

# 9

## Coding Form Transaksi Penjualan

Materi Pembahasan :

1. Coding Form Cari Pelanggan dan Form Cari Barang
2. Koneksi Form Transaksi Penjualan
3. Membuat Prosedur pada Form Transaksi Penjualan
4. Coding pada Form Transaksi Penjualan

### 9.1 Coding Form Cari Pelanggan dan Form Cari Barang

1. Buka Form Cari Pelanggan (FCariPelanggan), 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 FCariPelanggan : Form {**

4 references

```
public partial class FCariPelanggan : Form
{
    //2.tambahkan beberapa variabel yang digunakan untuk proses Read Data
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
    public string kodeplg, namaplg = "";

    //3.import kelas koneksi.cs yang berasal dari folder Kelas menjadi objek --> konn
    Kelas.Koneksi konn = new Kelas.Koneksi();
```

4. Pada baris berikutnya, buat prosedur refresh\_pelanggan() yang berfungsi untuk memperbaharui data pada DataGridViewPelanggan.

```
//4.membuat prosedur refresh pelanggan
1 reference
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");
```

```
        DataGridPelanggan.DataSource = ds;  
        DataGridPelanggan.DataMember = "pelanggan";  
        DataGridPelanggan.Refresh();  
    }  
    catch (Exception e)  
    {  
        MessageBox.Show(e.ToString());  
    }  
    finally  
    {  
        conn.Close();  
    }  
}
```

5. Pada baris berikutnya, cari kode berikut :

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

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

```
1 reference  
public FCariPelanggan()  
{  
    InitializeComponent();  
    //5.membersihkan komponen inputan dan panggil proc refresh pelanggan  
    TxtCariPelanggan.Clear();  
    refresh_pelanggan();  
}
```

7. Kembali ke desain form cari pelanggan (FCariPelanggan), klik 2x TxtCariPelanggan, lalu ketik kode berikut :

```
1 reference  
private void TxtCariPelanggan_TextChanged(object sender, EventArgs e)  
{  
    //6.membuat perintah pencarian data pelanggan berdasarkan kode dan nama  
    SqlConnection conn = konn.GetConn();  
    conn.Open();  
    cmd = new SqlCommand("select * from pelanggan where NamaPelanggan Like '%" +  
        TxtCariPelanggan.Text + "%' OR KodePelanggan Like '%" +  
        TxtCariPelanggan.Text + "%'", conn);  
    ds = new DataSet();  
    da = new SqlDataAdapter(cmd);  
    da.Fill(ds, "pelanggan");  
    DataGridPelanggan.DataSource = ds;  
    DataGridPelanggan.DataMember = "pelanggan";  
    DataGridPelanggan.Refresh();  
    conn.Close();  
}
```

8. Kembali ke desain form cari pelanggan (FCariPelanggan), klik komponen datagridview DataGridPelanggan, klik 2x pada **Event CellClick**  
Ketik kode berikut :

```
1reference
private void DataGridPelanggan_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //7.mengambil nilai dari record yang dipilih menuju ke variabel global kodeplg dan namaplg
    try
    {
        DataGridViewRow row = this.DataGridPelanggan.Rows[e.RowIndex];
        kodeplg = row.Cells["KodePelanggan"].Value.ToString();
        namaplg = row.Cells["NamaPelanggan"].Value.ToString();
        this.Close(); //menutup form cari pelanggan
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

9. Pada baris berikutnya, buat function untuk mengambil nilai yang berasal dari variabel kodeplg dan namaplg.

```
//8.membuat function ambil kodepelanggan
1reference
public String ambil_kode_pelanggan
{
    get
    {
        return kodeplg;
    }
}
```

```
//9.membuat function ambil namapelanggan
1reference
public String ambil_nama_pelanggan
{
    get
    {
        return namaplg;
    }
}
```

10. Tampilan pada form cari pelanggan (FCariPelanggan) seperti pada gambar 9.1 berikut :

... Data Pelanggan ...

Cari Berdasarkan Kode>Nama Pelanggan :

	KodePelanggan	NamaPelanggan	AlamatPelanggan
▶	P0001	aditya	Bandar Lampung
	P0002	nandi	Lampung Barat
	P0003	indraiat	Lampung Selatan
	P0004	ali	Bandar Lampung
	P0005	aquna p	Lampung Selatan
	P0006	dedi	Lampung Selatan
	P0007	rhendy	Lampung Selatan
	P0008	lulus	Lampung Timur
	P0009	alipi	Bandar Lampung
	P0010	indra	Bandar Lampung

**Gambar 9.1. Tampilan Form Cari Pelanggan**

11. Buka Form Cari Barang (FCariBarang), lalu pindah ke bagian kode program / view code (F7)

12. 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;
```

13. Ketikkan kode berikut di bawah **public partial class FCariBarang : Form {**

```
4 references
public partial class FCariBarang : Form
{
    //2.tambahkan beberapa variabel yang digunakan untuk proses Read Data
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
    public string kodebrg, namabrg, hargabrg = "";

    //3.import kelas koneksi.cs yang berasal dari folder Kelas menjadi objek --> konn
    Kelas.Koneksi konn = new Kelas.Koneksi();
```

14. Pada baris berikutnya, buat prosedur refresh\_barang() yang berfungsi untuk memperbaharui data pada DataGridViewBarang.

```
//4.membuat prosedur refresh barang
1 reference
void refresh_barang()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM barang", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "barang");
        DataGridViewBarang.DataSource = ds;
        DataGridViewBarang.DataMember = "barang";
        DataGridViewBarang.Refresh();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    finally
    {
        conn.Close();
    }
}
```

15. Pada baris berikutnya, cari kode berikut :

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

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

```
1 reference
public FCariBarang()
{
    InitializeComponent();
    //5.membersihkan komponen inputan dan panggil proc refresh barang
    TxtCariBarang.Clear();
    refresh_barang();
}
```

17. Kembali ke desain form cari barang (FCariBarang), klik 2x TxtCariBarang, lalu ketik kode berikut :

```
1 reference
private void TxtCariBarang_TextChanged(object sender, EventArgs e)
{
    //6.membuat perintah pencarian data barang berdasarkan kode atau nama
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("select * from barang where NamaBarang Like '%" +
        TxtCariBarang.Text + "%' OR KodeBarang Like '%" +
        TxtCariBarang.Text + "%'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "barang");
    DataGridBarang.DataSource = ds;
    DataGridBarang.DataMember = "barang";
    DataGridBarang.Refresh();
    conn.Close();
}
```

18. Kembali ke desain form cari barang (FCariBarang), klik komponen datagridview DataGridBarang, klik 2x pada **Event CellClick**  
 Ketik kode berikut :

```
1 reference
private void DataGridBarang_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //7.mengambil nilai dari record yang dipilih menuju ke variabel global
    try
    {
        DataGridViewRow row = this.DataGridBarang.Rows[e.RowIndex];
        kodebrg = row.Cells["KodeBarang"].Value.ToString();
        namabrg = row.Cells["NamaBarang"].Value.ToString();
        hargabrg = row.Cells["HargaBarang"].Value.ToString();
        this.Close(); //menutup form cari barang
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

19. Pada baris berikutnya, buat function untuk mengambil nilai yang berasal dari variabel kodebrg, namabrg dan hargabrg.

```
//8.membuat function ambil kode barang
1 reference
public String ambil_kode_barang
{
    get
    {
        return kodebrg;
    }
}
```

```
//9.membuat function ambil nama barang
1 reference
public String ambil_nama_barang
{
    get
    {
        return namabrg;
    }
}
```

```
//10.membuat function ambil harga barang
1 reference
public String ambil_harga_barang
{
    get
    {
        return hargabrg;
    }
}
```

20. Tampilan pada form cari barang (FCariBarang) seperti pada gambar 9.2 berikut :

... Data Barang ...

Cari Berdasarkan Kode/Nama Barang :

	KodeBarang	NamaBarang	HargaBarang	StokBarang	Sat
▶	B0001	Xiaomi Redmi 5A	1210000	124	pcs
	B0002	Xiaomi Redmi 4X	1650000	99	pcs
	B0003	HP Pavilion 14	3925000	22	pcs
	B0004	Acer Aspire One	3420000	9	pcs
	B0005	Canon EOS 700D	6200000	4	pcs
	B0006	ADATA Premiere	651000	6	pcs
	B0007	RJ45	50000	95	pac
*					

< >

Gambar 9.2. Tampilan Form Cari Barang

## 9.2 Koneksi Form Transaksi Penjualan

1. Buka Form Transaksi (FTransaksi), 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 FTransaksi : Form {**

```
6 references  
public partial class FTransaksi : Form  
{  
    //2. Import Kelas Koneksi dari Folder (Kelas/Koneksi.cs) menjadi objek --> konn  
    Kelas.Koneksi konn = new Kelas.Koneksi();  
  
    //3. Tambahkan Variabel untuk Perintah SQL  
    private SqlCommand cmd;  
    private SqlDataReader rd;  
    private DataSet ds;  
    private SqlDataAdapter da;
```

## 9.3 Membuat Prosedur pada Form Transaksi Penjualan

Pada form transaksi penjualan akan dibuat 8 prosedur yaitu prosedur **bersih**, **awal**, **auto\_number**, **simpan\_MasterPenjualan**, **simpan\_DetailPenjualan**, **refresh\_penjualan**, **RefreshTransaksi**, dan **TotalSeluruh**.

1. Pada baris berikutnya, ketik kode berikut untuk membuat prosedur **bersih** :

```
//4. Prosedur bersih  
1 reference  
private void Bersih()  
{  
    TxtNoKwitansi.Text = "";  
    DtpTanggalKwitansi.Value = DateTime.Now;  
    TxtKodePelanggan.Text = "";  
    TxtNamaPelanggan.Text = "";  
    TxtKodeBarang.Text = "";  
    TxtNamaBarang.Text = "";  
    TxtHargaBarang.Text = "0";  
    TxtJumlahJual.Text = "0";  
    LblTotalBayar.Text = "-";  
}
```

**Prosedur Bersih()** berisi perintah untuk membersihkan komponen inputan (textbox, label, datetimerpicker dan combobox) pada form transaksi.



2. Pada baris berikutnya, ketik kode berikut untuk membuat prosedur **awal** :

*\*prosedur **awal** berisi perintah untuk memanggil prosedur bersih, dan menampilkan data pada DataGridView yang berasal dari tabel kwitansi. Prosedur awal akan digunakan nanti pada saat form transaksi pertama kali aktif.*

```
//5. Prosedur awal
3 references
private void awal()
{
    //panggil prosedur bersih di dalam prosedur awal
    Bersih();
    DataGridPenjualan.Enabled = false; //menonaktifkan datagrid penjualan
    SqlConnection conn = konn.GetConn();
    {
        try
        {
            conn.Open();
            cmd = new SqlCommand("SELECT * FROM kwitansi order by NomorKwitansi DESC", conn);
            ds = new DataSet();
            da = new SqlDataAdapter(cmd);
            da.Fill(ds, "kwitansi");
            DataGridPenjualan.DataSource = ds;
            DataGridPenjualan.DataMember = "kwitansi";
            DataGridPenjualan.Refresh();
        }

        catch (Exception e)
        {
            MessageBox.Show(e.ToString());
        }
        finally
        {
            conn.Close();
        }
    }
}
```

3. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **auto\_number** :

*\*prosedur **auto\_number** berisi perintah untuk membuat penomoran kwitansi secara otomatis*

```
//6. Prosedur auto_number
1 reference
private void auto_number()
{
    long hitung;
    string urut;
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("select NomorKwitansi from kwitansi where NomorKwitansi in(SELECT MAX(NomorKwitansi) from kwitansi) order by NomorKwitansi DESC", conn);
    rd = cmd.ExecuteReader();
    rd.Read();
    if (rd.HasRows) //jika sudah ada data transaksi sebelumnya
    {
        hitung = Convert.ToInt64(rd[0].ToString().Substring(rd["NomorKwitansi"].ToString().Length - 4, 4)) + 1;
        string joinstr = "0000" + hitung;
        urut = "K" + joinstr.Substring(joinstr.Length - 4, 4);
    }
    else //jika belum ada data transaksi
    {
        urut = "K0001";
    }
    rd.Close();
    TxtNoKwitansi.Text = urut;
    conn.Close();
}
```

4. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **Simpan\_MasterPenjualan** :  
*\*prosedur **Simpan\_MasterPenjualan** berisi perintah untuk menyimpan data ke dalam tabel kwitansi menggunakan stored procedure **simpan\_kwitansi** yang ada di SQL Server.*

```
//7. Prosedur Simpan Master Penjualan
1 reference
private void Simpan_MasterPenjualan()
{
    //mempersiapkan sqlconnection
    SqlConnection conn = konn.GetConn();
    {
        //mempersiapkan sqlcommand yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_kwitansi '" + TxtNoKwitansi.Text + "','" +
            + DtpTanggalkwitansi.Text + "','" + TxtKodePelanggan.Text + "'", conn);
        //membuka koneksi database
        conn.Open();
        //menjalankan perintah query
        cmd.ExecuteNonQuery();
    }
}
```

5. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **Simpan\_DetailPenjualan** :  
*\*prosedur **Simpan\_DetailPenjualan** berisi perintah untuk menyimpan data ke dalam tabel detail\_kwitansi menggunakan stored procedure **simpan\_detailkwitansi** yang ada di SQL Server.*

```
//8. Prosedur Simpan Detail Penjualan
2 references
private void Simpan_DetailPenjualan()
{
    SqlConnection conn = konn.GetConn();
    {
        //mempersiapkan sqlcommand yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_detailkwitansi '" + TxtNoKwitansi.Text + "','" +
            + TxtKodeBarang.Text + "','" + TxtJumlahJual.Text + "'", conn);
        //membuka koneksi database
        conn.Open();
        //menjalankan perintah query
        cmd.ExecuteNonQuery();
    }
}
```

6. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **refresh\_penjualan** :

```
//9. Prosedur refresh_penjualan
1 reference
private void refresh_penjualan()
{
    SqlConnection conn = konn.GetConn();
    {
        try
        {
            conn.Open();
            /*note : buka microsoft sqlserver, pastikan sudah membuat view_detailkwitansi
            cmd = new SqlCommand("SELECT * FROM view_detailkwitansi where NomorKwitansi= '" +
                + TxtNoKwitansi.Text + "'", conn);
            ds = new DataSet();
            da = new SqlDataAdapter(cmd);
            da.Fill(ds, "view_detailkwitansi");
        }
    }
}
```

```

        DataGridPenjualan.DataSource = ds;
        DataGridPenjualan.DataMember = "view_detailkwitansi";
        DataGridPenjualan.Refresh();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    finally
    {
        conn.Close();
    }
}

```

7. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **RefreshTransaksi** :

*\*prosedur **RefreshTransaksi** berisi perintah untuk membersihkan beberapa komponen input pada form terkait transaksi penjualan barang*

```

//10. Prosedur Refresh Transaksi Per Barang
2 references
private void RefreshTransaksi()
{
    //memanggil proc refresh penjualan dan bersihkan beberapa komponen transaksi barang
    refresh_penjualan();
    TxtKodeBarang.Clear();
    TxtNamaBarang.Clear();
    TxtHargaBarang.Text = "0";
    TxtJumlahJual.Text = "0";
    TxtKodeBarang.Focus();
}

```

8. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur **totalseluruh** :

*\*prosedur **totalseluruh** berisi perintah untuk menghitung total bayar yang didapat dari jumlah keseluruhan kolom jumlah bayar per nomor kwitansi, total bayar yang didapat akan ditampilkan pada label LblTotalBayar dalam format rupiah.*

```

//11. Prosedur refresh totalseluruh
3 references
private void totalseluruh()
{
    SqlConnection conn = konn.GetConn();
    {
        conn.Open();
        cmd = new SqlCommand("select sum(JumlahBayar) as TotalBayar " +
            " from view_detailkwitansi where NomorKwitansi= '" + TxtNoKwitansi.Text + "' ", conn);
        cmd.Connection = conn;
        int result = ((int)cmd.ExecuteScalar());
        String TotalSeluruh = result.ToString();
        LblTotalBayar.Text = result.ToString("#,##0");
        conn.Close();
    }
}

```

## 9.4 Coding pada Form Transaksi Penjualan

1. Kembali ke desain Form Transaksi, klik 2 kali **button browse cari pelanggan (BtnBrowsePelanggan)**

Ketik kode berikut :



```
1 reference
private void BtnBrowsePelanggan_Click(object sender, EventArgs e)
{
    //12.memanggil form cari pelanggan
    FCariPelanggan cariplg = new FCariPelanggan();
    cariplg.ShowDialog();
    TxtKodePelanggan.Text = cariplg.ambil_kode_pelanggan;
    TxtNamaPelanggan.Text = cariplg.ambil_nama_pelanggan;
    TxtKodeBarang.Focus();
}
```

2. Kembali ke desain Form Transaksi, klik 2 kali **button browse cari barang (BtnBrowseBarang)**

Ketik kode berikut :

```
1 reference
private void BtnBrowseBarang_Click(object sender, EventArgs e)
{
    //13.memanggil form cari barang
    FCariBarang caribrg = new FCariBarang();
    caribrg.ShowDialog();
    TxtKodeBarang.Text = caribrg.ambil_kode_barang;
    TxtNamaBarang.Text = caribrg.ambil_nama_barang;
    TxtHargaBarang.Text = caribrg.ambil_harga_barang;
    TxtJumlahJual.Focus();
}
```

3. Kembali ke desain form Transaksi, klik 2 kali button **Baru**, ketik kode berikut :

```
1 reference
private void BtnBaru_Click(object sender, EventArgs e)
{
    //14. memanggil prosedur awal dan auto_number ketika button baru dijalankan
    awal();
    auto_number();
    DtpTanggalKwitansi.Focus();
}
```

4. Kembali ke desain Form Transaksi, klik 2 kali **button Tambah (BtnTambah)**

Ketik kode berikut :



```
1 reference
private void BtnTambah_Click(object sender, EventArgs e)
{
    //15.SIMPAN DATA PENJUALAN PER BARANG (BUAT TRIGGER DAHULU DI SQL SERVER UNTUK AUTOMASI STOK)
    DataGridPenjualan.Enabled = true; //mengaktifkan datagrid penjualan
    SqlDataReader reader = null;
    SqlConnection conn = konn.GetConn();
    {
        //jika ada data yang kosong
        if (TxtNoKwitansi.Text.Trim() == "" || TxtKodePelanggan.Text.Trim() == "" ||
            TxtKodeBarang.Text.Trim() == "" || TxtJumlahJual.Text.Trim() == "0")
        {
            //pesan data belum lengkap
            MessageBox.Show("Data belum lengkap, lengkapi data", "Peringatan", MessageBoxButtons.OK,
                MessageBoxIcon.Exclamation);
        }
    }
}
```

```
//jika data sudah terisi semua
else
{
    conn.Open();
    cmd = new SqlCommand("select * from kwitansi where NomorKwitansi = '" +
        TxtNoKwitansi.Text + "'", conn);
    cmd.ExecuteNonQuery();
    reader = cmd.ExecuteReader();
    if (reader.Read()) //jika data penjualan sudah ada
    {
        //refresh koneksi
        conn.Close();
        conn.Open();

        cmd = new SqlCommand("select * from detail_kwitansi where NomorKwitansi = '" +
            TxtNoKwitansi.Text + "' AND KodeBarang = '" + TxtKodeBarang.Text + "'", conn);
        cmd.ExecuteNonQuery();
        reader = cmd.ExecuteReader();
    }
}
```

```
if (reader.Read()) //jika data barang sudah diinputkan ke tabel detail_kwitansi
{
    MessageBox.Show("Data barang "+TxtNamaBarang.Text+" sudah diinputkan, harap pilih
        barang yang lain", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
else //jika data barang belum diinputkan ke tabel detail_kwitansi
{
    //panggil prosedur detail penjualan
    Simpan_DetailPenjualan();
    //panggil prosedur totalseluruh
    totalseluruh();
}
}
else //jika data penjualan belum ada
{
    //panggil prosedur simpan master dan detail penjualan
    Simpan_MasterPenjualan();
    Simpan_DetailPenjualan();
    //panggil prosedur totalseluruh
    totalseluruh();
}
//panggil prosedur refresh transaksi
RefreshTransaksi();
}
}
```

5. Kembali ke desain Form Transaksi, klik 2 kali **button Kurang (BtnKurang)**

Ketik kode berikut :



1 reference

```
private void BtnKurang_Click(object sender, EventArgs e)
{
    //16.HAPUS DATA PENJUALAN PER BARANG
    //mempersiapkan koneksi database
    SqlDataReader reader = null;
    SqlConnection conn = konn.GetConn();
    {
        //query menghapus data berdasarkan NomorKwitansi dan KodeBarang pada tbl detail
        cmd = new SqlCommand("EXECUTE hapus_detailkwitansi '" + TxtNoKwitansi.Text + "', '" +
            TxtKodeBarang.Text + "'", conn);
        //buka koneksi database
        conn.Open();
        //eksekusi query
        cmd.ExecuteNonQuery();
    }
}
```

```
//CEK DATA DI DETAIL KWITANSI
cmd = new SqlCommand("select * from detail_kwitansi where NomorKwitansi = '"
    + TxtNoKwitansi.Text + "'", conn);
cmd.ExecuteNonQuery();
reader = cmd.ExecuteReader();
if (reader.Read()) //jika ada data detail kwitansi
{
    //panggil prosedur totalseluruh
    totalseluruh();
}
else //jika belum ada data detail kwitansi
{
    LblTotalBayar.Text = "0";
}
//memanggil proc refresh transaksi
RefreshTransaksi();
}
```

6. Kembali ke desain form Transaksi, klik **event CellClick** pada DataGridViewPenjualan  
 Ketik kode berikut :

```
1 reference
private void DataGridViewPenjualan_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //17.TAMPILKAN DATA PENJUALAN PER BARANG
    DataGridViewRow row = this.DataGridViewPenjualan.Rows[e.RowIndex];
    TxtKodeBarang.Text = row.Cells["KodeBarang"].Value.ToString();
    TxtNamaBarang.Text = row.Cells["NamaBarang"].Value.ToString();
    TxtHargaBarang.Text = row.Cells["HargaBarang"].Value.ToString();
    TxtJumlahJual.Text = row.Cells["JumlahJual"].Value.ToString();
}
```

7. Kembali ke desain form Transaksi, klik 2 kali **button Simpan (BtnSimpan)**, ketik kode berikut :

```
1 reference
private void BtnSimpan_Click(object sender, EventArgs e)
{
    //18. transaksi selesai dan cetak laporan
    MessageBox.Show("Transaksi Selesai");
    awal();
}
```

8. Kembali ke desain form Transaksi, klik 2 kali **button Keluar (BtnKeluar)**, ketik kode berikut :

```
1 reference
private void BtnKeluar_Click(object sender, EventArgs e)
{
    //19. keluar form transaksi
    Close();
}
```

9. Masih di mode view code, cari kode berikut

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

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

```
1 reference
public FTransaksi()
{
    InitializeComponent();
    //20. memanggil prosedur awal
    awal();
}
```

11. **Running program**, gambar 9.3 merupakan tampilan form transaksi yang sudah terkoneksi dengan (database) tabel kwitansi dan detail\_kwitansi

NomorKwitansi	TanggalKwitansi	KodePelang
K0006	4/26/2018	P0006
K0005	4/24/2018	P0002
K0004	4/21/2018	P0003
K0003	4/20/2018	P0003
K0002	4/20/2018	P0002
K0001	4/20/2018	P0001

**Gambar 9.3. Form Transaksi yang Terkoneksi Database**