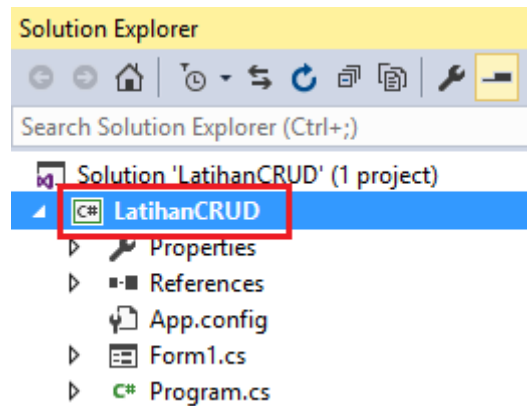


Materi:

- ✓ ADO.NET
- ✓ CRUD

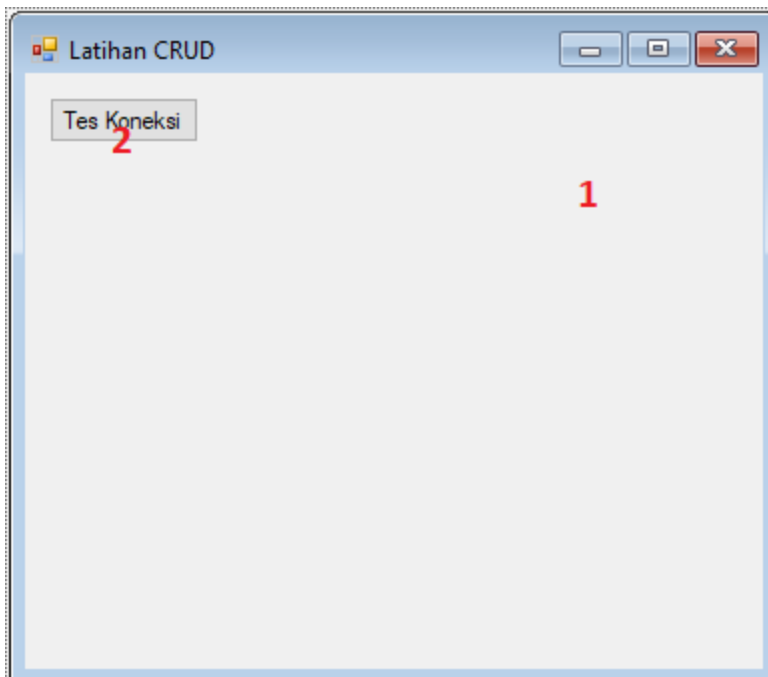
Pada praktikum kali ini, kita akan mempelajari cara menggunakan teknologi akses data ADO.NET untuk membuat operasi CRUD (Create, Read, Update dan Delete) sederhana. Buat project baru Windows Forms Application dengan nama **LatihanCRUD**



Kemudian kerjakan latihan-latihan berikut:

Latihan 1 – Tes Koneksi ke Database

1. Design Form



2. Properties

No	Komponen	Property	Nilai/Value
1	Form1	Text StartPosition	Latihan CRUD CenterScreen
2	Button1	Name Text	btnTesKoneksi Tes Koneksi

3. Source code

Di bagian deklarasi namespace kita tambahkan dua namespace yaitu *System.IO* dan *System.Data.OleDb*. Namespace *System.IO* kita tambahkan agar kita bisa menggunakan method *GetCurrentDirectory* dari class *Directory*. Method ini berfungsi untuk mendapatkan lokasi folder aplikasi secara relative.

Kemudian namespace *System.Data.OleDb* kita tambahkan agar bisa mengakses provider OLEDB untuk koneksi ke database Microsoft Access.

```
using System.Windows.Forms;
using System.IO;
using System.Data.OleDb;

namespace LatihanCRUD
{
    public partial class Form1 : Form
```

Selanjutnya kita akan menambahkan kode pada button Tes Koneksi. Pada saat button Tes Koneksi diklik maka akan menampilkan pesan "Koneksi ke database berhasil !" atau "Koneksi ke database gagal !!!" jika gagal melakukan koneksi ke database. Caranya adalah dengan mengklik ganda button **Tes Koneksi** untuk membuka editor code dan menambahkan *event handler click* untuk merespon ketika ada user mengklik tombol **Tes Koneksi**.

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

}
```

Kemudian lengkapi kodenya seperti berikut:

```
private void btnTesKoneksi_Click(object sender, EventArgs e)
{
    var dbName = Directory.GetCurrentDirectory() + "\\DbPerpustakaan.mdb";

    var connectionString = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source={0}", dbName);

    _conn = GetOpenConnection(connectionString);

    if (_conn.State == ConnectionState.Open)
    {
        MessageBox.Show("Koneksi ke database berhasil !", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Koneksi ke database gagal !!!", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
}
```

Agar kode di atas bisa berjalan, tambahkan juga method *GetOpenConnection* yang berfungsi untuk melakukan koneksi ke database.

```
private OleDbConnection GetOpenConnection(string connectionString)
{
    OleDbConnection conn = null;

    try
    {
        conn = new OleDbConnection(connectionString);
        conn.Open();
    }
    catch
    {
    }

    return conn;
}
```

Selain itu kita juga perlu mendeklarasikan objek *_conn* yang berfungsi untuk handle koneksi ke database.

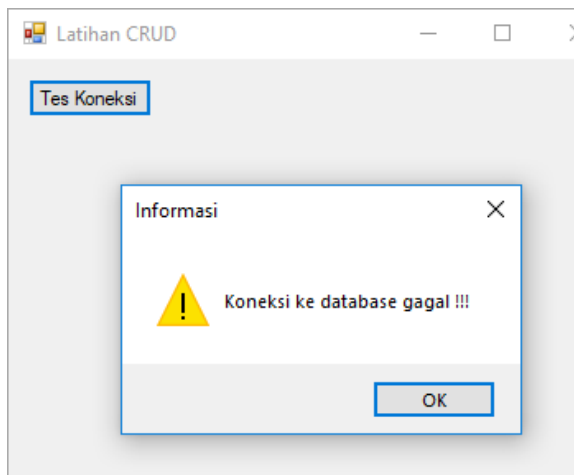
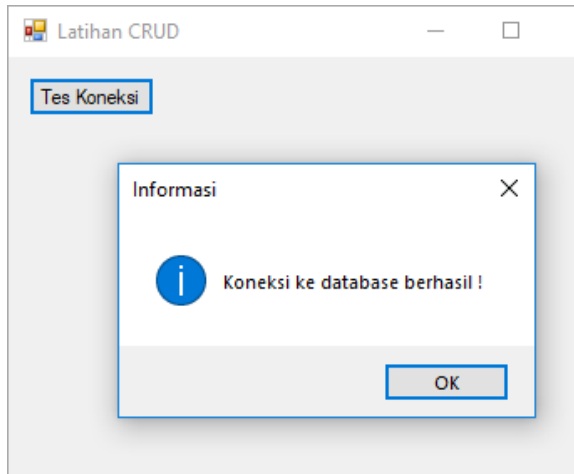
```
using System.IO;
using System.Data.OleDb;

namespace LatihanCRUD
{
    public partial class Form1 : Form
    {
        // deklarasi objek _conn untuk handle
        // koneksi ke database
        private OleDbConnection _conn;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnTesKoneksi_Click(object sender, EventArgs e)
        {
            // Logika koneksi ke database
        }
    }
}
```

4. Output

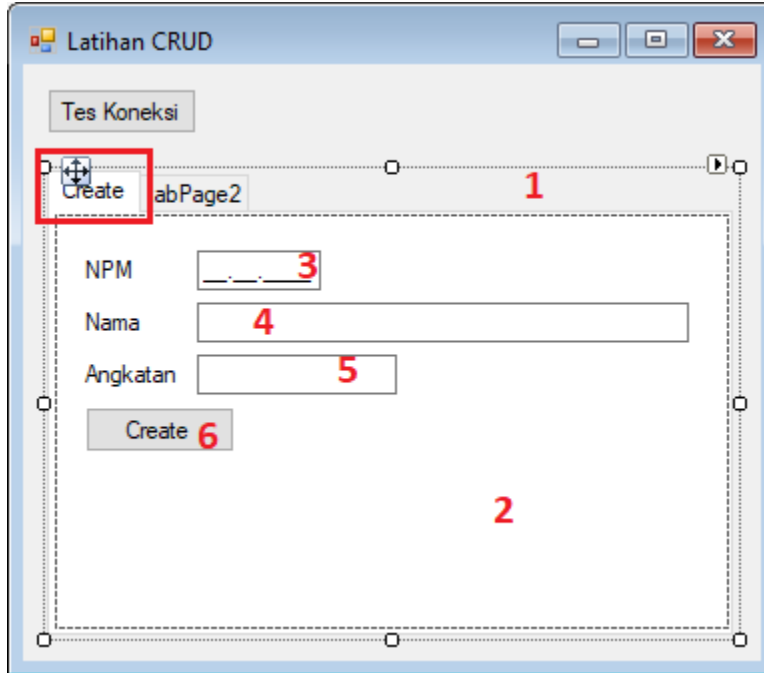


Jika koneksi ke **database** masih gagal, jangan lanjut dulu ke latihan 2 ya :-D

Latihan 2 – Menambahkan Fungsi Create

Lakukan modifikasi desain pada form latihan CRUD yang telah dibuat sebelumnya, untuk menambahkan fungsi create (menambah data baru).

1. Revisi Design Form



2. Properties

No	Komponen	Property	Nilai/Value
1	tabControl1	-	-
2	tabPage1	Text	Create
3	maskedTextBox1	Name Mask	mskNpm1 ##.##.####4
4	textBox1	Name	txtNama1
5	textBox2	Name	txtAngkatan1
6	button1	Name Text	btnCreate Create

3. Source Code

Pada latihan 2 ini kita akan menambahkan kode untuk menambahkan data baru ke tabel mahasiswa, dengan cara mengklik ganda button **Create** untuk membuka editor code dan menambahkan *event handler click* untuk merespon ketika ada user mengklik tombol **Create**.

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

Kemudian lengkapi kodenya seperti berikut:

```
private void btnCreate_Click(object sender, EventArgs e)
{
    var result = 0;

    // validasi npm harus diisi
    if (!mskNpm1.MaskFull)
    {
        MessageBox.Show("NPM harus diisi !!!", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);

        mskNpm1.Focus();
        return;
    }

    // validasi nama harus diisi
    if (txtNama1.Text.Length == 0)
    {
        MessageBox.Show("Nama harus diisi !!!", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);

        txtNama1.Focus();
        return;
    }

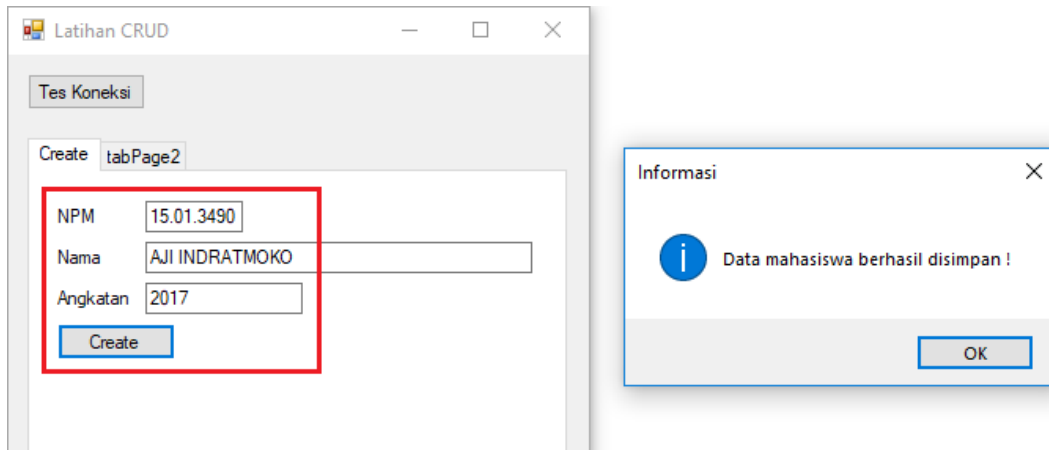
    var sql = @"insert into mahasiswa (npm, nama, angkatan)
                values (@npm, @nama, @angkatan)";
    using (var cmd = new OleDbCommand(sql, _conn))
    {
        cmd.Parameters.AddWithValue("@npm", mskNpm1.Text);
        cmd.Parameters.AddWithValue("@nama", txtNama1.Text);
        cmd.Parameters.AddWithValue("@angkatan", txtAngkatan1.Text);

        result = cmd.ExecuteNonQuery();
    }

    if (result > 0)
    {
        MessageBox.Show("Data mahasiswa berhasil disimpan !", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);

        // reset form
        mskNpm1.Clear();
        txtNama1.Clear();
        txtAngkatan1.Clear();
        mskNpm1.Focus();
    }
    else
        MessageBox.Show("Data mahasiswa gagal disimpan !!!", "Informasi",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
}
```

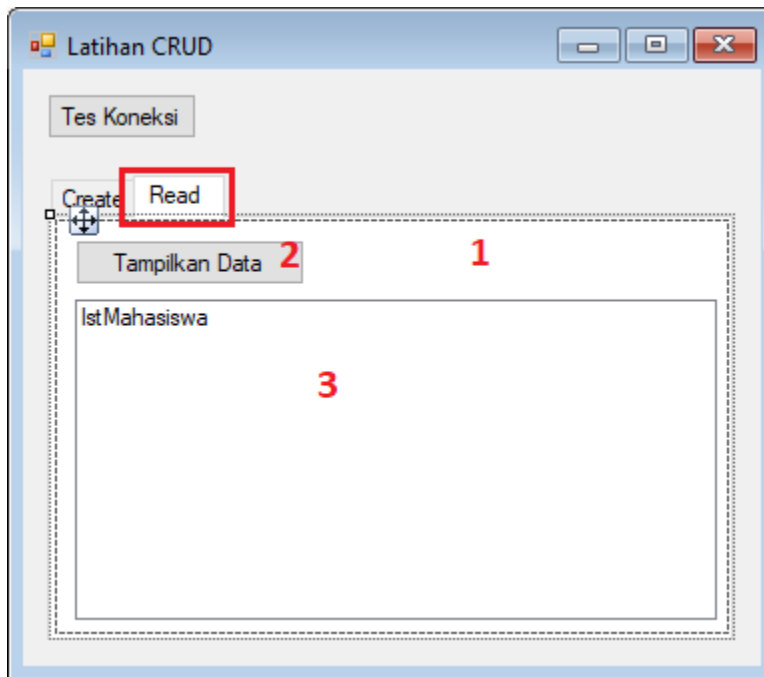
4. Output



Latihan 3 – Menambahkan Fungsi Read

Lakukan modifikasi desain pada form latihan CRUD yang telah dibuat sebelumnya, untuk menambahkan fungsi read (menampilkan data).

1. Revisi Design Form



2. Properties

No	Komponen	Property	Nilai/Value
1	tabPage2	Text	Read
2	button1	Name Text	btnTampilkanData Tampilkan Data
3	listBox1	Name	IstMahasiswa

3. Source Code

Pada latihan 3 ini kita akan menambahkan kode untuk menampilkan data, dengan cara mengklik ganda button **Tampilkan Data** untuk membuka editor code dan menambahkan *event handler click* untuk merespon ketika ada user mengklik tombol **Tampilkan Data**.

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

Kemudian lengkapi kodenya seperti berikut:

```
private void btnTampilkanData_Click(object sender, EventArgs e)
{
    lstMahasiswa.Items.Clear();

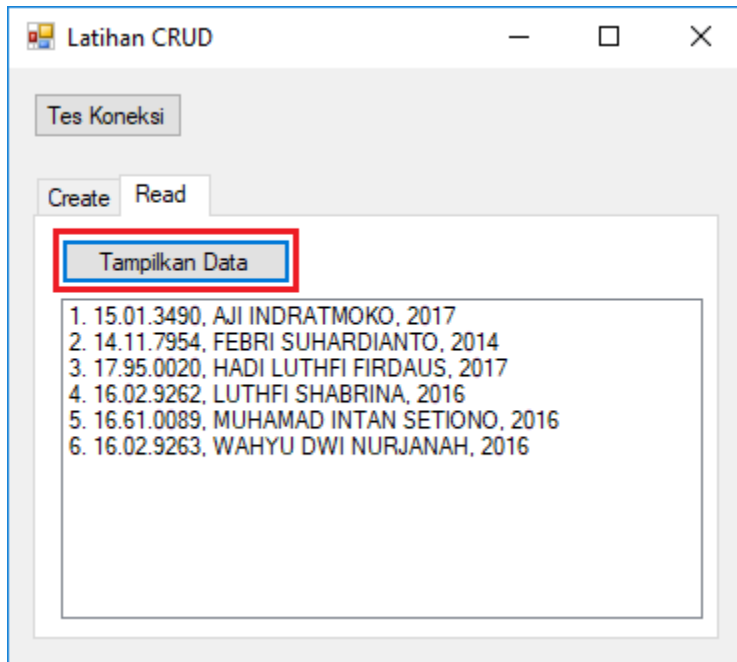
    var sql = @"select npm, nama, angkatan
                from mahasiswa
                order by nama";
    using (var cmd = new OleDbCommand(sql, _conn))
    {
        using (var dtr = cmd.ExecuteReader())
        {
            var noUrut = 1;

            while (dtr.Read())
            {
                var data = string.Format("{0}. {1}, {2}, {3}",
                    noUrut, dtr["npm"].ToString(), dtr["nama"].ToString(),
                    dtr["angkatan"].ToString());

                lstMahasiswa.Items.Add(data);

                noUrut++;
            }
        }
    }
}
```

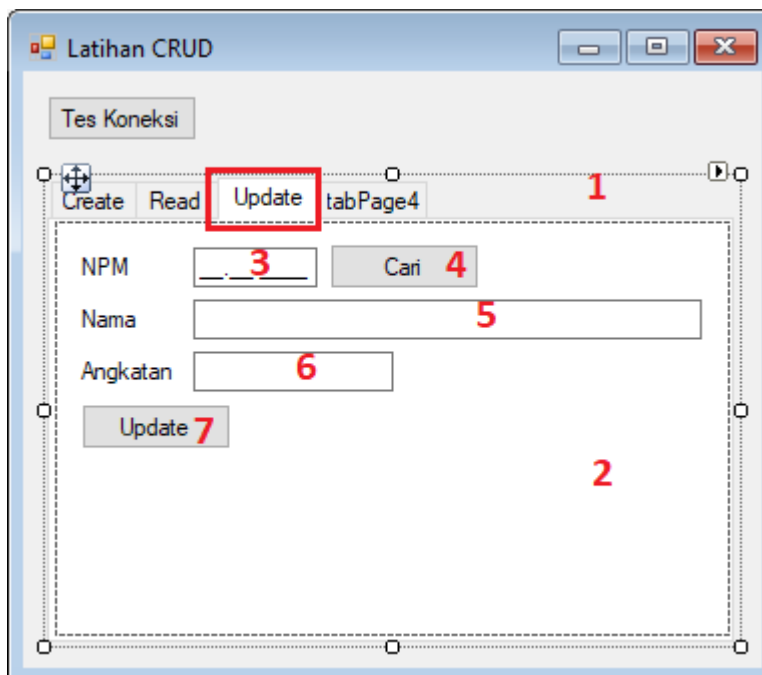

4. Output



Latihan 4 – Menambahkan Fungsi Update

Lakukan modifikasi desain pada form latihan CRUD yang telah dibuat sebelumnya, untuk menambahkan fungsi update (mengupdate data yang sudah ada).

1. Revisi Design Form



2. Properties

No	Komponen	Property	Nilai/Value
1	tabControl1	TabPage	Klik tombol ..., untuk menambahkan 2 tabPage lagi
2	tabPage3	Text	Update
3	maskedTextBox1	Name Mask	mskNpm2 ##.##.####
4	button1	Name Text	btnCari1 Cari
5	textBox1	Name	txtNama2
6	textBox2	Name	txtAngkatan2
7	Button2	Name Text	btnUpdate Update

3. Source Code

Pada latihan 4 ini kita akan menambahkan dua blok kode, yang pertama kode untuk mencari data mahasiswa berdasarkan npm. Caranya dengan mengklik ganda button **Cari** untuk membuka editor code dan menambahkan *event handler click* untuk merespon ketika ada user mengklik tombol **Cari**.

```
private void btnCari1_Click(object sender, EventArgs e)
{
}
}
```

Kemudian lengkapi kodenya seperti berikut:

```
private void btnCari1_Click(object sender, EventArgs e)
{
    var sql = @"select npm, nama, angkatan
                from mahasiswa
                where npm = @npm";
    using (var cmd = new OleDbCommand(sql, _conn))
    {
        cmd.Parameters.AddWithValue("@npm", mskNpm2.Text);

        using (var dtr = cmd.ExecuteReader())
        {
            if (dtr.Read())
            {
                mskNpm2.Text = dtr["npm"].ToString();
                txtNama2.Text = dtr["nama"].ToString();
                txtAngkatan2.Text = dtr["angkatan"].ToString();
            }
            else
            {
                MessageBox.Show("Data mahasiswa tidak ditemukan !",
                                "Informasi", MessageBoxButtons.OK,
                                MessageBoxIcon.Information);
            }
        }
    }
}
```

Kode kedua yang akan kita tambahkan yaitu kode untuk Update. Caranya sama dengan mengklik ganda button **Update** untuk membuka editor code dan menambahkan *event handler click* untuk merespon ketika ada user mengklik tombol **Update**.

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

Kemudian lengkapi kodenya seperti berikut:

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    var result = 0;

    // validasi npm harus diisi
    if (!mskNpm2.MaskFull)
    {
        MessageBox.Show("NPM harus !!!", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);

        mskNpm2.Focus();
        return;
    }

    // validasi nama harus diisi
    if (txtNama2.Text.Length == 0)
    {
        MessageBox.Show("Nama harus !!!", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);

        txtNama2.Focus();
        return;
    }

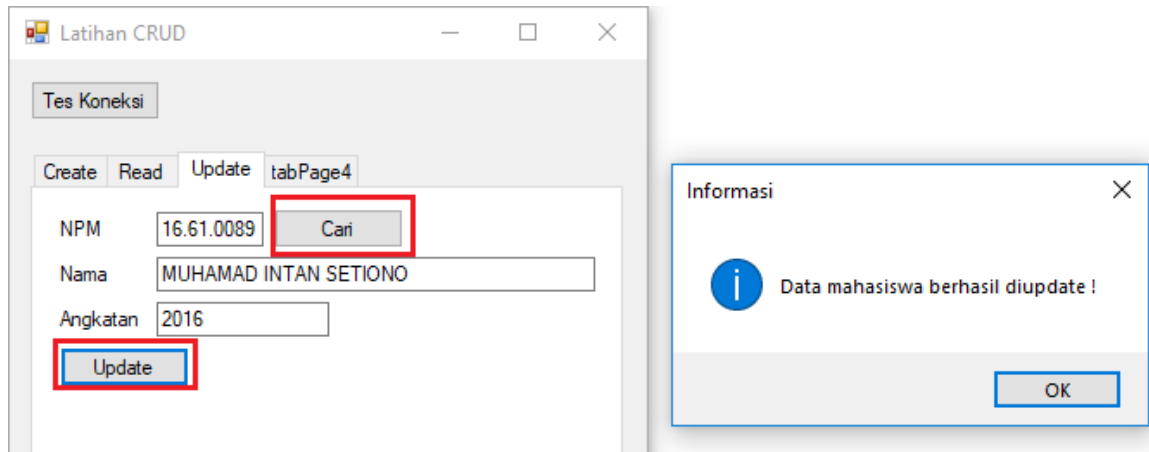
    var sql = @"update mahasiswa set nama = @nama, angkatan = @angkatan
                where npm = @npm";
    using (var cmd = new OleDbCommand(sql, _conn))
    {
        cmd.Parameters.AddWithValue("@nama", txtNama2.Text);
        cmd.Parameters.AddWithValue("@angkatan", txtAngkatan2.Text);
        cmd.Parameters.AddWithValue("@npm", mskNpm2.Text);

        result = cmd.ExecuteNonQuery();
    }

    if (result > 0)
    {
        MessageBox.Show("Data mahasiswa berhasil diupdate !", "Informasi",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);

        // reset form
        mskNpm2.Clear();
        txtNama2.Clear();
        txtAngkatan2.Clear();
        mskNpm2.Focus();
    }
    else
    {
        MessageBox.Show("Data mahasiswa gagal diupdate !!!", "Informasi",
            MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
}
```

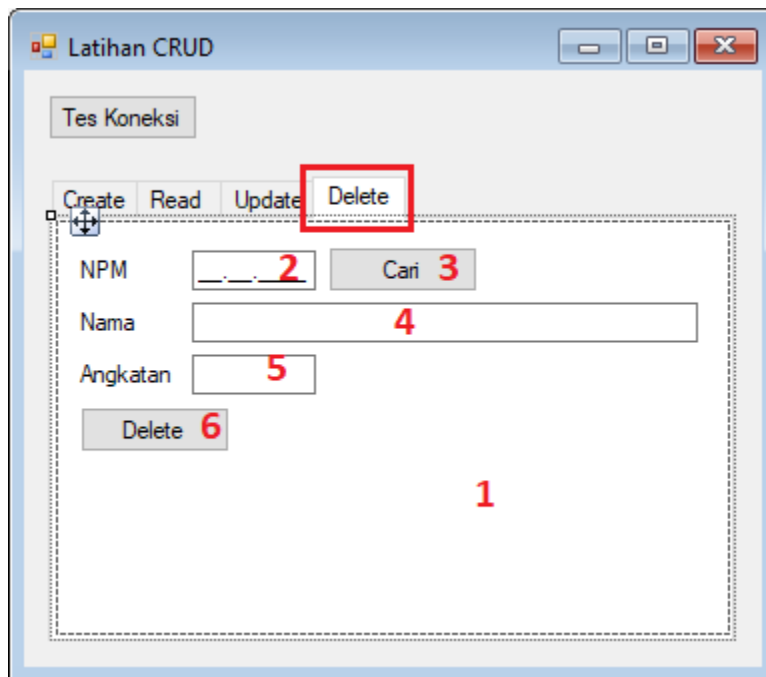
4. Output



Latihan 5 – Menambahkan Fungsi Delete

Lakukan modifikasi desain pada form latihan CRUD yang telah dibuat sebelumnya, untuk menambahkan fungsi delete (menghapus data).

1. Revisi Design Form



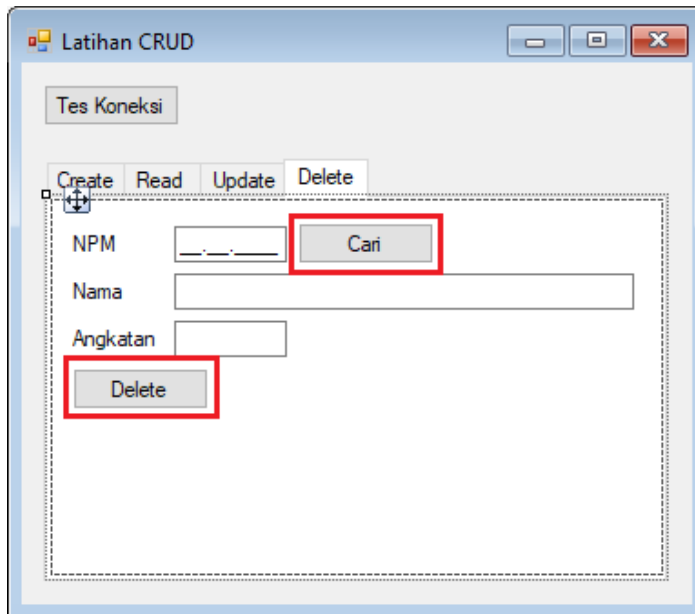
2. Properties

No	Komponen	Property	Nilai/Value
1	tabPage4	Text	Delete
2	maskedTextBox1	Name Mask	mskNpm3 ##.##.####
3	button1	Name Text	btnCari2 Cari

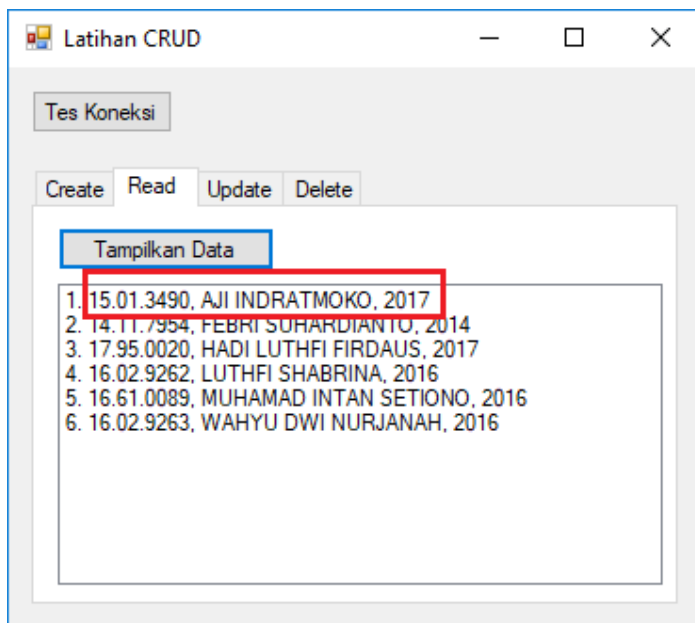
4	textBox1	Name	txtNama3
5	textBox2	Name	txtAngkatan3
6	Button2	Name Text	btnHapus Delete

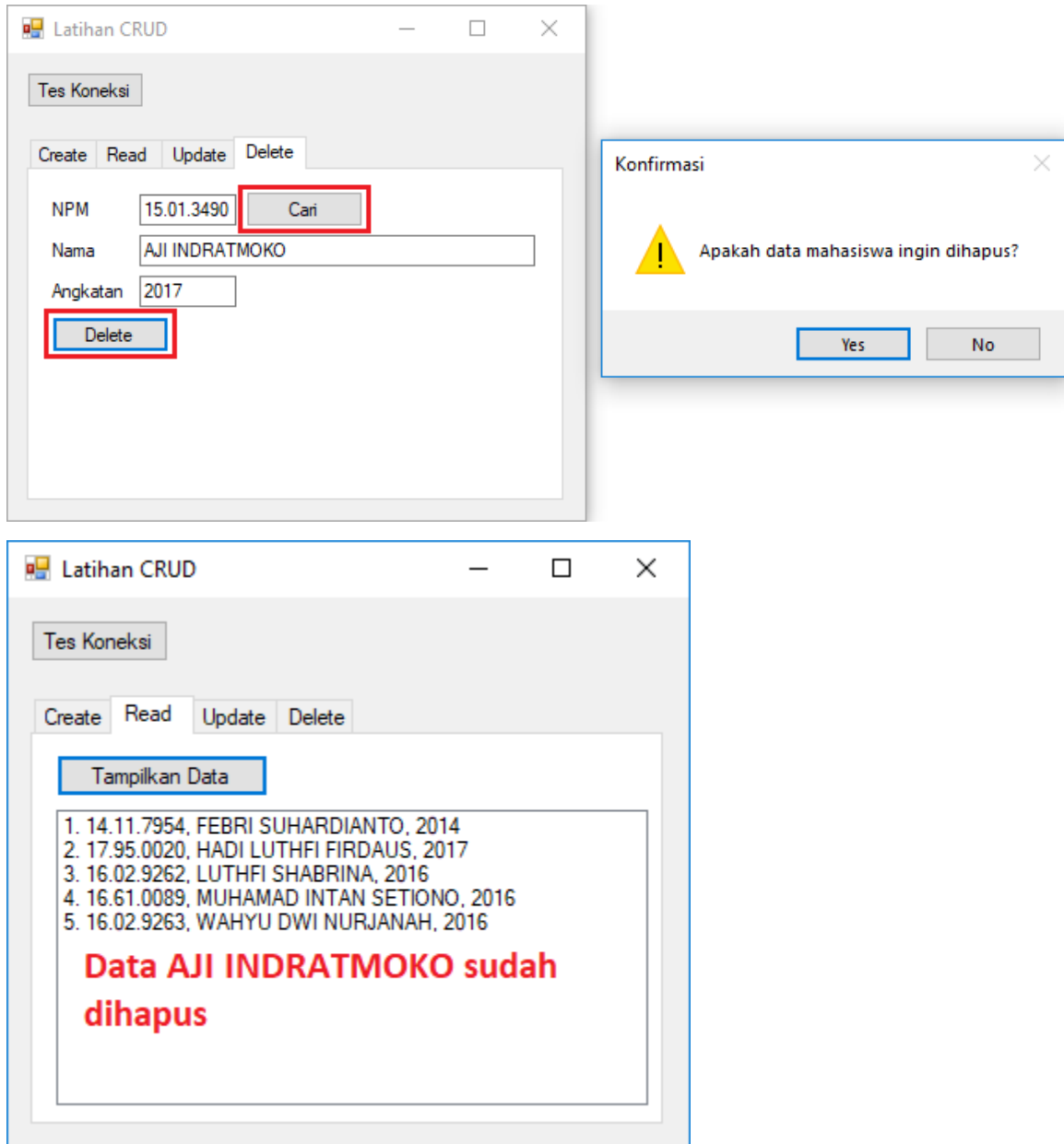
3. Source Code

Pada latihan 5 ini tugas Anda adalah melengkapi blok kode untuk pencarian (button cari) dan hapus data (button delete).



4. Output





Selesai ☺

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