**MEMBUAT PROGRAM CONSOLE SEDERHANA DENGAN BAHASA PEMROGRAMAN C#**

PRAKTIKUM 3

Laporan ini dibuat untuk memenuhi persyaratan kelulusan

mata kuliah pemrograman II



NAMA : WILDAN AZRIL ARVANY

NPM : 1204011

KELAS : 2A

PRODI : D-IV TEKNIK INFORMATIKA

**PROGRAM STUDI D IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**2021**

**DAFTAR ISI**

**DAFTAR ISI1**

**TABEL AKTIVITAS2**

**BAB I PENDAHULUAN3**

1.1 Data Field3

1.2 Method3

1.3 Contructor4

**BAB II PEMBAHASAN5**

2.1 Percobaan 15

2.2 Percobaan 28

2.1 Percobaan 310

2.1 Percobaan 412

2.1 Jawaban Praktikum15

**BAB III PENUTUP15**

3.1 Kesimpulan15

3.2 Saran16

**DAFTAR PUSTAKA17**

**TABEL AKTIVITAS**

|  |  |
| --- | --- |
| **TANGGAL** | **CATATAN** |
| 19 November 2021 | Mempelajari Modul dan Mengerjakan Laporan Bab 1 |
| 20 November 2021 | Mengerjakan Laporan Bab 1 dan Bab 3 |
| 21 November 2021 | Mengerjakan Percobaan 1 dan 2 |
| 22 November 2021 | Mengerjakan Percobaan 3 dan 4 |
| 23 Novemver 2021 | Mengisi laporan bab 2 dan selesai |

**BAB I**

**PENDAHULUAN**

* 1. **DATA FIELD**

Secara umum, Data Field digunakan untuk mendefinisikan variabel di kelas dengan aksesibilitas sebagai pribadi atau dilindungi. Data Field yang perlu diekspos di mana saja di luar kelas dapat dienkapsulasi sebagai metode publik, properti, atau pengindeks. Data Field juga dikenal sebagai variabel tingkat kelas atau variabel anggota.

Sebuah data field, atau biasa disebut field, sejatinya adalah sebuah konstanta atau variabel dengan tipe data apapun yang dideklarasikan langsung pada sebuah class. Field digunakan untuk merepresentasikan karakteristik atau kondisi sebuah objek yang dibuat dari class tersebut. Field biasanya menyimpan data yang harus bisa diakses oleh lebih dari satu method yang didefinisikan di dalam sebuah class. Apabila Anda menginginkan sebuah variabel yang hanya digunakan oleh satu buah method saja, lebih baik Anda mendeklarasikan variabel tersebut sebagai variabel lokal di dalam method yang menggunakannya saja.

Data Field, dalam C#, adalah anggota kelas atau objek jenis apa pun yang mewakili lokasi memori untuk menyimpan nilai. Data Field digunakan untuk menyimpan data yang harus dapat diakses oleh beberapa metode kelas dan tersedia sepanjang masa hidup suatu objek. Data Fields memungkinkan kelas atau struct untuk merangkum data dengan opsi untuk menentukan aksesibilitasnya di berbagai tingkat.

* 1. **METHOD**

Untuk mempresentasikan perilaku sebuah objek yang dibuat dari sebuah class, dapat didefinisikan dalam method di dalam class tersebut. Method berisi intruksi intruksi yang membentuk sebuah mekanisme untuk menjalankan operasi tertentu. Penggunaan metode yang tepat membawa keuntungan sebagai berikut:

* + Mengurangi duplikasi kode
  + Menguraikan masalah yang kompleks menjadi bagian-bagian yang lebih sederhana
  + Meningkatkan kejelasan kode
  + Penggunaan kembali kode
  + Penyembunyian informasi

Dalam pemrograman berorientasi objek bekerja dengan objek. Objek adalah blok bangunan dasar dari sebuah program. Objek terdiri dari data dan metode. Metode mengubah keadaan objek yang dibuat. Metode adalah bagian dinamis dari objek; data adalah bagian statis. Metode adalah blok kode yang berisi serangkaian pernyataan. Metode harus dideklarasikan di dalam kelas atau struktur. Ini adalah praktik pemrograman yang baik bahwa metode hanya melakukan satu tugas tertentu. Metode membawa modularitas ke program.

* 1. **CONSTRUCTOR**

Dalam pemrograman C#, constructor merupakan sebuah method khusus yang akan dipanggil pada saat sebuah objek (instance) dari sebuah class dibuat. Constructor mempunyai nama atau identifier yang sama persis dengan identifier class -nya. Constructor bermanfaat untuk menginisialisasi beberapa field yang membutuhkan nilai default. Dengan kata lain, constructor diperlukan untuk “membangun” (construct) objek dari sebuah class.

Pada dasarnya sebuah class akan memiliki minimal sebuah constructor. Jadi, meskipun tidak mendefinisikan sebuah constructor secara eksplisit, secara otomatis compiler akan membuat default constructor dari class yang dideklarasikan. Meskipun constructor merupakan sebuah method, sebuah constructor tidak bisa memiliki tipe nilai balik (return type) apapun, bahkan void sekalipun.

Di c#, Konstruktor adalah metode yang akan dipanggil secara otomatis setiap kali instance kelas atau struct dibuat. Konstruktor akan memiliki nama yang sama dengan kelas atau struct, dan berguna untuk menginisialisasi dan menetapkan nilai default untuk anggota data objek baru. Jika kita membuat sebuah kelas tanpa memiliki konstruktor, maka kompiler akan secara otomatis membuat satu konstruktor default untuk kelas tersebut. Jadi, selalu ada satu konstruktor yang akan ada di setiap kelas. Dalam c#, sebuah kelas dapat berisi lebih dari satu konstruktor dengan tipe argumen yang berbeda. Konstruktor tidak akan pernah mengembalikan apa pun, jadi kita tidak perlu menggunakan tipe pengembalian apa pun, bahkan void, saat mendefinisikan metode konstruktor di kelas.

**BAB II**

**PEMBAHASAN**

**2.1 PERCOBAAN 1**

|  |  |
| --- | --- |
| **Repository Github** | <https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_3/tree/main/P3_1_1204011> |
| **Last Commit** | **23 November 2021** |

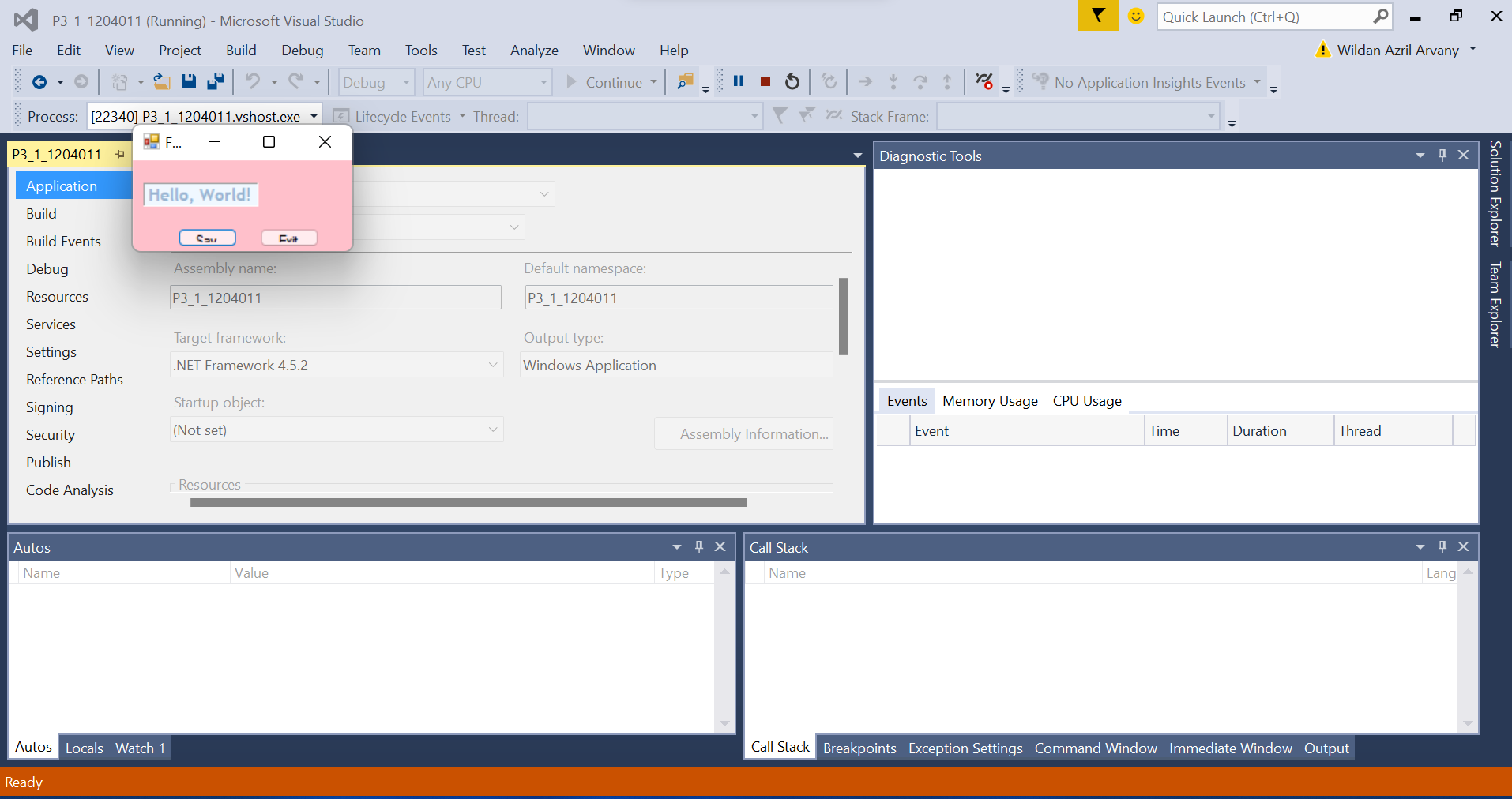
**Pertanyaan :**

* Berdasarkan topik sebelumnya mengenai OOP, menurut Anda Form1 merupakan sebuah kelas atau Obyek? Jelaskan! Pada bagian mana di kode yang menunjukkan hal tersebut? Form adalah sebuah class. Lihat pada ujung atas code window dari sebuah form: Public Class Form1: “Public” Artinya class ini bisa di akses oleh class lainya.“Form1” adalah nama dari class ini. Pada bagian paling bawah code window dari form ini, didapati “End Class”, menandakan berakhirnya code dari class form1.

Ditunjukkan dengan kode “**Public Partial Class Form1 : Form”**

* Berdasarkan topik OOP sebelumnya, HelloButton\_Click() ini merupakan sebuah ? Method karena ada berada didalam class

**Output :**

****

|  |  |
| --- | --- |
| using System.Threading.Tasks;**;** | |
| **Penjelasan** | Threading Tasks adalah suatu kelas yang memungkinkan untuk membuat tugas dan menjalankannya secara asinkron. Tugas tersebut merupakan objek yang mewakili beberapa pekerjaan yang harus dilakukan, serta dapat memberi informasi jika pekerjaan selesai dan operasi mengembalikan hasil |

|  |  |
| --- | --- |
| namespace P1\_1\_1204011 | |
| **Penjelasan** | Namespace pada C# berfungsi untuk mengatur banyak kelas sehingga mudah menangani aplikasi |

|  |  |
| --- | --- |
| public Form1()  {  InitializeComponent();  } | |
| **Penjelasan** | Constructor dari Form1 yang memanggil function InitializeComponent yang memanggil metode yang dibuat dan dikelola secara otomatis oleh desainer Windows Forms dan mendefinisikan semua yang Anda lihat di form. |

|  |  |
| --- | --- |
| private void HelloButton\_Click(object sender, EventArgs e)  {  OutputLabel.Text = "Hello, World";  } | |
| **Penjelasan** | Sebuah event catcher untuk tombol HelloButton\_Click yang akan mengubah OutputLabel.Text menjadi Hello World. |

|  |  |
| --- | --- |
| private void ExitButton\_Click(object sender, EventArgs e)  {  this.Close();  } | |
| **Penjelasan** | Sebuah function ExitButton\_Click event catcher untuk tombol Exit yang memanggil metode untuk menutup form. |

|  |  |
| --- | --- |
| [STAThread] | |
| **Penjelasan:** | pada dasarnya merupakan persyaratan untuk pompa pesan Windows untuk berkomunikasi dengan komponen COM |

|  |  |
| --- | --- |
| Application.EnableVisualStyles(); | |
| **Penjelasan:** | Mengaktifkan metode memungkinkan gaya visual untuk aplikasi. Gaya visual adalah warna, font, dan elemen visual lainnya yang membentuk form dengan tema sistem operasi |

|  |  |
| --- | --- |
| Application.SetCompatibleTextRenderingDefault(false); | |
| **Penjelasan:** | Properti UseCompatibleTextRendering dimaksudkan untuk menyediakan kompatibilitas visual antara kontrol Windows Forms yang membuat teks menggunakan kelas TextRenderer dan aplikasi .NET Framework 1.0 dan .NET Framework 1.1 yang melakukan rendering teks kustom menggunakan kelas Grafik. |

|  |  |
| --- | --- |
| Application.Run(new Form1()); | |
| **Penjelasan:** | Mengeksekusi aplikasi dengan basis Form1 |

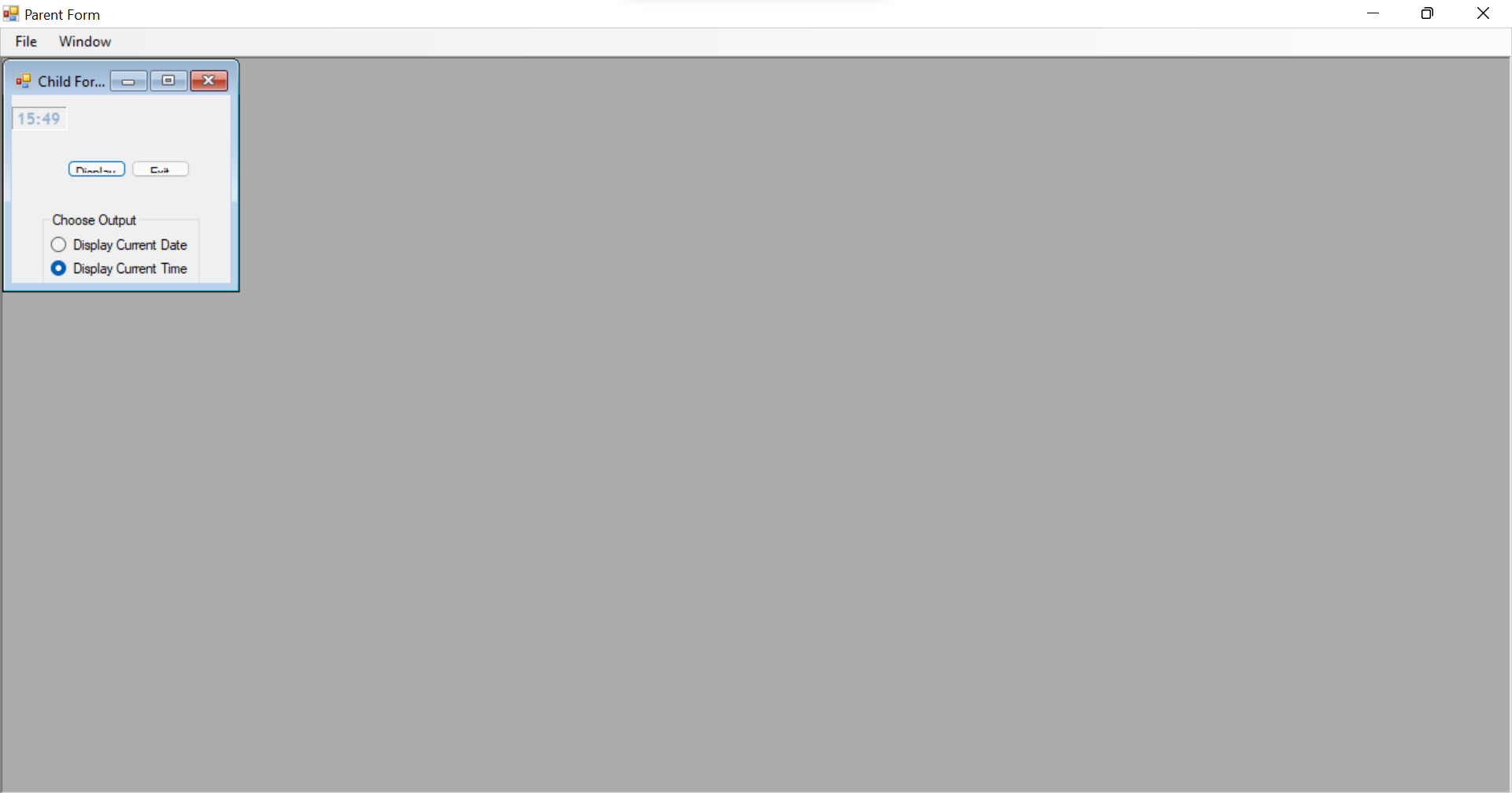
**2.2 PERCOBAAN 2**

|  |  |
| --- | --- |
| **Repository Github** | <https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_3/tree/main/P3_2_1204011> |
| **Last Commit** | **23 November 2021** |

**Pertanyaan :**

**•** Apa yang terjadi? Mengapa terjadi demikian? Jelaskan dan perbaiki! Mengganti AutoSize menjadi False karena autosize akan langsung mengubah ukuran size dan Terjadi error, karena tidak terdapat pperintah untuk menampilkan form dari ChildFrom, dan untuk menampilkannya perlu ditambahkan perintah private void ChildForm\_Load(object sender, EventArgs e

**Output :**

****

|  |  |
| --- | --- |
| public partial class ParentForm : Form | |
| **Penjelasan:** | Ini memberikan kemampuan khusus untuk mengimplementasikan  fungsionalitas kelas tunggal ke dalam beberapa file dan semua file ini  digabungkan menjadi file kelas tunggal saat aplikasi dikompilasi |

|  |  |
| --- | --- |
| private void ExitButton\_Click(object sender, EventArgs e)  {  this.Close();  }  private void DisplayButton\_Click(object sender, EventArgs e)  {  OutputLabel.Text = outputText;  } | |
| **Penjelasan:** | Ini adalah event – event handler untuk tombol display dan tombol exit |

|  |  |
| --- | --- |
| private void ExitMenuItem\_Click(object sender, EventArgs e)  {  this.Close();  } | |
| **Penjelasan:** | Menu item yang berada di menustrip yang men-close program |

|  |  |
| --- | --- |
| private void DateOption\_Click(object sender, EventArgs e)  {  outputText = DateTime.Now.ToString("d");  }  private void TimeOption\_Click(object sender, EventArgs e)  {  outputText = DateTime.Now.ToString("t");  } | |
| **Penjelasan:** | Ini adalah event – event handler untuk radiobutton Date dan Time |

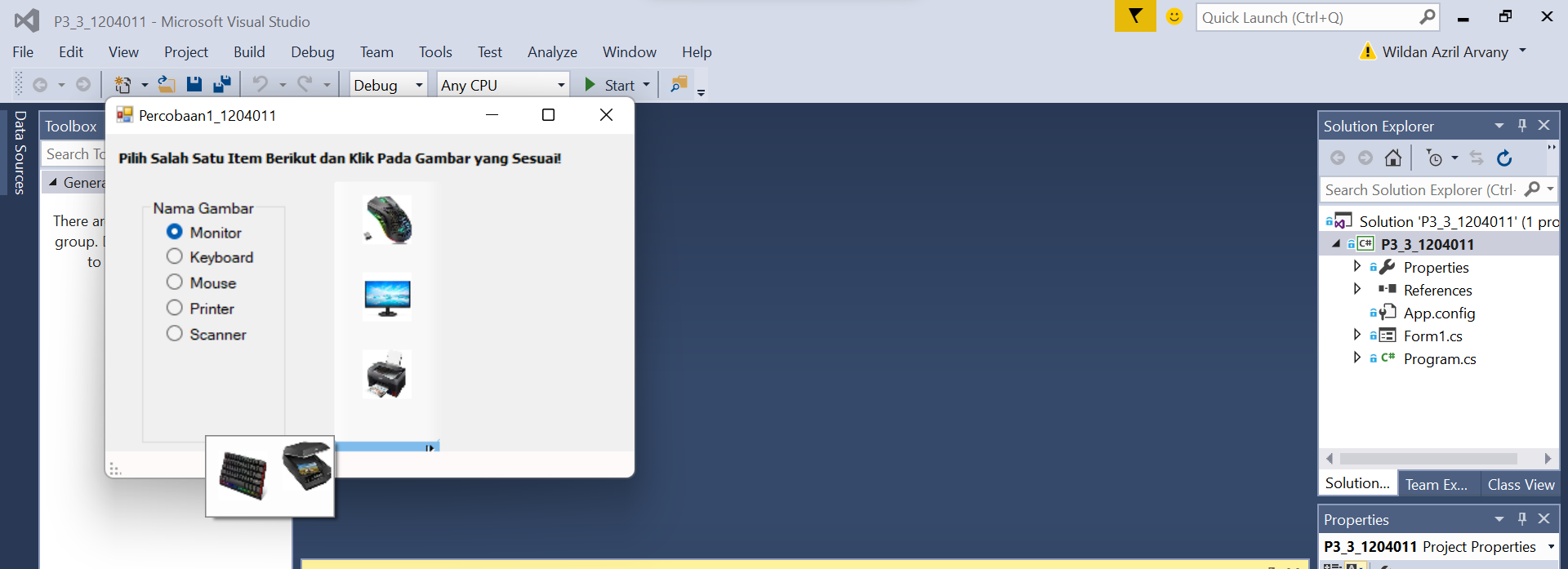
|  |  |
| --- | --- |
| private void NewMenuItem\_Click(object sender, EventArgs e)  {  ChildForm childform = new ChildForm();  childform.MdiParent = this;  childform.Show();  } | |
| **Penjelasan:** | Menu item new yang akan membuat Childform yang berada di mdiparent |

|  |  |
| --- | --- |
| private void WindowCascadeMenuItem\_Click(object sender, EventArgs e)  {  this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);  }  private void WindowTileMenuItem\_Click(object sender, EventArgs e)  {  this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal);  } | |
| **Penjelasan:** | Menustrip yang berisi Casade dan Window yang akan mengubah ukuran window childform |

**2.3 PERCOBAAN 3**

|  |  |
| --- | --- |
| **Repository Github** | <https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_3/tree/main/P3_3_1204011> |
| **Last Commit** | **23 November 2021** |

**Output :**

****

|  |  |
| --- | --- |
| private void tbsmouse\_Click(object sender, EventArgs e)  {  if (rbMouse.Checked == true)  {  tslKetGambar.Text = "Cocok! ini adalah gambar mouse";  }  else  {  tslKetGambar.Text = "Tidak Cocok! Anda memilih gambar yang salah \nIni adalah gambar mouse";  }  } | |
| **Penjelasan:** | Sebuah private function yang menjadi event catcher untuk toolstripbutton mouse |

|  |  |
| --- | --- |
| private void tbsprinter\_Click(object sender, EventArgs e)  {  if (rbPrinter.Checked == true)  {  tslKetGambar.Text = "Cocok! ini adalah gambar Printer";  }  else  {  tslKetGambar.Text = "Tidak Cocok! Anda memilih gambar yang salah \nIni adalah gambar Printer";  }  } | |
| **Penjelasan:** | Sebuah private function yang menjadi event catcher untuk toolstripbutton printer |

|  |  |
| --- | --- |
| private void tbskeyboard\_Click(object sender, EventArgs e)  {  if (rbKeyboard.Checked == true)  {  tslKetGambar.Text = "Cocok! ini adalah gambar Keyboard";  }  else  {  tslKetGambar.Text = "Tidak Cocok! Anda memilih gambar yang salah \nIni adalah gambar Keyboard";  }  } | |
| **Penjelasan:** | Sebuah private function yang menjadi event catcher untuk toolstripbutton keyboard |

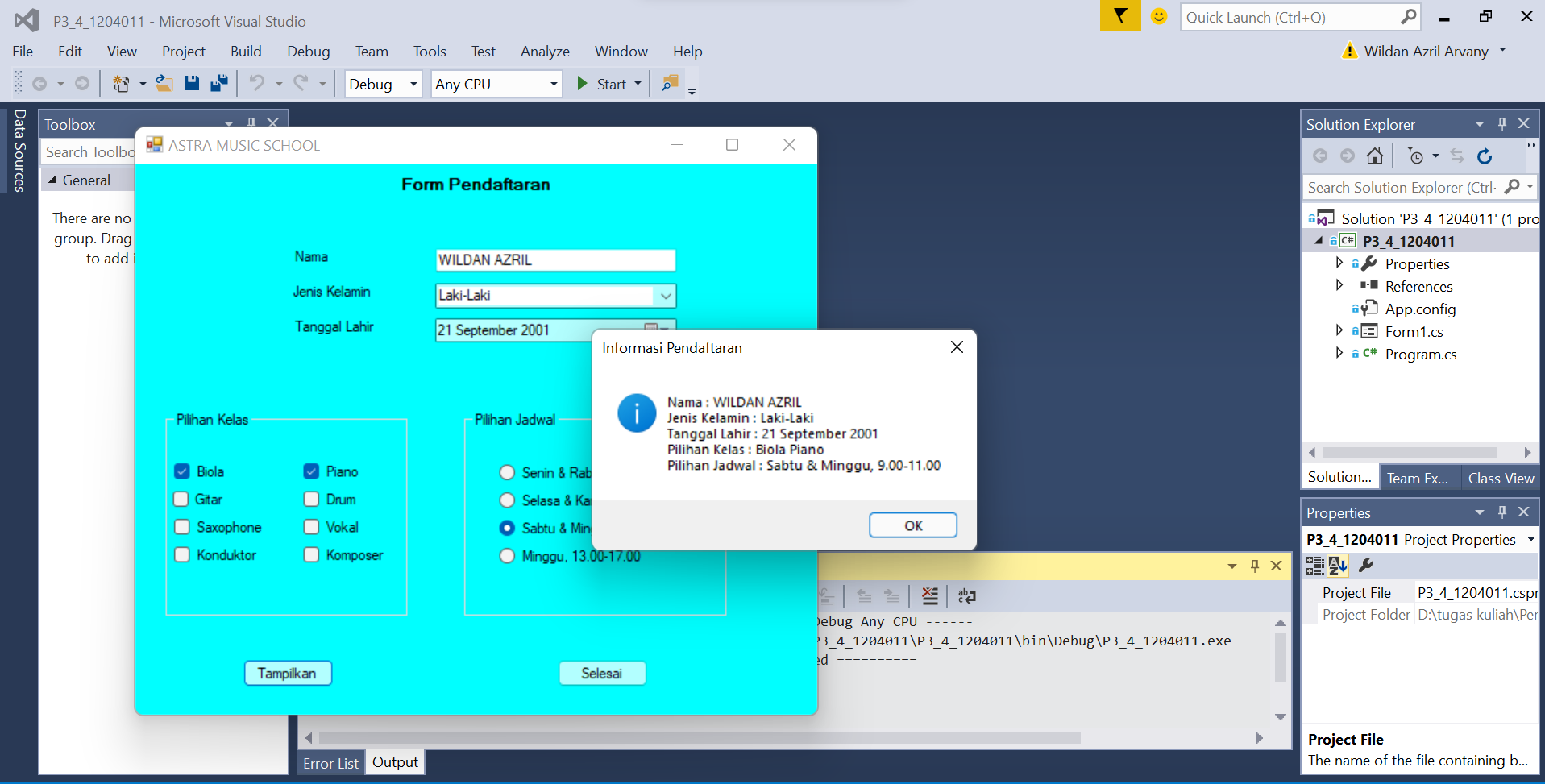
|  |  |
| --- | --- |
| private void tsbScanner\_Click(object sender, EventArgs e)  {  if (rbScanner.Checked == true)  {  tslKetGambar.Text = "Cocok! ini adalah gambar Scanner";  }  else  {  tslKetGambar.Text = "Tidak Cocok! Anda memilih gambar yang salah \nIni adalah gambar Scanner";  }  } | |
| **Penjelasan:** | Sebuah private function yang menjadi event catcher untuk toolstripbutton scanner |

|  |  |
| --- | --- |
| private void tbsMonitor\_Click(object sender, EventArgs e)  {  if (rbMonitor.Checked == true)  {  tslKetGambar.Text = "Cocok! ini adalah gambar Monitor";  }  else  {  tslKetGambar.Text = "Tidak Cocok! Anda memilih gambar yang salah \nIni adalah gambar Monitor";  }  } | |
| **Penjelasan:** | Sebuah private function yang menjadi event catcher untuk toolstripbutton monitor |

**2.4 PERCOBAAN 4**

|  |  |
| --- | --- |
| **Repository Github** | <https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_3/tree/main/P3_4_1204011> |
| **Last Commit** | **23 November 2021** |

**Output :**

****

|  |  |
| --- | --- |
| private void show\_Click(object sender, EventArgs e)  {  string kelas = "";  string jadwal = ""; | |
| **Penjelasan:** | Sebuah event catcher untuk tombol tampilkan yang memiliki attribut string kelas dan jadwal |

|  |  |
| --- | --- |
| if (jadwal1.Checked)  {  jadwal = "Senin & Rabu, 14.00-16.00";  }  else if (jadwal2.Checked)  {  jadwal = "Selasa & Kamis, 14.00-16.00";  }  else if (jadwal3.Checked)  {  jadwal = "Sabtu & Minggu, 9.00-11.00";  }  else if (jadwal4.Checked)  {  jadwal = "Minggu, 13.00-17.00";  } | |
| **Penjelasan:** | Control flow untuk radiobutton jadwal yang meng-overwrite isi value jadwal |

|  |  |
| --- | --- |
| if (biola.Checked)  {  kelas += "Biola ";  }  if (drum.Checked)  {  kelas += "Drum ";  }  if (gitar.Checked)  {  kelas += "Gitar ";  }  if (kom.Checked)  {  kelas += "Komposer ";  }  if (sax.Checked)  {  kelas += "Saxophone ";  }  if (vokal.Checked)  {  kelas += "Vokal ";  }  if (kon.Checked)  {  kelas += "Konduktor ";  }  if (piano.Checked)  {  kelas += "Piano ";  }  if(kelas == "")  {  MessageBox.Show("Harus Memilih salah satu dari pilihan kelas ", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);  } | |
| **Penjelasan:** | Sebuah control flow checkbox kelas yang menambahkan value ke dalam attribut kelas |

|  |  |
| --- | --- |
| MessageBox.Show("Nama : " + namabox.Text +  "\nJenis Kelamin : " + jk.Text  "\nTanggal Lahir : "+ dateTimePicker1.Text +  "\nPilihan Kelas : " + kelas +  "\nPilihan Jadwal : " + jadwal +  "Informasi Pendaftaran", MessageBoxButtons.OK, MessageBoxIcon.Information);  } | |
| **Penjelasan:** | Menampilkan MessageBox yang berisi Nama, Tanggal Lahir, Jenis Kelamin, Pilihan Kelas, Pilihan Jadwal, dan Informasi Pendaftaran |

|  |  |
| --- | --- |
| private void finish\_Click(object sender, EventArgs e)  {  this.Close();  } | |
| **Penjelasan:** | Event catcher untuk tombol finish yang akan menutup window |

**2.5 JAWABAN PRAKTIKUM**

1. Maxlength berfungsi untuk membatasi jumlah maximal karakter yang dapat diketikan pada textBox.

2. Berfungsi untuk mengisi yang akan ditampilkan dalam combo box berbentuk opsi saat dipilih

3. Value adalah Data yang diseleksi oleh pengguna. Format berfungsi Menetapkan format dari tanggal dan/ atau waktu yang digunakan untuk opsi pengguna. Custom format Menetapkan format string kustom untuk opsi pengguna.

4. checked di radio button dan checkbox sendiri berfungsi agar box akan terisi tanda centang.

5. Radio button adalah elemen graphical user interface ( GUI ) yang memungkinkan pengguna untuk memilih single item dari daftar opsi yang ditampilkan. Elemen radio button akan ditampilkan dalam kelompok yang terdiri dari beberapa opsi atau setidaknya dua opsi. Radio button ini berbentuk lingkaran kecil. Ketika lingkaran tersebut dipilih oleh user maka tampilan radio button akan terlihat seperti lingkaran dengan titik di dalamnya. Radio button bersifat mutually exclusive. Jadi, ketika ketika pengguna memilih satu opsi, maka opsi lain yang terpilih dalam group yang sama akan dibatalkan. Dengan kata lain, radio button hanya memungkinkan pengguna untuk memilih satu item saja dari opsi yang disediakan dalam group radio button. Berbeda dengan combo box memilih namun tidak mencentang seperti radio button atau check box.

**BAB III**

**PENUTUP**

**3.1 KESIMPULAN**

Secara umum, Data Field digunakan untuk mendefinisikan variabel di kelas dengan aksesibilitas sebagai pribadi atau dilindungi. Data Field yang perlu diekspos di mana saja di luar kelas dapat dienkapsulasi sebagai metode publik, properti, atau pengindeks. Data Field juga dikenal sebagai variabel tingkat kelas atau variabel anggota.

Dalam pemrograman berorientasi objek bekerja dengan objek. Objek adalah blok bangunan dasar dari sebuah program. Objek terdiri dari data dan metode. Metode mengubah keadaan objek yang dibuat. Metode adalah bagian dinamis dari objek; data adalah bagian statis. Metode adalah blok kode yang berisi serangkaian pernyataan. Metode harus dideklarasikan di dalam kelas atau struktur. Ini adalah praktik pemrograman yang baik bahwa metode hanya melakukan satu tugas tertentu. Metode membawa modularitas ke program. Method merupakan sebutan untuk behavioral/function. Method ini bisa diidentifikasi dengan adanya data type di depan nama method-nya dan kata kunci return di dalam method-nya.

Dalam pemrograman C#, constructor merupakan sebuah method khusus yang akan dipanggil pada saat sebuah objek (instance) dari sebuah class dibuat. Constructor mempunyai nama atau identifier yang sama persis dengan identifier class -nya. Constructor bermanfaat untuk menginisialisasi beberapa field yang membutuhkan nilai default. Dengan kata lain, constructor diperlukan untuk “membangun” (construct) objek dari sebuah class.

**3.2 SARAN**

Saran yang bisa penulis berikan perlu adanya pembelajaran lebih lanjut akan pembelajaran mengenai Pemrograman II sebagai salah satu cara memaksimalakan pembelajaran dalam mata kuliah.

**DAFTAR PUSTAKA**

Artikel Doavers Development Team, 2016

<https://www.doavers.com/blog/belajar-pemrograman-berbasis-objek-oop-di-c-bagian-2>

Mudjihartono, Rekayasa Pembalikan Kode Berorientasi Objek ke Desain Kelas Dengan Struktur Data Graf, Jurnal Buana Informatika, 2010

<https://ojs.uajy.ac.id/index.php/jbi/article/view/301/381>

Artikel ajengtriansari, Class,Attribute dan Methode, 2012

<https://ajengtriansari.wordpress.com/2012/10/11/71/>

Artikel Dasar Komputer Buat Pemula, 2016

<https://pccontrol.wordpress.com/dasar-c/>

Artikel Class Members, tecnopedia, 2011

<https://www.techopedia.com/definition/25589/class-members-c-sharp>

Artikel Class dan Objek pada c#, 2013

[https://icodeformoney.com/tuts/Class-dan-objek-pada-CSharp#](https://icodeformoney.com/tuts/Class-dan-objek-pada-CSharp)

Artikel Belajar c++, 2019

<https://www.belajarcpp.com/tutorial/cpp/access-modifier/>