

Setelah melaksanakan praktikum ini diharapkan mahasiswa dapat :

- ✓ Menggunakan class collection yang berfungsi sebagai container untuk menampung banyak objek.
- ✓ Mengetahui cara menambah dan menghapus objek dari dalam collection.

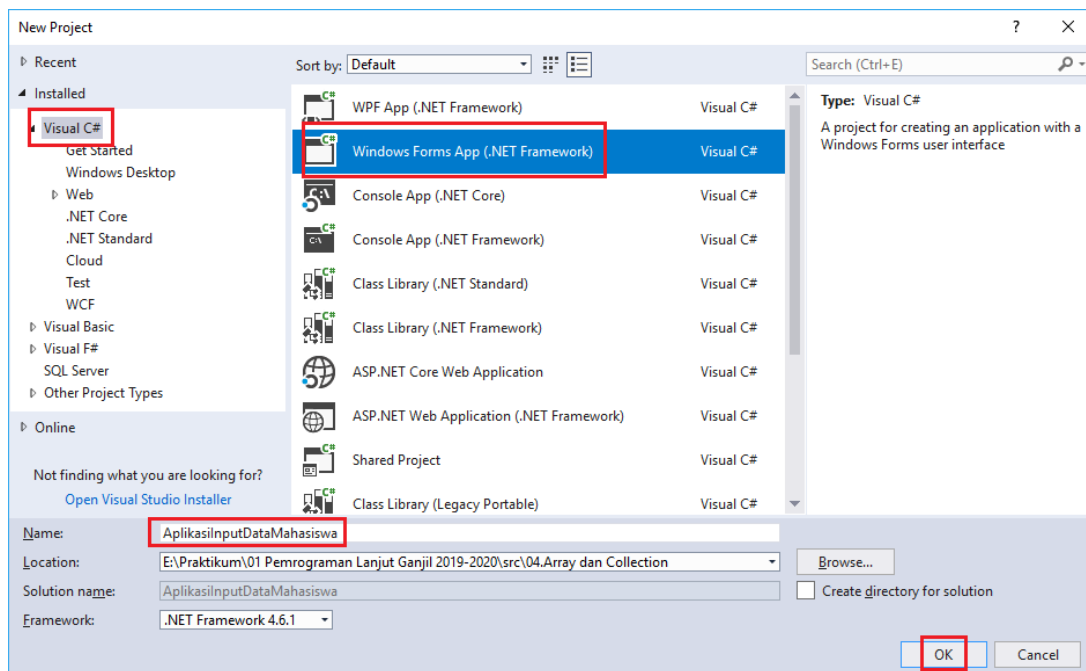
## Teori Singkat

Membuat aplikasi dengan Visual Studio .NET sangat sederhana. Hanya dengan 4 langkah dasar, akan didapat aplikasi yang selanjutnya dapat dijalankan tanpa memerlukan lingkungan pemrograman Visual Studio .NET. 4 langkah dasar tersebut adalah :

- Membuat project menggunakan Visual Studio .NET
- Membuat antarmuka pengguna (user interface) dari aplikasi
- Menuliskan kode program (source code) dari aplikasi yang akan membuat aplikasi menjalankan apa yang seharusnya dikerjakan.
- Mengkompilasi aplikasi menjadi file yang executable sehingga dapat dijalankan sebagai aplikasi yang berdiri sendiri (standalone) di luar IDE Visual Studio .NET

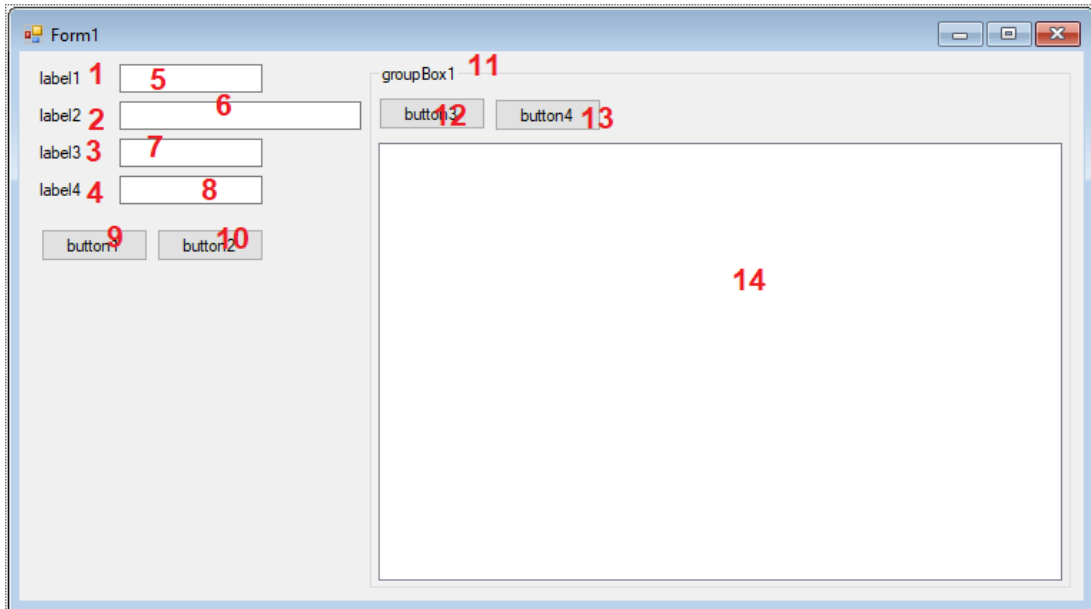
## Latihan 4.1

1. Awali selalu pekerjaan dgn doa, mudah-mudahan diberi kemudahan dan dapat memberikan manfaat
2. Jalankan aplikasi Microsoft Visual Studio .NET
3. Buat Project baru dengan: Klik File -> New -> Project



Isian Name diisi dengan **AplikasiInputDataMahasiswa**. 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. Kemudian rancang tampilan aplikasi seperti gambar di bawah ini:

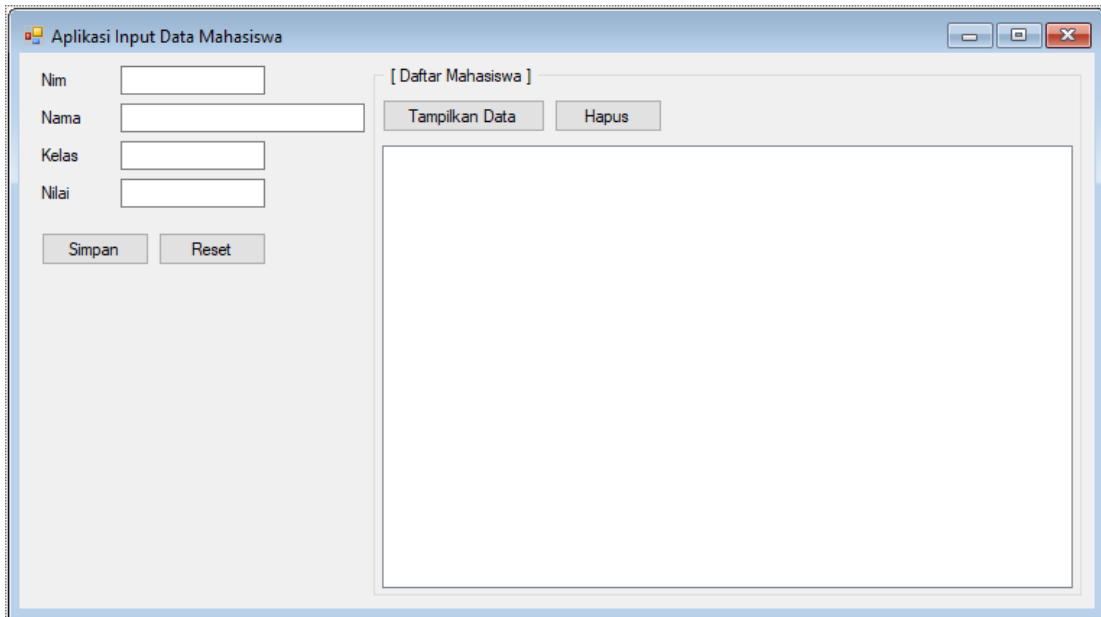


5. Kemudian atur properties masing-masing komponen melalui properties window dengan aturan sesuai tabel 4.1

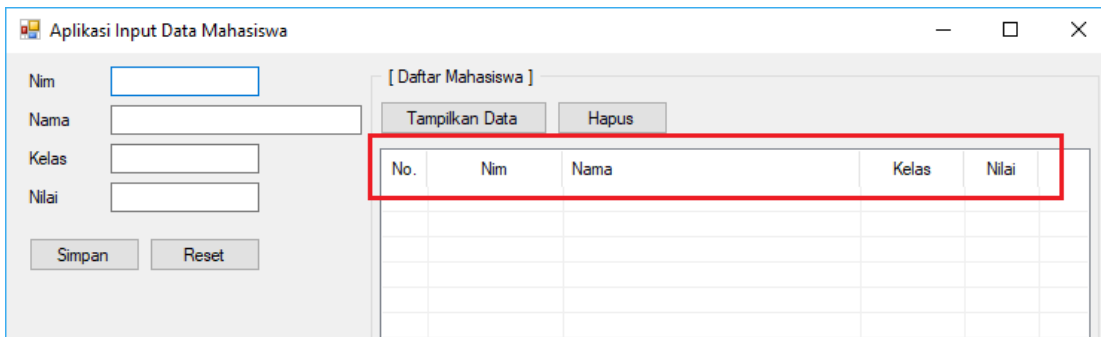
Tabel 4.1

Komponen	Properties	Nilai
Form	Text	Aplikasi Input Data Mahasiswa
	StartPosition	CenterScreen
1	Text	Nim
2	Text	Nama
3	Text	Kelas
4	Text	Nilai
5	Name	txtNim
6	Name	txtNama
7	Name	txtKelas
8	Name	txtNilai
9	Name	btnSimpan
	Text	Simpan
10	Name	btnReset
	Text	Reset
11 (GroupBox)	Text	[ Daftar Mahasiswa ]
12	Name	btnTampilkanData
	Text	Tampilkan Data
13	Name	btnHapus
	Text	Hapus
14 (ListView)	Name	lvwMahasiswa

6. Setelah pengaturan properties akan didapat tampilan sbb :



7. Hal pertama yang kita lakukan adalah mengatur property ListView agar mempunyai tampilan 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("Nim", 91, HorizontalAlignment.Center);
    lvwMahasiswa.Columns.Add("Nama", 200, HorizontalAlignment.Left);
    lvwMahasiswa.Columns.Add("Kelas", 70, HorizontalAlignment.Center);
    lvwMahasiswa.Columns.Add("Nilai", 50, HorizontalAlignment.Center);
}
```

Untuk petunjuknya lihat gambar berikut:

```
public partial class Form1 : Form
{
    // constructor
    public Form1()
    {
        InitializeComponent();
    }

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

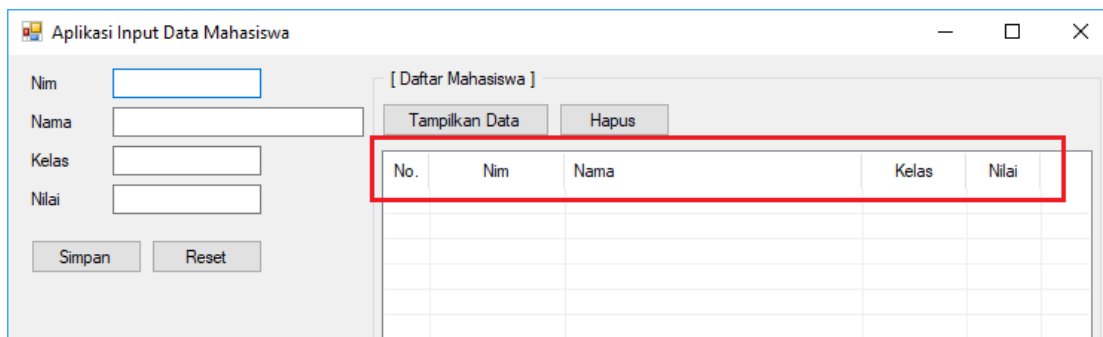
Setelah itu panggil method *InisialisasiListView* dari dalam constructor.

```
public partial class Form1 : Form
{
    // constructor
    public Form1()
    {
        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 Form1 dibuat secara otomatis method *InisialisasiListView* akan dipanggil untuk memformat tampilan ListView.

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



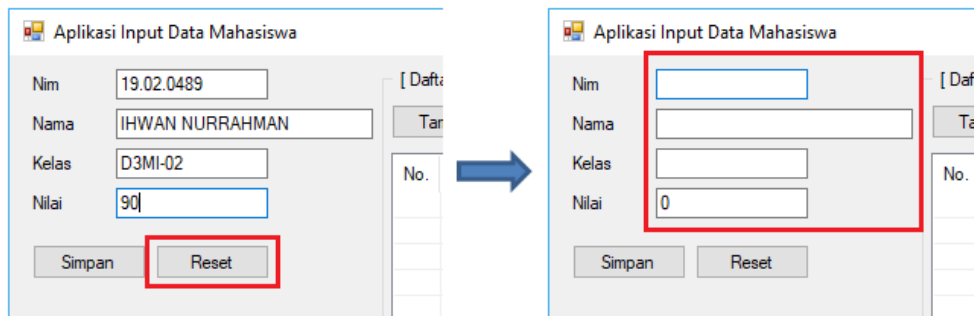
9. Kembali lagi ke editor code, kemudian kita tambahkan sebuah method baru dengan nama *ResetForm*. Lokasinya di bawah method *InisialisasiListView*. Method ini berfungsi untuk mengosongkan inputan data mahasiswa.

```
private void ResetForm()
{
    txtNim.Clear();
    txtNama.Clear();
    txtKelas.Clear();
    txtNilai.Text = "0";
    txtNim.Focus();
}
```

Kemudian panggil method tersebut di dalam event btnReset\_Click.

```
private void btnReset_Click(object sender, EventArgs e)
{
    ResetForm();
}
```

10. Jalankan kembali aplikasi untuk mengetes apakah tombol Resetnya, sudah berfungsi sesuai dengan yang diinginkan.



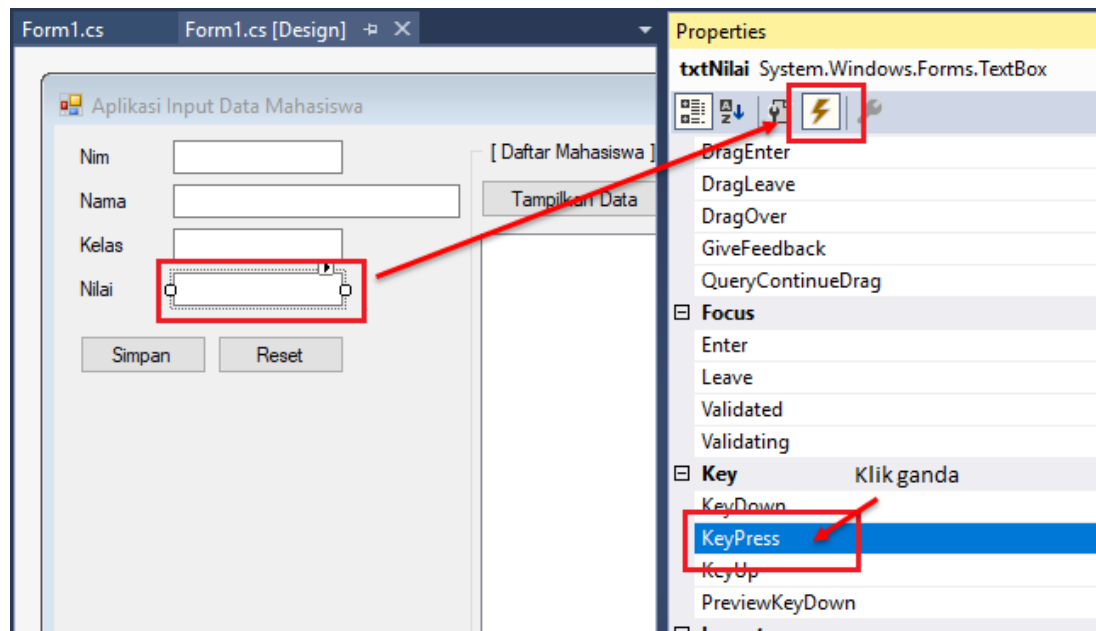
11. Berikutnya kita akan menambahkan sebuah method baru untuk memvalidasi inputan nilai. Di mana inputan yang diperbolehkan hanya berupa angka. Aktifkan editor code kemudian ketik atau copy kode berikut:

```
private bool NumericOnly(KeyPressEventArgs e)
{
    var strValid = "0123456789";

    if (!(e.KeyChar == Convert.ToChar(Keys.Back)))
    {
        // inputan selain angka
        if (strValid.IndexOf(e.KeyChar) < 0)
        {
            return true;
        }

        return false;
    }
    else
        return false;
}
```

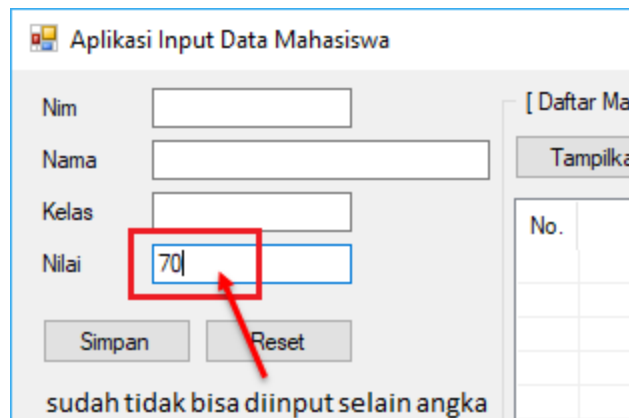
Setelah itu method `NumericOnly` di atas akan dipanggil di dalam event `txtNilai_KeyPress`, tapi sebelum itu kita harus mengaktifkan terlebih dulu event `txtNilai_KeyPress` dengan cara seperti gambar berikut:



Kemudian lengkapi kodenya seperti berikut:

```
private void txtNilai_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = NumericOnly(e);
}
```

12. Jalankan kembali aplikasi untuk mengetes apakah fungsi validasi angkanya sudah berfungsi sesuai dengan yang diinginkan atau belum.



13. Kembali ke editor code, selanjutnya kita akan menambahkan kode untuk tombol Simpan. Caranya Anda aktifkan terlebih dulu event `btnSimpan_Click` kemudian lengkapi kodenya seperti berikut:

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // membuat objek mahasiswa
    Mahasiswa mhs = new Mahasiswa();

    // set nilai masing-masing propertynya
    // berdasarkan inputan yang ada di form
    mhs.Nim = txtNim.Text;
    mhs>Nama = txtNama.Text;
    mhs.Kelas = txtKelas.Text;
    mhs.Nilai = int.Parse(txtNilai.Text);

    // tambahkan objek mahasiswa ke dalam collection
    list.Add(mhs);

    var msg = "Data mahasiswa berhasil disimpan.";

    // tampilkan dialog informasi
    MessageBox.Show(msg, "Informasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information);

    // reset form input
    ResetForm();
}
```

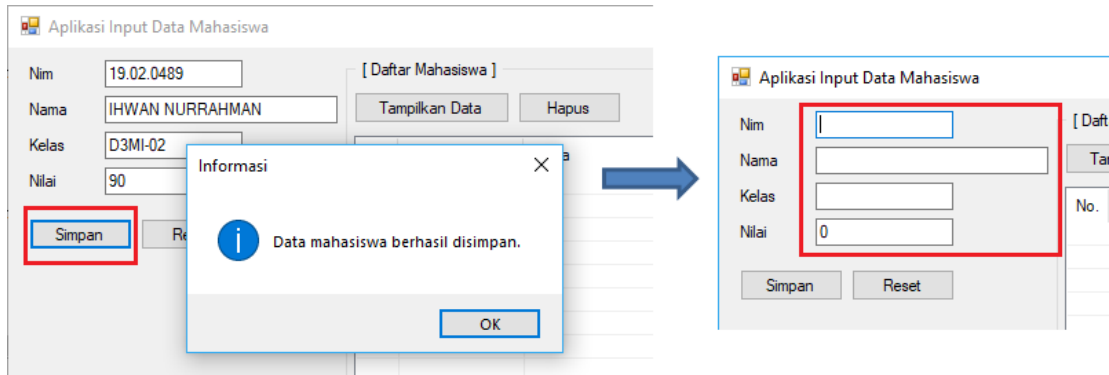
Pada kode di atas terdapat perintah `list.Add(mhs)` yang berfungsi untuk menambahkan objek mahasiswa ke dalam collection. Agar perintah ini tidak error, kita harus menambahkan deklarasi objek list tersebut di bagian atas constructor.

```
public partial class Form1 : Form
{
    // deklarasi objek collection
    private List<Mahasiswa> list = new List<Mahasiswa>();

    // constructor
    public Form1()
    {
        InitializeComponent();
    }
}
```

Selain itu Anda juga harus menambahkan sendiri class Mahasiswa yang mempunyai 4 buah property yaitu Nim, Nama, Kelas dan Nilai.

14. Jalankan kembali aplikasi untuk mengetes tombol simpannya.



Sampai di sini, data mahasiswa yang sudah disimpan datanya tidak langsung tampil datanya ke dalam ListView karena kita memang sengaja memisahkan prosesnya antara menambahkan data/objek ke dalam collection dan mengambil data dari dalam collection.

15. Berikutnya kita akan menambahkan kode untuk menampilkan data. Caranya aktifkan terlebih dulu editor code, kemudian tambahkan method berikut ini:

```
private void TampilkanData()
{
    // kosongkan data listview
    lvwMahasiswa.Items.Clear();

    // lakukan perulangan untuk menampilkan data mahasiswa ke listview
    foreach (var mhs in list)
    {
        var noUrut = lvwMahasiswa.Items.Count + 1;

        var item = new ListViewItem(noUrut.ToString());
        item.SubItems.Add(mhs.Nim);
        item.SubItems.Add(mhs>Nama);
        item.SubItems.Add(mhs.Kelas);
        item.SubItems.Add(mhs.Nilai.ToString());

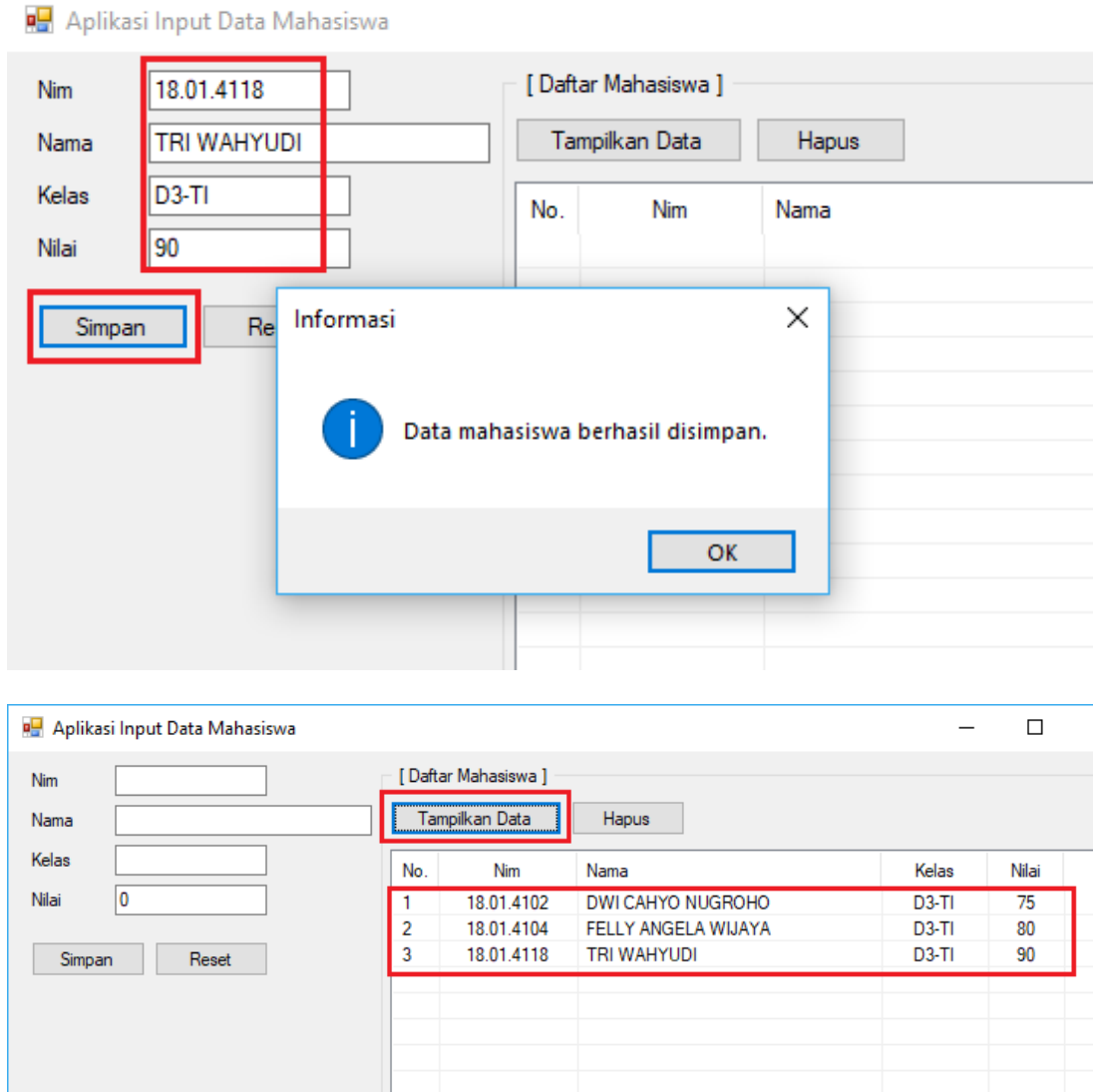
        lvwMahasiswa.Items.Add(item);
    }
}
```

Kemudian panggil method tersebut di dalam event btnTampilkanData\_Click

```
private void btnTampilkanData_Click(object sender, EventArgs e)
{
    TampilkanData();
}
```

16. Jalankan kembali aplikasi untuk mengetes apakah tombol tampilkan data berfungsi dengan benar atau tidak. Caranya adalah dengan menginputkan beberapa data mahasiswa 2 atau 3 orang, setelah itu klik tombol tampilkan data.





17. Terakhir kita akan menambahkan kode untuk event `btnHapus_Click` yang berfungsi untuk menghapus objek mahasiswa dari dalam objek collection. Untuk kode lengkapnya seperti berikut:

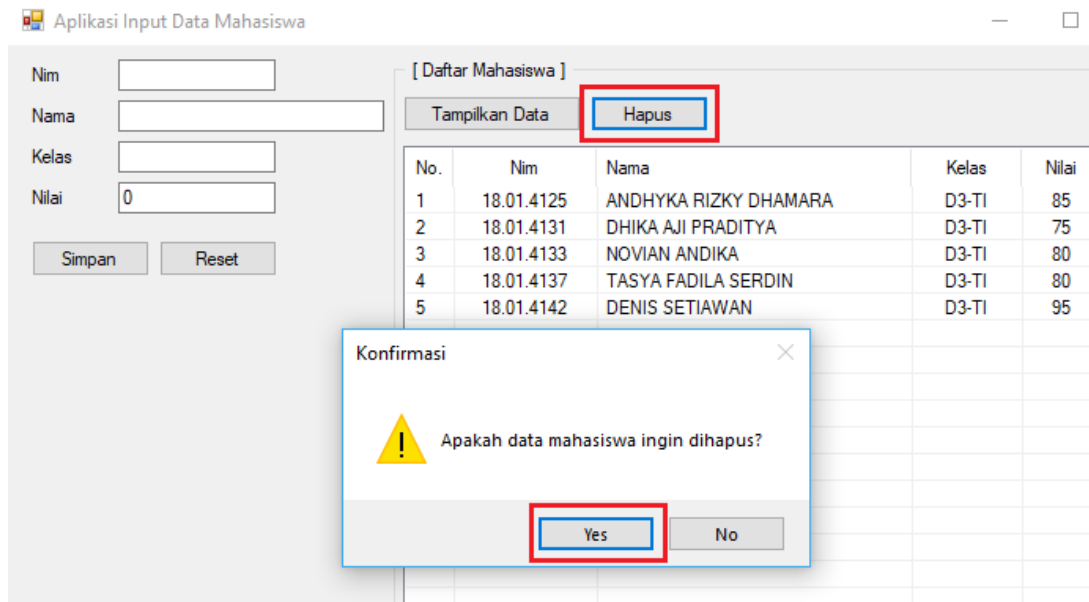
```
private void btnHapus_Click(object sender, EventArgs e)
{
    // cek apakah data mahasiswa sudah dipilih
    if (lvwMahasiswa.SelectedItems.Count > 0)
    {
        // tampilkan konfirmasi
        var konfirmasi = MessageBox.Show("Apakah data mahasiswa ingin dihapus?", "Konfirmasi",
            MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation);

        if (konfirmasi == DialogResult.Yes)
        {
            // ambil index list yang di pilih
            var index = lvwMahasiswa.SelectedIndices[0];

            // hapus objek mahasiswa dari list
            list.RemoveAt(index);

            // refresh tampilan listview
            TampilkanData();
        }
    }
    else // data belum dipilih
    {
        MessageBox.Show("Data mahasiswa belum dipilih !!!", "Peringatan",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

18. Jalankan kembali aplikasi untuk mengetes apakah tombol hapus.

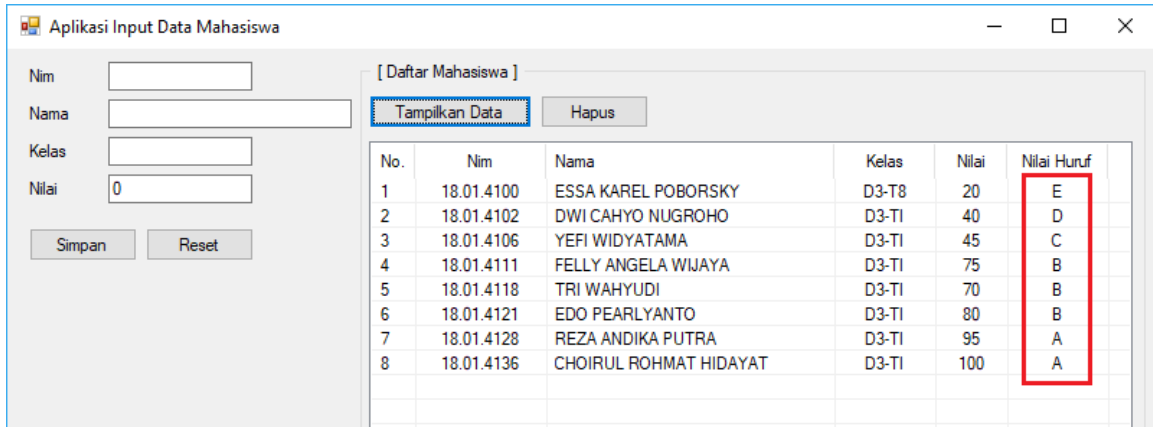


## Tugas 4.1

Revisi latihan di atas dengan menambahkan method baru untuk mendapatkan *nilai huruf* berdasarkan *nilai angka* yang diinputkan.

Adapun logic untuk mendapatkan nilai huruf berdasarkan nilai angka seperti berikut:

- Nilai 0 s.d 20 E
- Nilai 21 s.d 40 D
- Nilai 41 s.d 60 C
- Nilai 61 s.d 80 B
- Nilai 81 s.d 100 A



No.	Nim	Nama	Kelas	Nilai	Nilai Huruf
1	18.01.4100	ESSA KAREL POBORSKY	D3-T8	20	E
2	18.01.4102	DWI CAHYO NUGROHO	D3-TI	40	D
3	18.01.4106	YEFI WIDYATAMA	D3-TI	45	C
4	18.01.4111	FELLY ANGELA WIJAYA	D3-TI	75	B
5	18.01.4118	TRI WAHYUDI	D3-TI	70	B
6	18.01.4121	EDO PEARLYANTO	D3-TI	80	B
7	18.01.4128	REZA ANDIKA PUTRA	D3-TI	95	A
8	18.01.4136	CHOIRUL ROHMAT HIDAYAT	D3-TI	100	A

Selesai 😊

Kamarudin, M.Kom

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