burungBeo As Aves  
burungBeo = New Aves("Bengkok", "Pendek",  
Color.Black)
```

Properties

```
Dim burungElang As New Aves()  
burungElang.bentukParuh = "Runcing"  
MessageBox.Show("Bentuk Paruh dari Burung  
Elang adalah " & _  
burungElang.bentukParuh)
```

Methods

```
Dim burungElang As New Aves()  
MessageBox.Show("Apakah Burung Elang bisa  
terbang ? " & _  
burungElang.bisaTerbang())
```

6.2. Mengapa Harus OOP ?

Kalo ditanya kenapa sih harus pake OOP/PBO, apa emang lagi nge-trend ato emang ada keuntungan lain dibandingin dengan bahasa

pemrograman terstruktur. Hmm.. beberapa list jawaban berikut pasti bisa membantu:

1. Modularity (Pemodulan)

Karena kita sering bikin program maka semakin lama coding yang kita buat akan semakin banyak, bahkan bisa saja aplikasi yang dibuat tadi terdiri dari ribuan baris coding. Permasalahan akan mulai muncul, seperti baris coding yang sangat banyak dalam satu file dan juga kebingungan tentang keterkaitan antara coding yang satu dengan coding yang lain. Dengan OOP, pengkodean akan makin mudah, karna kita menggunakan class yang berfungsi sebagai modul. Kita bisa memecah beberapa coding yang mempunyai fungsi tertentu dalam beberapa modul berupa class-class. Dengan demikian kita tidak harus menaruh semua coding dalam satu file.

2. Reusability (Penggunaan Kembali)

Reusability berarti coding yang telah dibuat dapat dipakai kembali di masa yang akan datang. Dengan adanya kemampuan ini, maka pembuatan suatu aplikasi dapat lebih mudah dan lebih singkat. Dimana untuk memudahkan dalam pencarian class yang telah dibuat, maka class yang telah dibuat tersebut dapat dikelompokkan dalam suatu Namespaces.

3. Extendibility (Perluasan)

Setiap aplikasi yang kita buat pasti berbeda-beda, sehingga diperlukan kemampuan untuk dapat memperluas class yang ada agar sesuai dengan keperluan saat ini.

Latihan

- 1 Untuk lebih memahami konsep class, sekarang buatlah sebuah solution baru kemudian buatlah sebuah class dengan klik kanan pada project dalam solution explorer kemudian add new dan pilihlah class. Beri nama class baru anda cPraktikan.
- 2 Ketentuan dari class yang akan anda buat adalah :
Class cPraktikan memiliki data member berupa nama dengan type string, nim dengan tipe string, group dengan type string, nharian dengan type integer, nUjian dengan type integer dan nakhir dengan type integer.
- 3 Buatlah sebuah constructor untuk class anda dengan ketentuan :
 - constructor tanpa parameter
 - constructor dengan parameter nim dan nama
 - constructor dengan parameter nim, nama, group
- 4 Setelah property data member nim dengan ketentuan inputan harus memiliki panjang karakter 6. (cek panjang karakter value dengan menggunakan method length. Misal value bernama x maka cara mencari panjang karakter `x.length`)
- 5 setelah property data member group dengan syarat panjang karakter yang dimasukkan adalah 8 karakter dengan format harus GP6106XX, dengan XX terserah. (untuk pengecekan pertama cek dulu panjang dari value yang diinputkan apakah telah sesuai yaitu 8 karakter jika ya maka lakukan pengecekan untuk 6 karakter pertama apakah benar GP6106 dengan menggunakan method `Microsoft.VisualBasic.Left (nama value yang akan dicek, panjang karakter yang diambil dari kiri), misalkan Microsoft.VisualBasic.Left(x,2)`
- 6 setelah property nharian dan nujian agar inputan antara 0 sampai 100 .
- 7 buatlah sebuah function untuk menghitung nakhir dengan rumus 50% nilai harian dan 50% nilai ujian.

Public Class cPraktikan

```
Dim nama As String
Dim nim As String
Dim group As String
Dim nHarian As Integer
Dim nUjian As Integer
Dim nAkhir As Integer
```

```
Sub New()
    nama = "test"
    nim = "0123"
    group = "123"
    nHarian = 0
    nUjian = 0
    nAkhir = 0
```

```
End Sub
```

```
Sub New(ByVal nm As String, ByVal pnim As String)
    nama = nm
    nim = pnim
End Sub
```

```
Sub New(ByVal nm As String, ByVal pnim As String,
ByVal g As String)
    nama = nm
    nim = pnim
    group = g
End Sub
```

```
Public Property propNIM() As String
    Get
        Return nim
    End Get
    Set(ByVal Value As String)
        If Value.Length = 6 Then
            nim = Value
        Else
            MsgBox("nim salah")
            nim = ""
        End If
    End Set
End Property
```

```
Public Property propNAMA() As String
    Get
        Return nama
    End Get
    Set(ByVal Value As String)
```

```
        nama = Value
    End Set
End Property

Public Property propGroup() As String
    Get
        Return group
    End Get
    Set(ByVal Value As String)
        If Value.Length = 8 Then
            If Microsoft.VisualBasic.Left(Value,
6).Equals("GP6106") Then
                group = Value
            Else
                MsgBox("group salah")
                group = ""
            End If
        Else
            MsgBox("group salah")
            group = ""
        End If
    End Set
End Property

Public Property propNharian() As Integer
    Get
        Return nHarian
    End Get
    Set(ByVal Value As Integer)
        If Value < 100 And Value > 0 Then
            nHarian = Value
        Else
            nHarian = 0
        End If
    End Set
End Property

Public Property propNujian() As Integer
    Get
        Return nUjian
    End Get
    Set(ByVal Value As Integer)
        If Value < 100 And Value > 0 Then
            nUjian = Value
        Else
            nUjian = 0
        End If
    End Set
End Property
```

```
Public Property propNAkhir() As Integer
    Get
        Return nAkhir
    End Get
    Set(ByVal Value As Integer)
        nAkhir = Value
    End Set
End Property

Function hitung() As Integer
    nAkhir = 0.5 * nHarian + 0.5 * nUjian
    Return nAkhir
End Function
```

End Class

- 8 Sekarang anda telah memiliki sebuah class cPraktikan. Kita akan coba menerapkan class yang kita buat dalam sebuah form. Anda akan memasukkan nilai pada masing -masing textbox kemudian saat tombol simpan ditekan data tersebut akan diset sebagai nilai dari data member pada cPraktikum, kemudian dilakukan perhitungan nakhir dengan pemanggilan function cpraktikum, kemudian nilai data member dari class cpraktikum tersebut akan ditampilkan pada listbox.
- 9 Buatlah sebuah form seperti gambar berikut ini

The image shows a Windows application window titled "Form Input Nilai Praktikum". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main content area has a light beige background and is titled "Input Nilai Praktikum". Inside this area, there is a section titled "Data Praktikan" which contains five text input fields labeled "Group", "NIM", "Nama", "Nilai Harian", and "Nilai Ujian". Below these fields are two buttons: "Constructor" and "Non Constructor". At the bottom of the form is a large empty text area labeled "ListBox1" with scrollbars.

- 10 Pada program yang akan kita buat nantinya object dari class Cpraktikan hanya akan dipanggil pada saat button klik, tidak ada method atau function atau event lain yang memanggil sehingga kita definisikan pada button klik saja, namun jika digunakan oleh salah satu atau lebih dari hal yang disebutkan diatas maka harus anda definisikan agar dapat digunakan oleh seluruh method yang ada di form tersebut yaitu di luar method apapun tetapi di dalam class.
- 11 Pada tombol konstruktor digunakan untuk instance-kan sebuah object dari class cpraktikum dengan menggunakan konstruktor dengan parameter nim, nama dan group, sedangkan button non konstruktor digunakan untuk instance-kan sebuah objek dari

class praktikum tanpa konstruktor. Dengan kode program sebagai berikut:

```
Dim namaobject as new namaclass
```

Dim iclass as new cPraktikum

- 12 Untuk melakukan pemanggilan pada property data member atau procedure yang terdapat dalam class lakukan dengan

```
namaclass.namamemberclass
```

dimana namamemberclass adalah function maupun property yang telah anda deskripsikan pada class, saat anda mengetik . seluruh member class akan terlihat anda tinggal pilih salah satu. Contoh :



- 13 Tombol simpan digunakan untuk menghitung nilai akhir dan memasukkan data - data dalam listbox dengan ketentuan tampilan layout:

```
Nim nama nharian nujian nakhir
```

Gunakan vbTab untuk tabular string pada textbox.

```
Public Class Form3
    Inherits System.Windows.Forms.Form

    Dim iclass As cPraktikan

    Private Sub konstruktor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        iclass = New cPraktikan("jack", "01234", "GP610601")
        iclass.propNharian = TextBox4.Text
    End Sub
End Class
```

```
        iclass.propNujian = TextBox5.Text
        iclass.hitung()

        If iclass.propGroup.Equals("") Or
        iclass.propNIM.Equals("") Then
            MsgBox("data ada yang salah")
        Else
            ListBox1.Items.Add(iclass.propNAMA & vbTab &
            iclass.propNIM & vbTab & iclass.propGroup & vbTab &
            iclass.propNharian & vbTab & iclass.propNujian & vbTab &
            iclass.propNAKhir)
        End If

    End Sub

    Private Sub nonKonstruktor_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button2.Click
        iclass = New cPraktikan
        iclass.propNAMA = TextBox3.Text
        iclass.propNIM = TextBox2.Text
        iclass.propGroup = TextBox1.Text
        iclass.propNharian = TextBox4.Text
        iclass.propNujian = TextBox5.Text
        iclass.hitung()

        If iclass.propGroup.Equals("") Or
        iclass.propNIM.Equals("") Then
            MsgBox("data ada yang salah")
        Else
            ListBox1.Items.Add(iclass.propNAMA & vbTab &
            iclass.propNIM & vbTab & iclass.propGroup & vbTab &
            iclass.propNharian & vbTab & iclass.propNujian & vbTab &
            iclass.propNAKhir)
        End If
    End Sub
End Class
```

MODUL 7

CONTROL, PROPERTIES, METHOD, EVENT

*" Semangat yang kuat mampu mengatasi apa
saja cobaan yang datang"
--Anonymous--*



TUJUAN

Mahasiswa dapat memahami penggunaan control, properties, method dan event dalam VB.NET
Mahasiswa dapat menggunakan MenuStrip, ToolStrip, MDI Form, dan StatusStrip.

Materi

MenuStrip, ToolStrip, MDI Form, StatusStrip

Referensi

<http://www.startvbdotnet.com/controls/menus.aspx>
<http://www.sampublishing.com/articles/article.asp?p=25741&rl=1>
<http://www.startvbdotnet.com/forms/mdi.aspx>
http://www.geekpedia.com/tutorial100_Using-the-StatusBar-control.html

7.1. Control dan Properties

Control adalah sebuah object yang dapat diletakkan dan menjadi bagian dari sebuah form sehingga user dapat berinteraksi dengan aplikasi yang kita buat. Dalam visual basic.net kita dapat bekerja sebuah control dengan dua jalan, saat design time atau run time. Bekerja dengan sebuah control saat design time berarti control akan tampak pada form design, kita dapat men-drag drop control atau double klik dari toolbox dan melakukan setting terhadap properties dari properties window. Sedangkan bekerja dengan sebuah control saat run time berarti control tidak akan tampak pada form design, dibuat dan di setting properties melalui koding dan akan tampak saat program dijalankan. Contoh dari control adalah textbox, label, button, dll.

Setiap manusia pasti memiliki identitas dan keunikan, demikian juga sebuah control, control adalah sebuah object yang memiliki identitas dan ciri khas yang membedakan antara satu control dengan control yang lain walaupun dengan basic class yang sama. Misalnya dua object button, masing-masing memiliki ukuran, text, warna yang berbeda satu sama lain. Atau jika dianalogikan dengan manusia, setiap manusia walaupun dalam satu keluarga pasti masing-masing memiliki ciri khas satu sama lain, misalnya nama, tinggi badan, bentuk wajah dsb.

Ciri khas tersebut (nama,tinggi badan, ukuran, text, warna, dsb) disebut properties.

Kita dapat mengatur properties dengan dua jalan, melalui properties window atau melalui code editor.

Method dan Event

Method merupakan kejadian yang dapat dilakukan oleh sebuah control. Atau dapat kita analogikan dengan manusia, manusia memiliki control yaitu tangan dan tangan memiliki banyak sekali method yaitu diangkat, diputar, dicuci, dsb.

Event adalah kejadian yang dilakukan oleh sebuah control yang dipicu oleh suatu hal. Jika dianalogikan dengan manusia, sebelumnya manusia memiliki tangan sebagai control dan beberapa method kemudian ada suatu hal pemicu misal ada orang yang akan memukul kita, otomatis tangan kita angkat, nah saat kita

mengangkat tangan untuk menangkis tersebut sebuah event terjadi yaitu mengangkat tangan.

MenuStrip

Setiap pemakai komputer tentunya tidak asing dengan menu. Menu –menu yang ada pada sebuah aplikasi memungkinkan pengguna untuk memilih apa yang akan dilakukan pada aplikasi kita.

Dalam VB.net MenuStrip komponen adalah kontainer untuk struktur sebuah menu dalam sebuah form. Default event dari ToolStrip MenuItem adalah Click event

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

Beberapa properties dalam menu item :

Checked: Default value adalah false. Jika kita merubah menjadi true maka checkmark akan tampak di sebelah kiri menu.

DropDownItems: Digunakan untuk menambah, mengubah, menghapus sub item atau sub menu dari menu yang dipilih.

ShortcutKeys: Untuk membuat short key sebagai shortcuts untuk menu item.

ToolTipText: Untuk memberikan tooltip pada menu ketika mouse diarahkan ke menu

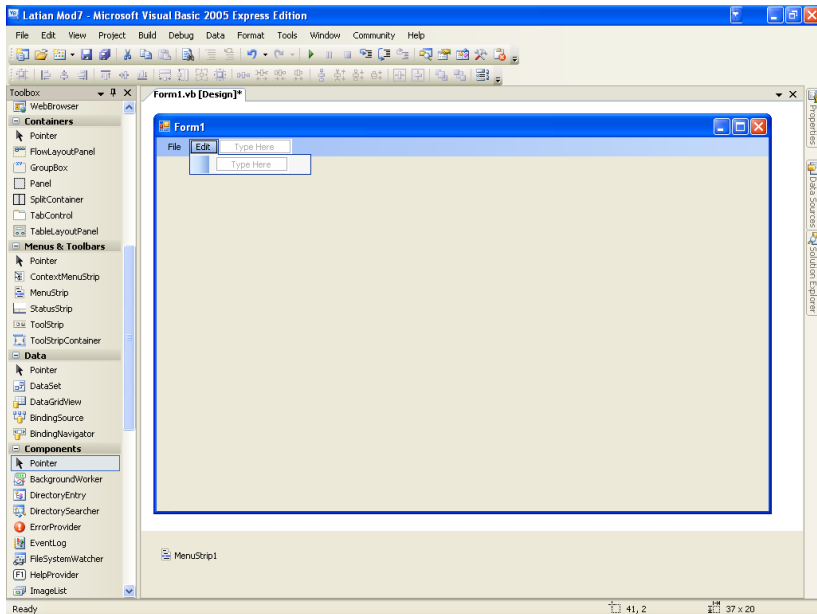
Berikut contoh pembuatan sebuah menu :

- Drag and Drop MenuStrip dari toolbox pada tab Menus & Toolbars dalam form.

- Drag and Drop sebuah textbox dalam form

Setelah mainmenu berada pada form, kita akan melihat tulisan "Type Here" pada pojok kiri atas, klik tulisan tersebut kemudian ketik text apa yang akan anda buat dalam menu misalkan file.

Untuk membuat submenu item dapat dilakukan dengan menekan enter atau menggunakan panah ke bawah pada keyboard. Untuk membuat menu item lain gunakan panah ke kanan pada keyboard



Untuk contoh ini kertiakkan File sebagai menu item, “Text” sebagai sub item yang pertama dan “Exit” sebagai sub item yang kedua.

Double klik pada menu untuk masuk dalam kode editor.

Kita akan membuat program saat memilih Text maka pada textbox1 akan tertulis ‘welcome to Menu” jika kita memilih Exit maka akan keluar dari aplikasi.

Berikut syntax pada kode editor untuk sub menu item textbox dan out :

```
Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripMenuItem.Click
    TextBox1.Text = "Welcome to Menus"
End Sub

Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripMenuItem.Click
    Me.Close()
End Sub
```

ToolBar / ToolStrip

Sama halnya dengan menu, pengguna windows tentunya tidak asing dengan ToolStrip yaitu tombol - tombol yang berada dibawah menu.

Menggunakan ToolStrip sama dengan menggunakan komponen lain dalam VB.Net dengan memasukkan komponen dalam form, setting properties dan menuliskan beberapa kode program pada code editor.

Dalam ToolStrip kita menggunakan image sebagai penanda misalnya image disket untuk menyimpan. Untuk hal ini kita akan menggunakan komponen imagelist.

Berikut langkah yang akan kita lakukan untuk membuat sebuah ToolStrip :

Masukkan komponen ToolStrip dalam form

Masukkan komponen Imagelist dalam form. Pada properties Image masukkan image-image yang ingin anda letakkan dalam image list.

Untuk menambahkan button pada ToolStrip klik dropdown pada ToolStrip, kemudian pilih Button. Untuk menambahkan gambar, kita bisa gunakan ImageList atau menggunakan properties Image yang ada pada ToolStrip. Dan setting pada properties agar ImageScaling 'SizeToFit'.

Untuk contoh kali ini masukkan 2 button pada ToolStrip kemudian setting properties text New dan Open.

Masuk dalam code Editor dengan Double klik ToolStrip.

Kita akan melakukan pengecekan button mana yang dieksekusi dengan menyeleksi event argument yang dikirim berdasarkan text yang terdapat pada button apakah button New atau Open, Anda juga dapat melakukan seleksi berdasarkan tag atau properties lain yang dapat kita beri nilai unik untuk setiap button. Sintax pada ToolStrip :

```
Private Sub ToolStripButton1_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton1.Click
    MsgBox("Choose New Button")
End Sub
```

```
Private Sub ToolStripButton2_Click(ByVal sender  
As System.Object, ByVal e As System.EventArgs)  
Handles ToolStripButton2.Click  
    MsgBox("Choose Open Button")  
End Sub
```

7.2. MDI Form

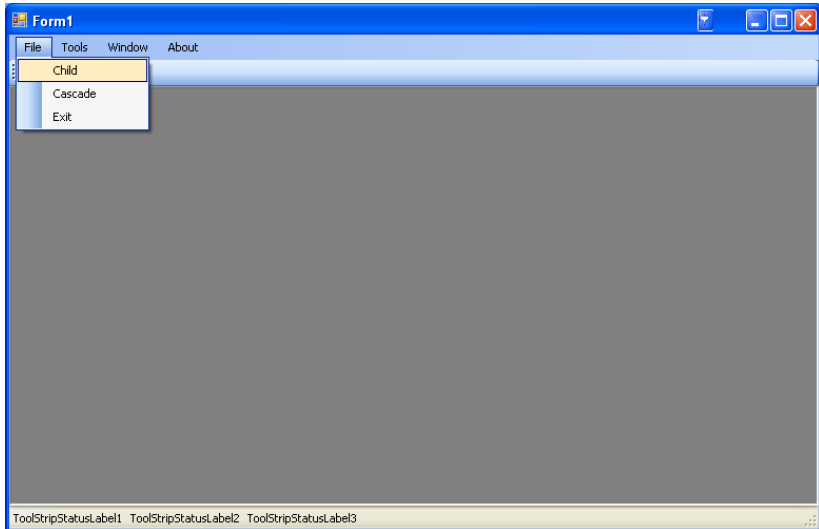
MDI (Multiple Document Interface) adalah aplikasi dimana kita dapat melihat dan bekerja dengan beberapa dokumen pada satu waktu. Contoh dari sebuah aplikasi MDI adalah aplikasi Excel. Lawan dari MDI application adalah SDI (Single Document Application) dimana kita hanya dapat bekerja pada satu dokumen saja misalnya microsoft word.

Untuk membuat sebuah form sebagai container menu setting properties `IsMdiContainer` pada form menjadi true.

Agar lebih mudah dimengerti kita coba membuat sebuah MDI Form

Setelah membuat window application baru kemudian setting **properties `IsMdiContainer` menjadi true**

Masukkan MainMenu kemudian buat menu Item File dengan sub menu "Child" , "Cascade Window", dan "Exit".

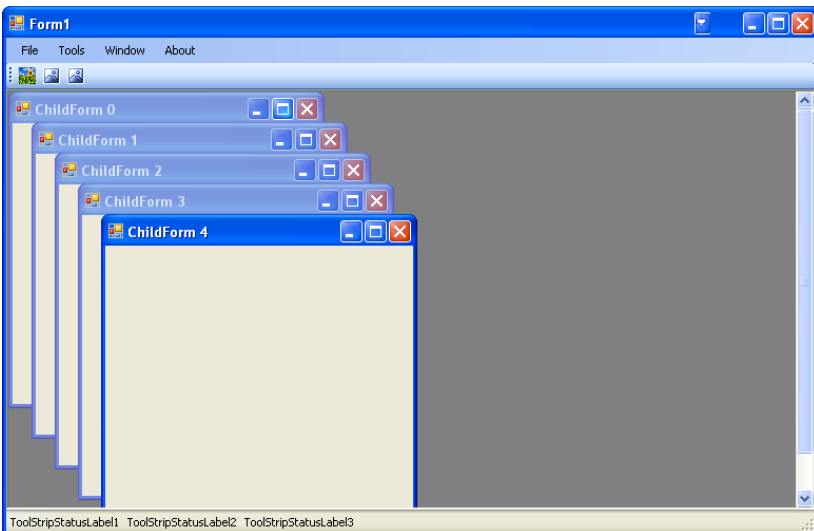


Tulis kode program berikut pada code Editor :

```
Public Class Form1
    Dim childForm As Integer = 0
    Dim childForms(5) As Form
    Private Sub ChildToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChildToolStripMenuItem.Click
        childForms(childForm) = New Form
        childForms(childForm).Text = "ChildForm" & Str(childForm)
        'setting title for child windows and incrementing the number with an array
        childForms(childForm).MdiParent = Me
        childForms(childForm).Show()
        childForm += 1
    End Sub
    Private Sub CascadeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CascadeToolStripMenuItem.Click
```

```
CascadeToolStripMenuItem.Click  
Me.LayoutMdi (MdiLayout.Cascade)  
    'arranging child windows on the parent  
    form with predefined LayoutMdi method  
    'Different layouts available are,  
    ArrangeIcons, Cascade, TileHorizontal,  
    TileVertical  
End Sub  
  
Private Sub ExitToolStripMenuItem1_Click (ByVal  
sender As System.Object, ByVal e As  
System.EventArgs) Handles  
ExitToolStripMenuItem1.Click  
  
    Me.Close ()  
    'closing the application  
End Sub
```

Dari kode program diatas jika kita memilih “Child” maka output yang akan muncul



StatusBar / StatusStrip

StatusStrip digunakan untuk menampilkan pesan pada bagian bawah dari form. Pada umumnya StatusStrip digunakan untuk

menampilkan informasi berupa waktu saat ini, halaman, pesan error, dll.

Propeties yang penting pada StatusStrip :

Items : untuk menambahkan panel atau item-item dalam StatusStrip

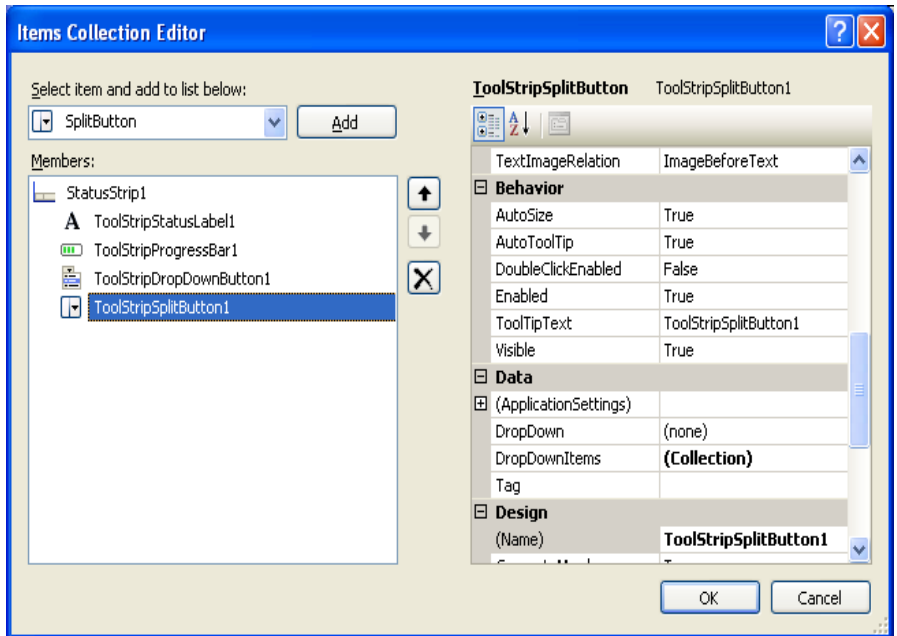
Pada StatusStrip terdapat 4 jenis Status yang dapat ditambahkan di dalam form, yaitu StatusLabel, ProgressBar, DropDownButton, SplitButton.

Mengedit item status pada StatusStrip :

Untuk memasukkan item dalam StatusStripe dapat digunakan dengan dua cara melalui properties window atau melalui koding.

Melalui properties window :

Pilih properties items, kemudian tambahkan items sesuai dengan yang diinginkan. Berikut contoh propeties items pada StatusStrip :

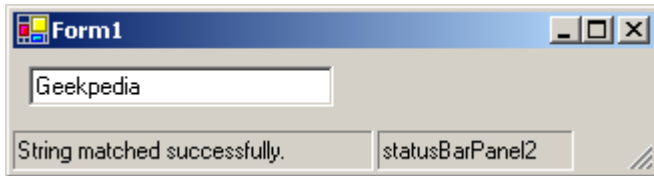


Melalui kode program :

Contoh sintax yang digunakan untuk menambah item label baru :

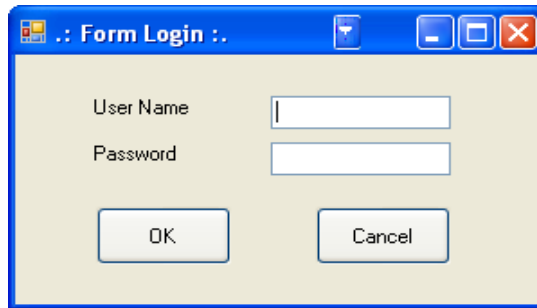
```
StatusStrip1.Items.Add("text yang tertulis  
pada item label Status Strip")
```

Berikut contoh status strip yang menggunakan items :



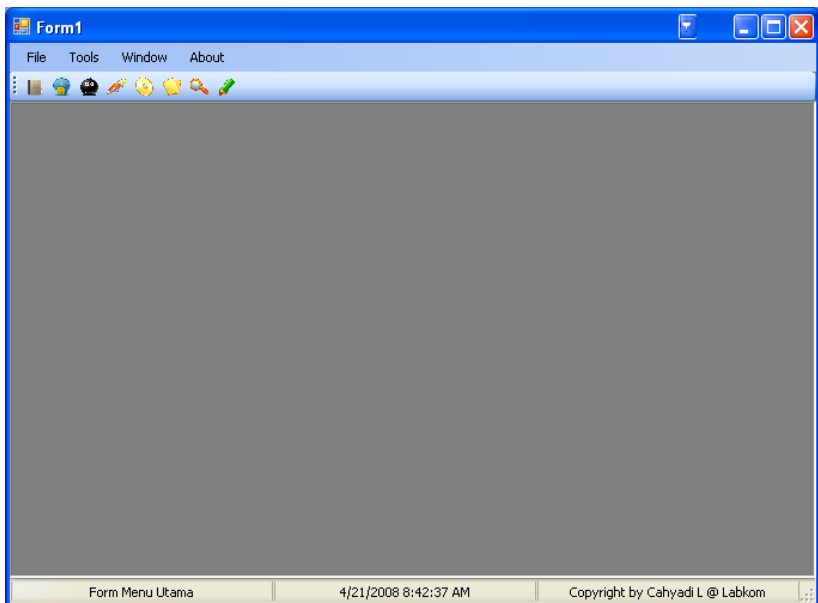
Latihan

1. Buatlah aplikasi windows dengan tampilan form Login sebagai berikut :



The image shows a standard Windows-style dialog box with a blue title bar containing the text ': Form Login :'. The dialog has a light beige background. It features two text input fields: the first is labeled 'User Name' and the second is labeled 'Password'. Below these fields are two rectangular buttons with rounded corners, labeled 'OK' and 'Cancel'.

Setelah Login (status login true), ketika login dengan username = admin dan password = admin maka status login akan true dan menampilkan form Aplikasi sebagai berikut dan semua menu pada form aplikasi dapat diakses kecuali menu login: **(Nilai 30)**



Pada Form tersebut tambahkan menu sebagai berikut : **(Nilai 30)**

File

- Log In → Memanggil form Login
- Log Out → Menutup form child dan mematikan menuExit
- Keluar dari aplikasi

Tools

- New Form → Memanggil form Baru
- Remove Active Form → Menghapus form child yang aktif

Remove All Form → Menutup semua form child

Windows

- Cascade → Mengatur tampilan secara Cascade
- Tile Horizontal → Mengatur tampilan secara Horizontal
- Tile Vertical → Mengatur tampilan secara Vertical

About

Jika memilih menu Log Out menu yang dapat diakses hanya menu Login dan Menu About. **(Nilai 05)**

ToolStrip berfungsi seperti menu yang telah anda buat. **(Nilai 10)**

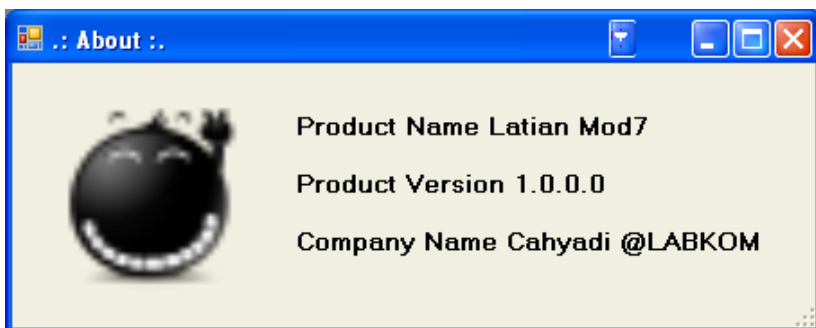
Status Strip berisi tiga item Label dengan ketentuan : **(Nilai 20)**

Item pertama berisi text dari MDI Child yang aktif

Item kedua berisi Tanggal dan jam system

Item ketiga berisi text "Copyright @ Nama Anda"

Tampilan untuk Form About adalah seperti Berikut ini :



Informasi yang ditampilkan pada form about didapat dari file AssemblyInfo.vb

MODUL 8

DEPLOYMENT

*" Terdapat banyak kemungkinan untuk gagal
kerana kejayaan hanya boleh dicapai dengan
satu perkara yaitu USAHA"
--anonymous--*



TUJUAN

Mahasiswa mampu melakukan deploy terhadap dekstop application

Materi

Deployment dekstop application

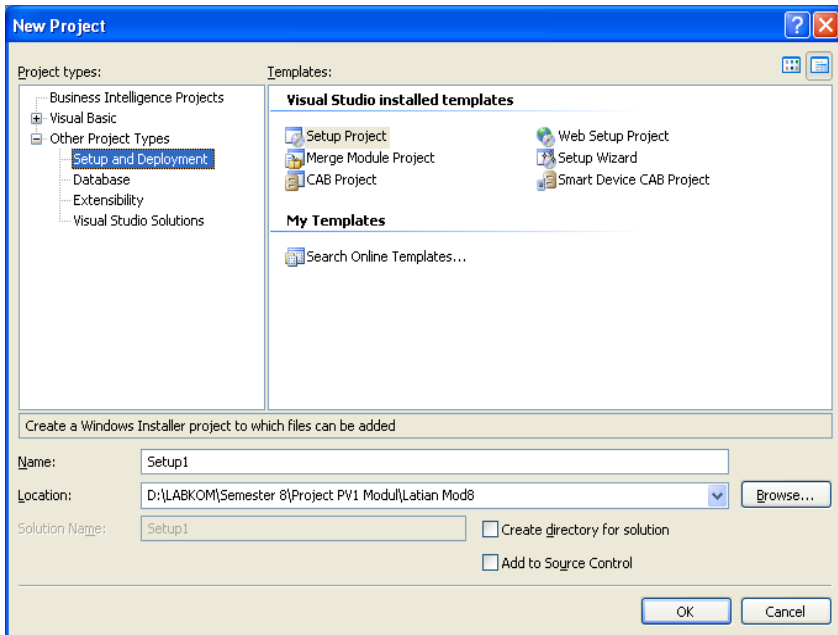
Referensi

Developing Microsoft .NET Applications for Windows (Chapter 9)

8.1. Visual Studio .NET Deployment Project Templates

Deployment adalah proses distribusi aplikasi yang telah dibuat di suatu komputer ke komputer lain dengan cara meng-install-nya. Keuntungan utama dari kegiatan deployment adalah memudahkan dan mempercepat proses distribusi aplikasi yang telah kita buat.

Pada Visual Studio .NET 2005 versi Express tidak terdapat deployment project. Visual Studio .NET 2005 mempunyai enam jenis templates untuk digunakan pada pembuatan penyebaran aplikasi. Adapun keenam jenis templates tersebut adalah :



Setup Project Template

Digunakan untuk membuat windows installer standar.

Web Setup Project Template

Digunakan untuk membuat program setup windows installer yang dapat digunakan untuk meng-install suatu aplikasi Web ke suatu directory virtual pada web server.

Merge Module Project Template

Merge modul memungkinkan satu set files dibungkus ke dalam suatu file “easy-to-use” yang dapat digunakan kembali dan dishare pada program setup berbasis teknologi windows installer. Biasanya digunakan untuk mempacketkan komponen dan semua yang berkaitan kedalam suatu unit.

Setup Wizard Template

Digunakan untuk membuat templates project melalui suatu wizard.

CAB Project Template

Digunakan untuk membuat file cabinet (.cab). Sebuah file cabinet dapat berisi banyak file, tetapi bukan merupakan logic instalasi. Umumnya digunakan untuk mempacketkan komponen ke dalam suatu file, yang kemudian disebar pada suatu web server yang memungkinkan client memdownloadnya.

Smart Device CAB Project Template

Digunakan untuk membuat file cabinet (.cab) untuk melakukan deployment Pocket PC, Smartphone, dan aplikasi yang berbasis Windows CE.

8.2. Membuat Installer Package

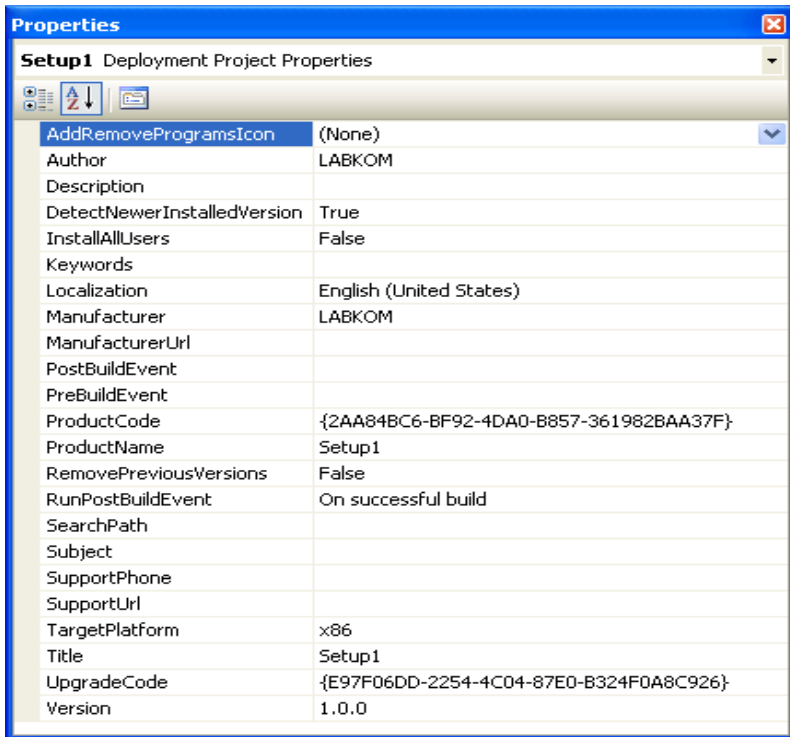
Untuk memulai membuat paket installer, maka langkah-langkah yang harus dilakukan adalah :

Klik menu File, New, Project

Pada dialog box New Project, pilih Tab Other Project Types, klik Setup and Deployment pada panel Other Project Types

Pilih Setup Project, dan isi nama dari aplikasi installer yang akan dibuat

Konfigurasi Project Properties



Berikut ini beberapa properties yang ada pada suatu project installer, dengan fungsi-fungsi sebagai berikut :

1. AddremoveProgramIcon : Icon yang akan ditampilkan pada Add/Remove Programs
2. Author : Nama pembuat aplikasi
3. Description : Deskripsi yang ditampilkan pada saat instalasi
4. Manufacturer menentukan instansi pembuat aplikasi
5. Product name : Menentukan nama produk dari aplikasi.

Jenis-Jenis Setup Editors

Dalam membuat suatu project installer, Anda harus mengatur beberapa pilihan pada editor yang berbeda-beda sesuai dengan kebutuhan. Adapun beberapa editor berikut dengan fungsinya dapat dijelaskan sebagai berikut :

File System Editor

Untuk menambah file dan membuat shortcut pada paket instalasi

Registry Editor

Untuk memanipulasi entri registry pada komputer tujuan

File Types Editor

Berhubungan dengan custom file pada aplikasi. Digunakan jika aplikasi yang dibuat ingin dihubungkan dengan suatu custom file tertentu.

User Interface Editor

Berhubungan dengan pengaturan tampilan pada saat instalasi

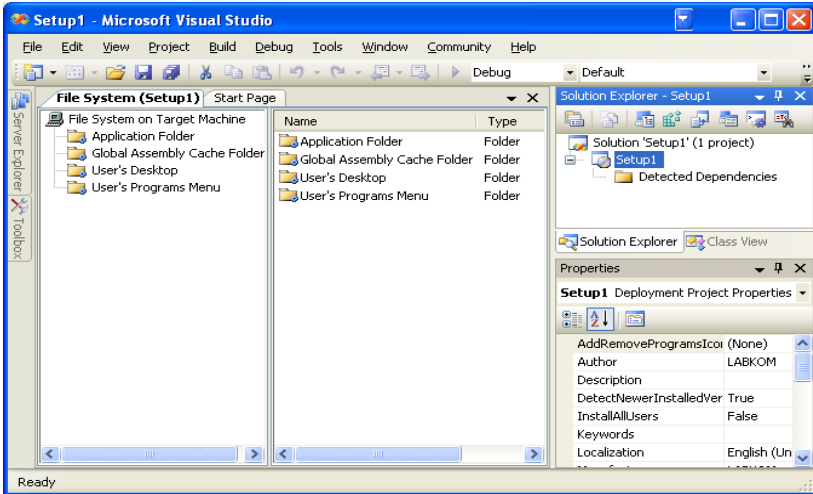
Custom Actions Editor

Untuk menjalankan program lain selama proses install dan uninstall dijalankan.

File System Editor

Pada File System Editor ini Anda dapat menambahkan project output files, assemblies dan file lain pada deployment project yang akan dibuat. Anda juga dapat menentukan lokasi directory untuk menempatkan file hasil instalasi pada komputer target.

Berikut ini tampilan pada File System Editor :



Application Folder

Secara default folder aplikasi ini akan mengacu pada path [ProgramFilesFolder][Manufacturer]\[ProductName].

Global Assembly Cache Folder

Folder ini digunakan untuk mengatur assemblies apa yang harus diinstall pada komputer target.

User's Desktop

Folder ini merupakan placeholder untuk file dan folder yang akan muncul pada dekstop komputer target. Secara default mengacu pada folder [DriveName]\Documents and Settings\[UserName]\Desktop

User's Programs Menu

Folder ini merupakan placeholder yang akan ditampilkan pada programs group. Secara default mengacu pada folder

[DriveName]\Documents and Settings\[UserName]\Start Menu\Programs

Dengan menggunakan File System Editor, Anda dapat menambahkan beberapa item berikut ini pada folder tertentu yang dipilih :

Folder

Untuk membuat folder pada directory tertentu

Project Output

Menentukan letak dari output suatu project (file .dll atau exe) pada solution, yang akan di buat pada komputer target.

File

Untuk membuat file yang tidak tergantung pada instalasi komputer target, seperti Readme.txt

Assembly

Untuk menentukan assembly dan menambah reference assembly yang dibutuhkan

Berikut ini langkah-langkah yang harus dilakukan untuk mengatur file-file apa saja yang akan dicopy pada komputer target :

Menambah Primary Output, Pilih Menu **Project, Add, Project Output**. Setelah memilih directory Application Folder, pilih **Primary Output** pada **Add Project Output Group**.

Untuk menjalankan aplikasi yang dibuat dengan Visual Basic .NET, membutuhkan .NET runtime yang otomatis ditambahkan pada project installer yang dibuat. Merge module yang berisi .NET runtime adalah *dotnetfxredist_x86_enu.msm* yang dapat dilihat pada solution explorer. .NET runtime harus sudah ada pada komputer target sebelum instalasi dilakukan, karena itu Anda harus menset properti Exclude pada file tersebut menjadi true.

Untuk menambahkan folder atau files tertentu, Anda tinggal klik kanan pada tertentu dan memilih **Add, Folder** atau **File**.

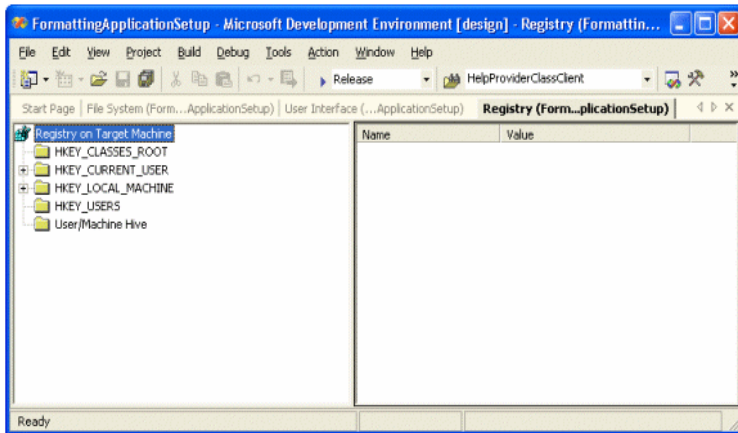
Untuk file Readme.txt seharusnya tersedia sebagai file individual sehingga pemakai dapat membacanya sebelum proses instalasi. Maka Anda harus mengubah properti **PackageAs** menjadi **vsdpaLoose**

Untuk membuat shortcut pada desktop atau program menu, Anda tinggal klik kanan pada file **Primary Output** dan pilih **Create Shortcut**. Ubah nama shortcut yang terbentuk serta drag dan drop shortcut tersebut pada folder yang sesuai (user's desktop atau user's programs menu)

Registry Editor

Pada Registry Editor ini Anda dapat mengatur setting Registry pada komputer target dimana aplikasi akan diinstall. Secara default, Registry Editor menampilkan standard Windows Registry Key, seperti HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, and HKEY_USERS. Dengan menggunakan Registry Editor, Anda dapat menambahkan Custom Registry Key di bawah key standard tersebut.

Berikut ini tampilan Registry Editor :



Untuk menambahkan informasi Registry tambahan yang dibutuhkan oleh Aplikasi Anda pada saat runtime, Anda dapat menggunakan HKEY_LOCAL_MACHINE\Software\[Manufacturer] atau HKEY_CURRENT_USER\Software\[Manufacturer].

File Types Editor

Pada File Types Editor, Anda dapat memilih file yang berasosiasi pada komputer target dengan memanggil suatu aplikasi dengan suatu extension file. Klik ganda file untuk menampilkan aplikasi yang sesuai. Ketika inialisasi dilaksanakan, maka extension dan deskripsi dari tipe file akan muncul pada daftar tipe file di Windows Explorer.

Custom Actions Editor

Pada Custom Actions Editor Anda dapat membuat link pada program yang lain, dimana program tersebut akan dijalankan pada saat aplikasi diinstall. Untuk membuat sebuah custom action, Anda membutuhkan file .dll atau .exe dan menambahkannya pada saat mendeploy project. Custom actions dapat diasosiasikan dengan salah satu dari empat hasil instalasi berikut : Install, Commit, Rollback dan Uninstall.

Launch Conditions Editor

Dengan menggunakan Launch Conditions Editor, Anda dapat menspesifikasi kondisi yang harus dijumpai secara berurutan pada saat setup dijalankan. Jika user berusaha menginstall aplikasi pada suatu sistem yang tidak ada launch conditions-nya, maka setup tidak akan dijalankan. Selama setting launch conditions, Anda dapat menspesifikasi pencarian terhadap suatu file tertentu, Registry Key, Component dan lainnya.

User Interface Editor

Sesuai dengan namanya, pada User Interface Editor ini akan diatur urutan dari dialog interface yang akan ditampilkan selama proses install aplikasi dijalankan pada komputer target. Terdiri dari dua model level :

Install – Section ini terdiri dari dialog boxes yang akan ditampilkan ketika user menjalankan installer

Administrative Install – Section ini terdiri dari dialog boxes yang akan ditampilkan ketika system administrator mengupload installer pada sebuah lokasi network.

Pada section Install dan Administrative Install juga terdiri dari sub section sebagai berikut :

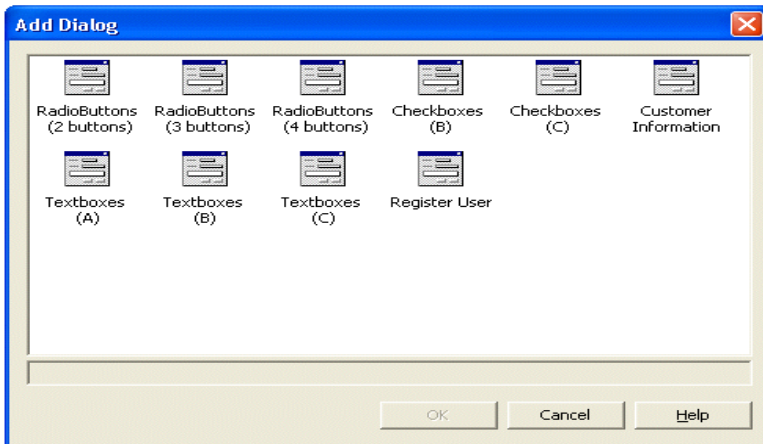
Start Dialog Boxes – Ditampilkan sebelum instalasi dimulai

Progress Dialog Boxes – Membuat Anda dapat menyediakan feedback pada user selama proses instalasi

End Dialog Boxes - Menginformasikan kepada user bahwa instalasi telah sukses dilakukan. Juga dapat digunakan untuk menampilkan file Readme.

Menambah Additional Custom Dialogs

Untuk menambah Dialog Boxes selain yang sudah ada selama proses instalasi sangat gampang. Dengan memilih menu **Action** atau malalui klik kanan, **Add Dialog** maka akan muncul pilihan dialog boxes seperti tampilan di bawah ini :



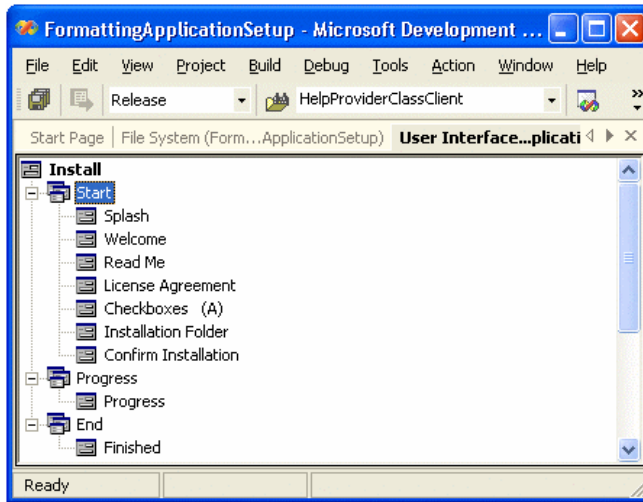
Berikut ini beberapa contoh penggunaannya :

Tambahkan beberapa Custom Dialog, seperti : Checkboxes (A), License Agreement, Read Me dan Splash dialog boxes sesuai dengan urutan pada gambar di bawah ini

Pilih Dialog Boxes License Agreement dan ubah properti LicenseFile dengan file .rtf yang sudah disediakan.

Pilih Dialog Boxes Read Me, dan ubah properti ReadMeFile dengan file .rtf yang sudah disediakan.

Dialog Boxes Checkboxes(A) digunakan untuk menanyakan user apakah ingin menampilkan shortcut pada Desktop atau tidak. Untuk melakukan hal itu maka ubahlah properti Checkboxes(A) seperti tampilan di bawah ini :



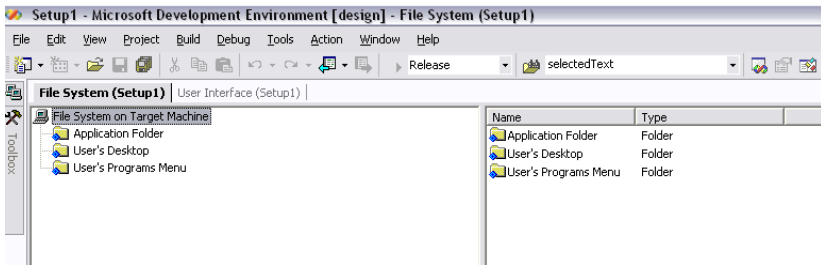
Jangan lupa mengubah properti **Condition** pada User's Desktop Folder pada File System Editor dengan SHORTCUTDEMO.

Membangun Installer Package

Setelah semua Editor pada pembuatan instalasi aplikasi diatur, maka untuk membangun paket installer Anda tinggal memilih menu **Build, Build Solution**. Selanjutnya jalankan file Setup.exe atau ApplicationSetup.msi

Latihan

1. Kita akan mencoba membuat sebuah installer paket
2. Kita akan membuat set up dari aplikasi yang telah anda buat pada modul sebelumnya. Sekarang buatlah sebuah solution baru saat memilih tipe project yang anda inginkan pilihlah **SetupAplikasi** yang bertipe **Setup and Deployment Projects**.
3. Anda akan tampil pada window seperti gambar berikut :

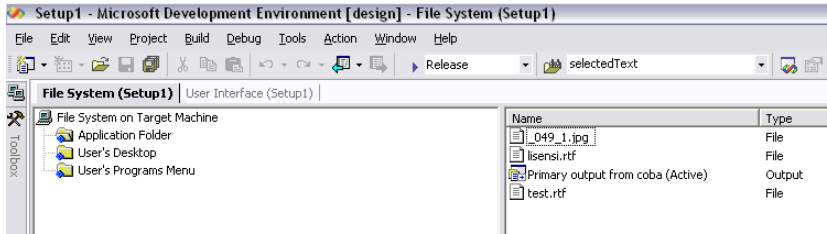


Untuk menambahkan project yang anda ingin install. Klik kanan pada solution explorer yang terletak di sebelah kanan window. Pilih add kemudian existing project. Setelah itu maka akan bertambah 1 project baru pada tree di solution anda.

Klik kanan Application Folder yang terlihat pada gambar diatas, lalu pilih add kemudian project output. Akan tampil window addprojectoutput group, jika kita telah menambahkan project seperti yang telah dijelaskan diatas, maka combo box project akan terisi sesuai dengan project yang kita tambahkan tadi. Pilih primary output, lalu tekan ok.

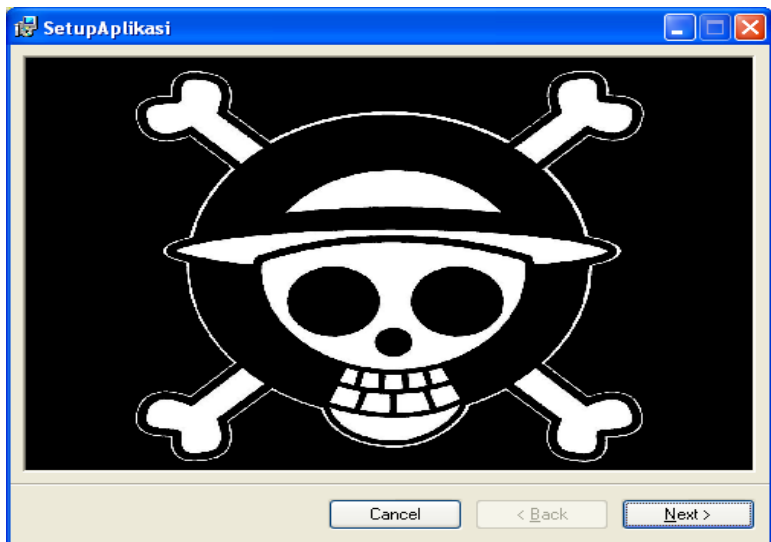
Jika ingin menggunakan gambar - gambar pada instalasi, menggunakan file rtf untuk readme dan lisensi, file - file yang anda butuhkan tersebut anda tambahkan pada application folder, dengan melakukan klik kanan add kemudian pilih file.

Berikut contoh capture dari penjelasan diatas.



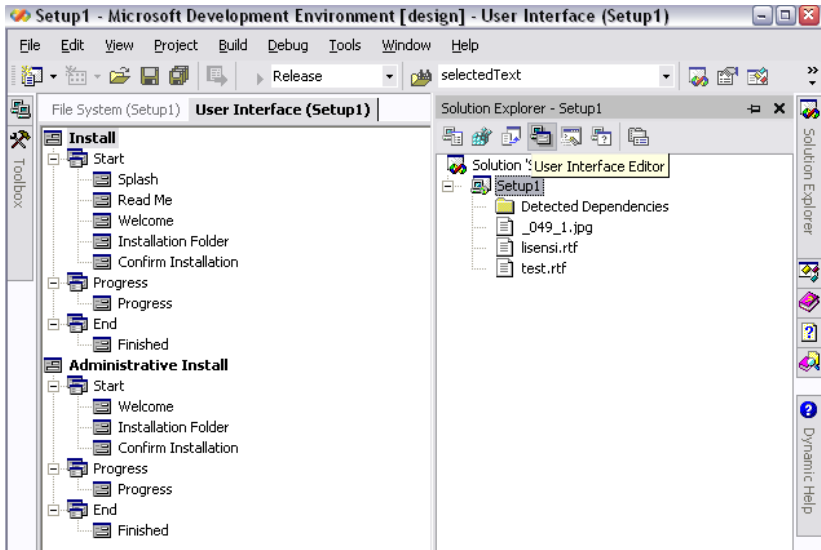
Urutan tampilan DialogBox pada saat instalasi dapat dilihat pada gambar-gambar di bawah ini.

Misalkan untuk tampilan splash adalah seperti berikut



Anda dapat menggunakan gambar lain yang anda suka.

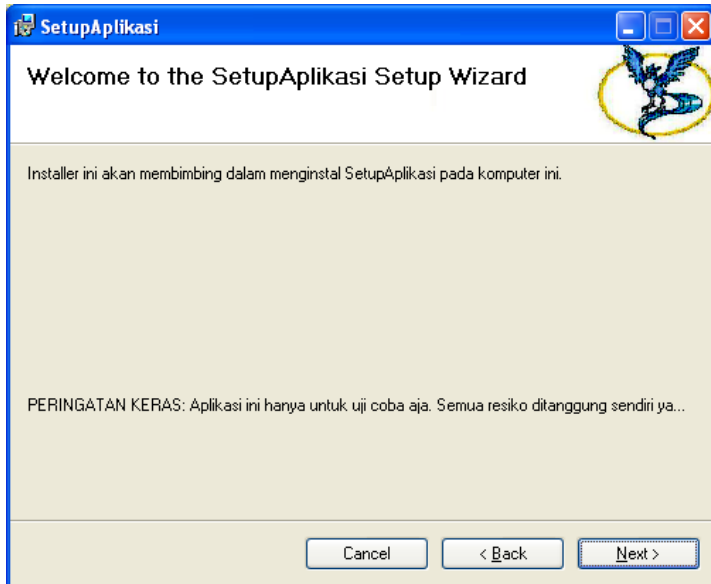
Untuk hal diatas, tambahkan file gambar yang diinginkan pada application folder. Lalu pindahlah ke tab user interface editor. Akan tampak seperti gambar berikut



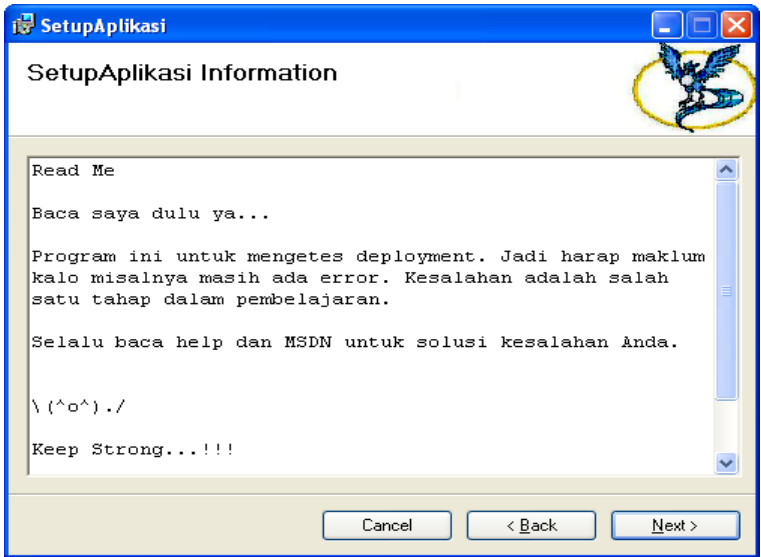
Pada gambar diatas treeview yang menunjukkan urutan instalasi yang akan dilakukan. Jika ingin menambahkan, misalnya splash maka anda klik kanan start pada treeview lalu pilih add Dialog. Untuk mengganti urutan instalasi, misal readme dipindah setelah welcome, anda tinggal drag tree readme ke bawah welcome.

Aturlah properties splash bitmap sesuai dengan gambar yang anda inginkan.

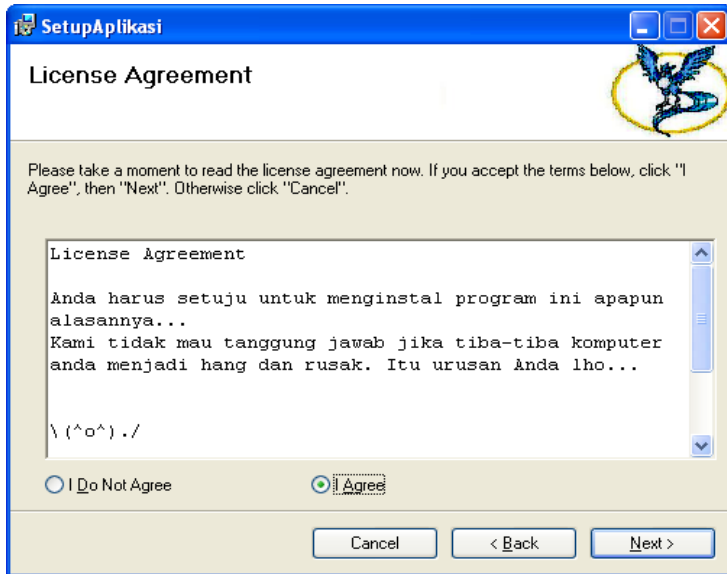
4. Atur properties dari DialogBox Welcome agar bisa menampilkan gambar banner serta bisa menampilkan Copyright Warning dan Welcome Text seperti gambar berikut:



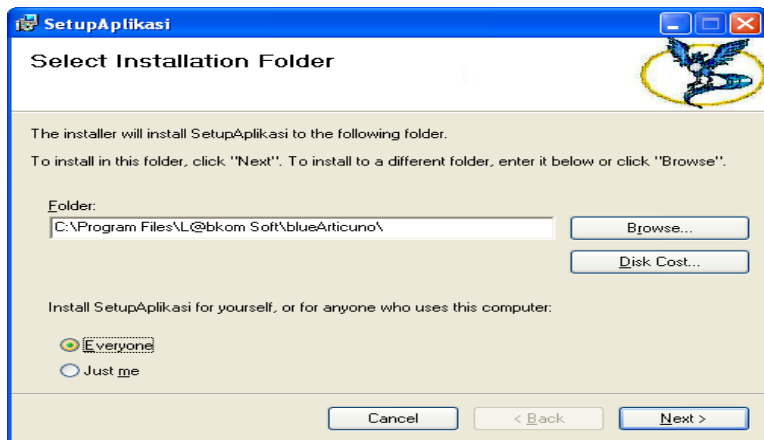
5. Buat file ReadMe.rtf dengan WordPad lalu tampilkan pada DialogBox Read Me seperti gambar di bawah ini, cobalah membuat file lain dengan extensi yang berbeda, .doc, .txt apakah dapat terbaca pada set up. Anda harus telah menambahkan file rtf pada application folder, lalu aturlah properties pada user interface, pada bagian readme.



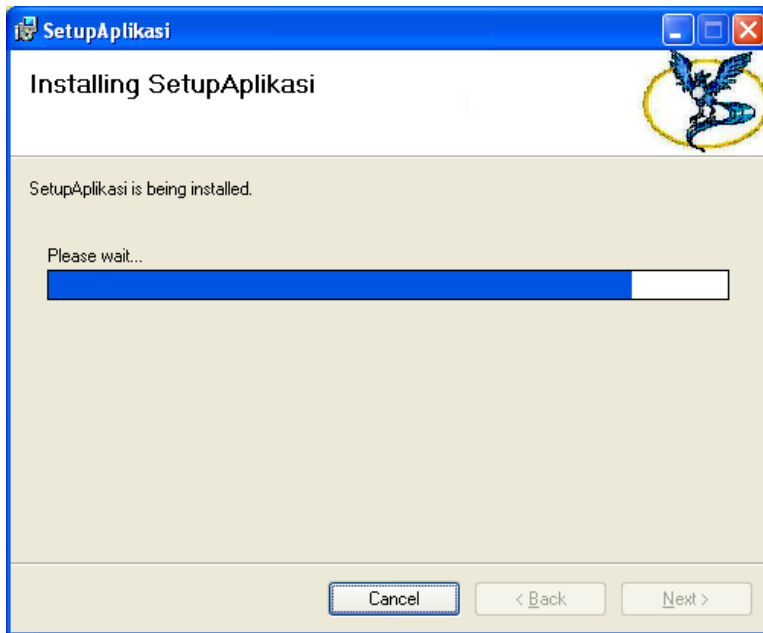
6. Buat file License.rtf dengan WordPad lalu masukkan pada DialogBox License Agreement seperti gambar berikut, caranya sama seperti saat anda menambahkan file readme:



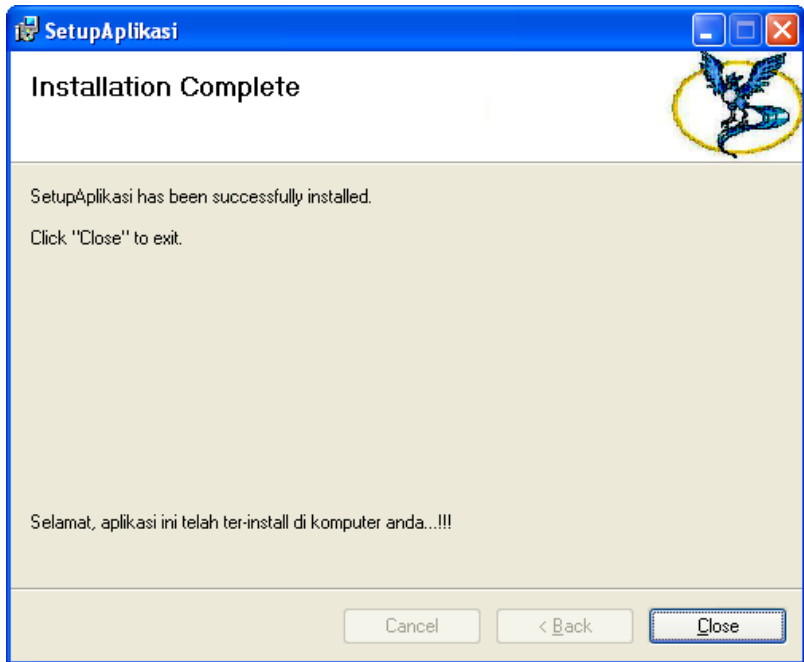
7. DialogBox Installation Folder tampak seperti gambar dibawah:



8. DialogBox Progress tampak seperti gambar di bawah:



10. Setelah proses instalasi selesai tampilan akan seperti di bawah ini :



11. Aturlah properties agar ketika di-install pada folder Program Files, aplikasi tadi mempunyai folder "L@bkom Soft". Anda atur properties pada project setup anda pada bagian manufacturer.

Setelah anda selesai melakukan setting pada instalation folder, build solution anda. Klik kanan project setup anda lalu pilih install untuk melihat hasil instalasi.