

WinForms Framework dan Basic Controls

Pengembangan Aplikasi Berbasis Desktop

UPH Medan Campus

Adi

Overview

- Aplikasi desktop pada windows dapat dibangun dengan menggunakan framework WinForms yang merupakan bagian dari .NET Framework. Framework ini memiliki koleksi component/controls yang berbasis Graphical User Interface (GUI). Pengguna dapat berinteraksi dengan sekumpulan component/controls visual (seperti: button, textbox, toolbar, menu, dll) untuk membuat aplikasi yang dapat bekerja sesuai dengan persyaratan dan kebutuhan pengguna.

Objectives

Setelah menyelesaikan Unit ini, mahasiswa diharapkan mampu:

- Mengetahui langkah-langkah pengembangan suatu aplikasi desktop yang berbasis winforms framework
- Membedakan Object, Property, Method, dan Event
- Memahami dasar-dasar pengoperasian Microsoft Visual Studio 2022
- Memahami jenis-jenis kontrol dan cara penamaan pada sebuah kontrol
- Memahami karakteristik dan penggunaan dari form
- Memahami karakteristik dan penggunaan dari kontrol Button, Label dan TextBox

Contents

- Langkah-langkah pengembangan aplikasi desktop dengan framework winforms.
- Object, Property, Method & Event.
- IDE Microsoft Visual Studio 2022
- Mengenal Kontrol
- Form
- Button, Label, dan TextBox

WinForms ?

Windows Forms

- Framework yang dapat digunakan untuk membangun aplikasi **smart clients**.

*smart clients ?

- *Smart clients* are **graphically rich applications** that are easy to deploy and update, can work when they are connected to or disconnected from the Internet, and can access resources on the local computer in a more secure manner than traditional Windows-based applications

Karakteristik:

- Graphical User Interface
 - interactive user interface
- Mudah dalam development dan deployment.
- Bisa running baik dalam lingkungan yang terkoneksi atau tidak terkoneksi dari dunia internet.
- Event based.
- Tersedia banyak component / controls siap pakai
 - Plus banyak component/controls thirdparty

Langkah-Langkah Pengembangan Aplikasi Desktop

1. Merancang user interface dengan memanfaatkan component/controls yang ada pada form.
2. Mengatur property yang dibutuhkan pada masing-masing component/controls.
3. Menulis kode program berdasarkan event yang diperlukan pada masing-masing component/controls.

Object, property, method dan event

- Aplikasi yang dibangun dengan winforms, bersifat event-driven, yang berarti aplikasi tidak akan melakukan apapun sampai aplikasi tersebut dipanggil untuk merespon suatu event (*event-handler) (misal: menekan tombol, mengisi kotak isian textbox, memilih menu, dan lain sebagainya)

Project

Aplikasi (project) terdiri dari :

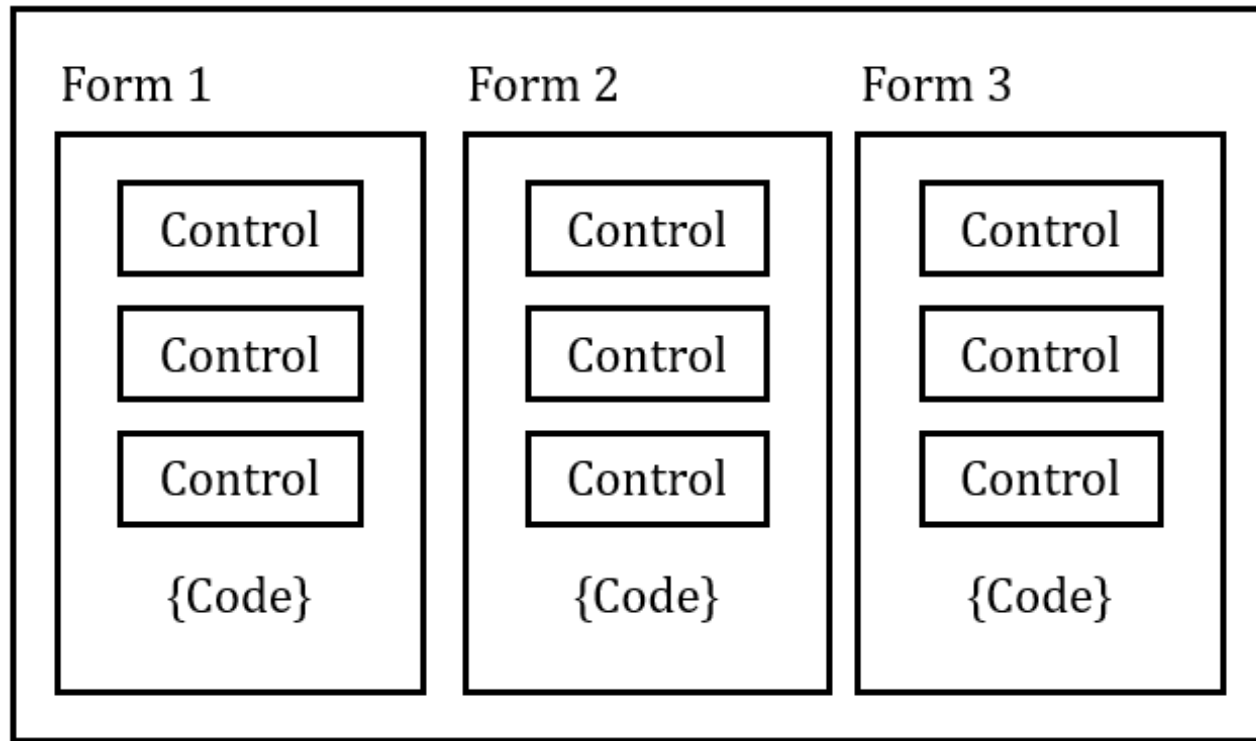
- Objects
 - Forms : windows yang dirancang sebagai antarmuka aplikasi
 - Controls : fitur grafis yang dimasukkan ke dalam form untuk memungkinkan interaksi dengan pengguna (misal: textbox, label, button, dan lain-lain)
- Properties
 - Setiap karakteristik dari form atau controls.
 - Contohnya: name, text, size, color, dll.
 - Setiap controls memiliki nilai default.
 - Nilai pada property dapat diganti Ketika mendesain aplikasi ataupun dengan coding.

Project (Cont.)

- Methods
 - metode built-in yang dapat dipanggil untuk memberikan beberapa tindakan ke kontrol atau objek tertentu
- Events
 - tergantung pada object atau control. Kode akan dieksekusi ketika event tertentu terjadi.

Project (Cont.)

Project



Project (Cont.)

- **Object** (Objek): Komponen di dalam sebuah program.
- **Property** (Properti): sifat atau karakteristik yang dimiliki objek.
- **Method** (Metoda): aksi, tindakan, atau perbuatan yang dapat dilakukan oleh objek.
- **Event** (Kejadian): Kejadian yang dapat diterima atau dialami oleh objek (biasanya dilakukan oleh user terhadap object).

Object: Mobil

Method:

- Maju
- Mundur
- Berhenti



Property:

- Merek: Toyota
- Type: Sedan
- Warna: Hijau

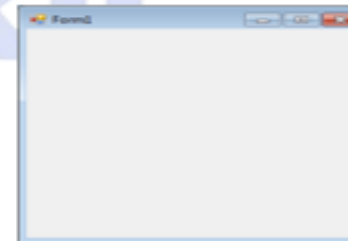
Event:

- Pintu dibuka
- Ditabrak
- Didorong

Object: Form

Method:

- Show
- Hide
- Close
- Move



Property:

- Name
- BackColor
- FormBorderStyle
- Text

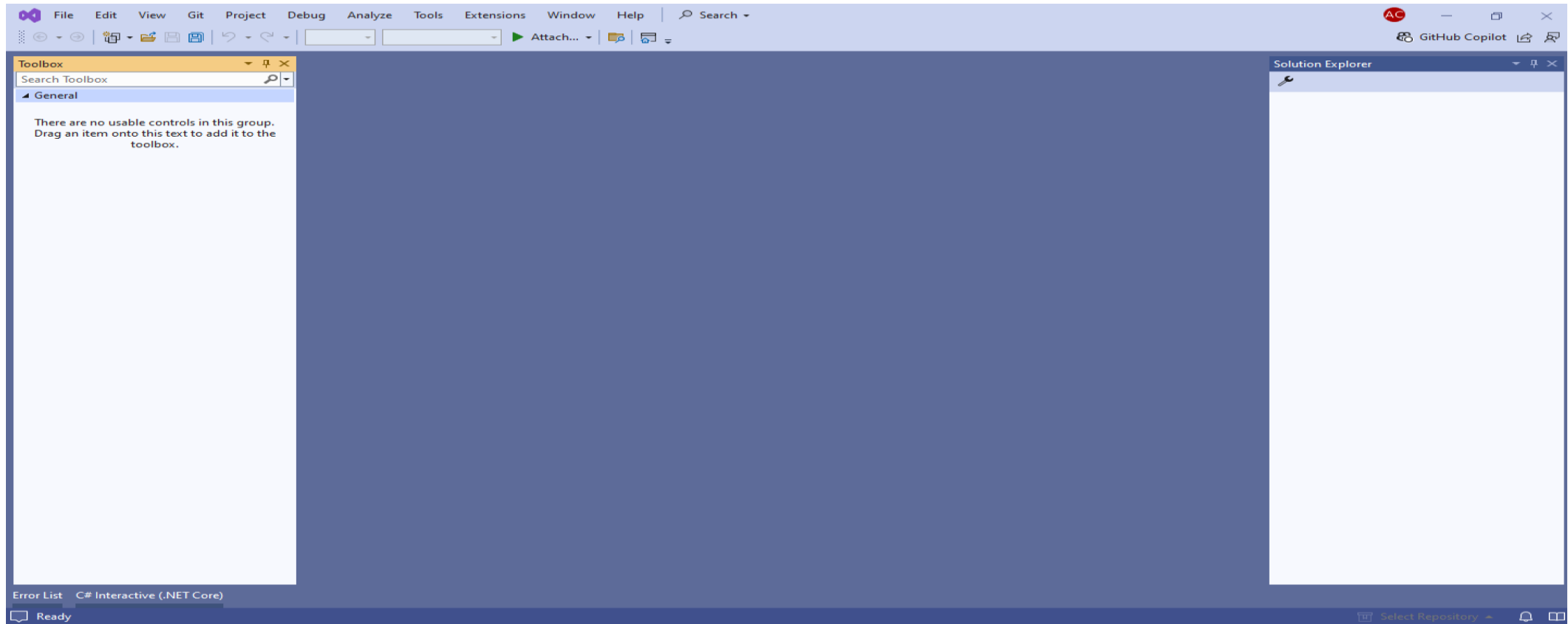
Event:

- Load
- Activated
- Click

IDE Microsoft Visual Studio 2022

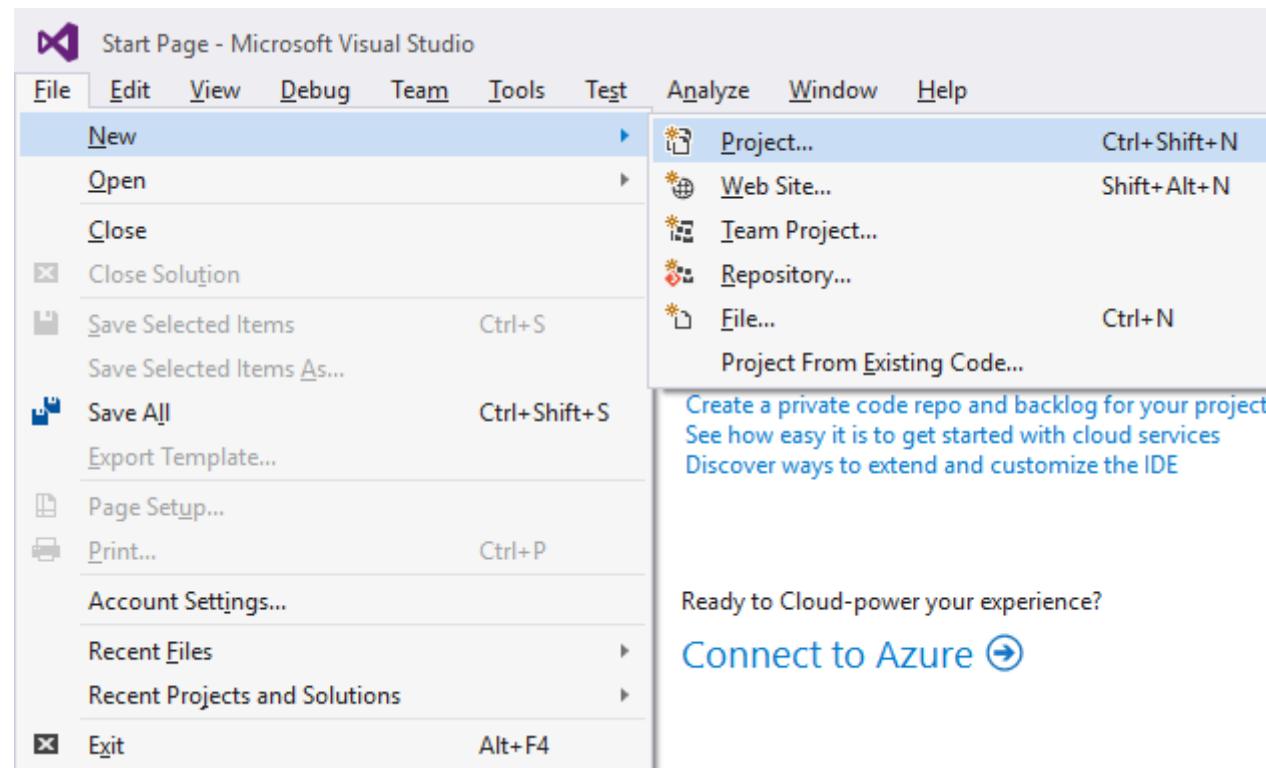
- IDE
 - Integrated Development Environment.
- Bacaan :
- https://en.wikipedia.org/wiki/Integrated_development_environment

Tampilan awal visual studio 2022



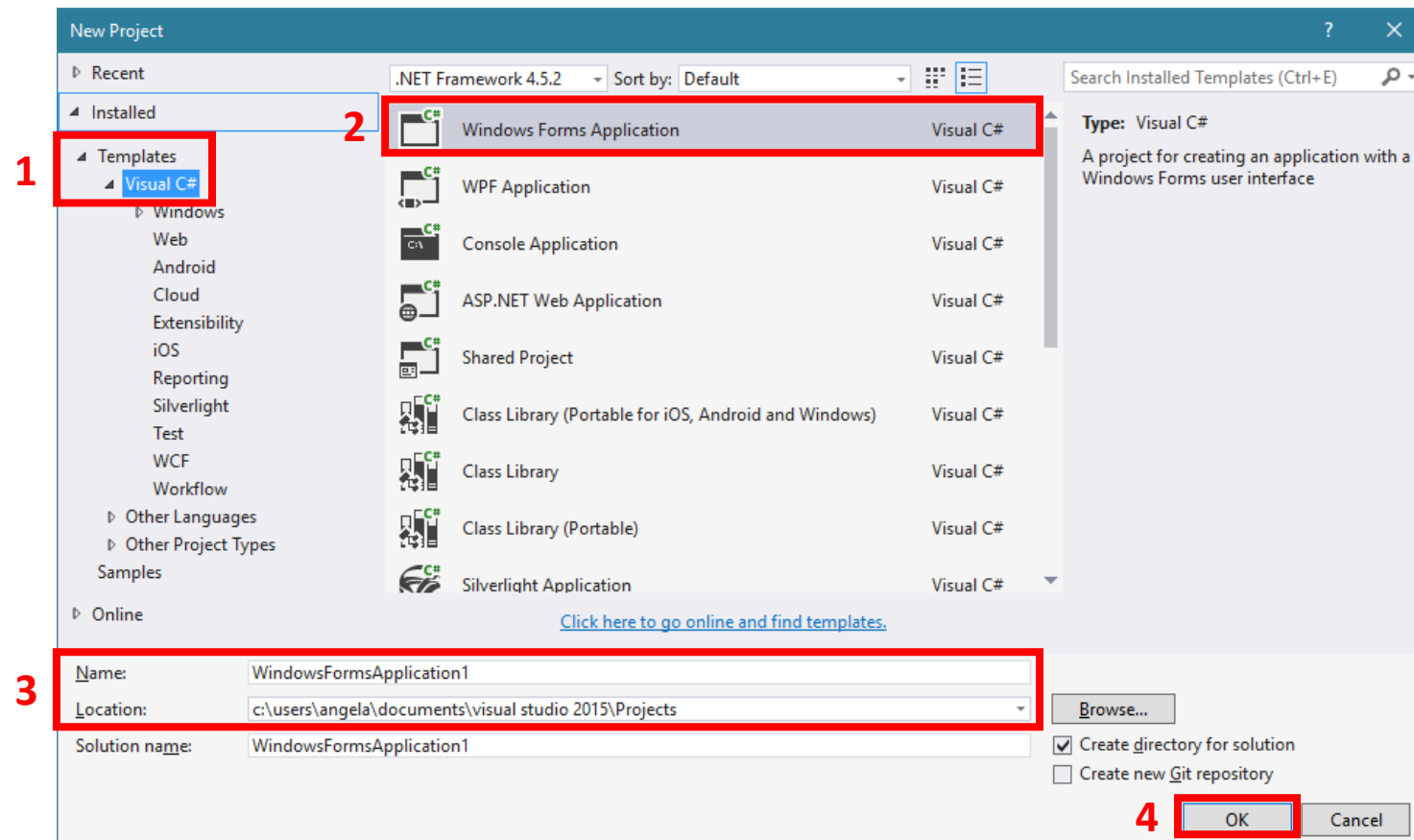
Membuat Project Baru

1. Untuk membuat project baru, kita dapat menekan tombol New Project... yang berada di sebelah kiri Start Page atau dengan memilih menu **File** → **New** → **Project (Ctrl + Shift + N)** untuk menampilkan kotak dialog new project



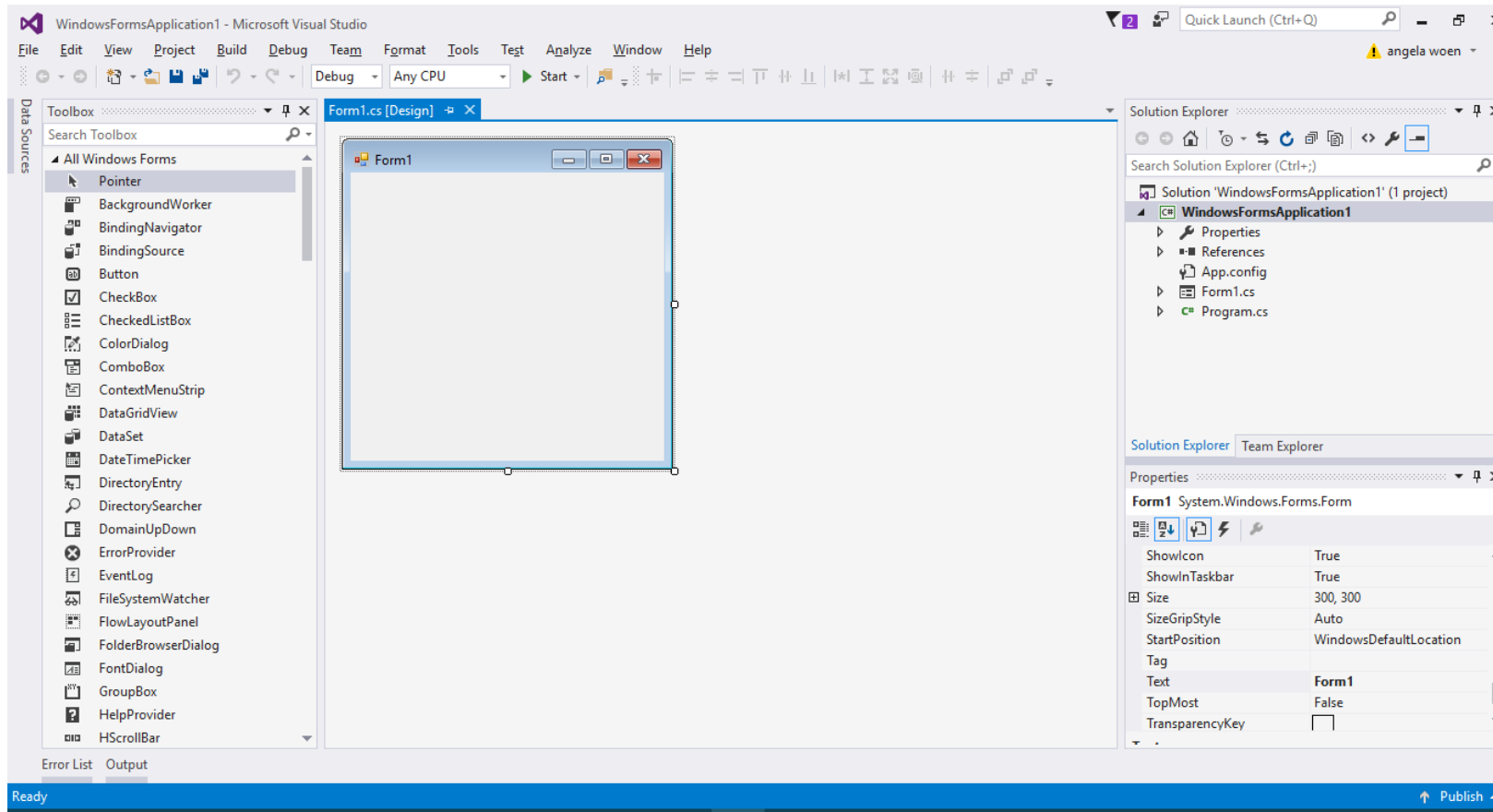
Membuat Project Baru (Cont.)

2. Pilih template **Visual C# → Windows Forms Application**. Pada kotak input **Name**, ketikkan nama project. Masukkan tempat penyimpanan project pada **Location** dengan menekan tombol **Browse**. Klik OK.

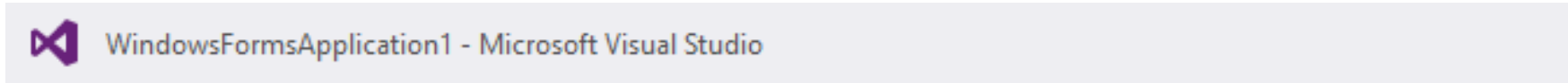


Membuat Project Baru (Cont.)

3. IDE Microsoft Visual Studio akan menampilkan template aplikasi windows form.



Title Bar



- Area Title Bar menunjukkan nama dari project. Ketika project dijalankan, kita dapat melihat tulisan (Running) atau (Debugging) yang menunjukkan mode operasi Visual C#. Visual C# beroperasi dalam 3 mode:
 - **Design mode** : digunakan untuk membangun aplikasi (mode ketika tidak sedang menjalankan aplikasi)
 - **Running mode** : digunakan untuk menjalankan aplikasi
 - **Debugging mode** : sering juga disebut Break mode, yaitu kondisi ketika aplikasi berhenti pada saat dijalankan karena terdapat debugger (biasa digunakan untuk mencari kesalahan pada kode program)

Menu Bar

File Edit View Project Build Debug Team Format Tools Test Analyze Window Help

Menu	Fungsi
File	Digunakan untuk membuka / menutup project dan file, juga untuk keluar dari Visual Studio
Edit	Digunakan ketika menulis kode program untuk melakukan pengeditan seperti copy, cut, paste, dan menghapus teks
View	Menyediakan akses ke berbagai windows dalam IDE
Project	Digunakan untuk menambah / menghapus komponen dari project
Build	Mengontrol proses kompilasi
Debug	Membantu melacak kesalahan / error pada kode program
Team	Digunakan ketika beberapa orang bekerja bersama pada 1 project
Format	Digunakan untuk memodifikasi tampilan pada graphic interface

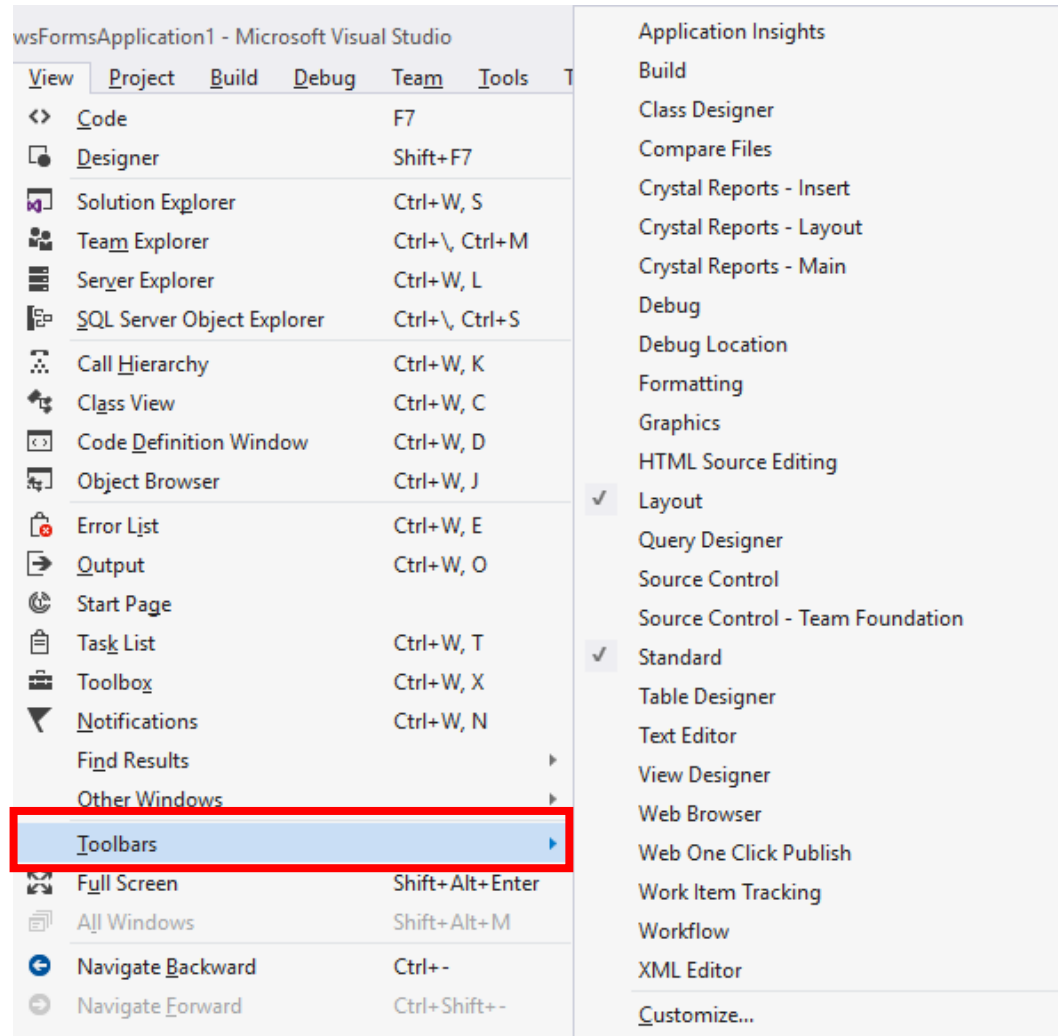
Menu Bar (Cont.)

File Edit View Project Build Debug Team Format Tools Test Analyze Window Help

Menu	Fungsi
Tools	Memungkinkan konfigurasi kustom terhadap IDE. Submenu Options paling banyak dipakai untuk memodifikasi IDE sesuai dengan kebutuhan personal masing-masing orang.
Test	Digunakan untuk mengkompilasi dan menjalankan aplikasi yang lengkap
Analyze	Melakukan analisis terhadap kode program
Window	Digunakan untuk mengubah layout windows dalam IDE
Help	Menyediakan akses ke dokumentasi online Visual C# melalui konten bantuan, indeks ataupun pencarian.

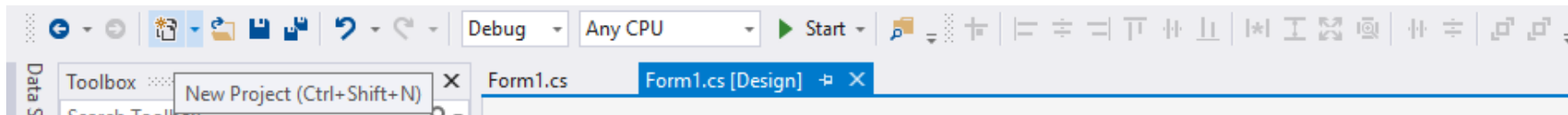
Toolbar

- Menu View juga memungkinkan kita untuk memilih berbagai toolbar yang tersedia pada Visual C# IDE.



Toolbar (Cont.)

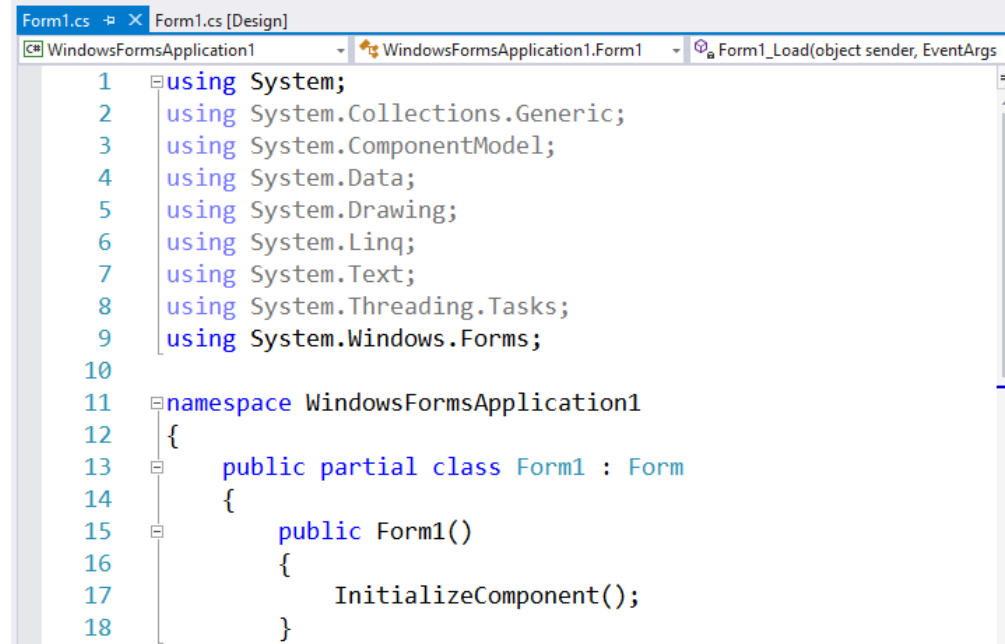
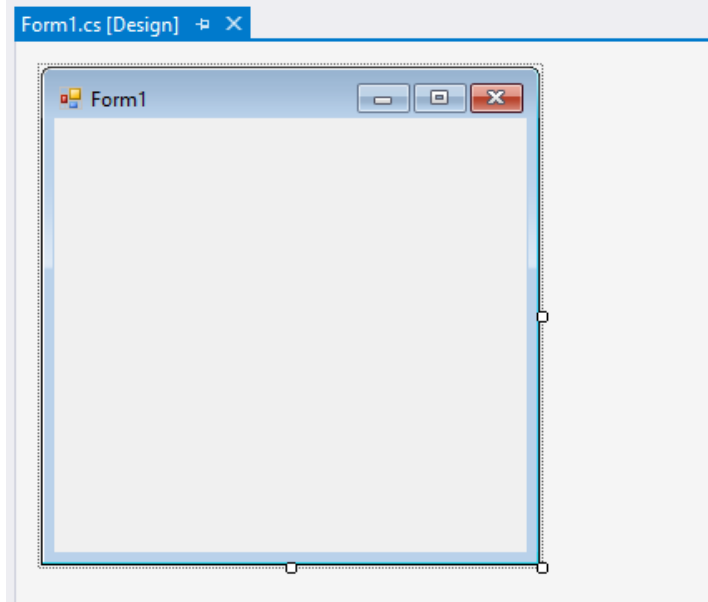
- Toolbar menyediakan akses yang cepat ke banyak fitur.
- Standard (Default) Toolbar dapat dilihat pada gambar dan letaknya berada di bawah menu bar.



- Apabila kita tidak mengetahui fungsi tombol toolbar, kita dapat mengarahkan mouse ke tombol tersebut dan tooltip yang berisi keterangan tombol akan muncul.

Form

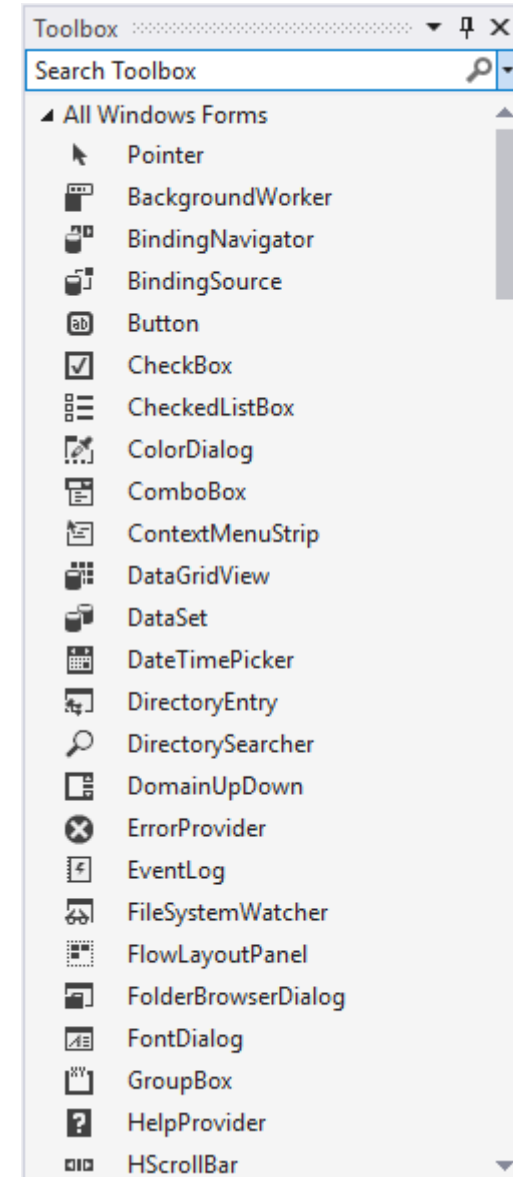
- Merupakan jendela utama pengembangan Visual C# yang terdiri dari dua bagian yaitu:
 - Layar design (Form Designer)
 - Layar penulisan kode program (Code Editor)

A screenshot of the Visual Studio Code Editor window. The title bar shows 'Form1.cs [Design]'. The main area displays the C# source code for 'Form1'. The code includes several 'using' statements for System, System.Collections.Generic, System.ComponentModel, System.Data, System.Drawing, System.Linq, System.Text, System.Threading.Tasks, and System.Windows.Forms. It also shows a namespace declaration for 'WindowsFormsApplication1' and a partial class definition for 'Form1' which inherits from 'Form'. The class has a constructor 'Form1()' that calls 'InitializeComponent()'.

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace WindowsFormsApplication1
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19     }
20 }
```

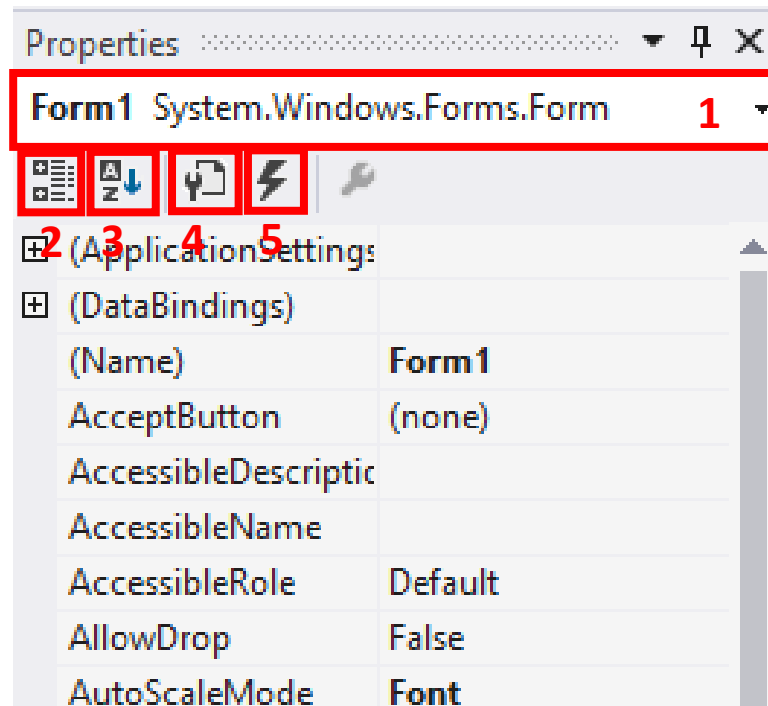
ToolBox

- Berisi kontrol-kontrol yang digunakan dalam aplikasi yang berbasis graphical user interface.
- Toolbox hanya aktif ketika form ditampilkan dalam layar design (Form Designer).
- Untuk menambahkan kontrol dari Toolbox ke form dapat digunakan dengan cara:
 - Klik pada kontrol yang diinginkan, lalu klik atau gambarkan pada form.
 - Drag kontrol dari toolbox.
 - Klik ganda pada kontrol yang diinginkan.



Properties Window

- Mengatur suatu nilai pada property objek dan event.
- Sangat membantu saat bekerja dengan Form Designer, karena kita dapat mengatur properti dengan mudah tanpa harus berurusan langsung dengan pemrograman.

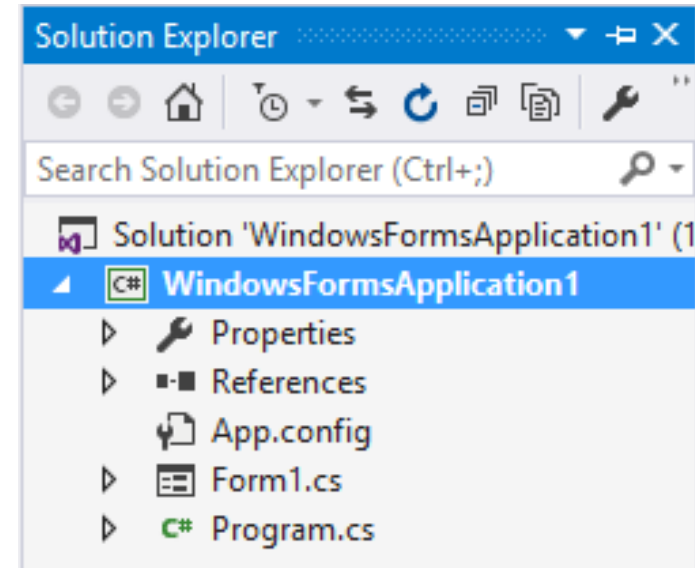


Properties Window (Cont.)

1. Drop-down pada bagian atas berisi daftar semua kontrol yang terdapat dalam form.
2. Property dapat disusun berdasarkan kategori (Categorized).
3. Property dapat disusun berdasarkan alfabet dari A sampai Z (Alphabetic).
4. Menampilkan semua karakteristik (properties) yang dimiliki oleh kontrol yang terpilih.
5. Menampilkan semua event yang dimiliki oleh kontrol yang terpilih.

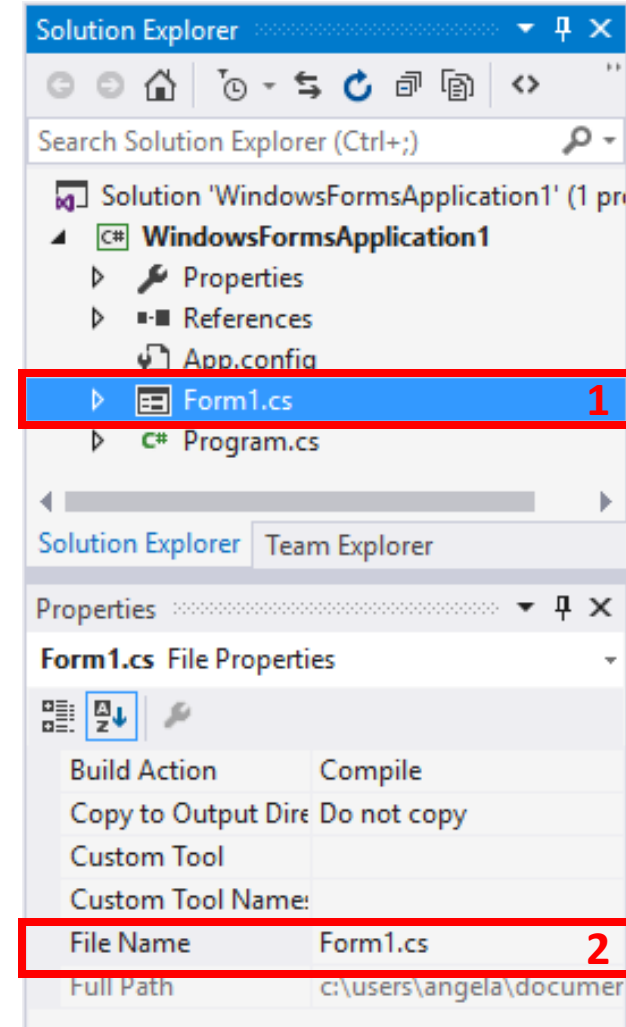
Solution Explorer

- Merupakan window yang membantu developer dalam mengelola project maupun solution yang sedang aktif.
- Menampilkan semua form dan file yang terdapat dalam project.
- Untuk menampilkan form, double klik pada nama form di Solution Explorer.
- Kita dapat memilih tampilan design ataupun kode (View Code).



Mengganti Nama File

- Untuk mengganti nama file, klik Form1 di jendela **Solution Explorer**. Selanjutnya nama file akan ditampilkan di jendela **Properties**. Ganti properti **File Name** dengan nama file yang diinginkan.
- Alternatif: klik kanan pada Form1 di jendela **Solution Explorer**, pilih **Rename**, lalu ganti nama filenya.



Menyimpan Project

- Pilih menu **Save All** pada Toolbar atau dapat juga dengan memilih dari menu **File** → **Save All** .





- Pastikan untuk menyimpan project sebelum menjalankan aplikasi atau keluar dari Visual Studio.










Struktur Project

- Visual C# menggunakan struktur folder yang sangat spesifik untuk menyimpan keseluruhan komponen pada suatu aplikasi.
- Ketika menyimpan suatu project baru (new project / solution), anda akan diminta untuk memasukkan **Name** dan **Location**. Folder dengan nama **Name** akan terbentuk pada lokasi **Location**. Folder tersebut akan berisi seluruh solution files, project files, dan file-file lain.
- Pada folder tersebut, terdapat 2 folder dengan nama: **Bin** dan **Obj**.

Struktur Project (Cont.)

 WindowsFormsApplication1
 WindowsFormsApplication1.sln



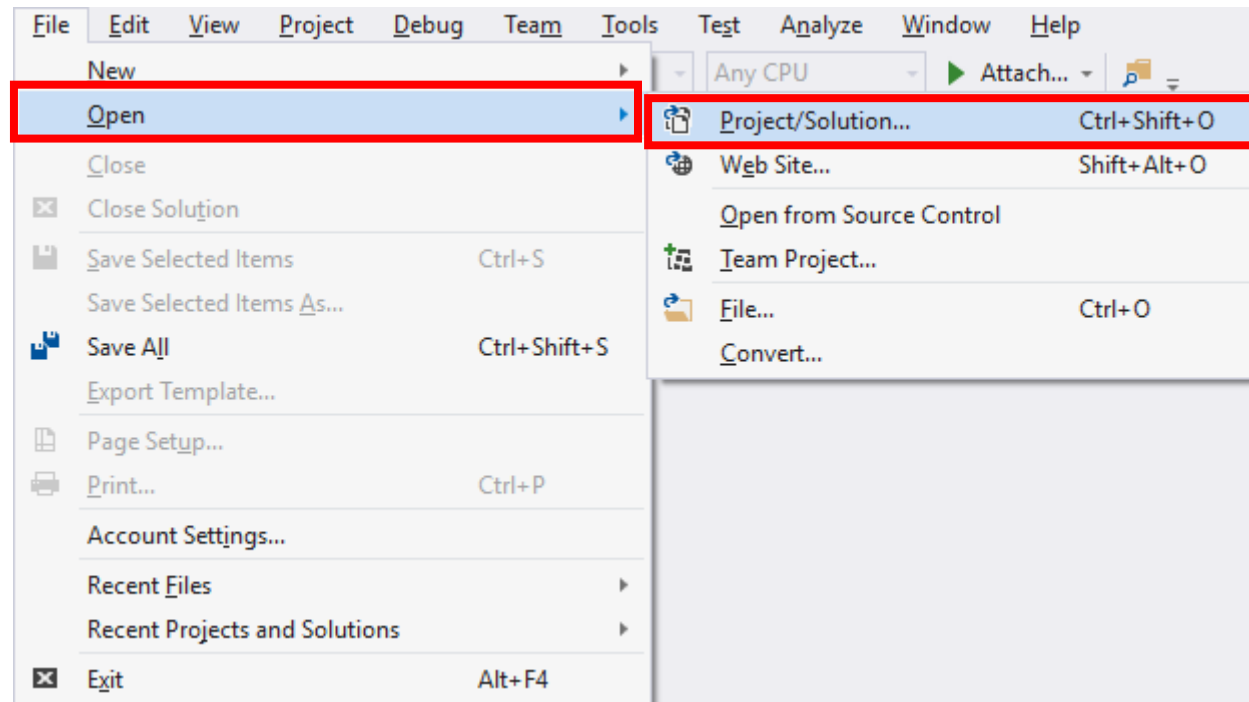
-  bin
-  obj
-  Properties
-  App.config
-  Form1.cs
-  Form1.Designer.cs
-  Form1.resx
-  Program.cs
-  WindowsFormsApplication1.csproj

Struktur Project (Cont.)

File	Keterangan
Program.cs	Information on how things fit together
SolutionName.sln	Solution file for solution named SolutionName
SolutionName.suo	Solution options file
ProjectName.resx	Project file – one for each project in solution
FormName.resx	Form resources file – one for each form
FormName.cs	Form code file – one for each form
FormName.Designer.cs	File holding design information for form
App.ico	Icon used to represent the application

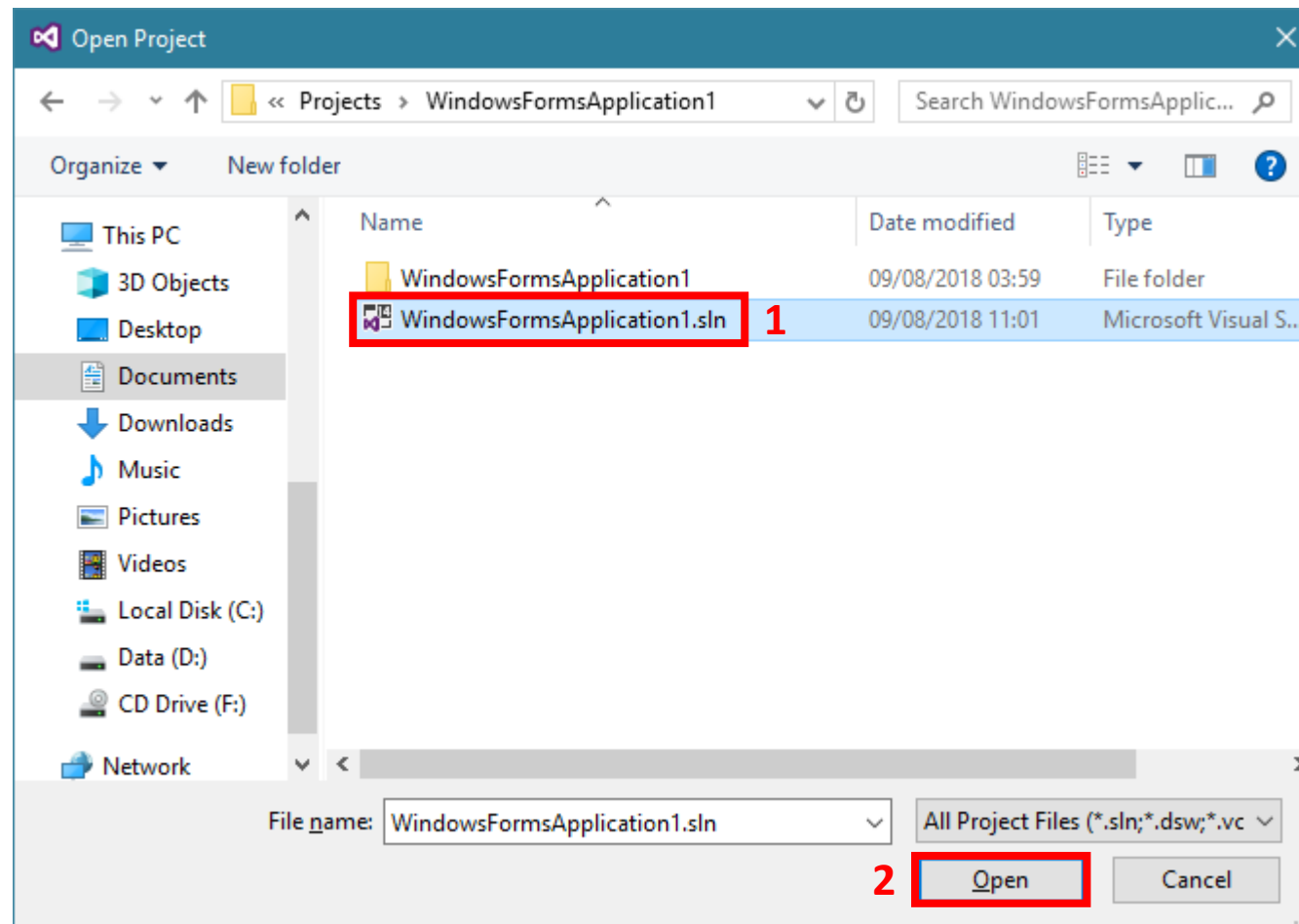
Membuka Saved Project

1. Pilih menu **File** → **Open** → **Project/Solution (Ctrl + Shift + O)**



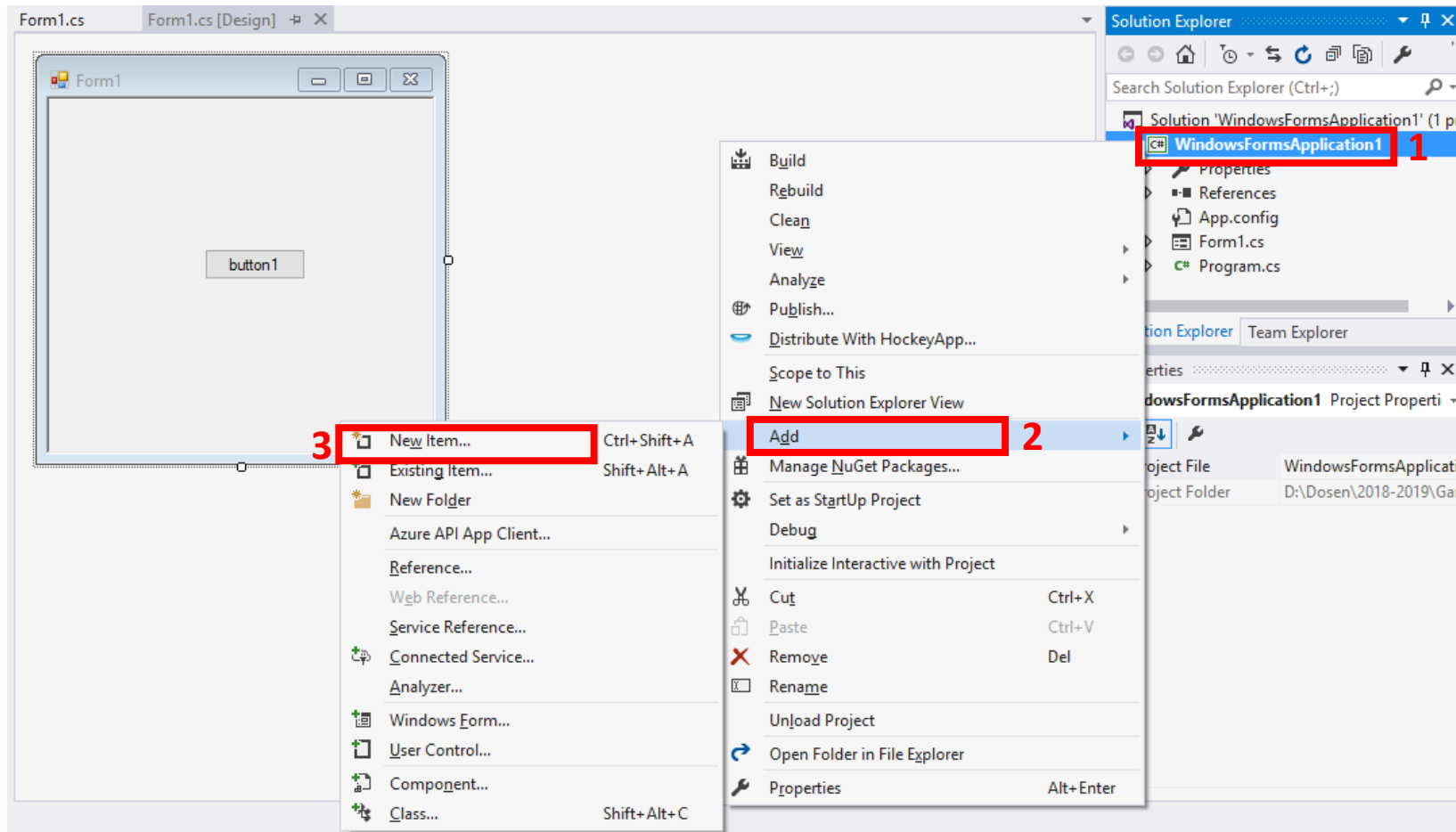
Membuka Saved Project (Cont.)

2. Kotak dialog Open Project akan terbuka. Cari lokasi penyimpanan project. Kemudian pilih file Solution (.sln), dan klik **Open**.



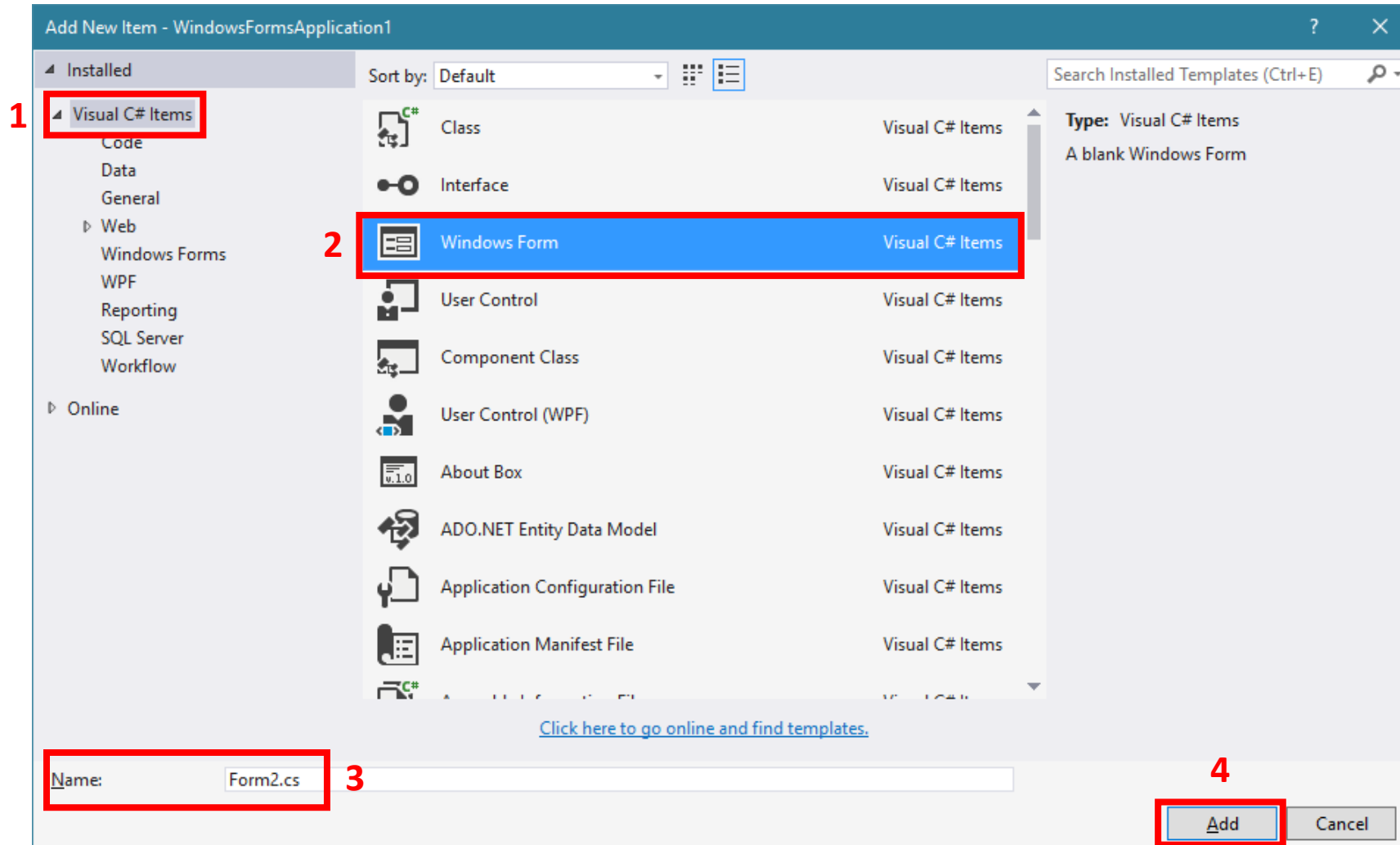
Menambahkan Item Baru

1. Klik kanan pada **nama project** pada **solution explorer** kemudian klik **add** → **new item** untuk menampilkan jendela **add new item**.



Menambahkan Item Baru (Cont.)

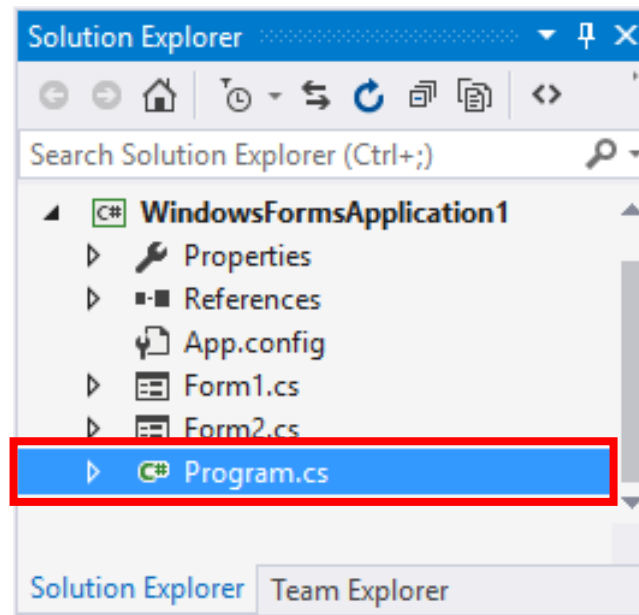
2. Pilih item yang akan ditambahkan, **ubah nama item**, kemudian klik **button Add**.



Memilih Form StartUp

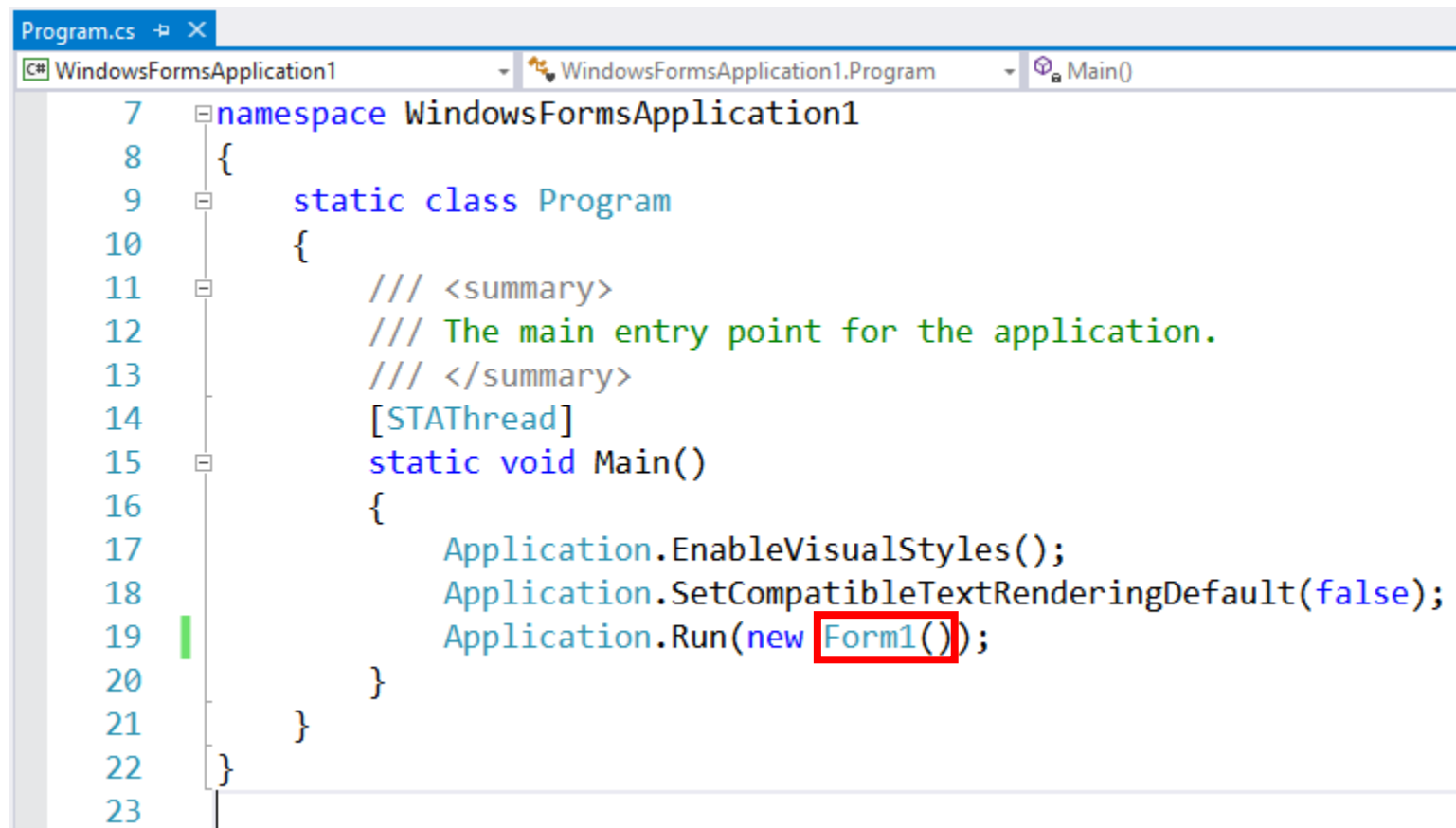
Apabila dalam suatu project terdapat beberapa form, kita dapat mengatur form mana yang muncul ketika aplikasi dijalankan.

1. Double klik **Program.cs** pada Solution Explorer



Memilih Form StartUp (Cont.)

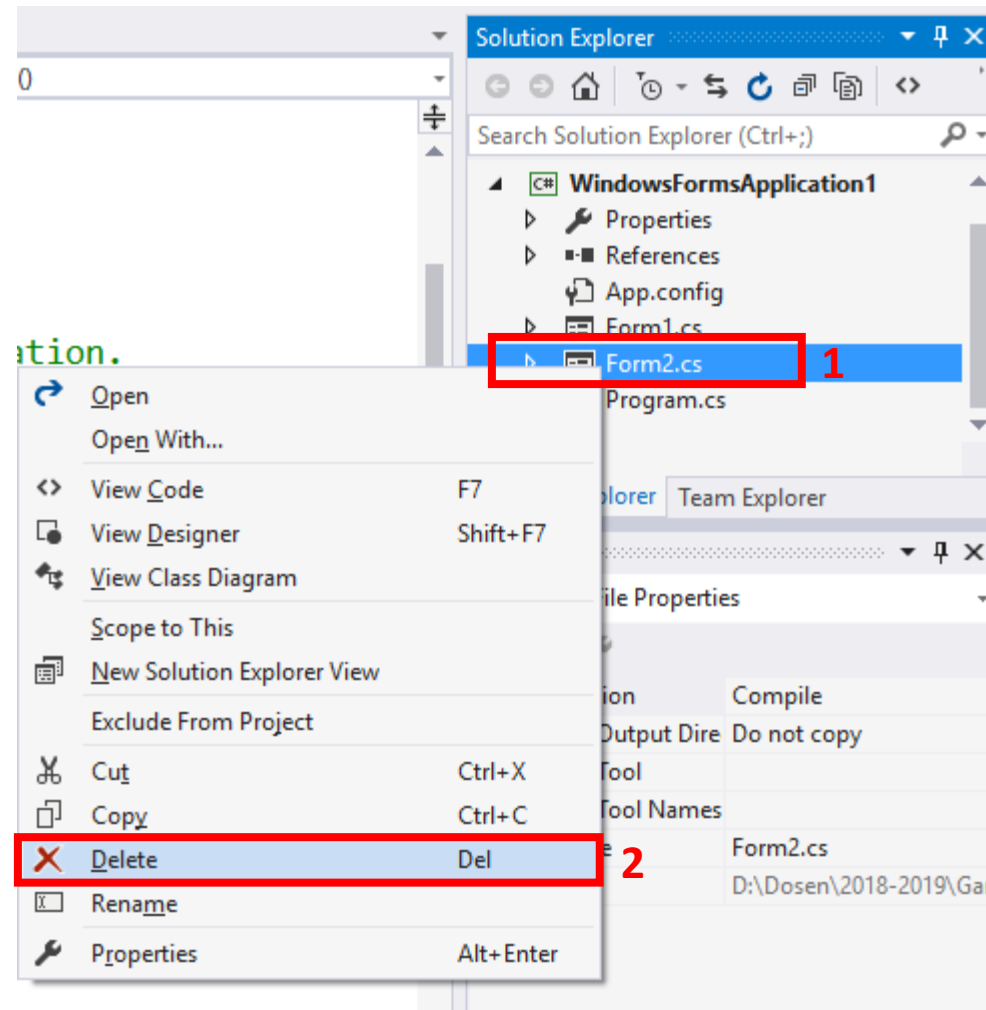
- Ubah nama form pada kotak berwarna merah menjadi nama form yang ingin dijalankan.



```
Program.cs
C# WindowsFormsApplication1 WindowsFormsApplication1.Program Main()
7 namespace WindowsFormsApplication1
8 {
9     static class Program
10    {
11        /// <summary>
12        /// The main entry point for the application.
13        /// </summary>
14        [STAThread]
15        static void Main()
16        {
17            Application.EnableVisualStyles();
18            Application.SetCompatibleTextRenderingDefault(false);
19            Application.Run(new Form1());
20        }
21    }
22 }
23
```

Menghapus Item

- Klik kanan pada **nama item** pada solution explorer yang akan dihapus kemudian klik **delete**.



Mengenal Component/Control

- Kontrol adalah objek yang terkait dengan user interface yang dapat ditambahkan pada form, baik untuk menampilkan informasi, menerima input user, atau keduanya. Kontrol disediakan oleh .NET Framework Class Library. Visual C# dapat menggunakan class library yang disediakan oleh .NET Framework untuk Windows Forms maupun kontrol. Kontrol terkait erat dengan class `System.Windows.Forms.Control`, karena semua class untuk kontrol diturunkan dari class tersebut. Kontrol dapat ditambahkan dengan mudah pada Form apabila menggunakan Form Designer. Namun, adakalanya harus berurusan dengan kode pemrograman secara langsung dan tidak menggunakan Form Designer maupun Properties Window. Selain kontrol untuk user interface, terdapat juga kontrol nonuser interface dan kontrol ActiveX.

Penamaan Control

- Property yang paling penting dari setiap kontrol adalah property **Name**. Pada saat menulis kode program, property Name digunakan untuk merujuk pada kontrol yang diinginkan.
- Terdapat aturan dalam pemberian nama kontrol. Aturan ini diterapkan dengan menambah awalan sebanyak 3 huruf kecil (lowercase) yang menunjukkan jenis kontrol, kemudian diikuti dengan nama kontrol yang ingin diberikan.

Penamaan Control (Cont.)

Kontrol	Prefix	Contoh
Form	frm	frmMahasiswa
Button	btn	btnHitung
Label	lbl	lblNama
TextBox	txt	txtAlamat
Menu	mnu	mnuSimpan
CheckBox	chk	chkMerah

- Disarankan untuk menggabungkan huruf besar dan huruf kecil supaya lebih mudah dibaca, tetapi ingat bahwa C# bersifat case-sensitive, yang berarti lblNama berbeda dengan LBLNAMA

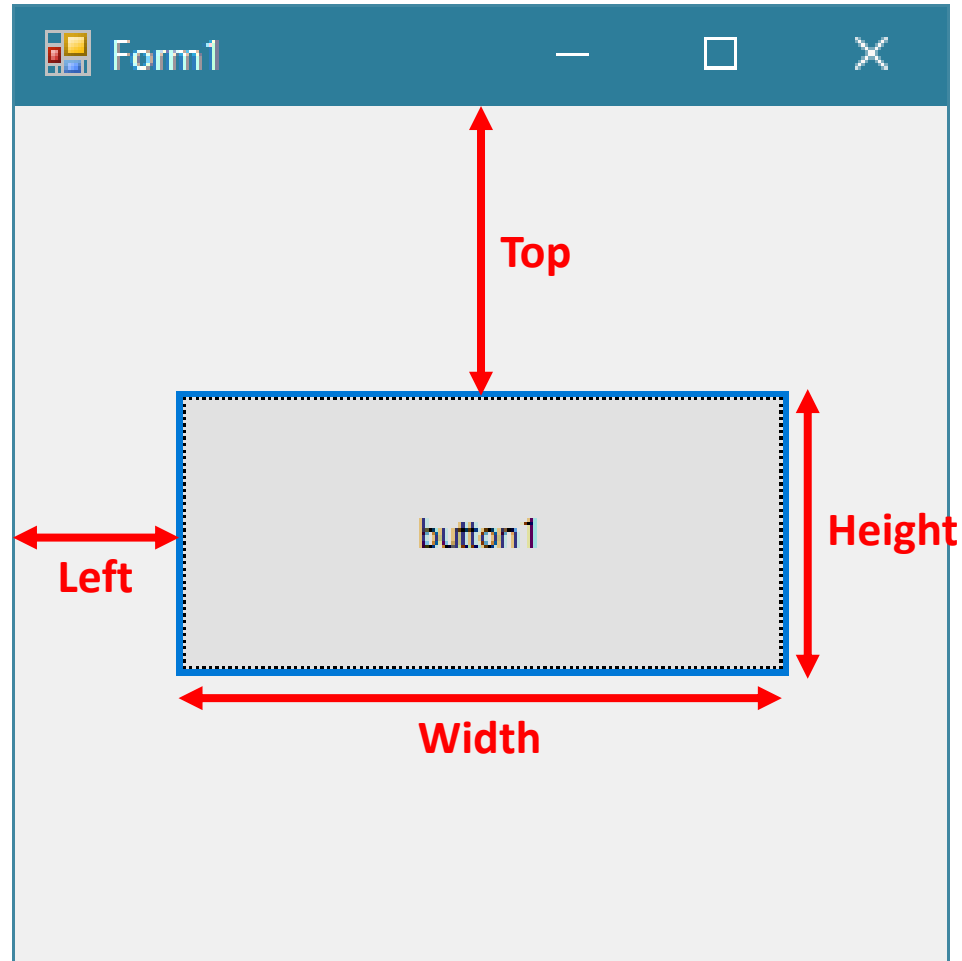
Penamaan Control (Cont.)

- Property Name pada kontrol dapat menampung 40 karakter dan harus dimulai dengan huruf dan hanya boleh terdiri dari huruf, angka, dan karakter underscore “ ”
—
- Berikan property name yang bermakna yang dapat membantu kita ataupun orang lain untuk memahami jenis kontrol dan fungsi dari kontrol bersangkutan.

Property Umum Class Control

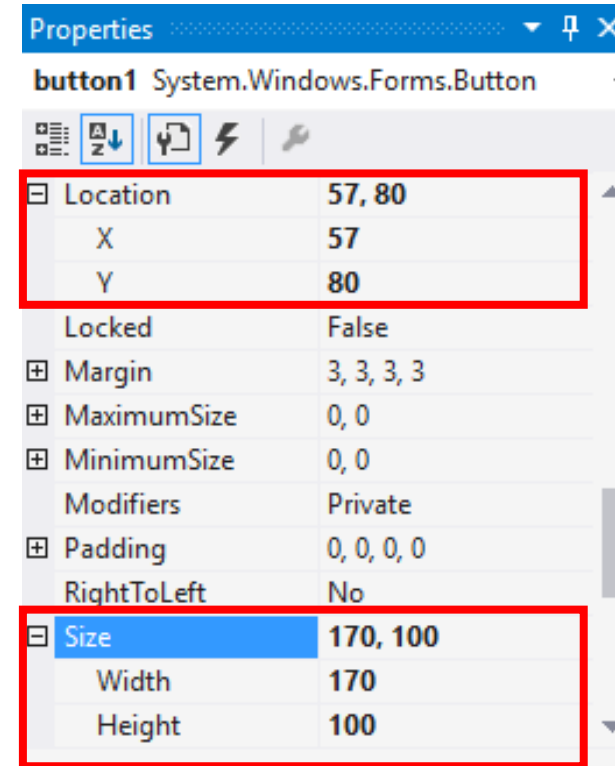
- Property yang umumnya terdapat pada kontrol adalah property yang berhubungan dengan posisi (**position**) dan ukuran (**size**).
- Terdapat 4 property (dalam satuan pixel), yaitu:
 - **Left** : jarak dari sisi kiri form ke sisi kiri kontrol
 - **Top** : jarak dari judul form ke sisi atas kontrol
 - **Width** : lebar kontrol
 - **Height** : tinggi kontrol

Property Umum Class Control (Cont.)



Property Umum Class Control (Cont.)

- Untuk melihat property Left dan Top pada Properties Window, tekan tanda + di samping property **Location**. Nilai X adalah property Left, sedangkan nilai Y adalah property Top.
- Untuk melihat property Width dan Height pada Properties Window, tekan tanda + di samping property **Size**. Nilai Width dan Height akan ditampilkan, dan nilai ini dapat diubah.



Mengatur Nilai Properties

- Mengatur nilai properties dapat dilakukan pada saat design melalui Properties Window ataupun ketika aplikasi dijalankan melalui kode program.

- Format penulisan kode program:

```
objectName.PropertyName = NewValue;
```

- Contoh:

```
btnStart.BackColor = Color.Blue;
```

Property Name Pada Event

- Untuk menambahkan event baru pada kontrol, dapat melalui 2 cara.
- Cara pertama yaitu dengan double klik pada kontrol untuk memunculkan event default dari kontrol tersebut.
- Kode yang muncul ketika kita melakukan double klik pada kontrol akan mengikuti format sebagai berikut:

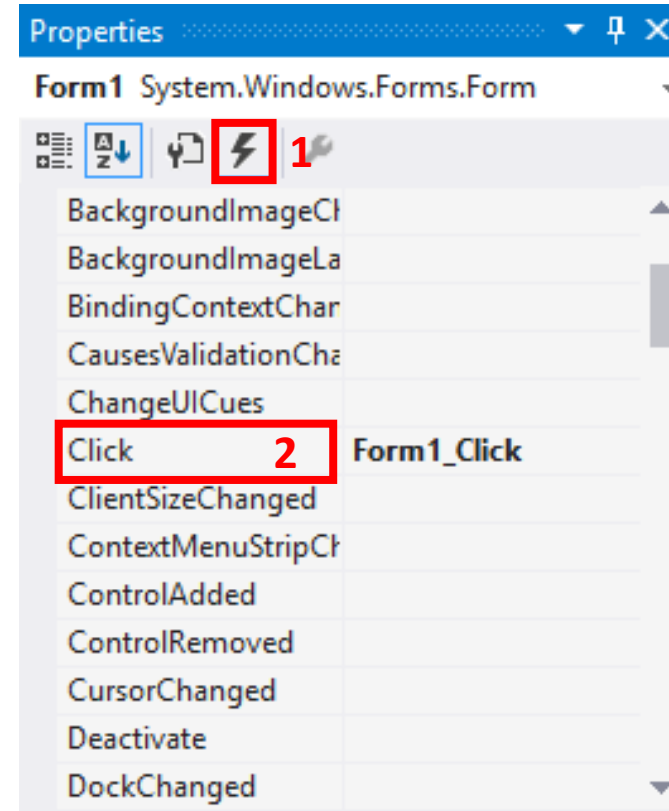
```
private void ControlName_Event(Arguments)
{

}
```

- Keseluruhan baris koding yang dijalankan ketika event terjadi, diletakkan di antara kedua kurung kurawal.

Property Name Pada Event (Cont.)

- Cara kedua adalah melalui **Properties Window**. Cara ini terutama digunakan apabila kita ingin menggunakan event yang lain dari kontrol (bukan event default).
- Pilih tombol **event** pada Properties Window dan double klik pada event yang diinginkan. Kode program akan langsung muncul pada Code Window.

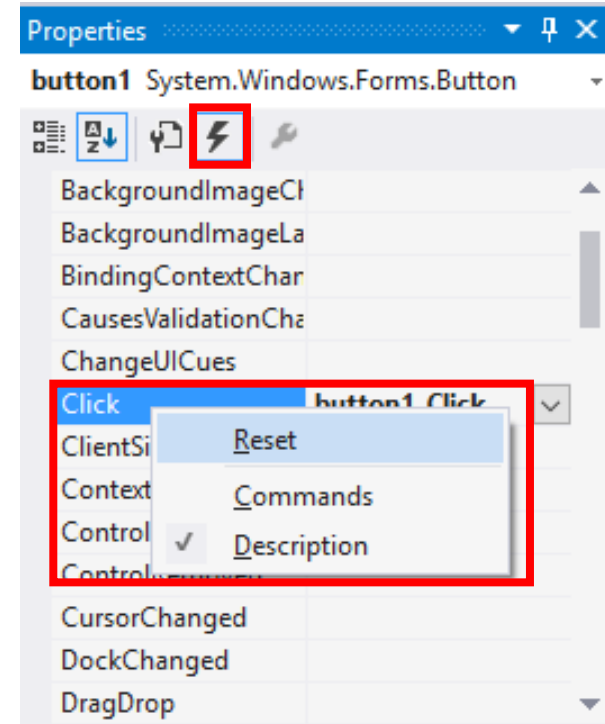


Property Name Pada Event (Cont.)

- Setelah kita menambahkan sebuah event pada aplikasi, Visual C# secara otomatis menuliskan baris koding untuk menghubungkan metode (baris kode yang ditulis) dengan event yang dipakai.
- Baris koding tersebut akan tetap ada walaupun kita telah menghapus metode (baris kode) sehingga akan terjadi error pada saat aplikasi dijalankan.

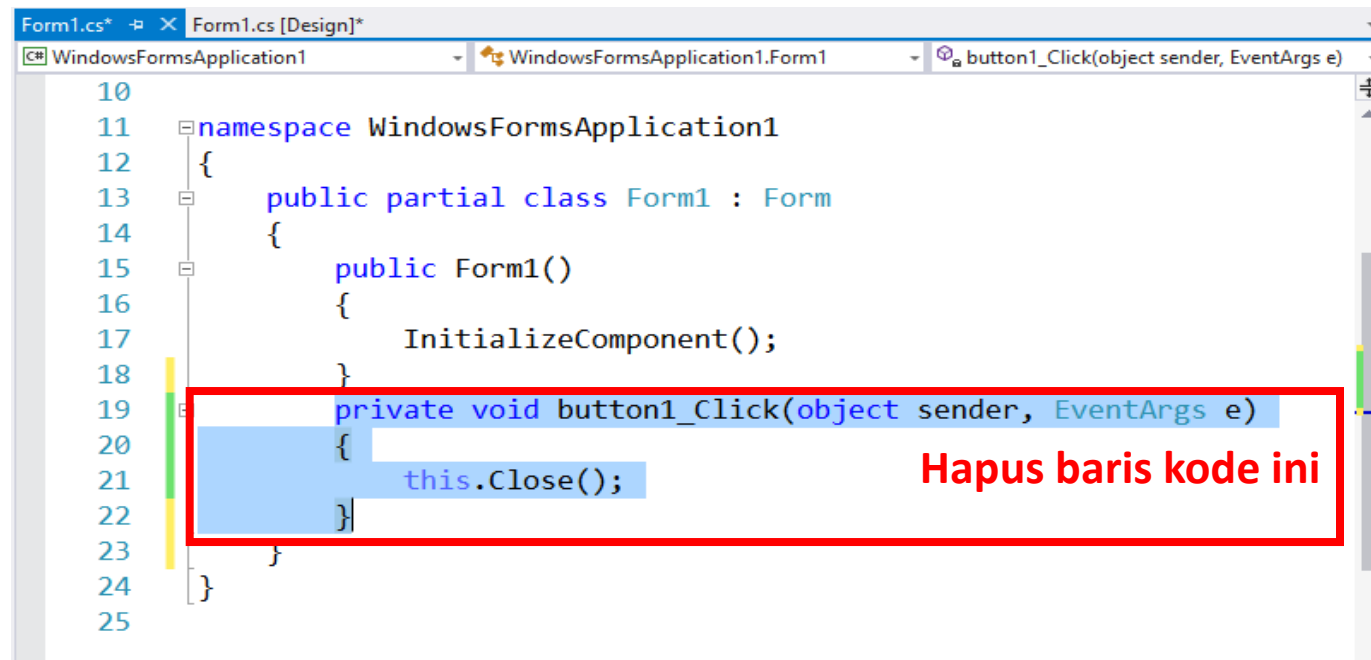
Property Name Pada Event (Cont.)

- Untuk mengatasinya, pertama-tama tekan tombol Event pada Properties Window, kemudian klik kanan pada event yang ingin dihapus, lalu pilih **Reset**.
- Proses ini akan menghapus baris coding yang menghubungkan metode dengan event.



Property Name Pada Event (Cont.)

- Setelah itu, baru hapus metode (baris kode) yang berhubungan dengan event tersebut pada Code Window.



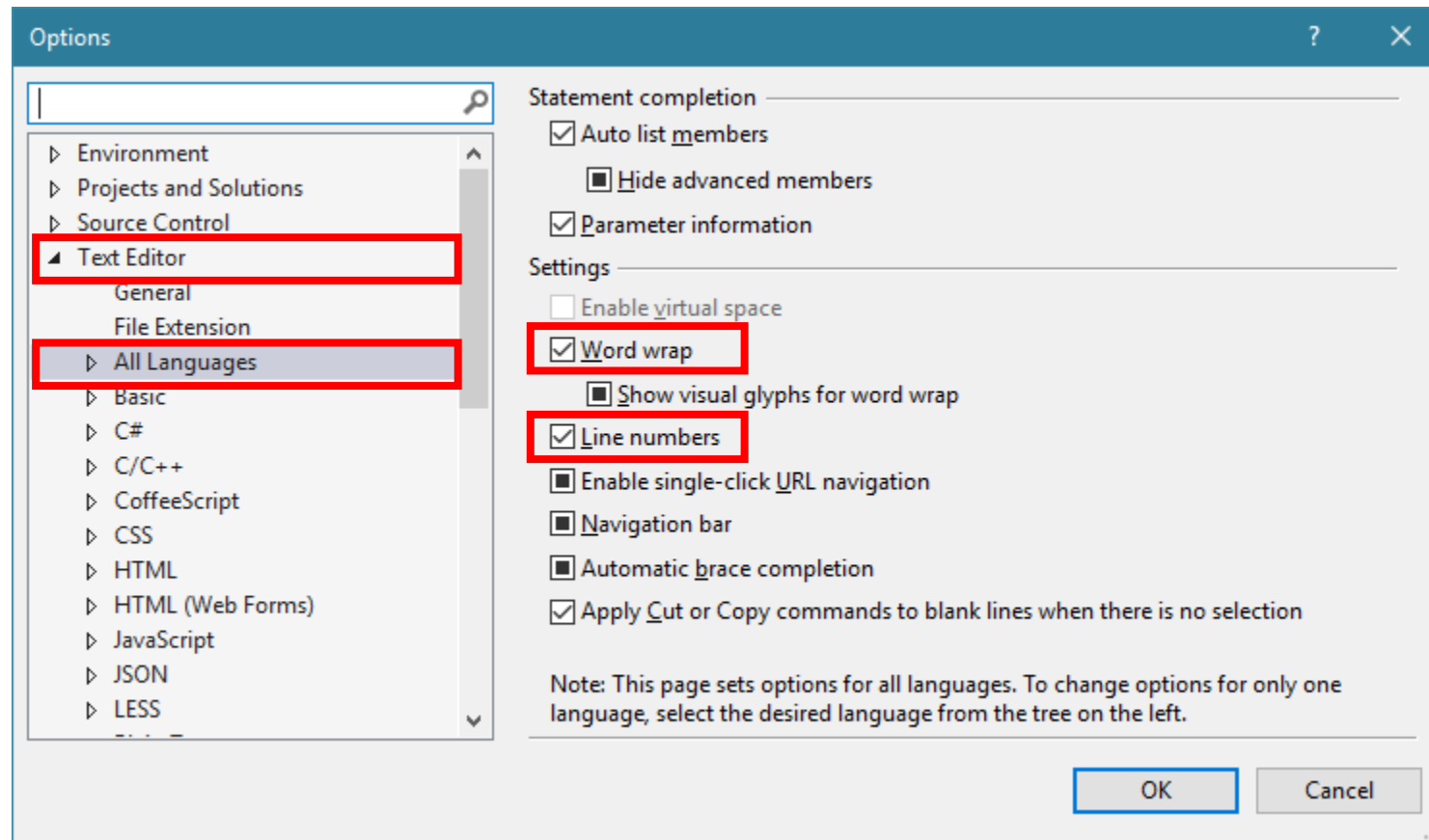
```
10
11 namespace WindowsFormsApplication1
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19         private void button1_Click(object sender, EventArgs e)
20         {
21             this.Close();
22         }
23     }
24 }
25
```

Hapus baris kode ini

Penulisan Kode Program

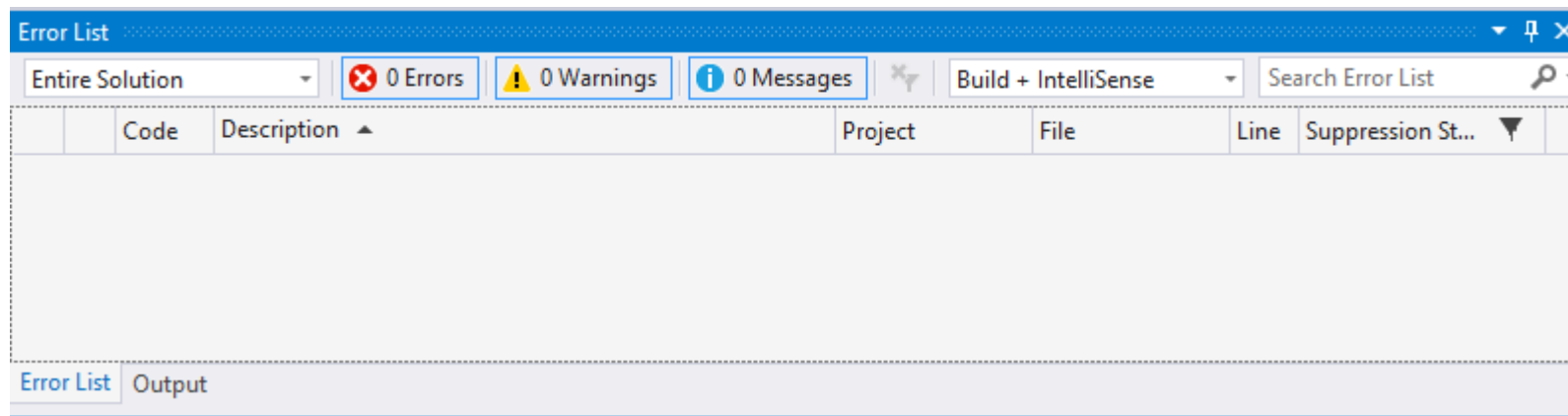
- Langkah terakhir dalam membangun aplikasi berbasis Visual C# adalah menuliskan kode program dengan menggunakan Bahasa C#.
- Fitur yang berguna dalam penulisan kode program adalah dengan menampilkan nomor baris (line number) di samping kode program yang ditulis.
- Cara menampilkannya adalah dengan memilih menu **Tools → Options**. Pada bagian kiri windows, pilih **Text Editor**, kemudian **All Languages**, lalu centang **Line numbers** pada sisi sebelah kanan.
- Kita juga dapat mencentang pilihan **Word wrap** agar penulisan kode program bisa secara otomatis enter ke bawah apabila telah mencapai ujung kanan layar.

Penulisan Kode Program (Cont.)



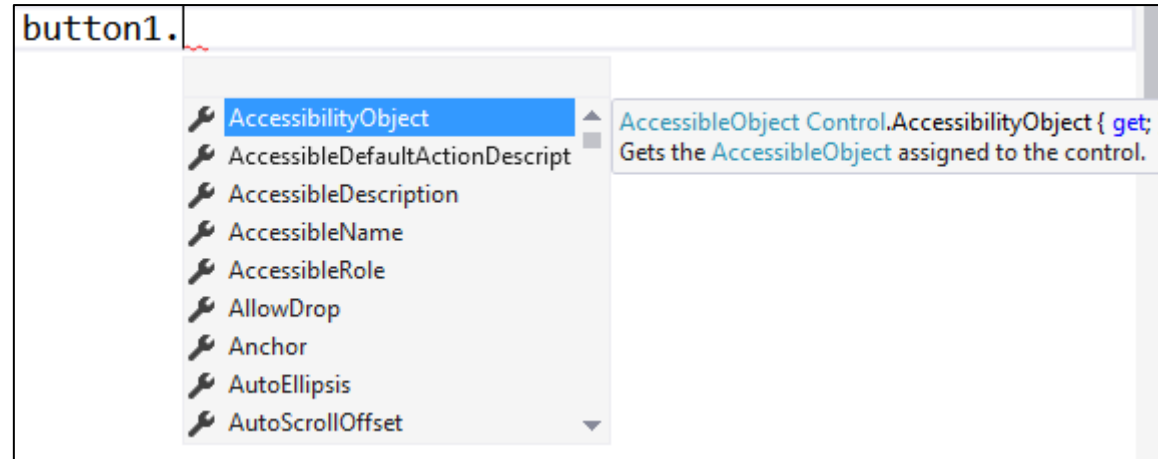
Penulisan Kode Program (Cont.)

- Line numbers tersebut akan berguna untuk mengidentifikasi di baris nomor berapa suatu error terjadi pada kode program.
- Informasi tersebut dapat dilihat pada **Error List Window**. Untuk mengaktifkannya, pilih menu **View → Error List**.



Fitur Intellisense

- Merupakan salah satu fasilitas yang terdapat pada layar kode.
- Berguna untuk menyelesaikan penulisan baris kode.
- Contoh, ketika kita mengetikkan nama kontrol diikuti dengan tanda titik “.”, maka Code Editor akan menampilkan informasi properties dan methods yang berhubungan dengan kontrol tersebut secara otomatis. Apabila kita menggunakan methods, informasi mengenai arguments (parameter) akan dimunculkan. Kesalahan penulisan syntax akan diidentifikasi dan error yang mungkin muncul akan diinformasikan.



Mengenal Form

- Windows Form adalah platform baru untuk pembuatan aplikasi Windows berbasis .NET Framework. Framework ini menyediakan sekelompok class yang berorientasi objek dan dapat dikembangkan lebih lanjut untuk membangun aplikasi windows yang lengkap. Namespace dari class library yang disediakan oleh .NET Framework adalah System.Windows.Forms.

Form

- Form merupakan tempat mendesain user interface. Bagian dari tampilan di layar, biasanya berbentuk kotak, yang dapat digunakan untuk menampilkan informasi ke pengguna dan menerima input dari pengguna.
- Pada form tersedia tombol Minimize, Maximize/Restore, dan Close.

Form (Cont.)

- Property umum form:

Property	Keterangan
Name	Mengatur nama form (Form diawali dengan frm)
AcceptButton	Mengatur button mana dalam form yang ditekan apabila user menekan tombol Enter
BackColor	Mengatur warna latar form
CancelButton	Mengatur button mana dalam form yang ditekan apabila user menekan tombol Esc
ControlBox	Mengatur apakah control box (Minimize, Maximize/Restrore, Close) ditampilkan pada form atau tidak
Enabled	Jika diatur menjadi False, maka semua kontrol dalam form menjadi tidak aktif
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan form

Form (Cont.)

Property	Keterangan
FormBorderStyle	Mengatur bingkai form apakah ukurannya bisa diubah atau tidak (fixed atau sizeable)
Height	Mengatur tinggi form dalam satuan pixel
Icon	Mengatur icon form
Left	Mengatur jarak dari sisi kiri layar ke sisi kiri form dalam satuan pixel
MaximizeButton	Mengatur apakah tombol Maximize/Restrore ditampilkan pada form atau tidak
MinimizeButton	Mengatur apakah tombol Minimize ditampilkan pada form atau tidak
StartPosition	Mengatur posisi form pada saat aplikasi dijalankan
Text	Mengatur judul form
Top	Mengatur jarak dari sisi atas layar ke sisi atas form dalam satuan pixel
Width	Mengatur lebar form dalam satuan pixel

Form (Cont.)

- Metode umum form:

Method	Keterangan
Close	Menutup form
Focus	Mengatur fokus pada form
Hide	Menyembunyikan form
Show	Menampilkan form

- Sintaks penggunaan metode khusus form:

```
this.method();
```

Form (Cont.)

- Event umum form:

Event	Keterangan
Activated	Terjadi ketika form diaktifkan melalui koding atau oleh user
Click	Terjadi ketika form di-klik oleh user
FormClosing	Terjadi ketika form sedang ditutup
DoubleClick	Terjadi ketika form di-doubleklik oleh user
Load (default)	Terjadi ketika form dijalankan pertama kali

Form (Cont.)

- Form dikenal sebagai objek **container**, karena berfungsi menampung kontrol-kontrol yang lain.
- Dampaknya adalah kontrol-kontrol yang diletakkan dalam form akan mewarisi property **BackColor**, **ForeColor**, dan **Font** dari form.
- Selain itu, apabila container diatur property **Enabled** menjadi False, maka semua kontrol yang ada di dalam container akan menjadi Disabled. Demikian pula dengan property **Visible**.

Kontrol Button, Label, dan Textbox

Button

- Biasanya digunakan untuk mengeksekusi suatu aksi dengan cara mengklik.
- Property umum pada button:

Property	Keterangan
Name	Mengatur nama button (Button diawali dengan btn)
BackColor	Mengatur warna latar button
Enabled	Jika diatur menjadi False, button kelihatan tetapi tidak dapat menerima klik
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Image	Mengatur gambar yang akan ditampilkan pada button
Text	Mengatur tulisan yang tampil pada button
TextAlign	Mengatur perataan teks pada button

Button (Cont.)

- Metode umum pada button:

Method	Keterangan
Focus	Mengatur fokus ke button
PerformClick	Menjalankan event click pada button

Event	Keterangan
Click (default)	Terjadi ketika button ditekan

Label

- Kontrol Label digunakan untuk menampilkan teks atau gambar yang tidak dapat diedit oleh user.
- Property umum pada label:

Property	Keterangan
Name	Mengatur nama label (Label diawali dengan lbl)
AutoSize	Mengatur apakah ukuran label berubah secara otomatis sesuai dengan isi label atau tidak. Secara default, kontrol label akan berubah secara otomatis mengikuti isi label sehingga untuk mengubah ukuran label, property AutoSize harus bernilai False
BackColor	Mengatur warna latar label
BorderStyle	Mengatur tipe bingkai pada label

Label (Cont.)

- Metode umum pada button:

Property	Keterangan
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Text	Mengatur tulisan yang akan ditampilkan pada label
TextAlign	Mengatur perataan teks pada label

- Event umum pada label:

Event	Keterangan
Click (default)	Terjadi ketika user menekan label
DoubleClick	Terjadi ketika user menekan label 2x secara cepat (double klik)

TextBox

- Digunakan untuk mendapatkan input dari user atau menampilkan teks.
- Teks di dalam TextBox dapat di-edit dan dapat juga dibuat menjadi read-only.
- TextBox dapat menampilkan satu baris atau banyak baris.
- Property umum pada textbox:

Property	Keterangan
Name	Mengatur nama TextBox (TextBox dimulai dengan txt)
AutoSize	Mengatur apakah ukuran tinggi dari textbox akan secara otomatis mengikuti tinggi tulisan
BackColor	Mengatur warna latar textbox
BorderStyle	Mengatur tipe bingkai textbox

TextBox (Cont.)

Property	Keterangan
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
HideSelection	Mengatur apakah tulisan yang diblok pada textbox akan tetap dalam kondisi terblok setelah textbox lost focus
MaxLength	Mengatur jumlah karakter maksimum yang dapat ditampung textbox
MultiLine	Mengatur apakah textbox menampilkan 1 baris atau dapat lebih dari 1 baris
PasswordChar	Mengatur karakter yang akan dimunculkan dalam textbox
ReadOnly	Mengatur apakah tulisan dalam textbox dapat diedit atau tidak
ScrollBars	Mengatur apakah scrollbar akan muncul pada textbox multiline

TextBox (Cont.)

Property	Keterangan
Text	Mengatur tulisan yang tampil pada textbox
TextAlign	Mengatur perataan tulisan pada textbox
TextLength	Mengatur panjang tulisan pada textbox

- Metode umum pada textbox:

Method	Keterangan
AppendText	Menambahkan teks baru pada teks yang sudah ada di textbox
Clear	Membersihkan textbox
Copy	Meng-copy teks yang terpilih ke clipboard
Cut	Memindahkan teks yang terpilih ke clipboard
Focus	Meletakkan kursor pada textbox yang bersangkutan
Paste	Menggantikan teks yang terpilih di textbox dengan konten yang ada di clipboard

TextBox (Cont.)

Method	Keterangan
SelectAll	Memilih semua teks yang ada di textbox
Undo	Membatalkan operasi edit terakhir

- Event umum pada textbox:

Event	Keterangan
Click	Terjadi ketika user menekan textbox
Enter	Terjadi ketika textbox menerima focus
KeyDown	Terjadi ketika tombol keyboard ditekan pada saat textbox menerima focus
KeyPress	Terjadi ketika tombol keyboard yang memiliki kode ASCII ditekan pada saat textbox menerima focus
Leave	Terjadi ketika textbox kehilangan focus. Ini merupakan saat yang tepat untuk memeriksa isi textbox.
TextChanged (default)	Terjadi ketika nilai property Text berubah

TextBox (Cont.)

- Penggunaan kontrol textbox harus diminimalisir jika memungkinkan karena apabila kita mengizinkan user untuk mengetikkan sesuatu, pekerjaan kita sebagai programmer akan lebih sulit. Kita harus memvalidasi semua inputan user untuk memastikan informasi tersebut dapat bekerja pada koding kita.
- Disarankan untuk menggunakan kontrol lain yang bersifat 'point and click' untuk menggantikan textbox.

Latihan

Latihan 1

Validasi Password

Masukkan password anda:

Validasi Masuk Keluar

Validasi Password

Masukkan password anda:

Validasi Masuk Keluar

Password salah! Input ulang!

Validasi Password

Masukkan password anda:

Validasi Masuk Keluar

Password benar!

Form2

**Selamat datang di
aplikasi Visual C#
pertama anda!**

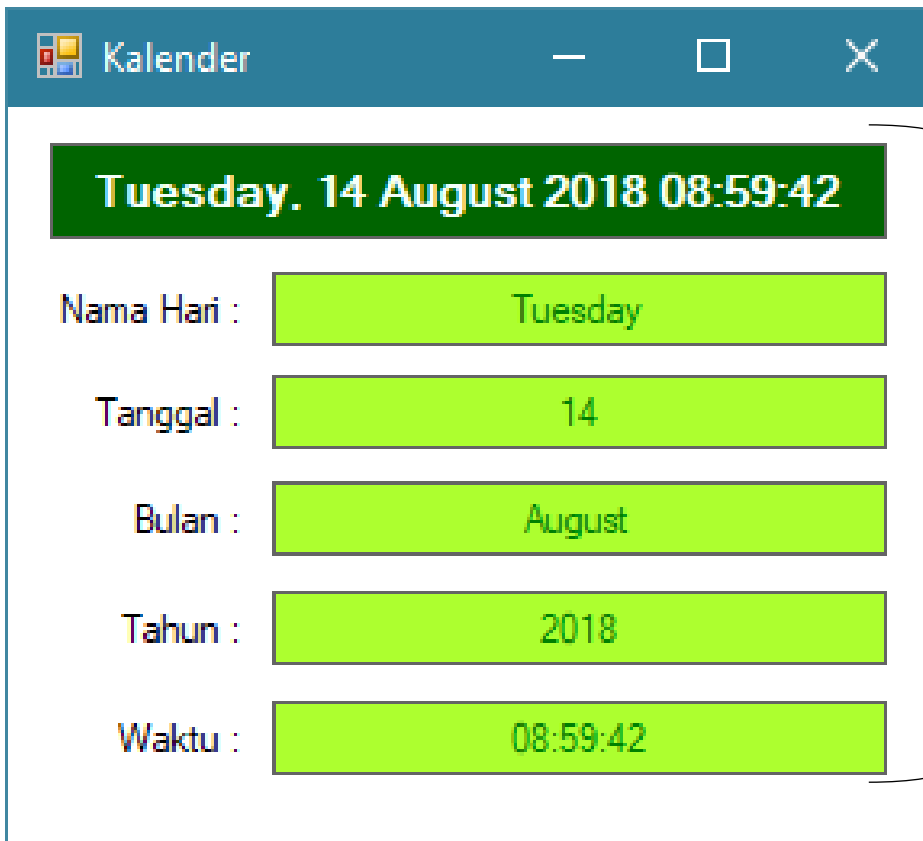
Tutup

Latihan 1 (Cont.)

Form1	Name	frmLatihan1a
	AcceptButton	btnValidasi
	CancelButton	btnKeluar
	Text	Validasi Password
Label1	Text	Masukkan password anda:
TextBox1	Name	txtPassword
	PasswordChar	*
Button1	Name	btnValidasi
	Text	&Validasi
Button2	Name	btnMasuk
	Text	&Masuk
Button3	Name	btnKeluar
		&Keluar

Label2	Name	lblKeterangan
	AutoSize	False
	BorderStyle	FixedSingle
	TextAlign	MiddleCenter
Form2	Name	frmLatihan1b
	ControlBox	False
Label1	AutoSize	False
	FontStyle	Bold
	FontSize	16
	Text	Selamat datang di aplikasi Visual C# pertama anda!
	TextAlign	MiddleCenter
Button1	Name	btnTutup
	Text	&Tutup

Latihan 2



The screenshot shows a Windows application window titled "Kalender". Inside the window, there is a dark green rectangular area at the top displaying the text "Tuesday. 14 August 2018 08:59:42" in a light green, bold font. Below this area, there are five input fields, each with a label to its left: "Nama Hari :", "Tanggal :", "Bulan :", "Tahun :", and "Waktu :". The input fields contain the values "Tuesday", "14", "August", "2018", and "08:59:42" respectively. The input fields have a yellow background and a black border. A blue bracket on the right side of the input fields points to a list of properties for a Label control.

Label
Name: lblTglLengkap, lblHari, lblTanggal, lblBulan, lblTahun, lblWaktu
AutoSize: False
BorderStyle: FixedSingle
TextAlign: MiddleCenter
BackColor: DarkGreen (lblTglLengkap), GreenYellow
ForeColor: MintCream (lblTglLengkap), Green
Font: Bold, 10pt (lblTglLengkap)

Label

Name: lblTglLengkap, lblHari, lblTanggal, lblBulan, lblTahun, lblWaktu

AutoSize: False

BorderStyle: FixedSingle

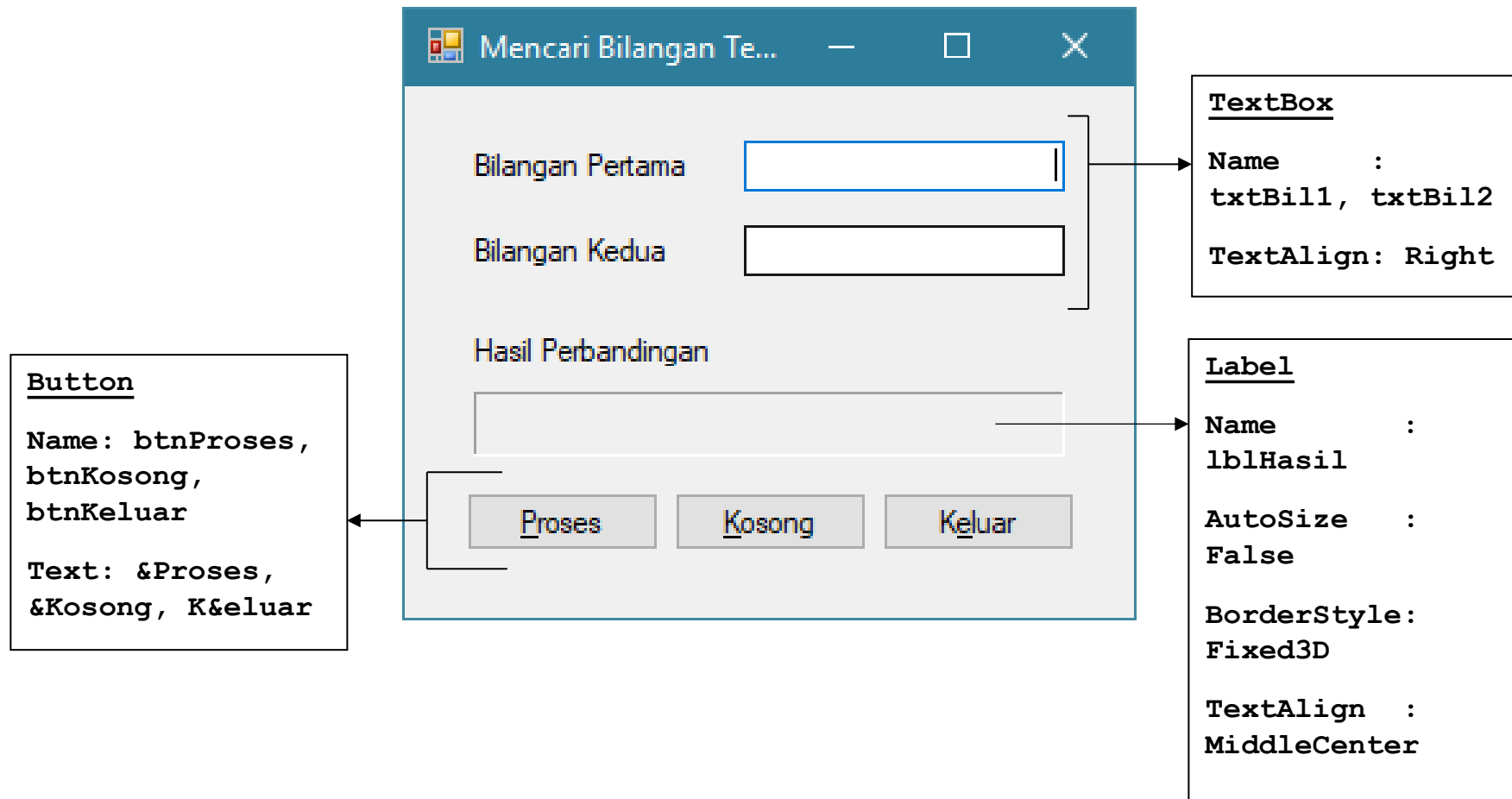
TextAlign: MiddleCenter

BackColor: DarkGreen (lblTglLengkap), GreenYellow

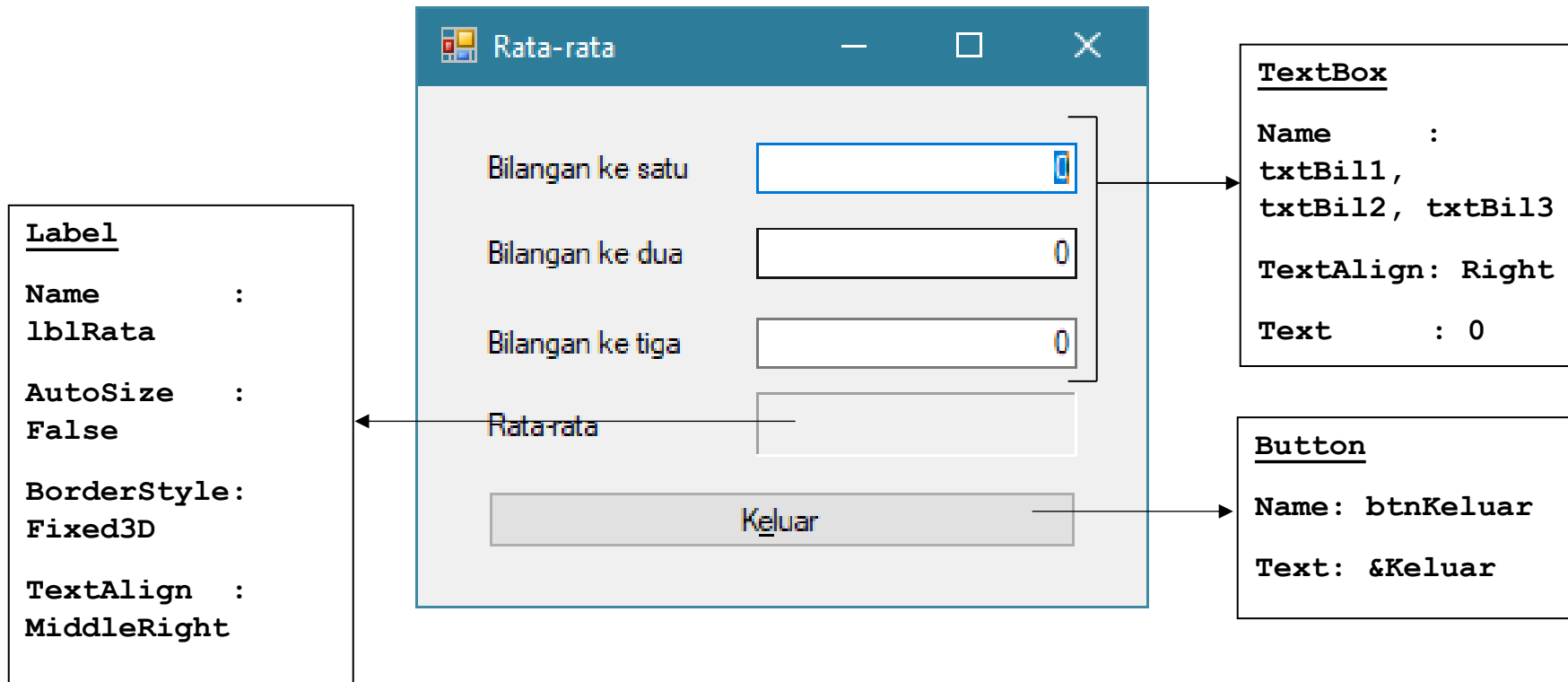
ForeColor: MintCream (lblTglLengkap), Green

Font: Bold, 10pt (lblTglLengkap)

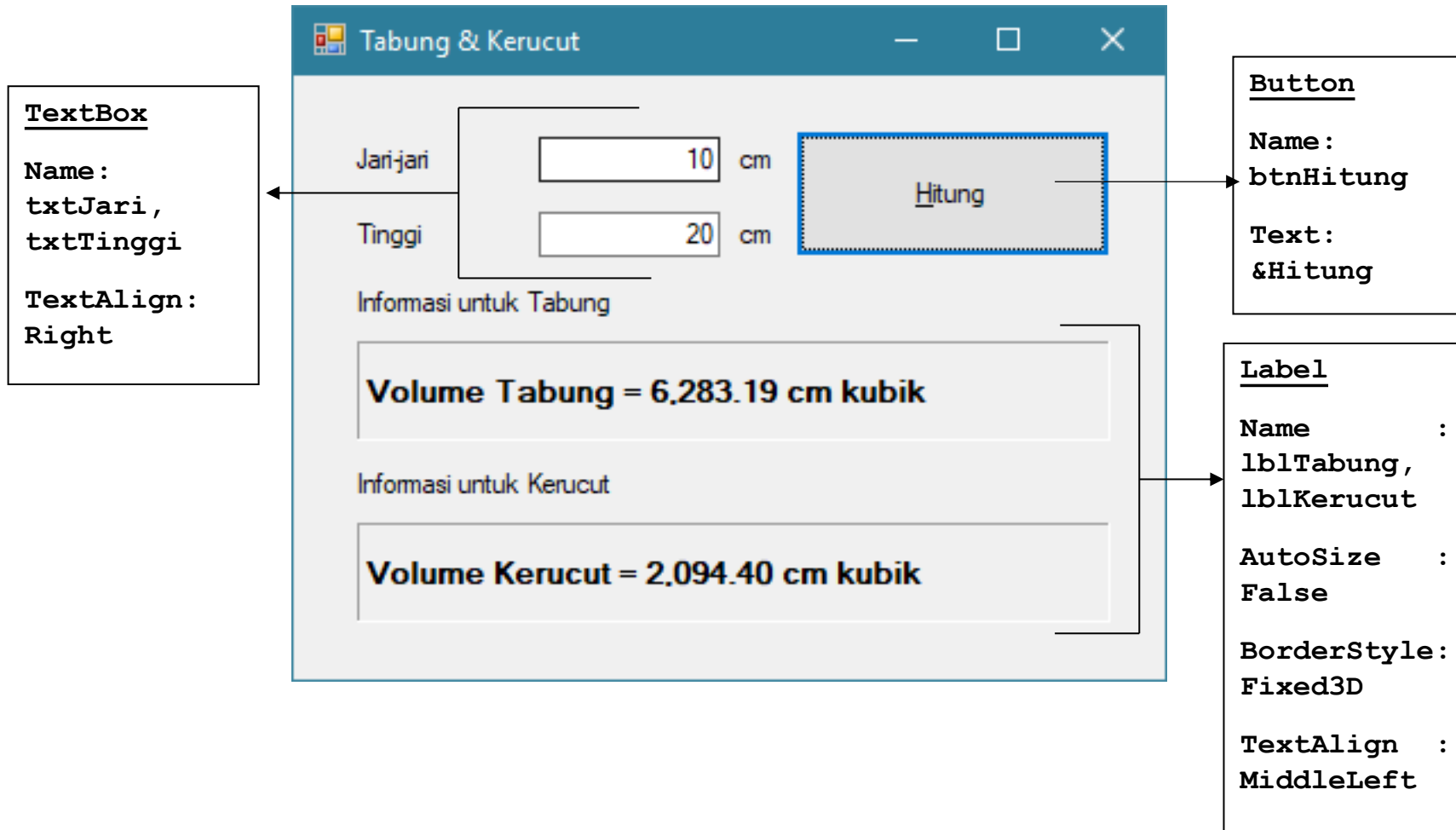
Latihan 3



Latihan 4



Latihan 5



Latihan 6

The screenshot shows a Windows application window titled "Pembelian". Inside the window, there are three input fields: "Harga Beli / Unit (Rp)" with the value "250000", "Jumlah Unit" with the value "5", and "Potongan (%)" with the value "10". To the right of the "Potongan (%)" field is a label "Rp 125,000.00". Below these fields is a button labeled "Hitung". At the bottom of the window is a label "Keterangan" with the text "Besaran Total dari Pembelian Anda adalah Rp 1,125,000.00".

TextBox

Name: txtHargaBeli, txtUnit, txtPotongan

TextAlign: Right

Button

Name: btnHitung

Text: &Hitung

Label

Name : lblPotongan, lblKeterangan

AutoSize : False

BorderStyle: Fixed3D

TextAlign : MiddleRight (lblPotongan), MiddleCenter (lblKeterangan)

Latihan 7

