

# 8

## Coding Form Master Pelanggan

Materi Pembahasan :

1. Koneksi Form Master Pelanggan
2. Membuat Prosedur Bersih, Atur Tombol dan Refresh Pelanggan
3. Coding Simpan, Edit, Hapus dan Cari

### 8.1 Koneksi Form Master Pelanggan

1. Buka Form Master Pelanggan, lalu pindah ke bagian kode program / view code (F7)
2. Ketik kode **using System.Data.SqlClient;** di bagian header, yang merupakan kode untuk menambahkan library koneksi ke SQL Server

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
//1.tambahkan library sqlserver
using System.Data.SqlClient;
```

3. Ketikkan kode berikut di bawah **public partial class FPelanggan : Form {**

```
4 references
public partial class FPelanggan : Form
{
    //2.import kelas koneksi.cs yang berasal dari folder Kelas menjadi objek --> konn
    Kelas.Koneksi konn = new Kelas.Koneksi();

    //3.tambahkan beberapa variabel yang digunakan untuk proses CRUD
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
```

### 8.2 Membuat Prosedur Bersih, Atur Tombol dan Refresh Pelanggan

Prosedur merupakan bagian kecil dari kode program yang terpisah dalam blok sendiri yang berisi perintah-perintah yang dapat digunakan/dipanggil berulang kali. Pada form master pelanggan akan dibuat 3 prosedur yaitu prosedur bersih, atur tombol dan refresh pelanggan.

1. Pada baris berikutnya, ketik kode berikut :

```
//4.membuat prosedur bersih
5 references
void BERSIH()
{
    txtKodePelanggan.Clear();
    txtNamaPelanggan.Clear();
    txtAlamatPelanggan.Clear();
    txtCariPelanggan.Clear();
}
```

**Prosedur Bersih()** berisi perintah untuk membersihkan komponen inputan (textbox dan combobox) pada form pelanggan.

2. Pada baris berikutnya, ketik kode berikut :

```
//5.membuat prosedur atur_tombol  
5 references  
void ATUR_TOMBOL(Boolean status)  
{  
    BtnEdit.Enabled = status;  
    BtnHapus.Enabled = status;  
}
```

**Prosedur Atur\_Tombol()** berisi perintah untuk mengatur ketersediaan (aktif / non aktif) button edit dan button hapus pada form pelanggan.

3. Pada baris berikutnya, Ketik kode berikut :

```
//6.membuat prosedur refresh_pelanggan  
4 references  
void refresh_pelanggan()  
{  
    SqlConnection conn = konn.GetConn();  
    try  
    {  
        conn.Open();  
        cmd = new SqlCommand("SELECT * FROM pelanggan", conn);  
        ds = new DataSet();  
        da = new SqlDataAdapter(cmd);  
        da.Fill(ds, "pelanggan");  
        ListPelanggan.DataSource = ds;  
        ListPelanggan.DataMember = "pelanggan";  
        ListPelanggan.Refresh();  
    }  
    catch (Exception e)  
    {  
        MessageBox.Show(e.ToString());  
    }  
    finally  
    {  
        conn.Close();  
    }  
}
```

**Prosedur Refresh\_Pelanggan()** berisi perintah untuk memperbaharui data pada listpelanggan. Data yang diperbaharui merupakan data yang ada pada tabel pelanggan.

### 8.3 Coding Simpan, Edit, Hapus dan Cari

1. Kembali ke desain Form Pelanggan, klik 2 kali **button Baru**

Ketik kode berikut :

```
1 reference  
private void BtnBaru_Click(object sender, EventArgs e)  
{  
    //7.memanggil prosedur bersih dan atur_tombol saat klik tombol baru  
    BERSIH();  
    txtKodePelanggan.Focus();  
    ATUR_TOMBOL(false);  
    BtnSimpan.Enabled = true;  
}
```

- Kembali ke desain Form Pelanggan, klik 2 kali **button Simpan**  
 Ketik kode berikut :

```
1 reference
private void BtnSimpan_Click(object sender, EventArgs e)
{
    //8.perintah simpan data pelanggan
    if(txtKodePelanggan.Text.Trim() == "" || txtNamaPelanggan.Text.Trim() == ""
        || txtAlamatPelanggan.Text.Trim() == "")
    {
        //tampilkan pesan bahwa data belum lengkap
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
}
```

```
else
{
    //mempersiapkan sqlconnection
    SqlConnection conn = konn.GetConn();
    try
    {
        //mempersiapkan sqlcommand yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_pelanggan '" + txtKodePelanggan.Text + "','" +
            txtNamaPelanggan.Text + "','" + txtAlamatPelanggan.Text + "'", conn);
        //membuka koneksi database
        conn.Open();
        //menjalankan perintah query
        cmd.ExecuteNonQuery();
        //menampilkan pesan berhasil disimpan
        MessageBox.Show("Data berhasil disimpan", "INFORMASI", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        //memanggil prosedur BERSIH dan refresh_pelanggan
        BERSIH();
        refresh_pelanggan();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
}
```

**EXECUTE simpan\_pelanggan**  
 merupakan perintah untuk  
 mengeksekusi **Stored Procedure**  
**simpan\_pelanggan** yang ada pada

- Kembali ke desain form Pelanggan, klik komponen **Datagridview ListPelanggan**, lalu klik 2x **event CellClick**, ketik kode berikut :

```
1 reference
private void ListPelanggan_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //9.mengambil nilai dari record yang dipilih menuju ke textbox
    try
    {
        BtnSimpan.Enabled = false;
        ATUR_TOMBOL(true);
        DataGridViewRow row = this.ListPelanggan.Rows[e.RowIndex];
        txtKodePelanggan.Text = row.Cells["KodePelanggan"].Value.ToString();
        txtNamaPelanggan.Text = row.Cells["NamaPelanggan"].Value.ToString();
        txtAlamatPelanggan.Text = row.Cells["AlamatPelanggan"].Value.ToString();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

4. Kembali ke desain Form Pelanggan, klik 2 kali **button Edit**  
 Ketik kode berikut :

```
1 reference
private void BtnEdit_Click(object sender, EventArgs e)
{
    //10.perintah edit data pelanggan
    if (txtKodePelanggan.Text.Trim() == "" || txtNamaPelanggan.Text.Trim() == ""
        || txtAlamatPelanggan.Text.Trim() == "")
    {
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }

    else
    {
        SqlConnection conn = konn.GetConn();
        try
        {
            cmd = new SqlCommand("EXECUTE ubah_pelanggan '" + txtKodePelanggan.Text + "', '" +
                txtNamaPelanggan.Text + "', '" + txtAlamatPelanggan.Text + "'", conn);
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Data berhasil dirubah", "INFORMASI", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            BERSIH();
            refresh_pelanggan();
            //memanggil prosedur atur_tombol
            ATUR_TOMBOL(false);
            BtnSimpan.Enabled = true;
        }
        catch (Exception x)
        {
            MessageBox.Show(x.ToString());
        }
    }
}
```

5. Kembali ke desain form Pelanggan, klik 2 kali **button Hapus**, ketik kode berikut :

```
1 reference
private void BtnHapus_Click(object sender, EventArgs e)
{
    //11.perintah hapus data pelanggan
    if (MessageBox.Show("Yakin ingin menghapus data pelanggan ?", "Konfirmasi",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        SqlConnection conn = konn.GetConn();
        cmd = new SqlCommand("EXECUTE hapus_pelanggan '" + txtKodePelanggan.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data pelanggan berhasil dihapus", "informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        BERSIH();
        refresh_pelanggan();
        //memanggil prosedur atur_tombol
        ATUR_TOMBOL(false);
        BtnSimpan.Enabled = true;
    }
}
```

6. Kembali ke desain form Pelanggan, klik 2 kali **button Keluar**  
Ketik kode berikut :

```
1 reference
private void BtnKeluar_Click(object sender, EventArgs e)
{
    //12.keluar form pelanggan
    Close();
}
```

7. Kembali ke desain form Pelanggan, klik 2 kali **txtcaripelanggan**, ketik kode berikut :

```
1 reference
private void txtCariPelanggan_TextChanged(object sender, EventArgs e)
{
    //13.perintah cari nama pelanggan
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("EXECUTE cari_pelanggan '" + txtCariPelanggan.Text + "'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "pelanggan");
    ListPelanggan.DataSource = ds;
    ListPelanggan.DataMember = "pelanggan";
    ListPelanggan.Refresh();
    conn.Close();
}
```

8. Masih di mode view code, cari kode berikut

```
1 reference
public FPelanggan()
{
    InitializeComponent();
}
```

9. Ketik kode berikut di bawah **InitializeComponent();**

```
1 reference
public FPelanggan()
{
    InitializeComponent();
    //14.panggil prosedur pada saat form pelanggan pertama kali dijalankan
    BERSIH();
    ATUR_TOMBOL(false);
    refresh_pelanggan();
}
```

10. **Running program**, gambar 8.1 merupakan tampilan form master pelanggan yang sudah terkoneksi dengan (database) tabel pelanggan

KodePelanggan	NamaPelanggan	AlamatPela
P0001	aditya	Bandar Lam
P0002	nandi	Lampung B
P0003	indrajat	Lampung Se
P0004	ali	Bandar Lam
P0005	agung p	Lampung Se
P0006	dedi	Lampung Se
P0007	rhendy	Lampung Se
P0008	lulus	Lampung Ti

**Gambar 8.1. Form Master Pelanggan yang Terkoneksi Database**