

# **PROGRAM PENJUALAN KUE**

## **ANGGOTA KELOMPOK**

Nama : Riski Aji Kusuma  
NPM : 17312003

Nama : Panji Setiawan Munandar  
NPM : 17312123



**INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS TEKNOKRAT INDONESIA  
BANDAR LAMPUNG  
2018-2019**

# **BAB 1 PENDAHULUAN**

## **1.1 LATAR BELAKANG**

Studi kasus program ini adalah toko yang bergerak dalam bidang penjualan kue. Dalam menjalankan rutinitasnya, toko ini masih menggunakan sistem manual sehingga sering menimbulkan masalah antara lain kesalahan dalam perhitungan penjualan, ketidakakuratan dalam perhitungan stok, kurangnya pengawasan terhadap karyawan dan lain-lain.

## **1.2 RUMUSAN MASALAH**

Berdasarkan latar belakang, dapat di ambil rumusan masalah sebagai berikut :

1. Bagaimana membuat suatu program yang dapat membantu transaksi penjualan pada toko ABC.
2. Bagaimana membuat perhitungan penjualan dan perhitungan penambahan stok.
3. Bagaimana membuat suatu program secara otomatis dapat menampilkan stok barang.

## **1.3 BATASAN MASALAH**

Keunggulan :

1. Dapat memilih menu kue yang tertera, dengan jumlah yang di inginkan.
2. Dapat mengurangi dan menambahkan menu yang telah di pesan .

Kekurangan :

1. Dalam program ini pelanggan harus ke toko langsung untuk melakukan transaksi.
2. Aplikasi dibuat hanya untuk digunakan pada toko ABC.

## **1.4 TUJUAN DAN MANFAAT**

1. Untuk membuat suatu program yang dapat membantu transaksi penjualan pada toko ABC.
2. Untuk membuat perhitungan penjualan dan perhitungan penambahan stok.
3. Untuk membuat suatu program secara otomatis dapat menampilkan stok barang.

Manfaat dari program ini adalah dapat mengetahui kekurangan dan kelemahan yang ada pada sistem yang sedang berjalan. sehingga dapat diatasi dan mampu menghindarkan kesalahan dalam perhitungan ataupun kesalahan dalam penyajian laporan. Selain itu dapat memberikan informasi yang benar dan tepat waktu kepada pihak manajemen sehingga dapat menunjang pengambilan keputusan yang terbaik dan perancangan program ini dapat berguna bagi toko kue dalam meningkatkan efektifitas dan efisiensi dalam siklus penjualan.

## **BAB 2 TEORI**

### **2.1 DATABASE (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.

### **2.2 PEMROGRAMAN VISUAL C#**

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).

## **BAB 3 PERANCANGAN**

### **3.1 PROSES BISNIS**

Dalam studi kasus ini, aplikasi yang dibuat berbasis desktop. Aplikasi yang dibuat **tidak membahas** bagaimana proses pembuatan kue pada toko ABC. Jadi, **hanya membahas** bagaimana proses penjualan kue yang disuplai oleh supplier.

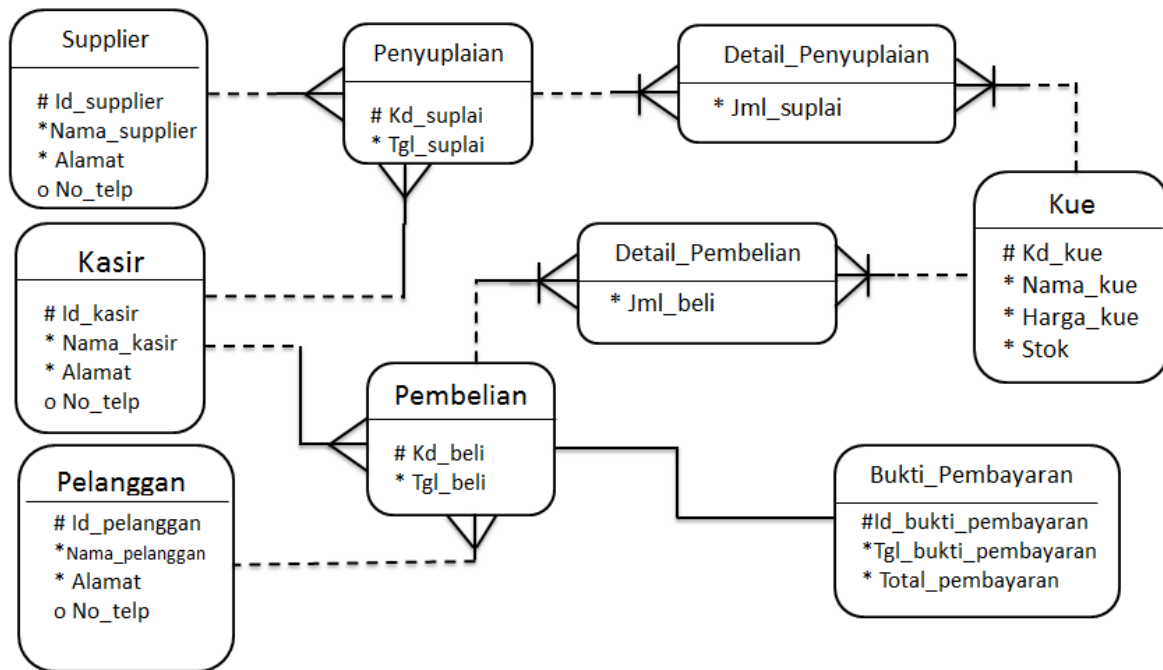
Ketentuan proses bisnisnya, sebagai berikut :

#### **Transaksi penyuplaian antara supplier dengan kasir :**

1. Supplier akan datang ke toko untuk melakukan transaksi penyuplaian.
2. Kasir akan menginputkan data supplier dengan data berupa id supplier, nama supplier, alamat, dan nomor telepon.
3. Selanjutnya kasir menginputkan semua data kue yang telah dibawa supplier dengan data berupa kode kue, nama kue, harga kue, dan stok.
4. Satu supplier boleh melakukan banyak transaksi penyuplaian.
5. Satu transaksi penyuplaian dilakukan oleh satu supplier.
6. Satu transaksi penyuplaian boleh terdiri dari banyak kue.
7. Satu kue boleh terdapat pada beberapa transaksi penyuplaian.

#### **Transaksi pembelian antara pelanggan dengan kasir :**

1. Pelanggan akan datang ke toko untuk melakukan transaksi pembelian.
2. Kasir akan menginputkan data pelanggan. Pelanggan memiliki data berupa id pelanggan, nama pelanggan, alamat, dan nomor telepon.
3. Satu pelanggan boleh melakukan beberapa transaksi pembelian.
4. Satu transaksi pembelian dilakukan oleh satu pelanggan.
5. Selanjutnya pelanggan akan memilih kue yang ingin dibeli.
6. Kasir akan menginputkan data kue yang ingin dibeli berdasarkan kode kue.
7. Satu transaksi pembelian boleh terdiri dari banyak kue.
8. Satu kue boleh terdapat pada beberapa transaksi pembelian.
9. Kemudian mesin secara otomatis mencetak bukti pembayaran.
10. Satu transaksi pembelian memiliki satu bukti pembayaran.
11. Satu bukti pembayaran dimiliki satu transaksi pembelian.



**Gambar ERD Bisnis Penjualan Toko Kue**

### 3.2 Membuat Tabel

#### 1. Ketik query berikut untuk membuat database

```
--Membuat database
CREATE DATABASE db_penjualan_kue
--Menggunakan database
USE db_penjualan_kue;
```

#### 2. Membuat Table

```
-- Membuat tabel kasir --
CREATE TABLE kasir
(
    Id_kasir CHAR(5) PRIMARY KEY,
    Nama_kasir VARCHAR(25) NOT NULL,
    Alamat_kasir VARCHAR(50) NOT NULL,
    No_telp VARCHAR(15)
)
```

```
-- Membuat tabel supplier --
CREATE TABLE supplier
(
    Id_supplier CHAR(5) PRIMARY KEY,
    Nama_supplier VARCHAR(25) NOT NULL,
    Alamat_supplier VARCHAR(50) NOT NULL,
    No_telp VARCHAR(15)
)
```

```
-- Membuat tabel pelanggan --
CREATE TABLE pelanggan
( Id_pelanggan CHAR(5) PRIMARY KEY,
  Nama_pelanggan VARCHAR(25) NOT NULL,
  Alamat_pelanggan VARCHAR(50) NOT NULL,
  No_telp VARCHAR(15))
```

```
-- Membuat tabel kue --
CREATE TABLE kue
(Kd_kue CHAR(5) PRIMARY KEY,
 Nama_kue VARCHAR(25) NOT NULL,
 Harga_kue INT NOT NULL,
 Jenis_kue VARCHAR(30) NOT NULL,
 Stok_kue INT NOT NULL )
```

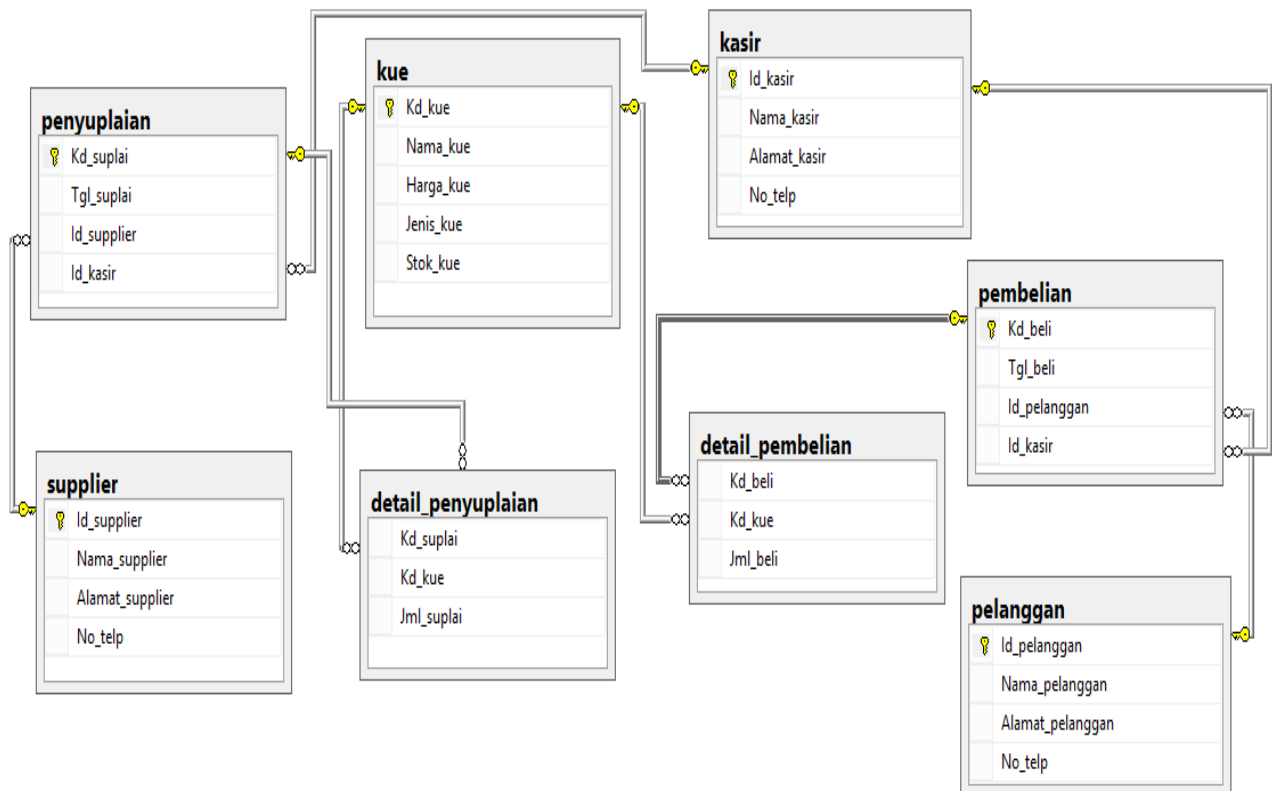
```
-- Membuat tabel transaksi penyuplaian --
CREATE TABLE penyuplaian
( Kd_suplai CHAR(5) PRIMARY KEY,
  Tgl_suplai DATE NOT NULL,
  Id_supplier CHAR(5) FOREIGN KEY REFERENCES
  supplier(Id_supplier),
  Id_kasir CHAR(5) FOREIGN KEY REFERENCES
  kasir(Id_kasir))
```

```
-- Membuat tabel transaksi pembelian --
CREATE TABLE pembelian
(Kd_beli CHAR(5) PRIMARY KEY,
 Tgl_beli DATE NOT NULL,
 Id_pelanggan CHAR(5) FOREIGN KEY REFERENCES
 pelanggan(Id_pelanggan),
 Id_kasir CHAR(5) FOREIGN KEY REFERENCES
 kasir(Id_kasir))
```

```
-- Membuat tabel detail_penyuplaian -
CREATE TABLE detail_penyuplaian
(Kd_suplai CHAR(5) FOREIGN KEY REFERENCES
 penyuplaian(Kd_suplai),
 Kd_kue CHAR(5) FOREIGN KEY REFERENCES
 kue(Kd_kue),
 Jml_suplai INT NOT NULL)
```

```
-- Membuat tabel detail_pembelian --
CREATE TABLE detail_pembelian
(Kd_beli CHAR(5) FOREIGN KEY REFERENCES
 pembelian(Kd_beli),
 Kd_kue CHAR(5) FOREIGN KEY REFERENCES
 kue(Kd_kue),
 Jml_beli INT NOT NULL)
```

3. Skema Diagram dari tabel yang di buat sebagai berikut :



Gambar Skema Database Diagram

4. kembali ke tab Query, lalu masukan dan execute perintah berikut :

```

-- Memasukkan data supplier --
INSERT INTO supplier VALUES
('S0001', 'Andik', 'Way Halim', '089672762170'),
('S0002', 'Putra', 'Pramuka', '081232412567'),
('S0003', 'Law', 'Kemiling', '082281525383'),
('S0004', 'Sandi', 'Pahoman', '085711989435'),
('S0005', 'Tama', 'Rajabasa', '0896452098989')
-- Memasukkan data kasir --
INSERT INTO kasir VALUES
('K0001', 'Riski', 'Gunung Terang', '081272762172'),
('K0002', 'Ajik', 'Palapa', '0857324125334'),
('K0003', 'Kusuma', 'Teluk', '081281525383'),
('K0004', 'Panjik', 'PU', '085711989477'),
('K0005', 'Setiawan', 'Kedaton', '0896452098912')
-- Memasukkan data pelanggan --
INSERT INTO pelanggan VALUES
('P0001', 'Putri', 'Kedaton', '089672762171'),
('P0002', 'Siti', 'Palapa', '081232412563'),
('P0003', 'Rara', 'Teluk', '082281525343'),
('P0004', 'Inka', 'PU', '085711989499'),
('P0005', 'Ririn', 'Antasari', '0896452098901'),
('P0006', 'Joko', 'Kedaton', '0812452098982'),
('P0007', 'Otong', 'PU', '0822452098923'),
('P0008', 'Dicky', 'Antasari', '085745209891'),
('P0009', 'Rio', 'Gedong Aer', '0812452098985'),
('P0010', 'Sherly', 'Teluk', '0812452098555')
  
```

5. Buka aplikasi Ms. Office Excel, lalu buat data sebagai berikut :  
 Jangan lupa untuk mengubah ekstensi file menjadi .csv (comma delimited)

Kd_kue	Nama_kue	Harga_kue	Jenis_kue	Stok_kue
KU001	Bolu Coklat	94000	Kue Bolu	20
KU002	Bolu Keju	11000	Kue Bolu	40
KU003	Kue Bolu Pandan	98000	Kue Bolu	53
KU004	Bolu Kukus Pelangi	90000	Kue Bolu	12
KU005	Black Forest	310000	kue Ulang Tahun	75
KU006	Cheese Cake	450000	kue Ulang Tahun	23
KU007	Brownis Selai Kacang	45000	Kue Brownis	20
KU008	Red Velvet	500000	kue Ulang Tahun	15
KU009	Brownis RHUM Raisin	75000	Kue Brownis	30
KU010	Brownis Tutup Keju	65000	Kue Brownis	25

6. Kembali ke tab Query :

```
-- Memasukkan data kue dari MS.Excel --
BULK INSERT kue
FROM 'E:\Program_PBD_2018\datakue.csv'
WITH
(
  FIRSTROW = 2,
  FIELDTERMINATOR = ';',
  ROWTERMINATOR = '\n'
)
GO
```

Alamat penyimpanan file .csv

### 3.3 Membuat View

View merupakan virtual table, view berisi query yang dapat dipanggil berulang kali pada saat di butuhkan (Forta, 2007 ). View yang akan dibuat akan berisi query join untuk menggabungkan table pada database.

1. Ketik Query sebagai berikut :

```
-- Membuat view penyuplaian --
CREATE VIEW view_penyuplaian AS
SELECT detail_penyuplaian.Kd_suplai, penyuplaian.Tgl_suplai,
supplier.Id_supplier, supplier>Nama_supplier,
kasir.Id_kasir, kasir>Nama_kasir,
detail_penyuplaian.Kd_kue, kue>Nama_kue, kue>Jenis_kue,
kue.Harga_kue, detail_penyuplaian.jml_suplai
FROM detail_penyuplaian
JOIN penyuplaian
ON penyuplaian.Kd_suplai = detail_penyuplaian.Kd_suplai
JOIN kue
ON kue.Kd_kue = detail_penyuplaian.Kd_kue
JOIN supplier
ON supplier.Id_supplier = penyuplaian.Id_supplier
JOIN kasir
ON kasir.Id_kasir = penyuplaian.Id_kasir
```



2. Sebelum menampilkan view\_penyuplai, tambahkan terlebih dahulu data kedalam data penyuplai dan detail\_penyuplai, sebagai berikut :

```
-- Memasukkan data penyuplaian --
INSERT INTO penyuplaian VALUES
('PE001', '2018/08/08', 'S0001', 'K0001'),
('PE002', '2018/08/15', 'S0002', 'K0002'),
('PE003', '2018/08/22', 'S0003', 'K0003'),
('PE004', '2018/08/29', 'S0004', 'K0004'),
('PE005', '2018/08/05', 'S0005', 'K0005')
```

```
-- Memasukkan data detail penyuplaian --
INSERT INTO detail_penyuplaian VALUES
('PE001', 'KU001', 5),
('PE002', 'KU002', 10),
('PE003', 'KU003', 15),
('PE003', 'KU004', 8),
('PE003', 'KU005', 2),
('PE004', 'KU006', 6),
('PE004', 'KU007', 5),
('PE004', 'KU008', 7),
('PE005', 'KU009', 8),
('PE005', 'KU010', 10)
```

```
-- Memanggil view --
SELECT * FROM view_penyuplaian;
```

```
-- Membuat view detail penyuplaian --
CREATE VIEW view_detailpenyuplaian AS
SELECT detail_penyuplaian.Kd_suplai, detail_penyuplaian.Kd_kue,
kue>Nama_kue, kue.Harga_kue, detail_penyuplaian.jml_suplai
FROM kue INNER JOIN
detail_penyuplaian ON kue.Kd_kue = detail_penyuplaian.Kd_kue;
```

```
SELECT * FROM view_detailpenyuplaian;
```

```
-- Membuat view pembelian --
CREATE VIEW view_pembelian AS
SELECT detail_pembelian.Kd_beli, pembelian.Tgl_beli,
pelanggan.Id_pelanggan, pelanggan>Nama_pelanggan,
kasir.Id_kasir, kasir>Nama_kasir,
detail_pembelian.Kd_kue, kue>Nama_kue, kue.Jenis_kue,
kue.Harga_kue, detail_pembelian.jml_beli,
kue.Harga_kue * detail_pembelian.Jml_beli AS Total_bayar
FROM detail_pembelian
JOIN pembelian
ON pembelian.Kd_beli = detail_pembelian.Kd_beli
JOIN kue
ON kue.Kd_kue = detail_pembelian.Kd_kue
JOIN pelanggan
ON pelanggan.Id_pelanggan = pembelian.Id_pelanggan
JOIN kasir
ON kasir.Id_kasir = pembelian.Id_kasir
```

```
-- Memasukkan data pembelian --
INSERT INTO pembelian VALUES
('PM001', '2018/08/15', 'P0001', 'K0001'),
('PM002', '2018/08/22', 'P0002', 'K0002'),
('PM003', '2018/08/25', 'P0003', 'K0003'),
('PM004', '2018/08/26', 'P0004', 'K0004'),
('PM005', '2018/08/29', 'P0005', 'K0005')
```

```
-- Memasukkan data detail pembelian --
INSERT INTO detail_pembelian VALUES
('PM001','KU001',10),
('PM002','KU002',1),
('PM003','KU003',5),
('PM003','KU004',2),
('PM003','KU005',3),
('PM004','KU006',4),
('PM004','KU007',6),
('PM004','KU008',9),
('PM005','KU009',10),
('PM005','KU010',7)
```

```
SELECT * FROM view_pembelian;
```

```
-- Membuat view detail pembelian --_
CREATE VIEW view_detailpembelian AS
SELECT detail_pembelian.Kd_beli, detail_pembelian.Kd_kue,
kue>Nama_kue, kue.Harga_kue, detail_pembelian.Jml_beli
FROM kue INNER JOIN
detail_pembelian ON kue.Kd_kue = detail_pembelian.Kd_kue;

SELECT * FROM view_detailpembelian;
```

### 3.4 Membuat Stored Procedure

Stored Procedure merupakan kumpulan dari satu atau lebih perintah SQL yang dapat dieksekusi lebih cepat (Forti,2007) , Berikut langkah-langkah untuk membuat Stored Procedure :

#### 1. Ketik dan execute

```
-- Membuat prosedur SP_View_Penyuplaian --
CREATE PROCEDURE SP_View_Penyuplaian AS
BEGIN
    SELECT * FROM view_penyuplaian;
END;
```

```
EXECUTE SP_View_Penyuplaian;
```

```
-- Membuat prosedur SP_View_Pembelian --
CREATE PROCEDURE SP_View_Pembelian AS
BEGIN
    SELECT * FROM view_pembelian;
END;
```

```
EXECUTE SP_View_Pembelian;
```

#### 2. Membuat Store Procedure Simpan\_supplier, hapus supplier, ubah supplier, cari supplier, sebagai berikut :

```
-- Membuat prosedur simpan_supplier --
CREATE PROCEDURE simpan_supplier
@Id_supplier CHAR(5),
@Nama_supplier VARCHAR(25),
@Alamat_supplier VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
    INSERT INTO supplier VALUES
```

```
(@Id_supplier, @Nama_supplier, @Alamat_supplier,
  @No_telp);
END;
```

```
-- Membuat prosedur hapus_supplier --
CREATE PROCEDURE hapus_supplier
@Id_supplier CHAR(5)
AS
BEGIN
  DELETE FROM supplier WHERE Id_supplier = @Id_supplier;
END;
```

```
-- Membuat prosedur ubah_supplier --
CREATE PROCEDURE ubah_supplier
@Id_supplier CHAR(5),
@Nama_supplier VARCHAR(25),
@Alamat_supplier VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
  UPDATE supplier SET Nama_supplier = @Nama_supplier,
    Alamat_supplier = @Alamat_supplier, No_telp = @No_telp
  WHERE Id_supplier = @Id_supplier;
END;
```

```
-- Membuat prosedur cari_supplier --
CREATE PROCEDURE cari_supplier
@Nama_supplier VARCHAR(25)
AS
BEGIN
  SELECT * FROM supplier
  WHERE Nama_supplier LIKE '%' + @Nama_supplier + '%';
END;
```

2. Membuat Store Procedure simpan\_pelanggan, hapus pelanggan, ubah pelanggan, cari pelanggan, sebagai berikut :

```
-- Membuat prosedur simpan_pelanggan --
CREATE PROCEDURE simpan_pelanggan
@Id_pelanggan CHAR(5),
@Nama_pelanggan VARCHAR(25),
@Alamat_pelanggan VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
  INSERT INTO pelanggan VALUES
    (@Id_pelanggan, @Nama_pelanggan, @Alamat_pelanggan,
    @No_telp);
END;
```

```
-- Membuat prosedur hapus_pelanggan --
CREATE PROCEDURE hapus_pelanggan
@Id_pelanggan CHAR(5)
AS
BEGIN
  DELETE FROM pelanggan WHERE Id_pelanggan = @Id_pelanggan;
END;
```

```
-- Membuat prosedur ubah_pelanggan --
CREATE PROCEDURE ubah_pelanggan
@Id_pelanggan CHAR(5),
@Nama_pelanggan VARCHAR(25),
@Alamat_pelanggan VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
    UPDATE pelanggan SET Nama_pelanggan = @Nama_pelanggan,
        Alamat_pelanggan = @Alamat_pelanggan, No_telp = @No_telp
    WHERE Id_pelanggan = @Id_pelanggan;
END;
```

```
-- Membuat prosedur cari_pelanggan --
CREATE PROCEDURE cari_pelanggan
@Nama_pelanggan VARCHAR(25)
AS
BEGIN
    SELECT * FROM pelanggan
    WHERE Nama_pelanggan LIKE '%' + @Nama_pelanggan + '%';
END;
```

3. Membuat Store Procedure simpan\_kasir, hapus kasir, ubah kasir, cari kasir, sebagai berikut :

```
-- Membuat prosedur simpan_kasir --
CREATE PROCEDURE simpan_kasir
@Id_kasir CHAR(5),
@Nama_kasir VARCHAR(25),
@Alamat_kasir VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
    INSERT INTO kasir VALUES
        (@Id_kasir, @Nama_kasir, @Alamat_kasir, @No_telp);
END;
```

```
-- Membuat prosedur hapus_kasir --
CREATE PROCEDURE hapus_kasir
@Id_kasir CHAR(5)
AS
BEGIN
    DELETE FROM kasir WHERE Id_kasir = @Id_kasir;
END;
```

```
-- Membuat prosedur ubah_kasir --
CREATE PROCEDURE ubah_kasir
@Id_kasir CHAR(5),
@Nama_kasir VARCHAR(25),
@Alamat_kasir VARCHAR(50),
@No_telp VARCHAR(15)
AS
BEGIN
    UPDATE kasir SET Nama_kasir = @Nama_kasir,
        Alamat_kasir = @Alamat_kasir, No_telp = @No_telp
    WHERE Id_kasir = @Id_kasir;
END;
```

```
-- Membuat prosedur cari_kasir --
CREATE PROCEDURE cari_kasir
@Nama_kasir VARCHAR(25)
AS
BEGIN
    SELECT * FROM kasir
    WHERE Nama_kasir LIKE '%' + @Nama_kasir + '%';
END;
```

3. Membuat Store Procedure simpan\_kue, hapus kue, ubah kue, cari kue, sebagai berikut :

```
-- Membuat prosedur simpan_kue --
CREATE PROCEDURE simpan_kue
@Kd_kue CHAR(5),
@Nama_kue VARCHAR(25),
@Harga_kue INT,
@Jenis_kue VARCHAR(15),
@Stok INT
AS
BEGIN
    INSERT INTO kue VALUES
    (@Kd_kue, @Nama_kue, @Harga_kue, @Jenis_kue, @Stok);
END;
```

```
-- Membuat prosedur hapus_kue --
CREATE PROCEDURE hapus_kue
@Kd_kue CHAR(5)
AS
BEGIN
    DELETE FROM kue WHERE Kd_kue = @Kd_kue;
END;
```

```
-- Membuat prosedur ubah_kue --
ALTER PROCEDURE ubah_kue
@Nama_kue VARCHAR(25),
@Harga_kue INT,
@Jenis_kue VARCHAR(15),
@Stok_kue INT,
@Kd_kue CHAR(5)
AS
BEGIN
    UPDATE kue SET Nama_kue = @Nama_kue,
    Harga_kue = @Harga_kue, Jenis_kue = @Jenis_kue,
    Stok_kue = @Stok_kue WHERE Kd_kue = @Kd_kue;
END;
```

```
-- Membuat prosedur cari_kue --
CREATE PROCEDURE cari_kue
@Nama_kue VARCHAR(25)
AS
BEGIN
    SELECT * FROM kue
    WHERE Nama_kue LIKE '%' + @Nama_kue + '%';
END;
```

#### 4. Membuat Store Procedure simpan penyuplai :

```
-- Membuat prosedur simpan penyuplaian --
CREATE PROCEDURE simpan_penyuplaian
@Kd_suplai CHAR(5),
@Tgl_suplai DATE,
@Id_supplier CHAR(5),
@Id_kasir CHAR(5)
AS
BEGIN
    INSERT INTO penyuplaian VALUES
    (@Kd_suplai, @Tgl_suplai, @Id_supplier, @Id_kasir);
END;
```

#### 5. Membuat Store Procedure simpan\_detail\_penyuplaian, hapus\_detail\_penyuplaian, sebagai berikut :

```
-- Membuat prosedur simpan_detail_penyuplaian --
CREATE PROCEDURE simpan_detail_penyuplaian
@Kd_suplai CHAR(5),
@Kd_kue CHAR(5),
@Jml_suplai INT
AS
BEGIN
    INSERT INTO detail_penyuplaian VALUES
    (@Kd_suplai, @Kd_kue, @Jml_suplai);
END;
```

```
-- Membuat prosedur hapus_detail_penyuplaian --
CREATE PROCEDURE hapus_detail_penyuplaian
@Kd_suplai CHAR(5),
@Kd_kue CHAR(5)
AS
BEGIN
    DELETE FROM detail_penyuplaian WHERE
    Kd_suplai = @Kd_suplai AND Kd_kue = @Kd_kue;
END;
```

#### 6. Membuat Store Procedure simpan pembelian :

```
-- Membuat prosedur simpan pembelian --
CREATE PROCEDURE simpan_pembelian
@Kd_beli CHAR(5),
@Tgl_beli DATE,
@Id_pelanggan CHAR(5), @Id_kasir CHAR(5)
AS
BEGIN
    INSERT INTO pembelian VALUES
    (@Kd_beli, @Tgl_beli, @Id_pelanggan, @Id_kasir);
END;
```

#### 7. Membuat Store Procedure simpan\_detail\_pembelian, sebagai berikut :

```
-- Membuat prosedur simpan_detail_pembelian --
CREATE PROCEDURE simpan_detail_pembelian
@Kd_beli CHAR(5),
@Kd_kue CHAR(5), @Jml_beli INT
AS
BEGIN
    INSERT INTO detail_pembelian VALUES
    (@Kd_beli, @Kd_kue, @Jml_beli);
END;
```

```
-- Membuat prosedur hapus_detail_pembelian --
CREATE PROCEDURE hapus_detail_pembelian
@Kd_beli CHAR(5),
@Kd_kue CHAR(5)
AS
BEGIN
    DELETE FROM detail_pembelian WHERE
        Kd_beli = @Kd_beli AND Kd_kue = @Kd_kue;
END;
```

### 3.5 Membuat Triger

Triger merupakan kumpulan pernyataan SQL yang berada di antara BEGIN dan END yang secara otomatis akan dijalankan oleh SQL SERVER sebagai respon dari kondisi AFTER INSERT, DELETE dan UPDATE.

```
-- Membuat trigger untuk penambahan stok otomatis --
CREATE TRIGGER AfterInsert_DetailPenyuplaian
ON detail_penyuplaian
AFTER INSERT
AS
BEGIN
    DECLARE
        @Jumlah INT, @Kode CHAR(5)
    SELECT @Jumlah = Jml_suplai,
           @Kode = Kd_kue FROM inserted

    UPDATE kue SET kue.Stok_kue =
        kue.Stok_kue + @Jumlah
    WHERE kue.Kd_kue = @Kode;
END;
```

```
-- Membuat trigger untuk pengurangan stok otomatis --
CREATE TRIGGER AfterInsert_DetailPembelian
ON detail_pembelian
AFTER INSERT
AS
BEGIN
    DECLARE
        @Jumlah INT, @Kode CHAR(5)
    SELECT @Jumlah = Jml_beli,
           @Kode = Kd_kue FROM inserted

    UPDATE kue SET kue.Stok_kue =
        kue.Stok_kue - @Jumlah
    WHERE kue.Kd_kue = @Kode;
END;
```

```
-- Membuat trigger AfterDelete_DetailPembelian --
CREATE TRIGGER AfterDelete_DetailPembelian
ON detail_pembelian
AFTER DELETE
AS
BEGIN
    DECLARE
        @Jumlah INT, @Kode CHAR(5)
    SELECT @Jumlah = Jml_beli,
           @Kode = Kd_kue FROM deleted

    UPDATE kue SET kue.Stok_kue =
        kue.Stok_kue + @Jumlah
    WHERE kue.Kd_kue = @Kode;
END;
```

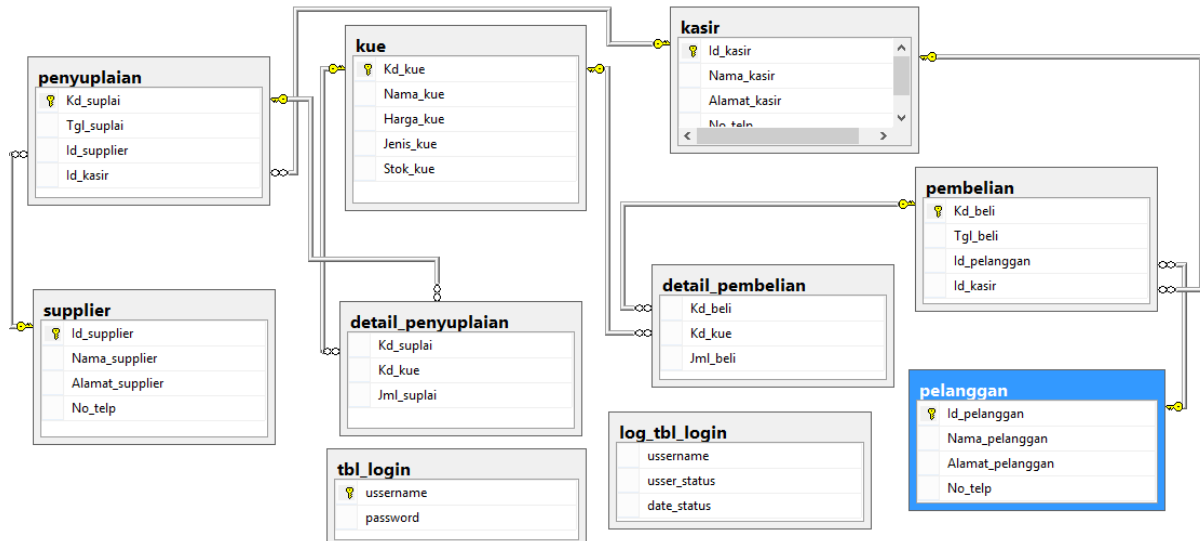
### 3.5 Membuat Login dan Log Login

```
-- Membual tabel login --
CREATE TABLE tbl_login (username VARCHAR(25) PRIMARY KEY,
password VARCHAR(25) NOT NULL);

-- Membuat tabel log login --
CREATE TABLE log_tbl_login (username VARCHAR(25) NOT NULL,
usser_status VARCHAR(10) NOT NULL,
date_status DATETIME DEFAULT GETDATE());

-- Membuat trigger insert login --
CREATE TRIGGER insert_login
ON tbl_login
AFTER INSERT
AS
BEGIN
    DECLARE @usser VARCHAR(25)
    SELECT @usser = username FROM inserted
    INSERT INTO log_tbl_login (username, usser_status)
    VALUES (@usser, 'REGISTERED');
END;

-- Simpan usser ke tbl_login --
INSERT INTO tbl_login VALUES ('panjik', '123');
INSERT INTO tbl_login VALUES ('ajik', '123');
-- Cek data di tabel log_tbl_login --
SELECT * FROM log_tbl_login;
```

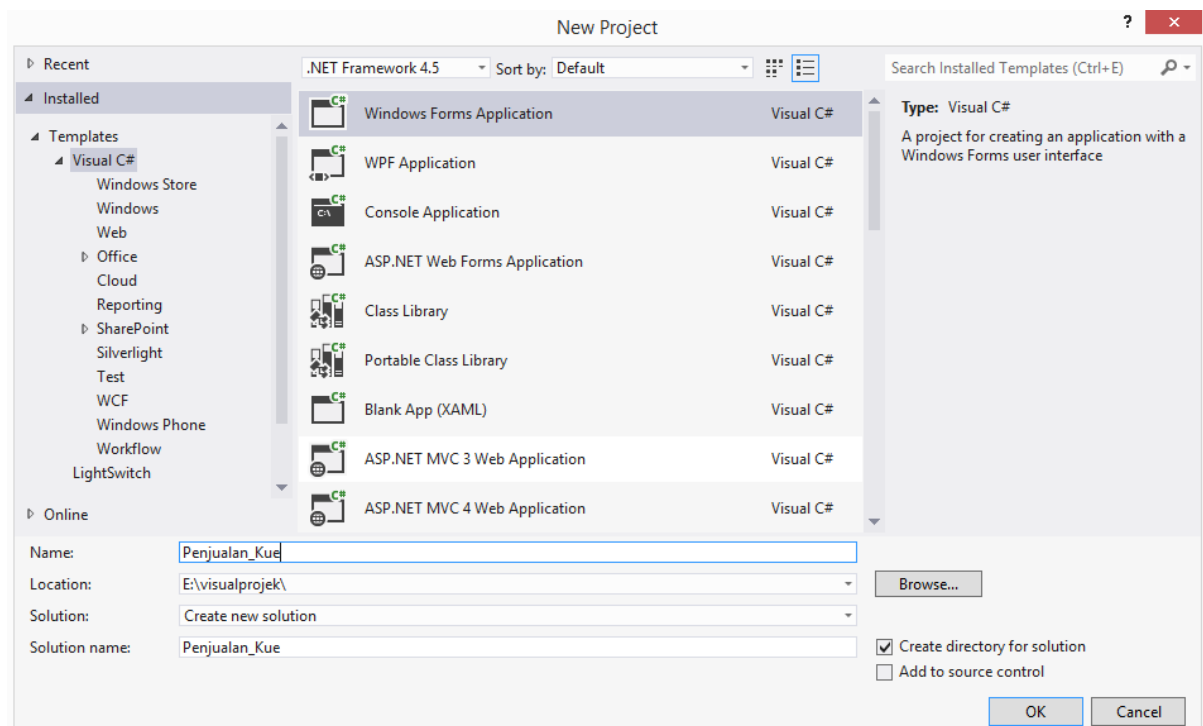




## Bab 4 Membuat Project Aplikasi Penjualan Kue

### 4.1 Membuat Project Baru

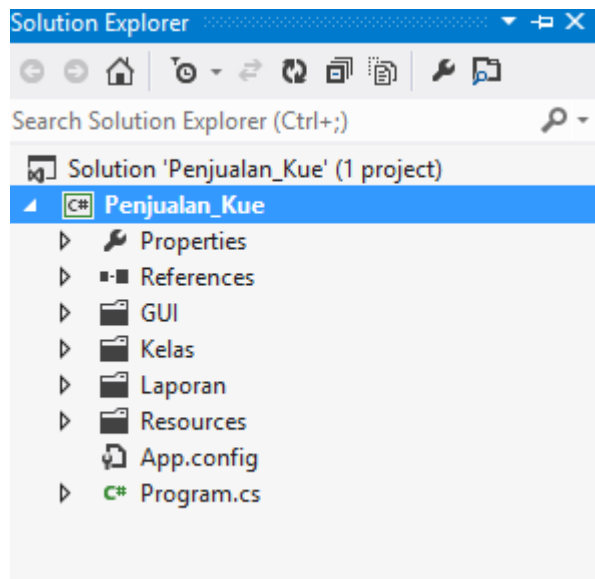
1. Buka Aplikasi Microsoft Visual Studio
2. Pilih File -> New -> Project
3. Pilih Template Project Visual C#, Lalu Pilih Windows Form Application
4. Berikan Nama Project = “(Tersebut Anda)”
5. Kemudian Browser Location penyimpanan project



### 4.2 Membuat Package

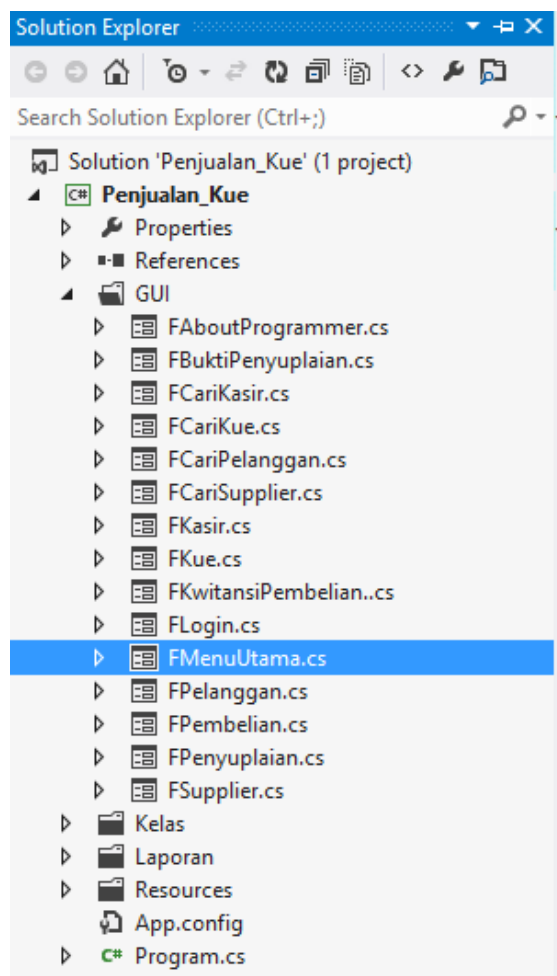
Package adalah suatu cara untuk memudahkan kita mengatur objek-objek atau Form yang di buat. Untuk membuat package pada project sebagai berikut :

1. Klik kanan pada project yang telah dibuat, lalu pilih “add” -> “New Folder”
2. Kemudian berikan nama package tersebut buat 3 package (GUI, Kelas, Laporan ) seperti berikut :



3. Package GUI akan di isi dengan desain Form project, package Kelas akan diisi dengan Objek kelas koneksi, dan package Laporan akan diisi dengan Objek desain Laporan .

4. Buatlah Windows Form di dalam Package GUI, dengan nama sebagai berikut :



#### 4.3 Membuat dan Codingan Menu Utama

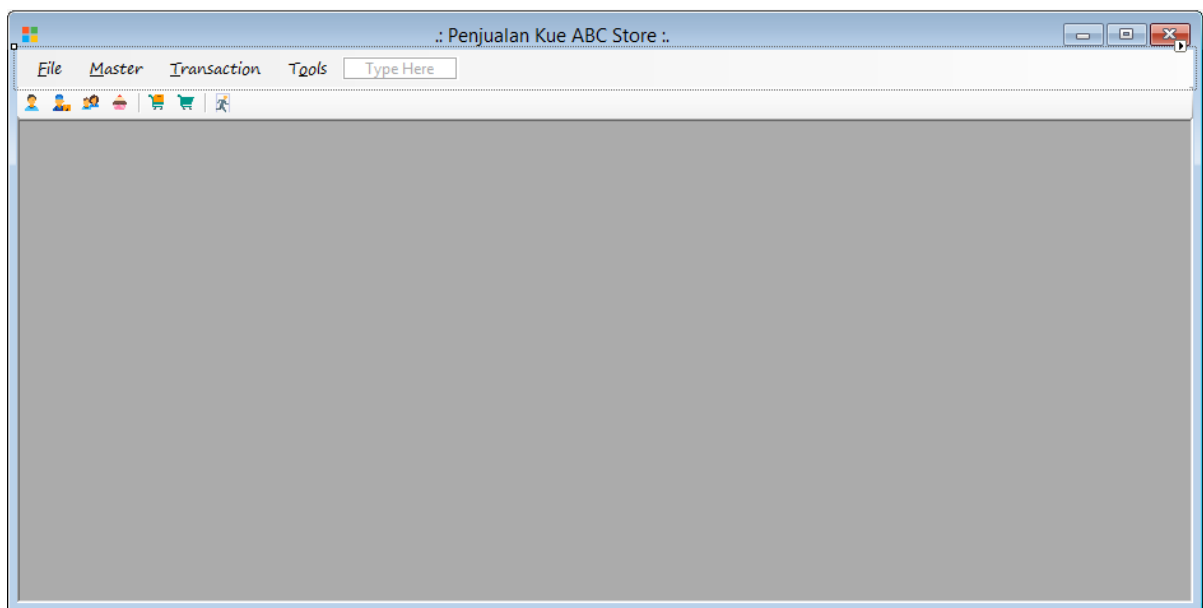
1. Buka Form MenuUtama.cs

2. Klik Form Menu Utama nya

3. Lalu Atur Properties Form “ FmenuUtama.cs” menjadi sebagai berikut :

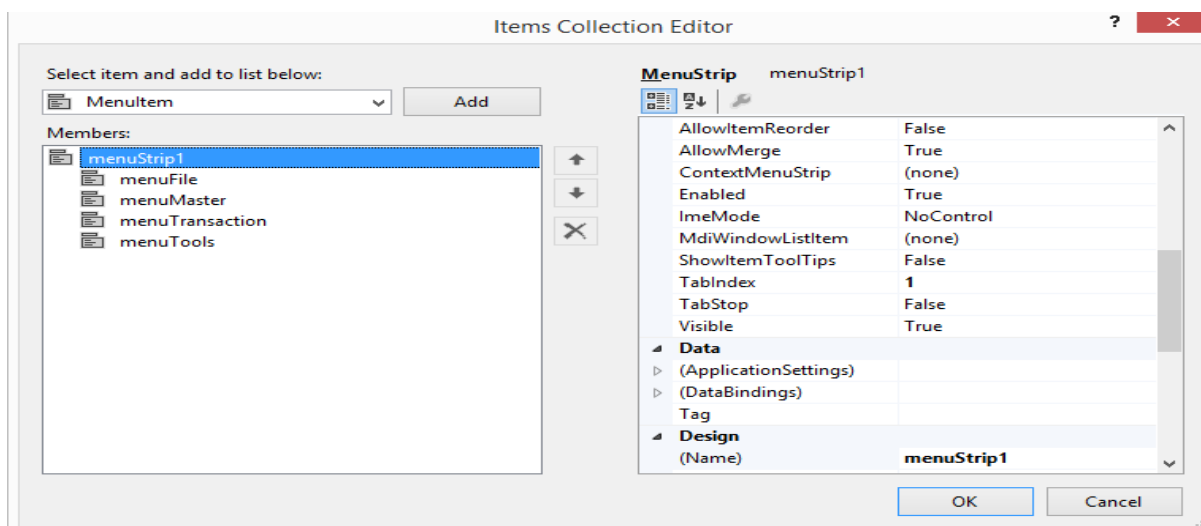
Properties	Values
Text	.: Penjualan Kue ABC Store :.
Windows State	Maximize
Start Position	CenterScreen
IsMdiContainer	True
Bacround Image	Pilih gambar untuk menjuadi Background utama
BacgroundImageLayout	Strerch

4. Tambahkan tool ke menu utama sebagai berikut :



5. Klik kanan pad menuStrip1, pilih Edit Items, lalu atur menu menjadi sebagai berikut :

\*Klik tombol add untuk menambahkan menu



6. Atur tiap properties sebagai berikut

Menu Item	Values
File	Text: &File Name: menuFile Modifier :Public
Master	Text : &Master Name : menuMaster Modifier :Public
Transaction	Text : &Master Name : &Master Modifier : Pubhlic
Tools	Text : T&ools Name : menuTools Modifier : Public

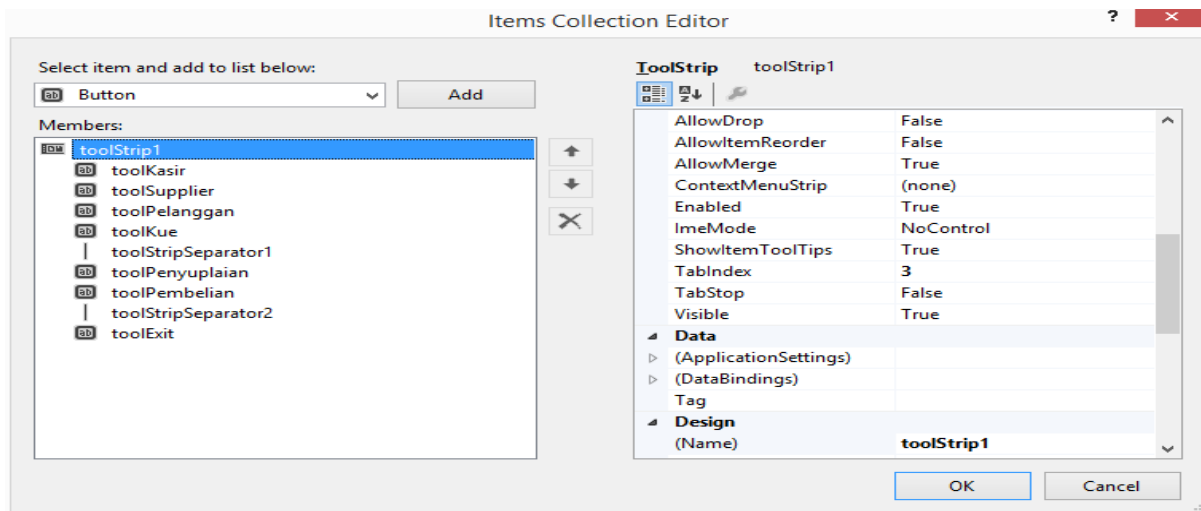
7. Selanjutnya buat submenu dari menu yang telah kita buat, contoh pada menu “File” tambahkan submenu “Login”, “Logout” dan submenu “Exit”



8. Berikut table untuk mengatur submenu

Menu Item	Submenu	Properties
File	Login	<ul style="list-style-type: none"> <li>Name : subMenuLogin</li> <li>ShortcutKey: Ctrl+L</li> <li>Modifier:Public</li> <li>Image:Penjualan_Kue.Properties.Resources.icons8_Lock_96px</li> </ul>
	Logout	<ul style="list-style-type: none"> <li>Name :submenuLogout</li> <li>ShortcutKey: Ctrl+X</li> <li>Modifier :Public</li> <li>Image:Penjualan_Kue.Properties.Resources.icons8_Padlock_96px</li> </ul>

	Exit	<ul style="list-style-type: none"> <li>• Name :subMenuExit</li> <li>• ShortcutKey: Atl+F4</li> <li>• Modifier: Public</li> <li>• Image:Penjualan_Kue.Properties.Resources.icons8_Exit_Sign_96px</li> </ul>
Master	Kasir	<ul style="list-style-type: none"> <li>• Name : subMenuKasir</li> <li>• ShortcutKey: Ctrl+K</li> <li>• Modifier: Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_User_96px</li> </ul>
	Supplier	<ul style="list-style-type: none"> <li>• Name : subMenuSupplier</li> <li>• ShortcutKey: Ctrl+S</li> <li>• Modifier:Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_Supplier_96px</li> </ul>
	Pelanggan	<ul style="list-style-type: none"> <li>• Name :subMenuPelanggan</li> <li>• ShortcutKey: Ctrl+P</li> <li>• Modifier: Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_Team_96px</li> </ul>
	Kue	<ul style="list-style-type: none"> <li>• Name :subMenuKue</li> <li>• ShortcutKey: Ctrl+I</li> <li>• Modifier: Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_Cupcake_96px</li> </ul>
Transaction	Penyuplaian	<ul style="list-style-type: none"> <li>• Name : subMenuPenyuplaian</li> <li>• ShortcutKey:Ctrl+B</li> <li>• Modifier:Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_Sell_Stock_96px</li> </ul>
	Pembelian	<ul style="list-style-type: none"> <li>• Name :subMenuPembelian</li> <li>• ShortcutKey: Ctrl+C</li> <li>• Modifier: Public</li> <li>• Image:Penjualan_Kue.Properties.Resources.icons8_Shopping_Cart_96px</li> </ul>
About	About	<ul style="list-style-type: none"> <li>• Name : subMenuAbout</li> <li>• ShortcutKey: Ctrl+ A</li> <li>• Modifier: Public</li> <li>• Image: Penjualan_Kue.Properties.Resources.icons8_About_96px</li> </ul>



9. Kemudian atur toolbar, dengan cara klik kanan pada “toolsStrip1”, lalu pilih “Edit Item”.

Item	Values
toolKair	ToolTipText : Tampilkan data kasir Name : toolKasir Modifier : Public Image:Penjualan_Kue.Properties.Resources.icons8_User_96px
toolSupplier	ToolTipText : Tampilkan data supplier Name : toolSupplier Modifier :Public Image:Penjualan_Kue.Properties.Resources.icons8_Supplier_96px
toolPelanggan	ToolTipText : Tampilkan data pelanggan Name : toolPelanggan Modifier : Public Image : Penjualan_Kue.Properties.Resources.icons8_Team_96px
toolKue	ToolTipText : Tampilkan data kue Name : toolKue Modifier : Public Image:Penjualan_Kue.Properties.Resources.icons8_Cupcake_96px
toolPenyuplain	ToolTipText : Tampilkan data transaksi penyuplaian Name :toolPenyuplain Modifier :Public Image:Penjualan_Kue.Properties.Resources.icons8_Sell_Stock_96px
toolPembelian	ToolTipText :Tampilkan data Transaksi Pembelian Name :toolPembelian Modifier : Public Image:Penjualan_Kue.Properties.Resources.icons8_Shopping_Cart_96px
toolExit	ToolTipText : Keluar Aplikasi Name :toolExit Modifier :Public Image : Penjualan_Kue.Properties.Resources.icons8_Exit_Sign_96px

10. Agar perogram dapat berjalan , kilk 2 pada “program.cs” kemudian atur kode program sebgai berikut :

```

static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new GUI.FMenuUtama());
}

```

11. Running project

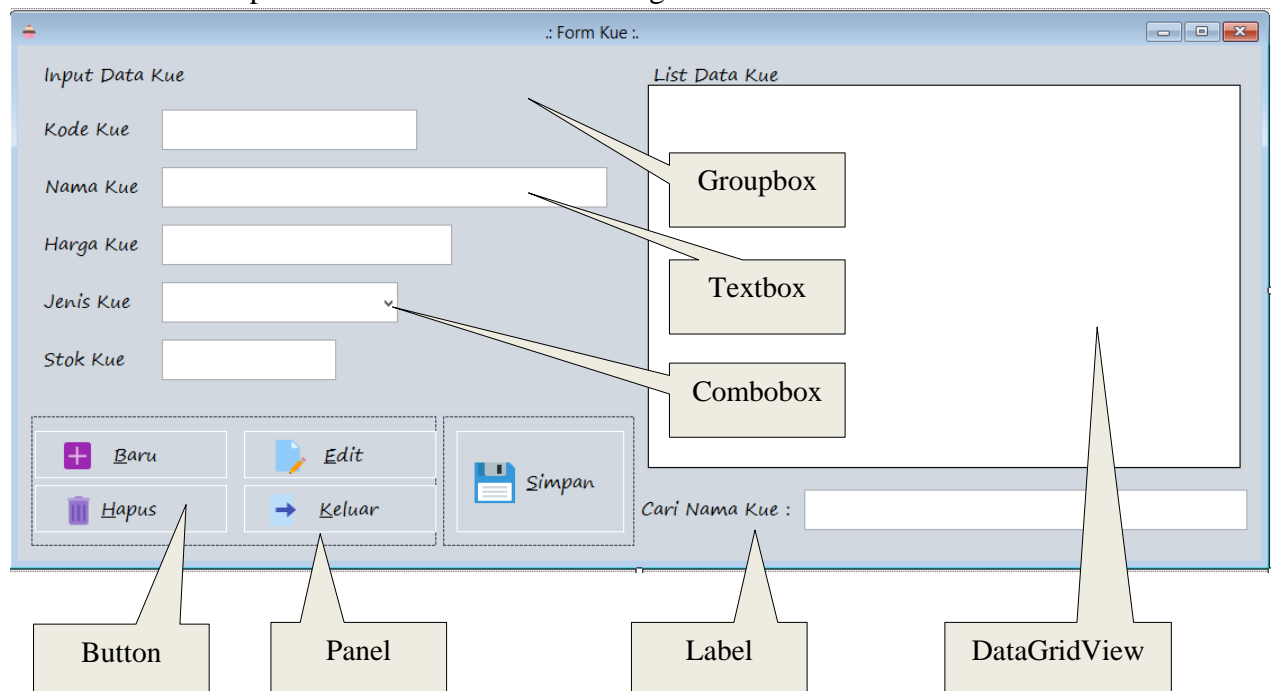
## Bab 5 Desain Form Master Aplikasi Penjualan Kue

### 5.1 Membuat Form Kue

1. Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
2. Berikan nama Fkue.cs
3. Atur Properties Fkue.cs Sebagai berikut :

Properties	Values
Text	.: Form Kue .:
Start Position	CenterScreen
Font	Segoe UI; 12pt

4. Tambahkan komponen ke dalam Form kue sebagai berikut :



5. Atur properties tiap komponen pada Form Kue sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1
	Text : Input Data Kue
	Name : groupBox2
	Text : List Data Kue

Label	Text = Kode Kue
	Text = Nama Kue
	Text = Harga Kue
	Text = Jenis Kue
	Text = Stok Kue
	Text = Cari Nama Kue :
Textbox	Name : txtKodeKue
	Name : txtNamaKue
	Name : txtHargaKue
	Name : txtJenisKue
	Name : txtStokKue
	Name : txtCariKue
ComboBox	Nama : cmbJenisKue Item : Kue Bolu,Kue Brownis,Kue Ulang Tahun
Panel	Nama : Panel1
	Nama : Panel2
Button	Nama : btnBaru Text :&Baru
	Nama : btnEdit Text :&Edit
	Nama : btnHapus Text :&Hapus
	Nama : btnKeluar Text :&Keluar
	Nama : btnSimpan Text :&Simpan
	** Untuk menambahkan icon, pilih properties image.
DatagridView	Nama : ListKue

## 5.2 Membuat Form Kasir

- 1.Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
- 2.Berikan nama Fkasir.cs
- 3.Atur Properties FKasir.cs Sebagai berikut :

Properties	Values
Text	.: Form Kasir .:
Star Position	CenterScreen
Font	Segoe UI; 12pt



4. Tambahkan komponen ke dalam Form kasir sebagai berikut :

5. Atur properties tiap komponen pada Form Kasir sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1 Text : Input Data Kasir
	Name : groupBox2 Text : List Data Kasir
Label	Text = Id Kasir
	Text = Nama Kasir
	Text = Alamat Kasir
	Text = Nomor Telepon
	Text = Cari Nama Kasir :
Textbox	Name : txtIdKasir
	Name : txtNamaKasir
	Name : txtAlamatKasir
	Name : txtJNomorTelepon
	Name : txtCariKasir
Panel	Nama : Panel2
	Nama : Panel3
Button	Nama : btnBaru Text :&Baru
	Nama : btnEdit Text :&Edit
	Nama : btnHapus Text :&Hapus
	Nama : btnKeluar Text :&Keluar
	Nama : btnSimpan Text :&Simpan
	** Untuk menambahkan icon, pilih properties image.

	Nama : ListKue
DatagridView	Nama :listKasir

### 5.3 Membuat Form Kasir

- 1.Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
- 2.Berikan nama FSupplier.cs
- 3.Atur Properties FSupplier.cs Sebagai berikut :

Properties	Values
Text	.: Form Supplier :.
Star Position	CenterScreen
Font	Segoe UI; 12pt

- 4.Tambahkan komponen ke dalam Form kasir sebagai berikut :

5. Atur properties tiap komponen pada Form Supplier sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1
	Text : Input Data Supplier
Label	Name : groupBox2
	Text : List Data Supplier
	Text = Id Supplier
	Text = Nama Supplier
	Text = Alamat Supplier
Textbox	Text = Nomor Telepon
	Text = Cari Nama Supplier :
	Name : txtIdSupplier
	Name : txtNamaSupplier
	Name : txtAlamatSupplier
Panel	Name : txtJNomorTelepon
	Name : txtCariSupplier
Button	Nama : Panel2
	Nama : Panel3
	Nama : btnBaru

	Text :&Baru
	Nama :btnEdit Text :&Edit
	Nama :btnHapus Text :&Hapus
	Nama :btnKeluar Text :&Keluar
	Nama :btnSimpan Text :&Simpan ** Untuk menambahkan icon, pilih properties image.
DatagridView	Nama :listSupplier

## 5.4 Membuat Form Pelanggan

- 1.Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
- 2.Berikan nama FPelanggan.cs
- 3.Atur Properties FPelanggan.cs Sebagai berikut :

Properties	Values
Text	.: Form Pelanggan .:
Star Position	CenterScreen
Font	Segoe UI; 12pt

- 4.Tambahkan komponen ke dalam Form kasir sebagai berikut :

5. Atur properties tiap komponen pada Form Pelanggan sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1 Text : Input Data Pelanggan
	Name : groupBox2 Text : List Data Pelanggan
Label	Text = Id Pelanggan
	Text = Nama Pelanggan
	Text = Alamat Pelanggan
	Text = Nomor Telepon
	Text = Cari Nama Pelanggan :

Textbox	Name : txtIdPelanggan
	Name : txtNamaPelanggan
	Name : txtAlamatPelanggan
	Name : txtJNomorTelepon
	Name : txtCariPelanggan
Panel	Nama : Panel2
	Nama : Panel3
Button	Nama : btnBaru Text :&Baru
	Nama : btnEdit Text :&Edit
	Nama : btnHapus Text :&Hapus
	Nama : btnKeluar Text :&Keluar
	Nama : btnSimpan Text :&Simpan ** Untuk menambahkan icon, pilih properties image.
DatagridView	Nama : listSupplier

## 5.5 Membuat Form Penyuplaian

1. Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
2. Berikan nama FPenyuplaian.cs
3. Atur Properties FPenyuplaian.cs Sebagai berikut :

Properties	Values
Text	.: Form Penyuplaian .:
Start Position	CenterScreen
Font	Segoe UI; 12pt

4. Tambahkan komponen ke dalam Form penyuplai sebagai berikut :

The screenshot shows a Windows Form titled "Data Penyuplaian". It contains several input fields and buttons. Callouts identify the following components:

- DateTimePicker**: Points to the "Tanggal" field showing "08/26/2018".
- Button**: Points to a button with a magnifying glass icon next to the "Id Supplier" field.
- Panel**: Points to the bottom section of the form containing three buttons: "Baru" (with a document icon), "Simpan" (with a floppy disk icon), and "Keluar" (with a right arrow icon).

Other visible fields include "Kode Suplai", "Nama Supplier", "Id Kasir", "Nama Kasir", "Kode Kue", "Nama Kue", "Jenis Kue", and "Jumlah Suplai". There is also a green "+" button next to the "Jumlah Suplai" field.

5. Atur properties tiap komponen pada Form Kue sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1 Text : Data Penyuplaian
	Name : groupBox2 Text : Data Supplier
	Name : groupBox3 Text : Data Kasir
	Name : groupBox4 Text : Data Kue
	Name : groupBox5 Text : Data Penyuplaian
Label	Text = Kode Suplai
	Text = Tanggal
	Text = Id Supplier
	Text = Nama Supplier
	Text = Id Kasir
	Text = Nama Kasir
	Text = Kode Kue
	Text = Nama Kue
	Text = Jenis kue
Textbox	Text = Jumlah Suplai
	Name : txtKodeSuplai
	Name : txtIdSupplier
	Name : txtNamaSupplier
	Name : txtIdKair
	Name : txtNamaKasir
	Name : txtKodeKue
	Name : txtNamaKue
Panel	Name : txtJenisKue
	Nama : Panel1
Button	Nama : btnBaru Text :&Baru
	Nama : btnEdit Text :&Edit
	Nama : btnKeluar Text :&Keluar
	Nama : btnSimpan Text :&Simpan
	** Untuk menanmbahkan icon, pilih properties image.
DatagridView	Nama : dataGridPenyuplaian

## 5.6 Membuat Form Pembelian

1. Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”
2. Berikan nama FPenyuplaian.cs
3. Atur Properties FPenyuplaian.cs Sebagai berikut :

Properties	Values
Text	.: Form Penyuplaian .:
Star Position	CenterScreen
Font	Segoe UI; 12pt

4. Tambahkan komponen ke dalam Form pembelian sebagai berikut :

The screenshot shows a Windows Form titled ': Form Pembelian :'. The form is divided into two main sections. The left section contains input fields for purchase data, organized into sections: 'Data Pembelian' (Kode Beli, Tanggal), 'Data Pelanggan' (Id Pelanggan, Nama Pelanggan), 'Data Kasir' (Id Kasir, Nama Kasir), 'Data Kue' (Kode Kue, Nama Kue, Harga Kue), and 'Jumlah Beli'. The right section is titled 'Data Pembelian' and is currently empty. At the bottom, there is a 'Total Bayar' section with a label 'Rp. lblTotalBayar' and a callout box labeled 'Label' pointing to it. Below the total section are three buttons: 'Baru' (with a plus icon), 'Simpan' (with a floppy disk icon), and 'Keluar' (with an arrow icon).

5. Atur properties tiap komponen pada Form Kue sesuai dengan tabel berikut :

Komponen	Properties
Groupbox	Name : groupBox1
	Text : Data Pembelian
	Name : groupBox2
	Text : Data Kasir
Label	Name : groupBox3
	Text : Data Kue
	Name : groupBox4
	Text : Data Pembelian
	Text = Kode Beli
	Text = Tanggal
	Text = Id Pelanggan
	Text = Nama Pelanggan
Label	Text = Id Kasir
	Text = Nama Kasir
	Text = Kode Kue
	Text = Nama Kue
	Text = Jenis kue

	Text = Jumlah Beli
Textbox	Name : txtKodeBeli
	Name : txtIdPelanggan
	Name : txtNamaPelanggan
	Name : txtIdKair
	Name : txtNamaKasir
	Name : txtKodeKue
	Name : txtNamaKue
	Name : txtJenisKue
Panel	Nama : Panel1
Button	Nama : btnBaru Text :&Baru
	Nama : btnSimpan Text :&Simpan
	Nama : btnKeluar Text :&Keluar
	** Untuk menambahkan icon, pilih properties image.
DatagridView	Nama : dataGridPembelian

### 5.7 Membuat Form Cari Kue

1. Klik kanan pada package "GUI", lalu pilih "add" – "Windows"
2. Berikan nama FCariKue.cs
3. Atur Properties FCariKue.cs Sebagai berikut :

Properties	Values
Text	.: Form Cari Kue .:
Star Position	CenterScreen
Font	Segoe UI; 12pt
FormBorderStyle	None

4. Tambahkan komponen ke dalam Form pembelian sebagai berikut :

*Data Kue*

*Cari Berdasarkan Kode / Nama Kue :*

5. Atur properties tiap komponen pada Form Cari Kue sesuai dengan tabel berikut :

Komponen	Properties
GroupBox	Name :groupbox1 Text = .: Data Kue .:
Label	Text = Cari Berdasarkan Kode>Nama Kue :
Textbox	Name : TxtCariPelanggan
DatagridView	Name : dataGridKue

## 5.8 Membuat Form Cari Kasir

1.Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”

2.Berikan nama FCariKasir.cs

3.Atur Properties FCariKasir.cs Sebagai berikut :

Properties	Values
Text	.: Form Cari Kasir .:
Star Position	CenterScreen
Font	Segoe UI; 12pt
FormBorderStyle	None

4.Tambahkan komponen ke dalam Form Cari Kasir sebagai berikut :

5. Atur properties tiap komponen pada Form Cari Kue sesuai dengan tabel berikut :

Komponen	Properties
GroupBox	Name :groupbox1 Text = .: Data Kasir .:
Label	Text = Cari Berdasarkan Kode>Nama Kasir :
Textbox	Name : TxtCariKasir
DatagridView	Name : dataGridKasir



## 5.9 Membuat Form Cari Supplier

1. Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”

2. Berikan nama FcariSupplier.cs

3. Atur Properties FcariSupplier.cs Sebagai berikut :

Properties	Values
Text	.: Form Cari Supplier :.
Start Position	CenterScreen
Font	Segoe UI; 12pt
FormBorderStyle	None

4. Tambahkan komponen ke dalam Form Cari Supplier sebagai berikut :

5. Atur properties tiap komponen pada Form Supplier sesuai dengan tabel berikut :

Komponen	Properties
GroupBox	Name : groupbox1 Text = .: Data Supplier :.
Label	Text = Cari Berdasarkan Kode/Nama Supplier :
Textbox	Name : TxtCariSupplier
DatagridView	Name : dataGridSupplier

## 5.10 Membuat Form Cari Pelanggan

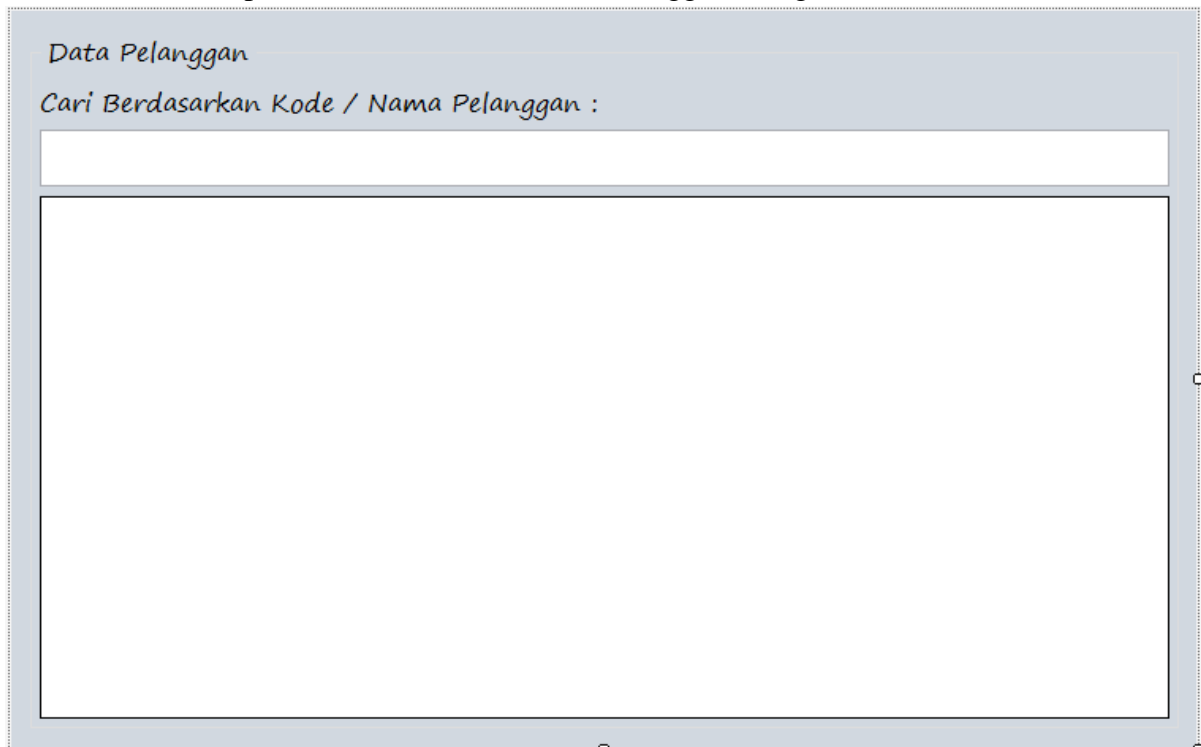
1. Klik kanan pada package “GUI”, lalu pilih “add” – “Windows”

2. Berikan nama FcariPelanggan.cs

3. Atur Properties FcariPelanggan.cs Sebagai berikut :

Properties	Values
Text	.: Form Cari Pelanggan .:
Start Position	CenterScreen
Font	Segoe UI; 12pt
FormBorderStyle	None

4. Tambahkan komponen ke dalam Form Cari Pelanggan sebagai berikut :



5. Atur properties tiap komponen pada Form Pelanggan sesuai dengan tabel berikut :

Komponen	Properties
GroupBox	Name : groupbox1 Text = .: Data Pelanggan .:
Label	Text = Cari Berdasarkan Kode/Nama Pelanggan :
Textbox	Name : TxtCariPelanggan
DatagridView	Name : dataGridPelanggan

# 6

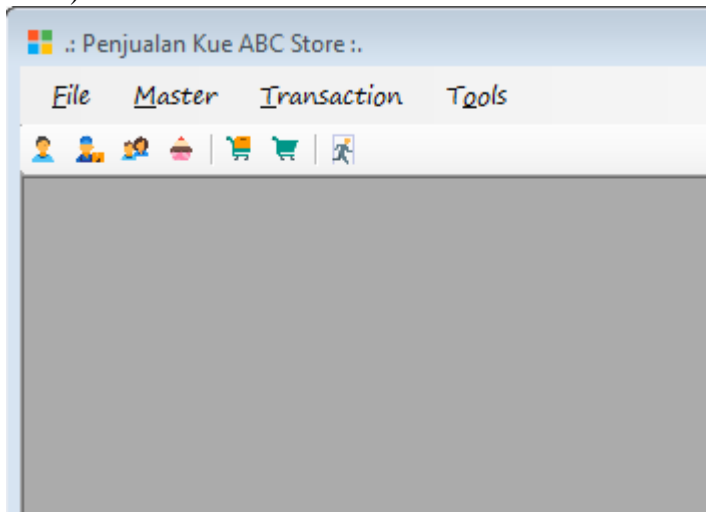
## Coding Form Menu Utama Aplikasi Penjualan Kue

Materi Pembahasan :

1. Memanggil Form Master
2. Memanggil Form Transaction

### 6.1 Menampilkan Kode Program (View Code)

1. Masuk ke Form Menu Utama, atau klik 2 kali FMenuUtama.cs yang berada di Solution Explorer
2. Klik Kanan Form Menu Utama pilih “view code” atau F7 (untuk versi Visual Studio 2012)



3. Ketikkan kode berikut, di bawah public partial class FMenuUtama : Form {  
Kode ini berfungsi untuk mendeklarasikan variabel buatan yang bernilai Form terkait yang ada di project, yang nantinya akan digunakan untuk pemanggilan form.

```
public partial class FMenuUtama : Form
{
    // Membuat variabel / objek dari form terkait
    public static FMenuUtama mdiObj;
    FPelanggan plg;
    FKasir ksr;
    FSupplier supp;
    FKue kue;
    FPenyuplaian suplai;
    FPembelian beli;
    FAboutProgrammer aboutProg;
```

4. Pada baris berikutnya, buat prosedur dengan event FormClosed yang berfungsi untuk membuat nilai variabel form (yang dibuat pada langkah ke-3) menjadi null pada saat form ditutup

```
// Membuat prosedur event form closed
void Proc_KasirClosed(Object sender, FormClosedEventArgs e)
{
    ksr = null;
}

void Proc_PelangganClosed(Object sender, FormClosedEventArgs e)
{
    plg = null;
}

void Proc_SupplierClosed(Object sender, FormClosedEventArgs e)
{
    supp = null;
}

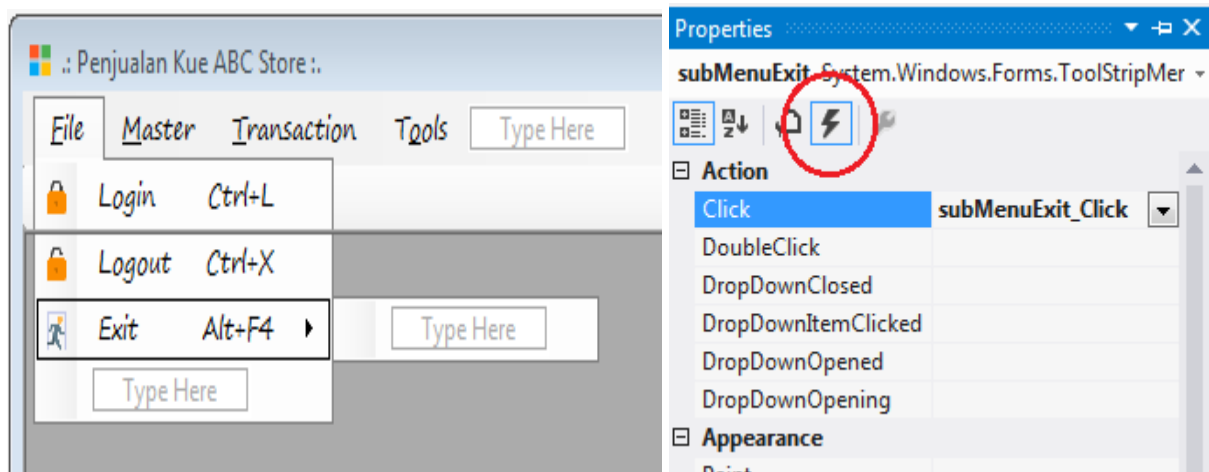
void Proc_KueClosed(Object sender, FormClosedEventArgs e)
{
    kue = null;
}

void Proc_PenyuplaianClosed(Object sender, FormClosedEventArgs e)
{
    suplai = null;
}

void Proc_PembelianClosed(Object sender, FormClosedEventArgs e)
{
    beli = null;
}

void Proc_AboutProgrammerClosed(Object sender, FormClosedEventArgs e)
{
    aboutProg = null;
}
```

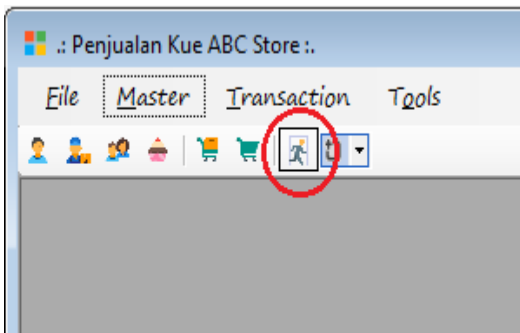
5. Kembali ke Desain Form Menu Utama, lalu pilih menu File – Exit, kemudian di halaman properties, aktifkan tombol “Event”, lalu klik 2 kali pada pilihan Event “Click”.



6. Ketikkan kode berikut pada Event Click menu Exit

```
private void subMenuExit_Click(object sender, EventArgs e)
{
    // Membuat kode keluar aplikasi
    if (MessageBox.Show("Keluar dari aplikasi ?", "Konfirmasi",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        Application.Exit();
    }
}
```

7. Kembali ke Desain Form Menu Utama, klik 2 kali toolbar Exit “ToolExit”, lalu ketik perintah berikut :



```
private void toolExit_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu exit
    subMenuExit_Click(sender, e);
}
```

## 6.2 Memanggil Form Master

1. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Master – Kasir

```
private void subMenuKasir_Click(object sender, EventArgs e)
{
    // Memanggil form kasir
    if (ksr == null)
    {
        ksr = new FKasir();
        ksr.MdiParent = this;
        ksr.FormClosed += new FormClosedEventHandler(Proc_KasirClosed);
        ksr.Show();
    }
    else
    {
        ksr.Activate();
    }
}
```

2. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolKasir

```
private void toolKasir_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu kasir
    subMenuKasir_Click(sender, e);
}
```

3. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Master – Supplier

```
private void subMenuSupplier_Click(object sender, EventArgs e)
{
    // Memanggil form supplier
    if (supp == null)
    {
        supp = new FSupplier();
        supp.MdiParent = this;
        supp.FormClosed += new FormClosedEventHandler(Proc_SupplierClosed);
        supp.Show();
    }
    else
    {
        supp.Activate();
    }
}
```

4. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolSupplier

```
private void toolSupplier_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu supplier
    subMenuSupplier_Click(sender, e);
}
```

5. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Master – Pelanggan

```
private void subMenuPelanggan_Click(object sender, EventArgs e)
{
    // Memanggil form pelanggan
    if (plg == null)
    {
        plg = new FPelanggan();
        plg.MdiParent = this;
        plg.FormClosed += new FormClosedEventHandler(Proc_PelangganClosed);
        plg.Show();
    }
    else
    {
        plg.Activate();
    }
}
}
```

6. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolSupplier

```
private void toolPelanggan_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu pelanggan
    subMenuPelanggan_Click(sender, e);
}
}
```

7. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Master – Kue

```
private void subMenuKue_Click(object sender, EventArgs e)
{
    // Memanggil form kue
    if (kue == null)
    {
        kue = new FKue();
        kue.MdiParent = this;
        kue.FormClosed += new FormClosedEventHandler(Proc_KueClosed);
        kue.Show();
    }
    else
    {
        kue.Activate();
    }
}
}
```

8. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolKue

```
private void toolKue_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu kue
    subMenuKue_Click(sender, e);
}
```

9. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Tools – About

```
private void subMenuAbout_Click(object sender, EventArgs e)
{
    // Memanggil form about programmer
    if (aboutProg == null)
    {
        aboutProg = new FAboutProgrammer();
        aboutProg.MdiParent = this;
        aboutProg.FormClosed += new FormClosedEventHandler(Proc_AboutProgrammerClosed);
        aboutProg.Show();
    }
    else
    {
        aboutProg.Activate();
    }
}
```

### 6.3 Memanggil Form Transaksi

1. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Transaction – Penyuplaian

```
private void subMenuPenyuplaian_Click(object sender, EventArgs e)
{
    // Memanggil form penyuplaian
    if (suplai == null)
    {
        suplai = new FPenyuplaian();
        suplai.MdiParent = this;
        suplai.FormClosed += new FormClosedEventHandler(Proc_PenyuplaianClosed);
        suplai.Show();
    }
    else
    {
        suplai.Activate();
    }
}
```

2. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolPenyuplaian



```
private void toolPenyuplaian_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu penyuplaian
    subMenuPenyuplaian_Click(sender, e);
}
```

3. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu Transaction – Pembelian

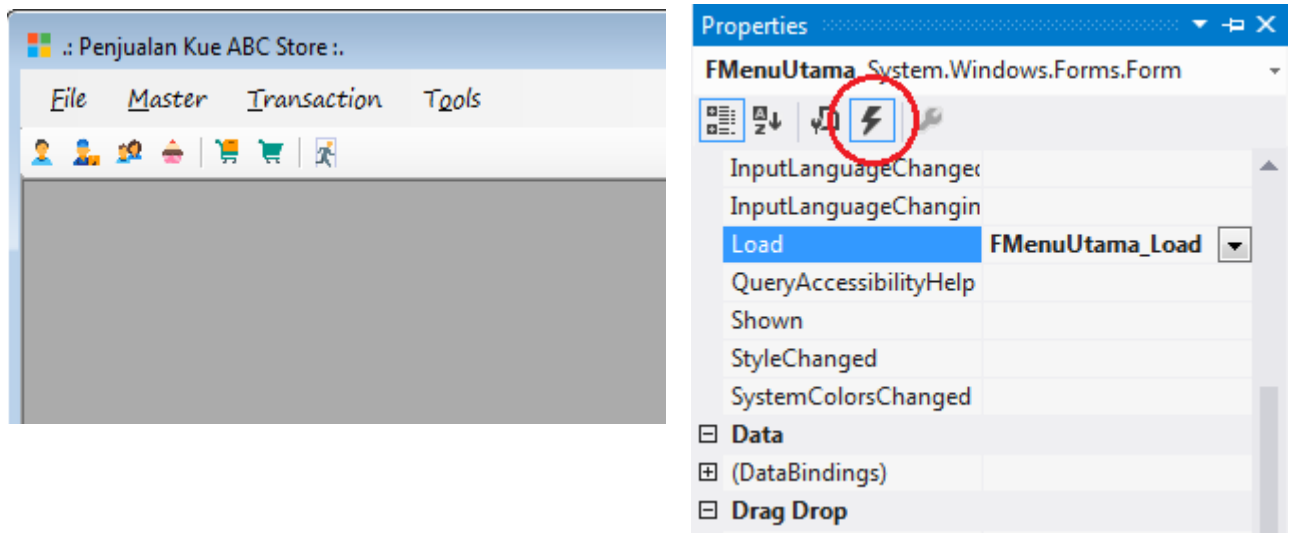
```
private void subMenuPembelian_Click(object sender, EventArgs e)
{
    // Memanggil form pembelian
    if (beli == null)
    {
        beli = new FPembelian();
        beli.MdiParent = this;
        beli.FormClosed += new FormClosedEventHandler(Proc_PembelianClosed);
        beli.Show();
    }
    else
    {
        beli.Activate();
    }
}
```

4. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Toolbar toolPembelian

```
private void toolPembelian_Click(object sender, EventArgs e)
{
    // Memanggil kode dari event click pada menu pembelian
    subMenuPembelian_Click(sender, e);
}
```

## 6.4 Memanggil Form Login

1. Kembali ke Desain Form Menu Utama, pastikan komponen yang aktif adalah form FMenuUtama (bukan komponen lain) kemudian di halaman properties, aktifkan tombol “Event”, lalu klik 2 kali pada pilihan Event “Load”.



2. Ketikkan kode berikut pada Event Load FMenuUtama

```
private void FMenuUtama_Load(object sender, EventArgs e)
{
    // Non aktifkan beberapa menu pada saat menu utama aktif
    menuMaster.Enabled = false;
    menuTransaction.Enabled = false;
    subMenuLogin.Enabled = true;
    subMenuLogout.Enabled = false;
    toolPelanggan.Enabled = false;
    toolKasir.Enabled = false;
    toolSupplier.Enabled = false;
    toolKue.Enabled = false;
    toolPenyuplaian.Enabled = false;
    toolPembelian.Enabled = false;

    // Memberikan nilai untuk var mdiObj adalah FMenuUtama
    mdiObj = this;

    // Menampilkan form login
    FLogin login = new FLogin();
    login.ShowDialog();
}
```

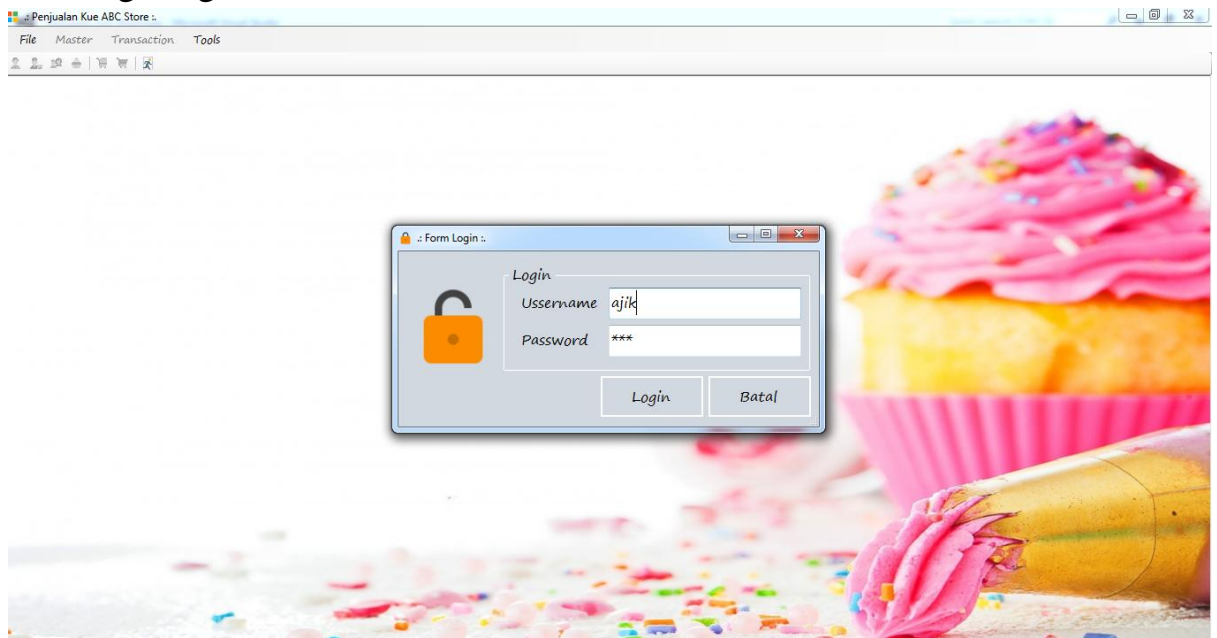
3. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu File – Logout

```
private void subMenuLogout_Click(object sender, EventArgs e)
{
    // Non aktifkan beberapa menu pada saat logout
    menuMaster.Enabled = false;
    menuTransaction.Enabled = false;
    subMenuLogin.Enabled = true;
    subMenuLogout.Enabled = false;
    toolPelanggan.Enabled = false;
    toolKasir.Enabled = false;
    toolSupplier.Enabled = false;
    toolKue.Enabled = false;
    toolPenyuplaian.Enabled = false;
    toolPembelian.Enabled = false;
}
```

4. Kembali ke Desain Form Menu Utama, isikan kode berikut pada Event Click Menu File – Login

```
private void subMenuLogin_Click(object sender, EventArgs e)
{
    // Memanggil form login dari menu login
    FLogin login = new FLogin();
    login.ShowDialog();
}
```

5. Running Program [Start]



# 7

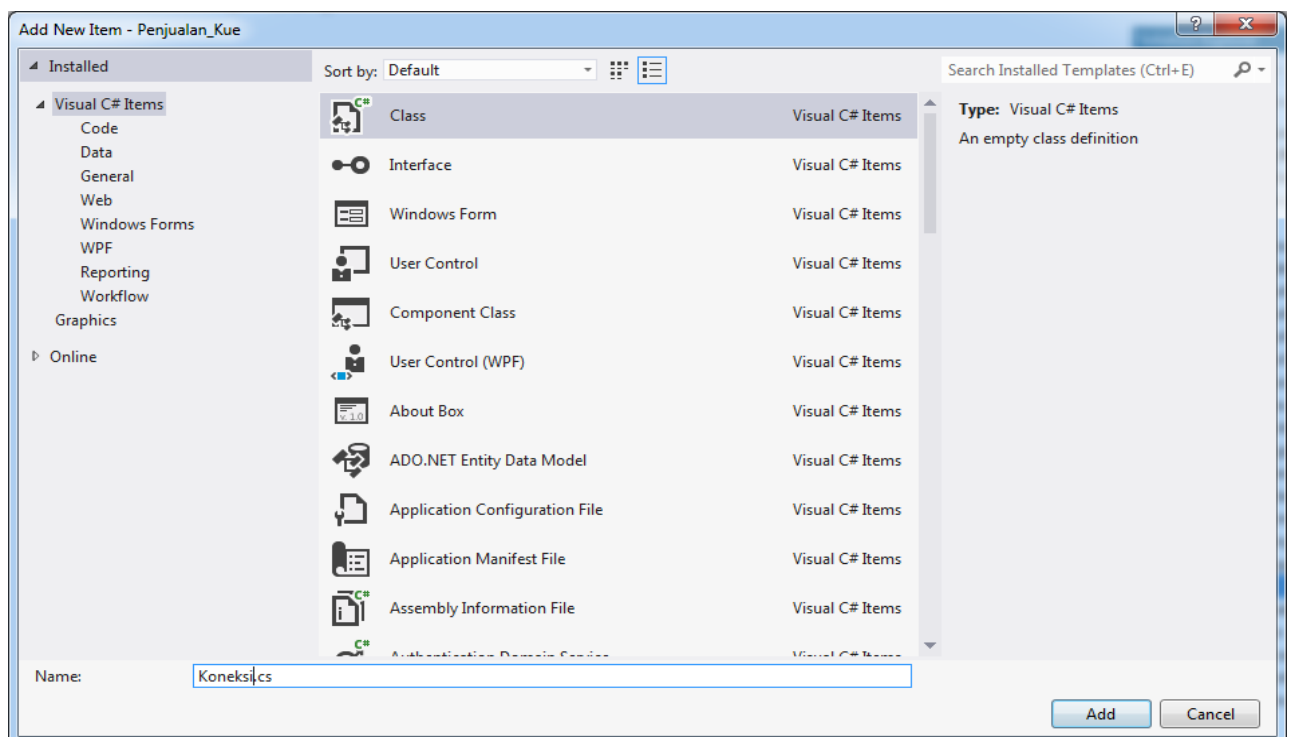
## Kelas Koneksi dan Coding Form Login

Materi Pembahasan :

1. Membuat Kelas Koneksi
2. Coding pada Form Login

### 7.1 Membuat Kelas Koneksi

1. Kelas koneksi akan berisi perintah untuk mengintegrasikan DBMS SQL SERVER khususnya database yang sudah dibuat pada modul 1 dengan project program penjualan. Kelas koneksi hanya dibuat sekali, namun akan digunakan oleh beberapa form yang butuh koneksi ke dalam database. Untuk membuat kelas koneksi, klik kanan pada package/folder Kelas, pilih Add – New Item – Class.
2. Simpan Class dengan nama Koneksi.cs



3. 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.Linq;
using System.Text;
// Tambahkan library koneksi ke SQL Server
using System.Data.SqlClient;

```

4. Ketikkan kode berikut di dalam class Koneksi { }

```

class Koneksi
{
    public SqlConnection GetConn()
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString = "Data Source = Acer; Initial Catalog = db_penjualan_kue; Integrated Security = true";
        return conn;
    }
}

```

## 7.2 Coding Pada Form Login

1. Aktifkan form login
2. Dari desain form login, pindah ke view code (F7)
3. Tambahkan kode berikut di bagian header untuk menambahkan library koneksi ke SQL Server

```

// Tambahkan library baru
using System.Data.SqlClient;

```

4. Ketikkan kode berikut di bawah public partial class FLogin : Form{

```

// Tambahkan variabel cmd untuk menampung perintah SQL
private SqlCommand cmd;
// Import kelas koneksi.cs menjadi objek --> konn
Kelas.Koneksi konn = new Kelas.Koneksi();

```

5. Kembali ke desain Form Login, klik 2 kali button login  
Ketik kode berikut :

```

private void btnLogin_Click(object sender, EventArgs e)
{
    // Coding button login
    SqlDataReader reader = null;
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("SELECT * FROM tbl_login WHERE username = '"
        + txtUsername.Text + "' AND password = '" + txtPassword.Text + "'", conn);
    cmd.ExecuteNonQuery();
    reader = cmd.ExecuteReader();
}

```

6. Di baris berikutnya, tambahkan kode berikut :

```
if (reader.Read()) // Jika usser dan password benar
{
    MessageBox.Show("Login Berhasil", "Informasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

    // Mengaktifkan semua menu dan toolbar yang sempat di non aktifkan
    FMenuUtama.mdiObj.menuMaster.Enabled = true;
    FMenuUtama.mdiObj.menuTransaction.Enabled = true;
    FMenuUtama.mdiObj.subMenuLogout.Enabled = true;
    FMenuUtama.mdiObj.subMenuLogin.Enabled = false;
    FMenuUtama.mdiObj.toolKasir.Enabled = true;
    FMenuUtama.mdiObj.toolPelanggan.Enabled = true;
    FMenuUtama.mdiObj.toolSupplier.Enabled = true;
    FMenuUtama.mdiObj.toolKue.Enabled = true;
    FMenuUtama.mdiObj.toolPenyuplaian.Enabled = true;
    FMenuUtama.mdiObj.toolPembelian.Enabled = true;
    this.Close(); // Form login keluar
}
```

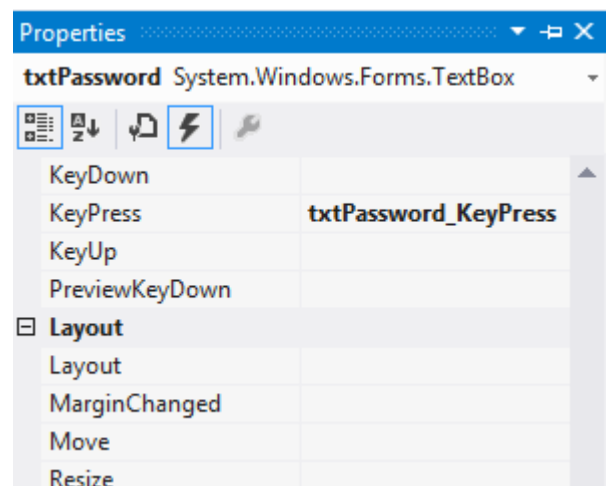
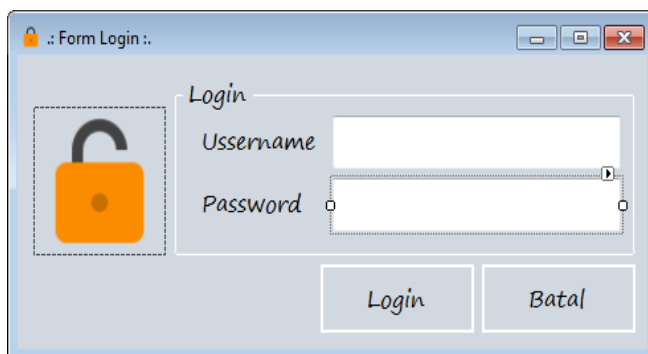
7. Di baris selanjutnya, ketikkan kode berikut :

```
else // Jika usser atau password salah
{
    MessageBox.Show("Ussername atau Password Salah !!!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    txtUsername.Text = "";
    txtPassword.Text = "";
    txtUsername.Focus();
}
conn.Close(); // Tutup koneksi
```

8. Kembali ke desain Form Login, klik 2 kali button batal  
Ketik kode berikut :

```
private void btnBatal_Click(object sender, EventArgs e)
{
    // Keluar dari form login
    this.Close();
}
```

9. Kembali ke desain form Login, klik txtpassword, lalu pilih  
Properties Event KeyPress



10. Klik 2 kali Event KeyPress, lalu ketik kode berikut :

```
private void txtPassword_KeyPress(object sender, KeyPressEventArgs e)
{
    // Memberikan event pada tombol enter
    if (e.KeyChar == 13)
    {
        btnLogin_Click(sender, e); // Memanggil kode dari button login
    }
}
```

11. Running Program [Start]

# 8

## Coding Form Master Kue

Materi Pembahasan :

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

### 8.1 Koneksi Form Master Kue

1. Buka Form Master Kue, 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;
```

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

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

    // 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 Kue

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 kue akan dibuat 3 prosedur yaitu prosedur bersih, atur tombol dan refresh kue.

1. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur bersih
void Bersih()
{
    txtKodeKue.Clear();
    txtNamaKue.Clear();
    txtHargaKue.Clear();
    cmbJenisKue.Text = "-Pilih Jenis Kue-";
    txtStokKue.Clear();
}
```

2. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur atur tombol
void AturTombol(Boolean status)
{
    btnEdit.Enabled = status;
    btnHapus.Enabled = status;
}
```

3. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur refresh kue
void RefreshKue()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM kue", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "kue");
        listKue.DataSource = ds;
        listKue.DataMember = "kue";
        listKue.Refresh();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    finally
    {
        conn.Close();
    }
}
```

## 8.3 Coding Simpan, Edit, Hapus, Cari

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

Ketik kode berikut :

```
private void btnBaru_Click(object sender, EventArgs e)
{
    // Memanggil prosedur bersih dan atur tombol saat klik tombol baru
    Bersih();
    txtKodeKue.Focus();
    AturTombol(false); // Membuat nilai status pada prosedur menjadi false
    btnSimpan.Enabled = true; // Mengaktifkan button simpan
}
```

2. Kembali ke desain Form Kue, klik 2 kali button Simpan

Ketik kode berikut :

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // Perintah simpan data kue
    // Jika ada komponen inputan yang kosong
    if (txtKodeKue.Text.Trim() == "" || txtNamaKue.Text.Trim() == ""
        || txtHargaKue.Text.Trim() == "" || txtStokKue.Text.Trim() == "" ||
        cmbJenisKue.Text.Trim() == "")
    {
        // Tampilkan pesan bahwa data belum lengkap
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
    else
    {
        // Mempersiapkan SQL Connection
        SqlConnection conn = konn.GetConn();
        try
        {
            // Mempersiapkan SQL Command yang akan dieksekusi
            cmd = new SqlCommand("EXECUTE simpan_kue '" + txtKodeKue.Text + "','"
                + txtNamaKue.Text + "','" + txtHargaKue.Text + "','"
                + cmbJenisKue.Text + "','" + txtStokKue.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 kue
            Bersih();
            RefreshKue();
        }
        catch (Exception x)
        {
            MessageBox.Show(x.ToString());
        }
    }
}
```

3. Kembali ke desain Form Kue, klik komponen DataGridView listKue, lalu klik 2x event CellClick, ketik kode berikut :

```
private void listKue_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke textbox
    try
    {
        btnSimpan.Enabled = false;
        AturTombol(true);
        DataGridViewRow row = this.listKue.Rows[e.RowIndex];
        txtKodeKue.Text = row.Cells["Kd_kue"].Value.ToString();
        txtNamaKue.Text = row.Cells["Nama_kue"].Value.ToString();
        txtHargaKue.Text = row.Cells["Harga_kue"].Value.ToString();
        cmbJenisKue.Text = row.Cells["Jenis_kue"].Value.ToString();
        txtStokKue.Text = row.Cells["Stok_kue"].Value.ToString();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

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

```
private void btnEdit_Click(object sender, EventArgs e)
{
    // Perintah edit data kue
    if (txtKodeKue.Text.Trim() == "" || txtNamaKue.Text.Trim() == ""
        || txtHargaKue.Text.Trim() == "" || cmbJenisKue.Text.Trim() == "" ||
        txtStokKue.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_kue '" + txtNamaKue.Text + "','"
                + txtHargaKue.Text + "','" + cmbJenisKue.Text + "','"
                + txtStokKue.Text + "','" + txtKodeKue.Text + "'", conn);
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Data berhasil dirubah", "INFORMASI", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            Bersih();
            RefreshKue();
            // Memanggil prosedur atur tombol
            AturTombol(false);
            btnSimpan.Enabled = true;
        }
    }
}
```

```

        catch (Exception x)
        {
            MessageBox.Show(x.ToString());
        }
    }
}

```

5. Kembali ke desain Form Kue, klik 2 kali button Hapus  
Ketik kode berikut :

```

private void btnHapus_Click(object sender, EventArgs e)
{
    // Perintah hapus data kue
    if (MessageBox.Show("Yakin ingin menghapus data kue ?",
        "Konfirmasi", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
    {
        SqlConnection conn = konn.GetConn();
        cmd = new SqlCommand("EXECUTE hapus_kue '" + txtKodeKue.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data kue berhasil dihapus", "informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshKue();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
}

```

6. Kembali ke desain Form Kue, klik 2 kali button Keluar  
Ketik kode berikut :

```

private void btnKeluar_Click(object sender, EventArgs e)
{
    // Keluar form kue
    Close();
}

```

7. Kembali ke desain Form Kue, klik 2 kali txtCariKue,  
Ketik kode berikut :

```

private void txtCariKue_TextChanged(object sender, EventArgs e)
{
    // Perintah cari nama kue
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("EXECUTE cari_kue '" + txtCariKue.Text + "'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "kue");
    listKue.DataSource = ds;
    listKue.DataMember = "kue";
    listKue.Refresh();
    conn.Close();
}

```

8. Masih di mode view code, cari kode berikut
9. Ketik kode berikut di bawah InitializeComponent();

```
public FKue()
{
    InitializeComponent();
    // Panggil prosedur pada saat form kue pertama kali dijalankan
    Bersih();
    AturTombol(false);
    RefreshKue();
}
```

10. Running program, gambar 7.1 merupakan tampilan Form Master Kue yang sudah terkoneksi dengan (database) tabel barang

The screenshot shows a Windows application titled "Form Kue". It features an "Input Data Kue" section on the left with five input fields: "Kode Kue", "Nama Kue", "Harga Kue", "Jenis Kue" (a dropdown menu currently showing "-Pilih Jenis Kue-"), and "Stok Kue". Below these fields are five buttons: "Baru" (with a plus icon), "Edit" (with a pencil icon), "Simpan" (with a floppy disk icon), "Hapus" (with a trash icon), and "Keluar" (with a right arrow icon). On the right side, there is a "List Data Kue" section containing a table with four columns: "Kd\_kue", "Nama\_kue", "Harga\_kue", and "Jenis\_kue". The table lists 10 items, with the first item "KU001 Bolu Coklat" highlighted. Below the table is a search bar labeled "Cari Nama Kue :".

Kd_kue	Nama_kue	Harga_kue	Jenis_kue
KU001	Bolu Coklat	94000	Kue Bolu
KU002	Bolu Keju	11000	Kue Bolu
KU003	Kue Bolu Pandan	98000	Kue Bolu
KU004	Bolu Kukus Pelangi	90000	Kue Bolu
KU005	Black Forest	310000	kue Ulang
KU006	Cheese Cake	450000	kue Ulang
KU007	Brownis Selai Kacang	45000	Kue Brown
KU008	Red Velvet	500000	kue Ulang
KU009	Brownis RHUM Raisin	75000	Kue Brown
KU010	Brownis Tutup Keju	65000	Kue Brown

# 9

## Coding Form Master Kasir

Materi Pembahasan :

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

### 9.1 Coding Form Master Kasir

1. Buka Form Master Kasir, 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.Windows.Forms;  
// Tambahkan library SQL Server  
using System.Data.SqlClient;
```

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

```
public partial class FKasir : Form  
{  
    // Import kelas koneksi.cs yang berasal dari folder kelas menjadi objek --> konn  
    Kelas.Koneksi konn = new Kelas.Koneksi();  
    // Tambahkan beberapa variabel yang digunakan untuk proses CRUD  
    private SqlCommand cmd;  
    private DataSet ds;  
    private SqlDataAdapter da;
```

### 9.2 Membuat Prosedur Bersih, Atur Tombol dan Refresh Kasir

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 kasir akan dibuat 3 prosedur yaitu prosedur bersih, atur tombol dan refresh pelanggan.

1. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur bersih
void Bersih()
{
    txtIdKasir.Clear();
    txtNamaKasir.Clear();
    txtAlamatKasir.Clear();
    txtNoTelpKasir.Clear();
}
```

2. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur atur tombol
void AturTombol(Boolean status)
{
    btnEdit.Enabled = status;
    btnHapus.Enabled = status;
}
```

3. Pada baris berikutnya, Ketik kode berikut :

```
// Membuat prosedur refresh kasir
void RefreshKasir()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM kasir", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "kasir");
        listKasir.DataSource = ds;
        listKasir.DataMember = "kasir";
        listKasir.Refresh();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    finally
    {
        conn.Close();
    }
}
```

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

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

Ketik kode berikut :

```
private void btnBaru_Click(object sender, EventArgs e)
{
    // Memanggil prosedur bersih dan atur tombol saat klik tombol baru
    Bersih();
    txtIdKasir.Focus();
    AturTombol(false);
    btnSimpan.Enabled = true;
}
```

2. Kembali ke desain Form Kasir, klik 2 kali button Simpan

Ketik kode berikut :

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // Perintah simpan data kasir
    if (txtIdKasir.Text.Trim() == "" || txtNamaKasir.Text.Trim() == ""
        || txtAlamatKasir.Text.Trim() == "" || txtNoTelpKasir.Text.Trim() == "")
    {
        // Tampilkan pesan bahwa data belum lengkap
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
```

```
else
{
    // Mempersiapkan SQL Connection
    SqlConnection conn = konn.GetConn();
    try
    {
        // Mempersiapkan SQL Command yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_kasir '" + txtIdKasir.Text + "','" +
            txtNamaKasir.Text + "','" + txtAlamatKasir.Text + "','" + txtNoTelpKasir.Text + "'", conn);
        // Membuka koneksi database
        conn.Open();
        // Menampilkan 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();
        RefreshKasir();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

3. Kembali ke desain Form Kasir, klik komponen DataGridView listKasir, lalu klik 2x event CellClick, ketik kode berikut :



```

private void listKasir_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke textbox
    try
    {
        btnSimpan.Enabled = false;
        AturTombol(true);
        DataGridViewRow row = this.listKasir.Rows[e.RowIndex];
        txtIdKasir.Text = row.Cells["Id_kasir"].Value.ToString();
        txtNamaKasir.Text = row.Cells["Nama_kasir"].Value.ToString();
        txtAlamatKasir.Text = row.Cells["Alamat_kasir"].Value.ToString();
        txtNoTelpKasir.Text = row.Cells["No_telp"].Value.ToString();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}

```

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

```

private void btnEdit_Click(object sender, EventArgs e)
{
    // Perintah edit data kasir
    if (txtIdKasir.Text.Trim() == "" || txtNamaKasir.Text.Trim() == ""
        || txtAlamatKasir.Text.Trim() == "" || txtNoTelpKasir.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_kasir '" + txtIdKasir.Text + "','" +
            txtNamaKasir.Text + "','" + txtAlamatKasir.Text + "','" + txtNoTelpKasir.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data berhasil dirubah", "INFORMASI", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshKasir();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
}

```

5. Kembali ke desain Form Kasir, klik 2 kali button Hapus, ketik kode berikut :

```
private void btnHapus_Click(object sender, EventArgs e)
{
    // Perintah hapus data kasir
    if (MessageBox.Show("Yakin ingin menghapus data kasir ?", "Konfirmasi",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        SqlConnection conn = konn.GetConn();
        cmd = new SqlCommand("EXECUTE hapus_kasir '" + txtIdKasir.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data kasir berhasil dihapus", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshKasir();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
}
```

6. Kembali ke desain Form Kasir, klik 2 kali button Keluar  
Ketik kode berikut :

```
private void btnKeluar_Click(object sender, EventArgs e)
{
    // Keluar form kasir
    Close();
}
```

7. Kembali ke desain Form Kasir, klik 2 kali txtCariKasir, ketik kode berikut :

```
private void txtCariKasir_TextChanged(object sender, EventArgs e)
{
    // Perintah cari nama kasir
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("EXECUTE cari_kasir '" + txtCariKasir.Text + "'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "kasir");
    listKasir.DataSource = ds;
    listKasir.DataMember = "kasir";
    listKasir.Refresh();
    conn.Close();
}
```

8. Masih di mode view code, cari kode berikut
9. Ketik kode berikut di bawah InitializeComponent();

```
public FKasir()  
{  
    InitializeComponent();  
    // Panggil prosedur bersih dan atur tombol pada saat form kasir pertama kali dijalankan  
    Bersih();  
    AturTombol(false);  
    RefreshKasir();  
}
```

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

Id_kasir	Nama_kasir	Alamat_kasir	No_telp
K0001	Riski	Gunung Terang	081272762
K0002	Ajik	Palapa	085732412
K0003	Kusuma	Teluk	081281525
K0004	Panjik	PU	085711989
K0005	Setiawan	Kedaton	089645209

# 10

## Coding Form Master Supplier

Materi Pembahasan :

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

### 10.1 Coding Form Master Supplier

1. Buka Form Master Supplier, 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.Windows.Forms;  
// Tambahkan library SQL Server  
using System.Data.SqlClient;
```

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

```
public partial class FSupplier : Form  
{  
    // Import kelas koneksi.cs yang berasal dari folder kelas menjadi objek --> konn  
    Kelas.Koneksi konn = new Kelas.Koneksi();  
    // Tambahkan beberapa variabel yang digunakan untuk proses CRUD  
    private SqlCommand cmd;  
    private DataSet ds;  
    private SqlDataAdapter da;
```

## 10.2 Membuat Prosedur Bersih, Atur Tombol dan Refresh Supplier

4. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur bersih
void Bersih()
{
    txtIdSupplier.Clear();
    txtNamaSupplier.Clear();
    txtAlamatSupplier.Clear();
    txtNoTelpSupplier.Clear();
}
```

5. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur atur tombol
void AturTombol(Boolean status)
{
    btnEdit.Enabled = status;
    btnHapus.Enabled = status;
}
```

6. Pada baris berikutnya, Ketik kode berikut :

```
// Membuat prosedur refresh supplier
void RefreshSupplier()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM supplier", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "supplier");
        listSupplier.DataSource = ds;
        listSupplier.DataMember = "supplier";
        listSupplier.Refresh();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    finally
    {
        conn.Close();
    }
}
```

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

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

Ketik kode berikut :

```
private void btnBaru_Click(object sender, EventArgs e)
{
    // Memanggil prosedur bersih dan atur tombol saat klik tombol baru
    Bersih();
    txtIdSupplier.Focus();
    AturTombol(false);
    btnSimpan.Enabled = true;
}
```

2. Kembali ke desain Form Supplier, klik 2 kali button Simpan

Ketik kode berikut :

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // Perintah simpan data supplier
    if (txtIdSupplier.Text.Trim() == "" || txtNamaSupplier.Text.Trim() == ""
        || txtAlamatSupplier.Text.Trim() == "" || txtNoTelpSupplier.Text.Trim() == "")
    {
        // Tampilkan pesan bahwa data belum lengkap
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
}
```

```
else
{
    // Mempersiapkan SQL Connection
    SqlConnection conn = konn.GetConn();
    try
    {
        // Mempersiapkan SQL Command yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_supplier '" + txtIdSupplier.Text + "','" +
            txtNamaSupplier.Text + "','" + txtAlamatSupplier.Text + "','" + txtNoTelpSupplier.Text + "'", conn);
        // Membuka koneksi database
        conn.Open();
        // Menampilkan 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();
        RefreshSupplier();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

3. Kembali ke desain Form Supplier, klik komponen DataGridView listSupplier, lalu klik 2x event CellClick, ketik kode berikut :

```
private void listSupplier_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke textbox
    try
    {
        btnSimpan.Enabled = false;
        AturTombol(true);
        DataGridViewRow row = this.listSupplier.Rows[e.RowIndex];
        txtIdSupplier.Text = row.Cells["Id_supplier"].Value.ToString();
        txtNamaSupplier.Text = row.Cells["Nama_supplier"].Value.ToString();
        txtAlamatSupplier.Text = row.Cells["Alamat_supplier"].Value.ToString();
        txtNoTelpSupplier.Text = row.Cells["No_telp"].Value.ToString();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

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

```
private void btnEdit_Click(object sender, EventArgs e)
{
    // Perintah edit data supplier
    if (txtIdSupplier.Text.Trim() == "" || txtNamaSupplier.Text.Trim() == ""
        || txtAlamatSupplier.Text.Trim() == "" || txtNoTelpSupplier.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_supplier '" + txtIdSupplier.Text + "','" +
            txtNamaSupplier.Text + "','" + txtAlamatSupplier.Text + "','" + txtNoTelpSupplier.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data berhasil dirubah", "INFORMASI", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshSupplier();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
}
```

5. Kembali ke desain Form Supplier, klik 2 kali button Hapus, ketik kode berikut :

```
private void btnHapus_Click(object sender, EventArgs e)
{
    // Perintah hapus data supplier
    if (MessageBox.Show("Yakin ingin menghapus data supplier ?", "Konfirmasi",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        SqlConnection conn = konn.GetConn();
        cmd = new SqlCommand("EXECUTE hapus_supplier '" + txtIdSupplier.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data supplier berhasil dihapus", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshSupplier();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
}
```

6. Kembali ke desain Form Supplier, klik 2 kali button Keluar

Ketik kode berikut :

```
private void btnKeluar_Click(object sender, EventArgs e)
{
    // Keluar form supplier
    Close();
}
```

7. Kembali ke desain Form Supplier, klik 2 kali txtCariSupplier,

Ketik kode berikut :

```
private void txtCariSupplier_TextChanged(object sender, EventArgs e)
{
    // Perintah cari nama supplier
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("EXECUTE cari_supplier '" + txtCariSupplier.Text + "'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "supplier");
    listSupplier.DataSource = ds;
    listSupplier.DataMember = "supplier";
    listSupplier.Refresh();
    conn.Close();
}
```

8. Masih di mode view code, cari kode berikut

9. Ketik kode berikut di bawah InitializeComponent();

```
public FSupplier()
{
    InitializeComponent();
    // Panggil prosedur bersih dan atur tombol pada saat form supplier pertama kali dijalankan
    Bersih();
    AturTombol(false);
    RefreshSupplier();
}
```



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

Input Data Supplier

Id Supplier

Nama Supplier

Alamat Supplier

Nomor Telepon

Baru Edit Simpan Hapus Keluar

List Data Supplier

	Id_supplier	Nama_supplier	Alamat_supplier	No.
▶	S0001	Andik	Way Halim	089
	S0002	Putra	Pramuka	081
	S0003	Law	Kemiling	082
	S0004	Sandi	Pahoman	085
	S0005	Tama	Rajabasa	089
*				

Cari Nama Supplier :

# 11

## Coding Form Master Pelanggan

Materi Pembahasan :

4. Koneksi Form Master Pelanggan
5. Membuat Prosedur Bersih, Atur Tombol dan Refresh Pelanggan
6. Coding Simpan, Edit, Hapus dan Cari

### 11.1 Coding 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;
```

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

```
public partial class FPelanggan : Form
{
    // Import kelas koneksi.cs yang berasal dari folder kelas menjadi objek --> konn
    Kelas.Koneksi konn = new Kelas.Koneksi();
    // Tambahkan beberapa variabel yang digunakan untuk proses CRUD
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
```

## 11.2 Membuat Prosedur Bersih, Atur Tombol dan Refresh Kasir

4. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur bersih
void Bersih()
{
    txtIdPelanggan.Clear();
    txtNamaPelanggan.Clear();
    txtAlamatPelanggan.Clear();
    txtNoTelpPelanggan.Clear();
}
```

5. Pada baris berikutnya, ketik kode berikut :

```
// Membuat prosedur atur tombol
void AturTombol(Boolean status)
{
    btnEdit.Enabled = status;
    btnHapus.Enabled = status;
}
```

6. Pada baris berikutnya, Ketik kode berikut :

```
// Membuat prosedur refresh pelanggan
void RefreshPelanggan()
{
    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();
    }
}
```

## 11.3 Coding Simpan, Edit, Hapus dan Cari

7. Kembali ke desain Form Pelanggan, klik 2 kali button Baru  
Ketik kode berikut :

```
private void btnBaru_Click(object sender, EventArgs e)
{
    // Memanggil prosedur bersih dan atur tombol saat klik tombol baru
    Bersih();
    txtIdPelanggan.Focus();
    AturTombol(false);
    btnSimpan.Enabled = true;
}
```

8. Kembali ke desain Form Pelanggan, klik 2 kali button Simpan  
Ketik kode berikut :

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // Perintah simpan data pelanggan
    if (txtIdPelanggan.Text.Trim() == "" || txtNamaPelanggan.Text.Trim() == ""
        || txtAlamatPelanggan.Text.Trim() == "" || txtNoTelpPelanggan.Text.Trim() == "")
    {
        // Tampilkan pesan bahwa data belum lengkap
        MessageBox.Show("Data belum lengkap, lengkapi data", "WARNING", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
    else
    {
        // Mempersiapkan SQL Connection
        SqlConnection conn = konn.GetConn();
        try
        {
            // Mempersiapkan SQL Command yang akan dieksekusi
            cmd = new SqlCommand("EXECUTE simpan_pelanggan '" + txtIdPelanggan.Text + "','" +
                txtNamaPelanggan.Text + "','" + txtAlamatPelanggan.Text + "','" + txtNoTelpPelanggan.Text + "'", conn);
            // Membuka koneksi database
            conn.Open();
            // Menampilkan 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();
            RefreshPelanggan();
        }
        catch (Exception x)
        {
            MessageBox.Show(x.ToString());
        }
    }
}
```

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

```
private void listPelanggan_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke textbox
    try
    {
        btnSimpan.Enabled = false;
        AturTombol(true);
        DataGridViewRow row = this.listPelanggan.Rows[e.RowIndex];
        txtIdPelanggan.Text = row.Cells["Id_pelanggan"].Value.ToString();
        txtNamaPelanggan.Text = row.Cells["Nama_pelanggan"].Value.ToString();
        txtAlamatPelanggan.Text = row.Cells["Alamat_pelanggan"].Value.ToString();
        txtNoTelpPelanggan.Text = row.Cells["No_telp"].Value.ToString();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

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

```
private void btnEdit_Click(object sender, EventArgs e)
{
    // Perintah edit data pelanggan
    if (txtIdPelanggan.Text.Trim() == "" || txtNamaPelanggan.Text.Trim() == ""
        || txtAlamatPelanggan.Text.Trim() == "" || txtNoTelpPelanggan.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 '" + txtIdPelanggan.Text + "', '" +
            txtNamaPelanggan.Text + "', '" + txtAlamatPelanggan.Text + "', '" + txtNoTelpPelanggan.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data berhasil dirubah", "INFORMASI", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshPelanggan();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
}
```

11. Kembali ke desain form Pelanggan, klik 2 kali button Hapus,  
Ketik kode berikut :

```
private void btnHapus_Click(object sender, EventArgs e)
{
    // 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 '" + txtIdPelanggan.Text + "'", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Data pelanggan berhasil dihapus", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        Bersih();
        RefreshPelanggan();
        // Memanggil prosedur atur tombol
        AturTombol(false);
        btnSimpan.Enabled = true;
    }
}
```

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

```
private void btnKeluar_Click(object sender, EventArgs e)
{
    // Keluar form pelanggan
    Close();
}
```

13. Kembali ke desain form Pelanggan, klik 2 kali txtcaripelanggan,  
Ketik kode berikut :

```
private void txtCariPelanggan_TextChanged(object sender, EventArgs e)
{
    // 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();
}
```

14. Masih di mode view code, cari kode berikut

15. Ketik kode berikut di bawah InitializeComponent();

```
public FPelanggan()
{
    InitializeComponent();
    // Panggil prosedur bersih dan atur tombol pada saat form pelanggan pertama kali dijalankan
    Bersih();
    AturTombol(false);
    RefreshPelanggan();
}
```

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

Form Pelanggan

Input Data Pelanggan

Id Pelanggan

Nama Pelanggan

Alamat Pelanggan

Nomor Telepon

Baru Edit Simpan Hapus Keluar

List Data Pelanggan

	Id_pelanggan	Nama_pelanggan	Alamat_pelanggan
▶	P0001	Putri	Kedaton
	P0002	Siti	Palapa
	P0003	Rara	Teluk
	P0004	Inka	PU
	P0005	Ririn	Antasari
	P0006	Joko	Kedaton
	P0007	Otonq	PU
	P0008	Dicky	Antasari
	P0009	Rio	Gedong Aer
	P0010	Sherly	Teluk
*			

Cari Nama Pelanggan :

# 12

## Coding Form Transaksi Penyuplaian

Materi Pembahasan :

1. Coding Form Cari Supplier dan Cari Kue
2. Koneksi Form Transaksi Penyuplaian
3. Membuat Prosedur pada Form Transaksi Penyuplaian
4. Coding Pada Form Transaksi Penyuplaian

### 12.1 Coding Form Cari Supplier, Cari Kasir dan Cari Kue

1. Buka Form Cari Supplier(FCariSupplier), 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;
```

3. Ketikkan kode berikut di bawah public partial class FCariSupplier : Form

```
public partial class FCariSupplier : Form
{
    // Tambahkan beberapa variabel yang digunakan untuk proses read data
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
    public string kodeSupp, namaSupp = "";

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

4. Pada baris berikutnya, buat prosedur RefreshSupplier() yang berfungsi untuk memperbaharui data pada DataGridViewSupplier.



```
// Membuat prosedur refresh supplier
void RefreshSupplier()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM supplier", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "supplier");
        dataGridSupplier.DataSource = ds;
        dataGridSupplier.DataMember = "supplier";
        dataGridSupplier.Refresh();
    }

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

5. Pada baris berikutnya, cari kode berikut :
6. Ketik kode berikut di bawah InitializeComponent();

```
public FCariSupplier()
{
    InitializeComponent();
    // Membersihkan komponen inputan dan panggil proc refresh supplier
    txtCariSupplier.Clear();
    RefreshSupplier();
}
```

7. Kembali ke desain form cari supplier (FCariSupplier), klik 2x txtCariSupplier, lalu ketik kode berikut :

```
private void txtCariSupplier_TextChanged(object sender, EventArgs e)
{
    // Membuat perintah pencarian data supplier berdasarkan kode dan nama
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("SELECT * FROM supplier WHERE Nama_supplier LIKE '%" +
        txtCariSupplier.Text + "%' OR Id_supplier LIKE '%" +
        txtCariSupplier.Text + "%'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "supplier");
    dataGridSupplier.DataSource = ds;
    dataGridSupplier.DataMember = "supplier";
    dataGridSupplier.Refresh();
    conn.Close();
}
```

8. Kembali ke desain form cari supplier (FCariSupplier), klik komponen datagridview DataGridSupplier, klik 2x pada Event CellClick Ketik kode berikut :

```
private void dataGridSupplier_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke variabel global kodeSupp dan namaSupp
    try
    {
        DataGridViewRow row = this.dataGridSupplier.Rows[e.RowIndex];
        kodeSupp = row.Cells["Id_supplier"].Value.ToString();
        namaSupp = row.Cells["Nama_supplier"].Value.ToString();
        this.Close(); // Menutup form cari supplier
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}
```

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

```
// Membuat function ambil kode supplier
public String ambilKodeSupplier
{
    get
    {
        return kodeSupp;
    }
}

// Membuat function ambil nama supplier
public String ambilNamaSupplier
{
    get
    {
        return namaSupp;
    }
}
```

10. Tampilan pada form cari supplier (FCariSupplier) seperti pada gambar berikut :

Data Supplier

Cari Berdasarkan Kode / Nama Supplier :

	Id_supplier	Nama_supplier	Alamat_supplier	No_telp
▶	S0001	Andik	Way Halim	089672762170
	S0002	Putra	Pramuka	081232412567
	S0003	Law	Kemiling	082281525383
	S0004	Sandi	Pahoman	085711989435
	S0005	Tama	Rajabasa	08964520989...
*				

11. Buka Form Cari Kasir (FCariKasir), 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;

```

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

```

public partial class FCariKasir : Form
{
    // Tambahkan beberapa variabel yang digunakan untuk proses read data
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
    public string kodeKsr, namaKsr = "";

    // Import kelas koneksi yang berasal dari folder kelas menjadi objek --> konn
    Kelas.Koneksi konn = new Kelas.Koneksi();
}

```

14. Pada baris berikutnya, buat prosedur RefreshKasir() yang berfungsi untuk memperbaharui data pada DataGridViewKasir.

```

// Membuat prosedur refresh kasir
void RefreshKasir()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM kasir", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "kasir");
        dataGridViewKasir.DataSource = ds;
        dataGridViewKasir.DataMember = "kasir";
        dataGridViewKasir.Refresh();
    }
}

```

```

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

```

15. Pada baris berikutnya, cari kode berikut :

16. Ketik kode berikut di bawah InitializeComponent();

```

public FCariKasir()
{
    InitializeComponent();
    // Membersihkan komponen inputan dan panggil proc refresh kasir
    txtCariKasir.Clear();
    RefreshKasir();
}

```

17. Kembali ke desain form cari kasir (FCariKasir), klik 2x txtCariKasir, lalu ketik kode berikut :

```

private void txtCariKasir_TextChanged(object sender, EventArgs e)
{
    // Membuat perintah pencarian data kasir berdasarkan kode dan nama
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("SELECT * FROM kasir WHERE Nama_kasir LIKE '%" +
        txtCariKasir.Text + "%' OR Id_kasir LIKE '%" +
        txtCariKasir.Text + "%'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "kasir");
    dataGridKasir.DataSource = ds;
    dataGridKasir.DataMember = "kasir";
    dataGridKasir.Refresh();
    conn.Close();
}

```

18. Kembali ke desain form cari kasir (FCariKasir), klik komponen datagridview DataGridKasir, klik 2x pada Event CellClick Ketik kode berikut :

```

private void dataGridKasir_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke variabel global kodeKsr dan namaKsr
    try
    {
        DataGridViewRow row = this.dataGridKasir.Rows[e.RowIndex];
        kodeKsr = row.Cells["Id_kasir"].Value.ToString();
        namaKsr = row.Cells["Nama_kasir"].Value.ToString();
        this.Close(); // Menutup form cari pelanggan
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}

```

19. Pada baris berikutnya, buat function untuk mengambil nilai yang berasal dari variabel kodeKsr dan namaKsr.

```

// Membuat function ambil kode kasir
public String ambilKodeKasir
{
    get
    {
        return kodeKsr;
    }
}

// Membuat function ambil nama kasir
public String ambilNamaKasir
{
    get
    {
        return namaKsr;
    }
}

```

20. Tampilan pada form cari kasir (FCariKasir) seperti pada gambar berikut :

Data Kasir

Cari Berdasarkan Kode / Nama Kasir :

	Id_kasir	Nama_kasir	Alamat_kasir	No_telp
▶	K0001	Riski	Gunung Terang	081272762172
	K0002	Ajik	Palapa	08573241253...
	K0003	Kusuma	Teluk	081281525383
	K0004	Panjik	PU	085711989477
	K0005	Setiawan	Kedaton	08964520989...
*				

21. Buka Form Cari Kue (FCariKue), lalu pindah ke bagian kode program / view code (F7)

22. 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;
```

23. Ketikkan kode berikut di bawah public partial class FCariKue : Form {

```
public partial class FCariKue : Form
{
    // Tambahkan beberapa variabel yang digunakan untuk proses read data
    private SqlCommand cmd;
    private DataSet ds;
    private SqlDataAdapter da;
    public string kodeKue, namaKue, jnsKue, hargaKue = "";

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

24. Pada baris berikutnya, buat prosedur RefreshKue() yang berfungsi untuk memperbaharui data pada DataGridViewKue.

```
// Membuat prosedur refresh kue
void RefreshKue()
{
    SqlConnection conn = konn.GetConn();
    try
    {
        conn.Open();
        cmd = new SqlCommand("SELECT * FROM kue", conn);
        ds = new DataSet();
        da = new SqlDataAdapter(cmd);
        da.Fill(ds, "kue");
        dataGridViewKue.DataSource = ds;
        dataGridViewKue.DataMember = "kue";
        dataGridViewKue.Refresh();
    }
```

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

25. Pada baris berikutnya, cari kode berikut :

26. Ketik kode berikut di bawah InitializeComponent();

```
public FCariKue()
{
    InitializeComponent();
    // Membersihkan komponen inputan dan panggil proc refresh kue
    txtCariKue.Clear();
    RefreshKue();
}
```

27. Kembali ke desain form cari kue (FCariKue), klik 2x txtCariKue, lalu ketik kode berikut :



```

private void txtCariKue_TextChanged(object sender, EventArgs e)
{
    // Membuat perintah pencarian data kue berdasarkan kode atau nama
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("SELECT * FROM kue WHERE Nama_kue LIKE '%" +
        txtCariKue.Text + "%' OR Kd_kue LIKE '%" +
        txtCariKue.Text + "%'", conn);
    ds = new DataSet();
    da = new SqlDataAdapter(cmd);
    da.Fill(ds, "kue");
    dataGridKue.DataSource = ds;
    dataGridKue.DataMember = "kue";
    dataGridKue.Refresh();
    conn.Close();
}

```

28. Kembali ke desain form cari kue (FCariKue), klik komponen datagridview DataGridKue, klik 2x pada Event CellClick Ketik kode berikut :

```

private void dataGridKue_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Mengambil nilai dari record yang dipilih menuju ke variabel global
    try
    {
        DataGridViewRow row = this.dataGridKue.Rows[e.RowIndex];
        kodeKue = row.Cells["Kd_kue"].Value.ToString();
        namaKue = row.Cells["Nama_kue"].Value.ToString();
        hargaKue = row.Cells["Harga_kue"].Value.ToString();
        jnsKue = row.Cells["Jenis_kue"].Value.ToString();
        this.Close(); // Menutup form cari kue
    }
    catch (Exception x)
    {
        MessageBox.Show(x.ToString());
    }
}

```

29. Pada baris berikutnya, buat function untuk mengambil nilai yang berasal dari variabel kodeKue, namaKue, jenisKue dan hargaKue.

```
// Membuat function ambil kode kue
```

```
public String ambilKodeKue
```

```
{
```

```
    get
```

```
    {
```

```
        return kodeKue;
```

```
    }
```

```
}
```

```
// Membuat function ambil nama kue
```

```
public String ambilNamaKue
```

```
{
```

```
    get
```

```
    {
```

```
        return namaKue;
```

```
    }
```

```
}
```

```
// Membuat function ambil jenis kue
```

```
public String ambilJenisKue
```

```
{
```

```
    get
```

```
    {
```

```
        return jnsKue;
```

```
    }
```

```
}
```

```
// Membuat function ambil harga kue
```

```
public String ambilHargaKue
```

```
{
```

```
    get
```

```
    {
```

```
        return hargaKue;
```

```
    }
```

```
}
```

30. Tampilan pada form cari kue (FCariKue) seperti pada gambar berikut :

Data Kue

Cari Berdasarkan Kode / Nama Kue :

	Kd_kue	Nama_kue	Harga_kue	Jenis_kue	Stok_kue
▶	KU001	Bolu Coklat	94000	Kue Bolu	20
	KU002	Bolu Keju	11000	Kue Bolu	40
	KU003	Kue Bolu Pan...	98000	Kue Bolu	50
	KU004	Bolu Kukus P...	90000	Kue Bolu	12
	KU005	Black Forest	310000	kue Ulang Ta...	75
	KU006	Cheese Cake	450000	kue Ulang Ta...	20
	KU007	Brownis Selai...	45000	Kue Brownis	20
	KU008	Red Velvet	500000	kue Ulang Ta...	15
	KU009	Brownis RHU...	75000	Kue Brownis	30
	KU010	Brownis Tut...	65000	Kue Brownis	25
*					

## 12.2 Koneksi Form Transaksi Penyuplaian

1. Buka Form Penyuplaian (FPenyuplaian), 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.Windows.Forms;
// Tambahkan library SQL Server
using System.Data.SqlClient;
```

3. Ketikkan kode berikut di bawah public partial class FPenyuplaian : Form

```
{
public partial class FPenyuplaian : Form
{
```

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

// Tambahkan variabel untuk perintah SQL
private SqlCommand cmd;
private SqlDataReader rd;
private DataSet ds;
private SqlDataAdapter da;
```

## 12.3 Membuat Prosedur Pada Form Transaksi Penyuplaian

1. Pada baris berikutnya, ketik kode berikut untuk membuat prosedur Bersih

:

```
// Prosedur bersih
private void Bersih()
{
    txtKodeSuplai.Text = "";
    dtpTanggalPenyuplaian.Value = DateTime.Now;
    txtIdSupplier.Text = "";
    txtNamaSupplier.Text = "";
    txtIdKasir.Text = "";
    txtNamaKasir.Text = "";
    txtKodeKue.Text = "";
    txtNamaKue.Text = "";
    txtJenisKue.Text = "";
    txtJumlahSuplai.Text = "0";
}
```

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

```
// Prosedur awal
private void Awal()
{
    // Panggil prosedur bersih di dalam prosedur awal
    Bersih();
    dataGridPenyuplaian.Enabled = false; // Menonaktifkan datagrid penyuplaian
    SqlConnection conn = konn.GetConn();
    {
        try
        {
            conn.Open();
            cmd = new SqlCommand("SELECT * FROM penyuplaian ORDER BY Kd_suplai DESC", conn);
            ds = new DataSet();
            da = new SqlDataAdapter(cmd);
            da.Fill(ds, "penyuplaian");
            dataGridPenyuplaian.DataSource = ds;
            dataGridPenyuplaian.DataMember = "penyuplaian";
            dataGridPenyuplaian.Refresh();
        }
    }
}
```

```

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

```

3. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur AutoNumber :

```

// Prosedur auto number
private void AutoNumber()
{
    long hitung;
    string urut;
    SqlConnection conn = konn.GetConn();
    conn.Open();
    cmd = new SqlCommand("SELECT Kd_suplai FROM penyuplaian WHERE Kd_suplai IN(SELECT MAX(Kd_suplai) FROM penyuplaian) ORDER BY Kd_suplai DESC", conn);
    rd = cmd.ExecuteReader();
    rd.Read();
    if (rd.HasRows) // Jika sudah ada data transaksi pembelian sebelumnya
    {
        hitung = Convert.ToInt64(rd[0].ToString().Substring(rd["Kd_suplai"].ToString().Length - 3, 3)) + 1;
        string joinstr = "000" + hitung;
        urut = "PE" + joinstr.Substring(joinstr.Length - 3, 3);
    }
    else // Jika belum ada data transaksi
    {
        urut = "PE001";
    }
    rd.Close();
    txtKodeSuplai.Text = urut;
    conn.Close();
}

```

4. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur SimpanMasterPenyuplaian:

```

// Prosedur simpan master penyuplaian
private void SimpanMasterPenyuplaian()
{
    // Mempersiapkan SQL Connection
    SqlConnection conn = konn.GetConn();
    {
        // Mempersiapkan SQL Command yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_penyuplaian '" + txtKodeSuplai.Text + "','" +
            + dtpTanggalPenyuplaian.Text + "','" + txtIdSupplier.Text + "','" + txtIdKasir.Text + "'", conn);
        // Membuka koneksi database
        conn.Open();
        // Menjalankan perintah query
        cmd.ExecuteNonQuery();
    }
}

```

5. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur SimpanDetailPenyuplaian:

```
// Prosedur simpan detail penyuplaian
private void SimpanDetailPenyuplaian()
{
    SqlConnection conn = konn.GetConn();
    {
        // Mempersiapkan SQL Command yang akan dieksekusi
        cmd = new SqlCommand("EXECUTE simpan_detail_penyuplaian '" + txtKodeSuplai.Text + "','" +
            + txtKodeKue.Text + "','" + txtJumlahSuplai.Text + "'", conn);
        // Membuka koneksi database
        conn.Open();
        // Menjalankan perintah query
        cmd.ExecuteNonQuery();
    }
}
```

6. Pada baris berikutnya, Ketik kode berikut untuk membuat prosedur RefreshPenyuplaian :

```
// Prosedur refresh penyuplaian
private void RefreshPenyuplaian()
{
    SqlConnection conn = konn.GetConn();
    {
        try
        {
            conn.Open();
            // Memanggil view_detail_penyuplaian
            cmd = new SqlCommand("SELECT * FROM view_detailpenyuplaian WHERE Kd_suplai = '" +
                + txtKodeSuplai.Text + "'", conn);
            ds = new DataSet();
            da = new SqlDataAdapter(cmd);
            da.Fill(ds, "view_detailpenyuplaian");
            dataGridPenyuplaian.DataSource = ds;
            dataGridPenyuplaian.DataMember = "view_detailpenyuplaian";
            dataGridPenyuplaian.Refresh();
        }
    }
}
```

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

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

```
// Prosedur refresh transaksi per kue
private void RefreshTransaksiPenyuplaian()
{
    // Memanggil proc refresh penyuplaian dan bersihkan beberapa komponen transaksi kue
    RefreshPenyuplaian();
    txtKodeKue.Clear();
    txtNamaKue.Clear();
    txtJenisKue.Text = "";
    txtJumlahSuplai.Text = "0";
    txtKodeKue.Focus();
}
```

## 12.4 Coding Pada Form Transaksi Penyuplaian

1. Kembali ke desain Form Penyuplaian, klik 2 kali button browse cari supplier (btnCariSupplier)

Ketik kode berikut :

```
private void btnCariSupplier_Click(object sender, EventArgs e)
{
    // Memanggil form cari supplier
    FCariSupplier cariSupp = new FCariSupplier();
    cariSupp.ShowDialog();
    txtIdSupplier.Text = cariSupp.ambilKodeSupplier;
    txtNamaSupplier.Text = cariSupp.ambilNamaSupplier;
    txtKodeKue.Focus();
}
```

2. Kembali ke desain Form Penyuplaian, klik 2 kali button browse cari kasir (btnCariKasir)

Ketik kode berikut :

```
private void btnCariKasir_Click(object sender, EventArgs e)
{
    // Memanggil form cari kasir
    FCariKasir cariKsr = new FCariKasir();
    cariKsr.ShowDialog();
    txtIdKasir.Text = cariKsr.ambilKodeKasir;
    txtNamaKasir.Text = cariKsr.ambilNamaKasir;
    txtKodeKue.Focus();
}
```

3. Kembali ke desain Form Penyuplaian, klik 2 kali button browse cari kue (btnCariKue)

Ketik kode berikut :

```
private void btnCariKue_Click(object sender, EventArgs e)
{
    // Memanggil form cari kue
    FCariKue cariKue = new FCariKue();
    cariKue.ShowDialog();
    txtKodeKue.Text = cariKue.ambilKodeKue;
    txtNamaKue.Text = cariKue.ambilNamaKue;
    txtJenisKue.Text = cariKue.ambilJenisKue;
    txtJumlahSuplai.Focus();
}
```

4. Kembali ke desain Form Penyuplaian, klik 2 kali button Baru, ketik kode berikut :

```
private void btnBaru_Click(object sender, EventArgs e)
{
    // Memanggil prosedur awal dan auto number ketika button baru dijalankan
    Awal();
    AutoNumber();
    dtpTanggalPenyuplaian.Focus();
}
```

5. Kembali ke desain Form Penyuplaian, klik 2 kali button Tambah (btnTambah)

Ketik kode berikut :

```
private void btnTambah_Click(object sender, EventArgs e)
{
    // Simpan data penyuplaian per barang (trigger untuk automasi stok)
    dataGridPenyuplaian.Enabled = true; // Mengaktifkan datagrid penyuplaian
    SqlDataReader reader = null;
    SqlConnection conn = konn.GetConn();
    {
        // Jika ada data yang kosong
        if (txtKodeSuplai.Text.Trim() == "" || txtIdSupplier.Text.Trim() == "" ||
            txtIdKasir.Text.Trim() == "" || txtKodeKue.Text.Trim() == "" ||
            txtJenisKue.Text.Trim() == "" || txtJumlahSuplai.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 penyuplaian WHERE Kd_suplai = '" +
        txtKodeSuplai.Text + "'", conn);
    cmd.ExecuteNonQuery();
    reader = cmd.ExecuteReader();
    if (reader.Read()) // Jika data penyuplaian sudah ada
    {
        // Refresh koneksi
        conn.Close();
        conn.Open();

        cmd = new SqlCommand("SELECT * FROM detail_penyuplaian WHERE Kd_suplai = '" +
            txtKodeSuplai.Text + "' AND Kd_kue = '" + txtKodeKue.Text + "'", conn);
        cmd.ExecuteNonQuery();
        reader = cmd.ExecuteReader();

        if (reader.Read()) // Jika data kue sudah diinputkan ke tabel detail_penyuplaian
        {
            MessageBox.Show("Data kue " + txtNamaKue.Text + " sudah diinputkan, harap pilih barang yang lain",
                "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
        else // Jika data kue belum diinputkan ke tabel detail_penyuplaian
        {
            // Panggil prosedur detail_penyuplaian
            SimpanDetailPenyuplaian();
        }
    }
}
```

```
else // Jika data penyuplaian belum ada
{
    // Panggil prosedur simpan master dan detail penyuplaian
    SimpanMasterPenyuplaian();
    SimpanDetailPenyuplaian();
}
// Panggil prosedur refresh transaksi penyuplaian
RefreshTransaksiPenyuplaian();
}
}
```



6. Kembali ke desain Form Penyuplaian, klik 2 kali button Kurang (btnKurang)

Ketik kode berikut :

```
private void btnKurang_Click(object sender, EventArgs e)
{
    // Hapus data penyuplaian per kue
    // Mempersiapkan koneksi database
    SqlDataReader reader = null;
    SqlConnection conn = konn.GetConn();
    {
        // Query menghapus data berdasarkan Kode Suplai dan Kode Kue pada tabel detail
        cmd = new SqlCommand("EXECUTE hapus_detailpenyuplaian '" + txtKodeSuplai.Text + "','" +
            txtKodeKue.Text + "'", conn);
        // Buka koneksi database
        conn.Open();
        // Eksekusi query
        cmd.ExecuteNonQuery();

        // Cek data di detail penyuplaian
        cmd = new SqlCommand("SELECT * FROM detail_penyuplaian WHERE Kd_suplai = '" +
            txtKodeSuplai.Text + "'", conn);
        cmd.ExecuteNonQuery();
        reader = cmd.ExecuteReader();

        // Memanggil proc refresh transaksi pembelian
        RefreshTransaksiPenyuplaian();
    }
}
```

7. Kembali ke desain Form Penyuplaian, klik event CellClick pada DataGridPenyuplaian

Ketik kode berikut :

```
private void dataGridPenyuplaian_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Tampilkan data penyuplaian per barang
    DataGridViewRow row = this.dataGridPenyuplaian.Rows[e.RowIndex];
    txtKodeKue.Text = row.Cells["Kd_kue"].Value.ToString();
    txtNamaKue.Text = row.Cells["Nama_kue"].Value.ToString();
    txtJenisKue.Text = row.Cells["Jenis_kue"].Value.ToString();
    txtJumlahSuplai.Text = row.Cells["Jml_suplai"].Value.ToString();
}
```

8. Kembali ke desain Form Penyuplaian, klik 2 kali button Simpan (btnSimpan), ketik kode berikut :

```
private void btnSimpan_Click(object sender, EventArgs e)
{
    // Transaksi selesai
    MessageBox.Show("Transaksi Selesai");

    // Report - Memanggil report bukti penyuplaian
    FBuktiPenyuplaian bp = new FBuktiPenyuplaian();
    bp.Show();
    Awal();
}
```

9. Kembali ke desain Form Penyuplaian, klik 2 kali button Keluar (btnKeluar), ketik kode berikut :

```
private void btnKeluar_Click(object sender, EventArgs e)
{
    // Keluar form transaksi penyuplaian
    Close();
}
```

10. Masih di mode view code, cari kode berikut

11. Ketik kode berikut di bawah InitializeComponent();

```
public FPenyuplaian()
{
    InitializeComponent();
    // Report - Mengisi variabel ObjTransaksi dengan nilai FPenyuplaian
    ObjTransaksi = this;
    // Memanggil prosedur awal
    Awal();
}
```

12. Running program

The screenshot shows the FPenyuplaian application interface. On the left, there are input fields for 'Data Penyuplaian' (Kode Suplai, Tanggal: 09/10/2018), 'Data Supplier' (Id Supplier, Nama Supplier), 'Data Kasir' (Id Kasir, Nama Kasir), and 'Data Kue' (Kode Kue, Nama Kue, Jenis Kue, Jumlah Suplai: 0). At the bottom left are buttons for 'Baru' (New), 'Simpan' (Save), and 'Keluar' (Exit). The main area on the right displays a table titled 'Data Penyuplaian' with columns: Kd\_suplai, TglSuplai, Id\_supplier, and Id\_kasir. The table contains five rows of data, with the first row (PE005, 08/05/18, S0005, K0005) selected.

Kd_suplai	TglSuplai	Id_supplier	Id_kasir
PE005	08/05/18	S0005	K0005
PE004	08/29/18	S0004	K0004
PE003	08/22/18	S0003	K0003
PE002	08/15/18	S0002	K0002
PE001	08/08/18	S0001	K0001

# 13

## Coding Form Transaksi Pembelian

Materi Pembahasan :

1. Coding Form Cari Pelanggan dan Cari Kue
2. Koneksi Form Transaksi Pembelian
3. Membuat Prosedur pada Form Transaksi Pembelian
4. Coding Pada Form Transaksi Pembelian

Pada dasar nya codingan transaksi pembelian sama dengan codingan transaksi penyuplaian hanya saja pada transaksi pembelian ditambahkan prosedur total seluruh untuk menjumlahkan total pembayaran oleh pelanggan. Pada transaksi pembelian yang melakukan transaksi adalah pelanggan dan kasir.

### 1. Prosedur TotalSeluruh

```
// Prosedur refresh total seluruh
private void totalseluruh()
{
    SqlConnection conn = konn.GetConn();
    {
        conn.Open();
        cmd = new SqlCommand("SELECT SUM(Total_bayar) " +
            " FROM view_pembelian WHERE Kd_beli = '" + txtKodeBeli.Text + "' ", conn);
        cmd.Connection = conn;
        int result = ((int)cmd.ExecuteScalar());
        String TotalSeluruh = result.ToString();
        lblTotalBayar.Text = result.ToString("#,##0");
        conn.Close();
    }
}
```