

# Pengembangan Aplikasi Bisnis

TI2126

**By:**

**Angela, Handoko, Ivan, Joy, Arsyad**

Course Version: 2021

Universitas Mikroskil, Copyright ©2021

# COURSE OVERVIEW

## COURSE OVERVIEW

Pada mata kuliah ini, mahasiswa mempelajari tentang cara membuat program desktop; cara mengakses file; menggunakan menu, toolbar, dan status bar; melakukan koneksi ke database dan manipulasi data; serta membuat laporan database dengan menggunakan bahasa pemrograman Visual C#.

## COURSE GOALS

Mata kuliah ini akan mempersiapkan mahasiswa untuk:

- Mempraktikkan penggunaan kontrol dan membuat aplikasi windows
- Mengakses database dari program dan membuat laporan

## COURSE OBJECTIVES

Mata kuliah ini akan mempersiapkan mahasiswa untuk:

- Mempraktikkan penggunaan kontrol dan membuat aplikasi windows
- Mengakses database dari program dan membuat laporan





# UNIT 1

## MENJELAJAHI VISUAL C# TOOLBOX

### UNIT OVERVIEW

- Aplikasi windows dibangun dengan menggunakan fitur Visual C# yang memiliki Graphical User Interface (GUI). Pengguna dapat berinteraksi dengan seperangkat tools visual (seperti: button, textbox, toolbar, menu, dan lain-lain) untuk membuat aplikasi yang dapat bekerja sesuai dengan persyaratan dan kebutuhan pengguna.
- Dalam sebuah Program biasanya user dapat bekerja dengan pilihan-pilihan. Pilihan tersebut dapat menggunakan kontrol RadioButton maupun CheckBox. Kedua kontrol tersebut disebut sebagai kontrol 'point and click' yang memudahkan user untuk membuat pilihan dengan mengklik menggunakan mouse sehingga mengurangi tingkat error pada aplikasi. Pilihan tersebut dapat dikelompokkan berdasarkan kriteria tertentu pada kontrol GroupBox maupun Panel.
- Kontrol CheckBox memang berguna untuk memilih item dari suatu daftar. Namun, bagaimana kalau daftar tersebut terdiri dari 100 item? Apakah kita akan menggunakan 100 CheckBox? ListBox dan ComboBox merupakan kontrol yang dapat menyelesaikan permasalahan tersebut. ListBox dan ComboBox digunakan sebagai tempat untuk menampilkan daftar pilihan. Dengan ListBox dan ComboBox user dapat memilih pilihan yang ada pada ListBox maupun ComboBox tersebut.
- Dalam aplikasi berbasis windows form, terdapat banyak kontrol yang disediakan oleh Visual C#, seperti MessageBox dan DialogResult. MessageBox digunakan untuk menyajikan pesan kepada pengguna. DialogResult digunakan untuk berinteraksi dengan pengguna dan mengambil informasi.

### UNIT OBJECTIVES

Setelah menyelesaikan Unit ini, mahasiswa diharapkan mampu:

- Membedakan Object, Property, Method, dan Event.
- Mengetahui langkah-langkah pengembangan suatu aplikasi bisnis.
- Memahami dasar-dasar pengoperasian Microsoft Visual Studio 2019.
- 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.
- Memahami karakteristik dan penggunaan dari kontrol GroupBox, Panel, CheckBox, dan RadioButton.
- Memahami karakteristik dan penggunaan dari kontrol ListBox dan ComboBox.
- Memahami karakteristik dan penggunaan dari kontrol MessageBox.
- Memahami karakteristik dan penggunaan dari kontrol DialogResult.

### UNIT CONTENTS

Lesson: Object, Property, Method & Event .....	1-6
Lesson: Langkah-langkah Pengembangan Aplikasi Bisnis .....	1-7
Lesson: IDE Microsoft Visual Studio 2019 .....	1-8
Lesson: Mengenal Kontrol .....	1-21
Lesson: Form .....	1-26
Lesson: Button, Label, dan TextBox .....	1-28
Lesson: GroupBox, Panel, CheckBox, dan RadioButton.....	1-31



Lesson: ListBox dan ComboBox .....	1-35
Lesson: MessageBox .....	1-40
Lesson: DialogBox .....	1-44



## QUESTION

1. Pada saat kursor memasuki txtPanjang (event: \_\_\_\_\_), maka tulisannya akan berubah menjadi warna Biru. Kemudian pada saat kursor meninggalkan txtPanjang (event: \_\_\_\_\_), maka tulisannya akan kembali menjadi warna Hitam.
  - ☐ A. Leave, Enter
  - ☐ B. KeyUp, KeyDown
  - ☐ C. Enter, Leave
  - ☐ D. TextChanged, Leave
  - ☐ E. TextChanged, TextChanged
2. Untuk membatasi agar pengguna hanya boleh memasukkan angka ke dalam TextBox, kita bisa melakukan coding pada event?
  - ☐ A. TextChanged
  - ☐ B. KeyDown
  - ☐ C. Input
  - ☐ D. KeyPress
  - ☐ E. Enter
3. Manakah di bawah ini property yang digunakan untuk mengatur warna tulisan?
  - ☐ A. Font
  - ☐ B. BackColor
  - ☐ C. ForeColor
  - ☐ D. Text
  - ☐ E. BorderStyle
4. Manakah di bawah ini event pada TextBox yang terjadi ketika TextBox kehilangan fokus?
  - ☐ A. Enter
  - ☐ B. Click
  - ☐ C. KeyPress
  - ☐ D. Leaver
  - ☐ E. TextChanged
5. Property yang digunakan untuk mengganti Nama File dari Form adalah?
  - ☐ A. Name
  - ☐ B. FileName
  - ☐ C. Text
  - ☐ D. Font
  - ☐ E. TextAlign
6. Manakah di bawah ini pernyataan yang benar?
  - ☐ A. Panel memiliki fungsi yang sama dengan GroupBox yaitu untuk mengelompokkan kontrol
  - ☐ B. Panel dan GroupBox bukan merupakan kontrol
  - ☐ C. Panel memiliki BorderStyle, sedangkan GroupBox tidak
  - ☐ D. Panel tidak memiliki property Text untuk memberi Judul, sedangkan GroupBox punya
  - ☐ E. Panel dapat mengelompokkan kontrol, sedangkan GroupBox tidak dapat mengelompokkan kontrol



7. Event ketika pengguna memilih RadioButton dan CheckBox adalah?
- ☐ A. Click
  - ☐ B. Checked
  - ☐ C. CheckChanged
  - ☐ D. TextChanged
  - ☐ E. Enter
8. Untuk menyimpan (Save) dan membuka (Open) file dengan memilih lokasi penyimpanan file yang diinginkan, kita dapat menggunakan?
- ☐ A. Open File & Save File
  - ☐ B. SaveFile dan OpenFile
  - ☐ C. SaveFileDialog dan OpenFileDialog
  - ☐ D. OpenSaveFileDialog
  - ☐ E. SaveOpenFileDialog
9. Untuk menghapus item di dalam ListBox, kita dapat menggunakan perintah?
- ☐ A. Remove
  - ☐ B. Text = "";
  - ☐ C. RemoveAt
  - ☐ D. Clear
  - ☐ E. Text = 0;
10. Manakah di bawah ini pernyataan yang benar?
- ☐ A. CheckBox digunakan untuk membuat serangkaian pilihan dimana user dapat memilih 1 atau lebih dari pilihan-pilihan yang ada
  - ☐ B. RadioButton digunakan untuk membuat serangkaian pilihan dimana user hanya dapat memilih 1 pilihan dari berbagai pilihan yang ada
  - ☐ C. Perbedaan ComboBox dan ListBox terdapat pada pilihan item. Pada ComboBox hanya satu item pilihan yang terlihat sebelum ComboBox diklik, sedangkan pada ListBox dapat menampilkan beberapa pilihan
  - ☐ D. Perbedaan ComboBox dan ListBox terdapat pada Event-nya
  - ☐ E. ComboBox dan ListBox penggunaan property dan eventnya hampir sama
11. MessageBox dapat digunakan untuk?
- ☐ A. Mengatasi error yang ada
  - ☐ B. Menampilkan pesan dan icon
  - ☐ C. Menampilkan pesan error
  - ☐ D. Menjelaskan masalah potensial
  - ☐ E. Menambah error
12. Cara untuk membersihkan RadioButton pada form di bawah ini adalah?
- ☐ A. Atur Properties Checked menjadi true
  - ☐ B. Atur Properties Text menjadi ""
  - ☐ C. Atur Properties Checked menjadi ""
  - ☐ D. Pakai metode Clear()
  - ☐ E. Atur Properties Checked menjadi false
13. GroupBox digunakan untuk?
- ☐ A. Mengelompokkan sekumpulan Event dalam aplikasi Visual C#
  - ☐ B. Mengelompokkan sekumpulan kontrol yang saling berhubungan dalam aplikasi Visual C#



- ☐ C. Mengelompokkan beberapa kontrol yang saling berhubungan dalam aplikasi Visual C#
  - ☐ D. Menggabungkan kontrol yang berbeda dalam aplikasi Visual C#
  - ☐ E. Memisahkan sekumpulan kontrol yang saling berhubungan dalam aplikasi Visual C#
14. Untuk menggunakan MessageBox, programmer harus menentukan ?
- ☐ A. Caption yang berisi judul MessageBox
  - ☐ B. Icon yang tidak sesuai
  - ☐ C. Text yang berisi pesan yang ingin disampaikan
  - ☐ D. Button yang berisi pilihan tombol yang tampil pada MessageBox
  - ☐ E. Default Button untuk menentukan tombol mana yang dijadikan default
15. DialogResult yang menyediakan fitur bagi user untuk mengganti tipe, style, ukuran, dan warna font adalah ?
- ☐ A. SaveFileDialog
  - ☐ B. OpenFileDialog
  - ☐ C. FontDialog
  - ☐ D. ColorDialog
  - ☐ E. PrintDialog

## CONTENT LESSON

### CASE STUDY / PROJECT

#### SKENARIO

Mira merupakan wanita yang terlahir dari keluarga pecinta kopi. Kopi menjadi minuman favorit Mira dan keluarga, sehingga stok kopi selalu ada dan tidak pernah kehabisan. Singkat cerita, Mira memiliki ide untuk membuka Coffee Shop di depan rumahnya. Setelah Coffee Shop dibuka, berawal dari sepi pembeli tak disangka dalam waktu dua minggu Coffee Shop milik Mira ramai pembeli. Banyak dari pembeli yang meminta menu baru dari kopi dan menggunakan topping agar tidak seperti kopi pada umumnya. Mira memilih untuk menyanggupi permintaan pembeli. Sebelumnya Mira masih mencatat penjualan di buku setiap harinya. Karena Coffee Shop milik Mira semakin ramai, Mira kewalahan dalam mencatat data pelanggan, Menu & Topping yang dipesan, jumlah pesanan, makan di tempat atau bawa pulang, dan menghitung total penjualan.

#### IDENTIFICATION CONCEPT OF PROBLEM / PROJECT

Masalah yang terlihat dari cerita di atas adalah kewalahan untuk mencatat data pelanggan, Menu & Topping yang dipesan, jumlah pesanan, makan di tempat atau bawa pulang, dan menghitung total penjualan karena masih mencatat penjualan di buku setiap harinya. Mira membutuhkan sebuah aplikasi sederhana yang dapat meng-handle penjualan Coffee Shop miliknya. Objek/Kontrol yang perlu ada pada aplikasi yang diinginkan Mira adalah:

- **Form** berguna sebagai objek container yang menampung beberapa kontrol serta dapat digunakan untuk menampilkan informasi dan menerima input dari pengguna.
- **Button** digunakan untuk mengeksekusi suatu aksi dengan cara mengklik.
- **Label** digunakan untuk menampilkan teks atau gambar yang tidak dapat diedit oleh user.
- **TextBox** digunakan untuk mendapatkan input dari user atau menampilkan teks.
- **GroupBox** atau **Panel** digunakan untuk mengelompokkan sekumpulan kontrol yang saling berhubungan dalam aplikasi Visual C#.
- **CheckBox** digunakan untuk membuat serangkaian pilihan dimana user dapat memilih 1 atau lebih dari pilihan-pilihan yang ada.





- **RadioButton** digunakan untuk membuat serangkaian pilihan dimana user hanya dapat memilih 1 pilihan yang ada.
- **ListBox** digunakan untuk menampilkan daftar dari sejumlah item atau pilihan menu dimana user dapat memilih 1 atau lebih item.
- **ComboBox** memungkinkan user memilih 1 item dari daftar; dan dalam beberapa kasus, memungkinkan user untuk mengetikkan pilihan alternatif.
- **MessageBox** digunakan untuk menyajikan pesan kepada pengguna.
- **DialogBox** digunakan untuk berinteraksi dengan pengguna dan mengambil informasi.

## LESSON 1: OBJECT, PROPERTY, METHOD & EVENT

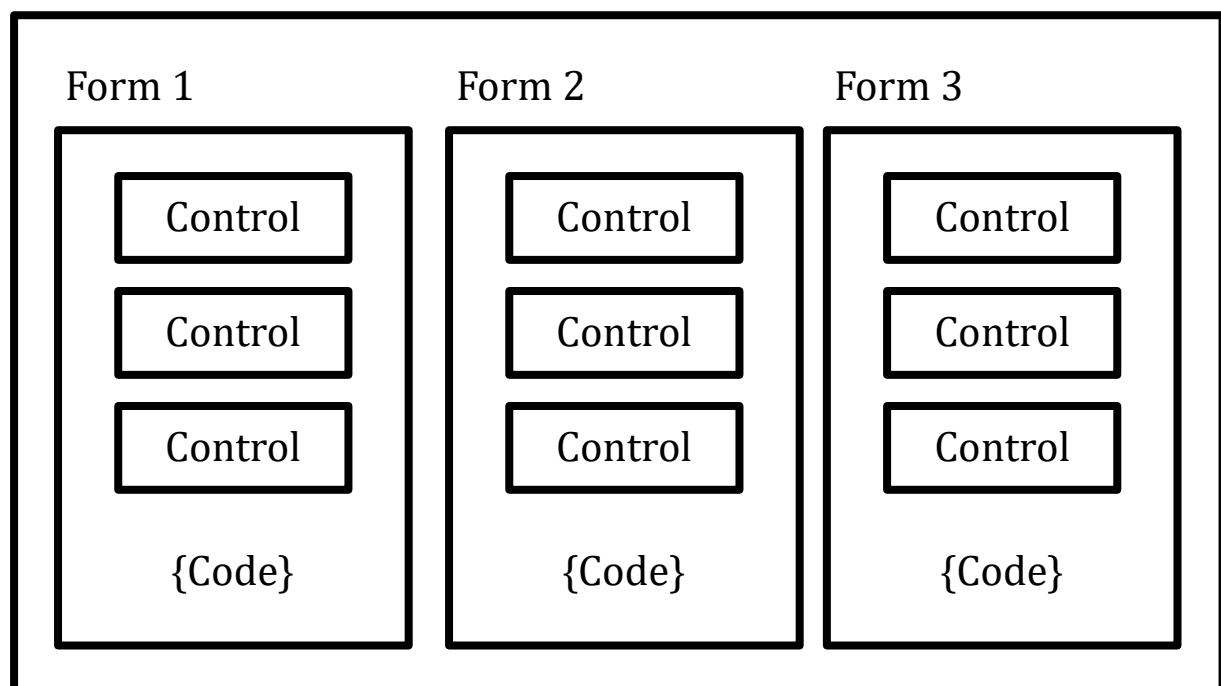
### LESSON 1: OVERVIEW

Aplikasi windows Visual C# bersifat event-driven, yang berarti aplikasi tidak akan melakukan apa pun sampai aplikasi tersebut dipanggil untuk merespon suatu event (mis: menekan tombol, memilih menu, dst.).

### TOPIC 1: PROJECT

Aplikasi (Project) terdiri dari:

- **Objects**
  - **Forms:** windows yang dirancang sebagai antarmuka aplikasi
  - **Controls:** fitur grafis yang dimasukkan dalam form untuk memungkinkan interaksi pengguna (mis: textbox, label, button, dll.)
- **Properties:** setiap karakteristik dari form atau control. Contoh: names, captions, size, color, position, dll. Dalam Visual C#, sudah terdapat nilai default pada property. Nilai pada property dapat diganti ketika mendesain aplikasi ataupun dengan coding.
- **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.



Gambar 1.1: Project dalam Visual C#

## TOPIC 2: OBJECT, PROPERTY, METHOD & EVENT

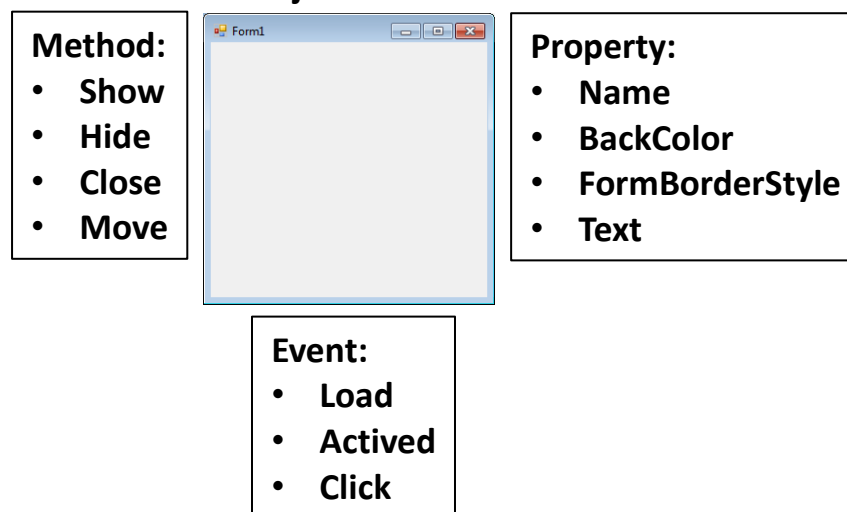
- Object (Objek): komponen di dalam sebuah program.
- Property (Properti): sifat atau karakteristik yang dimiliki objek.
- Methods (Metoda): aksi, tindakan, atau perbuatan yang dapat dilakukan oleh objek.
- Event (Kejadian): kejadian yang dapat diterima atau dialami oleh objek (biasanya dilakukan oleh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

### Object: Mobil



Gambar 1.2: Project dalam Visual C#

### Object: Form



Gambar 1.3: Ilustrasi Property, Method & Event pada Object Form

## LESSON 2: LANGKAH-LANGKAH PENGEMBANGAN APLIKASI BISNIS

### LESSON 2: OVERVIEW

Ketika pengembangan sistem untuk penyelesaian masalah diterapkan untuk pengembangan solusi sistem informasi terhadap masalah bisnis, maka hal ini disebut Information Systems Development (pengembangan sistem informasi) atau Application Development (pengembangan aplikasi).

---

## TOPIC 1: LANGKAH-LANGKAH PENGEMBANGAN APLIKASI BISNIS

Dalam konsep bisnis, perlu dilakukan analisis terhadap kasus dan permasalahan yang terjadi di dalam perusahaan. Salah satu solusi untuk menyelesaikan masalah yang terjadi dalam perusahaan adalah membangun aplikasi bisnis yang mampu menjawab permasalahan yang ada.

Adapun langkah-langkah dalam membangun suatu aplikasi untuk menyelesaikan masalah dalam kasus bisnis antara lain:

- Merancang *user interface* dengan memasukkan kontrol pada form.
- Mengatur *property* yang dibutuhkan pada kontrol.
- Menulis kode program berdasarkan *event* kontrol.

---

## LESSON 3: IDE MICROSOFT VISUAL STUDIO 2019

---

### LESSON 3: OVERVIEW

Beberapa fitur dari Visual C#:

- Integrated Development Environment (IDE) yang mudah digunakan
- Kontrol yang lengkap
- Merespon terhadap aksi mouse dan keyboard
- Akses ke clipboard dan printer
- Fungsi matematika, string, dan grafik yang lengkap
- Dapat bekerja dengan sekumpulan array, variabel dan objek
- Mendukung sequential file
- Fasilitas debugger dan penanganan kesalahan
- Tools grafik yang mudah digunakan
- Tools akses database yang berguna
- Kemampuan untuk mengembangkan aplikasi Windows dan internet dengan teknik yang sama
- Module common language runtime baru membuat distribusi aplikasi menjadi hal yang mudah

---

## TOPIC 1: ANTARMUKA PROGRAM

Ketika membangun aplikasi bisnis, pengembang aplikasi harus mempertimbangkan beberapa hal antara lain:

- Antarmuka program, yang terdiri dari:
  - Windows Form.
  - Web Service dan Web Forms (ASP .NET).
- Bahasa pemrograman yang dapat digunakan untuk membangun aplikasi .NET: APL (A Programming Language), Visual C#, Cobol, Component Pascal, Curriculum, Eiffel, Forth, Fortran (Formula Translator), Haskell, Java, Jscript, Mercury, Mondrian, Oberon, Oz, Pascal, Perl, Python, RPG, Scheme, Smalltalk, Standar ML, Visual Basic, dan Visual C++.

---

### Catatan

Dalam mengembangkan aplikasi bisnis, pengembang harus memperhatikan fungsionalitas, desain, serta ketepatan dalam pemilihan kontrol yang digunakan dalam aplikasi

---

## TOPIC 2: IDE MICROSOFT VISUAL STUDIO 2019

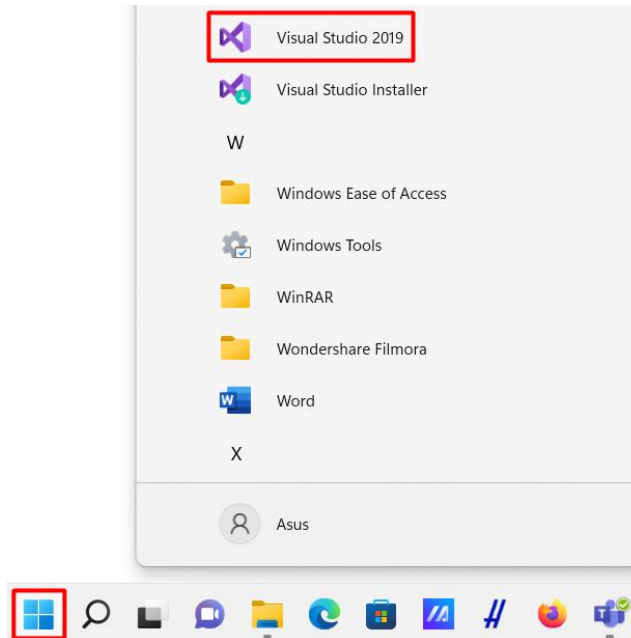
IDE dapat dikatakan sebagai lingkungan yang berisi tools maupun fitur dimana pengembang aplikasi dapat membangun, menjalankan, dan menguji program.



## MULAI MENGGUNAKAN IDE MICROSOFT VISUAL STUDIO 2019

Untuk mulai mengaktifkan IDE dari Microsoft Visual Studio 2019 dapat dilakukan melalui menu start.

- Pilih All Apps
- Kemudian cari Visual Studio 2019
- Klik pada Visual Studio 2019

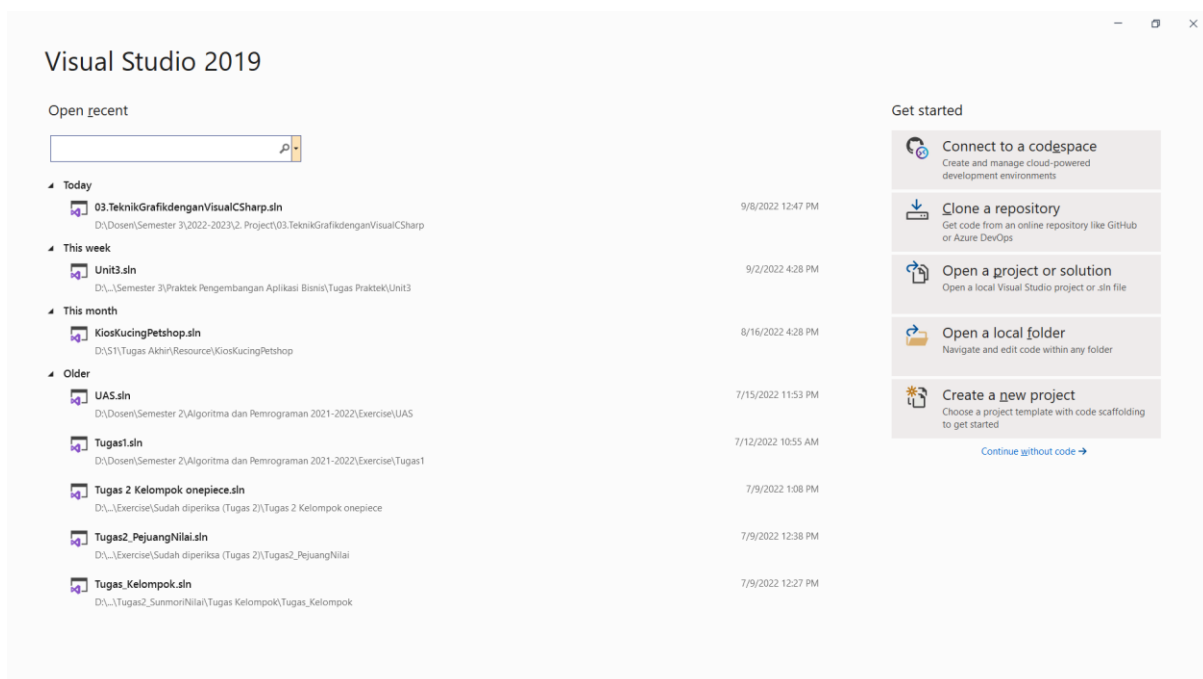


### Catatan

IDE merupakan singkatan dari Integrated Development Environment

Gambar 1.4: Memulai Mengaktifkan IDE Microsoft Visual Studio 2019

## TAMPILAN HALAMAN AWAL



Gambar 1.5: Tampilan Awal IDE Microsoft Visual Studio 2019



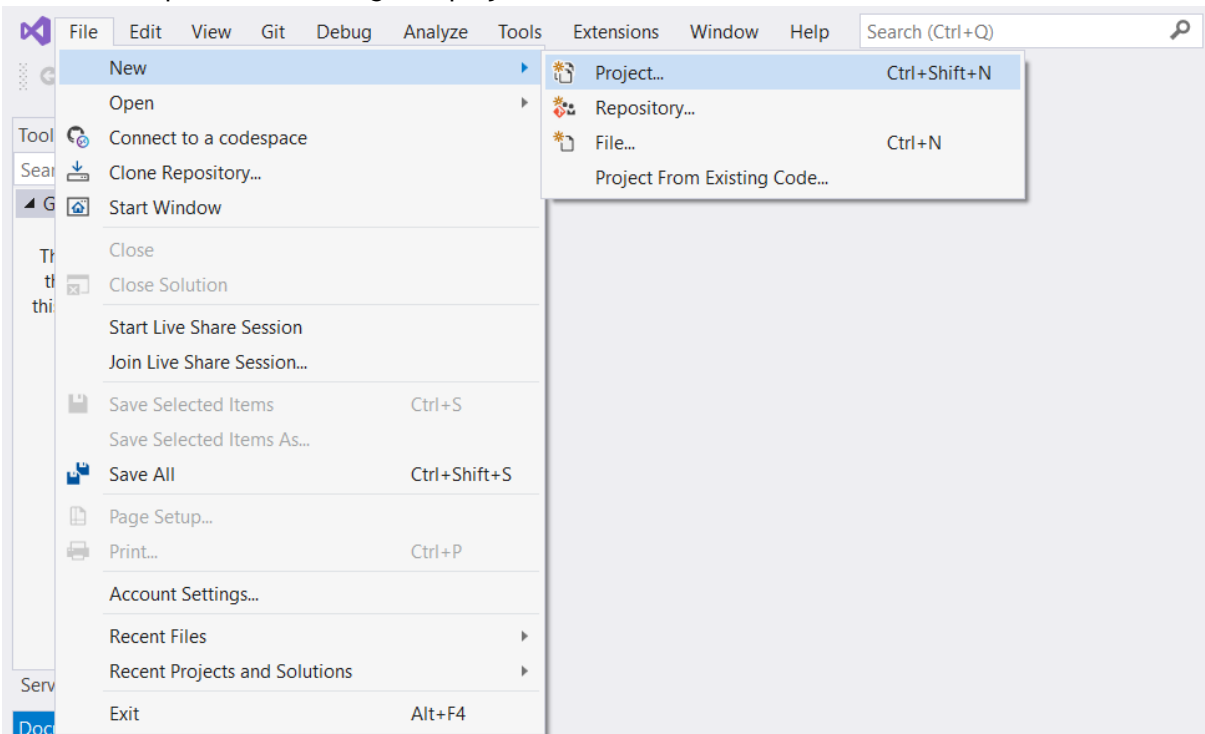
Ada banyak informasi yang berguna di halaman awal (start page):

1. Pada bagian kiri, terdapat fitur untuk menampilkan project-project yang terakhir kali dibuka.
2. Pada bagian kanan terdapat fitur untuk:
  - Connect to a codespace: membuat dan mengelola lingkungan pengembangan yang didukung cloud
  - Clone a repository: mendapatkan kode dari repositori online seperti GitHub atau Azure DevOps
  - Open a project or solution: membuka project Visual Studio lokal atau file .sln
  - Open a local folder: membuka dan mengedit kode di dalam folder mana pun
  - Create a new project: membuat project baru dengan memilih project template

---

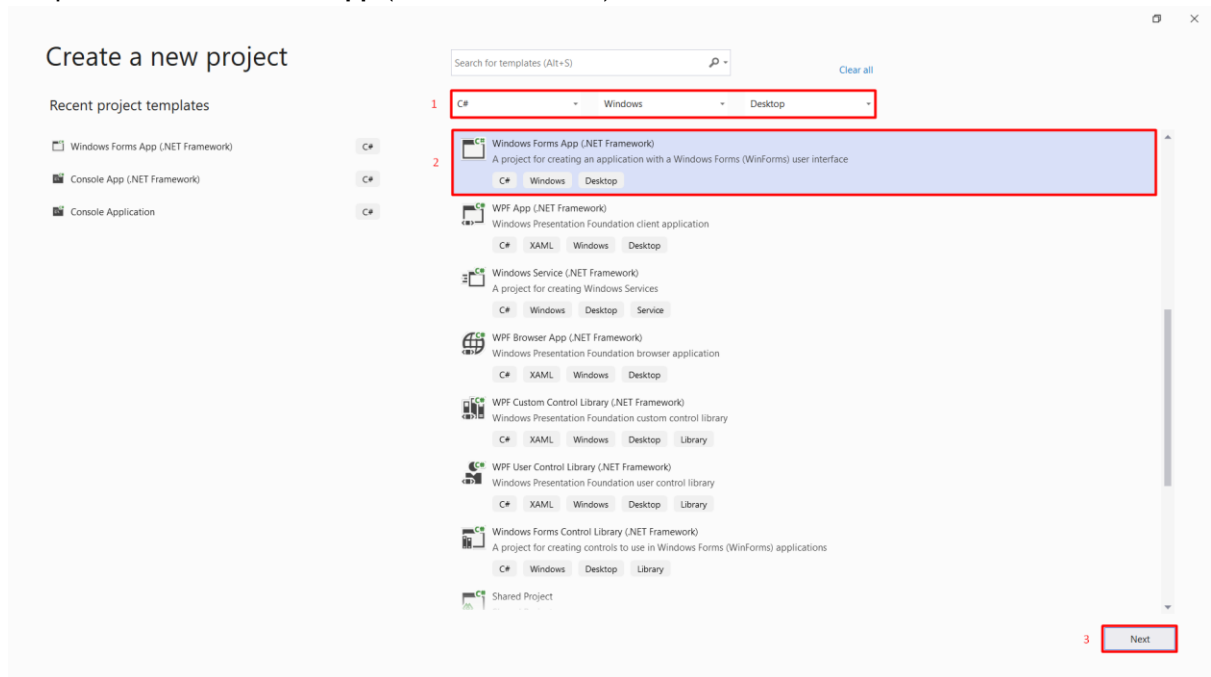
## MEMBUAT PROJECT BARU

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



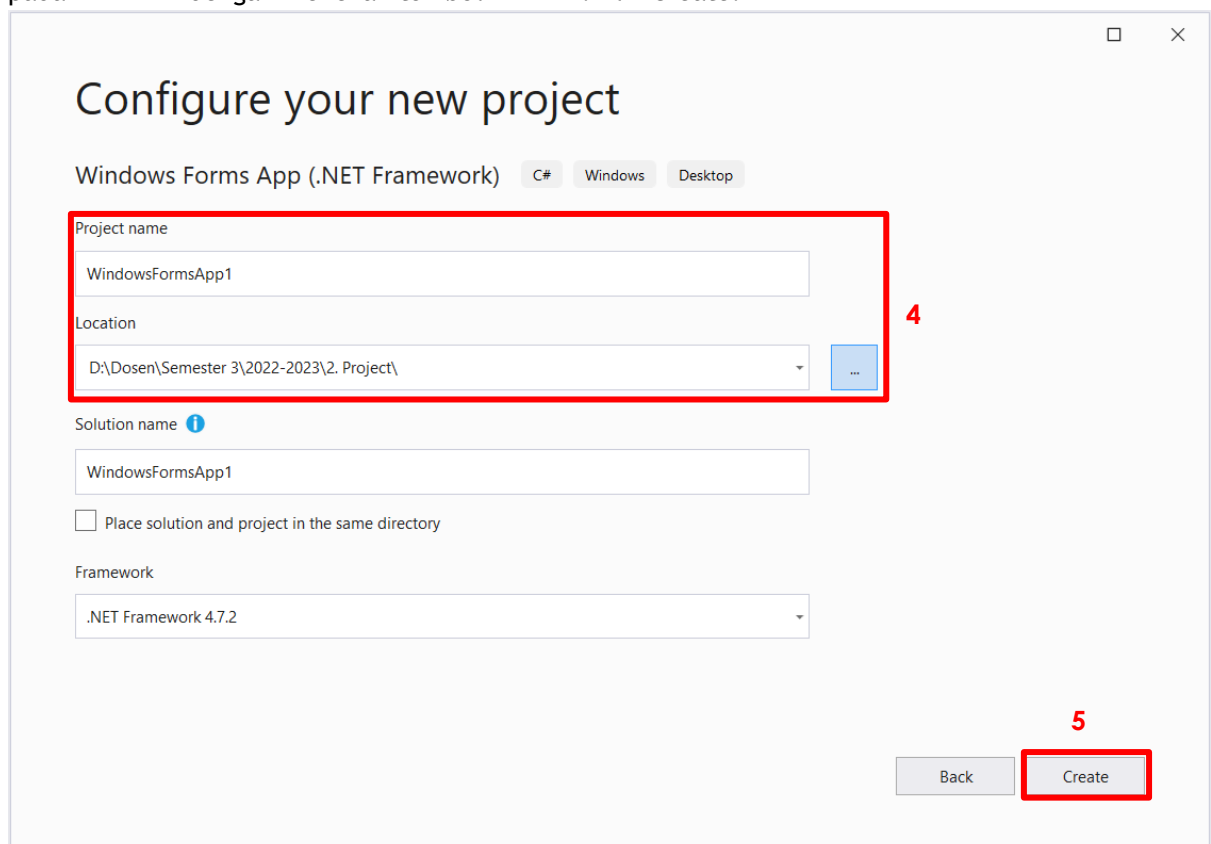
**Gambar 1.6: Memulai Project Baru**

2. Tentukan bahasa pemrograman **C#**, platform **Windows**, dan jenis project **Desktop** → pilih template **Windows Forms App (.NET Framework)**. klik **Next**.



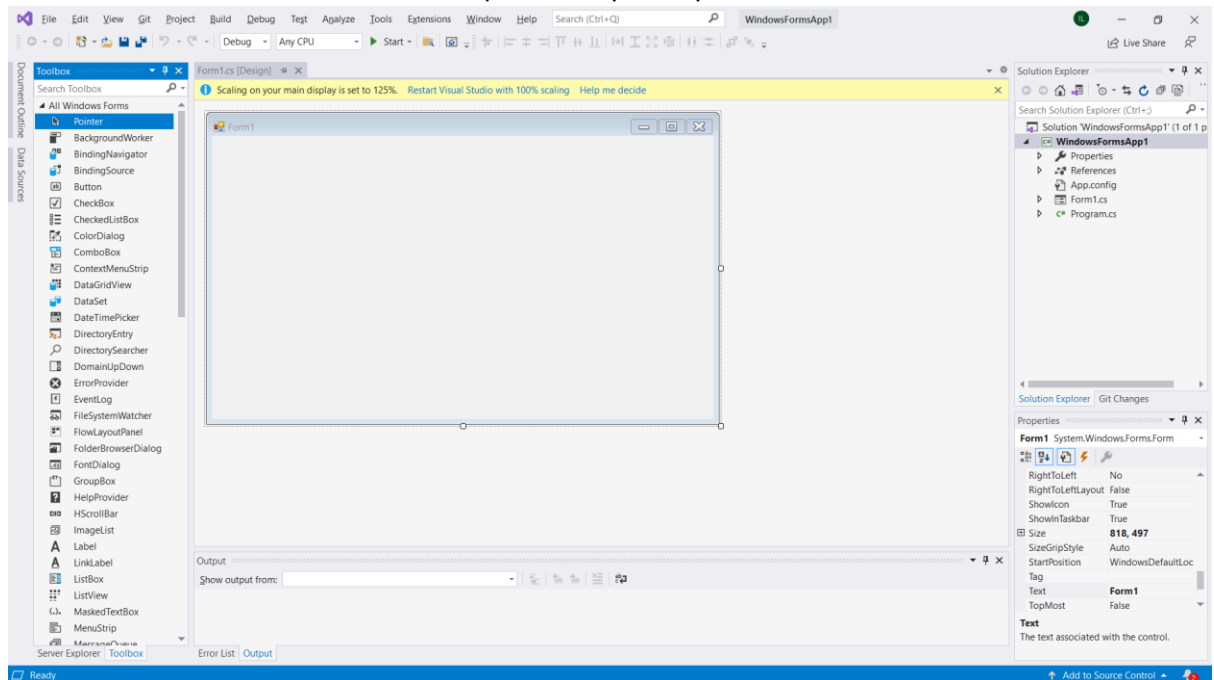
**Gambar 1.7: Membuat Project Baru**

3. Pada kotak input **Project name**, ketikkan nama project. Masukkan tempat penyimpanan project pada **Location** dengan menekan tombol **Browse**. Klik **Create**.



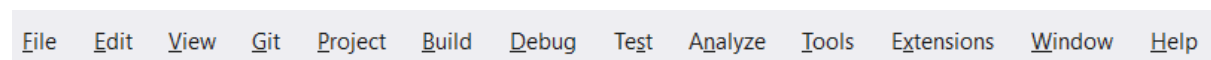
**Gambar 1.8: Mengonfigurasi Project Baru**

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



**Gambar 1.9: Template Aplikasi Windows Form**

## MENU BAR



**Gambar 1.10: Menu Bar**

**Tabel 1.1: Menu dalam Visual Studio 2019**

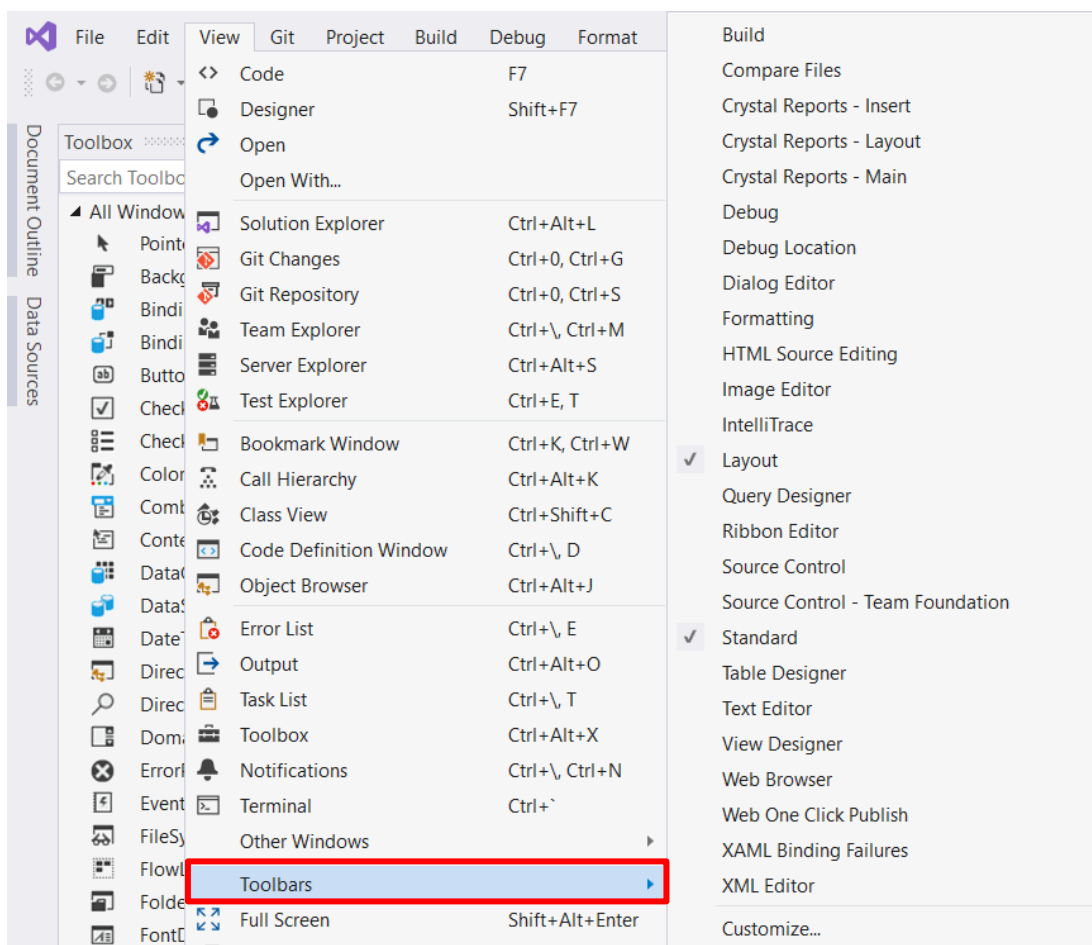
Menu	Fungsi
File	Digunakan untuk membuka / menutup project dan file, juga untuk keluar dari Visual Studio
Edit	Digunakan ketika menulis kode program untuk melakukan pengeditan seperti copy, cut, paste, dan menghapus teks
View	Menyediakan akses ke berbagai windows dalam IDE
Git	Digunakan untuk berinteraksi dengan folder dan file repositori
Project	Digunakan untuk menambah / menghapus komponen dari project
Build	Mengontrol proses kompilasi
Debug	Membantu melacak kesalahan / error pada kode program
Test	Digunakan untuk mengkompilasi dan menjalankan aplikasi yang lengkap
Analyze	Melakukan analisis terhadap kode program



Tools	Memungkinkan konfigurasi kustom terhadap IDE. Submenu Options paling banyak dipakai untuk memodifikasi IDE sesuai dengan kebutuhan personal masing-masing orang.
Window	Digunakan untuk mengubah layout windows dalam IDE
Extensions	Digunakan untuk menyesuaikan ekstensi
Help	Menyediakan akses ke dokumentasi online Visual C# melalui konten bantuan, indeks ataupun pencarian.

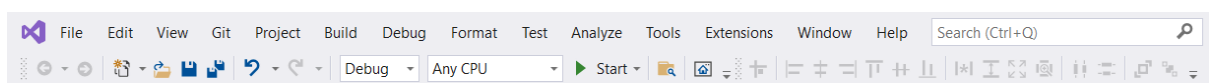
## TOOLBAR

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



**Gambar 1.11: Tampilan Menu View**

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



**Gambar 1.12: Standard Toolbar**

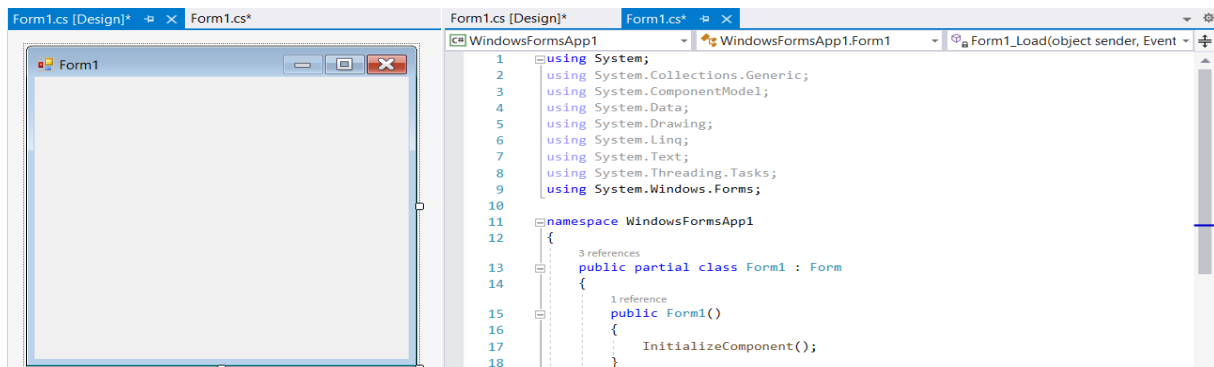


Apabila kita tidak mengetahui fungsi tombol toolbar, kita dapat mengarahkan mouse ke tombol tersebut dan tooltip yang berisi keterangan tombol akan muncul.

## FORM

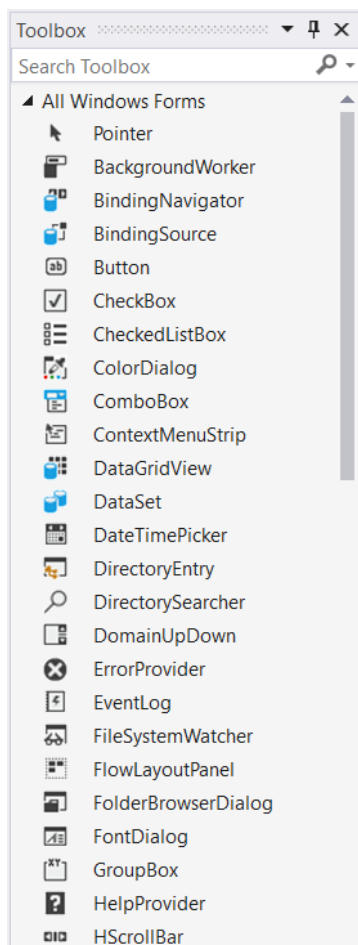
Merupakan jendela utama pengembangan Visual C# yang terdiri dari dua bagian yaitu:

- Layar design (Form Designer)
- Layar penulisan kode program (Code Editor)



**Gambar 1.13: Form - Layar Design (Kiri) dan Layar Penulisan Kode Program (Kanan)**

## TOOLBOX



**Gambar 1.14: Toolbox**



Toolbox berisi kontrol-kontrol yang digunakan dalam aplikasi yang berbasis graphical user interface. Toolbox hanya aktif ketika form ditampilkan dalam layar design (Form Designer).

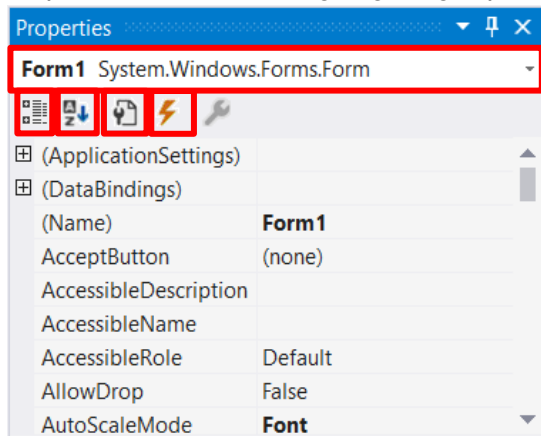
Untuk menambahkan kontrol dari Toolbox ke form dapat digunakan dengan cara:

- Klik pada kontrol yang diinginkan, lalu klik atau gambarkan pada form.
- Drag kontrol dari toolbox.
- Klik ganda pada kontrol yang diinginkan.

---

## PROPERTIES WINDOW

Properties Window digunakan untuk 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.

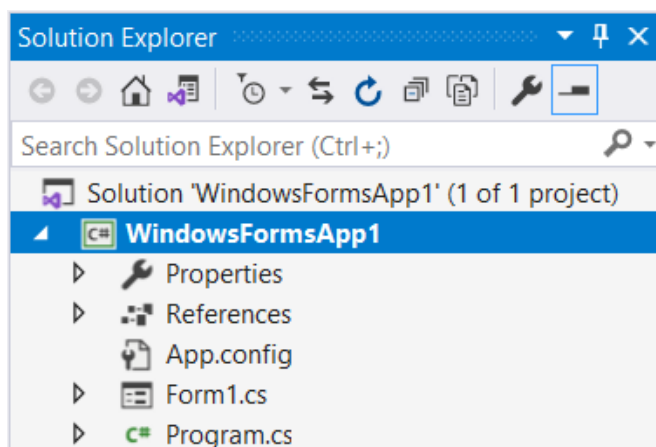


**Gambar 1.15: Properties Window**

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 alphabet 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

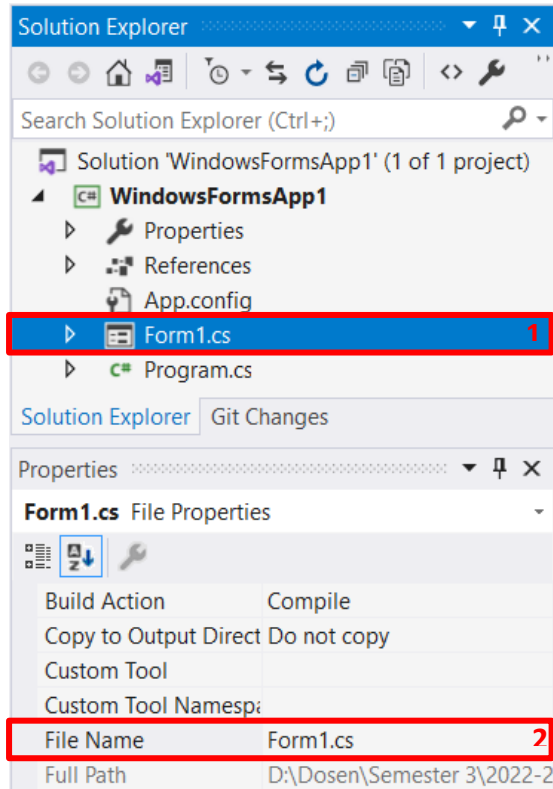


**Gambar 1.16: Jendela Solution Explorer**

Jendela 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

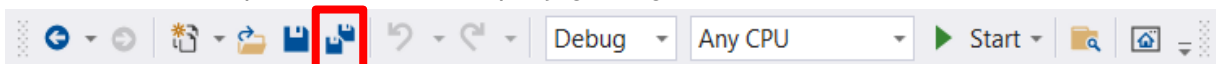
Untuk mengganti nama file, klik Form1 di jendela **Solution Explorer**. Selanjutnya nama file akan ditampilkan di jendela **Properties**. Ganti properti **File Name** dengan nama file yang diinginkan. Alternatif: klik kanan pada Form1 di jendela **Solution Explorer**, pilih **Rename**, lalu ganti nama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



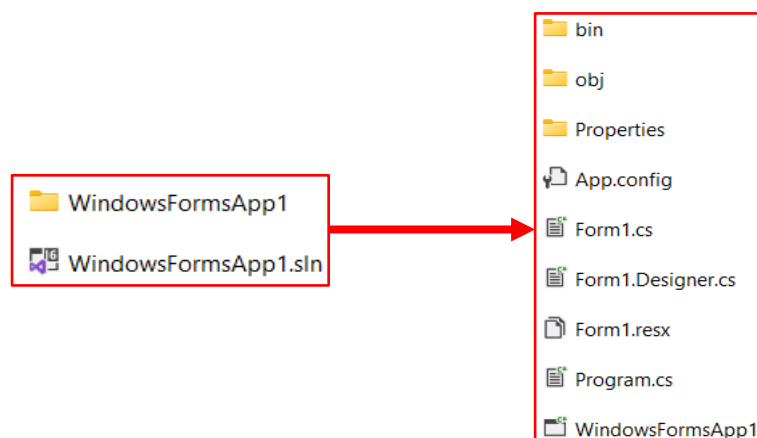
Gambar 1.17: Mengganti Nama File

## MENYIMPAN PROJECT

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



Gambar 1.18: Tombol Save All pada Standard Toolbar



Gambar 1.19: 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**.

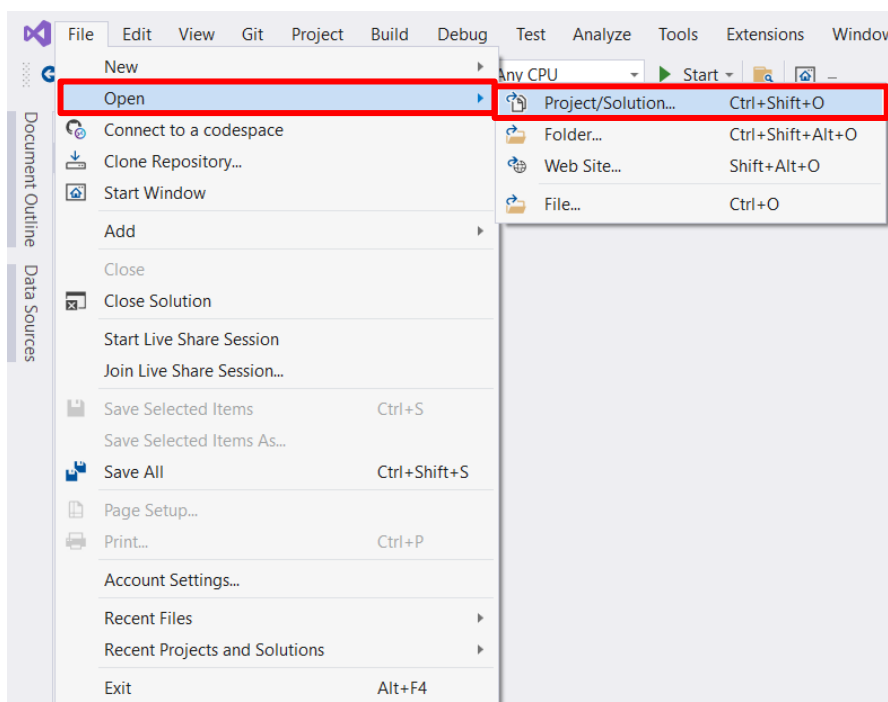
**Tabel 1.2: Struktur Project**

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

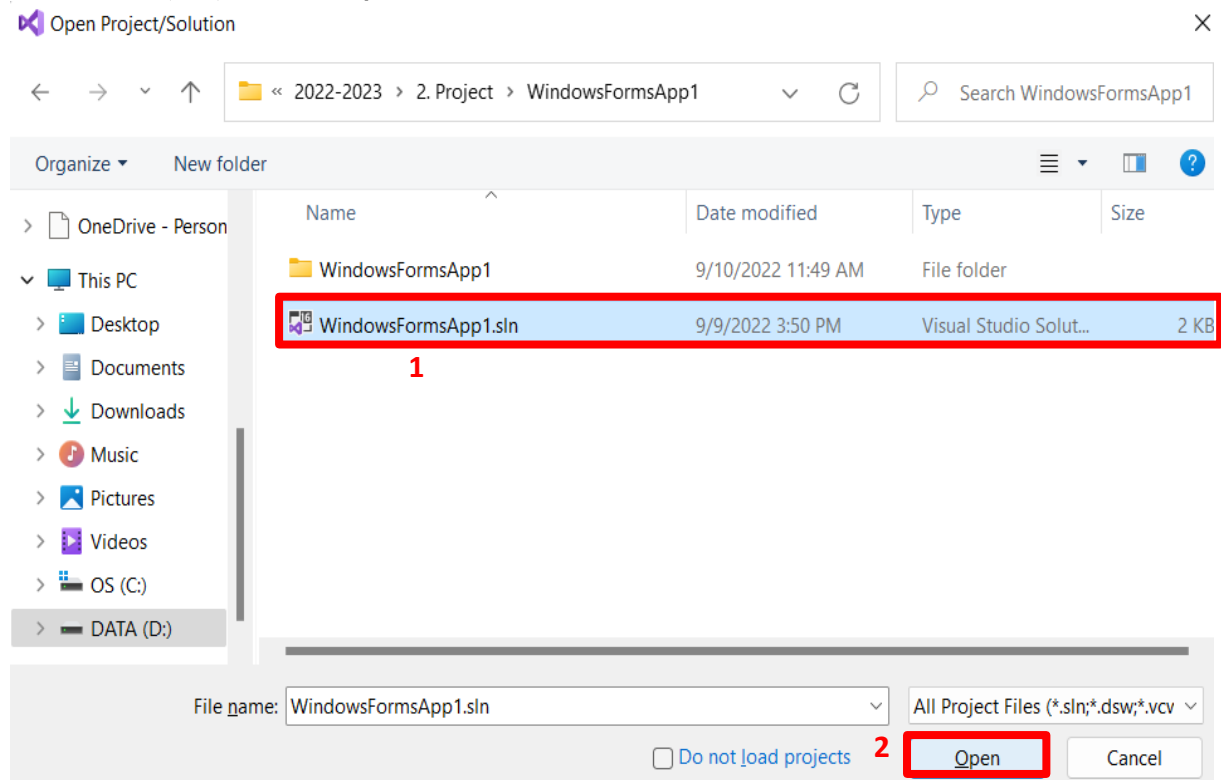
Langkah-langkah untuk membuka project yang sudah tersimpan:

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



**Gambar 1.20: Membuka Saved Project**

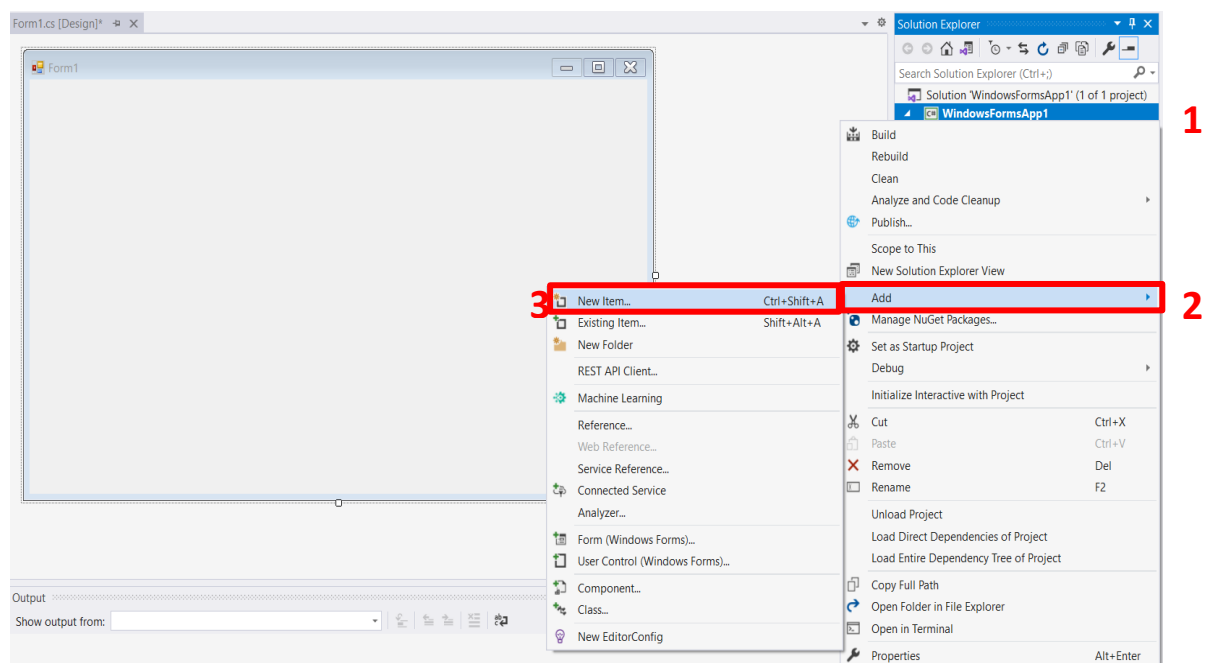
2. Kotak dialog Open Project akan terbuka. Cari lokasi penyimpanan project. Kemudian pilih file Solution (.sln), dan klik **Open**.



Gambar 1.21: Kotak Dialog Open Project

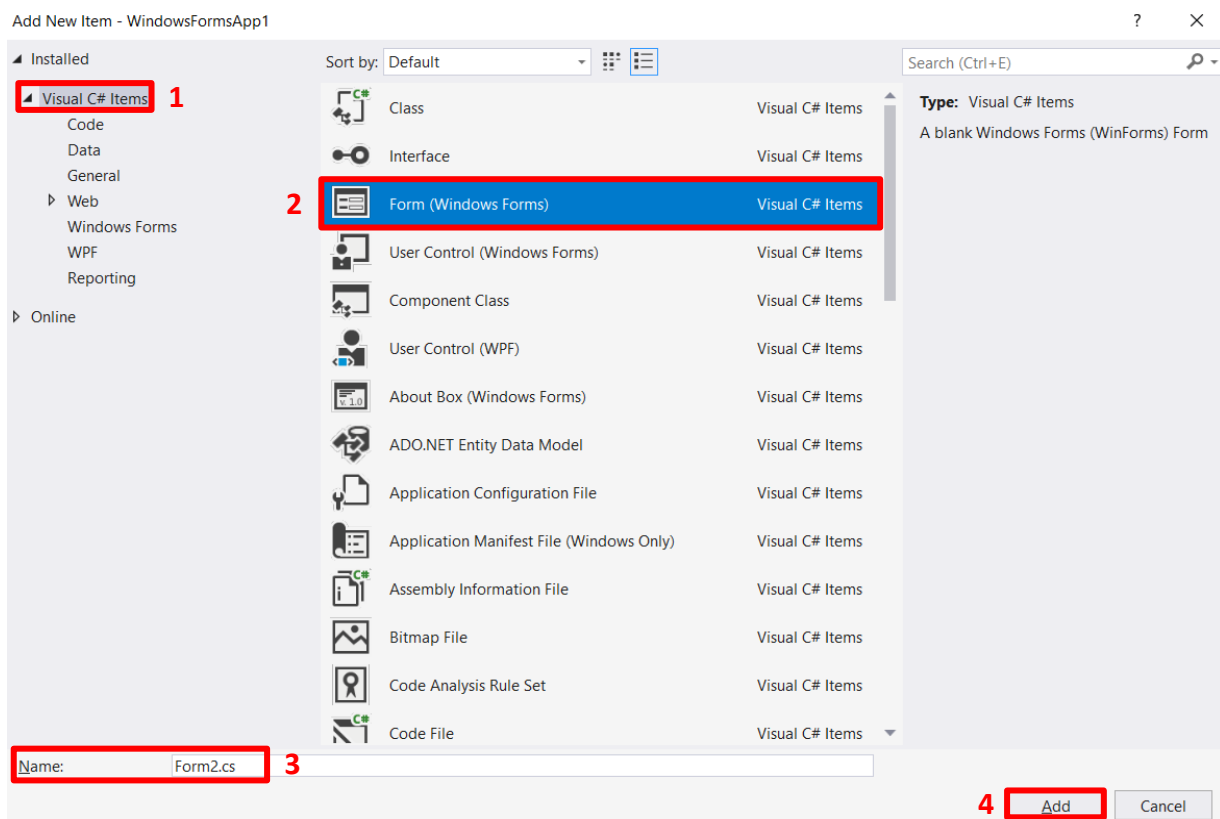
## MENAMBAHKAN ITEM BARU

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



Gambar 1.22: Menambahkan Item Baru

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

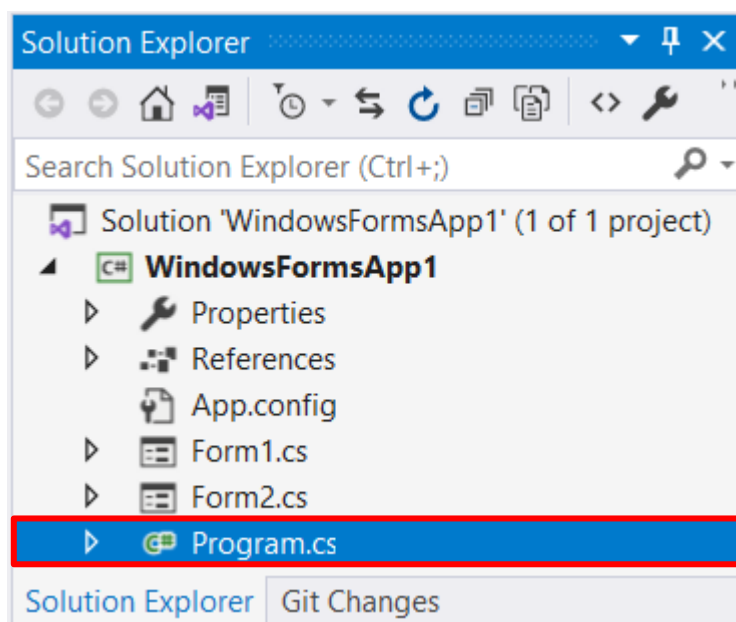


Gambar 1.23: Kotak Dialog Add New Item

## MEMILIH FORM STARTUP

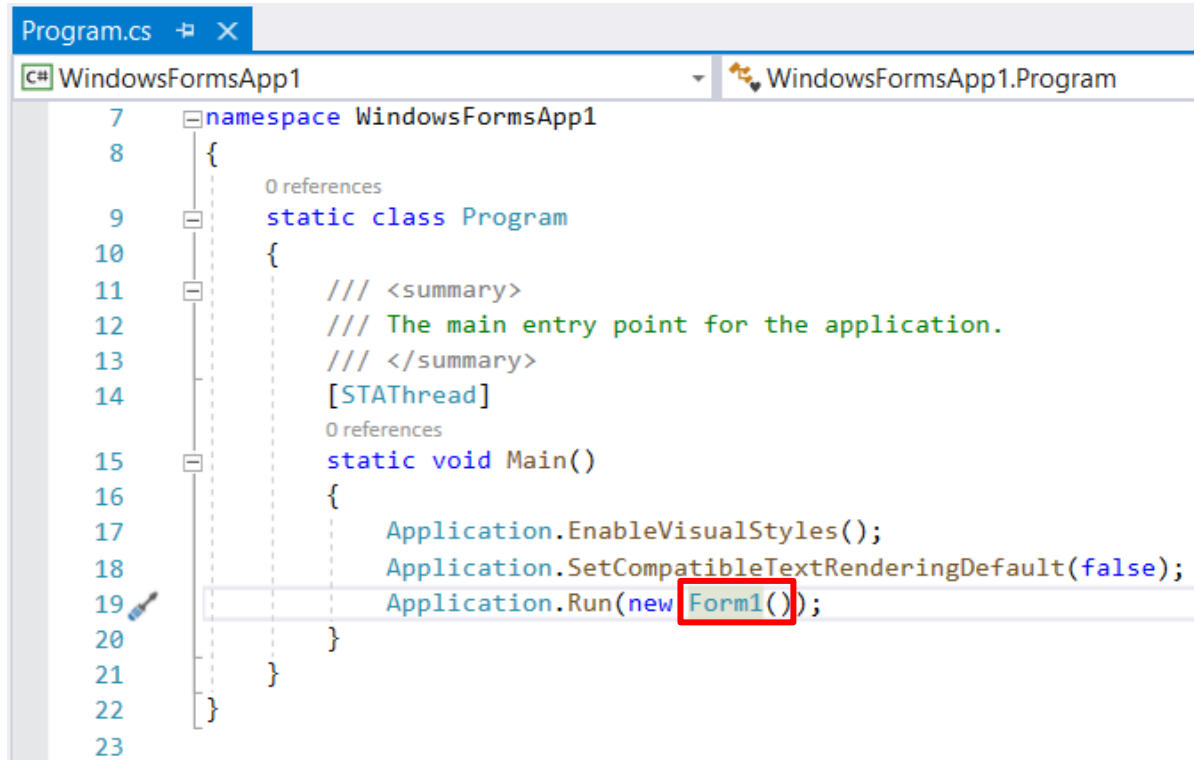
Apabila dalam suatu project terdapat beberapa form, kita dapat mengatur form mana yang muncul ketika aplikasi dijalankan.

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



Gambar 1.24: Program.cs pada Souldution Explorer

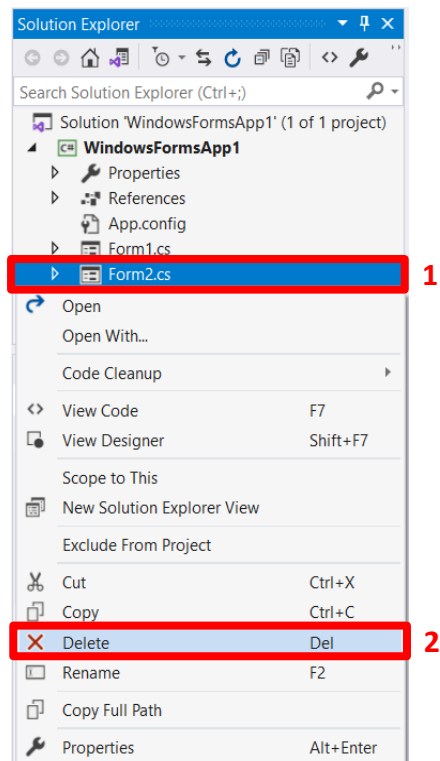
- Ubah nama form pada kotak berwarna merah menjadi nama form yang ingin dijalankan.



Gambar 1.25: Tampilan Program.cs

## MENGHAPUS ITEM

Untuk menghapus item, user hanya perlu melakukan klik kanan pada nama item pada solution explorer yang akan dihapus kemudian klik **delete**.



Gambar 1.26: Menghapus Item



## LESSON 4: OVERVIEW

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.

## TOPIC 1: 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.

Tabel 1.3: Penamaan Kontrol

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

## Catatan:

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`

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.

## TOPIC 2: PROPERTY UMUM CLASS CONTROL

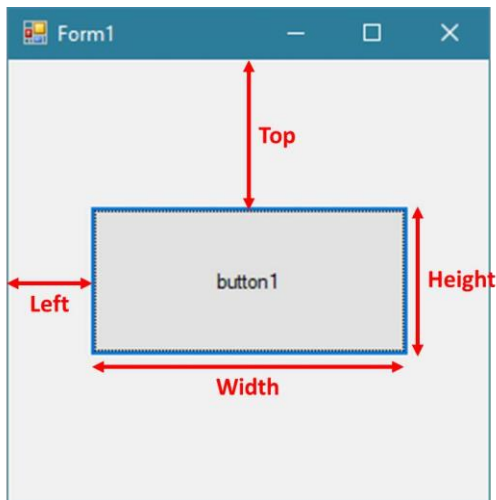
Property yang umumnya terdapat pada kontrol adalah property yang berhubungan dengan posisi (**position**) dan ukuran (**size**).

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



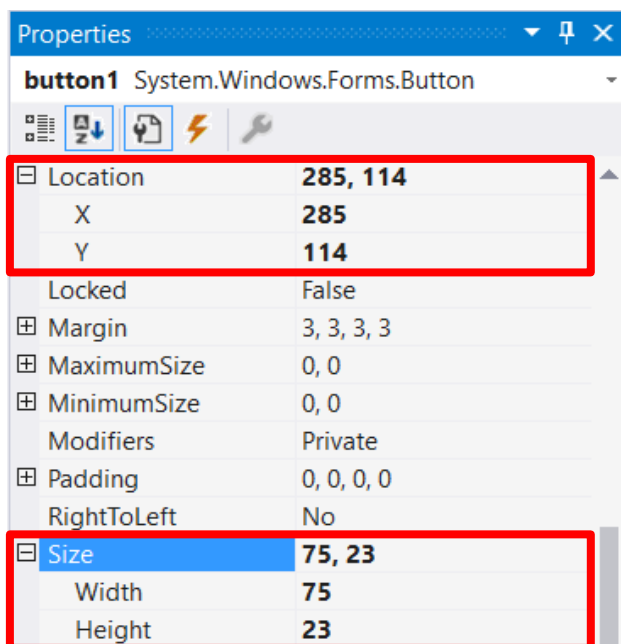




Gambar 1.27: Ilustrasi Property Position dan Size

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.



Gambar 1.28: Property Location dan Size di jendela Properties

### TOPIC 3: 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;
```

#### TOPIC 4: MENGATUR NAME PADA EVENT

Untuk menambahkan event baru pada kontrol, dapat melalui 2 cara.

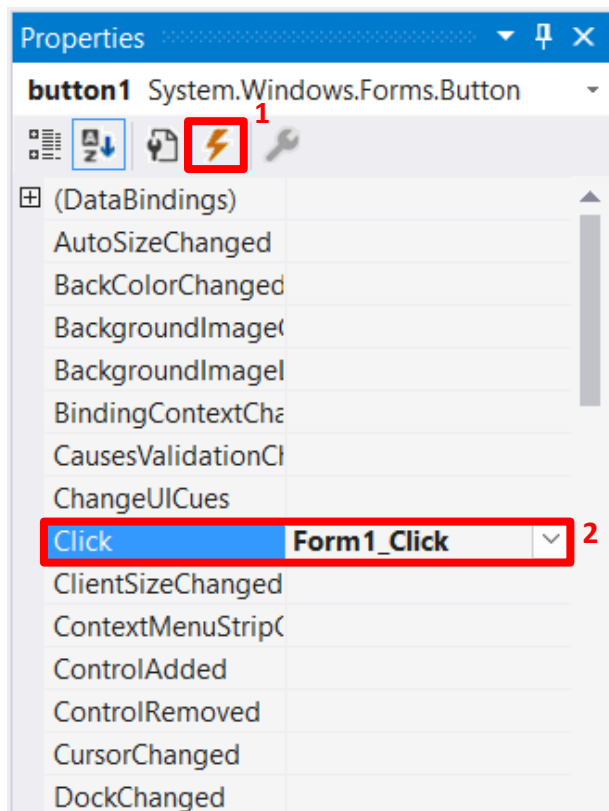
**Cara pertama** yaitu dengan double klik pada kontrol untuk memunculkan event default dari kontrol tersebut. Kode yang muncul ketika kita melakukan double klik pada kontrol akan mengikuti format sebagai berikut:

```
private void ControlName_Event(Arguments)
{
}
}
```

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

**Cara kedua** adalah melalui **Properties Window**. Cara ini terutama digunakan apabila kita ingin menggunakan event yang lain dari kontrol (bukan event default).

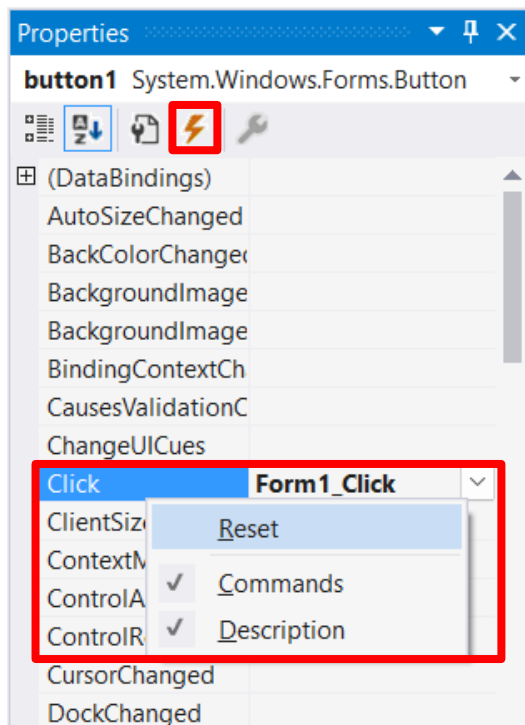
Pilih tombol **event** pada Properties Window dan double klik pada event yang diinginkan. Kode program akan langsung muncul pada Code Window.



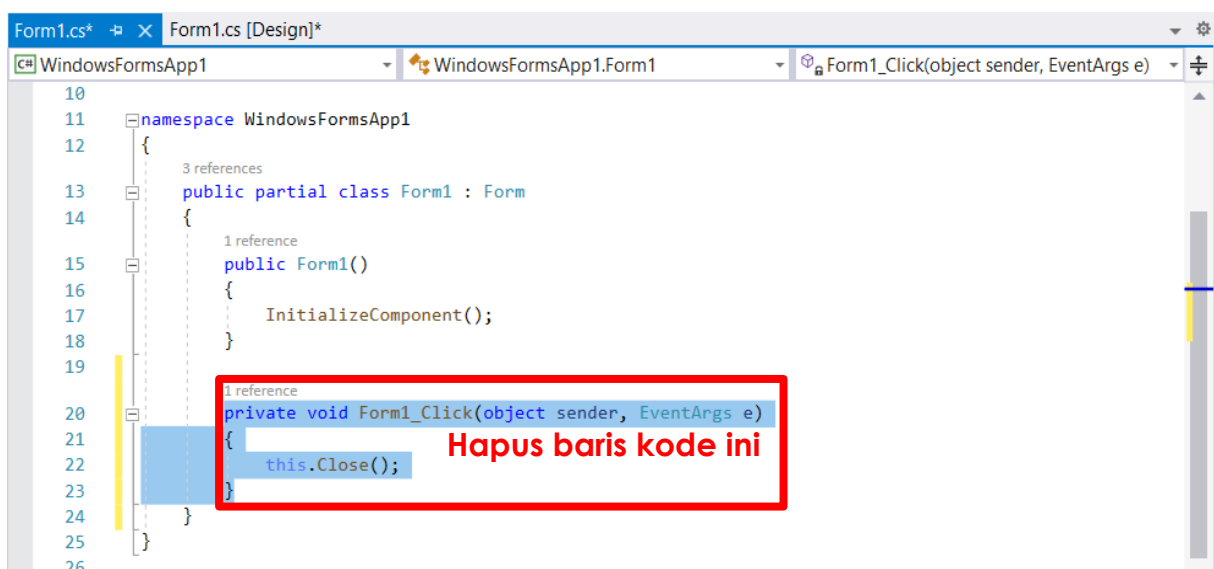
Gambar 1.29: Membuat Event Melalui Properties Window

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.

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 koding yang menghubungkan metode dengan event.



Gambar 1.30: Reset pada Event di Properties Window

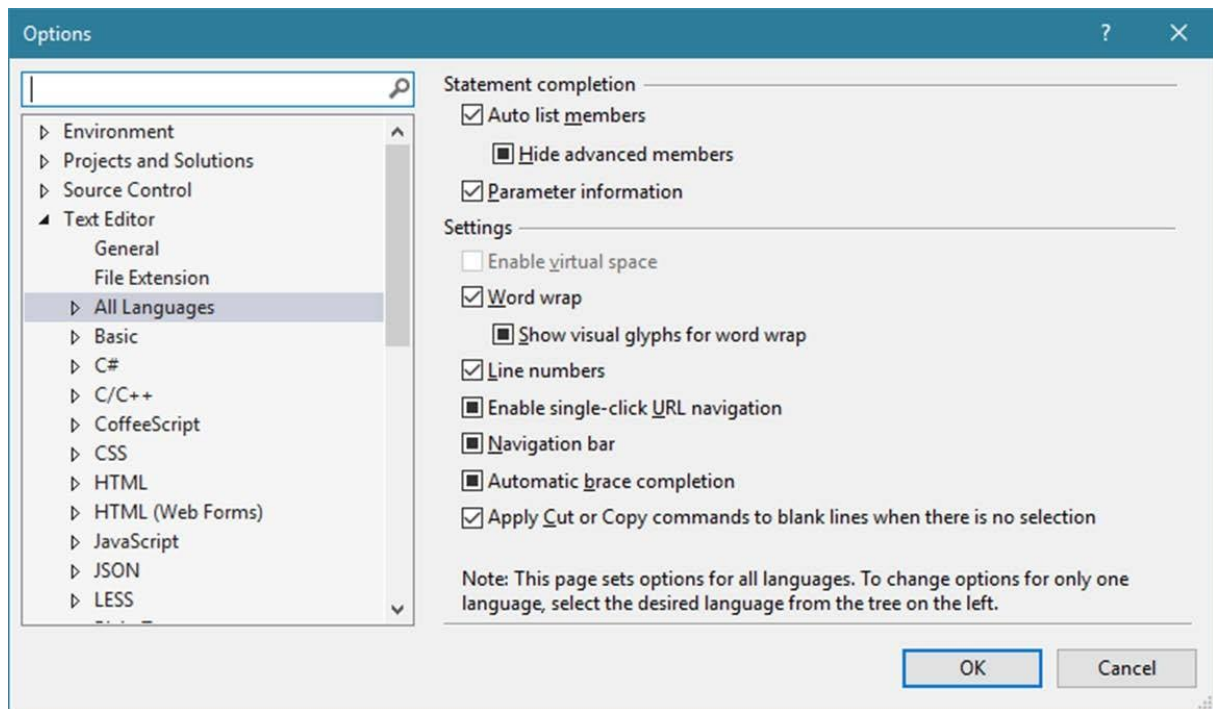


Gambar 1.31: Menghapus Event pada Jendela Code Editor

## TOPIC 5: PENULISAN KODE PROGRAM

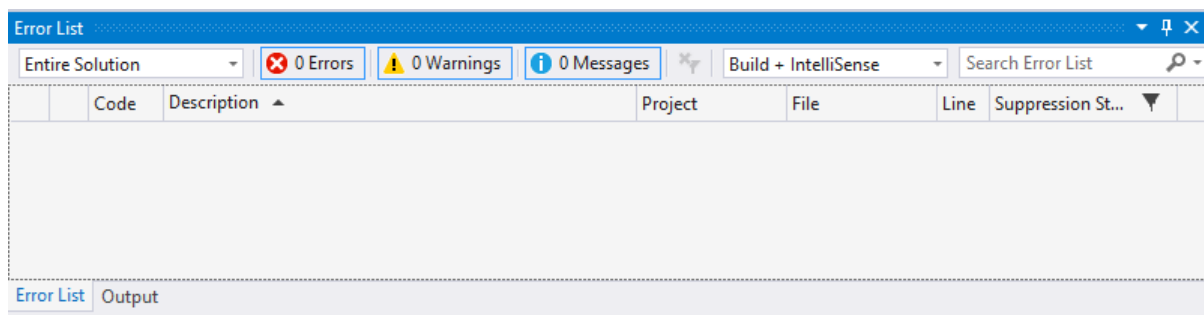
Langkah terakhir dalam membangun aplikasi berbasis Visual C# adalah menuliskan kode program dengan menggunakan Bahasa C#. Fitur yang berguna dalam penulisan kode program adalah dengan menampilkan nomor baris (line number) di samping kode program yang ditulis.

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.



Gambar 1.32: Kotak Dialog Options

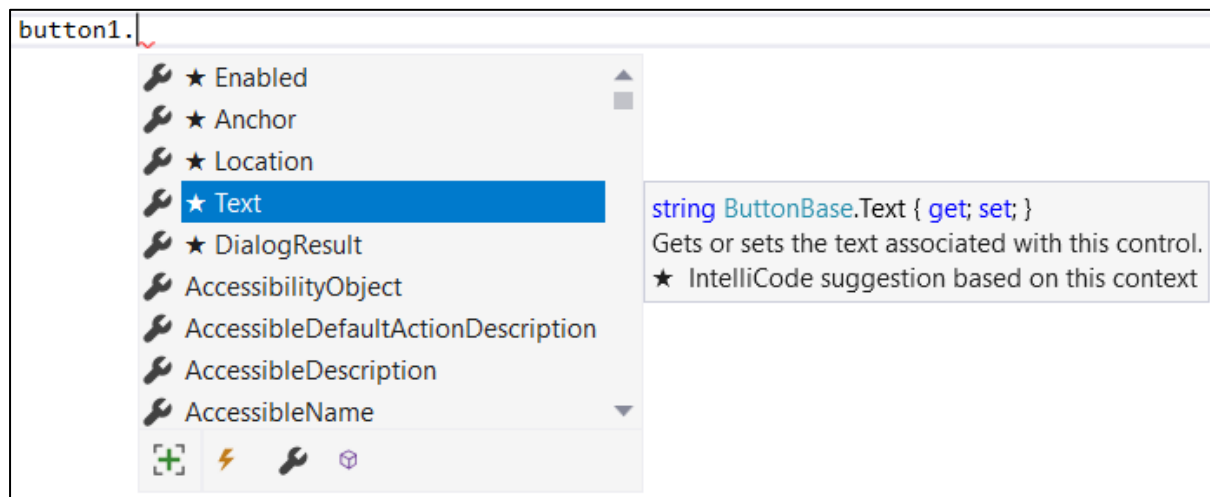
Line numbers tersebut akan berguna untuk mengidentifikasi di baris nomor berapa suatu error terjadi pada kode program. Informasi tersebut dapat dilihat pada **Error List Window**. Untuk mengaktifkannya, pilih menu **View** → **Error List**.



Gambar 1.33: Error List

## TOPIC 6: FITUR INTELLISENSE

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.



Gambar 1.34: Fitur Intellisense

## LESSON 5: MENGENAL FORM

### LESSON 5: OVERVIEW

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.

## TOPIC 1: 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.

Tabel 1.4: 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/Restore, 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
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

**Tabel 1.5: Metode Umum Form**

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

Sintaks penggunaan metode khusus form:

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



**Tabel 1.6: 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 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**

## LESSON 6: BUTTON, LABEL, DAN TEXTBOX

### LESSON 6: OVERVIEW

Kontrol yang paling umum untuk digunakan dalam sebuah program adalah Button, Label dan Textbox.

### TOPIC 1: BUTTON

Biasanya digunakan untuk mengeksekusi suatu aksi dengan cara mengklik.

**Tabel 1.7: Property Umum pada Button**

Property	Keterangan
Name	Mengatur nama button (Button diawali dengan <b>btn</b> )
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

**Tabel 1.8: Metode Umum pada Button**

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



**Tabel 1.9: Event Umum pada Button**

Event	Keterangan
Click (default)	Terjadi ketika button ditekan

## TOPIC 2: LABEL

Kontrol Label digunakan untuk menampilkan teks atau gambar yang tidak dapat diedit oleh user.

**Tabel 1.10: 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
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

**Tabel 1.11: Event Umum pada Label**

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

## TOPIC 3: 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.

**Tabel 1.12: 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





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

**Tabel 1.13: 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
SelectAll	Memilih semua teks yang ada di textbox
Undo	Membatalkan operasi edit terakhir

**Tabel 1.14: 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

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.

## LESSON 7: GROUPBOX, PANEL, CHECKBOX, DAN RADIOBUTTON

### LESSON 7: OVERVIEW

Dalam memberikan pilihan inputan pada program, dapat menggunakan kontrol RadioButton maupun CheckBox. Kedua kontrol tersebut disebut sebagai kontrol 'point and click' yang memudahkan user untuk membuat pilihan dengan meng-klik menggunakan mouse sehingga mengurangi tingkat error pada aplikasi. Pilihan tersebut dapat dikelompokkan berdasarkan kriteria tertentu pada kontrol GroupBox maupun Panel.

### TOPIC 1: GROUPBOX

Digunakan untuk mengelompokkan sekumpulan kontrol yang saling berhubungan dalam aplikasi VisualC#. Kontrol-kontrol yang saling berhubungan harus dimasukkan ke dalam groupbox sehingga groupbox dan kontrol dapat digerakkan secara bersamaan.

Sama seperti object form, kontrol groupbox merupakan kontrol container yang menampung kontrol-kontrol lain, sehingga property BackColor, ForeColor, dan Font akan diwariskan kepada kontrol. Hal yang sama berlaku untuk property Enabled dan Visible. Groupbox mempengaruhi cara kerja pilihan radiobutton.

Radiobutton dalam kontrol groupbox bekerja sebagai suatu group yang terpisah (independen) dengan radiobutton di kontrol groupbox lain. Demikian pula dengan radiobutton dalam form juga dianggap sebagai satu group.

**Tabel 1.15: Property Umum pada GroupBox**

Property	Keterangan
Name	Mengatur nama groupbox (GroupBox diawali dengan <b>grp</b> )
BackColor	Mengatur warna latar groupbox
Enabled	Jika diatur menjadi False, semua kontrol dalam groupbox kelihatan tetapi dalam kondisi disabled
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Text	Mengatur tulisan yang tampil pada judul groupbox
Visible	Jika bernilai False, groupbox (dan semua kontrol di dalamnya) menjadi tidak kelihatan



Kontrol GroupBox memiliki beberapa method dan event, akan tetapi method dan event pada GroupBox jarang digunakan. Programmer lebih berfokus pada method dan event yang berhubungan dengan kontrol yang terdapat di dalam GroupBox.

---

## TOPIC 2: PANEL

Fungsi dan perilakunya mirip dengan groupbox, yaitu digunakan untuk mengelompokkan kontrol. Perbedaan antara Panel dan GroupBox:

- Panel memiliki BorderStyle, sedangkan GroupBox tidak.
- GroupBox memiliki property Text untuk memberi judul (penjelasan isi atau fungsi GroupBox) dan tidak mengandung ScrollBar, sedangkan Panel dapat mengandung ScrollBar dan tidak dapat menampilkan Judul.

**Tabel 1.16: Property Umum pada Panel**

Property	Keterangan
Name	Mengatur nama panel (Panel diawali dengan pnl)
AutoScroll	Mengatur apakah panel memunculkan scrollbar secara otomatis atau tidak
BackColor	Mengatur warna latar panel
BorderStyle	Mengatur tipe bingkai panel
Enabled	Jika diatur menjadi False, semua kontrol dalam panel kelihatan tetapi dalam kondisi disabled
Visible	Jika bernilai False, panel (dan semua kontrol di dalamnya) menjadi tidak kelihatan

Kontrol Panel memiliki beberapa method dan event, akan tetapi method dan event pada Panel jarang digunakan. Programmer lebih berfokus pada method dan event yang berhubungan dengan kontrol yang terdapat di dalam Panel.

---

## TOPIC 3: CHECKBOX

Digunakan untuk membuat serangkaian pilihan dimana user dapat memilih 1 atau lebih dari pilihan-pilihan yang ada.

**Tabel 1.17: Property Umum pada CheckBox**

Property	Keterangan
Name	Mengatur nama checkbox (CheckBox diawali dengan chk)
AutoSize	Mengatur apakah ukuran kontrol berubah secara otomatis sesuai dengan isi kontrol atau tidak
BackColor	Mengatur warna latar checkbox
Checked	Mengatur apakah checkbox dalam kondisi terpilih (centang) atau tidak
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Text	Mengatur tulisan yang ditampilkan di samping checkbox
TextAlign	Mengatur perataan tulisan pada checkbox



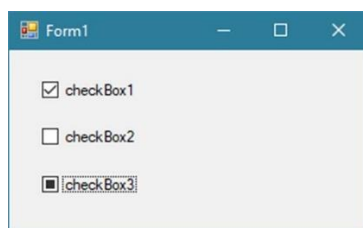
**Tabel 1.18: Metode Umum pada CheckBox**

Method	Keterangan
Focus	Mengatur focus pada checkbox

**Tabel 1.19: Event Umum pada CheckBox**

Event	Keterangan
CheckedChanged (default)	Terjadi ketika property Checked pada checkbox berubah (memunculkan / menghilangkan centang pada checkbox)

Secara default, kontrol CheckBox hanya menerima 2 state (kondisi), yaitu checked dan unchecked. CheckBox dapat dikonfigurasi untuk menerima 3 state (kondisi), yaitu **checked**, **unchecked**, dan **indeterminate**. Caranya adalah dengan mengatur property **ThreeState** menjadi **True**.



**Gambar 1.35: Property ThreeState pada CheckBox**

#### TOPIC 4: RADIOBUTTON

RadioButton digunakan untuk membuat serangkaian pilihan dimana user hanya dapat memilih 1 pilihan dari berbagai pilihan yang ada. RadioButton yang diletakkan dalam form akan bekerja sebagai suatu group, dimana jika salah satu radiobutton terpilih, maka otomatis radiobutton lainnya dalam kondisi tidak terpilih. Untuk membuat pilihan radiobutton yang berbeda, dibutuhkan bantuan kontrol GroupBox ataupun Panel.

**Tabel 1.20: Property Umum pada RadioButton**

Property	Keterangan
Name	Mengatur nama RadioButton (RadioButton diawali dengan <b>rdo</b> )
AutoSize	Mengatur apakah ukuran kontrol berubah secara otomatis sesuai dengan isi kontrol atau tidak
BackColor	Mengatur warna latar RadioButton
Checked	Mengatur apakah RadioButton dalam kondisi terpilih atau tidak
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Text	Mengatur tulisan yang ditampilkan di samping RadioButton
TextAlign	Mengatur perataan tulisan pada RadioButton



**Tabel 1.21: Metode Umum pada RadioButton**

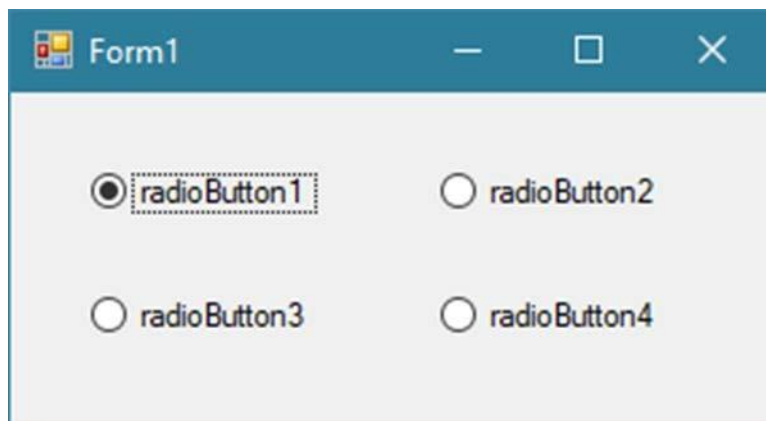
Method	Keterangan
Focus	Mengatur focus pada radiobutton

**Tabel 1.22: Event Umum pada RadioButton**

Event	Keterangan
CheckedChanged (default)	Terjadi ketika property Checked pada radiobutton berubah

## TOPIC 5: MENANGANI MULTIPLE EVENTS

Sebelumnya, setiap method akan menangani 1 event. Tetapi sekarang, dengan adanya kontrol seperti CheckBox dan RadioButton, akan lebih mudah apabila 1 buah method dapat menangani multiple events. Sebagai contoh, jika kita memiliki 4 RadioButton dalam sebuah group, ketika salah satu RadioButton dipilih, akan lebih mudah apabila terdapat 1 buah method untuk menentukan RadioButton mana yang dipilih, daripada harus membuat 4 buah method yang terpisah.

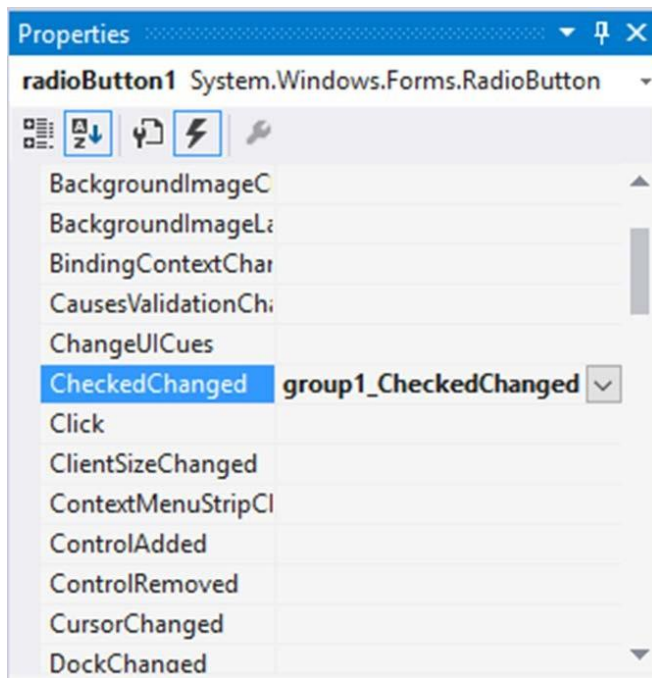


**Gambar 1.36: RadioButton**

Asumsikan kita memiliki 4 RadioButton yang saling berbagi event method **CheckedChanged**. Untuk melakukannya, pertama kita memilih radioButton1, kemudian buka Properties Window dan klik button Event. Cari event CheckedChanged.

Biasanya pada tahap ini, kita akan double klik pada event dan Visual C# akan membuat event method dengan nama radioButton1\_CheckedChanged.

Nama event method bersifat bebas. Kita dapat memberikan nama yang kita inginkan. Untuk menggunakan nama yang berbeda, kita mengetikkan nama pada isian di samping event CheckedChanged pada Properties Window. Nama yang digunakan sebaiknya dapat merepresentasikan 1 group RadioButton, bukan hanya 1 RadioButton.



**Gambar 1.37: Menangani Multiple Events**

Setelah mengetikkan nama untuk event pada Properties Window dan menekan tombol Enter, maka pada code window akan memunculkan:

```
private void group1_CheckedChanged(object sender, EventArgs e)
{
}

```

Method di atas hanya berlaku untuk radioButton1. Supaya kontrol lain dapat berbagi method ini, kita gunakan lagi Properties Window. Selanjutnya, pilih RadioButton yang lain. Buka Properties Window dan klik button Events, cari event CheckedChanged, kemudian pilih group1\_CheckedChanged. Lakukan langkah yang sama untuk semua RadioButton.

## LESSON 8: LISTBOX DAN COMBOBOX

### LESSON 8: OVERVIEW

Perbedaan ComboBox dan ListBox terdapat pada pilihan item. Pada ComboBox hanya satu item pilihan yang terlihat sebelum ComboBox diklik, sedangkan pada ListBox dapat menampilkan beberapa pilihan.

### TOPIC 1: LISTBOX

ListBox digunakan untuk menampilkan daftar dari sejumlah item atau pilihan menu dimana user dapat memilih 1 atau lebih item. Jika ListBox memiliki jumlah item yang melebihi ukurannya, secara otomatis akan menampilkan ScrollBar.



**Tabel 1.23: Property Umum pada ListBox**

Property	Keterangan
Name	Mengatur nama listbox (ListBox diawali dengan lst)
BackColor	Mengatur warna latar listbox
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Items	Mengatur item atau pilihan yang tampil pada listbox
SelectedIndex	Mengambil nilai indeks yang merupakan posisi item terpilih dalam listbox. Nilai indeks dimulai dari 0.
SelectedIndices	Mengambil nilai indeks yang merupakan posisi semua item terpilih dalam listbox. Item yang terpilih bisa lebih dari 1.
SelectedItem	Mengambil teks item terpilih dalam listbox
SelectedItems	Mengambil teks semua item terpilih dalam listbox. Item yang terpilih bisa lebih dari 1.
SelectionMode	Mengatur apakah user hanya dapat memilih 1 item atau boleh memilih lebih dari 1 item pada listbox
Sorted	Mengatur apakah item pada listbox diurutkan secara alfabetis
Text	Mengatur tulisan dari item terpilih pada listbox
TopIndex	Mengatur indeks dari item pertama yang terlihat di listbox

**Tabel 1.24: Metode Umum pada ListBox**

Method	Keterangan
ClearSelected	Menghilangkan seleksi semua item dalam ListBox
FindString	Menemukan indeks dari item pertama dalam listbox yang dimulai dengan string yang dicari
GetSelected	Mengembalikan nilai apakah item tertentu dalam kondisi terseleksi atau tidak
SetSelected	Menyeleksi atau menghilangkan seleksi item dalam listbox

**Tabel 1.25: Event Umum pada ListBox**

Event	Keterangan
SelectedIndexChanged (default)	Terjadi ketika property SelectedIndex berubah

Items pada ListBox memiliki property dan method sendiri untuk menambah atau menghapus item. Items pada ListBox dimulai dari indeks 0. Sehingga, item pertama pada ListBox dapat ditulis:

```
lstNama.Items[0]
```

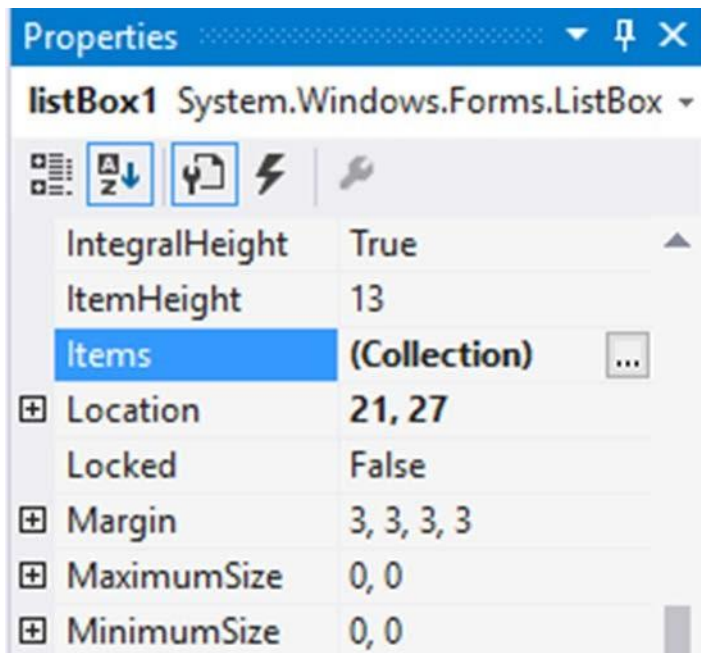


Item terakhir pada ListBox dapat ditulis:

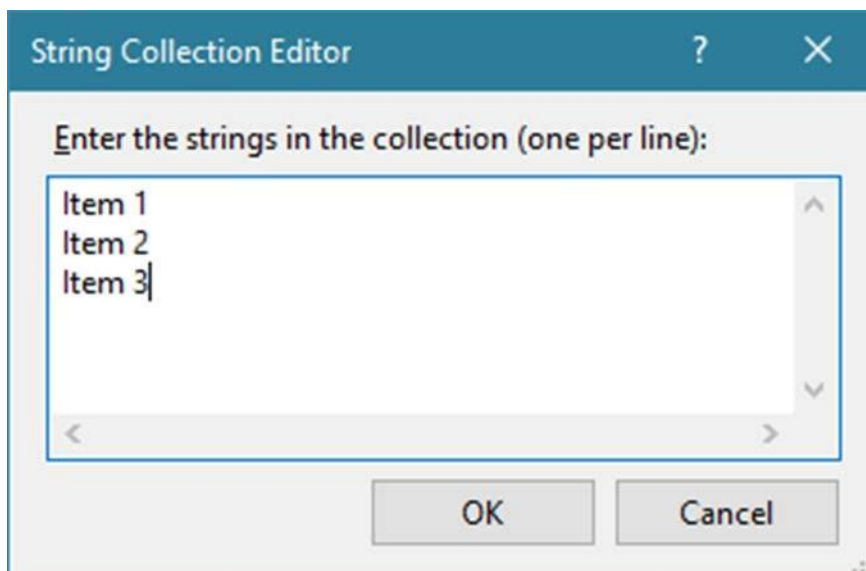
```
lstNama.Items[lstNama.Items.Count-1]
```

**Count** (property **Items**) menentukan jumlah items pada ListBox. Pengurangan 1 dibutuhkan karena indeks dimulai dari 0.

Pada design mode, pilih property **Items**, kemudian klik tombol browse yang muncul di samping tulisan **(Collections)**. Kita dapat menambahkan item satu per satu, dengan catatan 1 item menempati 1 baris. Klik **OK** setelah selesai.



Gambar 1.38: Property Items pada ListBox



Gambar 1.39: Menambahkan Item ListBox



Pada tampilan code editor:

1. Untuk menambah item, gunakan method **Add**.

```
lstNama.Items.Add(ItemToAdd);
```

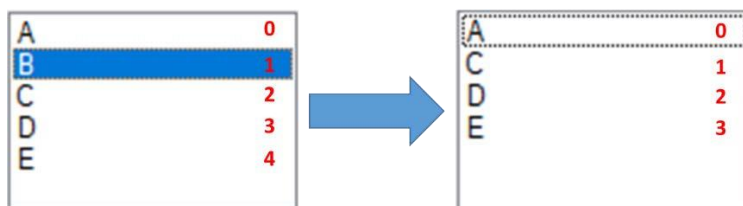
2. Untuk menghapus item, gunakan method **Remove** ataupun **RemoveAt**.

```
lstNama.Items.Remove(ItemToRemove);  
lstNama.Items.RemoveAt(IndexofItemToRemove);
```

3. Untuk menghapus semua item pada ListBox, gunakan method **Clear**.

```
lstNama.Items.Clear();
```

ListBox umumnya berisi tipe data string, walaupun tipe data yang lain juga memungkinkan. Sebagai catatan, ketika kita menghapus suatu item, maka index pada item-item berikutnya akan berubah karena proses penghapusan item tersebut. Contoh: apabila pada ListBox terdapat 5 item, dan kita menghapus item pada index 1, maka item-item berikutnya akan menempati index 1, 2, dan 3.



Gambar 1.40: Penghapusan Item di ListBox

Property **SelectionMode** menentukan apakah user hanya dapat memilih 1 item saja atau dapat memilih lebih dari 1 item.

- **SelectionMode.One**: user hanya dapat memilih 1 item saja.
- **SelectionMode.MultiExtended**: user dapat memilih lebih dari 1 item dengan cara menekan tombol Shift + mouse click atau tombol Shift + tombol panah untuk menambah item. User juga dapat menekan tombol Ctrl + mouse click untuk select atau deselect item.
- **SelectionMode.MultiSimple**: user dapat memilih lebih dari 1 item dengan cara mouse click atau menekan tombol space pada keyboard untuk select atau deselect item.

**SelectedItems** juga memiliki property tersendiri untuk menentukan item-item yang terpilih pada ListBox. Salah satu property adalah **Count** yang digunakan untuk mengetahui jumlah item yang terpilih. Nilai ini bersama dengan array **SelectedIndices** dapat mengidentifikasi serangkaian item yang terpilih.

Contoh:

```
for (int i = 0; i <= lstNama.SelectedItems.Count - 1; i++)  
{  
    lblTampil.Text += lstNama.SelectedItems[i] + "\n";  
}
```

## TOPIC 2: COMBOBOX

Apabila kontrol ListBox mirip dengan sekumpulan CheckBox, maka kontrol ComboBox mirip dengan sekumpulan RadioButton. ComboBox memungkinkan user memilih 1 item dari daftar, dan dalam beberapa kasus, memungkinkan user untuk mengetikkan pilihan alternatif.

Property pada ComboBox mirip dengan ListBox, hanya saja tidak ada property **SelectionMode** dan terdapat property **DropDownStyle**.

**Tabel 1.26: Property Umum pada ComboBox**

Property	Keterangan
Name	Mengatur nama combobox (ComboBox diawali dengan <b>cbo</b> )
BackColor	Mengatur warna latar combobox
DropDownStyle	Mengatur style dropdown dari combobox
Font	Mengatur jenis, style, dan ukuran tulisan
ForeColor	Mengatur warna tulisan
Items	Mengatur item atau pilihan yang tampil pada combobox
MaxDropDownItems	Mengatur jumlah item yang tampil pada bagian dropdown combobox. Property <b>IntegralHeight</b> harus diatur jadi <b>False</b> .
SelectedIndex	Mengatur nilai indeks yang merupakan posisi item terpilih dalam combobox. Nilai indeks dimulai dari 0.
SelectedItem	Mengatur item terpilih dalam combobox
SelectedText	Mengatur teks yang terpilih pada bagian combobox yang dapat diedit
Sorted	Mengatur apakah items diurutkan secara alfabetis
Text	Mengatur nilai string yang ditampilkan di combobox

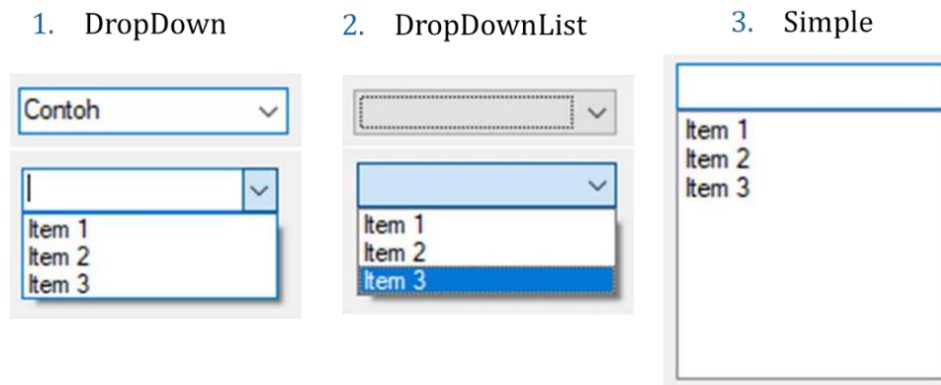
**Tabel 1.27: Event Umum pada ComboBox**

Event	Keterangan
KeyPress	Terjadi ketika tombol keyboard ditekan pada saat focus berada di kontrol combobox
SelectedIndexChanged (default)	Terjadi ketika property <b>SelectedIndex</b> berubah

**Items** pada kontrol ComboBox mirip dengan items pada kontrol ListBox sehingga cara untuk menambah dan menghapus item pada ComboBox juga sama seperti pada ListBox.



Property DropDownStyle pada ComboBox memiliki 3 pilihan nilai, yaitu:



Gambar 1.41: Property DropDownStyle ComboBox

## LESSON 9: MESSAGEBOX

### LESSON 9: OVERVIEW

Dalam aplikasi berbasis windows form, terdapat banyak kontrol yang disediakan oleh Visual C#, seperti MessageBox dan DialogBox. MessageBox digunakan untuk menyajikan pesan kepada pengguna. DialogBox digunakan untuk berinteraksi dengan pengguna dan mengambil informasi.

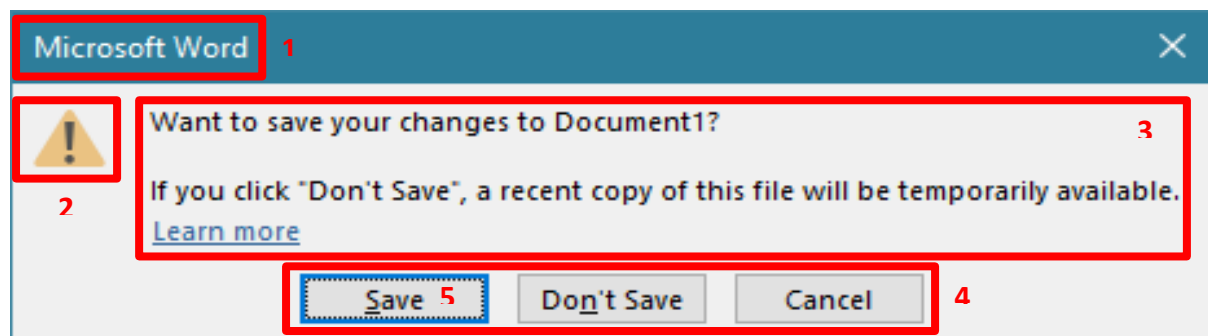
### TOPIC 1: MESSAGEBOX

Salah satu method yang paling banyak digunakan dalam Visual C# adalah objek MessageBox. Objek ini mengizinkan programmer untuk menampilkan pesan kepada user dan menerima feedback (respon) untuk informasi lebih lanjut. MessageBox dapat digunakan untuk menampilkan pesan error, menjelaskan masalah potensial, atau hanya untuk menampilkan hasil dari beberapa perhitungan.

Objek MessageBox sangat berguna dengan kemampuan untuk menampilkan pesan, icon, maupun berbagai pilihan jenis tombol. User akan merespon dengan menekan salah satu tombol pada MessageBox.

Untuk menggunakan MessageBox, programmer harus menentukan:

1. Caption yang berisi judul MessageBox
2. Icon yang sesuai
3. Text yang berisi pesan yang ingin disampaikan.
4. Button yang berisi pilihan tombol yang tampil pada MessageBox
5. Default Button untuk menentukan tombol mana yang dijadikan default



Gambar 1.42: Bagian MessageBox

Untuk menampilkan MessageBox, dilakukan melalui koding dengan menggunakan method **Show**. Ada beberapa cara untuk menggunakan method Show pada MessageBox:

```
MessageBox.Show("text");  
MessageBox.Show("text", "caption");  
MessageBox.Show("text", "caption", Buttons);  
MessageBox.Show("text", "caption", Buttons, Icon);  
  
MessageBox.Show("text", "caption", Buttons, Icon, DefaultButton);
```

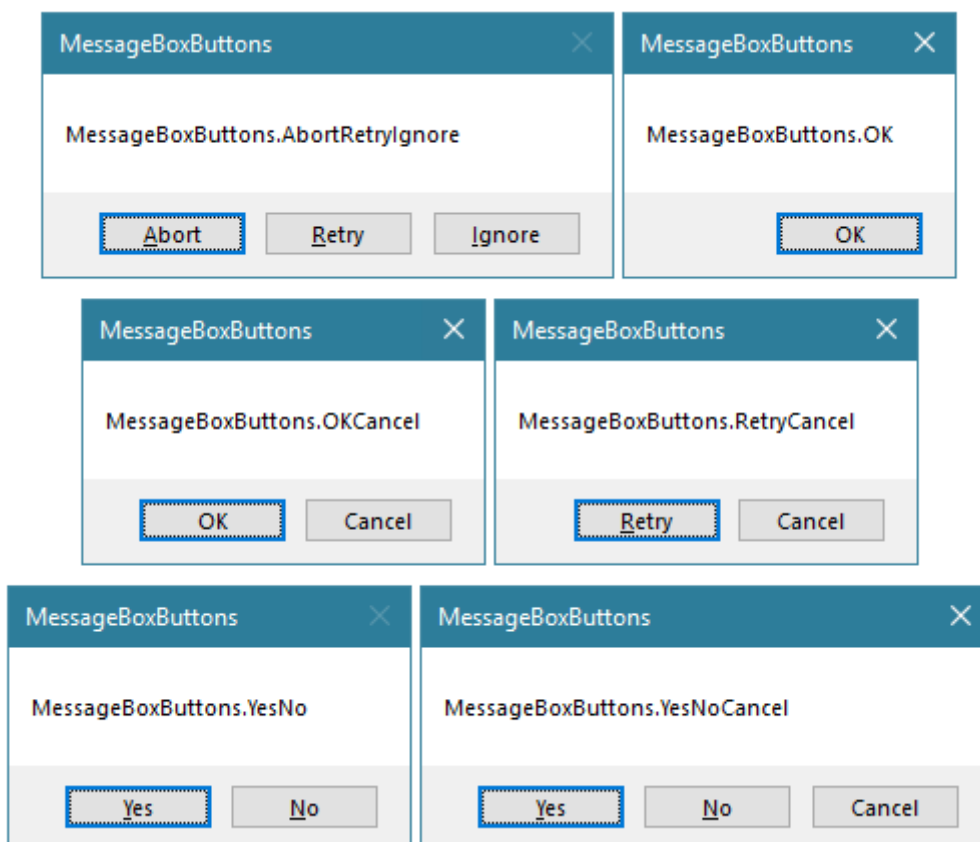
Dalam implementasi kontrol MessageBox, terdapat beberapa kondisi seperti:

- Jika parameter Default Button tidak diberikan, maka button pertama akan dijadikan sebagai default.
- Jika parameter Icon tidak diisi, tidak akan ada icon yang ditampilkan.
- Jika parameter Button tidak diisi, maka akan ditampilkan sebuah tombol OK.
- Jika parameter Caption tidak diisi, maka tidak akan ada judul yang tampil.

Parameter Text dan Caption dapat diisi sendiri dengan tipe data String, tetapi parameter yang lain harus diisi sesuai dengan konstanta yang sudah tersedia dalam Visual C#.

Nilai yang dapat diisi dalam parameter MessageBoxButtons:

1. `MessageBoxButtons.AbortRetryIgnore`
2. `MessageBoxButtons.OK`
3. `MessageBoxButtons.OKCancel`
4. `MessageBoxButtons.RetryCancel`
5. `MessageBoxButtons.YesNo`
6. `MessageBoxButtons.YesNoCancel`



Gambar 1.43: Tampilan MessageBoxButtons

Nilai yang dapat diisi dalam parameter MessageBoxIcon:

1. MessageBoxIcon.Asterisk
2. MessageBoxIcon.Information
3. MessageBoxIcon.Error
4. MessageBoxIcon.Hand
5. MessageBoxIcon.None
6. MessageBoxIcon.Stop
7. MessageBoxIcon.Exclamation
8. MessageBoxIcon.Warning
9. MessageBoxIcon.Question



Gambar 1.44: Tampilan MessageBoxIcon

Walaupun terdapat 8 nilai yang dapat diisi pada MessageBoxIcon, hanya ada 4 jenis icon yang akan tampil yaitu Information, Error, Exclamation, dan Question. Hal ini dikarenakan sistem operasi Windows saat ini hanya menyediakan 4 icon.

Nilai yang dapat diisi dalam parameter MessageBoxDefaultButton:

1. MessageBoxDefaultButton.Button1
2. MessageBoxDefaultButton.Button2
3. MessageBoxDefaultButton.Button3

Default Button yang ditentukan bersifat relatif pada Button yang ditampilkan dari kiri ke kanan. Misalnya jika kita memiliki MessageBox dengan button Yes, No, dan Cancel, maka Button1 adalah Yes, Button2 adalah No, dan Button3 adalah Cancel.

Ketika user memberikan respon terhadap MessageBox dengan menekan salah satu Button, maka untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult. Nilai DialogResult yang tersedia sesuai dengan tombol yang ditampilkan pada MessageBox:

1. DialogResult.Abort
2. DialogResult.Cancel
3. DialogResult.Ignore
4. DialogResult.No
5. DialogResult.OK
6. DialogResult.Retry
7. DialogResult.Yes

```
if (MessageBox.Show("Ini adalah contoh MessageBox", "Contoh MessageBox",  
    MessageBoxButtons.OKCancel, MessageBoxIcon.Information,  
    MessageBoxDefaultButton.Button1) == DialogResult.OK)  
  
{  
  
    //baris kode jika menekan tombol OK  
  
}
```

## LESSON 10: OVERVIEW

Kotak Dialog digunakan untuk berinteraksi dengan pengguna dan mengambil informasi. Kotak Dialog yang dibahas meliputi OpenFileDialog, SaveFileDialog, ColorDialog, FontDialog, Print Dialog, PrintPreviewDialog, dan PageSetupDialog. OpenFileDialog digunakan untuk memilih file yang akan dibuka. SaveFileDialog digunakan untuk memilih lokasi penyimpanan file. ColorDialog digunakan untuk memilih warna yang ingin dipakai. FontDialog digunakan untuk memilih font tulisan yang dipakai. PrintDialog digunakan untuk memilih printer, halaman yang diprint dan settingan terkait print lainnya. PrintPreviewDialog digunakan untuk menampilkan tampilan halaman yang akan diprint. PageSetupDialog digunakan untuk menampilkan pengaturan halaman dokumen yang mau diprint.

## TOPIC 1: SAVEFILEDIALOG

Dengan menggunakan SaveFileDialog, kita dapat menyimpan file dengan memilih lokasi penyimpanan file yang diinginkan. Apabila user menggunakan nama file yang sudah ada pada lokasi yang dipilih, maka file tersebut akan ditimpa dengan file baru.

Sebagai catatan, kontrol SaveFileDialog tidak memiliki interface (antarmuka) yang bisa dilihat pada saat mendesain form. Setelah user memasukkan kontrol SaveFileDialog ke dalam form, kontrol SaveFileDialog akan tampil di tray area.

Berikut adalah property umum pada SaveFileDialog:

**Tabel 1.28: Property SaveFileDialog**

Property	Keterangan
Name	Mengatur nama savefiledialog (SaveFileDialog umumnya dinamai dengan <b>dlgSave</b> )
AddExtension	Menentukan apakah savefiledialog akan menambahkan extension secara otomatis pada nama file jika user tidak memasukkan extension
CheckFileExists	Menentukan apakah savefiledialog akan menampilkan peringatan jika nama file yang ingin disimpan sudah ada pada lokasi penyimpanan
CheckPathExists	Menentukan apakah savefiledialog akan menampilkan peringatan jika user memasukkan lokasi yang tidak tersedia
CreatePrompt	Menentukan apakah kotak dialog akan meminta persetujuan user untuk membuat file baru apabila user memasukkan file yang belum ada
DefaultExt	Menentukan extension default pada file
FileName	Menentukan string yang berisi nama file yang dipilih di dalam kotak dialog
Filter	Menentukan pilihan tipe file yang dapat disimpan ke dalam kotak dialog
FilterIndex	Menentukan indeks pilihan filter yang terpilih pada kotak dialog
InitialDirectory	Menentukan folder yang akan ditampilkan pertama kali ketika kotak dialog muncul
OverwritePrompt	Mengindikasikan apakah kotak dialog akan memberikan peringatan jika user memasukkan nama file yang sudah ada
Title	Menentukan judul dari kotak dialog

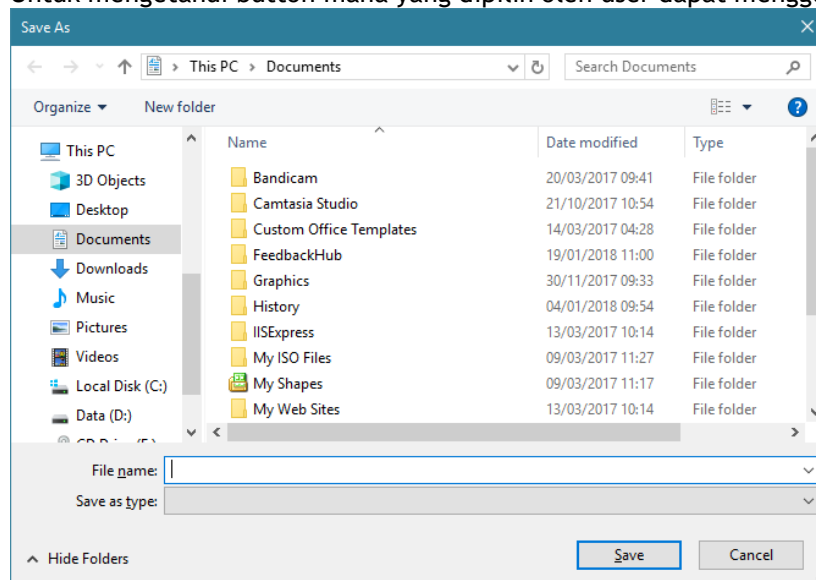


Berikut adalah method umum pada SaveFileDialog:

**Tabel 1.29: Metode SaveFileDialog**

Method	Keterangan
ShowDialog	Menampilkan kotak dialog, mengembalikan nilai yang mengindikasikan button mana yang dipilih oleh user: OK atau Cancel.

Untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult.



**Gambar 1.45: Tampilan Kontrol SaveFileDialog**

## TOPIC 2: OPENFILEDIALOG

Pada sebagian aplikasi, programmer membutuhkan file name dari user. Namun, jika meminta user untuk mengetikkan lokasi dan nama file, sangat rawan terjadi kesalahan sehingga dibutuhkan interface yang bersifat 'point-and-click' untuk mendapatkan nama file tersebut. Kontrol yang dapat digunakan adalah OpenFileDialog.

Sebagai catatan, kontrol OpenFileDialog tidak memiliki interface (antarmuka) yang bisa dilihat pada saat mendesain form. Setelah user memasukkan kontrol OpenFileDialog ke dalam form, kontrol OpenFileDialog akan tampil di tray area.

Berikut adalah property umum pada OpenFileDialog:

**Tabel 1.30: Property OpenFileDialog**

Property	Keterangan
Name	Mengatur nama openFileDialog (OpenFileDialog umumnya dinamai dengan <b>dlgOpen</b> )
AddExtension	Menentukan apakah openFileDialog akan menambahkan extension secara otomatis pada nama file jika user tidak memasukkan extension
CheckFileExists	Menentukan apakah openFileDialog akan menampilkan peringatan jika nama file yang ingin dibuka belum ada pada lokasi penyimpanan





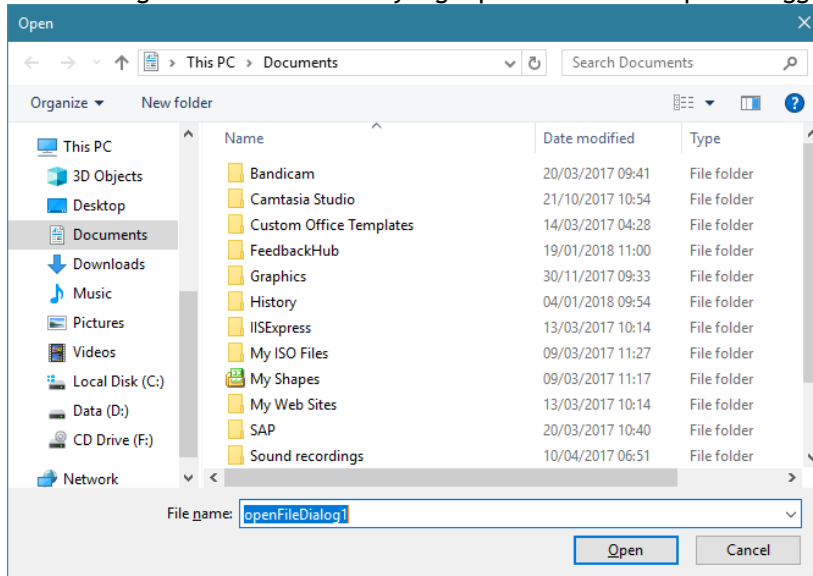
CheckPathExists	Menentukan apakah openFileDialog akan menampilkan peringatan jika user memasukkan lokasi yang tidak tersedia
DefaultExt	Menentukan extension default pada file
FileName	Menentukan string yang berisi nama file yang dipilih di dalam kotak dialog
Filter	Menentukan pilihan tipe file yang dapat dibuka ke dalam kotak dialog
FilterIndex	Menentukan indeks pilihan filter yang terpilih pada kotak dialog
InitialDirectory	Menentukan folder yang akan ditampilkan pertama kali ketika kotak dialog muncul
Title	Menentukan judul dari kotak dialog

Berikut adalah method umum pada OpenFileDialog:

**Tabel 1.31: Method OpenFileDialog**

Method	Keterangan
ShowDialog	Menampilkan kotak dialog, mengembalikan nilai yang mengindikasikan button mana yang dipilih oleh user: OK atau Cancel.

Untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult.



**Gambar 1.46: Tampilan Kontrol OpenFileDialog**

### TOPIC 3: FONTDIALOG

FontDialog menyediakan fitur bagi user untuk mengganti tipe, style, ukuran, dan warna font. Sebagai catatan, kontrol FontDialog tidak memiliki interface (antarmuka) yang bisa dilihat pada saat

mendesain form. Setelah user memasukkan kontrol FontDialog ke dalam form, kontrol FontDialog akan tampil di tray area.

Berikut adalah property umum pada FontDialog:



**Tabel 1.32: Property FontDialog**

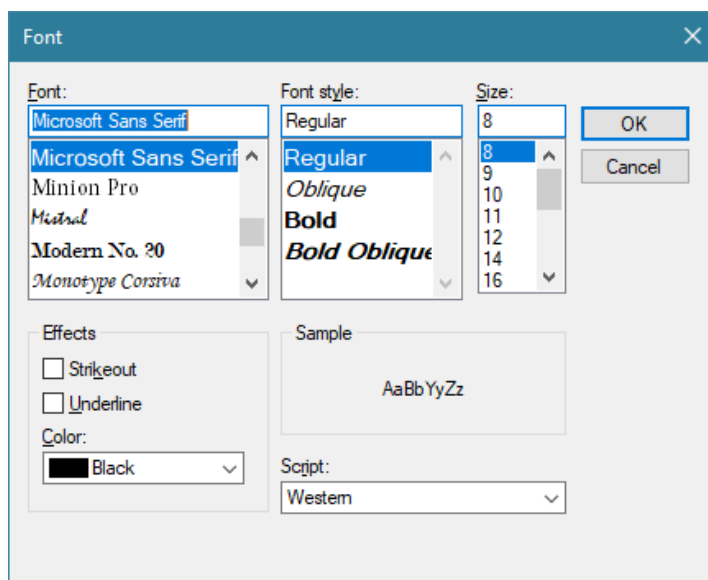
Property	Keterangan
Name	Mengatur nama fontdialog (FontDialog umumnya dinamai dengan <b>dlgFont</b> )
Color	Mengindikasikan warna font
Font	Mengindikasikan tipe font
MaxSize	Menentukan ukuran maksimum font dalam satuan point yang dapat dipilih
MinSize	Menentukan ukuran minimum font dalam satuan point yang dapat dipilih
ShowColor	Mengindikasikan apakah kotak dialog font menampilkan kotak pilihan warna. Nilai defaultnya adalah false.
ShowEffects	Mengindikasikan apakah kotak dialog font mengizinkan user untuk mengatur efek (strikeout, underline)

Berikut adalah method umum pada FontDialog:

**Tabel 1.33: Method FontDialog**

Method	Keterangan
ShowDialog	Menampilkan kotak dialog, mengembalikan nilai yang mengindikasikan button mana yang dipilih oleh user: OK atau Cancel.

Untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult.



**Gambar 1.47: Tampilan Kontrol FontDialog**

#### TOPIC 4: COLORDIALOG

ColorDialog mengizinkan user untuk memilih warna dari palet warna yang tersedia ataupun menambahkan warna custom ke palet tersebut. Warna yang terpilih dapat digunakan untuk mengatur property kontrol.

Sebagai catatan, kontrol ColorDialog tidak memiliki interface (antarmuka) yang bisa dilihat pada saat mendesain form. Setelah user memasukkan kontrol ColorDialog ke dalam form, kontrol ColorDialog akan tampil di tray area.

Berikut adalah property umum pada ColorDialog:

**Tabel 1.34: Property ColorDialog**

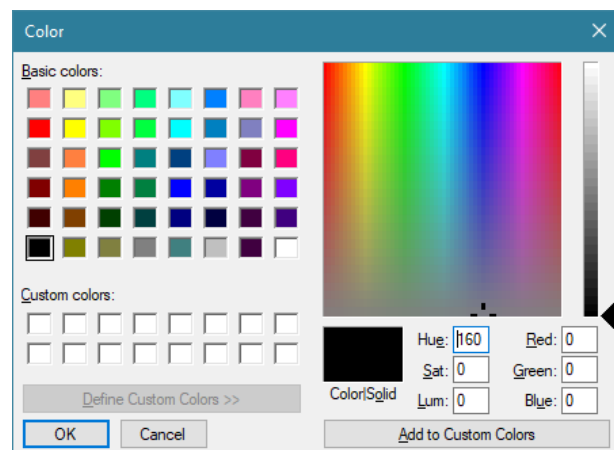
Property	Keterangan
Name	Mengatur nama colordialog (ColorDialog umumnya dinamai dengan <b>dlgColor</b> )
AllowFullOpen	Mengatur nilai yang mengindikasikan apakah user dapat menggunakan color dialog box untuk menentukan custom color
AnyColor	Mengatur nilai yang mengindikasikan apakah kotak dialog menampilkan seluruh warna yang tersedia pada kumpulan warna dasar
Color	Menentukan warna yang terpilih
CustomColors	Mengatur kumpulan custom color yang tampil pada kotak dialog
FullOpen	Mengatur nilai yang mengindikasikan apakah kontrol digunakan untuk menambahkan custom color ketika kotak dialog dibuka
SolidColorOnly	Mengatur nilai yang mengindikasikan apakah kotak dialog akan membatasi user untuk memilih warna solid saja.

Berikut adalah method umum pada ColorDialog:

**Tabel 1.35: Method ColorDialog**

Method	Keterangan
ShowDialog	Menampilkan kotak dialog, mengembalikan nilai yang mengindikasikan button mana yang dipilih oleh user: OK atau Cancel.

Untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult.



**Gambar 1.48: Tampilan Kontrol ColorDialog**

## TOPIC 5: PRINTDIALOG

PrintDialog mengizinkan user untuk memilih printer yang ingin digunakan, memilih orientasi halaman, rentang halaman yang ingin diprint, dan jumlah rangkap yang ingin diprint.

Sebagai catatan, kontrol PrintDialog tidak memiliki interface (antarmuka) yang bisa dilihat pada saat mendesain form. Setelah user memasukkan kontrol PrintDialog ke dalam form, kontrol PrintDialog akan tampil di tray area.

Property umum pada PrintDialog:

**Tabel 1.36: Property PrintDialog**

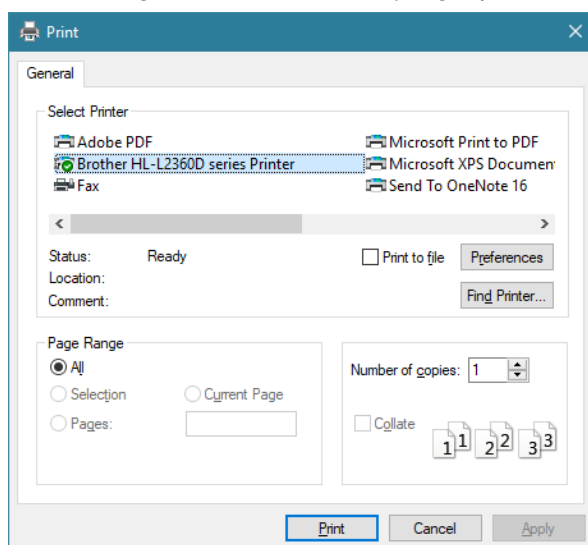
Property	Keterangan
Name	Mengatur nama printdialog (PrintDialog umumnya dinamai dengan <b>dlgPrint</b> )
AllowPrintToFile	Mengatur nilai yang mengindikasikan apakah checkbox print to file aktif atau tidak
AllowSelection	Mengatur nilai yang mengindikasikan apakah opsi From... To... Diaktifkan
AllowSomePages	Mengatur nilai yang mengindikasikan apakah tombol Pages option aktif atau tidak
Document	Sumber dokumen yang akan diprint
PrinterSettings	Mengatur PrinterSettings yang dimodifikasi kotak dialog
PrintToFile	Mengatur nilai yang mengindikasikan apakah checkbox print to file terpilih

Berikut adalah method umum pada PrintDialog:

**Tabel 1.37: Method PrintDialog**

Method	Keterangan
ShowDialog	Menampilkan kotak dialog, mengembalikan nilai yang mengindikasikan button mana yang dipilih oleh user: OK atau Cancel.

Untuk mengetahui button mana yang dipilih oleh user dapat menggunakan konstanta DialogResult.



**Gambar 1.49: Tampilan Kontrol PrintDialog**

## SOLUTION

Untuk menyelesaikan masalah yang ada pada case study di atas adalah dengan membuat aplikasi desktop sederhana menggunakan ADO .NET. Minimal di dalam aplikasi terdapat Form, serta beberapa kontrol lainnya, yaitu Form, Button, Label, TextBox, GroupBox atau Panel, CheckBox, RadioButton, ListBox, ComboBox, MessageBox, dan DialogBox.

## INSTRUCTION

1. Buka **IDE Microsoft Visual Studio 2015**
2. Buat sebuah project baru
3. Pilih Template Windows Forms App (.NET Framework)
4. Drag and Drop Kontrol yang diperlukan pada Form
5. Gunakan Aplikasi / Project yang telah dibuat

## EXERCISE

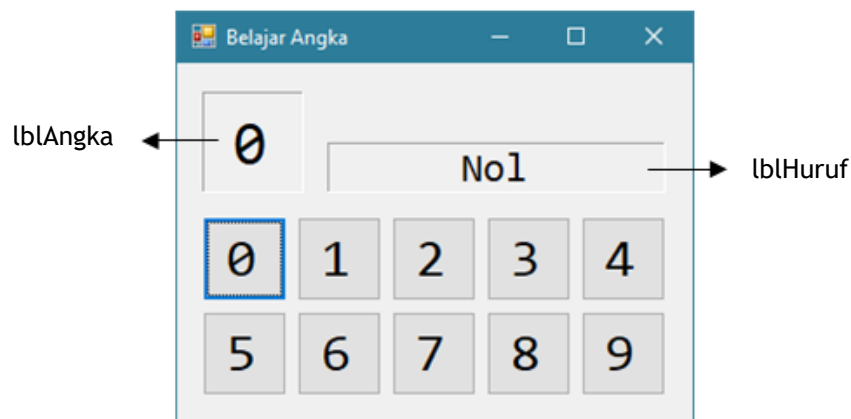
### EXERCISE OBJECTIVES

Mahasiswa diharapkan mampu:

- Mempraktikkan karakteristik dan penggunaan dari kontrol Button, Label, dan TextBox
- Mempraktikkan karakteristik dan penggunaan dari kontrol GroupBox, Panel, CheckBox, dan RadioButton
- Mempraktikkan karakteristik dan penggunaan dari kontrol ListBox dan ComboBox

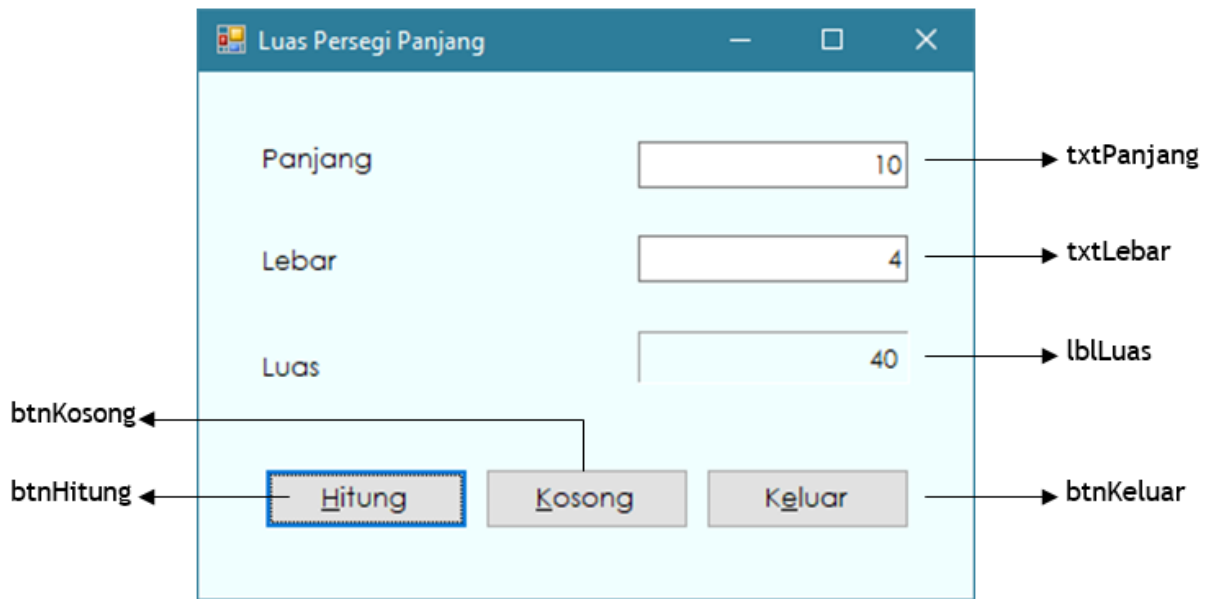
### TASK 1:

1. Buatlah aplikasi yang menampilkan angka dan teks dari Button angka yang diklik pada Label. Misalkan, Button 0 diklik, maka lblAngka menampilkan 0 dan lblHuruf menampilkan Nol.



2. Buatlah aplikasi yang dapat digunakan untuk menghitung luas persegi panjang. Berikan warna tulisan yang berbeda untuk TextBox yang sedang mendapatkan focus. Batasi nilai yang boleh diinput pada TextBox hanya berupa angka (numerik).  
Rumus: Luas persegi panjang = panjang x lebar.



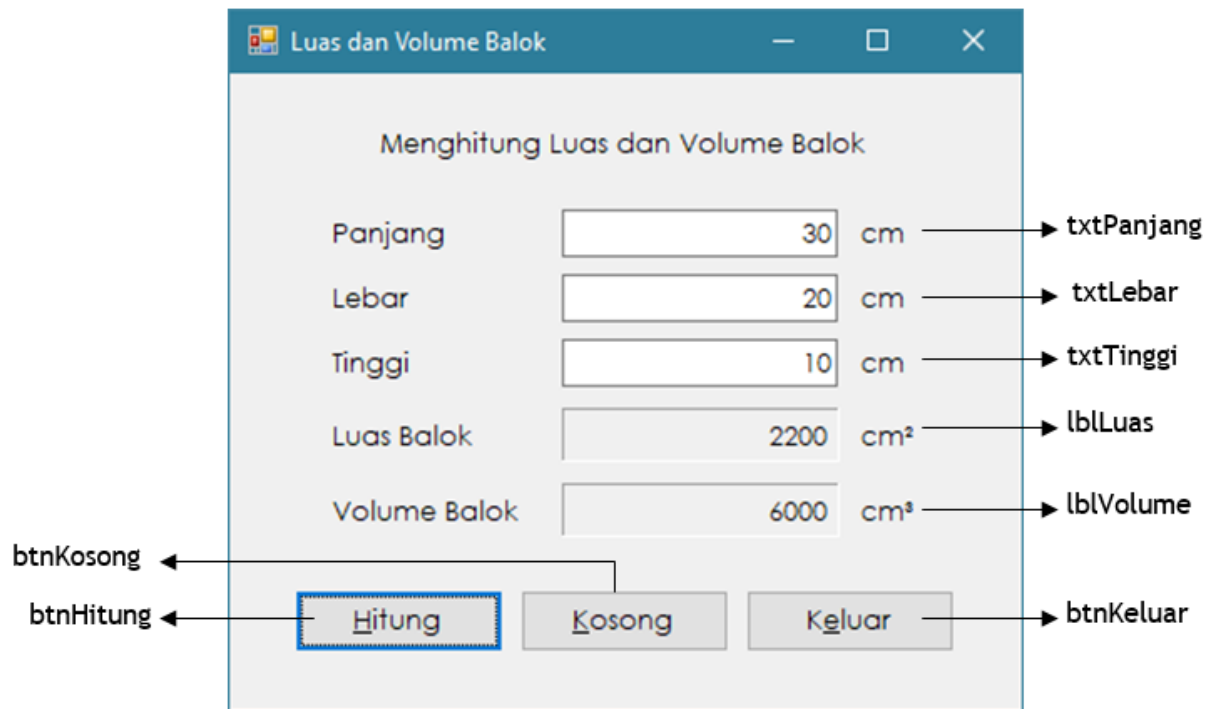


Atur property dari masing-masing object seperti pada tabel berikut ini.

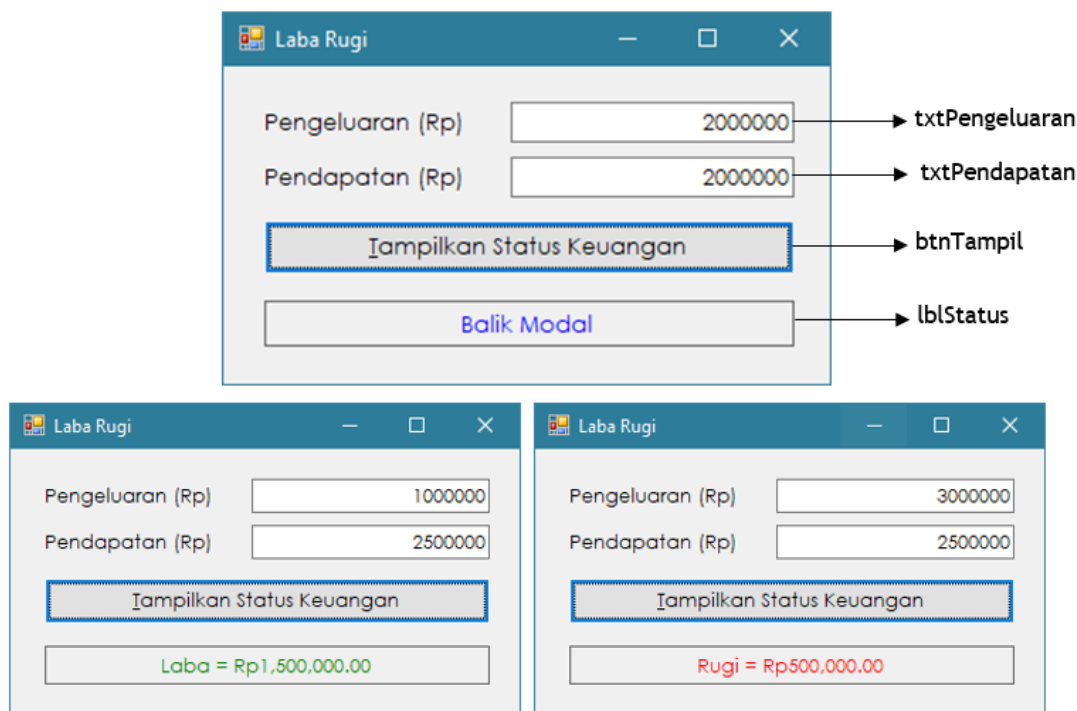
Object	Property	Nilai
frmTugas2	AcceptButton	btnHitung
	CancelButton	btnKeluar
	BackColor	Azure
	Font	Century Gothic, 10pt
	Text	Luas Persegi Panjang
txtPanjang, txtLebar	TextAlign	Right
lblLuas	AutoSize	False
	BorderStyle	Fixed3D
	TextAlign	MiddleRight
btnHitung	Text	&Hitung
btnKosong	Text	&Kosong
btnKeluar	Text	K&eluar

- Buatlah aplikasi yang dapat digunakan untuk menghitung luas dan volume balok. Berikan warna tulisan yang berbeda untuk TextBox yang sedang mendapatkan focus. Batasi nilai yang boleh diinput pada TextBox hanya berupa angka (numerik). Non-aktifkan Button Hitung apabila masih ada TextBox yang belum diisi.

Rumus: Luas Balok =  $2 \times (p \times l + p \times t + l \times t)$ , Volume Balok =  $p \times l \times t$ .

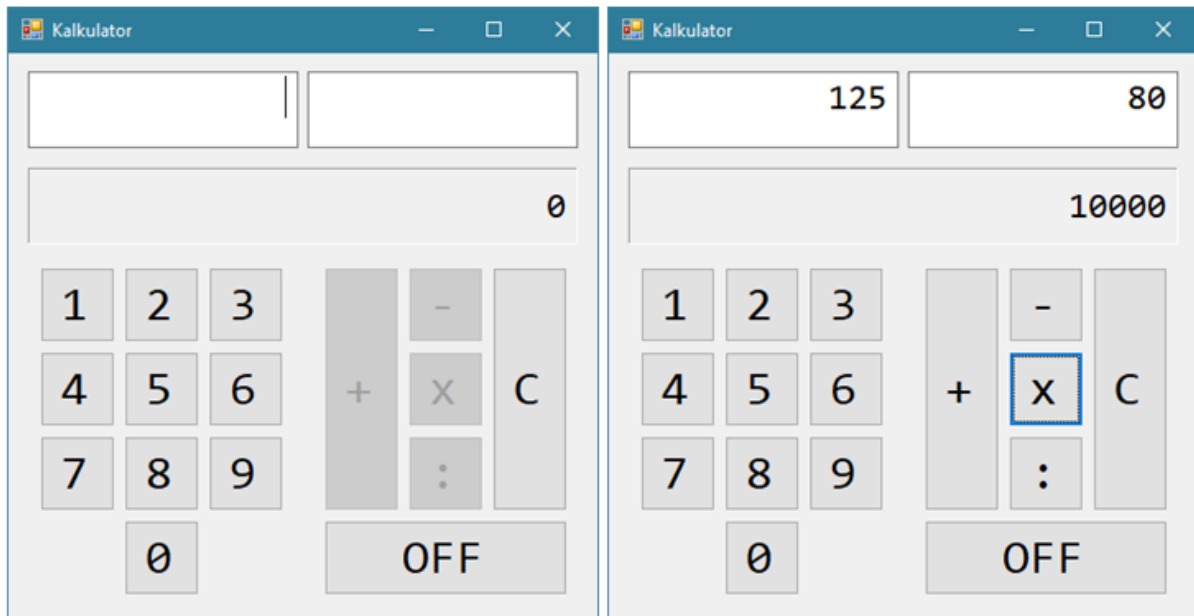


4. Buatlah aplikasi yang meminta nilai pengeluaran dan pendapatan dari sebuah perusahaan, dimana akan ditampilkan pesan “Balik Modal” jika pengeluaran = pendapatan, sedangkan jika sebaliknya akan ditampilkan pesan laba atau rugi. Berikan warna tulisan yang berbeda untuk TextBox yang sedang mendapatkan focus. Batasi nilai yang boleh diinput pada TextBox hanya berupa angka (numerik). Non-aktifkan Button Tampil apabila masih ada TextBox yang belum diisi.



5. Buatlah aplikasi kalkulator sederhana dimana user dapat menginput dua buah bilangan pada TextBox melalui button-button angka yang sudah disediakan. Batasi nilai yang boleh diinput pada TextBox hanya berupa angka (numerik). Button operator matematika (Tambah, Kurang, Kali, Bagi) baru aktif apabila user sudah mengisi kedua bilangan pada TextBox. Saat user menekan

salah satu button operator, lakukan operasi perhitungan yang sesuai (jika Button Tambah, maka lakukan penambahan bilangan, dst.), kemudian tampilkan hasil perhitungannya pada Label Hasil. Saat user menekan tombol C, maka kosongkan isian TextBox dan kembalikan kondisi Form seperti semula. Button OFF berfungsi untuk menutup aplikasi kalkulator.



## TASK 2:

1. Buatlah aplikasi yang dapat membantu seorang user untuk menghitung total harga yang harus dibayar dengan memilih berbagai pilihan menu makanan dan minuman yang tersedia.

Makanan			
Nasi Uduk	Rp	7000	<input checked="" type="checkbox"/>
Ayam Semur	Rp	23000	<input checked="" type="checkbox"/>
Ikan Asam Manis	Rp	35000	<input checked="" type="checkbox"/>
Bebek Bakar	Rp	40000	<input type="checkbox"/>
Udang Goreng	Rp	38000	<input checked="" type="checkbox"/>

Minuman			
Teh Manis	Rp	5000	<input checked="" type="checkbox"/>
Kopi	Rp	6500	<input type="checkbox"/>
Susu	Rp	6000	<input type="checkbox"/>
Jus Buah	Rp	9000	<input checked="" type="checkbox"/>
Soda	Rp	5500	<input checked="" type="checkbox"/>

Total : Rp122,500.00

2. Buatlah aplikasi untuk menampilkan Nama, Jenis Kelamin, dan Hobi yang di-input user pada sebuah Label saat user menekan Button Tampilkan Informasi.



Informasi Data Diri

Isi Data Diri

Nama: Sandy

Jenis Kelamin: ☐ Laki-Laki ☒ Perempuan

Hobi: ☒ Membaca ☒ Olahraga  
☒ Memasak ☐ Main Game  
☐ Melukis ☒ Lainnya

Tampilkan Informasi

```

=====
Nama : Sandy
Jenis Kelamin : Perempuan
=====
HOBII
=====
Membaca
Memasak
Olahraga
Lainnya
=====

```

3. Buatlah aplikasi untuk menampilkan mission-mission yang harus dijalani oleh seorang gamer. Game terdiri dari 10 mission. Aplikasi dapat menghitung jumlah mission yang sudah diselesaikan (Mission Completed) dan jumlah mission yang belum diselesaikan (Mission Not Completed).
  - Apabila gamer menyelesaikan 0-5 mission, tampilkan status “Not Completed” dengan latar berwarna merah.

Game

Mission

☒ Connect to Facebook  
☒ Complete the Tutorial  
☐ Fight 200 Times  
☐ Defeat 100 Enemies  
☐ Win 5 Consecutive Battles

☐ Eliminate Monsters 5 Times  
☒ Collect 100000 Coins  
☐ Upgrade Castle to Level 3  
☐ Recruit 15 Heroes  
☐ Reach Level 20

Mission Completed: 3

Mission Not Completed: 7

**Not Completed**

- Apabila gamer menyelesaikan 6-9 mission, tampilkan status “Almost There” dengan latar berwarna kuning.

The screenshot shows a window titled "Game" with a "Mission" section. It contains two columns of mission items, each with a checkbox. The first column has: ☒ Connect to Facebook, ☒ Complete the Tutorial, ☒ Fight 200 Times, ☐ Defeat 100 Enemies, and ☒ Win 5 Consecutive Battles. The second column has: ☐ Eliminate Monsters 5 Times, ☒ Collect 100000 Coins, ☒ Upgrade Castle to Level 3, ☒ Recruit 15 Heroes, and ☐ Reach Level 20. To the right, there are two input fields: "Mission Completed" with the value 7 and "Mission Not Completed" with the value 3. A large yellow button with the text "Almost There" is positioned below the input fields.

- Apabila gamer menyelesaikan 10 mission, tampilkan status “Mission Accomplished” dengan latar berwarna hijau.

The screenshot shows the same "Game" window as before, but now all mission checkboxes are checked, including "Reach Level 20". The "Mission Completed" input field now shows the value 10, and the "Mission Not Completed" input field shows the value 0. A large green button with the text "Mission Accomplished" is displayed below the input fields.

4. Buatlah aplikasi untuk menerima pesanan baju dengan inputan Nama, Ukuran, dan Pilihan Baju. Tanggal yang ditampilkan pada Label adalah tanggal sistem. Pilihan Baju ada 2: Biasa dan Custom. Apabila user memilih salah satu dari kedua Pilihan Baju tersebut, maka Panel di bawahnya akan aktif sehingga user dapat memilih lebih lanjut. TextBox di samping RadioButton baru akan aktif dan dapat diisi dengan keterangan apabila RadioButton bersangkutan dipilih. Button Pesan akan menampilkan pesanan baju pada Label di samping kanan. Button Kosong berfungsi membersihkan semua inputan user. Button Keluar berfungsi untuk menutup aplikasi.

The screenshot shows a window titled "Pemesanan Baju". It features a "Tanggal" label with a date input field showing "14/09/2018" and a "Nama" label with a text input field. Below these are two radio buttons: "Biasa" and "Custom". Under "Biasa", there are four radio buttons: "Merah", "Kuning", "Hijau", and "Lainnya" with an adjacent text input field. Under "Custom", there are four radio buttons: "Superhero", "Band", "Football", and "Quotes", each with an adjacent text input field. At the bottom, there are three buttons: "Pesan", "Kosong", and "Keluar". On the right side of the window is a large empty rectangular area for displaying the order details.

**Pemesanan Baju**

Tanggal: 14/09/2018  
 Nama: SpongeBob

Ukuran:  
☐ XL ☐ L ☐ M ☒ S

☒ Biasa ☐ Custom

☐ Merah  
☒ Kuning  
☐ Hijau  
☐ Lainnya:

☐ Superhero :  
☐ Band :  
☐ Football :  
☐ Quotes :

Pesan Kosong Keluar

Tanggal : 14/09/2018  
 Nama : SpongeBob

Ukuran : S  
 Pilihan : Biasa  
 Keterangan : Kuning

**Pemesanan Baju**

Tanggal: 14/09/2018  
 Nama: Tony Stark

Ukuran:  
☐ XL ☒ L ☐ M ☐ S

☐ Biasa ☒ Custom

☐ Merah  
☐ Kuning  
☐ Hijau  
☐ Lainnya:

☒ Superhero : IronMan  
☐ Band :  
☐ Football :  
☐ Quotes :

Pesan Kosong Keluar

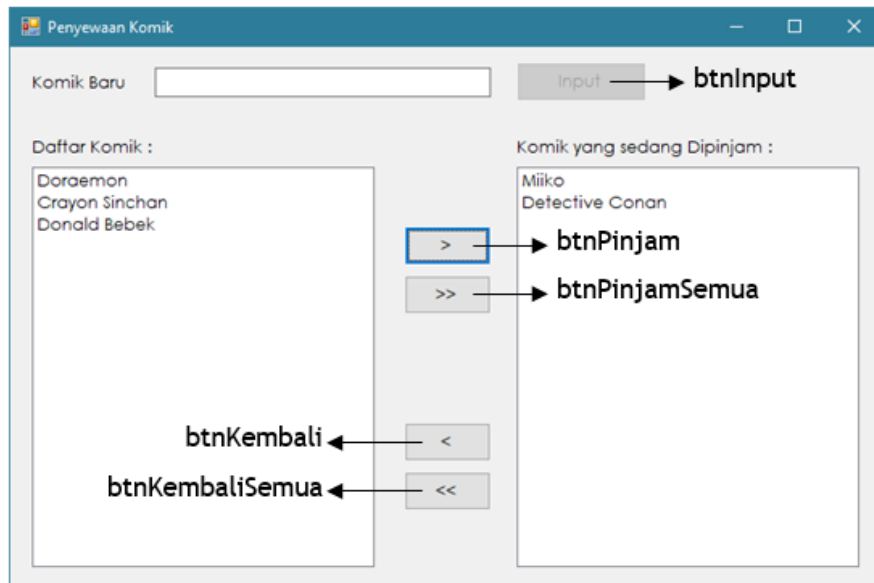
Tanggal : 14/09/2018  
 Nama : Tony Stark

Ukuran : L  
 Pilihan : Custom  
 Keterangan : Superhero - IronMan

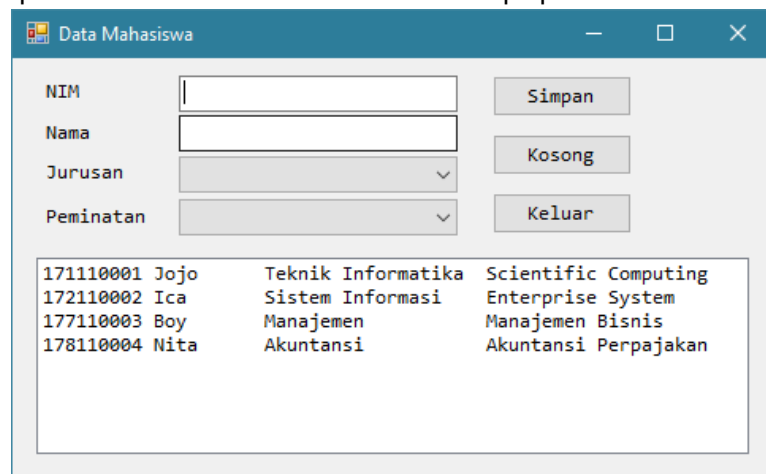
### TASK 3:

1. Buatlah aplikasi penyewaan komik yang dapat menampilkan Daftar Komik dan Komik yang sedang Dipinjam pada ListBox. User dapat menambahkan Komik Baru ke Daftar Komik dengan menginput judul pada TextBox dan menekan Button Input. Button Input baru aktif jika TextBox sudah terisi. Button Pinjam akan memindahkan komik yang dipilih dari Daftar Komik ke Komik yang sedang Dipinjam. Button Pinjam Semua akan memindahkan semua komik dari Daftar Komik ke Komik yang sedang Dipinjam. Button Kembali untuk mengembalikan komik yang dipilih dari Komik yang sedang Dipinjam ke Daftar Komik. Button Kembali Semua untuk mengembalikan semua komik dari Komik yang sedang Dipinjam ke Daftar Komik.





2. Buatlah aplikasi untuk menginput data mahasiswa dimana user dapat menginput NIM, Nama, Jurusan, dan Peminatan. Peminatan tergantung pada jurusan yang dipilih oleh user. Jurusan dan Peminatan yang tersedia terdiri dari:  
 Teknik Informatika : Scientific Computing, Mobile & Web  
 Sistem Informasi : e-Business, Enterprise System  
 Manajemen : Manajemen Bisnis, Manajemen Pemasaran  
 Akuntansi : Akuntansi Perpajakan, Akuntansi Keuangan  
 Button Simpan untuk menampilkan data mahasiswa ke ListBox. Button Kosong untuk membersihkan inputan user. Button Keluar untuk menutup aplikasi.



3. Buatlah aplikasi penjualan barang elektronik dimana Kategori Barang Elektronik terdiri dari: Televisi, Kulkas, dan AC. Merk Barang Elektronik yang tersedia di toko adalah: Samsung, LG, dan Sharp. List harga dari barang elektronik yang dijual adalah sebagai berikut:

Kategori	Merk	Harga
Televisi	Samsung	Rp2.500.000
	LG	Rp1.900.000
	Sharp	Rp1.500.000
Kulkas	Samsung	Rp5.600.000
	LG	Rp5.200.000
	Sharp	Rp6.000.000
AC	Samsung	Rp3.100.000
	LG	Rp3.700.000
	Sharp	Rp3.250.000

Harga akan otomatis muncul ketika user memilih Kategori dan Merk Barang Elektronik. Button Beli baru akan aktif apabila Kategori dan Merk sudah dipilih. Button Beli untuk menampilkan data barang yang dibeli ke Label di bawah. Button Kosong untuk membersihkan pilihan user. Button Keluar untuk menutup aplikasi.

4. Buatlah aplikasi untuk melakukan Konversi Satuan dimana satuan yang disediakan terdiri dari satuan Panjang dan Berat. Satuan Panjang yang ditampilkan pada ListBox adalah: Feet, Inci, dan Centimeter. Satuan Berat yang ditampilkan pada ListBox adalah: Ton, Kilogram, dan Ons. Button Konversi baru akan aktif apabila ListBox Dari dan ListBox Ke sudah dipilih satuannya, serta TextBox Nilai sudah diisi. Batasi nilai yang boleh diinput pada TextBox hanya berupa angka numerik). Apabila user menekan Button Konversi, tampilkan hasil perhitungan pada Label Hasil Konversi dengan acuan sebagai berikut:
  - Satuan Panjang:
    - Feet = 12 Inci
    - 1 Feet = 30.48 Centimeter

- 1 Inci = 0.0833333 Feet
- 1 Inci = 2.54 Centimeter
- 1 Centimeter = 0.0328084 Feet
- 1 Centimeter = 0.393701 Inci
- Satuan Berat:
  - 1 Ton = 1000 Kilogram
  - 1 Ton = 10000 Ons
  - 1 Kilogram = 0.001 Ton
  - 1 Kilogram = 10 Ons
  - 1 Ons = 0.0001 Ton
  - 1 Ons = 0.1 Kilogram

## UNIT SUMMARY

Sekarang mahasiswa mampu:

- Membedakan istilah Object, Property, Method, dan Event
- Mempraktikkan langkah-langkah pengembangan suatu aplikasi bisnis
- Mempraktikkan dasar-dasar pengoperasian Microsoft Visual Studio 2015
- Mempraktikkan jenis-jenis kontrol dan cara penamaan pada sebuah kontrol
- Mempraktikkan karakteristik dan penggunaan dari Form
- Mempraktikkan karakteristik dan penggunaan dari kontrol Button, Label, dan TextBox
- Mempraktikkan karakteristik dan penggunaan kontrol GroupBox, Panel, CheckBox, dan RadioButton



- Mengenal karakteristik dan penggunaan kontrol ListBox dan ComboBox
- Mempraktikkan karakteristik dan penggunaan dari kontrol MessageBox
- Mempraktikkan karakteristik dan penggunaan dari kontrol DialogBox

## POST LAB

### QUESTION

1. Pada saat kursor memasuki txtPanjang (event: \_\_\_\_\_), maka tulisannya akan berubah menjadi warna Biru. Kemudian pada saat kursor meninggalkan txtPanjang (event: \_\_\_\_\_), maka tulisannya akan kembali menjadi warna Hitam.
  - ☐ A. Leave, Enter
  - ☐ B. KeyUp, KeyDown
  - ☐ C. Enter, Leave
  - ☐ D. TextChanged, Leave
  - ☐ E. TextChanged, TextChanged
2. Untuk membatasi agar pengguna hanya boleh memasukkan angka ke dalam TextBox, kita bisa melakukan coding pada event?
  - ☐ A. TextChanged
  - ☐ B. KeyDown
  - ☐ C. Input
  - ☐ D. KeyPress
  - ☐ E. Enter
3. Manakah di bawah ini property yang digunakan untuk mengatur warna tulisan?
  - ☐ A. Font
  - ☐ B. BackColor
  - ☐ C. ForeColor
  - ☐ D. Text
  - ☐ E. BorderStyle
4. Manakah di bawah ini event pada TextBox yang terjadi ketika TextBox kehilangan fokus?
  - ☐ A. Enter
  - ☐ B. Click
  - ☐ C. KeyPress
  - ☐ D. Leaver
  - ☐ E. TextChanged
5. Property yang digunakan untuk mengganti Nama File dari Form adalah?
  - ☐ A. Name
  - ☐ B. FileName
  - ☐ C. Text
  - ☐ D. Font
  - ☐ E. TextAlign
6. Manakah di bawah ini pernyataan yang benar?
  - ☐ A. Panel memiliki fungsi yang sama dengan GroupBox yaitu untuk mengelompokkan kontrol
  - ☐ B. Panel dan GroupBox bukan merupakan kontrol
  - ☐ C. Panel memiliki BorderStyle, sedangkan GroupBox tidak



- ☐ D. Panel tidak memiliki property Text untuk memberi Judul, sedangkan GroupBox punya
  - ☐ E. Panel dapat mengelompokkan kontrol, sedangkan GroupBox tidak dapat mengelompokkan kontrol
7. Event ketika pengguna memilih RadioButton dan CheckBox adalah?
- ☐ A. Click
  - ☐ B. Checked
  - ☐ C. CheckedChanged
  - ☐ D. TextChanged
  - ☐ E. Enter
8. Untuk menyimpan (Save) dan membuka (Open) file dengan memilih lokasi penyimpanan file yang diinginkan, kita dapat menggunakan?
- ☐ A. Open File & Save File
  - ☐ B. SaveFile dan OpenFile
  - ☐ C. SaveFileDialog dan OpenFileDialog
  - ☐ D. OpenSaveFileDialog
  - ☐ E. SaveOpenFileDialog
9. Untuk menghapus item di dalam ListBox, kita dapat menggunakan perintah?
- ☐ A. Remove
  - ☐ B. Text = "";
  - ☐ C. RemoveAt
  - ☐ D. Clear
  - ☐ E. Text = 0;
10. Manakah di bawah ini pernyataan yang benar?
- ☐ A. CheckBox digunakan untuk membuat serangkaian pilihan dimana user dapat memilih 1 atau lebih dari pilihan-pilihan yang ada
  - ☐ B. RadioButton digunakan untuk membuat serangkaian pilihan dimana user hanya dapat memilih 1 pilihan dari berbagai pilihan yang ada
  - ☐ C. Perbedaan ComboBox dan ListBox terdapat pada pilihan item. Pada ComboBox hanya satu item pilihan yang terlihat sebelum ComboBox diklik, sedangkan pada ListBox dapat menampilkan beberapa pilihan
  - ☐ D. Perbedaan ComboBox dan ListBox terdapat pada Event-nya
  - ☐ E. ComboBox dan ListBox penggunaan property dan eventnya hampir sama
11. MessageBox dapat digunakan untuk?
- ☐ A. Mengatasi error yang ada
  - ☐ B. Menampilkan pesan dan icon
  - ☐ C. Menampilkan pesan error
  - ☐ D. Menjelaskan masalah potensial
  - ☐ E. Menambah error
12. Cara untuk membersihkan RadioButton pada form di bawah ini adalah?
- ☐ A. Atur Properties Checked menjadi true
  - ☐ B. Atur Properties Text menjadi ""
  - ☐ C. Atur Properties Checked menjadi ""
  - ☐ D. Pakai metode Clear()
  - ☐ E. Atur Properties Checked menjadi false





13. GroupBox digunakan untuk?

- ☐ A. Mengelompokkan sekumpulan Event dalam aplikasi Visual C#
- ☐ B. Mengelompokkan sekumpulan kontrol yang saling berhubungan dalam aplikasi Visual C#
- ☐ C. Mengelompokkan beberapa kontrol yang saling berhubungan dalam aplikasi Visual C#
- ☐ D. Menggabungkan kontrol yang berbeda dalam aplikasi Visual C#
- ☐ E. Memisahkan sekumpulan kontrol yang saling berhubungan dalam aplikasi Visual C#

14. Untuk menggunakan MessageBox, programmer harus menentukan ?

- ☐ A. Caption yang berisi judul MessageBox
- ☐ B. Icon yang tidak sesuai
- ☐ C. Text yang berisi pesan yang ingin disampaikan
- ☐ D. Button yang berisi pilihan tombol yang tampil pada MessageBox
- ☐ E. Default Button untuk menentukan tombol mana yang dijadikan default

15. DialogBox yang menyediakan fitur bagi user untuk mengganti tipe, style, ukuran, dan warna font adalah ?

- ☐ A. SaveFileDialog
- ☐ B. OpenFileDialog
- ☐ C. FontDialog
- ☐ D. ColorDialog
- ☐ E. PrintDialog





UNIVERSITAS  
**MIKROSKIL**