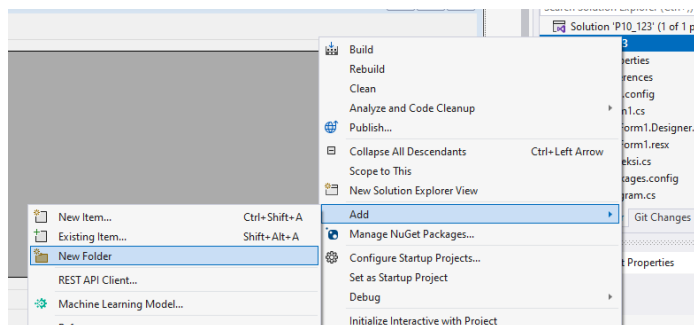


Universitas	: Universitas Logistik dan Bisnis Internasional (ULBI)
Program Studi	: D4 Teknik Informatika
Semester	: 3 (Tiga)
Mata Kuliah	: Pemrograman II (Pemrograman Berorientasi Objek)
Topik	: Praktikum 11 Membuat Program Windows C# dengan Basis Data (Insert, Update, Delete Data)

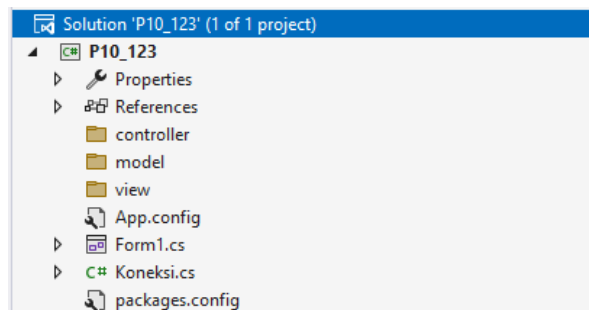
LATIHAN

INSERT DATA

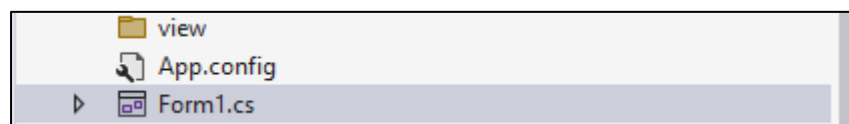
1. Pada praktikum ini kita akan terapkan konsep MVC dengan membuat folder controller, model, view untuk menyimpan code yang sudah dan yang akan dibuat. Pertama klik kanan pada project kalian → Add → New Folder → Beri nama **controller**, **model**, dan **view**



Sehingga menjadi seperti ini



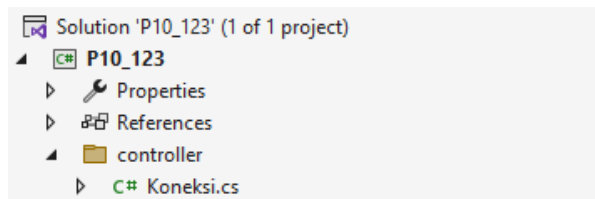
2. Selanjutnya adalah memindahkan file Form1.cs ke dalam folder view, dengan cara tarik(drag) Form1.cs pada solution explorer ke dalam folder view



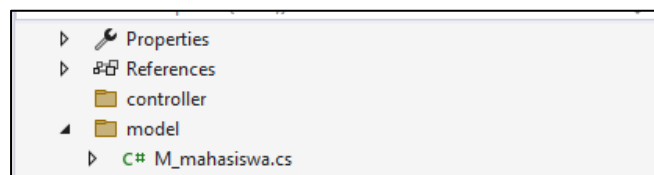
Sehingga menjadi seperti ini



3. Selanjutnya adalah memindahkan file Koneksi.cs ke dalam folder controller, caranya sama seperti langkah ke-2 dengan drag and drop



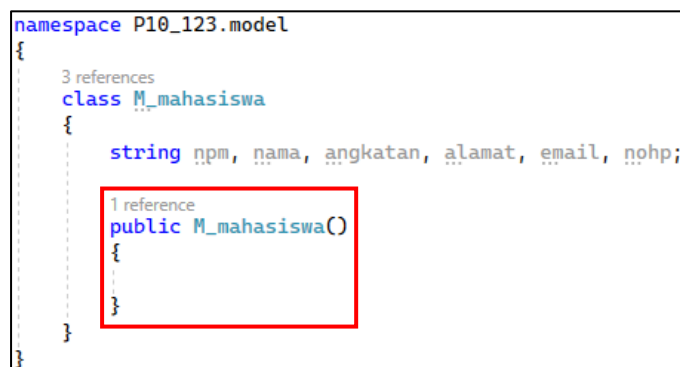
4. Langkah selanjutnya adalah dengan **membuat class baru di dalam folder model** dengan nama M_mahasiswa.cs. Pertama klik kanan pada folder model → Add → Class → Beri nama M_mahasiswa.cs → Add



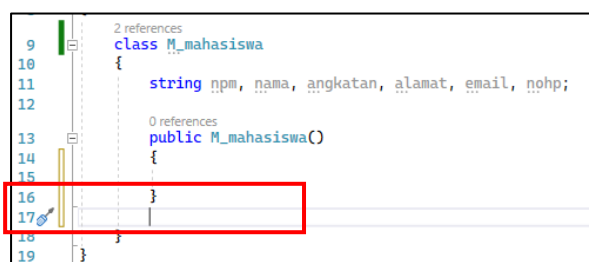
5. Kemudian kita lakukan deklarasi variabel secara eksplisit sesuai dengan yang ada di database yaitu : npm, nama, angkatan, alamat, email, nohp.



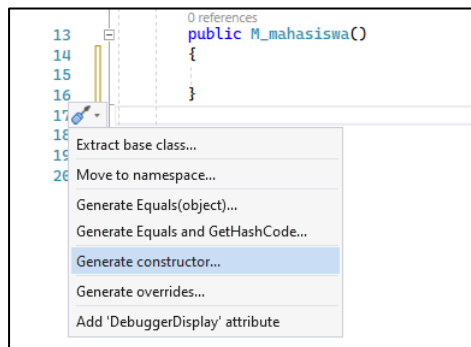
6. Selanjutnya adalah membuat method M_mahasiswa



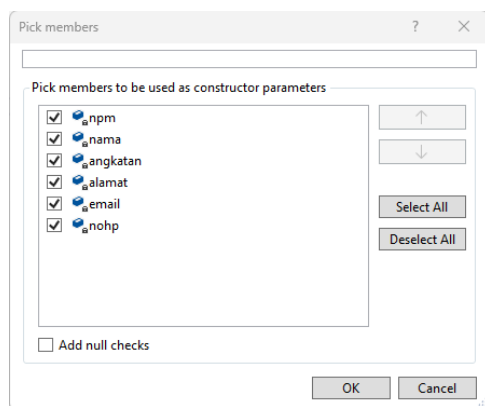
7. Jika sudah, selanjutnya adalah membuat method constructor dengan cara arahkan kursor mouse di bawah tutup kurung kurawal dari method M_mahasiswa yang telah dibuat seperti ini



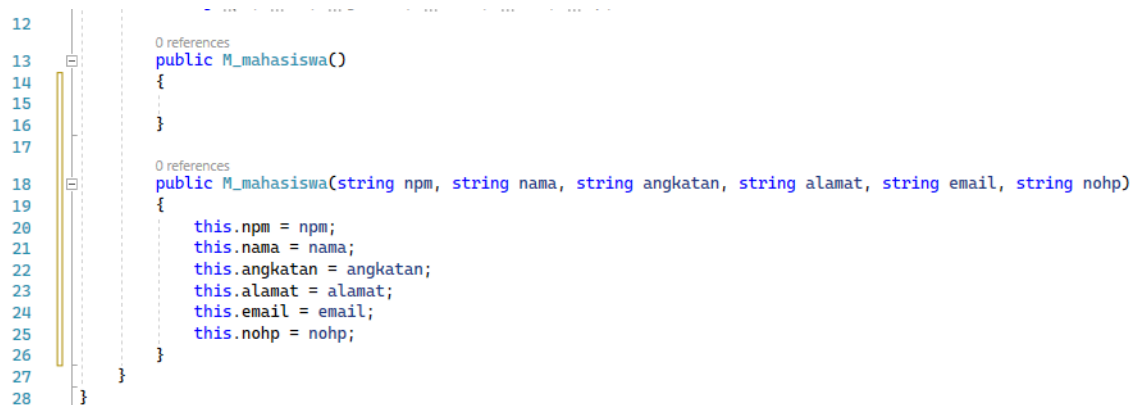
Jika terlihat logo atau gambar obeng sebelah angka row ke-17, klik pada gambar tersebut
→ Generate constructor...



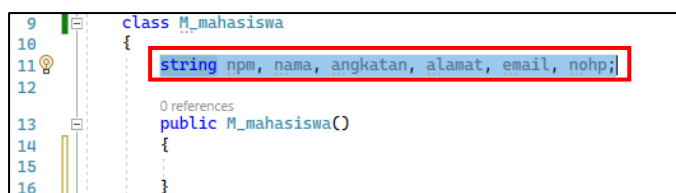
Kemudian akan muncul seperti gambar di bawah, lalu klik ok saja



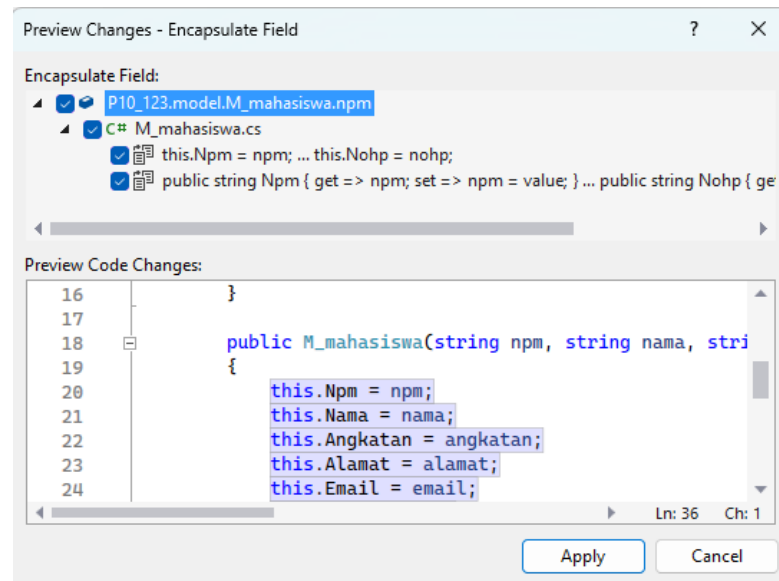
Sehingga akan seperti ini (constructor berhasil digenerate)



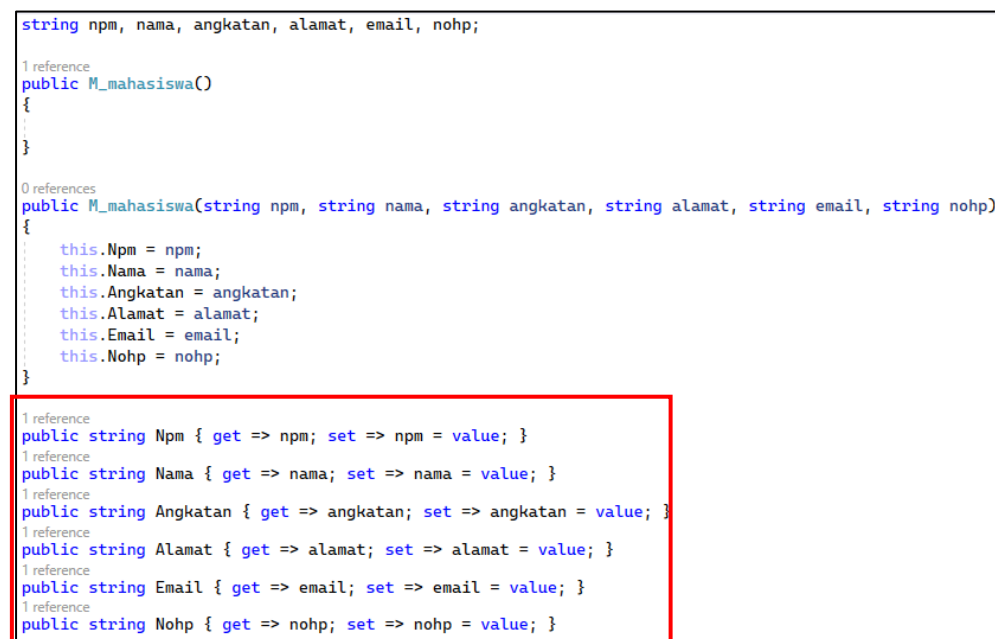
8. Selanjutnya adalah pembuatan proses enkapsulasi, pada langkah poin ke-5 kita sudah membuat deklarasi variabel, pada variabel-variabel tersebut kita lakukan blok menggunakan mouse seperti gambar di bawah



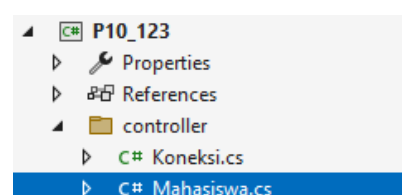
Setelah itu tekan 3 kombinasi tombol pada keyboard yaitu **CTRL + R + E** untuk melakukan proses enkapsulasi, jika sudah tampil seperti gambar di bawah, selanjutnya klik Apply



Sehingga akan tampil seperti gambar di bawah



9. Langkah selanjutnya adalah dengan **membuat class baru di dalam folder controller** dengan nama Mahasiswa.cs. Controller ini dibuat untuk mengatur fungsi CRUD atau sebagai jembatan antara model dan view. Pertama klik kanan pada folder model → Add → Class → Beri nama Mahasiswa.cs → Add



10. Jika sudah, isi code berikut pada Mahasiswa.cs

```
class Mahasiswa
{
    //Memanggil class Koneksi dan membuat objek baru
    Koneksi koneksi = new Koneksi();

    //Method insert
    0 references
    public bool Insert(M_mahasiswa mahasiswa)
    {
        Boolean status = false;
        try
        {
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message, "Gagal Insert", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        return status;
    }
}
```

Pada method insert di atas kita sudah membuat code di dalam catch jika kondisi penambahan data gagal.

11. Selanjutnya kita buat code untuk menambahkan datanya, copy paste code berikut di dalam kurung try

```
public bool Insert(M_mahasiswa mahasiswa)
{
    Boolean status = false;
    try
    {
    }
    catch(Exception e)
    {
        MessageBox.Show(e.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return status;
}
```

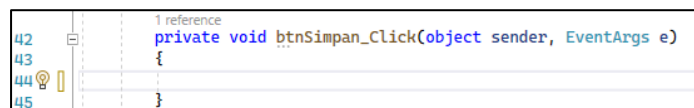
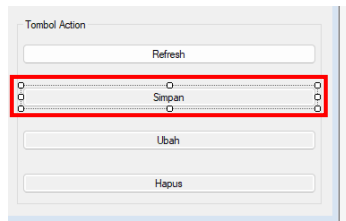
```
koneksi.OpenConnection();
koneksi.ExecuteNonQuery("INSERT INTO t_mahasiswa
(npm,nama,angkatan,alamat,email,nohp) VALUES ('" + mahasiswa.Npm + "','" +
mahasiswa>Nama + "','" + mahasiswa.Angkatan + "','" + mahasiswa.Alat +
"',"' + mahasiswa.Email + "','" + mahasiswa.Nohp + "')");
status = true;
MessageBox.Show("Data berhasil ditambahkan", "Informasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
koneksi.CloseConnection();
```

Sehingga akan seperti gambar di bawah

```
public bool Insert(M_mahasiswa mahasiswa)
{
    Boolean status = false;
    try
    {
        koneksi.OpenConnection();
        koneksi.ExecuteNonQuery("INSERT INTO t_mahasiswa (npm,nama,angkatan,alamat,email,nohp) VALUES ('" + mahasiswa.Npm + "
        status = true;
        MessageBox.Show("Data berhasil ditambahkan", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
        koneksi.CloseConnection();
    }
    catch(Exception e)
    {
        MessageBox.Show(e.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return status;
}
```

Jadi, setiap ada method yang mengakses database, harus dibuka terlebih dahulu koneksinya dengan cara memanggil method `OpenConnection` yang ada pada `Koneksi.cs`, jadi jika data berhasil disimpan ke database maka akan menampilkan message box dengan pesan “Data berhasil ditambahkan”. Setelah itu, jangan lupa untuk menutup koneksi dengan memanggil method `CloseConnection`.

12. Kemudian buka `Form1.cs` [Design], kemudian klik 2x pada button simpan untuk membuat event click pada button simpan.



13. Selanjutnya disini adalah membuat validasi untuk setiap form wajib diisi

```

private void btnSimpan_Click(object sender, EventArgs e)
{
    if(npm.Text == "" || nama.Text == "" || angkatan.SelectedIndex == -1 ||
    alamat.Text == "" || email.Text == "" || nohp.Text == "")
    {
        MessageBox.Show("Data tidak boleh kosong", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
    }
}

```

Copy code di atas sehingga menjadi seperti ini

```

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (npm.Text == "" || nama.Text == "" || angkatan.SelectedIndex == -1 || alamat.Text == "" || email.Text == "" || nohp.Text == "")
    {
        MessageBox.Show("Data tidak boleh kosong", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
    }
}

```

14. Selanjutnya pada else nya kita gunakan untuk mengirimkan data ke database. Sebelum itu, kita harus memanggil class `M_mahasiswa` yang ada di model pada file `Form1.cs`

```

public partial class Form1 : Form
{
    Koneksi koneksi = new Koneksi();
    M_mahasiswa m_mhs = new M_mahasiswa();

    1 reference
    public void Tampil()
    {
    }
}

```

Kembali pada bagian else, yang dilakukan pertama kali adalah membuat sebuah objek baru dari kelas Mahasiswa yang berada pada folder controller dan tuliskan code seperti gambar di bawah

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    if (npm.Text == "" || nama.Text == "" || angkatan.SelectedIndex == -1 || alamat.Te
    {
        MessageBox.Show("Data tidak boleh kosong", "Peringatan", MessageBoxButtons.OK,
    }
    else
    {
        Mahasiswa mhs = new Mahasiswa();
        m_mhs.Npm = npm.Text;
        m_mhs>Nama = nama.Text;
        m_mhs.Angkatan = angkatan.Text;
        m_mhs.Alatat = alamat.Text;
        m_mhs.Email = email.Text;
        m_mhs.Nohp = nohp.Text;

        mhs.Insert(m_mhs);
    }
}
```

Setelah insert, selanjutnya adalah membuat component menjadi kosong seperti awal membuka form.

```
1 reference
public Form1()
{
    InitializeComponent();
}

0 references
public void ResetForm()
{
    npm.Text = "";
    nama.Text = "";
    angkatan.SelectedIndex = -1;
    alamat.Text = "";
    email.Text = "";
    nohp.Text = "";
}
```

Kemudian panggil Method ResetForm dan Tampil pada bagian else setelah insert seperti gambar berikut

```

else
{
    Mahasiswa mhs = new Mahasiswa();
    m_mhs.Npm = npm.Text;
    m_mhs>Nama = nama.Text;
    m_mhs.Angkatan = angkatan.Text;
    m_mhs.Alatnat = alamat.Text;
    m_mhs.Email = email.Text;
    m_mhs.Nohp = nohp.Text;

    mhs.Insert(m_mhs);

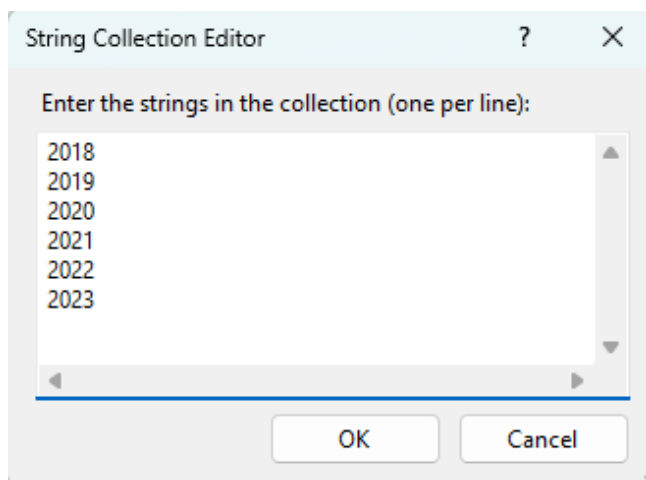
    ResetForm();
    Tampil();
}

```

15. Kemudian kembali buka Form1.cs [Design], lalu klik pada combobox angkatan dan tambahkan data angkatan pada properties Items.



Isikan itemnya kurang lebih seperti gambar di bawah



16. Kemudian jalankan aplikasinya, jika salah satu form tidak diisi dan menekan tombol Simpan maka akan tampil validasi warning seperti ini

The screenshot shows the ULBI application interface. At the top, there is a table titled "Table Data Mahasiswa" with columns: NPM, Nama, Angkatan, Alamat, Email, and No HP. The table contains three rows of data. Below the table is a form titled "Form Data Mahasiswa" with input fields for NPM, Nama, Angkatan (dropdown), Alamat, Email, and No HP. To the right of the form is a "Tombol Action" section with buttons for Refresh, Simpan, Ubah, and Hapus. A warning dialog box titled "Peringatan" is displayed in the center, with a yellow warning icon and the message "Data tidak boleh kosong". The dialog has an "OK" button.

	NPM	Nama	Angkatan	Alamat	Email	No HP
	714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
	714220202	Lionel Akbar	2022	Banjarmasin	leoakbar@ulbi.ac.id	081224544200
▶	714220203	Karin Nurtaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
*						

Form Data Mahasiswa

NPM: 714220204

Nama:

Angkatan:

Alamat:

Email:

No HP:

Tombol Action

Refresh

Simpan

Ubah

Hapus

Peringatan

! Data tidak boleh kosong

OK

Jika data yang diinputkan lengkap maka akan tampil seperti ini

The screenshot shows the ULBI application interface. At the top, there is a table titled "Table Data Mahasiswa" with columns: NPM, Nama, Angkatan, Alamat, Email, and No HP. The table contains three rows of data. Below the table is a form titled "Form Data Mahasiswa" with input fields for NPM, Nama, Angkatan (dropdown), Alamat, Email, and No HP. To the right of the form is a "Tombol Action" section with buttons for Refresh, Simpan, Ubah, and Hapus. An information dialog box titled "Informasi" is displayed in the center, with a blue information icon and the message "Data berhasil ditambahkan". The dialog has an "OK" button.

	NPM	Nama	Angkatan	Alamat	Email	No HP
	714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
	714220202	Lionel Akbar	2022	Banjarmasin	leoakbar@ulbi.ac.id	081224544200
▶	714220203	Karin Nurtaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
*						

Form Data Mahasiswa

NPM: 714220204

Nama: Pragaries Septian

Angkatan: 2022

Alamat: Bandung

Email: akucintadia@ulbi.ac.id

No HP: 08123456789

Tombol Action

Refresh

Simpan

Ubah

Hapus

Informasi

i Data berhasil ditambahkan

OK

Ketika menekan tombol OK seharusnya data yang diinputkan akan langsung terlihat pada datagridview

Table Data Mahasiswa						
	NPM	Nama	Angkatan	Alamat	Email	No HP
▶	714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
	714220202	Lionel Akbar	2022	Banjarmasin	leoakbar@ulbi.ac.id	081224544200
	714220203	Karin Nurtaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
	714220204	Praganes Septian	2022	Bandung	akucintadia@ulbi.ac.id	08123456789
*						

UPDATE DATA

1. Buka folder controller → Mahasiswa.cs. Untuk update kita akan membuat method baru untuk mengupdate data yang ada di database. Untuk codenya hampir sama seperti insert, jadi kita lakukan copy paste pada method insert dengan melakukan blok menggunakan mouse seperti gambar di bawah

```
//Method insert
1 reference
public bool Insert(M_mahasiswa mahasiswa)
{
    Boolean status = false;
    try
    {
        koneksi.OpenConnection();
        koneksi.ExecuteNonQuery("INSERT INTO t_mahasiswa (npm,nama,angkatan,alamat,email,nohp) VALUES ('" + mahasiswa.Npm + "','" + mahasiswa.Nama
        status = true;
        MessageBox.Show("Data berhasil ditambahkan", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
        koneksi.CloseConnection();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return status;
}
```

Kemudian lakukan paste → ubah nama method menjadi Update

```
34 //Method update
35 0 references
36 public bool Update(M_mahasiswa mahasiswa)
37 {
38     Boolean status = false;
39     try
40     {
41         koneksi.OpenConnection();
42         koneksi.ExecuteNonQuery("INSERT INTO t_mahasiswa (npm,nama,angkatan,alamat,email,nohp) VALUES ('" + mahasiswa.Npm + "','" + mahasiswa.Nama
43         status = true;
44         MessageBox.Show("Data berhasil ditambahkan", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
45         koneksi.CloseConnection();
46     }
47     catch (Exception e)
48     {
49         MessageBox.Show(e.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
50     }
51     return status;
}
```

2. Tambah parameter npm pada method Update

```
35 //Method Update
36 0 references
37 public bool Update(M_mahasiswa mahasiswa, string npm)
38 {
    Boolean status = false;
```

3. Ubah query updatenya menjadi

```
koneksi.ExecuteNonQuery("UPDATE t_mahasiswa SET nama='" + mahasiswa.Nama + "','"
+ "angkatan='" + mahasiswa.Angkatan + "','" + "alamat='" + mahasiswa.Alatat +
"'," + "email='" + mahasiswa.Email + "','" + "nohp='" + mahasiswa.Nohp + '"
WHERE npm='" + npm_mhs + '"");
```

Sehingga akan seperti ini

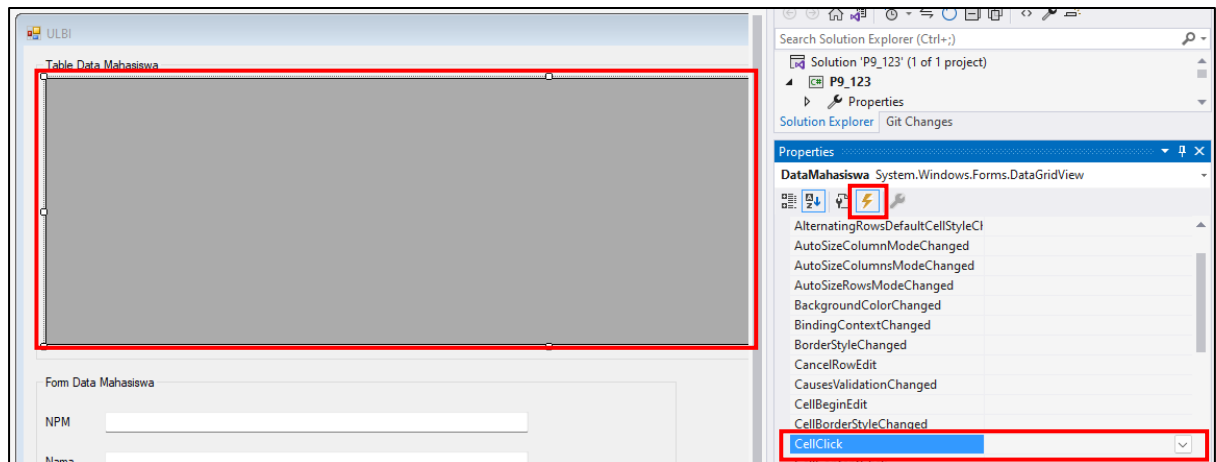
```
//Method update
0 references
public bool Update(M_mahasiswa mahasiswa, string npm)
{
    Boolean status = false;
    try
    {
        koneksi.OpenConnection();
        koneksi.ExecuteNonQuery("UPDATE t_mahasiswa SET npm='" + mahasiswa.Npm + "','" + "nama='" + mahasiswa.Nama + "','" + "angkata
        status = true;
        MessageBox.Show("Data berhasil ditambahkan", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
        koneksi.CloseConnection();
    }
}
```

Dan jangan lupa untuk mengubah pesan pada MessageBox

```
MessageBox.Show("Data berhasil diubah", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
MessageBox.Show(e.Message, "Gagal Update", MessageBoxButtons.OK, MessageBoxIcon.Error);
```

4. Buka Form1.cs [Design], disini kita akan menggunakan event yang ada pada datagridview, maka klik terlebih dahulu datagridviewnya → lihat pada tab properties → klik Events (logo petir kuning) → cari CellClick → klik 2x pada CellClick



5. Jika sudah maka akan tampil seperti gambar di bawah

```
77 private void DataMahasiswa_CellClick(object sender, DataGridViewCellEventArgs e)
78 {
79
80 }
```

Fungsi Event CellClick pada DataGridView akan dipicu (triggered) ketika salah satu sel di dalam DataGridView diklik.

Ketika sebuah sel di DataGridView ditekan (clicked), event CellClick dijalankan. Saat event ini dipicu, kode yang ada pada method DataMahasiswa_CellClick akan dieksekusi.

6. Isi code seperti berikut dimulai dengan id hingga nohp sesuai dengan yg ada di database

```
private void DataMahasiswa_CellClick(object sender, DataGridViewCellEventArgs e)
{
    npm.Text = DataMahasiswa.Rows[e.RowIndex].Cells[0].Value.ToString();
    nama.Text = DataMahasiswa.Rows[e.RowIndex].Cells[1].Value.ToString();
    angkatan.Text = DataMahasiswa.Rows[e.RowIndex].Cells[2].Value.ToString();
    alamat.Text = DataMahasiswa.Rows[e.RowIndex].Cells[3].Value.ToString();
    email.Text = DataMahasiswa.Rows[e.RowIndex].Cells[4].Value.ToString();
    nohp.Text = DataMahasiswa.Rows[e.RowIndex].Cells[5].Value.ToString();
}
```

7. Coba running aplikasi, dan kalian coba pilih data yang ada di datagridview, data yang kalian pilih akan tampil pada form di bawahnya. Pada gambar di bawah, kita melakukan klik pada data dengan npm 714220203 dan bisa dilihat data yang dipilih akan ditampilkan pada form.

NPM	Nama	Angkatan	Alamat	Email	No HP
714220201	Dahlan Fernando	2022	Bandung	dahlanfermando@ulbi.ac.id	081222210909
714220202	Lionel Akbar	2022	Banjarmasin	leoakbar@ulbi.ac.id	081224544200
714220203	Karin Nurlaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
714220204	Pragianes Septian	2022	Bandung	akucintadia@ulbi.ac.id	08123456789

NPM	714220203
Nama	Karin Nurlaela
Angkatan	2022
Alamat	Bogor
Email	inikarin@ulbi.ac.id
No HP	085722900707

Tombol Action

Refresh

Simpan

Ubah

Hapus

8. Selanjutnya adalah eksekusi method update. Caranya sama seperti insert, buka Form1.cs[Design] → klik 2 kali pada button Ubah

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

9. Lakukan copy pada event click button simpan

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    if (npm.Text == "" || nama.Text == "" || angkatan.SelectedIndex == -1 || alamat.Text == "" || email.Text == "")
    {
        MessageBox.Show("Data tidak boleh kosong", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        Mahasiswa mhs = new Mahasiswa();
        m_mhs.Npm = npm.Text;
        m_mhs>Nama = nama.Text;
        m_mhs.Angkatan = angkatan.Text;
        m_mhs.Alatat = alamat.Text;
        m_mhs.Email = email.Text;
        m_mhs.Nohp = nohp.Text;

        mhs.Insert(m_mhs);

        ResetForm();
        Tampil();
    }
}
```

Kemudian paste pada event button ubah, yang harus diubah hanya pada bagian mhs.Insert menjadi mhs.Update

```
private void btnUbah_Click(object sender, EventArgs e)
{
    if (npm.Text == "" || nama.Text == "" || angkatan.SelectedIndex == -1 || alamat.Text == "" || email.Text == "")
    {
        MessageBox.Show("Data tidak boleh kosong", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        Mahasiswa mhs = new Mahasiswa();
        m_mhs.Npm = npm.Text;
        m_mhs>Nama = nama.Text;
        m_mhs.Angkatan = angkatan.Text;
        m_mhs.Alatamat = alamat.Text;
        m_mhs.Email = email.Text;
        m_mhs.NoHp = nohp.Text;

        mhs.Update(m_mhs);

        ResetForm();
        Tampil();
    }
}
```

17. Build aplikasi dan jalankan

18. Apa yang terjadi? Error? Mengapa terjadi demikian? Jelaskan dan perbaiki!

19. Setelah diperbaiki, jalankan aplikasinya, jika salah satu form tidak diisi dan menekan tombol Ubah maka akan tampil validasi warning seperti ini

The screenshot shows a Windows application titled 'ULBI'. It contains a 'Table Data Mahasiswa' with columns: NPM, Nama, Angkatan, Alamat, Email, and No HP. The table has four rows of data. Below the table is a 'Form Data Mahasiswa' with input fields for NPM, Nama, Angkatan, Alamat, Email, and No HP. The 'Alamat' field is highlighted with a red rectangle. A 'Peringatan' (Warning) dialog box is displayed in the center, with a yellow warning icon and the text 'Data tidak boleh kosong' (Data cannot be empty). The dialog has an 'OK' button. To the right of the form is a 'Tombol Action' (Action Button) section with buttons for Refresh, Simpan (Save), Ubah (Update), and Hapus (Delete).

NPM	Nama	Angkatan	Alamat	Email	No HP
714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
714220202	Lionel Akbar	2022	Banjarnasin	leoakbar@ulbi.ac.id	081224544200
714220203	Karin Nurfaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
714220204	Pragaries Septian	2022	Bandung	akucintadia@ulbi.ac.id	08123456789

Form Data Mahasiswa

NPM: 714220204

Nama: Pragaries Septian

Angkatan: 2022

Alamat:

Email: akucintadia@ulbi.ac.id

No HP: 08123456789

Tombol Action

Refresh

Simpan

Ubah

Hapus

Lalu kita coba update angkatan dan no hp untuk NPM 714220204

Table Data Mahasiswa

NPM	Nama	Angkatan	Alamat	Email	No HP
714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
714220202	Lionel Akbar	2022	Bandung	leoakbar@ulbi.ac.id	081224544200
714220203	Karin Nurfaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
714220204	Praganes Septian	2022	Bandung	akucintadia@ulbi.ac.id	08123456789

Form Data Mahasiswa

NPM: 714220204

Nama: Praganes Septian

Angkatan: 2021

Alamat: Bandung

Email: akucintadia@ulbi.ac.id

No HP: 089876543211

Informasi

Data berhasil diubah

OK

Tombol Action

Refresh

Simpan

Ubah

Hapus

Setelah klik OK maka data pada datagridview akan otomatis ter update

Table Data Mahasiswa

NPM	Nama	Angkatan	Alamat	Email	No HP
714220201	Dahlan Fernando	2022	Bandung	dahlanfernando@ulbi.ac.id	081222210909
714220202	Lionel Akbar	2022	Bandung	leoakbar@ulbi.ac.id	081224544200
714220203	Karin Nurfaela	2022	Bogor	inikarin@ulbi.ac.id	085722900707
714220204	Praganes Septian	2021	Bandung	akucintadia@ulbi.ac.id	089876543211

20. CHALLENGE : Buat event click pada tombol Refresh kemudian isikan dengan code untuk melakukan refresh dan reload data pada datagridview dan form inputan menjadi kosong.

Contoh hasil tombol Refresh [DISINI](#)

DELETE DATA

1. Buka folder controller → Mahasiswa.cs. Untuk delete kita akan membuat method baru untuk menghapus data yang ada di database. Untuk codenya hampir sama seperti sebelumnya, jadi kita lakukan copy paste pada method update dengan melakukan blok menggunakan mouse seperti gambar di bawah

```
//Method Update
1 reference
public bool Update(M_mahasiswa mahasiswa, string npm_mhs)
{
    Boolean status = false;
    try
    {
        koneksi.OpenConnection();
        koneksi.ExecuteNonQuery("UPDATE t_mahasiswa SET nama='" + mahasiswa.Nama + "', " + "angkatan=" +
        status = true;
        MessageBox.Show("Data berhasil diubah", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
        koneksi.CloseConnection();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Gagal Update", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return status;
}
```

Kemudian lakukan paste → ubah nama method menjadi Delete

```
//Method Delete
0 references
public bool Delete(M_mahasiswa mahasiswa, string npm_mhs)
{
}
```

2. Ubah pada bagian parameter hingga menyisakan parameter npm_mhs saja

```
//Method Delete
0 references
public bool Delete(string npm_mhs)
{
}
```

3. Ubah query delete dan pesan dari message box seperti gambar di bawah

```
koneksi.ExecuteNonQuery("DELETE FROM t_mahasiswa WHERE npm='" + npm_mhs + "'");

//Method Delete
1 reference
public bool Delete(string npm_mhs)
{
    Boolean status = false;
    try
    {
        koneksi.OpenConnection();
        koneksi.ExecuteNonQuery("DELETE FROM t_mahasiswa WHERE npm='" + npm_mhs + "'");
        status = true;
        MessageBox.Show("Data berhasil dihapus", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);
        koneksi.CloseConnection();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Gagal Hapus", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return status;
}
```


4. Selanjutnya adalah eksekusi method delete. Caranya sama seperti insert dan update, buka Form1.cs[Design] → klik 2 kali pada button Hapus

```
123
124 1 reference
125 private void btnHapus_Click(object sender, EventArgs e)
126 {
127 }
```

5. Sebelum data dihapus, disini kita akan membuat pesan apakah yakin akan menghapus data yang dipilih dengan menggunakan DialogResult

```
private void btnHapus_Click(object sender, EventArgs e)
{
    DialogResult pesan = MessageBox.Show(
        "Apakah yakin akan menghapus data ini?",
        "Perhatian",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);
}
```

6. Selanjutnya adalah membuat kondisi jika DialogResult yang dipilih adalah Yes

```
private void btnHapus_Click(object sender, EventArgs e)
{
    DialogResult pesan = MessageBox.Show(
        "Apakah yakin akan menghapus data ini?",
        "Perhatian",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);

    if (pesan == DialogResult.Yes)
    {
        Mahasiswa mhs = new Mahasiswa();
        mhs.Delete(npm.Text);
        ResetForm();
        Tampil();
    }
}
```

7. Build aplikasi dan jalankan, lalu coba hapus data yang ada pada datagridview

Pengumpulan Hasil :

Jika sudah berhasil, hasil praktikum dimasukkan ke dalam github masing-masing. Gunakan repository bernama PemrogramanII_PBO.

Buat folder dengan nama **Pertemuan11** kemudian di dalamnya buat 2 folder yaitu **folder Praktikum** dan **Hasil**. Isi **folder Praktikum** adalah aplikasi C# yang sudah dibuat. Isi **folder Hasil** adalah output dari aplikasi yang sudah dibuat.