

Setelah melaksanakan praktikum ini diharapkan mahasiswa dapat :

- ✓ Mengingat kembali penggunaan collection yang berfungsi sebagai container untuk menyimpan banyak object.
- ✓ Mengingat kembali manfaat constructor untuk inisialisasi data
- ✓ Membuat form entry data
- ✓ Memahami cara komunikasi antar form
- ✓ Memahami menggunakan event dan delegate untuk membuat custome event

Teori Singkat

Pada praktikum kali ini, kita akan membuat project baru (**AkademikApp**) dengan studi kasus membuat form input data mahasiswa. Project ini mengimplementasikan konsep dasar operasi CRUD (Create, Read, Update, Delete) yang biasanya digunakan dalam pembuatan aplikasi database. Karena belum menggunakan database sebagai media penyimpanan, sebagai gantinya kita akan menggunakan object collection untuk media penyimpanan sementara.

Di praktikum ini kita juga menggunakan *delegate* dan *event* untuk membuat sendiri event yang dibutuhkan seperti event *OnCreate* yang akan dipanggil ketika input data baru dan *OnUpdate* ketika mengedit data. Penggunaan event merupakan salah satu solusi untuk pertukaran data antar class, dalam hal ini class berupa sebuah Form.

Sebagai gambaran, berikut adalah tampilan aplikasi yang akan kita buat.

The screenshot shows a Java Swing application window titled "Data Mahasiswa". Inside the window is a table with four columns: "No.", "Npm", "Nama", and "Angkatan". The table contains four rows of student data. A modal dialog box titled "Tambah Data Mahasiswa" is displayed in front of the table. This dialog has three text input fields: "NPM" with the value "18.01.4117", "Nama" with the value "SHODIK AGIK UTOMO", and "Angkatan" with the value "2018". Below these fields are two buttons: "Simpan" and "Selesai". At the bottom of the main application window, there are four buttons: "Tambah", "Perbaiki", "Hapus", and "Selesai".

No.	Npm	Nama	Angkatan
1	17.11.0919	HERDIANTO	2017
2	17.11.0920	SYELVI NUR DWI JULIANA	2017
3	17.11.0921	MUHAMMAD KHAIRUL RIJAL	2017
4	17.11.0922	YOHANES FIDES ANAS SUSSETA	2017

Tambah Data Mahasiswa

NPM: 18.01.4117

Nama: SHODIK AGIK UTOMO

Angkatan: 2018

Simpan Selesai

Tambah Perbaiki Hapus Selesai

Aplikasi ini terdiri dari dua form yaitu:

1. FrmMahasiswa

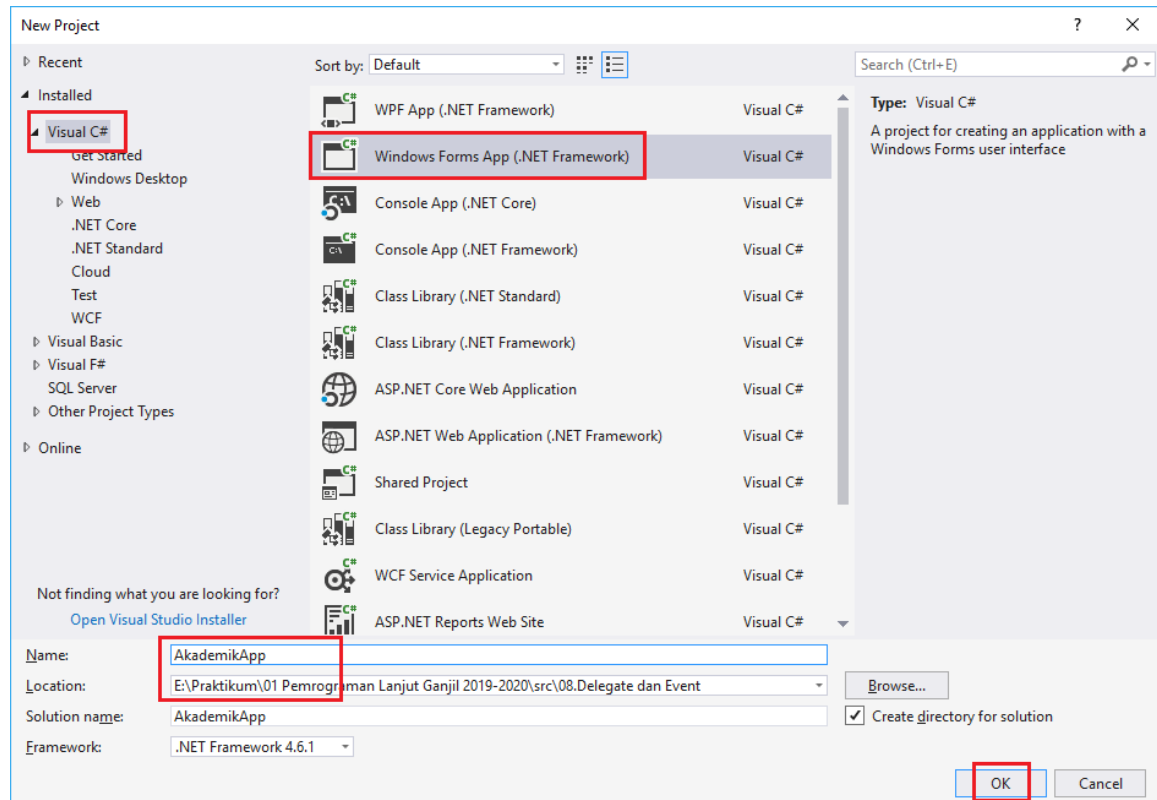
Berfungsi sebagai form induk yang digunakan untuk menampilkan data mahasiswa.

2. FrmEntryMahasiswa

Form anak yang berfungsi sebagai form entri data.

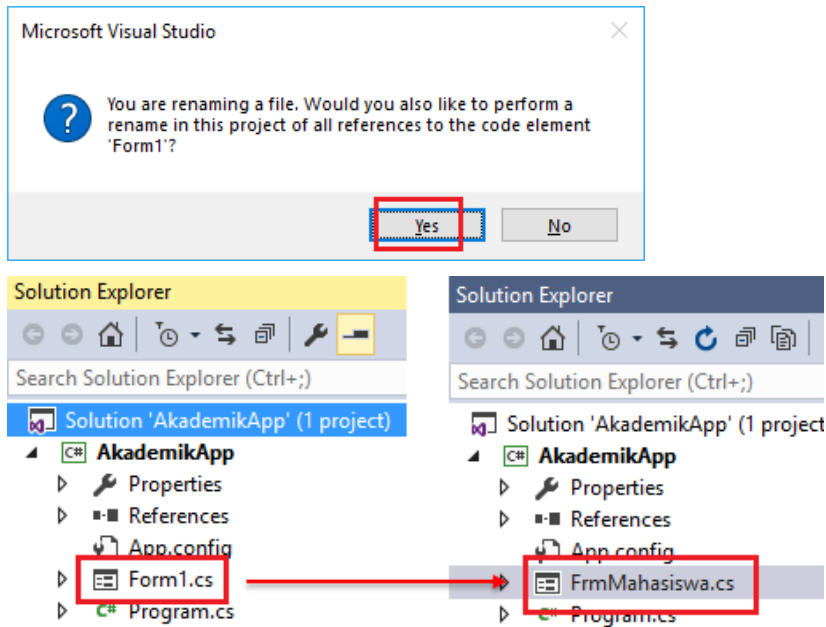
Latihan 8.1 – Membuat Project Baru

1. Awali selalu pekerjaan dgn doa, mudah-mudahan diberi kemudahan dan dapat memberikan manfaat
2. Jalankan aplikasi Visual Studio .NET
3. Buat project baru dengan cara klik menu File -> New -> Project -> Windows Forms Application



Isian Name diisikan dengan **AkademikApp**. Untuk penamaan project perlu diperhatikan huruf besar kecilnya karena nama project secara default akan menjadi namespace yang akan berpengaruh terhadap penulisan kode program. Untuk isian location menyesuaikan.

4. Setelah berhasil menambahkan project baru, kemudian ganti nama Form1 menjadi FrmMahasiswa, dengan cara klik kanan Form1 -> Rename, kemudian klik tombol Yes pada dialog konfirmasi.



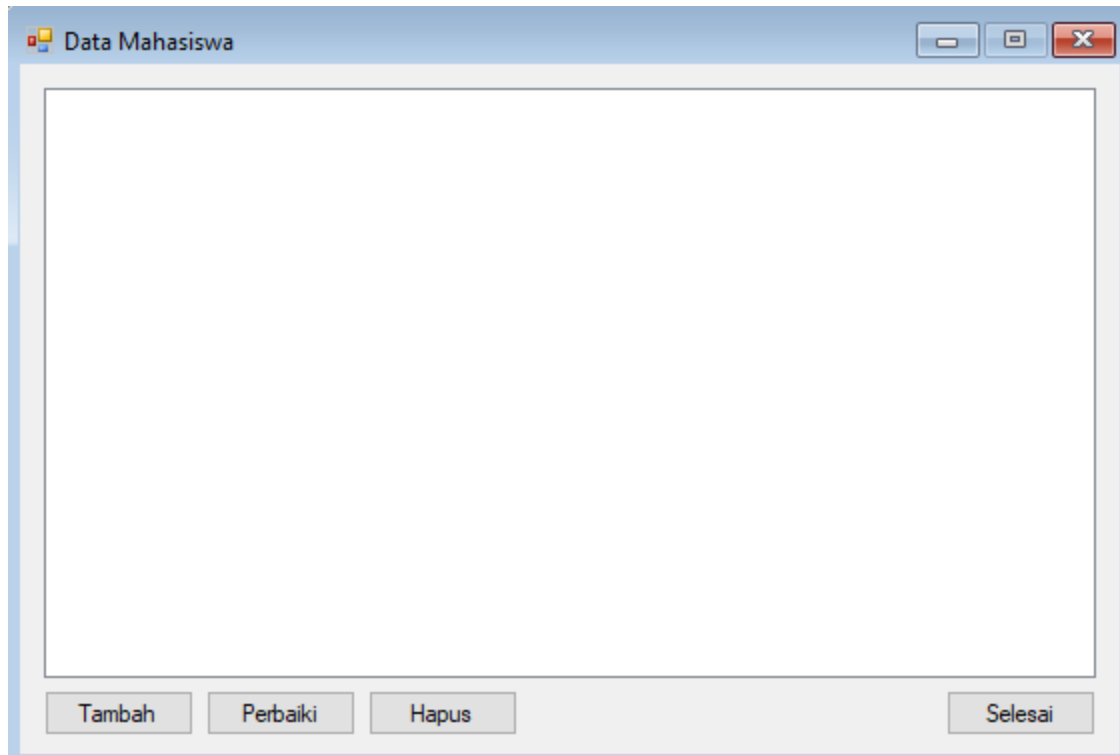
5. Kemudian rancang tampilan aplikasi seperti gambar di bawah ini:



6. Setelah itu atur properties masing-masing komponen melalui properties window dengan aturan sesuai tabel berikut:

No	Komponen	Property	Nilai/Value
-	Form	Text StartPosition MaximizeBox	Data Mahasiswa CenterScreen False
1	ListView	Name	lvwMahasiswa
2	Button	Name Text	btnTambah Tambah
3	Button	Name Text	btnPerbaiki Perbaiki
4	Button	Name Text	btnHapus Hapus
5	Button	Name Text	btnSelesai Selesai

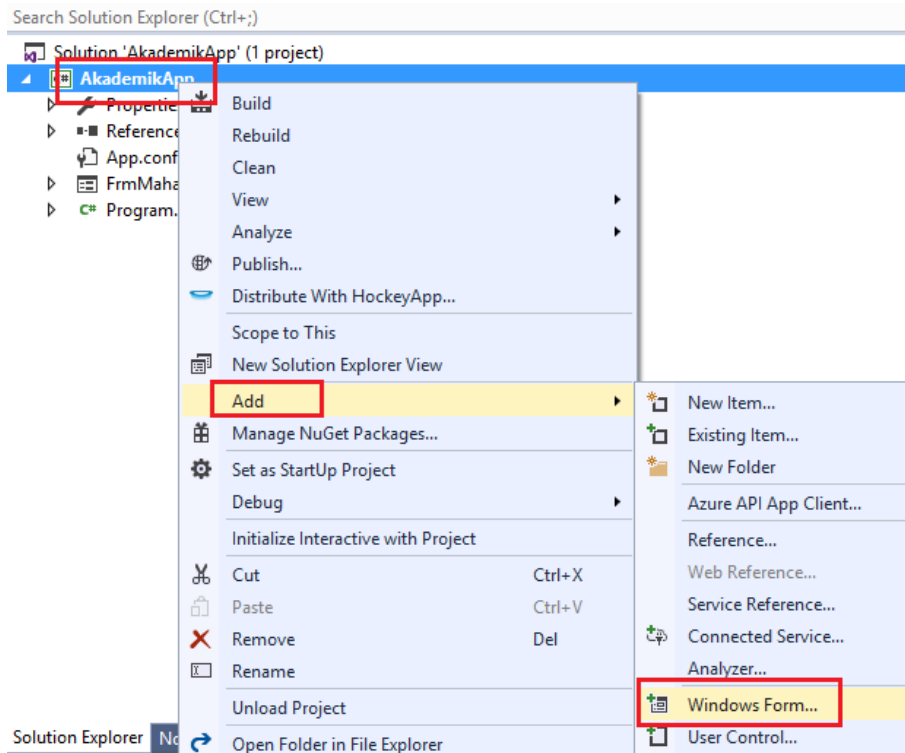
7. Setelah pengaturan properties akan didapat tampilan seperti berikut:



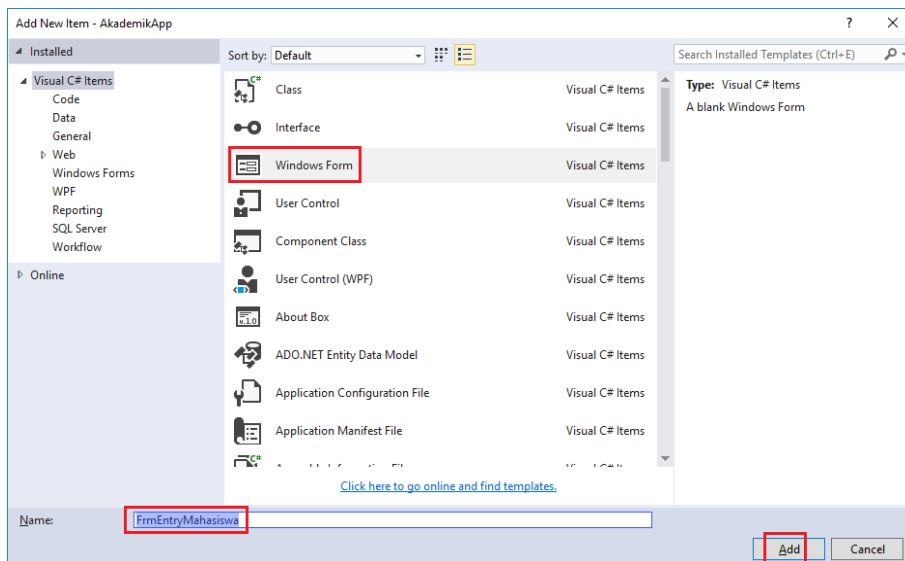
8. Kemudian arahkan entry point dari class Program ke Form Mahasiswa

Latihan 8.2 – Menambahkan Form Entry Data Mahasiswa

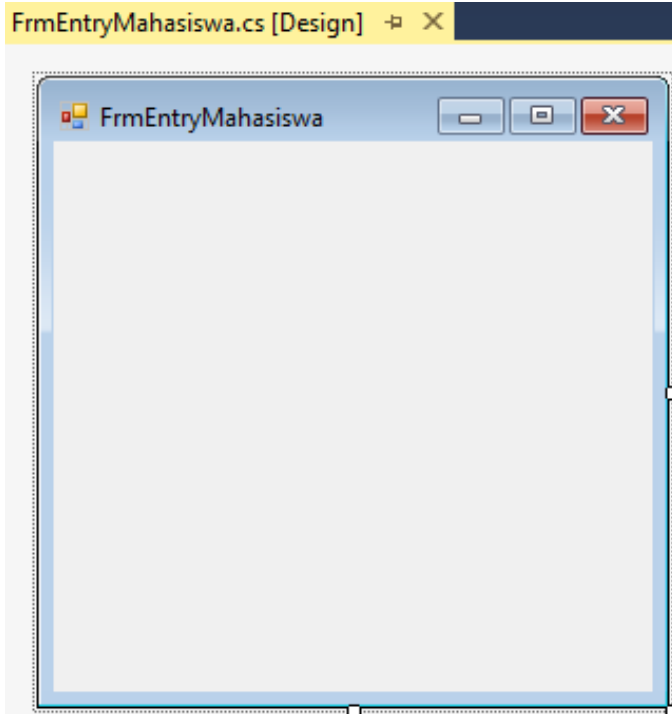
1. Klik kanan project -> Add -> Windows Form



Kemudian pilih Windows Forms, untuk isian Name diisi dengan **FrmEntryMahasiswa**.



Setelah itu akan tampil form entry mahasiswa



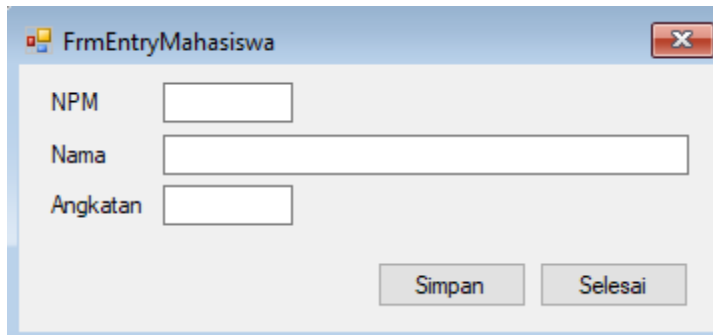
2. Kemudian rancang tampilan aplikasi seperti gambar di bawah ini:



3. Setelah itu atur properties masing-masing komponen melalui properties window dengan aturan sesuai tabel berikut

No	Komponen	Property	Nilai/Value
-	Form	FormBorderStyle StartPosition MinimizeBox MaximizeBox	FixedSingle CenterScreen False False
1	TextBox	Name	txtNpm
2	TextBox	Name	txtNama
3	TextBox	Name	txtAngkatan
4	Button	Name Text	btnSimpan Simpan
7	Button	Name Text	btnSelesai Selesai

4. Setelah pengaturan properties akan didapat tampilan sebagai berikut:

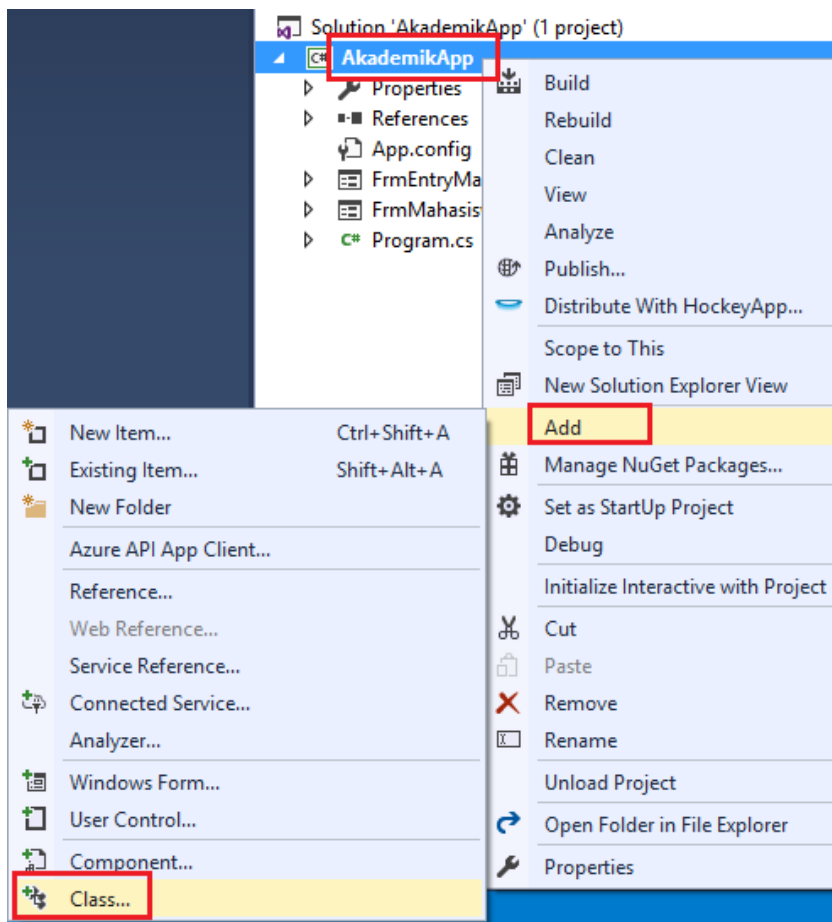


The screenshot shows a Windows form titled 'FrmEntryMahasiswa'. It contains three text input fields: 'NPM', 'Nama', and 'Angkatan'. Below the fields are two buttons: 'Simpan' (Save) and 'Selesai' (Finish).

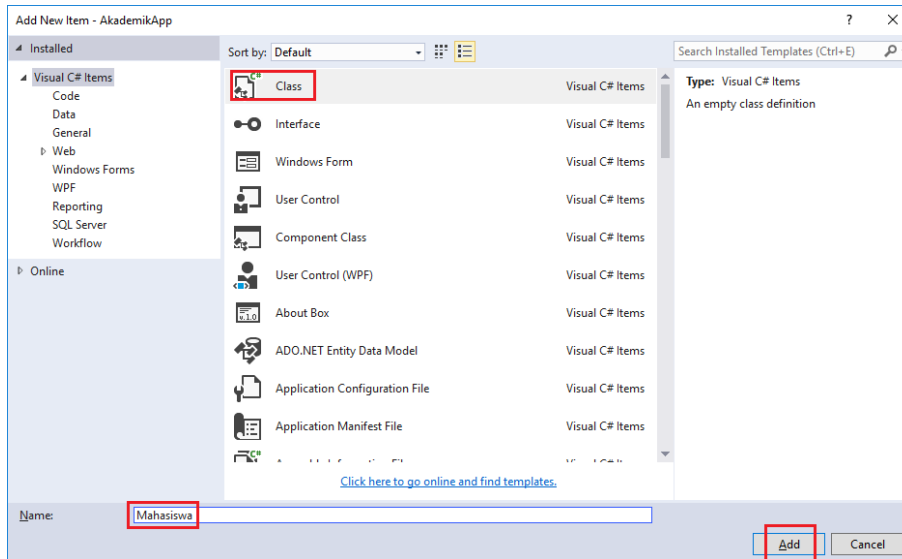
Latihan 8.3 – Menambahkan Class Mahasiswa

Class Mahasiswa digunakan untuk membuat objek mahasiswa. Objek mahasiswa ini nantinya akan kita gunakan untuk menampung data yang diinputkan sebelum akhirnya disimpan ke dalam objek collection.

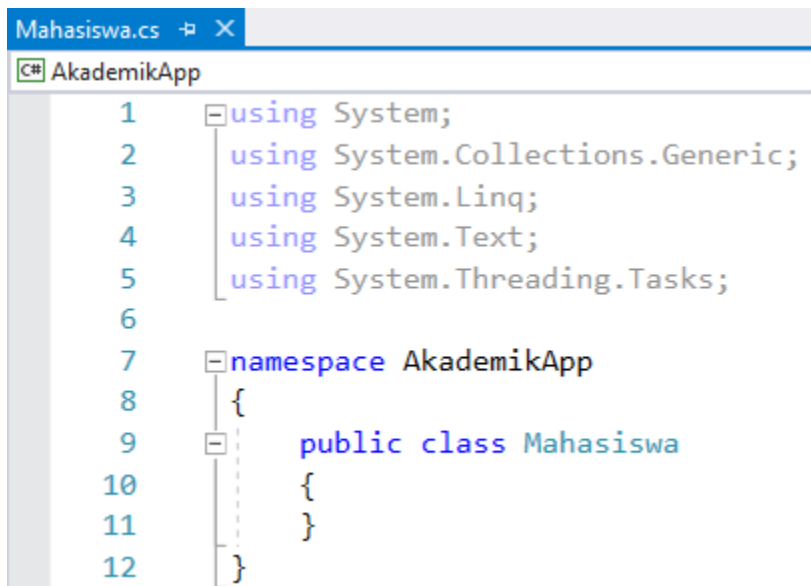
1. Klik kanan project -> Add -> Class...



Kemudian pilih Class, untuk isian Name diisi dengan **Mahasiswa**.



Setelah itu akan tampil editor code class Mahasiswa.



Kemudian lengkapi kodenya seperti berikut:

```
public class Mahasiswa
{
    public string Npm { get; set; }
    public string Nama { get; set; }
    public string Angkatan { get; set; }
}
```

Pada kode di atas, kita menambahkan beberapa property untuk menyimpan informasi yang berhubungan dengan data mahasiswa seperti Npm, Nama, dan Angkatan.

Latihan 8.4 – Melengkapi Kode untuk Form Entry Mahasiswa

1. Di langkah pertama ini kita akan mendeklarasikan *delegate* dan *event*. Ada 2 *event* yang akan kita buat yaitu *event OnCreate* dan *OnUpdate*. Kedua *event* ini dipanggil pada saat user menginputkan data baru (*OnCreate*) dan ketika user mengupdate data (*OnUpdate*). Sebelum mendeklarasikan *event* terlebih dulu kita harus mendeklarasikan tipe dari *event* tersebut dengan menggunakan *delegate*.

Aktifkan editor code form entry mahasiswa, kemudian lengkapi kode untuk mendeklarasikan *delegate* dan *event* seperti berikut:

```
public partial class FrmEntryMahasiswa : Form
{
    // deklarasi tipe data untuk event OnCreate dan OnUpdate
    public delegate void CreateUpdateEventHandler(Mahasiswa mhs);

    // deklarasi event ketika terjadi proses input data baru
    public event CreateUpdateEventHandler OnCreate;

    // deklarasi event ketika terjadi proses update data
    public event CreateUpdateEventHandler OnUpdate;

    // deklarasi variabel untuk menyimpan status entry data (input baru atau
    update)
    private bool isNewData = true;

    // deklarasi variabel/objek mhs untuk meyimpan data mahasiswa
    private Mahasiswa mhs;

    // Constructor default
    public FrmEntryMahasiswa()
    {
        InitializeComponent();
    }
}
```

Pada kode di atas, selain mendeklarasikan *delegate* dan *event*, kita juga mendeklarasikan dua buah field yaitu *isNewData* dan *mhs*. Field *isNewData* berfungsi untuk menyimpan status entri data apakah sedang menginput data baru (create) atau mengupdate data yang sudah ada (update). Sedangkan field *mhs* berfungsi untuk menyimpan objek dari class Mahasiswa.

2. Selanjutnya kita akan menambahkan dua buah constructor lagi yang digunakan untuk keperluan inisialisasi data ketika terjadi proses input data baru (create) atau edit data (update). Masih di editor code yang sama, kemudian tambahkan dua constructor berikut setelah constructor default.

```
// Constructor untuk inisialisasi data ketika entri data baru
public FrmEntryMahasiswa(string title)
    : this()
{
    // ganti text/judul form
    this.Text = title;
}

// Constructor untuk inisialisasi data ketika mengedit data
public FrmEntryMahasiswa(string title, Mahasiswa obj)
    : this()
{
    // ganti text/judul form
    this.Text = title;

    isNewData = false; // set status edit data
    mhs = obj; // set objek mhs yang akan diedit

    // untuk edit data, tampilkan data lama
    txtNpm.Text = mhs.Npm;
    txtNama.Text = mhs>Nama;
    txtAngkatan.Text = mhs.Angkatan;
}
```

Constructor adalah method khusus yang mempunyai nama yang sama dengan nama classnya dan dipanggil secara otomatis ketika sebuah objek dibuat. Jadi ketika objek FrmEntryMahasiswa dibuat, maka secara otomatis semua kode yang ada di blok constructor tersebut akan dieksekusi.

- Langkah berikutnya kita akan melengkapi kode untuk tombol Simpan, caranya adalah dengan mengaktifkan event handler *btnSimpan_Click*.

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

Kemudian lengkapi kodenya seperti berikut:

Caranya dengan mengaktifkan editor code, kemudian tambahkan method InisialisasiListView berikut di bawah constructor.

```
// atur kolom listview
private void InisialisasiListView()
{
    lvwMahasiswa.View = View.Details;
    lvwMahasiswa.FullRowSelect = true;
    lvwMahasiswa.GridLines = true;

    lvwMahasiswa.Columns.Add("No.", 30, HorizontalAlignment.Center);
    lvwMahasiswa.Columns.Add("Npm", 91, HorizontalAlignment.Center);
    lvwMahasiswa.Columns.Add("Nama", 300, HorizontalAlignment.Left);
    lvwMahasiswa.Columns.Add("Angkatan", 80, HorizontalAlignment.Center);
}
```

Untuk petunjuknya lihat gambar berikut:

```
public partial class FrmMahasiswa : Form
{
    // constructor
    public FrmMahasiswa()
    {
        InitializeComponent();
    }

    // atur kolom listview
    private void InisialisasiListView()
    {
        lvwMahasiswa.View = View.Details;
        lvwMahasiswa.FullRowSelect = true;
        lvwMahasiswa.GridLines = true;
    }
}
```

Setelah itu panggil method InisialisasiListView dari dalam constructor.

```
namespace AkademikApp
{
    public partial class FrmMahasiswa : Form
    {
        // constructor
        public FrmMahasiswa()
        {
            InitializeComponent();
            InisialisasiListView();
        }

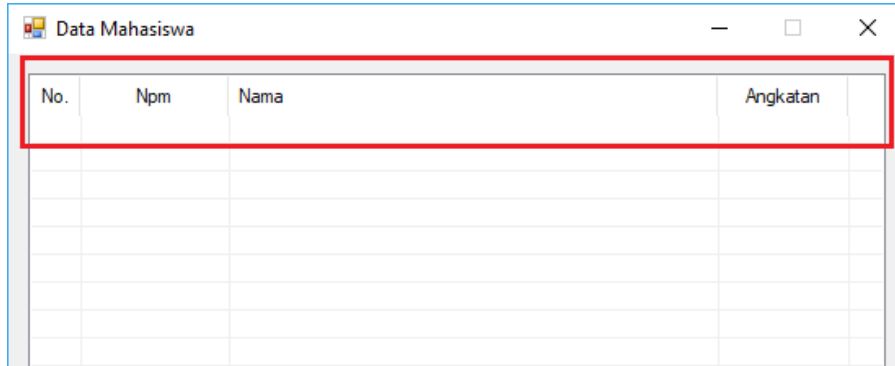
        // atur kolom listview
        private void InisialisasiListView()
        {

```

Constructor adalah method khusus yang mempunyai nama yang sama dengan nama classnya dan dipanggil secara otomatis ketika sebuah objek dibuat. Jadi

ketika objek FrmMahasiswa dibuat, secara otomatis method InisialisasiListView akan dipanggil untuk memformat tampilan ListView.

2. Kemudian jalankan aplikasi untuk melihat hasil sementara dengan menekan tombol F5 (Start Debugging)



3. Kembali lagi ke editor kode, kemudian tambahkan kode berikut untuk mendeklarasikan field dengan tipe collection yang khusus untuk menyimpan objek dari class Mahasiswa.

```
public partial class FrmMahasiswa : Form
{
    // deklarasi collection untuk menampung objek mahasiswa
    private IList<Mahasiswa> listOfMahasiswa = new List<Mahasiswa>();

    // constructor
    public FrmMahasiswa()
    {
        InitializeComponent();
    }
}
```

4. Langkah berikutnya kita akan menambahkan dua buah event handler yang berfungsi untuk merespon event *OnCreate* dan *OnUpdate* ketika dipanggil dari form entry mahasiswa.

```
// method event handler untuk merespon event OnCreate,  
// ketika di panggil dari form entry mahasiswa  
private void FrmEntry_OnCreate(Mahasiswa mhs)  
{  
    // tambahkan objek mhs yang baru ke dalam collection  
    listOfMahasiswa.Add(mhs);  
  
    int noUrut = lvwMahasiswa.Items.Count + 1;  
  
    // tampilkan data mhs yg baru ke list view  
    ListViewItem item = new ListViewItem(noUrut.ToString());  
    item.SubItems.Add(mhs.Npm);  
    item.SubItems.Add(mhs>Nama);  
    item.SubItems.Add(mhs.Angkatan);  
  
    lvwMahasiswa.Items.Add(item);  
}  
  
// method event handler untuk merespon event OnUpdate,  
// ketika di panggil dari form entry mahasiswa  
private void FrmEntry_OnUpdate(Mahasiswa mhs)  
{  
    // ambil baris data mhs yang edit  
    int row = lvwMahasiswa.SelectedIndices[0];  
  
    // update informasi mhs di listview  
    ListViewItem itemRow = lvwMahasiswa.Items[row];  
    itemRow.SubItems[1].Text = mhs.Npm;  
    itemRow.SubItems[2].Text = mhs>Nama;  
    itemRow.SubItems[3].Text = mhs.Angkatan;  
}
```

Kode di atas, ditambahkan setelah method InisialisasiListView.

- Setelah menambahkan event handler *OnCreate* dan *OnUpdate*, langkah berikutnya adalah melengkapi kode untuk tombol Tambah. Adapun kodenya seperti berikut:

```
private void btnTambah_Click(object sender, EventArgs e)  
{  
    // buat objek form entry data mahasiswa  
    FrmEntryMahasiswa frmEntry = new FrmEntryMahasiswa("Tambah Data  
Mahasiswa");  
  
    // mendaftarkan method event handler utk merespon event OnCreate  
    (subscribe)  
    frmEntry.OnCreate += FrmEntry_OnCreate;  
  
    // tampilkan form entry mahasiswa  
    frmEntry.ShowDialog();  
}
```

6. Kemudian lengkapi juga kode untuk tombol Perbaiki seperti berikut:

```
private void btnPerbaiki_Click(object sender, EventArgs e)
{
    if (lvwMahasiswa.SelectedItems.Count > 0)
    {
        // ambil objek mhs yang mau diedit dari collection
        Mahasiswa mhs = listOfMahasiswa[lvwMahasiswa.SelectedIndices[0]];

        // buat objek form entry data mahasiswa
        FrmEntryMahasiswa frmEntry = new FrmEntryMahasiswa("Edit Data
Mahasiswa", mhs);

        // mendaftarkan method event handler utk merespon event OnUpdate
        (subscribe)
        frmEntry.OnUpdate += FrmEntry_OnUpdate;

        // tampilkan form entry mahasiswa
        frmEntry.ShowDialog();
    }
    else // data belum dipilih
    {
        MessageBox.Show("Data belum dipilih", "Peringatan",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
}
```

7. Terakhir lengkapi juga kode untuk tombol Hapus dan Selesai.

```
private void btnHapus_Click(object sender, EventArgs e)
{
    if (lvwMahasiswa.SelectedItems.Count > 0)
    {
        // ambil objek mhs yang mau dihapus dari collection
        Mahasiswa obj = listOfMahasiswa[lvwMahasiswa.SelectedIndices[0]];

        string msg = string.Format("Apakah data mahasiswa '{0}' ingin
dihapus ?", obj>Nama);

        if (MessageBox.Show(msg, "Konfirmasi", MessageBoxButtons.YesNo,
            MessageBoxIcon.Exclamation) == DialogResult.Yes)
        {
            // hapus objek mahasiswa dari collection
            listOfMahasiswa.Remove(obj);

            lvwMahasiswa.Items.Clear();

            // refresh data mhs yang ditampilkan ke listview
            foreach (Mahasiswa mhs in listOfMahasiswa)
            {
                int noUrut = lvwMahasiswa.Items.Count + 1;

                ListViewItem item = new ListViewItem(noUrut.ToString());
                item.SubItems.Add(mhs.Npm);
                item.SubItems.Add(mhs>Nama);
                item.SubItems.Add(mhs.Angkatan);

                lvwMahasiswa.Items.Add(item);
            }
        }
        else // data belum dipilih
        {
            MessageBox.Show("Data belum dipilih", "Peringatan",
                MessageBoxButtons.OK,
                MessageBoxIcon.Exclamation);
        }
    }
}
```

```
private void btnSelesai_Click(object sender, EventArgs e)
{
    this.Close();
}
```

8. Setelah menyelesaikan kode program, jalankan aplikasi dengan menekan tombol F5.

Selesai 😊

Kamarudin, M.Kom
<http://coding4ever.net/>
<https://github.com/rudi-krsoftware/open-retail>