

# **MEMBUAT PROGRAM WINDOWS DENGAN BAHASA PEMROGRAMAN C# DAN DATABASE**

## **PRAKTIKUM 6**

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

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>TABEL AKTIVITAS.....</b>	<b>2</b>
<b>BAB I PENDAHULUAN.....</b>	<b>3</b>
1.1 Abstract Class.....	3
1.2 Menyediakan Method yang bisa ditimpa (Overriding) .....	3
1.3 Menggantikan Method pada Base Class .....	4
<b>BAB II PEMBAHASAN.....</b>	<b>5</b>
2.1 Percobaan 1 .....	5
2.2 Percobaan 2 .....	8
2.3 Latihan.....	10
2.4 Jawaban Praktikum .....	15
<b>BAB III PENUTUP.....</b>	<b>15</b>
3.1 Kesimpulan.....	15
3.2 Saran.....	16
<b>DAFTAR PUSTAKA.....</b>	<b>17</b>

### **TABEL AKTIVITAS**

<b>TANGGAL</b>	<b>CATATAN</b>
9 Desember 2021	Mengerjakan codingan praktikum
10 Desember 2021	Mengerjakan laporan dan selesai

# **BAB I**

## **PENDAHULUAN**

### **1.1 ABSTRACT CLASS**

Tujuan di buatnya class adalah hanya untuk mengumpulkan karakteristik dan perilaku yang bersifat umum dari class turunannya ke dalam satu wadah agar nantinya karakteristik dan perilaku tersebut bisa diwariskan. Dengan alasan tersebut lebih baik kita mencegah instansiasi objek dari class dengan cara mendeklarasikan class tersebut dengan kata kunci `abstract`. Sehingga, deklarasi class menjadi `abstract class { }`.

Apabila mencoba untuk membuat sebuah objek langsung dari class yang dideklarasikan dengan kata kunci `abstract`, compiler akan memberi Anda peringatan bahwa membuat sebuah objek (instance) dari `abstract class` itu tidak diperkenankan.

### **1.2 MENYEDIAKAN METHOD YANG BISA DITIMPA ( OVERRIDING)**

Di dalam semua bahasa pemrograman yang berbasis objek, teknik overriding adalah fitur yang memungkinkan kita untuk mengimplementasikan “ulang” fungsi/method pada sebuah `child class` atau kelas turunan yang sebenarnya fungsi tersebut telah didefinisikan di dalam `parent class` atau kelas induk.

Overriding sendiri memberikan keuntungan kepada kita karena kita bisa menduplikat kelas lain –sehingga kode program lebih singkat, dan kita juga bisa “membedakan” fungsi-fungsi tertentu pada kelas turunan yang sudah didefinisikan pada kelas Induk.

Hal ini memberikan kita keleluasaan untuk mendefinisikan berbagai kelas sesuai fungsinya masing-masing tanpa harus menulis kode yang sama berkali-kali. Oleh karena itu, overriding masih dikategorikan sebuah bagian dari pewarisan

### **1.3 MENGGANTIKAN METHOD PADA BASE CLASS**

Metode yang bersembunyi di `C#` mirip dengan fitur `override` fungsi di `C++`. Fungsi dari kelas dasarnya tersedia untuk kelas turunan. Jika kelas turunan tidak senang, salah satu fungsi tersedia untuknya dari kelas dasar dapat mendefinisikan versinya sendiri dari fungsi yang sama dengan tanda tangan fungsi yang sama, hanya berbeda di menerapkannya. Ini baru definisi menyembunyikan itu basis kelas definisi. Take itu mengikutiprogram misalnya

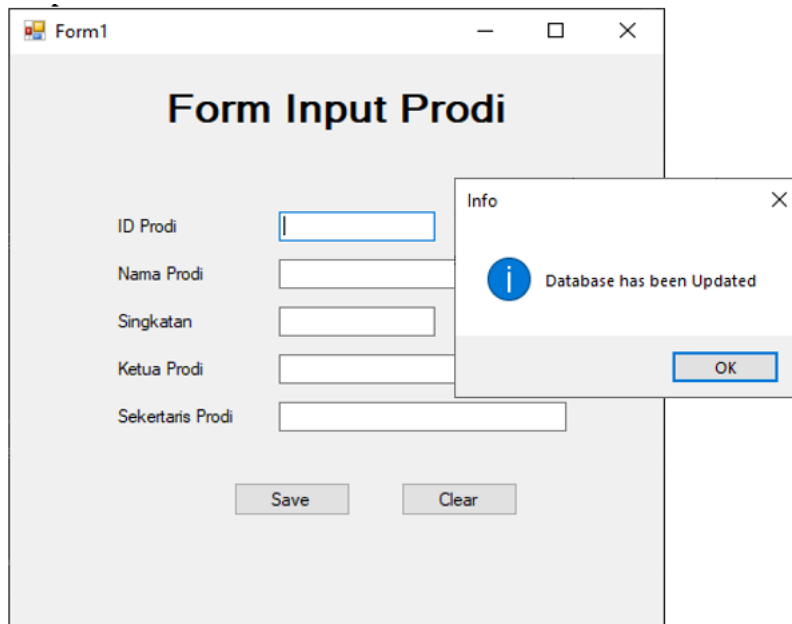
## BAB II

### PEMBAHASAN

#### 2.1 PERCOBAAN 1

<b>Repository Github</b>	<a href="https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_1_1204011">https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_1_1204011</a>
<b>Last Commit</b>	<b>10 Desember 2021</b>

**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 P6_1_1204011
```

#### **Penjelasan**

Namespace pada C# berfungsi untuk mengatur banyak kelas sehingga mudah menangani aplikasi

```
private string url = "server=localhost;userid=root;password=;database=p6_1204011";
```

#### **Penjelasan:**

Digunakan untuk mengatur database yang akan dipakai

<pre> private void UpdateDB (string cmd) {     try     {         MySqlConnection conn = new MySqlConnection(url);         conn.Open();          MySqlCommand command = new MySqlCommand();         command.Connection = conn;         command.CommandText = cmd;         command.ExecuteNonQuery();         MessageBox.Show("Database has been Updated", "Info",         MessageBoxButtons.OK, MessageBoxIcon.Information);     }     catch(Exception ex)     {         MessageBox.Show(ex.ToString(), "Error!", MessageBoxButtons.OK,         MessageBoxIcon.Error);     } } </pre>	
<b>Penjelasan:</b>	Digunakan untuk menghubungkan ke dalam database, serta akan muncul message box dengan tulisan Database has been Updated jika data yang di inputkan berhasil, dan akan muncul message box dengan tulisan Error jika data yang di inputkan gagal di inputkan

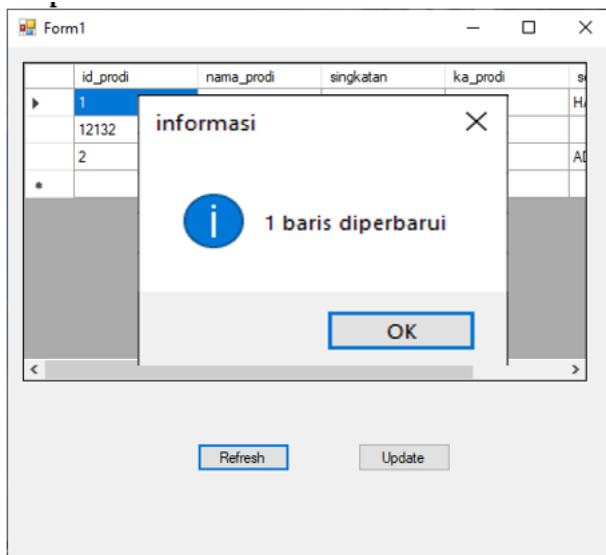
<pre> private void button1_Click(object sender, EventArgs e) {     string cmdsave = "insert into MSPRODI values('"+ txtIdProdi.Text + "','"+ + txtNamaProdi.Text+"', '"+ txtSingkatan.Text + "','"+txtKaProdi.Text+"', '"+txtSekProdi.Text+"')";     UpdateDB(cmdsave); } </pre>	
<b>Penjelasan:</b>	Event handler click button1 untuk menambah data ke dalam database ketika menekan button save

<pre> private void button2_Click(object sender, EventArgs e) {     txtIdProdi.Text = "";     txtKaProdi.Text = "";     txtNamaProdi.Text = "";     txtSekProdi.Text = "";     txtSingkatan.Text = ""; } </pre>	
<b>Penjelasan:</b>	Event handler btnclear untuk mengosongkan kembali data yang telah di ketik pada form

## 2.2 PERCOBAAN 2

<b>Repository Github</b>	<a href="https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_2_1204011">https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_2_1204011</a>
<b>Last Commit</b>	<b>10 Desember 2021</b>

**Output :**



```
private void button1_Click(object sender, EventArgs e)
{
    RefreshDataset();
}
```

**Penjelasan:** Digunakan untuk ketika di tekan button refresh, dan memanggil class RefreshDataset

```
private string url =
"server=localhost;userid=root;password=;database=p6_1204011";
private DataSet Getdatasets()
{
    DataSet mydataset = new DataSet();
    try
    {
        MySqlConnection conn = new MySqlConnection(url);
        conn.Open();

        MySqlCommand command = new MySqlCommand();
        command.Connection = conn;
        command.CommandText = "select * from msprodi";
        command.CommandType = CommandType.Text;

        MySqlDataAdapter dataadap = new MySqlDataAdapter();
        dataadap.SelectCommand = command;
        dataadap.TableMappings.Add("Table", "Prodi");

        dataadap.Fill(mydataset);
    }
}
```

```

        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString(), "Error!", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }

        return mydataset;
    }

```

**Penjelasan:** Sebuah method dataset yang mengambil data dari database kemudian menggunakan mysqldataadapter untuk mengisi dataset mydataset

```

private void RefreshData()
{
    dsProdi = Getdatasets();
    dgProdi.DataSource = dsProdi.Tables["Prodi"];
}

```

**Penjelasan:** Membuat method RefreshData yang digunakan untuk menampilkan data yang ada di dalam database dengan memanggil Getdataset

```

private void button1_Click(object sender, EventArgs e)
{
    RefreshData();
}

```

**Penjelasan:** Event handler click button1 yang akan memanggil RefreshData

```

private void button2_Click(object sender, EventArgs e)
{
    string url =
"server=localhost;userid=root;password=;database=p6_1204011";
    string myConnectionString = "integrated
security=true;datasource=localhost;initial catalog=p6_1204011";
    MySqlConnection conn = new MySqlConnection(url);

    conn.Open();

    MySqlDataAdapter myAdapter = new MySqlDataAdapter("select * from
msprodi", conn);
    MySqlCommandBuilder mycmdBuilder = new MySqlCommandBuilder(myAdapter);

    myAdapter.InsertCommand = mycmdBuilder.GetInsertCommand();
    myAdapter.DeleteCommand = mycmdBuilder.GetDeleteCommand();
    myAdapter.UpdateCommand = mycmdBuilder.GetUpdateCommand();

    MySqlTransaction myTrans;
    myTrans = conn.BeginTransaction();
    myAdapter.InsertCommand.Transaction = myTrans;
    myAdapter.DeleteCommand.Transaction = myTrans;
    myAdapter.UpdateCommand.Transaction = myTrans;

    try
    {
        int rowsupdate = myAdapter.Update(dsProdi, "Prodi");
        myTrans.Commit();
        MessageBox.Show(rowsupdate.ToString() + " baris diperbarui",
        "informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```



```

        RefreshData();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unable to update: " + ex.Message);
        myTrans.Rollback();
    }

    MessageBox.Show(myAdapter.InsertCommand.CommandText);
    MessageBox.Show(myAdapter.UpdateCommand.CommandText);
    MessageBox.Show(myAdapter.DeleteCommand.CommandText);
}

```

<b>Penjelasan:</b>	Event handler lick button2 yang akan meng update data yang ada dalam database dengan mengambil dari datagrid yang ditampilkan
--------------------	---

## 2.3 LATIHAN

Repository Github	<a href="https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_3_1204011">https://github.com/azrilarva21/TUGAS_PEMROGRAMAN2_6/tree/main/P6_3_1204011</a>
Last Commit	10 Desember 2021

Output :

Form1

### FORM INPUT DATA MAHASISWA

NPM:

Nama:

Tanggal Lahir:

Jenis Kelamin: ☐ Laki-Laki ☐ Perempuan

Alamat:

No. Telp:

Program Studi:

Save Cancel

Info: Data has been Updated OK

```
private string url =  
"server=localhost;userid=root;password=;database=p6_1204011";  
  
DataTable dataprodi = new DataTable();  
private void filldataset()  
{  
    using (MySqlConnection sqlConnection = new MySqlConnection(url))  
    {  
        MySqlCommand sqlCommand = new MySqlCommand("SELECT id_prodi, singkatan  
FROM msprodi", sqlConnection);  
        sqlConnection.Open();  
        MySqlDataAdapter sqlReader = new MySqlDataAdapter(sqlCmd);  
        sqlReader.Fill(dataprodi);  
  
        //while (sqlReader.Read())  
        //{  
        //    comboBox1.Items.Add(sqlReader["singkatan"].ToString());  
        //    comboBox1.Items.Add(new List)  
        //}  
    }  
}
```

```

        comboBox1.DataSource = dataprodi;
        comboBox1.DisplayMember = "singkatan";
        comboBox1.ValueMember = "id_prodi";
    }

```

**Penjelasan:** Digunakan untuk menghubungkan ke database

```

private void InsertDB(string cmd)
{
    try
    {
        MySqlConnection conn = new MySqlConnection(url);
        conn.Open();

        MySqlCommand command = new MySqlCommand();
        command.Connection = conn;
        command.CommandText = cmd;
        command.ExecuteNonQuery();
        MessageBox.Show("Data has been Updated", "Info",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString(), "Error!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

```

**Penjelasan:** Digunakan untuk perintah menginputkan data ke dalam database, serta akan muncul message box dengan tulisan data berhasil di update jika data yang di inputkan berhasil, dan akan muncul message box dengan tulisan Error jika data yang di inputkan gagal di inputkan

```

private void button1_Click(object sender, EventArgs e)
{
    string jk = "";
    if (pr.Checked) { jk = pr.Text; } else if (lk.Checked) { jk = lk.Text; }
    string sDate = tanggal.SelectionStart.ToString("yyyy\\MM\\dd");
    string alamat = textBox1.Text;
    string cmdsave = "insert into MSMHS values('" + tbnpm.Text + "','" +
    tbnama.Text + "','" + sDate + "','" + jk + "','" + alamat + "','" + tbtelp.Text +
    "','" + comboBox1.SelectedValue + "')";
    InsertDB(cmdsave);
}

```

**Penjelasan:** Event Handler Click button1 yang akan menginput data ke dalam database yang mengambil dari textbox

```

private void button2_Click(object sender, EventArgs e)
{
    tbnama.Clear();
    tbnpm.Clear();
    tbtelp.Clear();
    alamat.Text = "";
}

```

```

        comboBox1.Items.Insert(0, "-Please Select-");
        pr.Checked = false;
        lk.Checked = false;
    }

```

**Penjelasan:** Event handler click button2 yang akan menghapus isi dari textbox

```

private void seterror(Control control)
{
    epWarn.SetError(control, "Harus diisi");
}

private void remerror()
{
    epWarn.Clear();
}

private void tbnpm_Leave(object sender, EventArgs e)
{
    if(tbnpm.Text == "")
    {
        seterror(tbnpm);
    }
    else
    {
        remerror();
    }
}

private void tbnama_Leave(object sender, EventArgs e)
{
    if (tbnama.Text == "")
    {
        seterror(tbnama);
    }
    else
    {
        remerror();
    }
}

private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        seterror(textBox1);
    }
    else
    {
        remerror();
    }
}

private void tbtelp_Leave(object sender, EventArgs e)
{
    if (tbtelp.Text == "")
    {
        seterror(tbtelp);
    }
    else
    {

```

<pre> remerror();     } } </pre>	
<b>Penjelasan:</b>	Error provider yang digunakan untuk mengecek isi dari textbox

<pre> public Form1() {     InitializeComponent();     filldataset(); } </pre>	
<b>Penjelasan:</b>	Memanggil InitializeComponent(); kemudian memanggil filldataset(); untuk mengisi combobox1

## 2.4 JAWABAN PRAKTIKUM

Percobaan 1 :

17. Mengapa terjadi error? Jelaskan!

Jawaban : karena pada id prodi length max 5 dan yang di inputkan ada 6

## **BAB III**

### **PENUTUP**

#### **3.1 KESIMPULAN**

Praktikum 4 ini mempelajari tentang mengkoneksikan console c# ke dalam database, membuat form yang langsung tersimpan ke database juga.

#### **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/>