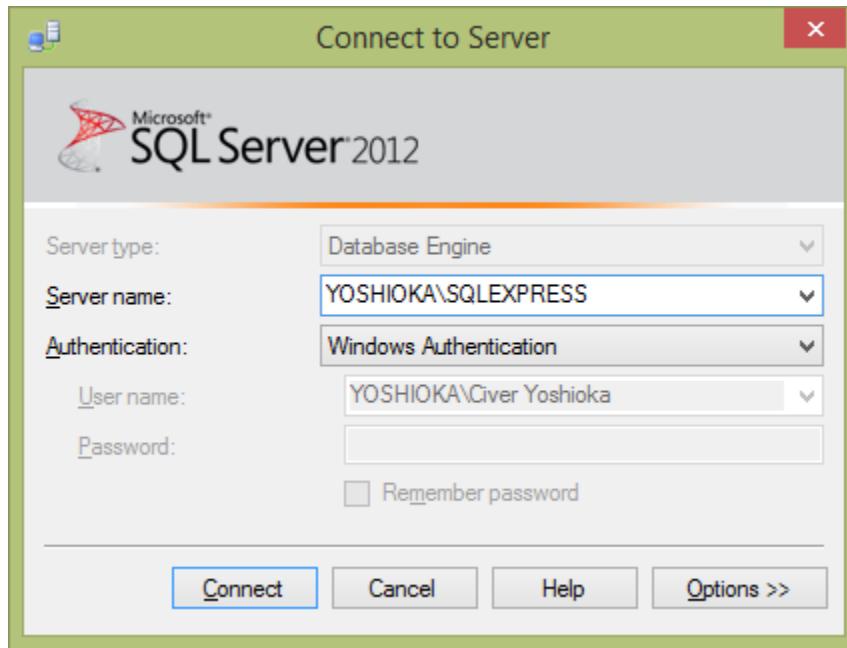


## ENTITY FRAMEWORK

Untuk melakukan manipulasi data pada database menggunakan **ADO.Net** adalah sebagai berikut.

1. Buka pada **SQL Server** pada komputer dan lakukan koneksi sesuai dengan settingan yang telah dibuat.



Pada contoh gambar di atas *nama Server Database Engine* adalah **YOSHIOKA\SQLEXPRESS**. **YOSHIOKA** adalah *nama computer* yang digunakan untuk instalasi, sedangkan **SQLEXPRESS** adalah *nama server* dari SQL server. Untuk memberikan nama pada server bisa disesuaikan pada saat Instalasi SQL server, pada contoh di atas adalah penamaan secara default dari program. Sedangkan untuk **Authentication**, pada contoh di atas adalah menggunakan *Windows Authentication*(autentifikasi dengan menggunakan username dan password dari komputer yang digunakan untuk instalasi), selain menggunakan Windows Authentication bisa menggunakan autentifikasi dengan melakukan setting username dan password pada SQL server pada saat instalasi. Karena untuk memudahkan cara autentifikasi maka pada modul ini menggunakan Windows Authentication, seharusnya untuk keamanan harus melakukan setting username dan password sendiri.

## 2. Membuat Database pada SQL Server

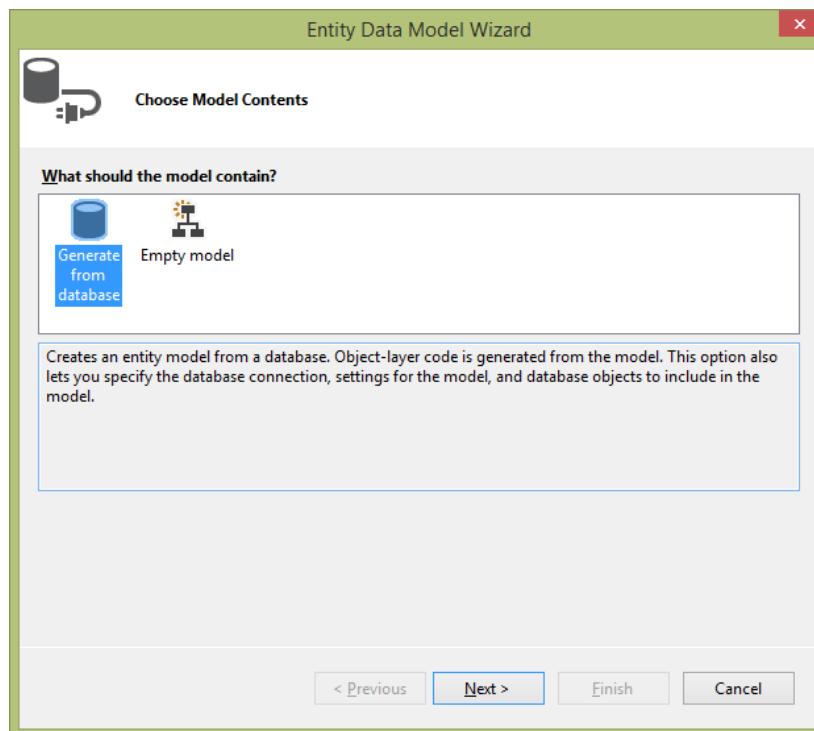
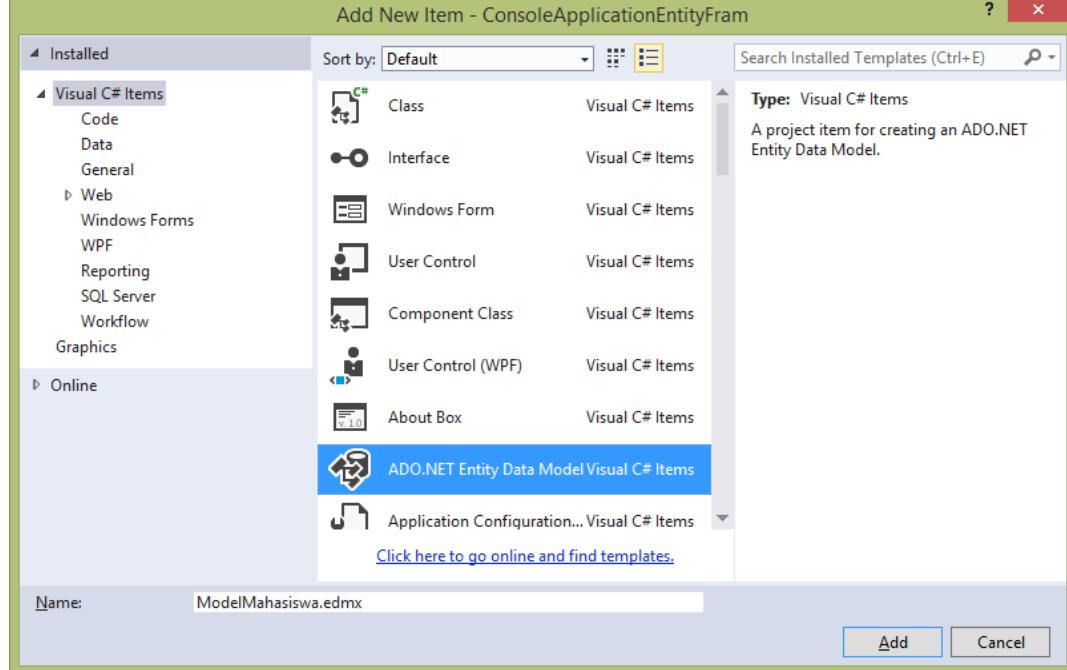
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "YOSHIOKA\SQLEXPRESS.db\_testing - dbo.Table\_1YOSHIOKA\SQLEXPRESS". The left pane, titled "Object Explorer", shows a tree structure of database objects. Under "YOSHIOKA\SQLEXPRESS (SQL Server 11)", there are "Databases" (including "System Databases", "my\_db", and "db\_testing"), "Tables" (including "System Tables", "FileTables", and "dbo.Mahasiswa"), and other object types like "Columns", "Keys", "Constraints", "Triggers", "Indexes", and "Statistics". The "dbo.Mahasiswa" table is selected in the Object Explorer. The right pane, titled "YOSHIOKA\SQLEXPRESS... - dbo.Mahasiswa", displays the table structure with five columns: "mahasiswa\_id" (int, primary key, not null), "mahasiswa\_nama" (nvarchar(50), nullable), "mahasiswa\_alamat" (nvarchar(50), nullable), "mahasiswa\_jurusan" (nvarchar(50), nullable), and "mahasiswa\_jeniskelamin" (nvarchar(50), nullable).

| Column Name            | Data Type    | Allow Nulls                         |
|------------------------|--------------|-------------------------------------|
| mahasiswa_id           | int          | <input type="checkbox"/>            |
| mahasiswa_nama         | nvarchar(50) | <input checked="" type="checkbox"/> |
| mahasiswa_alamat       | nvarchar(50) | <input checked="" type="checkbox"/> |
| mahasiswa_jurusan      | nvarchar(50) | <input checked="" type="checkbox"/> |
| mahasiswa_jeniskelamin | nvarchar(50) | <input checked="" type="checkbox"/> |
|                        |              | <input type="checkbox"/>            |

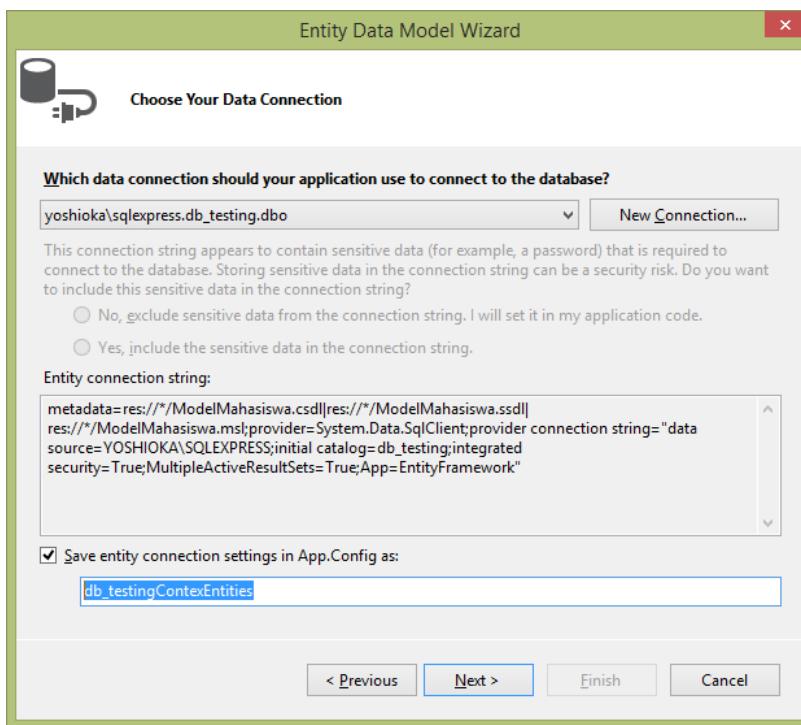
Pada contoh di atas, setelah masuk pada *SQL Server* berhasil dari langkah sebelumnya. Kemudian membuat database dan tabel sebagai bahan untuk ujicoba. Pada contoh di atas nama dabatase adalah **db\_testing**, kemudian menambahkan tabel **Mahasiswa** dan membuat tabel seperti pada contoh gambar di atas.

3. Install Entity Framework pada Visual Studio, TOOLS > Library Package Manager > Manage NuGet for Package Solution. Kemudian pilih Entity Framework
4. Membuat DbContext.

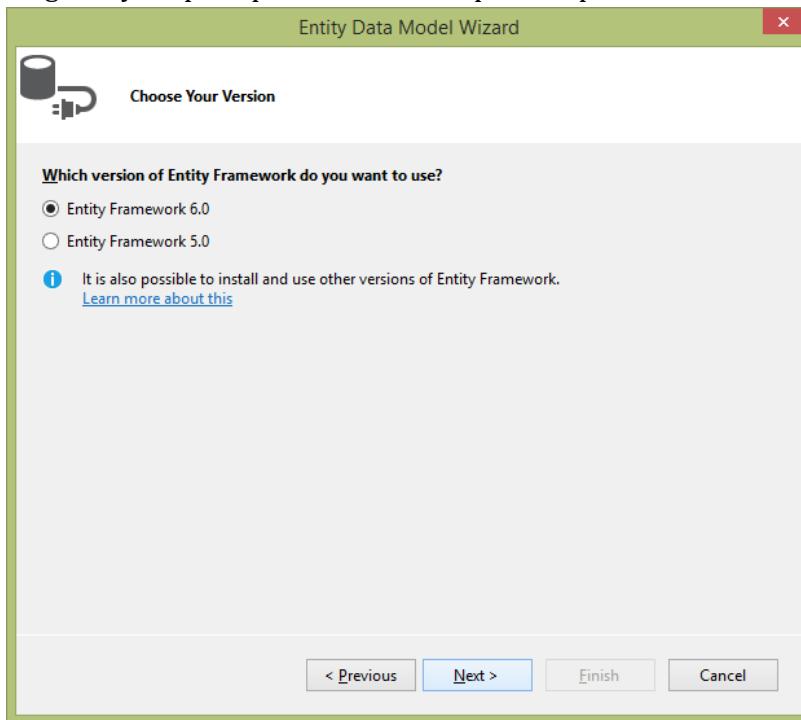
Klik Kanan Pada Projek > Add > New Item > ADO.Net Entity Data Model



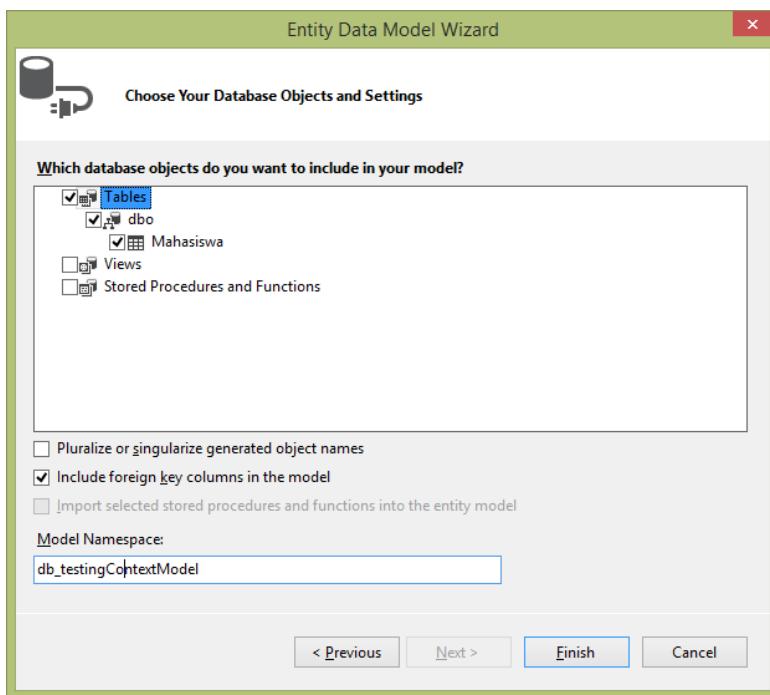
Kemudian pilih Generate From Database, perlu di ingat bahwa untuk contoh ini adalah men-generate data Model dari database.



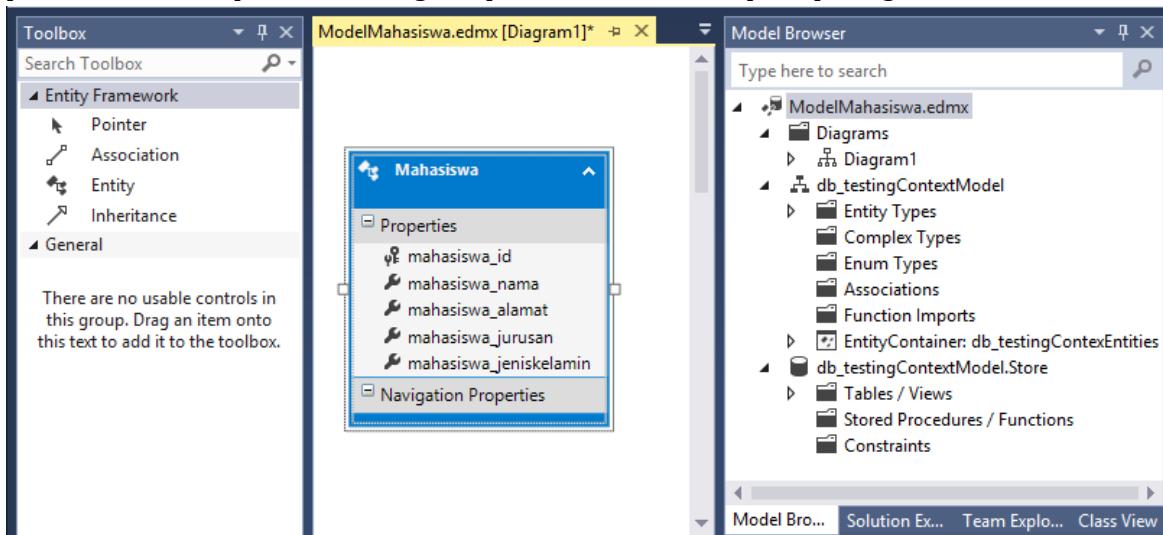
Jika koneksi ke database sudah dilakukan maka secara otomatis akan muncul seperti di atas pada Connection(koneksi sudah dilakukan pada prak sebelumnya BAB ADO.Net pada step ke-3). Jika koneksi belum di set silahkan klik pada New Connection dan silahkan diikuti langkahnya seperti pada BAB AD.Net pada step ke-3.



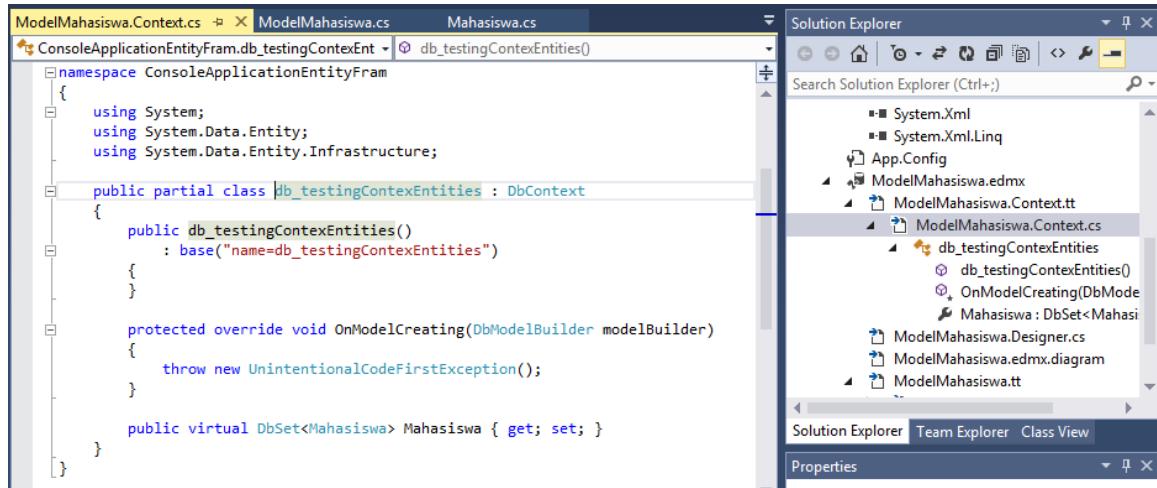
Pilihlah pada Entity Framework 6.0 dan kemudian pilih Next



Pilihlah database Objek yang ingin dilakukan import kedalam Model Diagram, kemudian pilih Finish. Maka pada Model Diagram pada akan terlihat seperti pada gambar berikut.



Perhatikan bagian kanan window Visual Studio pada Window Model Browser terlihat struktur dari Model Diagram. Pilihlah ke window Solution Explorer, terlihat pada projek yang telah dibuat terdapat tambahan file-file baru yang salah satunya adalah dile DbContext, yang jika dibuka isinya adalah sebagai berikut.



## 5. Membuat Connection String

Dengan menggunakan Entity Framework Programmer tidak perlu membuat koneksi secara manual. Secara default sudah ditangani oleh Entity Framework. Sehingga Programmer bisa langsung fokus ke proses manipulasi data ke Database.

## 6. Membuat Fungsi Untuk Insert Data

Untuk membuat fungsi tambah data pada C# menggunakan Entity Framework adalah sebagai berikut.

```

static void tambahData()
{
    var contex = new db_testingContextEntities();
    var mhs = new Mahasiswa();
    mhs.mahasiswa_id = 4;
    mhs.mahasiswa_nama = "Muhammad Alibaba";
    mhs.mahasiswa_alamat = "Jl Bumi Marina 46 Surabaya";
    mhs.mahasiswa_jurusan = "Teknik Informatika";
    mhs.mahasiswa_jeniskelamin = "Laki";
    contex.Mahasiswa.Add(mhs);
    contex.SaveChanges();
}

```

Meskipun demikian pada **Entity Framework** juga tetap menyediakan fungsi untuk memanggil perintah Query dengan menggunakan **ADO.Net**. Misalkan untuk kode di atas juga bisa ditulis seperti kode berikut di bawah.

```

static void tambahData()
{
    var contex = new db_testingContextEntities();
    string query = "INSERT INTO Mahasiswa VALUES(4, 'Muhammad Alibaba', 'Jl Bumi Marina 46 Surabaya', 'Teknik Informatika', 'Teknik Informatika')";
    contex.Database.ExecuteSqlCommand(query);
}

```

Jika program di atas berhasil dijalankan maka pada database SQL Server akan muncul data tersebut seperti pada gambar berikut.

|   | mahasiswa_id | mahasiswa_nama   | mahasiswa_alamat           | mahasiswa_jurusan  | mahasiswa_jeniskelamin |
|---|--------------|------------------|----------------------------|--------------------|------------------------|
| 1 | 2            | Elisa Yamada     | Jl Semilowaru 46 Surabaya  | Teknik Informatika | Perempuan              |
| 2 | 3            | Valentino Rossi  | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki                   |
| 3 | 4            | Muhammad Alibaba | Jl Bumi Marina 46 Surabaya | Teknik Informatika | Laki                   |

Terlihat pada gambar di atas data dengan mahasiswa\_id=4 telah masuk ke dalam SQL Server Database.

## 7. Membuat Fungsi untuk **Update Data**

Untuk membuat fungsi update data pada C# menggunakan Entity Framework adalah sebagai berikut.

```
static void updateData()
{
    var contex = new db_testingContexEntities();
    var mhs = (from a in contex.Mahasiswa where a.mahasiswa_id == 4
select a).Single();
    mhs.mahasiswa_nama = "Muhammad Ali";
    contex.SaveChanges();
}
```

Jika program di atas berhasil dijalankan maka pada database SQL Server akan muncul data tersebut seperti pada gambar berikut. Terihat data pada mahasiswa\_id=4 telah berubah mahasiswa\_nama menjadi "Muhammad Ali"

|   | mahasiswa_id | mahasiswa_nama  | mahasiswa_alamat           | mahasiswa_jurusan  | mahasiswa_jeniskelamin |
|---|--------------|-----------------|----------------------------|--------------------|------------------------|
| 1 | 2            | Elisa Yamada    | Jl Semilowaru 46 Surabaya  | Teknik Informatika | Perempuan              |
| 2 | 3            | Valentino Rossi | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki                   |
| 3 | 4            | Muhammad Ali    | Jl Bumi Marina 46 Surabaya | Teknik Informatika | Teknik Informatika     |

## 8. Membuat Fungsi untuk **Hapus Data**

Untuk membuat fungsi hapus data pada C# menggunakan Entity Framework adalah sebagai berikut.

```
static void hapusData()
{
    var contex = new db_testingContexEntities();
    var mhs = (from a in contex.Mahasiswa where a.mahasiswa_id ==
4 select a).Single();
```

```

        contex.Mahasiswa.Remove(mhs);
        contex.SaveChanges();
    }
}

```

Jika berhasil dijalankan program di atas, maka bisa dipastikan bahwa data pada SQL server dengan **mahasiswa\_id=4** telah terhapus dari database SQL Server.

## 9. Tampilan Kode Program Lengkap

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplicationEntityFram
{
    class Program
    {
        static void Main(string[] args)
        {
            hapusData();
        }
        static void tambahData()
        {
            var contex = new db_testingContexEntities();
            var mhs = new Mahasiswa();
            mhs.mahasiswa_id = 4;
            mhs.mahasiswa_nama = "Muhammad Alibaba";
            mhs.mahasiswa_alamat = "Jl Bumi Marina 46 Surabaya";
            mhs.mahasiswa_jurusan = "Teknik Informatika";
            mhs.mahasiswa_jeniskelamin = "Laki";
            contex.Mahasiswa.Add(mhs);
            contex.SaveChanges();

        }
        static void tambahDataADO()
        {
            var contex = new db_testingContexEntities();
            string query = "INSERT INTO Mahasiswa VALUES(4,'Muhammad Alibaba','Jl Bumi Marina 46 Surabaya','Teknik Informatika','Teknik Informatika')";
            contex.Database.ExecuteSqlCommand(query);
        }
        static void updateData()
        {
            var contex = new db_testingContexEntities();
            var mhs = (from a in contex.Mahasiswa where a.mahasiswa_id == 4 select a).Single();
            mhs.mahasiswa_nama = "Muhammad Ali";
            contex.SaveChanges();
        }
        static void hapusData()
        {
            var contex = new db_testingContexEntities();

```

```

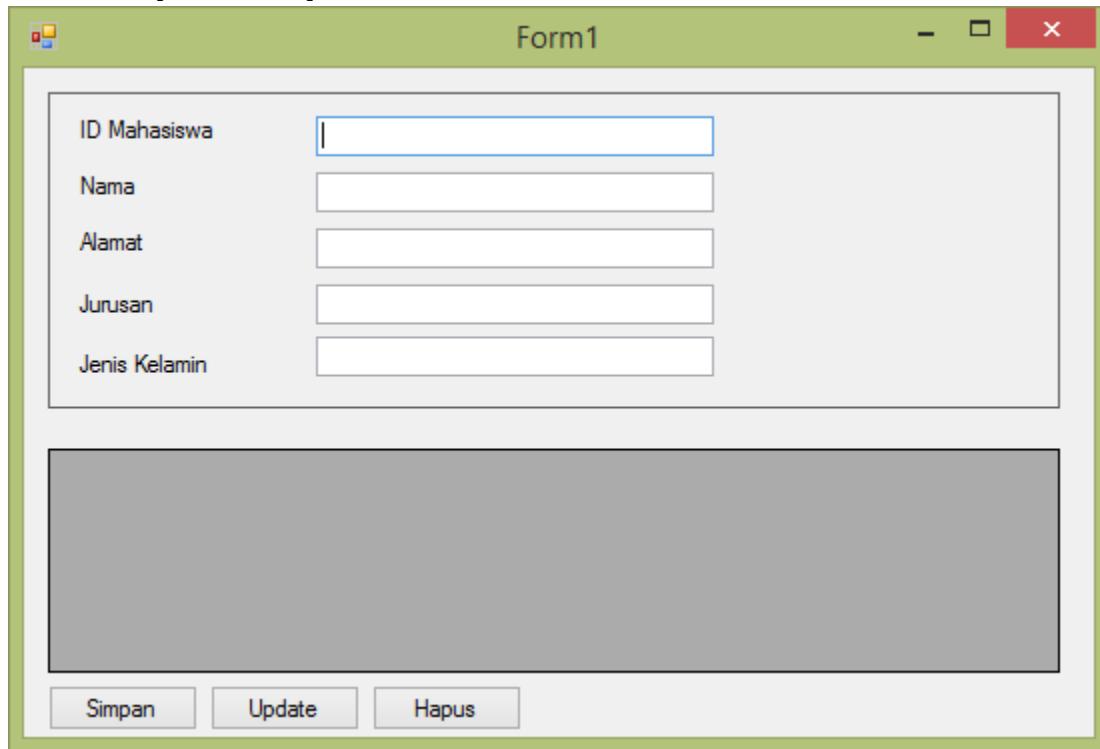
        var mhs = (from a in contex.Mahasiswa where a.mahasiswa_id
== 4 select a).Single();
contex.Mahasiswa.Remove(mhs);
contex.SaveChanges();
    }
}

```

## GUI Windows Application Form

Jika seorang Developer membuat aplikasi tettunya pasti tidak lepas dari peran GUI(Graphic User Interface) sebagai tatap muka untuk User. Pada bagian ini akan mencoba membuat aplikasi CRUD.

1. Desain Tampilan GUI seperti Berikut



Desain nama dari Control yang digunakan adalah sebagai berikut :

```

TextBox textBox_jeniskelamin;
TextBox textBox_jurusan;
TextBox textBox_alamat;
TextBox textBox_nama;
TextBox textBox_id;
DataGridView dataGridView1;
Button button_simpan;
Button button_hapus;

```

```
Button button_update;
```

Berikut adalah kode-kode yang harus dibuat pada Tampilan GUI di atas :

### Kode pada Koneksi

```
static SqlConnection koneksi()
{
    string connection_string = @"Data
Source=YOSHIOKA\SQLEXPRESS;Initial Catalog=db_testing;Integrated
Security=True";
    SqlConnection conn = new SqlConnection(connection_string);
    return conn;
}
```

Kode pada Button **Simpan**(double klik pada Button Simpan dan copy-kan kode berikut)

```
private void button_simpan_Click(object sender, EventArgs e)
{
    SqlConnection conn = koneksi();
    conn.Open();
    SqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "INSERT INTO Mahasiswa
VALUES ("+Convert.ToInt32(textBox_id.Text)+

", '"+textBox_nama.Text+"', '"+textBox_alamat.Text+"', '"+textBox_jurusan.
Text+"', '"+textBox_jeniskelamin.Text+"')";
    cmd.ExecuteNonQuery();
    conn.Close();
    display_data();
    MessageBox.Show("Simpan data berhasil!!");
}
```

Kode pada Button **Update**(double klik pada Button Update dan copy-kan kode berikut)

```
private void button_update_Click(object sender, EventArgs e)
{
    SqlConnection conn = koneksi();
    conn.Open();
    SqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "UPDATE Mahasiswa SET
mahasiswa_nama='"+textBox_nama.Text+
"', mahasiswa_alamat='"+textBox_alamat.Text+"',
mahasiswa_jurusan='"+textBox_jurusan.Text+
"', mahasiswa_jeniskelamin='"+textBox_jeniskelamin.Text+"'
WHERE mahasiswa_id="+Convert.ToInt32(textBox_id.Text);
    cmd.ExecuteNonQuery();
    conn.Close();
    display_data();
```

```
        MessageBox.Show("Update data berhasil!!");  
    }  
}
```

Kode pada Button **Hapus**(double klik pada Button Hapus dan copy-kan kode berikut)

```
private void button_hapus_Click(object sender, EventArgs e)  
{  
    SqlConnection conn = koneksi();  
    conn.Open();  
    SqlCommand cmd = conn.CreateCommand();  
    cmd.CommandText = "DELETE FROM Mahasiswa WHERE  
mahasiswa_id=" + Convert.ToInt32(textBox_id.Text);  
    cmd.ExecuteNonQuery();  
    conn.Close();  
    display_data();  
    MessageBox.Show("Hapus data berhasil!!");  
}  
}
```

Kode pada untuk **menampilkan** data pada **dataGridView**

```
public void display_data()  
{  
    SqlConnection conn = koneksi();  
    conn.Open();  
    SqlCommand cmd = conn.CreateCommand();  
    cmd.CommandText = "SELECT mahasiswa_id AS ID, mahasiswa_nama  
AS Nama, mahasiswa_alamat AS Alamat, mahasiswa_jurusan AS Jurusan,  
mahasiswa_jeniskelamin AS Jenis_Kelamin FROM Mahasiswa";  
    cmd.ExecuteNonQuery();  
    DataTable dt = new DataTable();  
    SqlDataAdapter da = new SqlDataAdapter(cmd);  
    da.Fill(dt);  
    dataGridView1.DataSource = dt;  
    conn.Close();  
}  
}
```

Jika nama Control yang dibuat berbeda dengan contoh di atas silahkan disesuaikan dengan nama Control yang telah Anda buat.

## 2. Tampilan pada saat Simpan Data

The screenshot shows a Windows application window titled "Form1". The interface consists of two main sections: a data entry form at the top and a grid view at the bottom.

**Data Entry Form:**

- ID Mahasiswa: An input field containing the character "I".
- Nama: An empty input field.
- Alamat: An empty input field.
- Jurusan: An empty input field.
- Jenis Kelamin: An empty input field.

**Grid View:**

|   | ID | Nama | Alamat | Jurusan | Jenis_Kelamin |
|---|----|------|--------|---------|---------------|
| * |    |      |        |         |               |
|   |    |      |        |         |               |

**Buttons:**

- Simpan
- Update
- Hapus

Gambar di atas adalah tampilan awal ketika program di jalankan

Form1

|               |                             |
|---------------|-----------------------------|
| ID Mahasiswa  | 1                           |
| Nama          | Civer Yoshioka              |
| Alamat        | Jl Mangga II Srengat Blitar |
| Jurusan       | Teknik Informatika          |
| Jenis Kelamin | Laki                        |

|   | ID | Nama           | Alamat               | Jurusan |
|---|----|----------------|----------------------|---------|
| ▶ | 1  | Civer Yoshioka | Jl Mangga II Sren... | Tekni   |
| * |    |                |                      |         |

**Simpan data berhasil!!**

**OK**

Simpan      Update      Hapus

Tampilan program ketika berhasil memasukan Data

Form1

|               |                        |
|---------------|------------------------|
| ID Mahasiswa  | 3                      |
| Nama          | Valentino Rossi        |
| Alamat        | Jl Bubutan 46 Surabaya |
| Jurusan       | Teknik Informatika     |
| Jenis Kelamin | Laki                   |

|   | ID | Nama            | Alamat                      | Jurusan            | Jenis_Kelamin |
|---|----|-----------------|-----------------------------|--------------------|---------------|
| ▶ | 1  | Civer Yoshioka  | Jl Mangga II Srengat Blitar | Teknik Informatika | Laki          |
|   | 2  | Elisa Yamada    | Jl Semilowaru 46 Surabaya   | Teknik Informatika | Perempuan     |
| * | 3  | Valentino Rossi | Jl Bubutan 46 Surabaya      | Teknik Informatika | Laki          |
|   |    |                 |                             |                    |               |

Simpan      Update      Hapus

Tampilan program setelah berhasil memasukan beberapa Data

|   | mahasiswa_id | mahasiswa_nama  | mahasiswa_alamat            | mahasiswa_jurusan  | mahasiswa_jeniskelamin |
|---|--------------|-----------------|-----------------------------|--------------------|------------------------|
| 1 | 1            | Civer Yoshioka  | Jl Mangga II Srengat Blitar | Teknik Informatika | Laki                   |
| 2 | 2            | Elisa Yamada    | Jl Semilowaru 46 Surabaya   | Teknik Informatika | Perempuan              |
| 3 | 3            | Valentino Rossi | Jl Bubutan 46 Surabaya      | Teknik Informatika | Laki                   |

Tampilan pada Database SQL Server

### 3. Tampilan Pada Saat Update Data

Misalkan pada contoh ini akan melakukan Update Data pada data Mahasiswa yang memiliki ID Mahasiswa=1.

The application window has a title bar 'Form1' and a standard Windows-style window frame.

**Input Form (Top):**

|               |                             |
|---------------|-----------------------------|
| ID Mahasiswa  | 1                           |
| Nama          | Jhono Yoshioka              |
| Alamat        | Jl Mangga II Srengat Blitar |
| Jurusan       | Teknik Elektronika          |
| Jenis Kelamin | Laki                        |

**Data Grid (Bottom):**

|   | ID | Nama            | Alamat                     | Jurusan            | Jenis_Kelamin |
|---|----|-----------------|----------------------------|--------------------|---------------|
| ▶ | 1  | Civer Yoshioka  | Jl Mangga II Srengat Bl... | Teknik Informatika | Laki          |
|   | 2  | Elisa Yamada    | Jl Semilowaru 46 Surab...  | Teknik Informatika | Perempuan     |
|   | 3  | Valentino Rossi | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki          |
| * |    |                 |                            |                    |               |

**Buttons at the bottom:**

- Simpan
- Update
- Hapus

User harus menginputkan ID Mahasiswa yang ingin dilakukan Update beserta Data yang Baru.

|   | ID | Nama            | Alamat                     | Jurusan            | Jenis_Kelamin |
|---|----|-----------------|----------------------------|--------------------|---------------|
| ▶ | 1  | Jhono Yoshioka  | Jl Mangga II Srengat Bl... | Teknik Elektronika | Laki          |
|   | 2  | Elisa Yamada    | Jl Semilowaru 46 Surab...  | Teknik Informatika | Perempuan     |
|   | 3  | Valentino Rossi | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki          |
| * |    |                 |                            |                    |               |

Tampilan setelah Berhasil Melakukan Update Data pada ID Mahasiswa=1.

4. Tampilan Pada Saat **Hapus Data**

Misalkan pada contoh ini akan melakukan Hapus Data pada data Mahasiswa yang memiliki ID Mahasiswa=1.

Form1

|               |                                |
|---------------|--------------------------------|
| ID Mahasiswa  | <input type="text" value="1"/> |
| Nama          | <input type="text" value=""/>  |
| Alamat        | <input type="text" value=""/>  |
| Jurusan       | <input type="text" value=""/>  |
| Jenis Kelamin | <input type="text" value=""/>  |

|   | ID | Nama            | Alamat                     | Jurusan            | Jenis_Kelamin |
|---|----|-----------------|----------------------------|--------------------|---------------|
| ▶ | 1  | Jhono Yoshioka  | Jl Mangga II Srengat Bl... | Teknik Elektronika | Laki          |
|   | 2  | Elisa Yamada    | Jl Semilowaru 46 Surab...  | Teknik Informatika | Perempuan     |
|   | 3  | Valentino Rossi | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki          |
| * |    |                 |                            |                    |               |

Untuk melakukan Hapus Data cukup memasukan ID Data yang ingin di hapus seperti contoh di atas.

Form1

|               |                                |
|---------------|--------------------------------|
| ID Mahasiswa  | <input type="text" value="1"/> |
| Nama          | <input type="text" value=""/>  |
| Alamat        | <input type="text" value=""/>  |
| Jurusan       | <input type="text" value=""/>  |
| Jenis Kelamin | <input type="text" value=""/>  |

|   | ID | Nama            | Alamat                     | Jurusan            | Jenis_Kelamin |
|---|----|-----------------|----------------------------|--------------------|---------------|
| ▶ | 1  | Jhono Yoshioka  | Jl Mangga II Srengat Bl... | Teknik Elektronika | Laki          |
|   | 2  | Elisa Yamada    | Jl Semilowaru 46 Surab...  | Teknik Informatika | Perempuan     |
|   | 3  | Valentino Rossi | Jl Bubutan 46 Surabaya     | Teknik Informatika | Laki          |
| * |    |                 |                            |                    |               |

Hapus data berhasil!!!

5. Tampilan pada **Database SQL Server** setelah data terhapus

100 % <

Results Messages

|   | mahasiswa_id | mahasiswa_nama  | mahasiswa_alamat          | mahasiswa_jurusan  | mahasiswa_jeniskelamin |
|---|--------------|-----------------|---------------------------|--------------------|------------------------|
| 1 | 2            | Elisa Yamada    | Jl Semilowaru 46 Surabaya | Teknik Informatika | Perempuan              |
| 2 | 3            | Valentino Rossi | Jl Bubutan 46 Surabaya    | Teknik Informatika | Laki                   |