Pemrograman Visual

Sesi 01

Dosen Pembina : Danang Junaedi

Pendahuluan

Overview

- Deskripsi
- Tujuan Instruksional & Kaitan Materi
- Penilaian & Grade
- Referensi
- Tata Tertib Perkuliahan

Deskripsi

- Dalam mata kuliah ini dibahas tentang definisi Pemrograman berbasis tampilan Visual, Tahapan Pengembangan Pemrograman berbasis tampilan Visual, Sejarah Visual Basic, Lingkungan Visual Basic, Event Driven, Obyek dalam Visual Basic (*Control*, *Form* dan GUI lainnya), Penanganan *Error* (*Error Handling*), serta Pengolahan *database* standar dan OLE

Tujuan Instruksional & Kaitan Materi

- **Tujuan Umum**

  Setelah proses perkuliahan, mahasiswa memiliki kemampuan untuk

  - **tingkat pemahaman**

    menjelaskan konsep cara perancangan dan analisis masalah, yang kemudian dipecahkan dengan menggunakan teknik pemrograman berbasis tampilan visual.

  - **tingkat aplikasi**

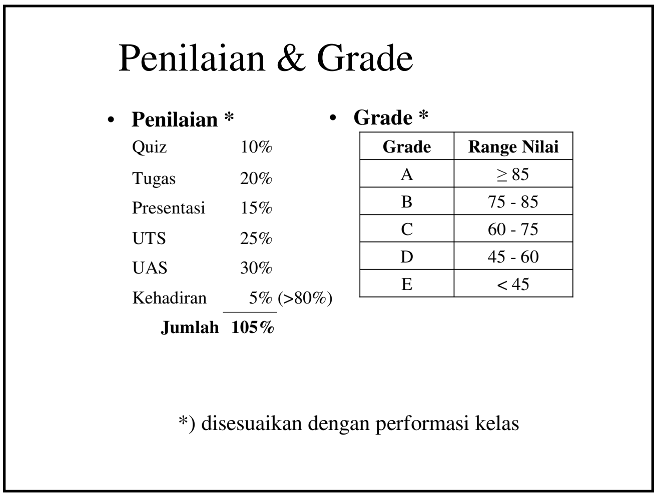
    mengimplementasikan konsep cara perancangan dan analisis masalah, yang kemudian dipecahkan dengan menggunakan teknik pemrograman berbasis tampilan visual dalam hal ini menggunakan bahasa pemrograman Visual Basic

- **Kaitan Materi**

  -

Urutan Pembahasan

Pertemuan	**Materi**
1	Pendahuluan (Susunan Materi, Aturan Perkuliahan, Aturan Penilaian, grade nilai, referensi), Intro
2 - 7	Dasar-dasar Pemrograman Visual; Presentasi(optional)
8	UTS
9	Pengelolaan File; Presentasi
10 - 13	Pengelolaan Database; Presentasi (optional)
14 -15	Class & Object; Interfacing & OLE Automation; Project
16	UAS



Penilaian & Grade

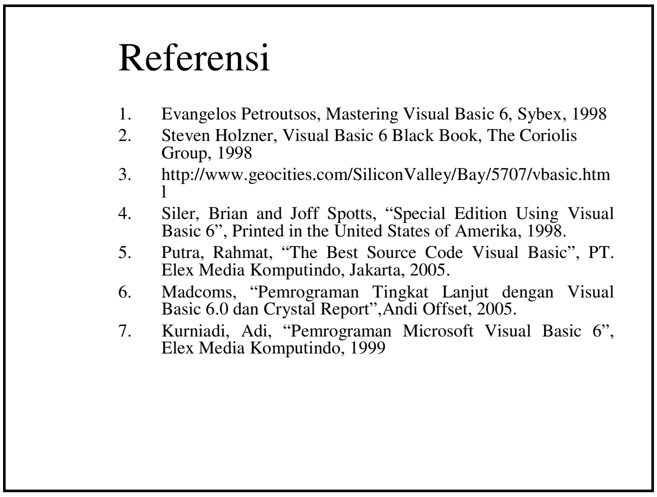
- **Penilaian ***

Quiz	10%
Tugas	20%
Presentasi	15%
UTS	25%
UAS	30%
Kehadiran	5% (>80%)
Jumlah	**105%**

- **Grade ***

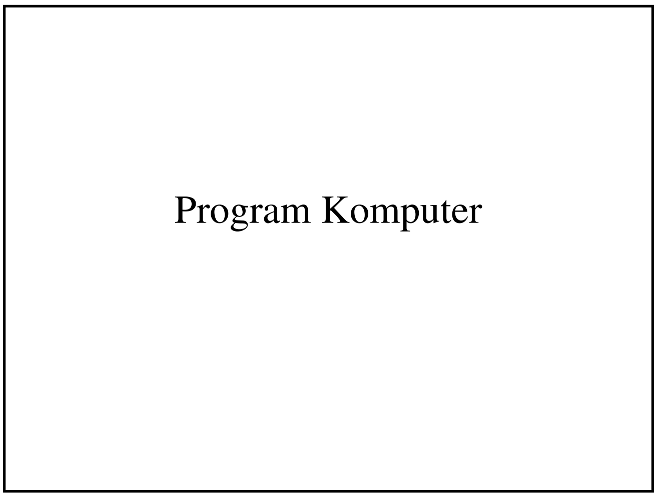
Grade	Range Nilai
A	≥ 85
B	75 - 85
C	60 - 75
D	45 - 60
E	< 45

*) disesuaikan dengan performasi kelas



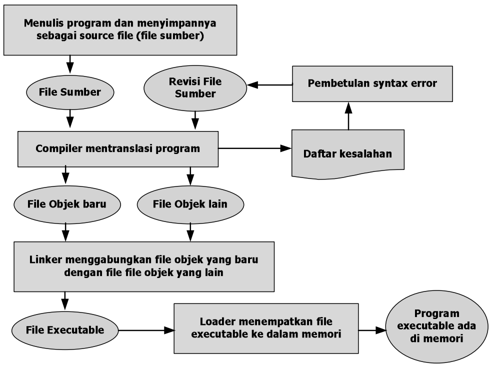
Referensi

1. Evangelos Petroutsos, Mastering Visual Basic 6, Sybex, 1998
2. Steven Holzner, Visual Basic 6 Black Book, The Coriolis Group, 1998
3. http://www.geocities.com/SiliconValley/Bay/5707/vbasic.html
4. Siler, Brian and Joff Spotts, "Special Edition Using Visual Basic 6", Printed in the United States of Amerika, 1998.
5. Putra, Rahmat, "The Best Source Code Visual Basic", PT. Elex Media Komputindo, Jakarta, 2005.
6. Madcoms, "Pemrograman Tingkat Lanjut dengan Visual Basic 6.0 dan Crystal Report",Andi Offset, 2005.
7. Kurniadi, Adi, "Pemrograman Microsoft Visual Basic 6", Elex Media Komputindo, 1999



Program Komputer

Prinsip Kerja


Menulis program dan menyimpannya sebagai source file (file sumber)
File Sumber
Revisi File Sumber
Pembetulan syntax error
Compiler mentranslasi program
Daftar kesalahan
File Objek baru
File Objek lain
Linker menggabungkan file objek yang baru dengan file file objek yang lain
File Executable
Loader menempatkan file executable ke dalam memori
Program executable ada di memori

Kelompok Bahasa Pemrograman

- ***Generasi bahasa pemrograman***:
  - Generasi I : *Machine Language*
  - Generasi II : *Assembly Llanguage* atau *Low Level Language* atau *symbolic Language* contohnya Assembler
  - Generasi III : *Middle Level Language*, contohnya C.
  - Generasi IV : *High Level Programming Language,* contohnya C, PASCAL, dsb.
  - Generasi IV : 4 GL (*fourth-generation language*): Prolog, SQL, Visual tool, dsb
- ***Metoda bahasa pemrograman*** :
  - Pemrograman tidak terstruktur, contohnya : : Basica, Fortran
  - Pemrograman terstruktur, contohnya : Pascal, C/C++
  - Pemrograman visual, contohnya : *VB, Delphi*
  - Pemrograman berorientasi objek, contohnya : C++, Java

Kelompok Bahasa Pemrograman

- ***Corak Kode*** :
  - Pemrograman prosedural, contohnya : Pascal, C
  - Pemrograman fungsional, contohnya Lisp
  - Pemrograman deklaratif, contohnya Prolog
  - Pemrograman berorientasi objek, contohnya : C++, Java
  - Pemrograman prosedural sekaligus berorientasi objek, contohnya : C++
- ***Tujuan (hasil)*** :
  - Pemrograman *stand-alone*
  - Pemrograman *client/serve*r
  - Pemrograman web : HTML, *Script*
  - Pemrograman jaringan
- ***Cara penterjemahan*** :
  - Interpreter : Basica, Foxpro, Matlab
  - Compiler : Turbo B*asic, Pascal, C/C++*

Kelompok Bahasa Pemrograman

- http://en.wikipedia.org (Array, Aspect-oriented, Assembly, Concatenative, Concurrent, Curly-Bracket, Data-structured, Dataflow, Declarative, Domain-specific, Dynamic, Educational, Esoteric, General Purpose, Imperative, Functional, Interface Description, Logic, Multiparadigm, Object Oriented, Protoype-based, Pattern directive invocation, Procedural, Quantum, Reflective, Scripting, Synchronous, Visual)
- **Bentuk Tampilan**
  - Berbasis Web (HTML, VBScript, ASP, PHP, CGI, Java Script)
  - Berbasis Visual (Visual Basic dkk, Power Builder, Delphi, Java)
  - Berbasis Teks (Clipper, Pascal, C/C++, Basic, Fortran, Cobol, assembly, Ada, Algol, Forth)
- Permograman **Berbasis Linux** (Perl, Phyton, Shell Programming, A+, Real Basic, KDevelop)

Jenis Program Komputer

- **Berdasarkan jumlah Pengguna**
  - *Multi User* : program yang digunakan oleh lebih dari satu orang pada saat yang bersamaan. Ex : DOS, Windows
  - Single User : program yang hanya digunakan oleh satu rang pada satu saat. Ex : Linux, UNIX
- **Berdasarkan Tampilan**
  - GUI (*Graphical User interface*) Ex :Visual Basic, delphi dll.
  - CUI (*Command User interface*) Ex : Pascal, C, Perl dll.
- **Berdasarkan Tempat Menjalankan Program**
  - 32 bit
  - 64 bit
- **Berdasarkan Konfigurasi**
  - *Stand Alone* : semua lapisan penyusun program terletak pada satu komputer
  - *Two Tier* (*Client Server*) : *Server* berfungsi sebagai penyedia informasi dan *Client* berfungsi melakukan penanganan data.
  - *Three Tier* : terdiri dari *Server* database, *Server* Program dan *Client*

Tahapan Pemrograman

- Mendefinisikan permasalahan
- Membuat rumusan untuk pemecahan masalah (algoritma)
- Membuat prototipe antarmuka
- Implementasi / *Coding* / Pembuatan Program
- Menguji hasil implementasi
- Membuat dokumentasi program

Common Programming Languages

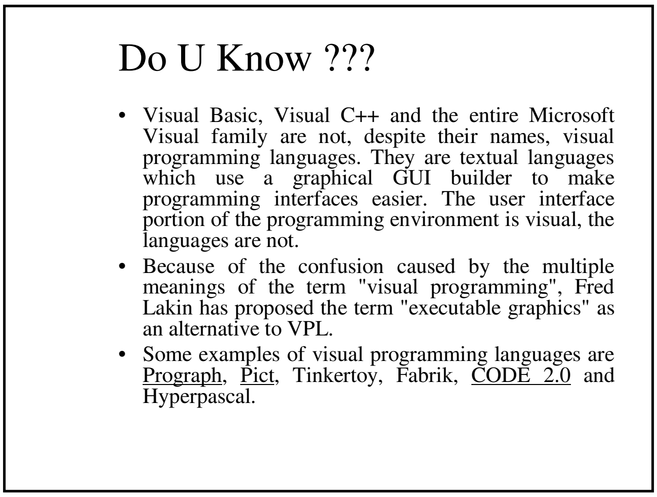
- BASIC
- FORTRAN
- COBOL
- Pascal
- C
- C++
- C#
- Java

- *Visual Basic is not just a programming language*
- *It's a programming environment with tools to:*
  - *Create screen elements*
  - *Write programming language statements*

Pemrograman Visual

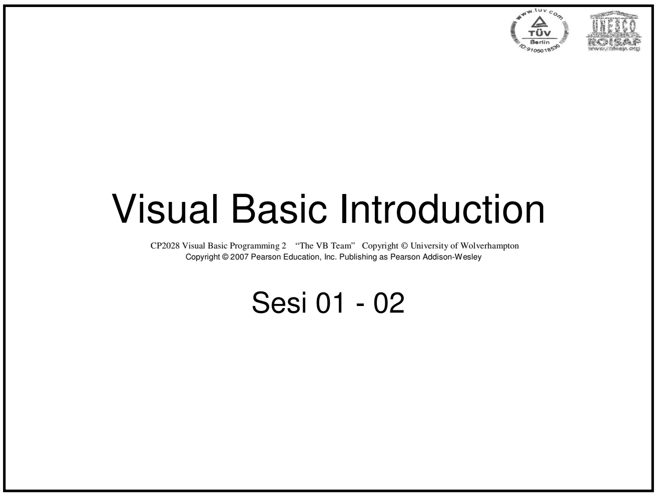
Definisi

- **Pemrograman visual** adalah metode pembuatan program dimana programmer membuat koneksi antara objek-objek dengan cara menggambar, menunjuk,, dan mengklik pada diagram dan ikon dan dengan berinteraksi dengan diagram alur.
- **Visual programming language** - (VPL) Any programming language that allows the user to specify a program in a two-(or more)-dimensionsional way. Conventional textual languages are not considered two-dimensional since the compiler or interpreter processes them as one-dimensional streams of characters. A VPL allows programming with visual expressions - spatial arrangements of textual and graphical symbols.
- VPLs may be further classified, according to the type and extent of visual expression used, into icon-based languages, form-based languages and diagram languages. Visual programming environments provide graphical or iconic elements which can be manipulated by the user in an interactive way according to some specific spatial grammar for program construction.



Do U Know ???

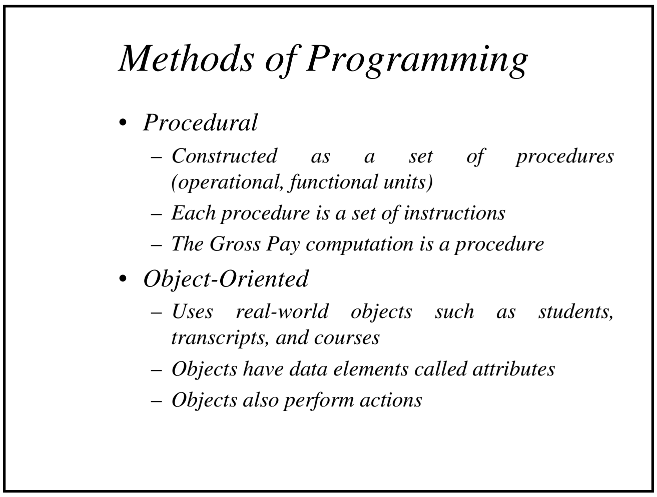
- Visual Basic, Visual C++ and the entire Microsoft Visual family are not, despite their names, visual programming languages. They are textual languages which use a graphical GUI builder to make programming interfaces easier. The user interface portion of the programming environment is visual, the languages are not.
- Because of the confusion caused by the multiple meanings of the term "visual programming", Fred Lakin has proposed the term "executable graphics" as an alternative to VPL.
- Some examples of visual programming languages are Prograph, Pict, Tinkertoy, Fabrik, CODE 2.0 and Hyperpascal.



Visual Basic Introduction

CP2028 Visual Basic Programming 2 "The VB Team" Copyright © University of Wolverhampton
Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

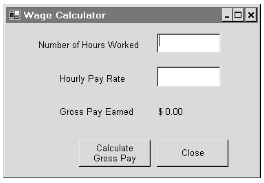
Sesi 01 - 02



Methods of Programming

- *Procedural*
  - *Constructed as a set of procedures (operational, functional units)*
  - *Each procedure is a set of instructions*
  - *The Gross Pay computation is a procedure*
- *Object-Oriented*
  - *Uses real-world objects such as students, transcripts, and courses*
  - *Objects have data elements called attributes*
  - *Objects also perform actions*

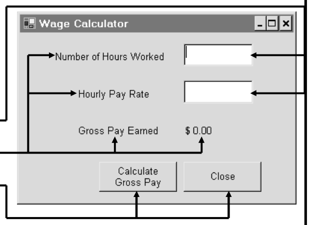
Example of an Object



- *This is a Visual Basic GUI object called a form*
- *Contains data and actions*
- *Data, such as Hourly Pay Rate, is a text property that determines the appearance of form objects*
- *Actions, such as Calculate Gross Pay, is a method that determines how the form reacts*
- *A form is an object that contains other objects such as buttons, text boxes, and labels*

Example of an Object

- *Form elements are objects called controls*
- *This form has:*
  - *Two TextBox controls*
  - *Four Label controls*
  - *Two Button controls*
- *The value displayed by a control is held in the text property of the control*
- *Left button text property is Calculate Gross Pay*
- *Buttons have methods attached to click events*



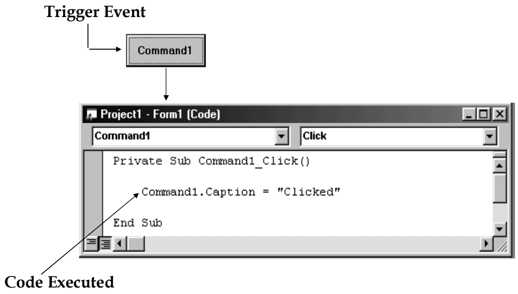
Methods of Programming (2)

- Pemrograman Visual
  - Penggunaan Ekspresi Visual (Seperti Gambar, Grafik atau ikon) dalam proses pemrograman
  - Mengacu pada aktivitas yang memungkinkan pengguna untuk membuat program dalam dua (atau lebih) dimensi

Event Driven Programming: Events

- *The GUI environment is event-driven*
- *An event is an action that takes place within a program*
  - *Clicking a button (a Click event)*
  - *Keying in a TextBox (a TextChanged event)*
- *Visual Basic controls are capable of detecting many, many events*
- *A program can respond to an event if the programmer writes an event procedure*

Visual Basic Event Processing



Trigger Event

```
Private Sub Command1_Click()

    Command1.Caption = "Clicked"

End Sub
```

Code Executed

Types of Events

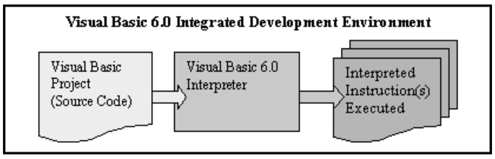
- **Events can be classified as:**
- **User generated**
  - **(e.g. command button click)** Command1
- **Computer generated**
  - **(e.g. specific time elapsed, from a timer control)**
- **Program generated**
  - **(i.e. program explicitly generates an event from within the code)**

Visual Basic Events

- ***The events that can happen to a control are pre-determined***
- ***Each type of control has a relevant set of events***
- ***The events that can happen to a Command Button***

Programmer’s Lingo

Programmer’s Lingo (1)


Visual Basic 6.0 Integrated Development Environment
Visual Basic Project (Source Code)
Visual Basic 6.0 Interpreter
Interpreted Instruction(s) Executed

Interpreter Process

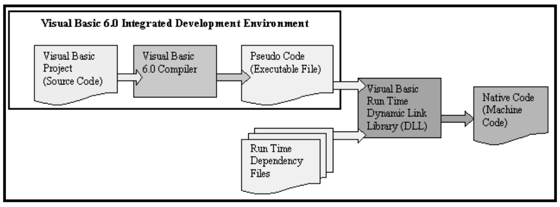
Programmer's Lingo (2)

Compiler *- translates source code into an executable file*

Executable file *- standalone file that runs outside the Visual Basic environment; Used to distribute most Windows applications*

Package and Deployment Wizard *creates all the files necessary to distribute your projects*

Programmer's Lingo (3)


Visual Basic 6.0 Integrated Development Environment
Visual Basic Project (Source Code)
Visual Basic 6.0 Compiler
Pseudo Code (Executable File)
Visual Basic Run Time Dynamic Link Library (DLL)
Native Code (Machine Code)
Run Time Dependency Files

Compiler Process

Programmer's Lingo (4)

Pseudo Code *(P-code) - intermediate step between source code and native code*

Native Code *(machine language) - instructions the computer can execute directly*

Dynamic Link Library *(DLL) - translates instructions in executable file at runtime*

Sejarah Visual Basic

Background of Visual Basic (1)

- *BASIC*
  - *Beginner's All-purpose Symbolic Instruction Code*
  - *By John Kemeny and Thomas Kurtz*
  - *Designed to teach programming to beginners*
  - *BASIC language interpreter, one of Microsoft's first products*

Background of Visual Basic (2)

- BASIC
  - Included with DOS
  - QBASIC - included with DOS Version 5 - subset of BASIC
- Visual Basic
  - 1992 - Visual Basic 1 for Windows 3
  - Visual Basic 4 - for Windows 95
  - Visual Basic 5 - for Office 97
  - Visual Basic 6 - for Windows 98 & Office 2000
  - Visual – net – beta version 2001****

Visual Basic Editions (1)

- *Learning Edition*
- *Professional Edition*
- *Enterprise Edition*
- *Included in Visual Studio suite (analogy Office for programmers)*

Visual Basic Editions (2)

- *Other Visual Basic packaging options*
  - *VBA (Visual Basic for Applications)*
  - *Windows CE Toolkit for Visual Basic*
  - *Visual Basic Scripting Edition*
  - *Control Creation Edition*

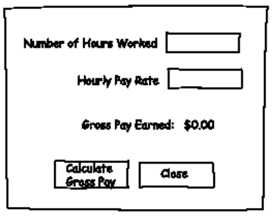
Tahapan Pembuatan Program

Step 1 of Developing an Application

- *Clearly define what the program is to do*
- *For example, the Wage Calculator program:*
  - *Purpose: To calculate the user’s gross pay*
  - *Input: Number of hours worked, hourly pay rate*
  - *Process: Multiply number of hours worked by hourly pay rate (result is the user’s gross pay)*
  - *Output: Display a message indicating the user’s gross pay*

Step 2 of Developing an Application

- *Visualize the application running on the computer and design its user interface*



Step 3 of Developing an Application

- *Make a list of the controls needed*

Type	***Name***	***Description***
TextBox	*txtHoursWorked*	*Allows the user to enter the number of hours worked.*
TextBox	*txtPayRate*	*Allows the user to enter the hourly pay rate*
Label	*lblGrossPay*	*Displays the gross pay, after the btnCalcGrossPay button has been clicked*
Button	*btnCalcGrossPay*	*When clicked, multiplies the number of hours worked by the hourly pay rate*
Button	*btnClose*	*When clicked, terminates the application*
Label	*(default)*	*Description for Number of Hours Worked TextBox*
Label	*(default)*	*Description for Hourly Pay Rate TextBox*
Label	*(default)*	*Description for Gross Pay Earned Label*
Form	*(default)*	*A form to hold these controls*

Step 4 of Developing an Application

- *Define values for each control's relevant properties:*

Control Type	*Control Name*	*Text*
Form	*(Default)*	*"Wage Calculator"*
Label	*(Default)*	*"Number of Hours Worked"*
Label	*(Default)*	*"Hourly Pay Rate"*
Label	*(Default)*	*"Gross Pay Earned"*
Label	*lblGrossPay*	*"$0.00"*
TextBox	*txtHoursWorked*	*""*
TextBox	*txtPayRate*	*""*
Button	*btnCalcGrossPay*	*"Calculate Gross Pay"*
Button	*btnClose*	*"Close"*

Step 5 of Developing an Application

- *List the methods needed for each control:*

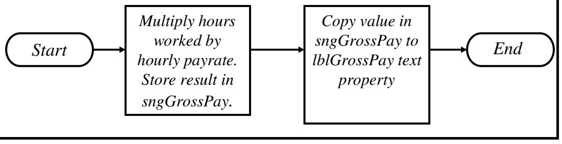
Method	*Description*
btnCalcGrossPay_Click	*Multiplies hours worked by hourly pay rate. These values are entered into the txtHoursWorked and txtPayRate TextBoxes Result is stored in lblGrossPay Text property*
btnClose_Click	*Terminates the application*

Step 6 of Developing an Application

- *Create pseudocode **or** a flowchart of each method:*
  - *Pseudocode is an English-like description in programming language terms*

```
Store Hours Worked x Hourly Pay Rate in
sngGrossPay.
Store the value of sngGrossPay in
lblGrossPay.Text.
```

  - *A flowchart is a diagram that uses boxes and other symbols to represent each step*



Start

Multiply hours worked by hourly payrate. Store result in sngGrossPay.

Copy value in sngGrossPay to lblGrossPay text property

End

Step 7 of Developing an Application

- *Check the code for errors:*
  - *Read the flowchart and/or pseudocode*
  - *Step through each operation as though **you** are the computer*
  - *Use a piece of paper to jot down the values of variables and properties as they change*
  - *Verify that the expected results are achieved*

Step 8 of Developing an Application

- *Use Visual Basic to create the forms and other controls identified in step 3*
  - *This is the first use of Visual Basic, all of the previous steps have just been on paper*
  - *In this step you develop the portion of the application the user will see*

Step 9 of Developing an Application

- *Use Visual Basic to write the code for the event procedures and other methods created in step 6*
  - *This is the second step on the computer*
  - *In this step you develop the methods behind the click event for each button*
  - *Unlike the form developed on step 8, this portion of the application is invisible to the user*

Step 10 of Developing an Application

- *Attempt to run the application - find syntax errors*
  - *Correct any syntax errors found*
  - *Syntax errors are the incorrect use of an element of the programming language*
  - *Repeat this step as many times as needed*
  - *All syntax errors must be removed before Visual Basic will create a program that actually runs*

Step 11 of Developing an Application

- *Run the application using test data as input*
  - *Run the program with a variety of test data*
  - *Check the results to be sure that they are correct*
  - *Incorrect results are referred to as a runtime error*
  - *Correct any runtime errors found*
  - *Repeat this step as many times as necessary*

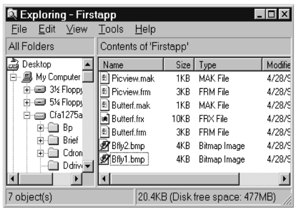
Visual Basic IDE *(Integrated Development Environment)*

The Visual Studio IDE

- *Visual Studio is an integrated development environment, often abbreviated as IDE*
- *Provides everything needed to create, test, and debug software including:*
  - *The Visual Basic language*
  - *Form design tools to create the user interface*
  - *Debugging tools to help find and correct programming errors*
- *Visual Studio supports other languages beside Visual Basic such as C++ and C#*

Visual Basic Program Structure.

- Project File
  - '.VBP'
- Form Files
  - '.FRM'
  - '.FRX'
- Modules
  - '.BAS'
- Custom Controls
  - '.VBX' files

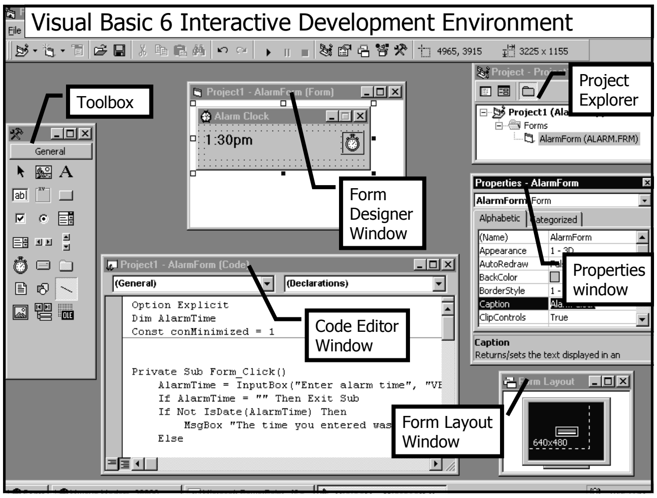


Contents of 'Firstapp'

Name	Size	Type	Modifie
Picview.mak	1KB	MAK File	4/28/9
Picview.frm	3KB	FRM File	4/28/9
Butterf.mak	1KB	MAK File	4/28/9
Butterf.frx	10KB	FRX File	4/28/9
Butterf.frm	3KB	FRM File	4/28/9
Bfly2.bmp	4KB	Bitmap Image	4/28/9
Bfly1.bmp	4KB	Bitmap Image	4/28/9

7 object(s) 20.4KB (Disk free space: 477MB)

Visual Basic 6 Interactive Development Environment



```
Option Explicit
Dim AlarmTime
Const conMinimized = 1

Private Sub Form_Click()
    AlarmTime = InputBox("Enter alarm time", "VB
    If AlarmTime = "" Then Exit Sub
    If Not IsDate(AlarmTime) Then
        MsgBox "The time you entered was
    Else
```

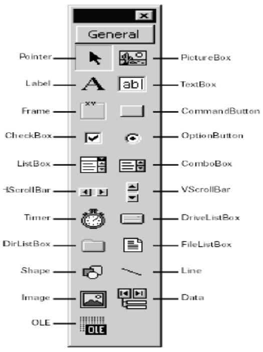
(Name)	AlarmForm
Appearance	1 - 3D
AutoRedraw	Fal
BackColor	
BorderStyle	1 -
Caption	Ala
ClipControls	True

Caption
Returns/sets the text displayed in an

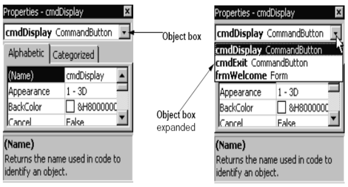
Elements of the Integrated Development Environment

- *Application icon*
- *Application name*
- *Context menus*
- *Controls*
- *Current project*
- *Design view mode*
- *Title bar*
- *Major windows*
  - *Form Designer*
  - *Form Layout*
  - *Toolbox*
  - *Project Explorer*
  - *Object Browser*
  - *Properties*
  - *Code Editor*
  - *Immediate, Locals, Watch*

Toolbox Window



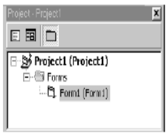
Properties Window



(Name)	cmdDisplay
Appearance	1 - 3D
BackColor	8H8000000C
Cancel	False

(Name)
Returns the name used in code to identify an object.

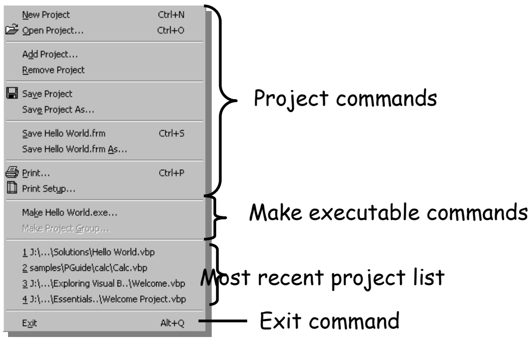
Project Explorer Window



Perhatikan Icon-icon berikut :

Icon	Fungsi
▤	View Code
▤	View Object
□	Toggle Forder

The File Menu


New Project Ctrl+N
Open Project... Ctrl+O
Add Project...
Remove Project
Save Project
Save Project As...
Save Hello World.frm Ctrl+S
Save Hello World.frm As...
Print... Ctrl+P
Print Setup...
Make Hello World.exe...
Make Project Group...
1 J:\...\Solutions\Hello World.vbp
2 samples\PGuide\calc\Calc.vbp
3 J:\...\Exploring Visual B..\Welcome.vbp
4 J:\...\Essentials..\Welcome Project.vbp
Exit Alt+Q

Project commands

Make executable commands

Most recent project list

Exit command

The File Menu

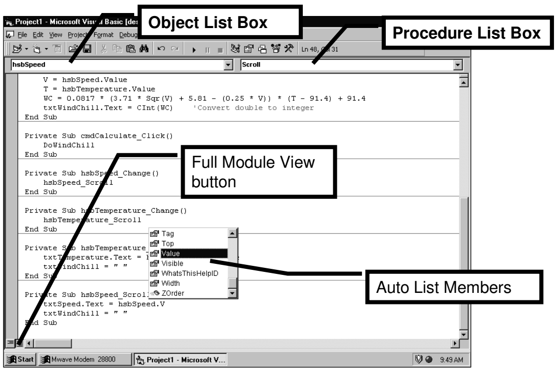
Template - *particular pattern for a project or a project element; start a project from a template saves time*

Existing tab *- displays VB projects*

Recent tab *- displays most recently accessed projects*

Project Groups *- a collection of several projects*

The Code Editor Window (1)



Object List Box

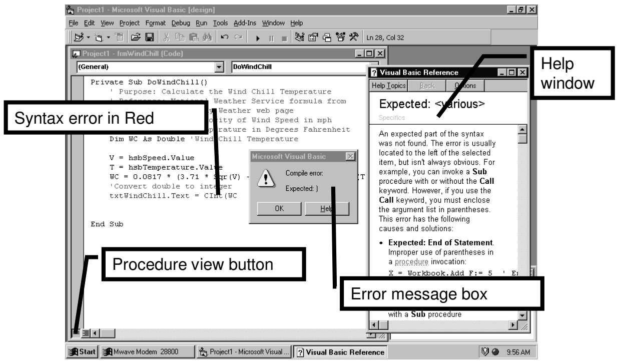
Procedure List Box

Full Module View button

Auto List Members

```
    V = hsbSpeed.Value
    T = hsbTemperature.Value
    WC = 0.0817 * (3.71 * Sqr(V) + 5.81 - (0.25 * V)) * (T - 91.4) + 91.4
    txtWindChill.Text = CInt(WC)     'Convert double to integer
End Sub

Private Sub cmdCalculate_Click()
    DoWindChill
End Sub

Private Sub hsbSpeed_Change()
    hsbSpeed_Scroll
End Sub

Private Sub hsbTemperature_Change()
    hsbTemperature_Scroll
End Sub

Private Sub hsbTemperature_
    txtTemperature.Text = 
    txtWindChill = " "
End Sub

Private Sub hsbSpeed_Scroll
    txtSpeed.Text = hsbSpeed.V
    txtWindChill = " "
End Sub
```

The Code Editor Window (2)



Help window

Syntax error in Red

Procedure view button

Error message box

```
Private Sub DoWindChill()
    ' Purpose: Calculate the Wind Chill Temperature
    ' Reference: National Weather Service formula from
                    Weather web page
                 ocity of Wind Speed in mph
                 perature in Degrees Fahrenheit
    Dim WC As Double 'Wind Chill Temperature

    V = hsbSpeed.Value
    T = hsbTemperature.Value
    WC = 0.0817 * (3.71 * Sqr(V) -
    'Convert double to integer
    txtWindChill.Text = CInt(WC

End Sub
```

Microsoft Visual Basic

Compile error:

Expected:)

Visual Basic Reference

Expected: <various>

An expected part of the syntax was not found. The error is usually located to the left of the selected item, but isn't always obvious. For example, you can invoke a **Sub** procedure with or without the **Call** keyword. However, if you use the **Call** keyword, you must enclose the argument list in parentheses. This error has the following causes and solutions:

- **Expected: End of Statement**. Improper use of parentheses in a procedure invocation:

  `X = Workbook.Add F:= 5  ' Er`

  with a **Sub** procedure

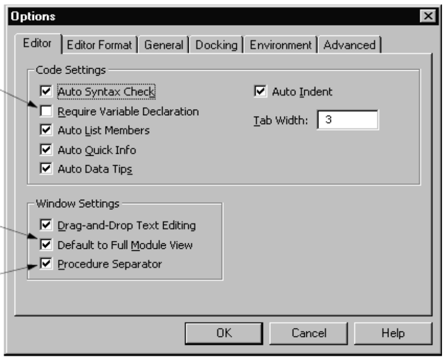
Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Tools Add-Ins Window Help
Ln 28, Col 32
Project1 - frmWindChill (Code)
(General)
DoWindChill
OK
Help
Help Topics
Back
Options
Specifics
Start
Mwave Modem 28800
Project1 - Microsoft Visual ...
Visual Basic Reference
9:56 AM

Option Window

Forces explicit declarations by adding the line **`Option Explicit`** to the general declaration (when checked)

Full Module View is the default when checked

When checked, displays **Procedure Separator** lines in **Full Module View**



Options

Editor | Editor Format | General | Docking | Environment | Advanced

Code Settings

- ☑ Auto Syntax Check
- □ Require Variable Declaration
- ☑ Auto List Members
- ☑ Auto Quick Info
- ☑ Auto Data Tips
- ☑ Auto Indent
- Tab Width: 3

Window Settings

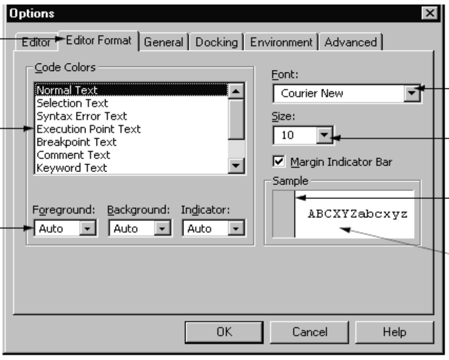
- ☑ Drag-and-Drop Text Editing
- ☑ Default to Full Module View
- ☑ Procedure Separator

Mengatur Jendela Editor

Editor Format tab

Element that is color highlighted

Color choices



Options

Editor | Editor Format | General | Docking | Environment | Advanced

Code Colors

- Normal Text
- Selection Text
- Syntax Error Text
- Execution Point Text
- Breakpoint Text
- Comment Text
- Keyword Text

Foreground: Auto   Background: Auto   Indicator: Auto

Font: Courier New

Size: 10

☑ Margin Indicator Bar

Sample: ABCXYZabcxyz

Font choices

Font size choices

Margin Indicator bar

Sample area

Using the Code Editor

GUI (Graphical User Interface) *- forms and windows that the user sees*

Property *- a characteristic or attribute of an object such as color and size*

Event *- a user action such as clicking a button*

Code Editor window *- editor specially designed to help you when writing code*

The Code Editor Window

- *View and edit code*
- *Views*
  - *Full Module View*
  - *Procedure View*
- *Features*
  - *Auto List Members*
  - *Auto Quick Info*
  - *Auto Syntax Check*

Syntax Boxes

- *Syntax for a Visual Basic statement is shown in a syntax box*
- *Reserved words are shown in **bold***
- *Programmer named words are shown in italics*

Testing & Debugging
Documenting Project
Distributing Project

Testing and Debugging

- *Humans are not perfect*
- *Rarely code programs right the first time*
- *Test to find problems*
- *Debug to correct problems*
- *First bug - Grace Hopper story*

Documenting your Project

- *Insert Banner Comments in your code*
  - *Comment - coding statement used by humans not the compiler or interpreter*
  - *Each organization may have different requirements, check with your instructor*
  - *Explain purpose of each major section*
  - *Example of Banner Comments*

```
`*************************************
` Programmer: Bracha Shapira
` Date:     June 30, 1998
` Source:           Hello World.vbp
` ID:               Chapter 2 Hands-On 1
` Purpose:     Practice steps to create
`              a Visual Basic project
` Inputs:      None
` Returns:     None
`***************************************
```

Documenting Your Project

- *Follow Code Conventions, and Print your Code*
  - *Conventions make your code easier to read and understand*
  - *Use block indenting*
  - *Print code and put in documentation folder*

Distributing Your Project

- *Remove all bugs first*
- *Create an executable file*
  - *Pull down File menu, and select Make Project exe command or*
  - *Use Package and Deployment Wizard (previously called Setup Wizard)*

Language Elements
Modules
Procedures

Language Elements

- *Keywords: Words with special meaning to Visual Basic (e.g.,* **`Private`***,* **`Sub`***)*
- *Programmer-defined-names: Names created by the programmer (e.g.,* **`sngGrossPay, btnClose`***)*
- *Operators: Special symbols to perform common operations (e.g., +, -, *, and /)*
- *Remarks: Comments inserted by the programmer – these are ignored when the program runs (e.g., any text preceded by a single quote)*
- *Syntax*
  - *defines the correct use of key words, operators, & programmer-defined names.*
  - *Similar to the syntax (rules) of English that defines correct use of nouns, verbs, etc.*
  - *A program that violates the rules of syntax will not run until corrected*

Five main types of code in Visual Basic (1)

- ***Procedure Calls:*** *You can call a procedure that you created, or you can call another control's event. Procedures include functions (return a value) and subs (don't return a value).*
- ***Compiler Functions:*** *If you know another form of BASIC then you are undoubtedly familar with it's simple functions like Chr$(), Asc(), and Ucase$(). These functions are like user-defined functions and always accept arguments and return values.*
- ***Keywords:*** *Like compiler functions, all versions of BASIC have built in keywords, or reserved words, like For, Next, Do, Loop, Select Case, Error, Goto, and Gosub. These appear in dark blue by default in the Visual Basic IDE.*

Five main types of code in Visual Basic (2)

- ***Control Properties****: You can set control properties that can be set at run time from within the code. For example, you could set the text property of a text box by typing txtInput.Text = "This is in the textbox", assuming that the na me of the textbox is txtInput. Notice the period. All properties are set after the control name with a seperating period. There are three times you will use periods after a word in Visual Basic: as a property, as a method, and as the member of a user-d efined type.*
- ***Methods:*** *Many objects in Visual Basic contain unique functions that can be performed on them. A function unique to an object is called a method. The method function is called after the name of the object and is set in dark blue by default in th e Visual Basic IDE. An example of a method might look like imgPicturebox.Zorder, which will set the image control to the front of the screen, assuming that the image control's name is imgPicturebox.*

Modules

Code *- the man behind the curtain*

Modules *- large units of code that comprise a Visual Basic application*

Form Modules *- contains a form and code for the form*

Standard Modules *- contains code that is typically used by other modules*

Modules Types (1)

- ***Base Code Modules:*** *To create a base code module, just click the second icon from the left on the toolbar. Base code modules do not belong to any form, and have a .bas extension. I n a base code module you define variables, types, and constants which are to be read from all other forms in the program. To declare a variable in a base code module, use the Global keyword instead of Dim. This will make the variable a global variable, which can be access from* ***anywhere*** *in the program.*

- ***Form Modules:*** *Like base code modules have definition blocks and procedures, forms also contain these. However, unlike in base code modules, Global variables cannot be defined here. Variables declared in the Definitions block of a form are local to that form with respect to the whole program (and global to that form period, but let's not get picky). This means that any variable defined here can be accessed by any function, procedure, or event from the en tire form. This is very usefull for cutting down on overhead usage. These variables do not take up as much overhead because they are allocated in memory only when the form is loaded.*

Modules Types (2)

If a form is unloaded, they loose their value. Furthermore, variable s defined in one form's Definitions block, those variables are not visible by other forms. You define procedures the same way as in a base code module, only as usual, they are only visible to that form.

- ***Local Blocks:*** *Local blocks include procedures (Functions and Subs) and events (which is really a procedure). Variables defined within a procedure are local to that procedure, meaning that they cannot be used by any other procedure besides the one they are defined in. This means that one procedure could define a variable, and another procedure could define a variable of the same name, and they would use them sperately. The only way a variable defined in one procedure can be accessed by other proc edures is if you pass it.*

Procedures

Procedures *- smaller units that comprise a module*

Event procedure *- automatically invoked by an event*

General procedure *- explicitly invoked by another procedure*

Private procedure *- accessible from within a module*

Public procedure *- accessible from anywhere*