

MEMBUAT PROGRAM CONSOLE SEDERHANA DENGAN BAHASA PEMROGRAMAN C#

PRAKTIKUM 5

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 ISI.....	1
TABEL AKTIVITAS.....	2
BAB I PENDAHULUAN.....	3
1.1 Mengenal Istilah <i>Assembly</i> di Lingkungan .NET.....	3
1.2 Mengakses Anggota Class	3
1.3 Properti	4
1.4 Mengekapsulasi Beberapa Fungsionalitas dengan <i>Utility Class</i>	4
BAB II PEMBAHASAN.....	5
2.1 Percobaan 1	5
2.2 Percobaan 2	8
2.1 Percobaan 3	10
2.1 Percobaan 4	12
2.1 Jawaban Praktikum	15
BAB III PENUTUP	15
3.1 Kesimpulan.....	15
3.2 Saran.....	16
DAFTAR PUSTAKA.....	17

TABEL AKTIVITAS

TANGGAL	CATATAN
03 Desember 2021	Mempelajari Modul dan Mengerjakan Laporan Bab 1
04 Desember 2021	Mengerjakan Laporan Bab 1 dan Bab 3
05 Desember 2021	Mengerjakan Percobaan 1 dan 2
06 Desember 2021	Mengerjakan Percobaan 3 dan 4
07 Desember 2021	Mengisi laporan bab 2 dan selesai
08 Desember 2021	Revisi praktikum dan mengumpulkan

BAB I

PENDAHULUAN

1.1 KELURGA OBJEK

Objek pada pemrograman berorientasi objek merupakan instans dari kelas (class). Dengan kata lain sebelum ada objek maka harus dibuat kelas terlebih dahulu (kelas adalah template/pola dari objek). Bahasa pemrograman berorientasi objek biasanya mengadopsi multi paradigma (lebih dari satu paradigma pemrograman). Bahasa pemrograman C# tidak terkecuali. Bahasa pemrograman C# adalah bahasa pemrograman modern dibuat oleh Microsoft sebagai salah satu bahasa yang dirancang untuk menggunakan framework .NET. Bahasa pemrograman C# diajukan dan disetujui oleh dua badan standar internasional Ecma (ECMA-334) dan ISO (ISO/IEC 23270:2006). Bahasa ini dirancang sebagai bahasa yang dapat digunakan untuk menyelesaikan berbagai macam masalah, sederhana, dan modern. Paradigma pemrograman berorientasi objek memiliki empat pilar pendukung. Empat pilar tersebut adalah abstraksi (abstraction), enkapsulasi (encapsulation), polimorfisme (polymorphism), dan pewarisan (inheritance).

1.2 GENERALISASI DAN SPESIALISASI

Class dan instannya (objek) tidak berdiri sendiri, seperti objek manusia yang merupakan makhluk sosial tidak hanya berdiri sendiri, objek akan saling berelasi dan berinteraksi, demikian pula pada OOP. Salah satu hubungan antar objek yang paling penting yang terjadi di dunia nyata adalah spesialisasi yang bisa didefinisikan sebagai is-a relationship, misal anjing adalah mamalia, jika anjing termasuk anggota mamalia maka dia memiliki ciri-ciri special khas yaitu menyusui, punya bulu. Selain sifat-sifat pada mamalia anjing juga memiliki ciri yang spesifik khas anjing yang tidak dipunyai anggota mamalia lainnya seperti kucing. Relasi spesialisasi dan generalisasi terjadi pada contoh diatas, anjing dan kucing secara general merupakan anggota mamalia tetapi juga terspesialisasi secara spesifik sebagai anjing dan kucing yang mempunyai ciri khas yang berbeda. Relasi tersebut dapat digambarkan secara hirarki.

1.3 PEWARISAN DALAM

Inheritance merupakan penurunan sifat atau pewarisan dimana dimisalkan suatu induk tentunya akan memiliki anak yang mewarisi sebagian atau seluruh sifat induknya. Dalam hal ini suatu class induk/parent disebut sebagai super class dan anak/child disebut sebagai sub class.

1.4 MENGIMPLEMENTASIKAN KONSEP PEWARISAN DI C#

Konsep inheritance (pewarisan) membuat kita dapat mendefinisikan sebuah class pada class lainnya. Dan ini memudahkan kita untuk membuat dan memelihara sebuah aplikasi. Konsep ini juga memberikan kesempatan penggunaan ulang code dan waktu implementasi lebih cepat.

Saat membuat class, daripada menulis lengkap data dan fungsi baru, programmer bisa mendesain sebuah kelas yang dapat mewariskan (inherit) bagian dari class yang sudah ada. Class yang sudah ada ini disebut base class (dasar) dan class yang baru disebut derived class (turunan).

Sebuah kelas dapat diturunkan dari lebih dari satu class atau interface, yang berarti bahwa ia dapat mewarisi data dan function dari beberapa base class atau interface.

1.5 PEWARISAN PADA CONSTRUCTOR

Constructor yang terdapat pada kelas induk dapat dipanggil dari kelas turunan menggunakan kata kunci `super`, Bentuk umum pemanggilannya: `super (daftar-parameter)` Daftar-parameter = daftar parameter yang didefinisikan constructor kelas induk Dalam constructor kelas turunan, `super()` harus ditempatkan pada bagian awal (baris pertama).

1.6 ANGGOTA CLASS DENGAN AKSES PRIVATE TIDAK BISA DIWARISKAN

Variable dan method yang diberikan hak akses `private` hanya bisa diakses oleh class itu sendiri, data-data tersebut tidak bisa diwariskan pada subclass atau class lainnya. Agar class lain bisa mengakses variable atau method tersebut maka perlu dibuatkan method yang mempunyai hak akses `public` (package yang berbeda) atau `protected` (pada package yang sama).

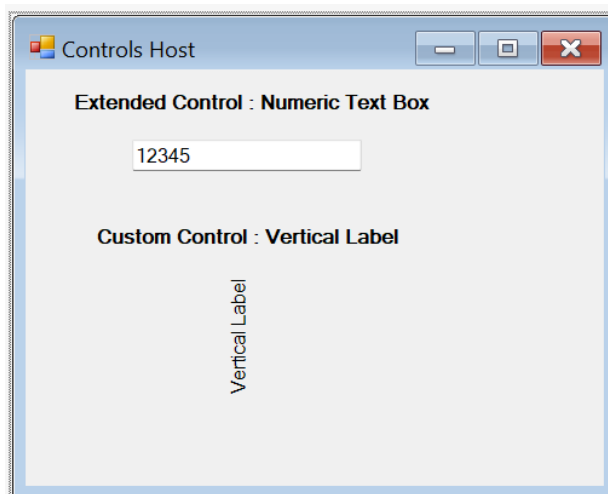
BAB II

PEMBAHASAN

2.1 PERCOBAAN 1

Repository Github	https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_5/tree/main/P5_1_1204011
Last Commit	08 Desember 2021

Output :



<code>using System.Threading.Tasks;;</code>	
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

<code>namespace P5_1_1204011</code>	
Penjelasan	Namespace pada C# berfungsi untuk mengatur banyak kelas sehingga mudah menangani aplikasi

<code>public partial class NumericTextBox: TextBox</code>	
Penjelasan	Kata kunci partial menunjukkan bahwa bagian lain dari kelas, struct, atau antarmuka dapat didefinisikan dalam namespace. Semua bagian harus menggunakan kata kunci parsial. Semua bagian harus tersedia pada waktu kompilasi untuk membentuk tipe akhir. Semua bagian harus memiliki aksesibilitas yang sama, seperti publik, privat, dan sebagainya. Dan digunakan juga untuk membuat objek baru yaitu text box baru.

```

public override string Text
{
    get
    {
        return base.Text;
    }
    set
    {
        try
        {
            int.Parse(value);
            base.Text = value;
            return;
        }
        catch
        { }
        if (value == null)
        {
            base.Text = value;
            return;
        }
    }
}

```

Penjelasan

Digunakan untuk membuat jika nanti pada design time hanya dapat menerima nilai angka.

```

e);
public event InvalidUserEntryEvent InvalidUserEntry;

protected override void OnKeyPress(System.Windows.Forms.KeyPressEventArgs e)
{
    int asciiInteger = Convert.ToInt32(e.KeyChar);
    if (asciiInteger >= 47 && asciiInteger <= 57)
    {
        e.Handled = false;
        return;
    }
    if (asciiInteger == 8)
    {
        e.Handled = false;
        return;
    }
    e.Handled = true;
    if (InvalidUserEntry != null)
        InvalidUserEntry(this, e);
}

```

Penjelasan

Digunakan untuk agar pada saat runtime dilakukan perubahan hanya bisa menerima nilai angka.

```
public partial class VerticalLabel : Control
```

Penjelasan: Digunakan agar VerticalLabel diturunkan dari kelas Control bukan dari UserControl

```
private string labelText;
[Category("VerticalLabel"), Description("Text is displayed in container")]
public override string Text
{
    get
    {
        return labelText;
    }
    set
    {
        labelText = value;
        Invalidate();
    }
}
```

Penjelasan: Digunakan untuk membuat field bertipe string labelText. Override property Text. Serta attribute untuk design time support, untuk memberikan penjelasan tentang property Text di Windows Properties.

```
protected override void OnPaint(System.Windows.Forms.PaintEventArgs e)
{
    float sngControlWidth;
    float sngControlHeight;
    float sngTransformX;
    float sngTransformY;

    Color labelColor = new Color();
    Pen labelBorderPen = new Pen(labelColor, 0);
    SolidBrush labelBackColorBrush = new SolidBrush(labelColor);
    SolidBrush labelForeColorBrush = new SolidBrush(base.ForeColor);

    if (this.DesignMode)
        base.ResizeRedraw = true;

    base.OnPaint(e);

    sngControlWidth = this.Size.Width;
    sngControlHeight = this.Size.Height;

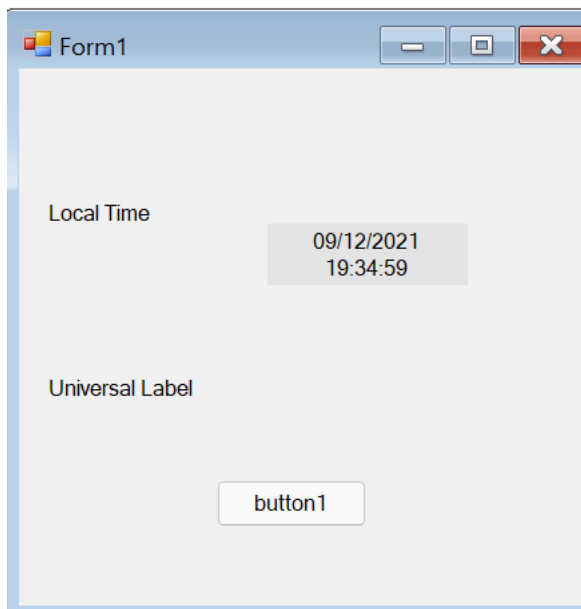
    e.Graphics.DrawRectangle(labelBorderPen, 0, 0, sngControlWidth,
sngControlHeight);
    e.Graphics.FillRectangle(labelBackColorBrush, 0, 0, sngControlWidth,
sngControlHeight);
    sngTransformX = 0;
    sngTransformY = sngControlHeight;
    e.Graphics.TranslateTransform(sngTransformX, sngTransformY);
    e.Graphics.RotateTransform(270);
    e.Graphics.DrawString(labelText, Font, labelForeColorBrush, 0, 0);
}
```

Penjelasan: Override event OnPaint untuk menggambar VerticalLabel

2.2 PERCOBAAN 2

Repository Github	https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_5/tree/main/P5_2_1204011
Last Commit	08 Desember 2021

Output :



```
public DigitalClock()
{
    InitializeComponent();
}
public bool Timer1_Enabled
{
    get
    { return Timer1.Enabled; }
    set
    {
        Timer1.Enabled = value;
    }
}
```

Penjelasan:	Digunakan untuk membuat property public bernama Timer1_Enabled yang mengekspos property boolean Enabled dari Timer1
--------------------	---------------------------------------------------------------------------------------------------------------------

```
public Color LocalTimeLabel_BackColor
{
    get
    {
        return LocalTimeLabel.BackColor;
    }
    set
    {
        LocalTimeLabel.BackColor = value;
    }
}
```

Penjelasan:	Digunakan untuk membuat property public bernama
--------------------	-------------------------------------------------

	LocalTimeLabel_BackColor yang mengekspos property BackColor dari LocalTimeLabel.
--	----------------------------------------------------------------------------------

```
public event System.EventHandler RaiseTimer1_Tick;
private void Timer1_Tick(object sender, EventArgs e)
{
    LocalTimeLabel.Text = System.DateTime.Now.ToString();
    if (RaiseTimer1_Tick != null)
    {
        RaiseTimer1_Tick(sender, e);
    }
}
```

Penjelasan:	Digunakan untuk mendeklarasikan event handler RaiseTimer1_Tick, dan untuk mengupdate property Text dari LocalTimeLabel agar menampilkan waktu saat ini.
--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------

```
private void digitalClock1_RaiseTimer1_Tick(object sender, EventArgs e)
{
    UniversalTimeLabel.Text = DateTime.Now.ToUniversalTime().ToString();
}
```

Penjelasan:	Digunakan untuk menampilkan universal time saat ini.
--------------------	------------------------------------------------------

```
private void button1_Click(object sender, EventArgs e)
{
    digitalClock1.Timer1_Enabled = !(digitalClock1.Timer1_Enabled);
}
```

Penjelasan:	Digunakan untuk memberhentikan dan memulai kembali universal time.
--------------------	--------------------------------------------------------------------

2.3 LATIHAN

Repository Github	https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_5/tree/main/P5_3_1204011
Last Commit	09 Desember 2021

Output :

Form1

NIM: 1204011 Program Studi: TEKNIK INFORMATIKA

Nama: Wildan Azril Tahun Akademik: 2021/2022

Jenis Kelamin: ☒ Laki-Laki ☐ Perempuan Semester: 3

Alamat: Tasikmalaya

Pilih Mata Kuliah

Kurikulum Pilihan

☐ Kurikulum 2006

☒ Kurikulum 2010

☐ Kurikulum 2014

Pemrograman

☐ Pemrograman 6

☒ Pemrograman 7

☐ Pengantar Logistik

☐ Jaringan Komputer

☐ Sistem Operasi

☐ Manajemen Rantai Pasok

☐ Pemrograman 4

☐ Pemrograman 5

Simpan Batal

Modal Box:

NIM : 1204011
 Nama : Wildan Azril
 Jenis Kelamin : Laki-Laki
 Alamat : Tasikmalaya
 Program Studi : Perempuan
 Tahun Akademik : 2021/2022
 Semester : 3
 Kurikulum : Kurikulum 2010
 Kelas : Pemrograman 1Pemrograman 7

OK

```
public partial class NumericTB: TextBox
{
    public NumericTB()
    {
        InitializeComponent();
    }

    public override string Text
    {
        get
        {
            return base.Text;
        }
        set
        {
            try
            {
                int.Parse(value);
                base.Text = value;
                return;
            }
            catch { }
            if (value == null)
            {
                base.Text = value;
                return;
            }
        }
    }
}
```

```

public delegate void InvalidUserEntryEvent(object sender, KeyPressEventArgs e);
public event InvalidUserEntryEvent InvalidUserEntry;

protected override void OnKeyPress(System.Windows.Forms.KeyPressEventArgs e)
{
    int asciiInteger = Convert.ToInt32(e.KeyChar);
    if (asciiInteger >= 47 && asciiInteger <= 57)
    {
        e.Handled = false;
        return;
    }
    if (asciiInteger == 0)
    {
        e.Handled = false;
    }

    e.Handled = true;

    if (InvalidUserEntry != null)
    {
        InvalidUserEntry(this, e);
    }
}
}

```

Penjelasan: Digunakan untuk membuat sebuah textbox yang hanya menerima ASCII yang berada diantara 47 sampai 57.

```

private void thak_TextChanged(object sender, EventArgs e)
{
    Regex regex = new Regex("^.*/*.*$", RegexOptions.IgnoreCase);
    if (regex.IsMatch(thak.Text))
    {
        epWarn.Clear();
    }
    else
    {
        epWarn.SetError(thak, "Regex Tidak Sesuai");
    }
}

```

Penjelasan: Digunakan untuk mengecek perubahan yang terjadi pada box tahun akademik menggunakan regex, jika sesuai akan menghapus errorProvider, jika tidak akan memunculkan epWarn dengan pesan Regex Tidak sesuai

```

private void button3_Click(object sender, EventArgs e)
{
    if ((lk.Checked == true || pr.Checked == true) && NIM.Text != "" &&
nama.Text != "" && alamat.Text != "" && progcb.Text != "" && thak.Text != "" &&
semes.Text != "")
    {
        if(k2006.Checked == true)
        {
            matkul.Enabled = false;
            matkul.Enabled = true;
            checkBox12.Enabled = false;
            checkBox9.Enabled = false;
        }
        else if (k2010.Checked == true)
        {
            matkul.Enabled = false;

```

<pre> matkul.Enabled = true; mtk.Enabled = false; checkBox9.Enabled = false; } else if (k2014.Checked == true) { matkul.Enabled = false; matkul.Enabled = true; mtk.Enabled = false; checkBox12.Enabled = false; } } } </pre>	
Penjelasan:	Digunakan untuk membuat mata kuliah pilihan enabled dan tidak sesuai kurikulum. Dan size digunakan untuk saat tombol mata kuliah di tekan maka size form akan berubah.

<pre> private void button1_Click(object sender, EventArgs e) { string kelas = " "; if (mtk.Checked) { kelas += mtk.Text; } if (pem1.Checked) { kelas += pem1.Text; } if (checkBox10.Checked) { kelas += checkBox10.Text; } if (checkBox11.Checked) { kelas += checkBox11.Text; } if (checkBox12.Checked) { kelas += checkBox12.Text; } if (checkBox3.Checked) { kelas += checkBox3.Text; } if (checkBox4.Checked) { kelas += checkBox4.Text; } if (checkBox5.Checked) { kelas += checkBox5.Text; } if (checkBox6.Checked) { kelas += checkBox6.Text; } if (checkBox7.Checked) { kelas += checkBox7.Text; } if (checkBox8.Checked) { kelas += checkBox8.Text; } if (checkBox9.Checked) { kelas += checkBox9.Text; } string jk = lk.Checked == true ? lk.Text : pr.Text; string kuri = ""; if (k2006.Checked){ kuri = k2006.Text; } else if (k2010.Checked) { kuri = k2010.Text; } else if (k2014.Checked) { kuri = k2014.Text; } string message = String.Format("NIM : {0}\nNama : {1}\nJenis Kelamin : {2}\nAlamat : {3}\nProgram Studi : {4}\nTahun Akademik : {5}\nSemester : {6}\nKurikulum : {7}\nKelas : {8}\n", NIM.Text, nama.Text, jk, alamat.Text, pr.Text, thak.Text, semes.Text, kuri, kelas); MessageBox.Show(message); } </pre>	
Penjelasan:	Sebuah event handler click tombol button1 dengan mengecek isi dari textbox, radiobutton, dan checkbox kemudian ditampilkan menggunakan messagebox

2.4 JAWABAN PRAKTIKUM

Percobaan 1 :

- Klik numericTextBox1 pada form tsb, lalu pada Windows Properties ubahlah property Text, ketikkan “Tes Huruf”. Perhatikan hasilnya. Apa yang terjadi? Mengapa demikian?

Jawaban : Karena tidak menampilkan huruf yang di ketikan, karena numeric text box hanya menerima character ASCII yang berada diantara 47 sampai 57

- Coba ketikkan “12345”. Perhatikan hasilnya. Apa yang terjadi? Mengapa hal tersebut terjadi?

Jawaban : Karena menampilkan angka yang di ketikan, karena numeric text box menerima character ascii yang berada diantara 47 sampai 57

BAB III

PENUTUP

3.1 KESIMPULAN

Praktikum 4 ini mempelajari toolbox – toolbox baru, mulai dari Numeric Text box, VerticalBox, cara mengatur waktu dan tanggal yang sesuai dan juga membuat form yang mempunyai banyak jenis jenis toolbox yang dipakai.

3.2 SARAN

Saran yang bisa penulis berikan perlu adanya pembelajaran lebih lanjut akan pembelajaran mengenai Pemrograman II sebagai salah satu cara memaksimalkan 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#>

Artikel Belajar c++, 2019

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