**BAB V**

**Integrasi Basis Data Dalam Pengembangan C# Windows Form Application Menggunakan Visual Studio**

* 1. **Tujuan**

1. Praktikan mampu menghubungkan Sql Server ke dalam aplikasi Windows Form Application menggunakan Visual Studio
2. Praktikan mampu melakukan operasi CRUD (*Create Read Update Delete*) terhadap basis data tertentu melalui aplikasi Windows Form Application menggunakan bahasa C# yang sederhana
3. Praktikan mampu memahami langkah-langkah dalam membuat aplikasi Windows Form Application menggunakan bahasa C#.
4. Praktikan mampu membuat aplikasi Windows Form Application yang sederhana menggunakan bahasa C#.
5. Praktikan mampu memahami langkah-langkah dalam membuat aplikasi Windows Form Application menggunakan bahasa C#.
   1. **Dasar Teori**

**5.2.1 Windows Form**

Windows Forms adalah sebuah nama yang diberikan untuk elemen antarmuka dengan pengguna berbasis grafik yang dimasukkan ke dalam Microsoft .NET Framework, yang mengizinkan akses terhadap elemen-elemen antarmuka grafik sistem operasi Windows dengan memasukkan Windows API yang telah ada ke dalam managed code.

*(Sumber: https://id.wikipedia.org/wiki/Windows\_Forms, diakses pada 20 Oktober 2018)*

**5.2.2 C#**

Bahasa C# adalah sebuah bahasa pemrograman modern yang bersifat general-purpose, ‎berorientasi objek, yang dapat digunakan untuk membuat program di atas arsitektur Microsoft .NET Framework. ‎Bahasa C# ini memiliki kemiripan dengan bahasa Java, C dan C++ (selengkapnya dapat dilihat pada Sejarah ‎Bahasa C#).

Bahasa pemrograman ini dikembangkan oleh sebuah tim pengembang di Microsoft yang dipimpin oleh ‎Anders Hejlsberg, seorang yang telah lama malang melintang di dunia pengembangan bahasa ‎pemrograman karena memang ialah yang membuat Borland Turbo Pascal, Borland Delphi, dan juga Microsoft J++.

Kini, C# telah distandarisasi oleh European Computer Manufacturer Association (ECMA) dan ‎juga International Organization for Standardization (ISO) dan telah menginjak versi 3.0 yang mendukung ‎beberapa fitur baru semacam Language Integrated Query (LINQ) dan lain-lainnya.‎

*(Sumber: https://id.wikibooks.org/wiki/Belajar\_Bahasa\_C\_sharp/Pendahuluan, diakses pada 20 Oktober 2018)*

**5.2.3 Microsoft Visual Studio**

Microsoft Visual Studio by merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

*(Sumber: https://id.wikipedia.org/wiki/Microsoft\_Visual\_Studio, diakses pada 20 Oktober 2018)*

**5.2.4 ADO.Net**

ADO.NET adalah teknologi akses data dari Microsoft .Net Framework. Yang menyediakan komunikasi antara sistem relasional dan non-relasional melalui seperangkat komponen. ADO.NET terdiri dari serangkaian Objek yang mengekspos layanan akses data ke lingkungan NET. ADO.NET dibangun untuk arsitektur terputus, sehingga memungkinkan benar-benar terputus dengan sumber data dapat dilakukan manipulasi melalui Objek Dataset, yang benar-benar independen dari Sumber Data(database).

*(Sumber: http://yoeone.blogspot.com/2011/11/pbd-resume-pertemuan-8.html, diakses pada 20 Oktober 2018)*

**5.2.5 SQL Server**

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (Tabular Data Stream). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (Open Database Connectivity), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data mirroring dan clustering. Pada versi sebelumnya, MS SQL Server 2000 terserang oleh cacing komputer SQL Slammer yang mengakibatkan kelambatan akses Internet pada tanggal 25 Januari 2003.

*(Sumber: https://id.wikipedia.org/wiki/Microsoft\_SQL\_Server, diakses pada 20 Oktober 2018)*

* 1. **Langkah Kerja**

**5.3.1 Langkah Persiapan**

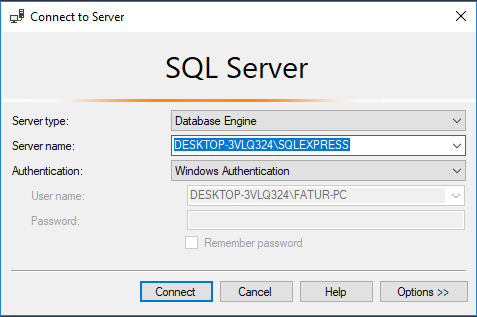
Pastikan setiap kelompok tersedia minimal 1 laptop yang mempunyai spesifikasi RAM 4 GB dan juga sudah terinstal aplikasi-aplikasi berikut:

1. Microsoft SQLServer 2017 (recommended SQL Server 2017)
2. Visual Studio 2017 Community (recommended Visual Studio 2017)
3. XAMPP (Apache, MySQL)
4. Aplikasi untuk *screenshot (sebagai contoh lightshot snipping tool)*

Untuk aplikasi 1 dan 2 diperbolehkan menggunakan versi dibawahnya, tetapi praktikan dapat mengikuti dan menyesuaikan.

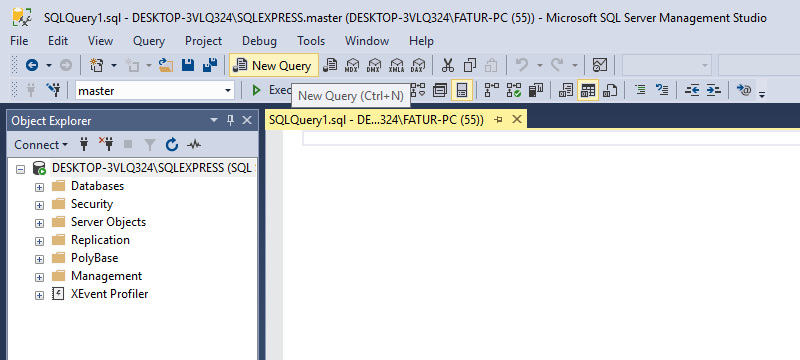
**5.3.2 Langkah Kerja Pertama**

1. Buka Microsoft SQL Server 2017, kemudian tunggu sampai keluar pop up seperti di bawah.



Gambar 5.1 connect SQL Server 2017

1. Selanjutnya klik kanan pada database setelah itu klik New Query



Gambar 5.2 Membuat query baru

1. Kemudian isikan query dengan source code di bawah untuk membuat database baru.

|  |
| --- |
| create database bab5\_kel8; |

1. Lalu kita akan mengaktifkan atau memilih database yang sudah dibuat dengan cara mengetikkan source code di bawah.

|  |
| --- |
| **USE** bab5\_kel8; |

1. Setelah itu kita akan membuat table mobil pada databse yang telah kita buat sebelumnya.

|  |
| --- |
| **CREATE TABLE** mobil(  no\_plat varchar(12) primary key not null,  jenis\_mobil varchar(50) not null,  merk\_mobil varchar(50) not null,  tahun\_buat varchar (4) not null,  harga\_sewa int not null,  lain\_lain varchar(255) not null,  stat varchar(15) not null ); |

1. Kedua, setelah membuat table mobil kita akan membuat table penyewa dengan source code sebagai berikut

|  |
| --- |
| **CREATE TABLE** penyewa(  no\_ktp varchar(18) primary key not null,  nama varchar(50) not null,  alamat varchar(200) not null,  no\_plat varchar(12) not null  ); |

1. Kemudian setelah membuat table mobil pada database, selanjutnya kita akan mengisi nilai/value dengan perintah sebagai berikut

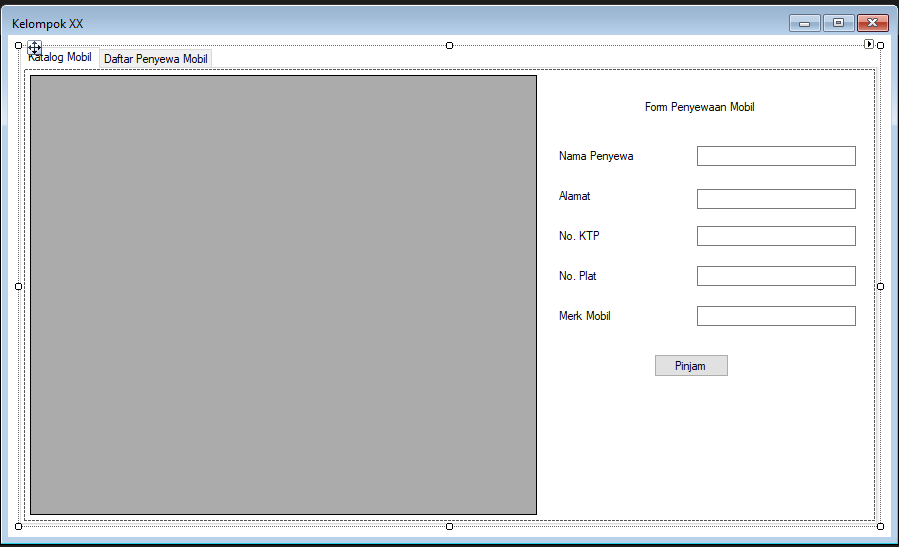
|  |
| --- |
| **INSERT INTO** mobil **VALUES**  ('H499xxs','Innova','Toyota','2014',500000,'warna hitam,BBM pertamas','masih'),  ('H869xxc','Avanza','Toyota','2013',250000,'warna hitam,BBM premium','masih'),  ('B459xx','Gran Max','Daihatsu','2012',300000,'warna putih,BBM pertamas','masih'),  ('F489yy','Xenia','Daihatsu','2015',200000,'warna hitam,BBM pertamas','masih'),  ('B392yy','APV','Suzuki','2015',400000,'warna putih,BBM pertamas','masih'); |

1. Selanjutnya adalah membuat perintah view pada database dengan nama penyewa\_mobil

|  |
| --- |
| create view penyewa\_mobil as  select a.no\_ktp as 'No. Ktp',a.nama as 'Nama',a.alamat as 'Alamat',b.no\_plat 'No. Plat',b.jenis\_mobil as 'Jenis Mobil',b.merk\_mobil as 'Merk Mobil',b.harga\_sewa as 'Harga Sewa'  from penyewa as a inner join mobil as b  on a.no\_plat = b.no\_plat |

**5.3.2 Langkah Kerja Kedua**

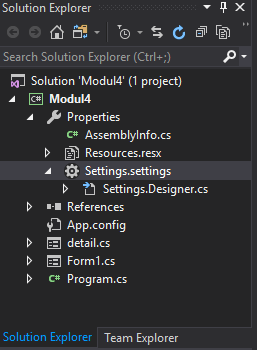
1. Buka project modul4 ke visual studio 2017



Gambar 5.3 project modul4

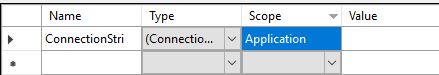
2. Kemudian buka Solution Explorer, pilih Modul4 > Properties > Settings.settings.

klik Settings.settings



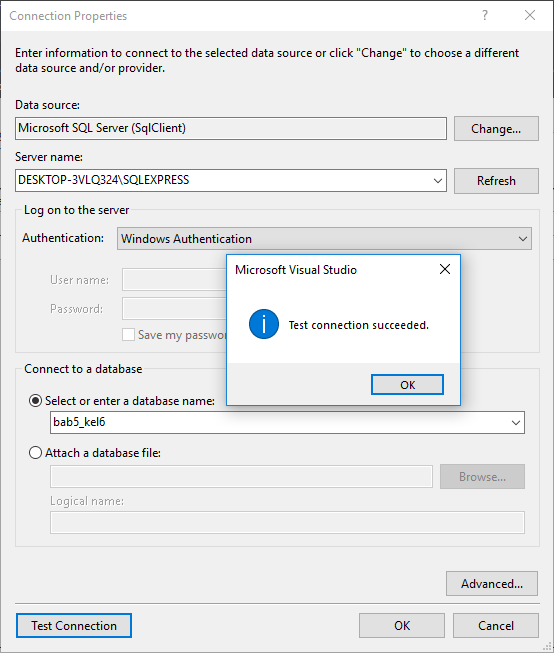
Gambar 5.4 settings.settings

3. Setelah klik settings.settings maka akan muncul form sebagai berikut, pastikan pengaturannya sudah seperti ini > Name : ConnectionString, Type : (Connection String), Scope : Application, Value : ikuti langkah 4



Gambar 5.5 form settings.settings

4. Klik pada form value, maka akan keluar dialog seperti gambar di 5.6, kemudian Isikan server name dengan server name dari SQL Server anda, setelah itu pilih database yang anda gunakan. Setalah memilih database, klik Test Connection. Jika sukses Klik OK dan OK lagi.



Gambar 5.6 pengaturan pada connection properties

5. Klik OK, Setelah itu jangan lupa save form Settings.settings.

6. Kemudian tambahkan source code berikut tepat di atas public String No\_ktp;

|  |
| --- |
| SqlConnection koneksi = new SqlConnection(Properties.Settings.Default.ConnectionString); |

7. Lalu klik kanan pada DataGrid bernama DataMobil pilih properties. Maka dibagian monitor sebelah kanan akan muncul sebuah dialog, setelah itu pilih bagian Events. Cari events bernama *CellDoubleClick* dan klik events tersebut. Setelah itu masukkan source code dibawah ini kedalam method dari dataMobil.

|  |
| --- |
| DataGridViewRow row = this.DataMobil.Rows[e.RowIndex];  No\_Plat = row.Cells[0].Value.ToString();  String jenis = row.Cells[1].Value.ToString();  String merk = row.Cells[2].Value.ToString();  String harga = row.Cells[3].Value.ToString();  detail tampil = new detail(No\_Plat,jenis,merk,harga);  tampil.Visible = true; |

8. Hampir sama dengan langkah sebelumnya, untuk langkah 8 kita cari event bernama CellClick. Dan masukkan senarai berikut.

|  |
| --- |
| DataGridViewRow row = this.DataMobil.Rows[e.RowIndex];  No\_Plat = row.Cells[0].Value.ToString();  plat\_input.Text = No\_Plat;  merk\_input.Text = row.Cells[2].Value.ToString();  nama\_input.Enabled = true;  alamat\_input.Enabled = true;  noktp\_input.Enabled = true;  btn\_pinjam.Visible = true; |

9. Selanjutnya klik 2x pada tombol pinjam, dan masukkan source code berikut ke

dalam tombol tersebut.

|  |
| --- |
| String kueri\_penyewa = "insert into penyewa values(@no\_ktp,@nama,@alamat,@no\_plat)";  using (SqlCommand tambah = new SqlCommand("", koneksi))  {  tambah.Parameters.AddWithValue("@no\_ktp", SqlDbType.VarChar).Value = noktp\_input.Text;  tambah.Parameters.AddWithValue("@nama", SqlDbType.VarChar).Value = nama\_input.Text;  tambah.Parameters.AddWithValue("@alamat", SqlDbType.VarChar).Value = alamat\_input.Text;  tambah.Parameters.AddWithValue("@no\_plat", SqlDbType.VarChar).Value = plat\_input.Text;  koneksi.Open();  tambah.CommandType = CommandType.Text;  tambah.CommandText = kueri\_penyewa;  tambah.ExecuteNonQuery();  tambah.CommandText = "update mobil set stat='dipinjam' where no\_plat = '" + plat\_input.Text + "'";  tambah.ExecuteNonQuery();  koneksi.Close();  tampil\_data();  btn\_pinjam.Visible = false;  nama\_input.Text = null;  nama\_input.Enabled = false;  alamat\_input.Enabled = false;  noktp\_input.Enabled = false;  alamat\_input.Text = null;  noktp\_input.Text = null;  plat\_input.Text = null;  merk\_input.Text = null;  } |

10. Kemudian pilih tab Daftar Penyewa Mobil. Klik kanan pada DataGrid bernama DataPenyewa pilih properties. Maka dibagian monitor sebelah kanan akan muncul sebuah dialog, setelah itu pilih bagian Events. Cari events bernama *CellClick* dan klik events tersebut. Setelah itu masukkan senarai dibawah ini kedalam method dari DataPenyewa.

|  |
| --- |
| DataGridViewRow row = this.DataPenyewa.Rows[e.RowIndex];  No\_ktp = row.Cells[0].Value.ToString();  No\_Plat = row.Cells[3].Value.ToString();  nama\_txt.Text = row.Cells[1].Value.ToString();  btn\_hapus.Visible = true;  nama\_edit.Enabled = true;  alamat\_edit.Enabled = true;  btn\_edit.Visible = true;  nama\_edit.Text = row.Cells[1].Value.ToString();  alamat\_edit.Text = row.Cells[2].Value.ToString(); |

11. Lalu klik 2x tombol hapus dan masukkan source code berikut kedalam tombol tersebut.

|  |
| --- |
| SqlCommand hapus = new SqlCommand();  hapus.Connection = koneksi;  koneksi.Open();  hapus.CommandType = CommandType.Text;  hapus.CommandText = "Delete from penyewa where no\_ktp = '"+No\_ktp+"'";  hapus.ExecuteNonQuery();  hapus.CommandText = "Update mobil set stat='masih' where no\_plat = '" + No\_Plat + "'";  hapus.ExecuteNonQuery();  koneksi.Close();  tampil\_data();  btn\_hapus.Visible = false;  nama\_edit.Enabled = false;  nama\_edit.Text = null;  alamat\_edit.Enabled = false;  alamat\_edit.Text = null;  btn\_edit.Visible = false;  nama\_txt.Text = null; |

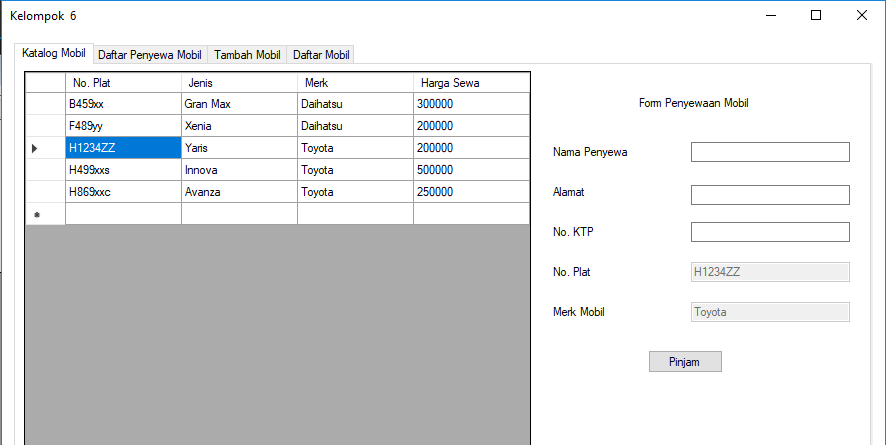
12. Klik 2x pada tombol Edit dan masukkan senarai berikut.

|  |
| --- |
| SqlCommand edit = new SqlCommand();  edit.Connection = koneksi;  koneksi.Open();  edit.CommandType = CommandType.Text;  edit.CommandText = "Update penyewa set nama=@nama,alamat=@alamat where no\_ktp='" + No\_ktp + "'";  edit.Parameters.AddWithValue("@nama", SqlDbType.VarChar).Value = nama\_edit.Text;  edit.Parameters.AddWithValue("@alamat", SqlDbType.VarChar).Value = alamat\_edit.Text;  edit.ExecuteNonQuery();  MessageBox.Show("Data berhasil diubah");  koneksi.Close();  btn\_hapus.Visible = false;  nama\_edit.Enabled = false;  nama\_edit.Text = null;  alamat\_edit.Enabled = false;  alamat\_edit.Text = null;  btn\_edit.Visible = false;  nama\_txt.Text = null;  tampil\_data(); |

13. Masukkan source code berikut, yang letaknya dibawah dari kode public String No\_Plat;

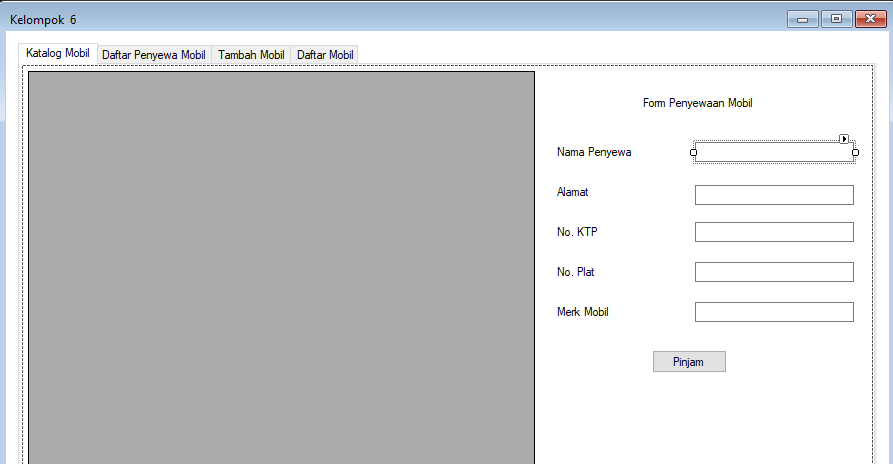
|  |
| --- |
| public void tampil\_data()  {  DataSet mobil = new DataSet();  DataSet penyewa = new DataSet();  var tabel = new SqlDataAdapter();  //menampilkan data pada tab 1 Katalog Mobil  tabel.SelectCommand = new SqlCommand("Select no\_plat as 'No. Plat',jenis\_mobil as 'Jenis', merk\_mobil as 'Merk',harga\_sewa from mobil where stat='masih'");  tabel.SelectCommand.Connection = koneksi;  tabel.SelectCommand.CommandType = CommandType.Text;  tabel.Fill(mobil);  DataMobil.DataSource = mobil.Tables[0];  DataMobil.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;  DataMobil.AutoResizeColumns();  tabel.SelectCommand = new SqlCommand("Select\*from penyewa\_mobil");  tabel.SelectCommand.Connection = koneksi;  tabel.SelectCommand.CommandType = CommandType.Text;  tabel.Fill(penyewa);  DataPenyewa.DataSource = penyewa.Tables[0];  DataPenyewa.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;  DataMobil.AutoResizeColumns();  } |

* 1. **Hasil Percobaan dan Analisa**
     1. **Hasil Percobaan**



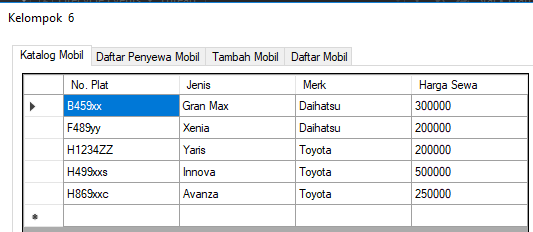
Gambar 5.7 Fungsi pinjam pada katalog mobil

Dalam program penyewaan mobil pada kolom katalog mobil terdapat fungsi pinjam, fungsi ini dapat dideklarasikan menggunakan source code private void btn\_pinjam\_Click(object sender, EventArgs e {} dan dari data pinjam ini akan dimasukkan ke kolom daftar penyewa mobil.



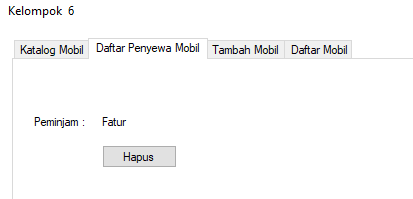
Gambar 5.8 Fungsi edit textbox

Kemudian untuk mendeklarasikan textboxnya kita menggunakan source code private void nama\_input\_TextChanged(object sender, EventArgs e)dan dari textbox ini dapat diketikan atau diisi oleh user.



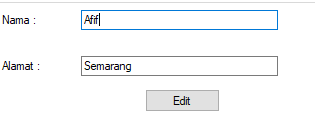
Gambar 5.9 Fungsi menampilkan data

Supaya data yang kita compile dari program dapat ditampilkan, kita dapat menggunakan source code sebagai berikut tampil\_data ();



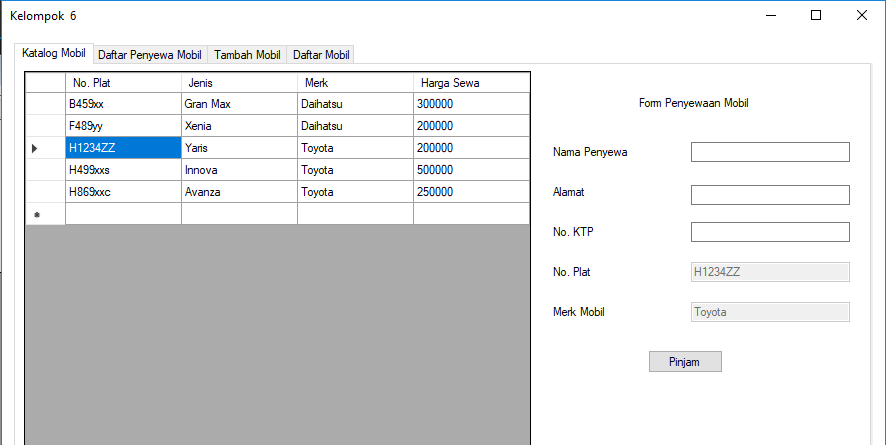
Gambar 5.10 Fungsi hapus data

Setelah data pinjam pada tab katalog mobil masuk ke tab daftar penyewa mobil, di sana disediakan tombol hapus untuk delete data yang sekiranya tidak ingin ditampilkan atau digunakan, dan untuk source codenya dapat didefinisikan dengan private void btn\_hapus\_Click(object sender, EventArgs e) {}

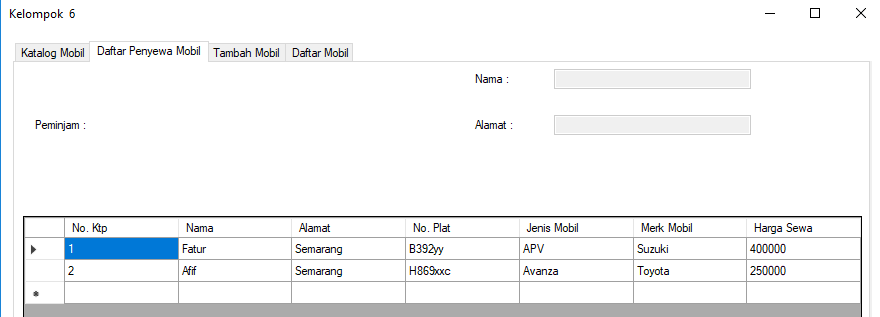


Gambar 5.11 Fungsi edit data

Adapun fungsi yang lainnya yaitu edit data penyewa, untuk mendefinisikannya dapat mengetikkan source code private void btn\_edit\_Click(object sender, EventArgs e){} kemudian setelah kita mengisikan perubahan atau hasil edit pada nama dan alamat, maka data dari daftar penyewa juga otomatis berganti.



Gambar 5.12 Fungsi cellclick pada data mobil

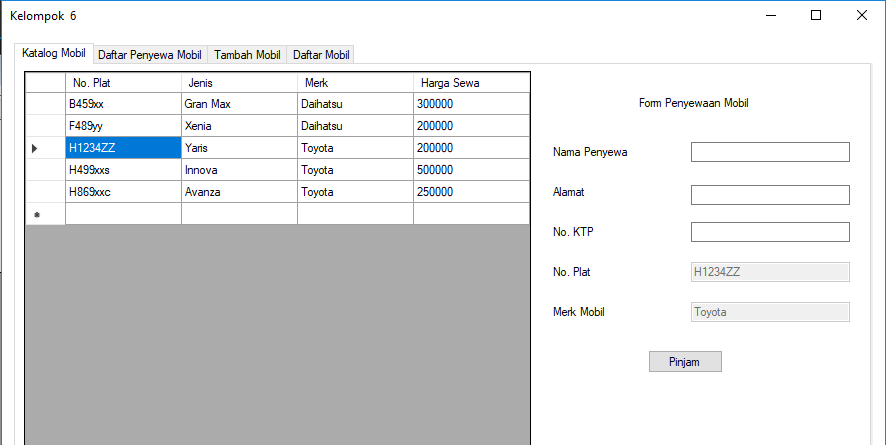


Gambar 5.13 Fungsi cellclick pada data penyewa

Apabila kita mengklik data pada cell maka data tersebut akan muncul pada textbox yang ditentukan, hal ini dapat terjadi karena telah mendefinisikan source code private void DataMobil\_CellClick(object sender, DataGridViewCellEventArgs e) {} pada data mobil dan juga private void DataPenyewa\_CellClick(object sender, DataGridViewCellEventArgs e) {} pada data penyewa

* + 1. **Analisa Percobaan**

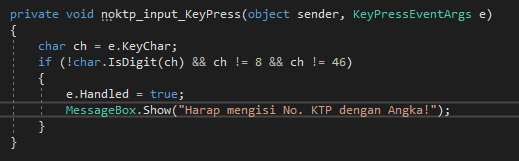
Pada praktikum kali ini kita akan membuat sebuah form dengan Visual Studio 2017 yaitu katalog mobil dan penyewa yang dimana pada form tersebut akan dihubungkan dengan database pada Microsoft SQL Server. Pada form ini berisi data Kalatog Mobil yang dimana data ini sudah kita isikan sebelumnya pada SQL Server yang berupa No.Plat, Jenis, Merk dan Harga Sewa. Dan juga terdapat Form Penyewaan Mobil yang dimana kita bisa mengisi identitas penyewa mobil mulai dari Nama, Alamat hingga No.KTP kemudian button Pinjam yang nantinya data ini akan muncul pada daftar penyewa mobil.



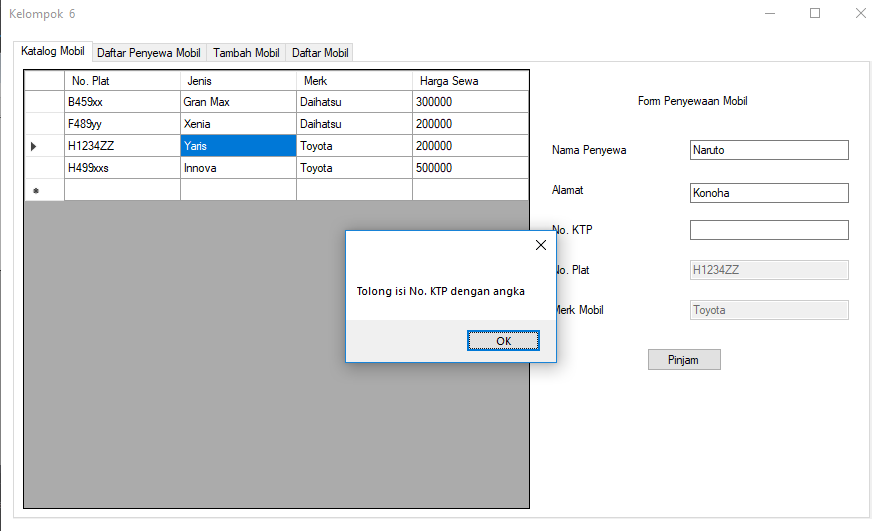
Gambar 5.14 Hasil compile progam

* 1. **Tugas**

1. Textbox no KTP hanya dapat di isi oleh angka



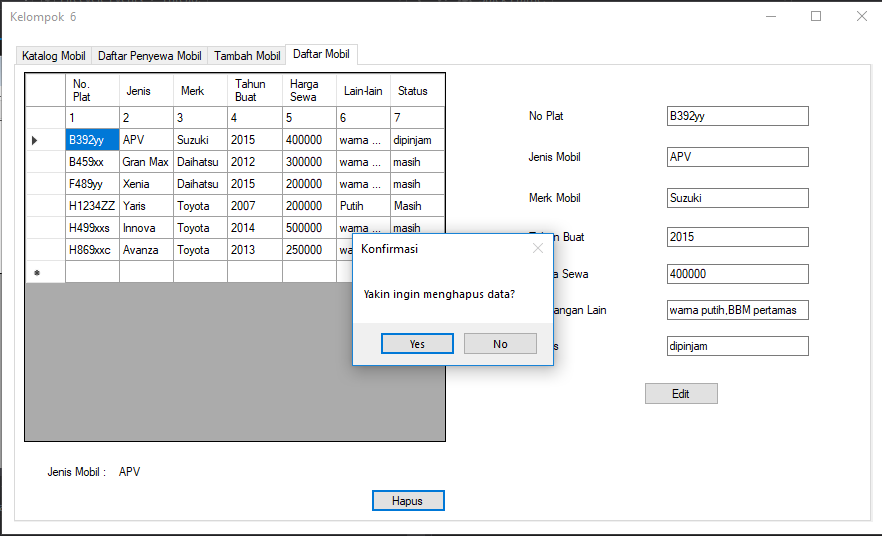
Gambar 5.15 source code pop up nomer KTP



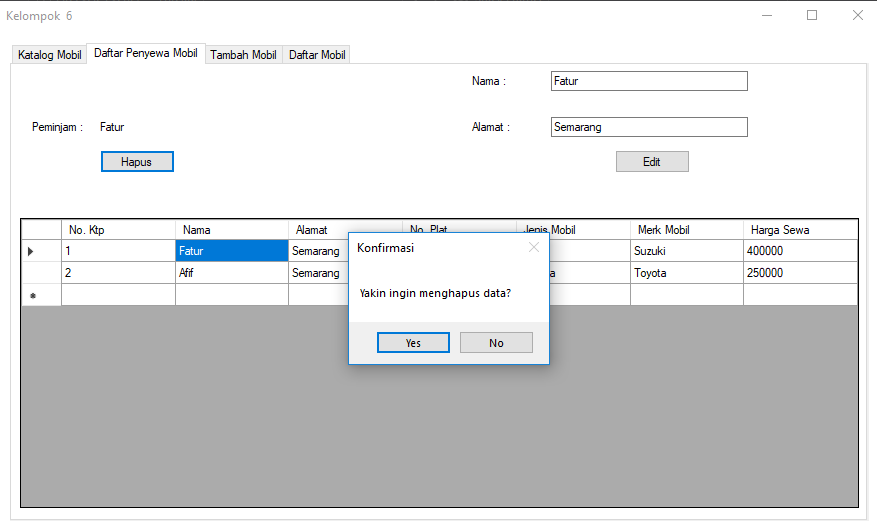
Gambar 5.16 Hasil pop up nomer KTP

Pada tahap ini, kita membuat suatu pop up yang aktif ketika user atau pengguna mengetikkan data selain menggunakan angka, maka akan keluar tulisan Harap mengisi No. KTP dengan angka. Untuk melakukannya dapat menggunakan source code e.Handled = true; MessageBox.Show("Harap mengisi No. KTP dengan Angka!");

1. Fungsi hapus diberikan messagebox *yes or no.*

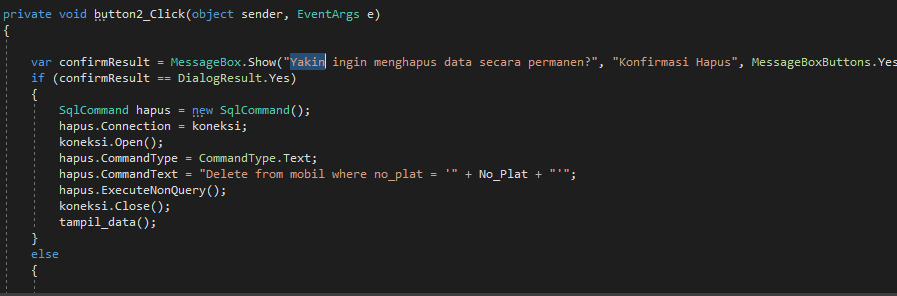


Gambar 5.17 pop up konfirmasi penghapusanpada katalog mobil

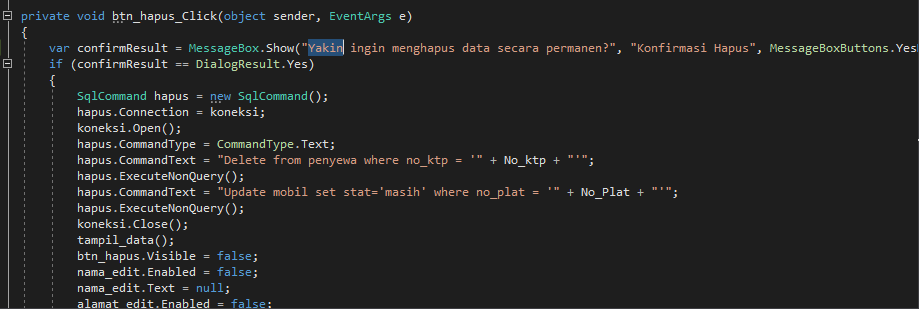


Gambar 5.18 pop up konfirmasi penghapusan pada data penyewa

Kemudian pada langkah selanjutnya, kita akan membuat suatu pop up dimana ketika user atau pengguna akan menghapus data akan muncul tulisan yakin ingin menghapus data secara permanen?.



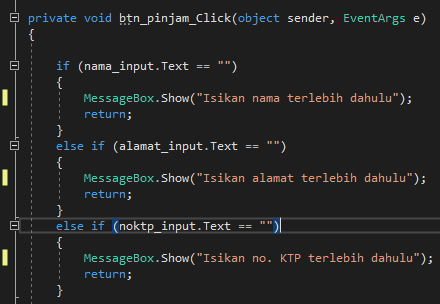
Gambar 5.19 mendefinisikan source code pop up konfirmasi penghapusan



Gambar 5.20 source code pop up konfirmasi penghapusan pada data penyewa mobil

Di atas adalah source code yang digunakan dalam pembuatan pop up konfirmasi penghapusan data secara permanen pada menu katalog mobil dan juga menu data penyewa mobil. Untuk melakukannya dapat menggunakan source code var confirmResult = MessageBox.Show("Yakin ingin menghapus data secara permanen?", "Konfirmasi Hapus", MessageBoxButtons.YesNo); if (confirmResult == DialogResult.Yes)

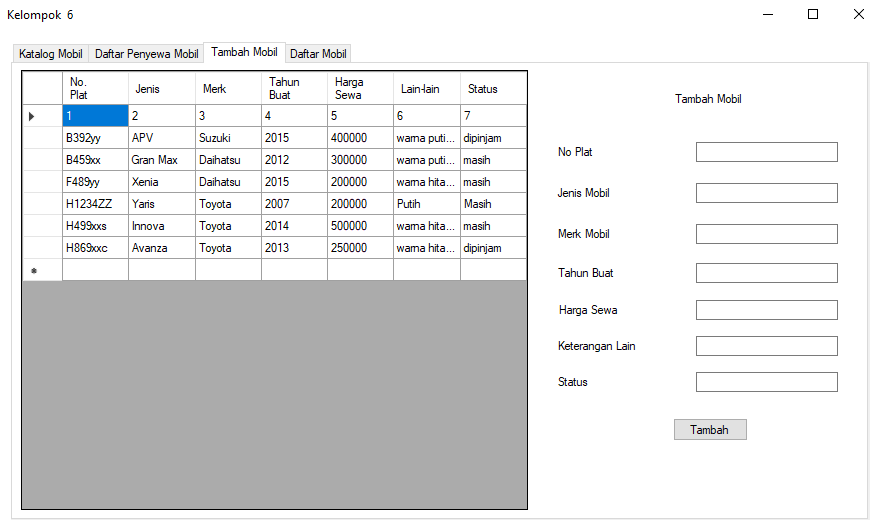
1. Tidak boleh ada textbox yang kosong



Gambar 5.21 source code pop up textbox tidak boleh kosong

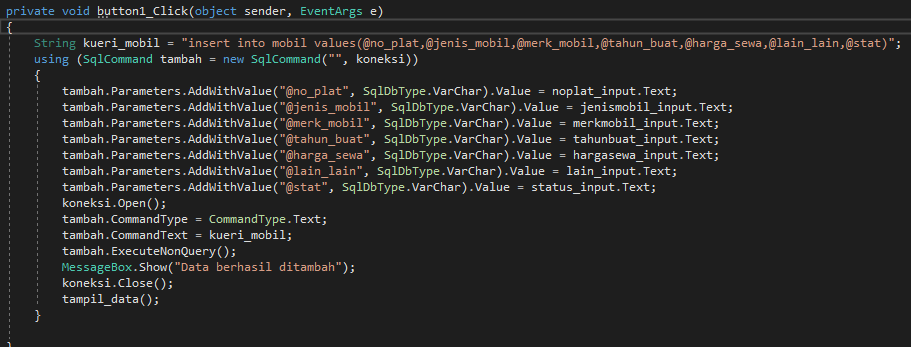
Selanjutnya pada tahap ini kita akan mendefinisikan source code yang digunakan agar user atau pengguna tidak dapat mengkosongkan data atau textbox yang tersedia pada katalog mobil. Salah satu contoh source codenya adalah MessageBox.Show("Isikan nama terlebih dahulu"); return;

1. Pembuatan menu tambah mobil



Gambar 5.22 hasil pembuatan menu tambah mobil

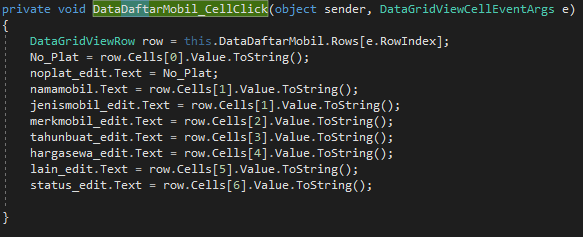
Gambar di atas adalah hasil dari pembuatan menu baru tambah mobil yang sebelumnya tidak ada.



Gambar 5.23 source code menu tambah mobil

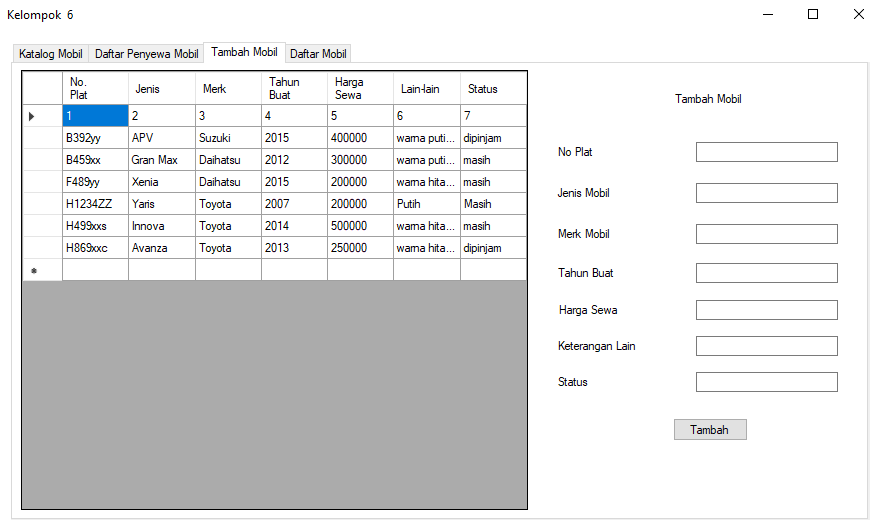
Kemudian untuk source codenya dapat didefinisikan sebagai berikut String kueri\_mobil = "insert into mobil values(@no\_plat,@jenis\_mobil,@merk\_mobil,@tahun\_buat,@harga\_sewa,@lain\_lain,@stat)"; using (SqlCommand tambah = new SqlCommand("", koneksi)) dimana memiliki fungsi untuk menambahkan no plat, jenis mobil, merk mobil, tahun buat, harga sewa, keterangan lain, dan juga status.

1. Pembuatan menu daftar mobil dan edit mobil



Gambar 5.24 source code menu daftar mobil

Lalu, kita juga akan membuat menu baru yaitu daftar mobil di sini kita dapat mengedit dan juga menghapus setiap data yang dirasa perlu ditambahkan data baru dan dirasa ada data yang ingin dihapus, dan berikut adalah source codenya private void DataDaftarMobil\_CellClick(object sender, DataGridViewCellEventArgs e) { DataGridViewRow row = this.DataDaftarMobil.Rows[e.RowIndex];.



Gambar 5.25 hasil pembuatan menu daftar mobil

Hasil compile atau pembuatan menu baru daftar mobil dengan fungsi edit data mobil dan juga hapus data mobil.

* 1. **Kesimpulan**

1. *ExecuteScalar* adalah perintah kebalikan dari *ExecuteNonQuery* yang hanya bisa menjalankan perintah select.
2. *ExecuteNonQuery* adalah perintah yang mengizinkan Windows Formmelakukan fungsi query CRUD melalui aplikasi langsung tetapi tidak bisa memberi izin melakukan Select.
3. Untuk memasukan query pada sebuah variable, kita harus membuat method terlebih dahulu.
4. Tampilkan\_data(); perlu didefinisikan supaya form yang telah kita buat dapat menampilkan data yang sudah diisi
5. SqlConnection koneksi = new SqlConnection(Properties.Settings.Default.ConnectionString); digunakan untuk mengkoneksikan antara sql server dengan data yang kita buat dalam visual studio 2017
6. Fungsi USE pada SQL Server ialah sebagai perintah untuk menentukan table dari database yang ditampilkan datanya.
7. Fungsi View pada SQL Server adalah tabel virtual (bukan tabel sebenarnya) yang dibuat dari beberapa tabel atau view lain.
8. Salah satu fungsi dari *DataGridView* adalah menampilkan data dari database. Selain menampilkan data, *DataGridView* juga dapat memanipulasi data (update, delete) yang berasal dari database. *DataGridView* juga dapat menyimpan data-data dalam bentuk tabel tanpa harus berinteraksi atau terkoneksi dengan database.