

BAB VIII

ADO.NET Database Connectivity

ADO.NET

Sebelum mempelajari lebih dalam mengenai ADO.NET, perlu untuk diketahui sebagai gambaran awal mengenai konsep koneksi dari sebuah aplikasi ke sebuah Database. Dalam bahasa pemrograman apapun untuk koneksi ke Database dilakukan melalui sekumpulan API(Application Program Interface) contohnya JAVA dengan JDBC(*Java Database Connectivity*) , ODBC, OLE DB dll.

ADO.NET adalah teknologi baru dari Microsoft. Meskipun ADO.NET adalah pengganti teknologi sebelumnya yaitu ADO. Tetapi ADO.NET adalah teknologi yang benar-benar berbeda dari sebelumnya(ADO) yang telah dikembangkan Microsoft untuk *Database Manipulation Process*. Kelebihan dari menggunakan ADO.NET adalah dengan adanya konsep Disconnected Architecture yang mana ketika Database diputus sementara Data dari database disimpan sementara pada Dataset dalam memori berbentuk XML format sehingga bisa dikenali semua platform, dengan adanya modelini dengan menggunakan ADO.NET bisa lebih baik dalam menangani Database yang di akses banyak user secara bersamaan.

Beberapa komponen dari ADO.NET adalah sebagai berikut :

- SqlConnection
- SqlCommand
- SqlDataAdapter
- OleDbConnection
- OleDbCommand
- OleDbDataAdapter
- DataSet
- DataView

Berikut adalah penjelasan dari Objek-objek yang sering digunakan dan dibutuhkan untuk proses manipulasi data pada Database dengan ADO.NET.

Connection

Objek *Connection* digunakan untuk menciptakan koneksi ke Database. ADO .NET menyediakan dua jenis Objek untuk koneksi ke Database yaitu : *SqlConnection* yang dikususkan untuk koneksi ke SQL Server dan *OleDbConnection* digunakan untuk koneksi ke berbagai jenis Database seperti Microsoft Access dan Oracle. Objek Connection berisi semua informasi mengani Sumber Database, Nama Database, Security, User Name, Password dll.

Command

Objek ini digunakan untuk membuat perintah ke Database. Perintah yang ditulis melalui SQL Statement akan dilewatkan melalui Objek *Connection* untuk membuka koneksi ke Database. Kemudian Statement dilewatkan melalui Objek *SqlCommand* untuk eksekusi perintah ke Database.

DataAdapter

DataAdapter adalah inti dari ADO.NET. *DataAdapter* adalah kelas yang akan menangani Data yang di ambil dari Database ketika koneksi terputus. *DataAdapter* juga bertanggungjawab komunikasi Data antara Database dan *DataSet*.

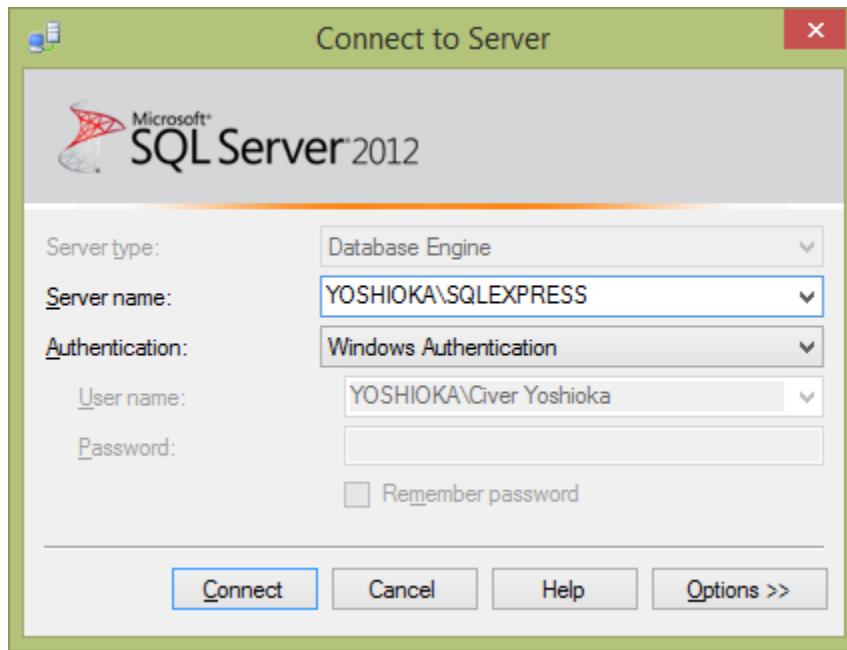
DataSet

Adalah objek yang terpisah dari Database. *DataSet* dapat digambarkan sebagai Objek Data hasil copy-an data yang diambil dari Database. Di dalam *DataSet* juga terdapat Baris, Kolom, Relasi, Constraint dll.

PERCOBAAN

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

1. Buka pada **SQL Server Management Studio** 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. Contoh penulisan Database Engine di atas bisa ditulis dengan format sebagai berikut, **.\SQLEXPRESS**. Maksudnya adalah semua Server Engine yang bernama SQLEXPRESS pada Komputer akan dikenali secara otomatis. 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

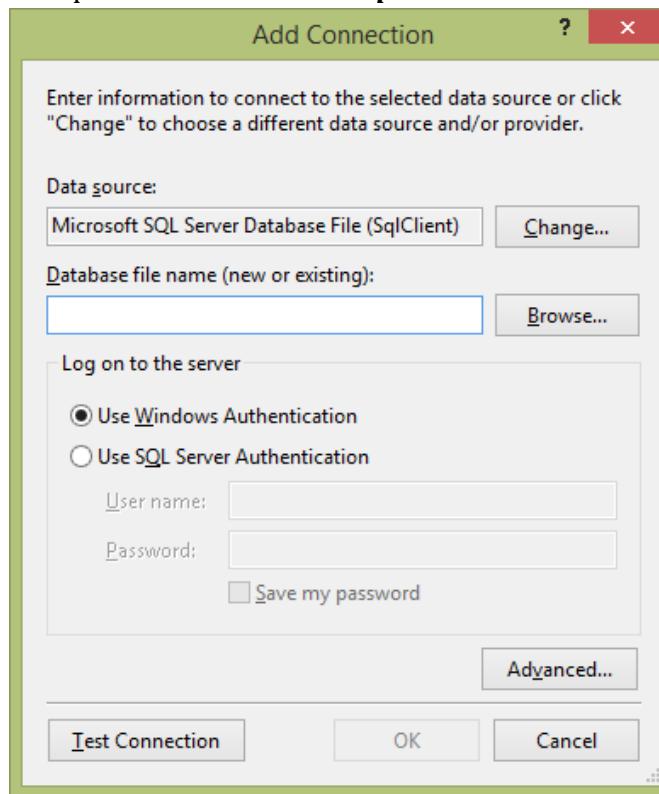
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "YOSHIOKA\SQLEXPRESS.db_testing - dbo.Table_1YOSHIOKA\SQLEXPRESS". The left pane, titled "Object Explorer", shows the database structure. It includes a tree view of "YOSHIOKA\SQLEXPRESS (SQL Server 11)", "Databases" (with "System Databases", "my_db", and "db_testing"), "Tables" (with "System Tables", "FileTables", and "dbo.Mahasiswa"), and sub-items for "dbo.Mahasiswa" such as "Columns", "Keys", "Constraints", "Triggers", "Indexes", and "Statistics". The right pane, titled "YOSHIOKA\SQLEXPRESS... - dbo.Mahasiswa", displays the table structure with the following columns:

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"/>

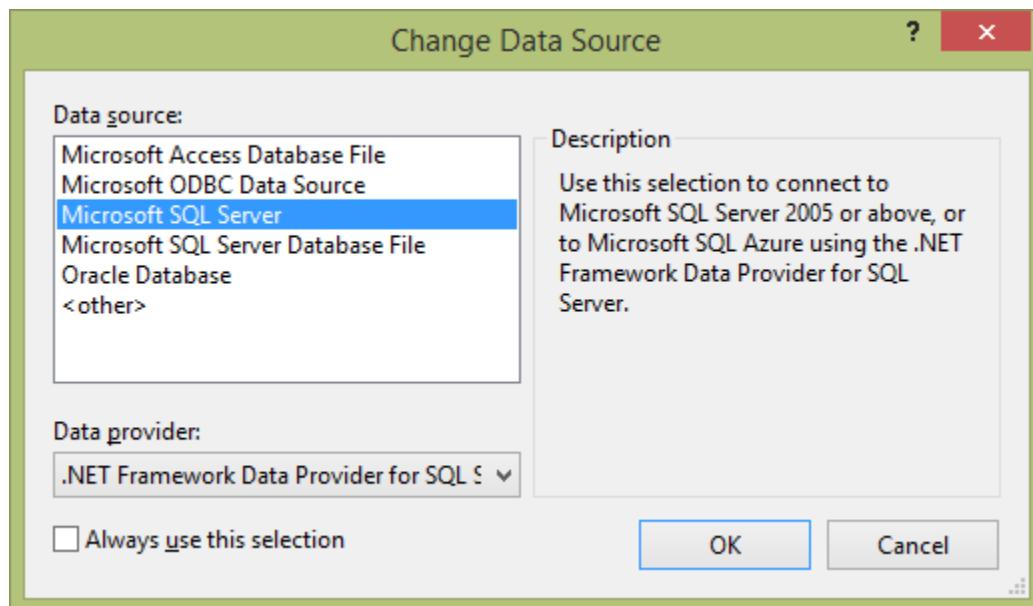
Pada contoh di atas, setelah masuk pada *SQL Server* berhasil dari langkah sebelumnya. Kemudian membuat database dan tabel sebagai bahan untuk ujicoba. Pada contoh di atas nama dabatase adalah **db_testing**, kemudian menambahkan tabel **Mahasiswa** dan membuat properti tabel seperti pada contoh gambar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

3. Membuat Projek pada Visual Studio dan melakukan setting koneksi ke database yang telah di buat.

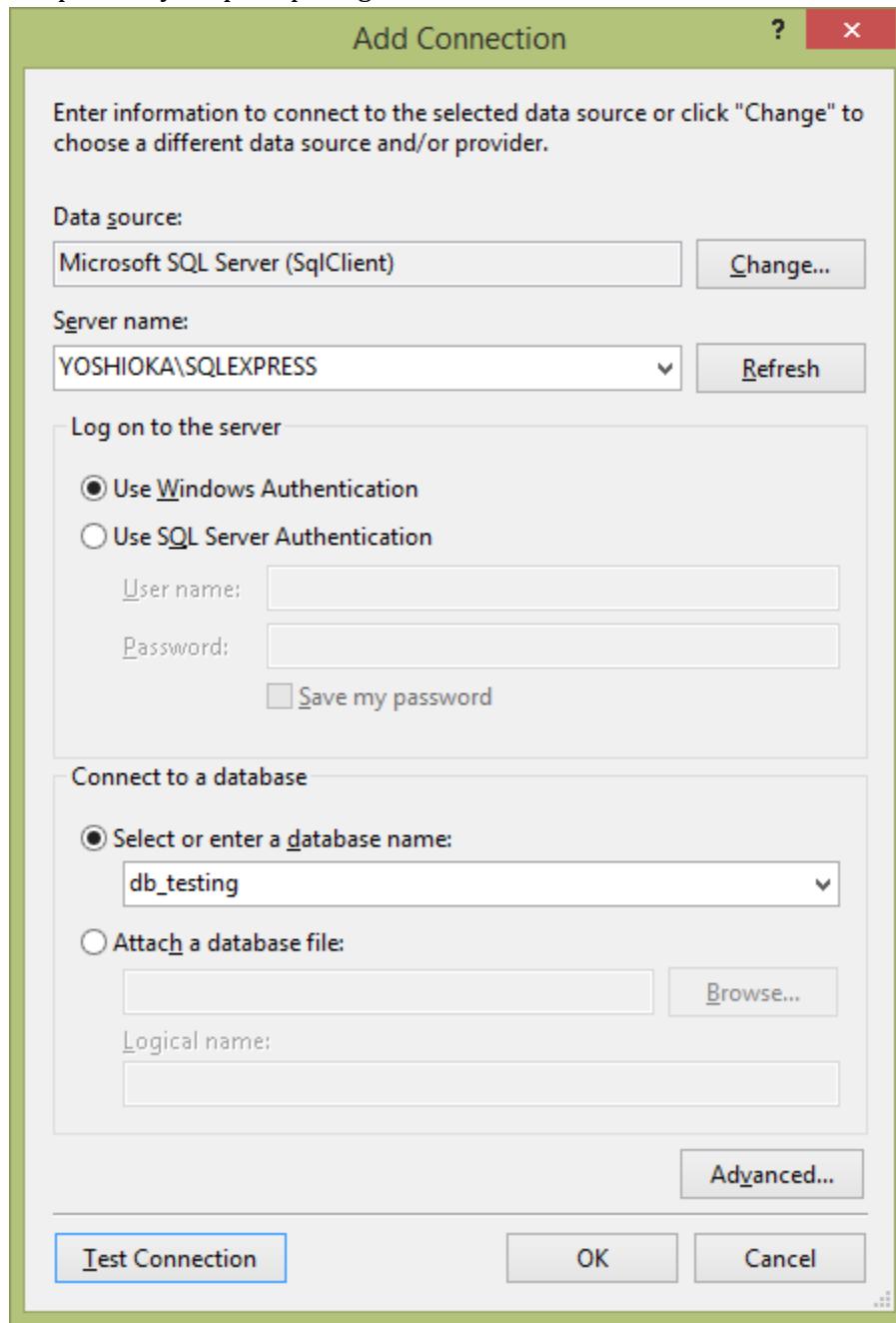
Pilih pada **Window Server Explorer -> Add Connection**



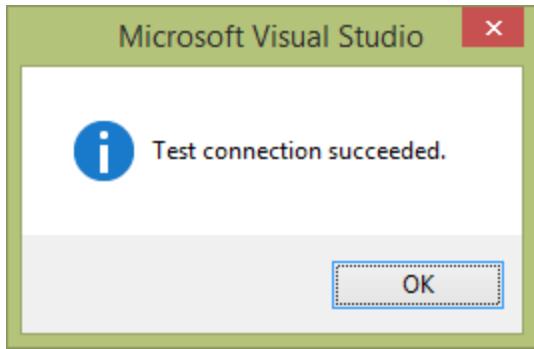
Untuk bagian **Data source**, silahkan pilih **Change**, maka akan muncul jendela untuk memilih **Data Source**. Karena pada percobaan ini menggunakan database langsung dari SQL Server maka pilih **Data Source** pada **Microsoft SQL Server** dan klik **OK**.



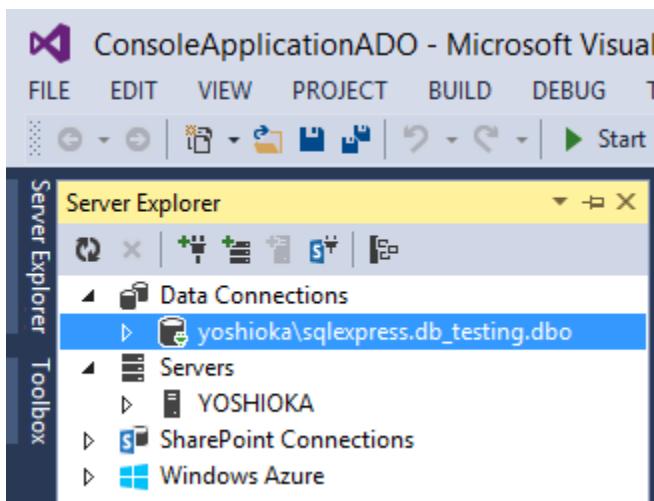
Kemudian pilih OK, maka akan muncul window yang sebelumnya tetapi sedikit berbeda tatapmukanya seperti pada gambar di bawah berikut



Kemudian pada window yang baru muncul, lakukan pengisian seperti pada contoh di atas. Yang harus dilakukan pengisian adalah **Server Name** dan **Database Name** seperti pada contoh di atas. Jika anda menggunakan **SQL Server Authentication** maka jangan lupa untuk mengganti pilihan pada Radio Button ke Use SQL Server Authentication. Untuk mengetes apakah koneksi telah berhasil klik pada Button **Test Connection**, jika muncul window seperti berikut maka setting yang telah dilakukan telah berhasil.

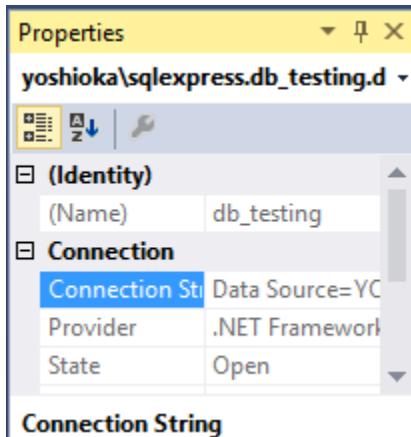


Maka pada window Server Explorer akan muncul **Database Connection** seperti berikut.



4. Membuat **Connection String**

Untuk membuat koneksi dari program Pilih pada **Server Explorer** -> **Properties** -> **Connection String**. Nilai pada *Connection String* inilah yang nanti akan digunakan pada program untuk membuat koneksi ke database. Perhatikan gambar berikut



Kode untuk membuat koneksi dengan menggunakan ADO.NET pada C#. Sebelum menulis kodi di bawah perlu diingat bahwa harus mengimpor terlebih dahulu library untuk koneksi **ke SQL Server** yaitu `using System.Data.SqlClient`.

```
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;
}
```

Perhatikan pada isi dari variabel `connection_string` yang berwarna merah. Value dari variebel di atas adalah nilai dari **Properties -> Connection String** pada **Server Explorer**.

*Note

Untuk membuat koneksi pada program sebenarnya tidak perlu repot-repot harus melalui pada langkah ke-3. Jika sudah hafal dengan konfigurasi SQL Server dan database bisa langsung masuk ke langkah ke-4 dan mengetikkan manual Connection String pada kode program seperti langkah di atas. Tetapi jika anda tidak yakin dengan konfigurasi pada komputer anda lebih aman menggunakan langkah ke-3. Pembuatan koneksi seperti pada langkah ke-3 berguna jika programmer menggunakan Tools dari Visual Studio untuk membuat Fungi CRUD(Creat Read Update Delete) yang akan di bahas pada BAB selanjutnya.

5. Membuat Fungsi Untuk **Insert Data**

Untuk membuat fungsi tambah data pada C# menggunakan ADO.NET adalah sebagai berikut.

```
static void tambahData()
{
    SqlConnection conn = koneksi();
    conn.Open();
    SqlCommand cmd = conn.CreateCommand();
    cmd.CommandText="INSERT INTO Mahasiswa VALUES(1,'Civer
Yoshioka','Jl Mangga II Srengat Blitar','Teknik Informatika','Laki')";
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

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

A screenshot of the SQL Server Management Studio interface. The 'Results' tab is selected. A table is displayed with five columns: 'ma...', 'mahasiswa_nama', 'mahasiswa_alamat', 'mahasiswa_jurusan', and 'mahasiswa_jeniskelamin'. There is one row of data with the following values: 1, Civer Yoshioka, Jl Mangga II Srengat Blitar, Teknik Informatika, Laki.

ma...	mahasiswa_nama	mahasiswa_alamat	mahasiswa_jurusan	mahasiswa_jeniskelamin
1	Civer Yoshioka	Jl Mangga II Srengat Blitar	Teknik Informatika	Laki

6. Membuat Fungsi untuk **Update Data**

Untuk membuat fungsi update data pada C# menggunakan ADO.NET adalah sebagai berikut.

```
static void updateData()
{
    SqlConnection conn = koneksi();
    conn.Open();
    SqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "UPDATE Mahasiswa SET mahasiswa_nama='Jhono Yoshioka' WHERE mahasiswa_id=1";
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

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

A screenshot of the SQL Server Management Studio interface. The 'Results' tab is selected. A table is displayed with five columns: 'maha...', 'mahasiswa_nama', 'mahasiswa_alamat', 'mahasiswa_jurusan', and 'mahasiswa_jeniskelamin'. There is one row of data with the following values: 1, Jhono Yoshioka, Jl Mangga II Srengat Blitar, Teknik Informatika, Laki.

maha...	mahasiswa_nama	mahasiswa_alamat	mahasiswa_jurusan	mahasiswa_jeniskelamin
1	Jhono Yoshioka	Jl Mangga II Srengat Blitar	Teknik Informatika	Laki

Terlihat pada gambar di atas, bahwa nama mahasiswa yang sebelumnya Civer Yoshioka telah ter-Update menjadi Jhono Yoshioka.

7. Membuat Fungsi untuk **Hapus Data**

Untuk membuat fungsi hapus data pada C# menggunakan ADO.NET adalah sebagai berikut.

```
static void hapusData()
{
    SqlConnection conn = koneksi();
    conn.Open();
    SqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "DELETE FROM Mahasiswa WHERE mahasiswa_id=1";
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

Jika berhasil dijalankan program di atas, maka bisa dipastikan bahwa data pada SQL server dengan **mahasiswa_id=1** telah terhapus dari database SQL Server.

8. Tampilan Kode Program Lengkap

```
using System;
using System.Text;
using System.Data.SqlClient;

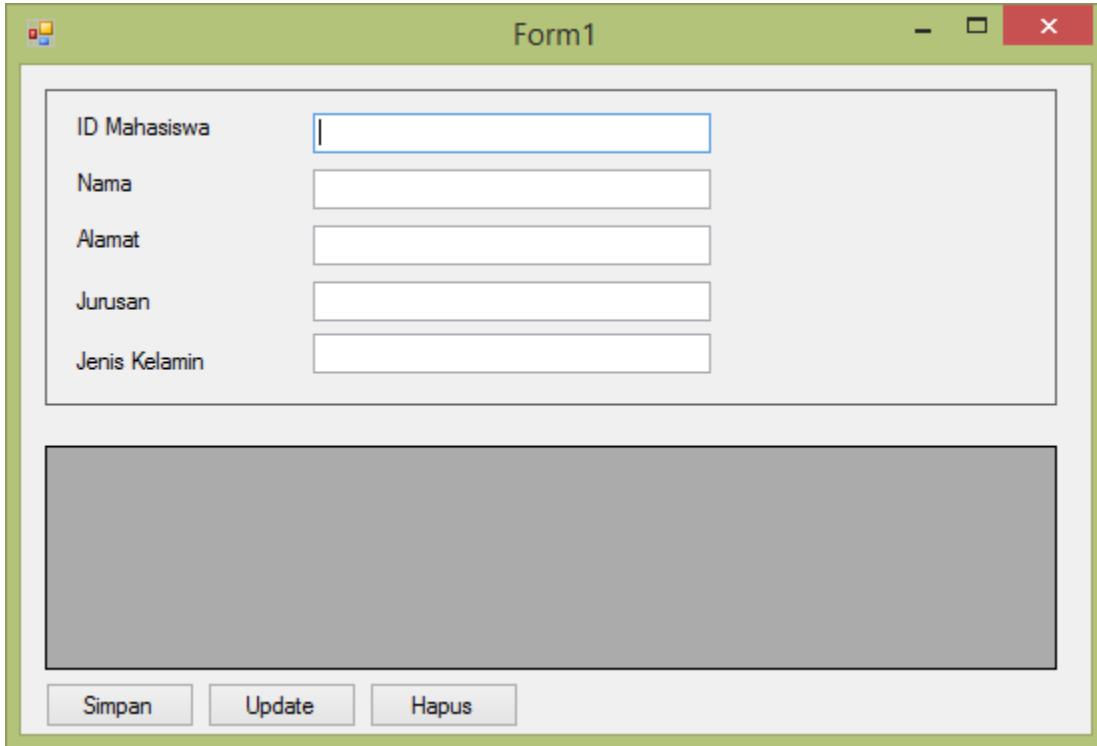
namespace ConsoleApplicationADO
{
    class Program
    {
        static void Main(string[] args)
        {
            //tambahData();
            //updateData();
            hapusData();
            Console.ReadKey();
        }
        static SqlConnection koneksi()
        {
            string connection_string = @"Data
Source=YOSHIOKA\SQLEXPRESS;Initial Catalog=db_testing;Integrated
Security=True";
            SqlConnection conn = new SqlConnection(connection_string);
            return conn;
        }
        static void tambahData()
        {
            SqlConnection conn = koneksi();
            conn.Open();
            SqlCommand cmd = conn.CreateCommand();
            cmd.CommandText="INSERT INTO Mahasiswa VALUES(1,'Civer
Yoshioka','Jl Mangga II Srengat Blitar','Teknik Informatika','Laki')";
            cmd.ExecuteNonQuery();
            conn.Close();
        }
        static void updateData()
        {
            SqlConnection conn = koneksi();
            conn.Open();
            SqlCommand cmd = conn.CreateCommand();
            cmd.CommandText = "UPDATE Mahasiswa SET
mahasiswa_nama='Jhono Yoshioka' WHERE mahasiswa_id=1";
            cmd.ExecuteNonQuery();
            conn.Close();
        }
        static void hapusData()
        {
            SqlConnection conn = koneksi();
            conn.Open();
            SqlCommand cmd = conn.CreateCommand();
            cmd.CommandText = "DELETE FROM Mahasiswa WHERE
mahasiswa_id=1";
            cmd.ExecuteNonQuery();
            conn.Close();
        }
    }
}
```



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

	ID	Nama	Alamat	Jurusan	Jenis_Kelamin
*					

Gambar di atas adalah tampilan awal ketika program di jalankan

▶	1	Civer Yoshioka	Jl Mangga II Sren...	Tekni
*				

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
*					

Tampilan program setelah berhasil memasukan beberapa Data

Results Messages

	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 *I*

Form1

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

	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 Infomatika	Perempuan
	3	Valentino Rossi	Jl Bubutan 46 Surabaya	Teknik Informatika	Laki
*					

Simpan Update Hapus

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

Form1

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

Update data berhasil!!

OK

	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	1
Nama	
Alamat	
Jurusan	
Jenis Kelamin	

	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.

	ID	Nama	Alamat	Jurusan	Jenis Kelamin
▶	2	Elisa Yamada	Jl Semilowaru 46 Surabaya	Teknik Informatika	Perempuan
	3	Valentino Rossi	Jl Bubutan 46 Surabaya	Teknik Informatika	Laki
*					

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

	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

PRAKTIKUM WORKSHOP

1. Cobalah program di atas dengan Database yang dibuat sendiri.