



Buku Latihan

# Visual Basic

untuk Mahasiswa

Inspirasi Membuat Aplikasi VB  
yang Mudah dan Cepat

Jubilee Enterprise



# **Buku Latihan Visual Basic untuk Mahasiswa**

**Jubilee Enterprise**

pustaka-indo.blogspot.com

PENERBIT PT ELEX MEDIA KOMPUTINDO



*KOMPAS GRAMEDIA*

## **Buku Latihan Visual Basic untuk Mahasiswa**

### **Jubilee Enterprise**

©2015, PT Elex Media Komputindo, Jakarta

Hak cipta dilindungi undang-undang

Diterbitkan pertama kali oleh

Penerbit PT Elex Media Komputindo

Kelompok Gramedia, Anggota IKAPI, Jakarta 2015

121150784

ISBN: 9786020263205

Sanksi Pelanggaran Pasal 113  
Undang-Undang Nomor 28 Tahun 2014  
tentang Hak Cipta

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta

Isi di luar tanggung jawab percetakan

# Kata Pengantar

Visual Basic masih memiliki daya tarik. Kemudahan yang dilihat dari sisi bahasa pemrogramannya membuat Visual Basic tetap eksis hingga sekarang. Anda bisa membuat aplikasi hitung-hitungan sederhana sampai aplikasi yang melibatkan banyak form dan koneksi database.

Buku ini ditujukan untuk semua pembaca, terutama mahasiswa yang ingin mencari pengetahuan tentang pemrograman Visual Basic. Dengan mempelajari Visual Basic, maka diharapkan para mahasiswa dapat mengerjakan tugas akhir pembuatan aplikasi yang melibatkan bahasa pemrograman tersebut.

Semoga dengan adanya buku ini, wawasan para pembaca bertambah dan bisa membuat aplikasi sendiri.

Selamat membaca!

Yogyakarta, 27 Februari 2015

Gregorius Agung

**Founder Jubilee Enterprise**

"Information Technology is Our Passion and Book is Our Way"

Do you need top-notch IT Book? Just .... [thinkjubilee.com](http://thinkjubilee.com)

# Daftar Isi

<b>Kata Pengantar.....</b>	<b>iii</b>
<b>Daftar Isi .....</b>	<b>iv</b>

<b>Bab 1 Selamat Datang di Visual Basic .....</b>	<b>1</b>
Sekilas MS Visual Studio 2013 .....	1
Download Visual Basic 2013 .....	2
Mengecek System Requirements.....	4
Instalasi Visual Studio Express 2013 .....	5
Meluncurkan Visual Studio 2013.....	7
Menggunakan SQL Server .....	9
Membuat Database .....	10
Menambahkan Tabel dalam Database .....	11
Menambahkan Data dalam Tabel.....	12
User Interface .....	14
Toolbar.....	15
Solution Explorer .....	16
Toolbox.....	17
Properties Window .....	18
Pembuatan Aplikasi Sederhana.....	19
Membuat Project Baru .....	19
Mengganti Nama MainWindow.xaml.....	21
Design User Interface.....	22
Menambahkan Sebuah Kontrol TextBlock .....	22
Kontrol Radio Button .....	23
Menambahkan Button .....	24
Menambahkan Program pada Button Display .....	25
Debugging.....	26
Debugging dengan Breakpoints .....	29

## **Bab2 Latihan Membuat Aplikasi-Aplikasi Sederhana .....31**

Aplikasi Penghitung Jumlah Tabungan .....	35
Bagaimana Script di Atas Bekerja? .....	37
Aplikasi untuk Menghitung Langganan PDAM .....	37
Bagaimana Script di Atas Bekerja? .....	40

## **Bab 3 Mengenal Database .....43**

SQL Server Management Studio.....	44
Mengambil Data Menggunakan Query SQL SELECT.....	56
Clause SELECT ... FROM .....	56
Clause WHERE .....	57

## **Bab 4 Latihan Pemrograman Visual Basic Database: Aplikasi Gudang .....61**

Menyiapkan Database .....	61
Menambahkan Tabel Barang .....	64
Menambahkan Tabel Suplier .....	66
Menambahkan Tabel Pelanggan .....	69
Menyiapkan Form Aplikasi .....	71
Form Utama .....	71
Form Barang .....	75
Form TambahBarang .....	78
Form HapusBarang .....	81
Form Pelanggan .....	83
Form TambahPelanggan .....	84
Form HapusPelanggan .....	86
Form Suplier .....	88
Form TambahSuplier .....	89
Form HapusSuplier .....	93
Menghubungkan Form dengan Database .....	95
Form Barang dengan Tabel Barang .....	97
Form Pelanggan dengan Tabel Pelanggan .....	102
Form Suplier dengan Tabel Suplier .....	107
Menambahkan Kode Program .....	111
Modul Koneksi .....	111
Kode Form Utama .....	117
Kode Form Barang .....	121
Kode Form Pelanggan .....	125
Kode Form Suplier .....	129
Kode Form Tambah Barang .....	133
Kode Form Tambah Pelanggan .....	137
Kode Form Tambah Suplier .....	141

Kode Form Hapus Barang .....	145
Kode Form Hapus Pelanggan .....	149
Kode Form Hapus Suplier .....	155
<b>Tentang Penulis.....</b>	<b>162</b>

# Bab 1

## Selamat Datang di Visual Basic

### Sekilas MS Visual Studio 2013

Jika Anda dulu pernah menggunakan Visual Basic dan bertahun-tahun kemudian tidak pernah mempelajari bahasa pemrograman ini, maka mungkin Anda akan sedikit terkecoh. Mengapa? Sebab Microsoft tidak lagi membuat Visual Basic sebagai sebuah aplikasi pemrograman.

Sebagai gantinya, Microsoft merilis MS Visual Studio yang salah satu fiturnya adalah untuk pemrograman Visual Basic. Jadi dapat diAndaikan seperti MS Office yang di dalamnya ada MS Word, MS Excel, dan seterusnya. Namun bedanya, Anda tidak bisa menginstal Visual Basic sebagai aplikasi yang terpisah dari Visual Studio. Saat buku ini ditulis, Visual Studio telah menginjak versi 2013.

Mengapa muncul Visual Studio? Sebab Microsoft ini mengintegrasikan banyak hal ke dalam “satu atap”. Visual Studio sendiri disebut dengan istilah Integrated Development Environment atau yang disingkat IDE. Salah satu wujudnya adalah, Anda bisa memasukkan script pemrograman non Visual Basic, misalnya C# dan C++, bersama-sama dengan script Visual Basic untuk membuat aplikasi yang lebih kompleks.

# Download Visual Basic 2013

Visual Basic 2013 dapat diunduh dari situs Microsoft. Microsoft menciptakan aplikasi untuk pemrograman Visual Basic 2013 yang dinamai Visual Studio. Dalam contoh buku ini, yang digunakan adalah Visual Studio Express 2013 yang dapat diunduh, diinstal, dan digunakan secara gratis selama 90 hari.

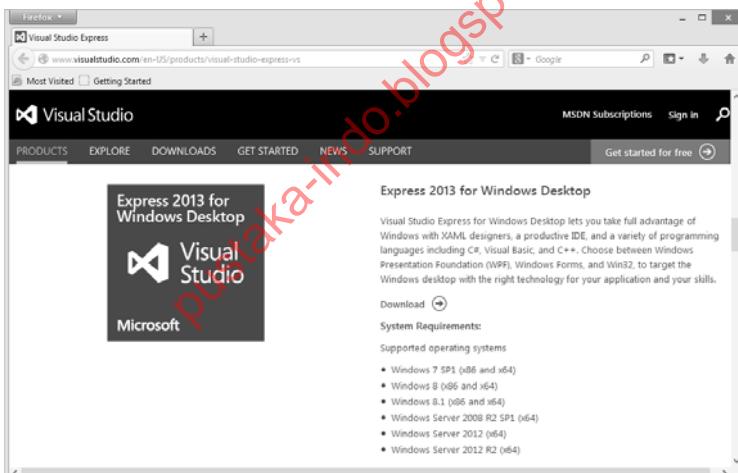
Aplikasi itu dapat diunduh dari internet. Oleh karena itu, koneksi internet diperlukan agar Anda bisa mendapatkan aplikasi ini.

Berikut langkah-langkahnya:

1. Buka browser Anda dan masuk ke dalam alamat berikut:

<http://www.visualstudio.com/en-US/products/visual-studio-express-vs>

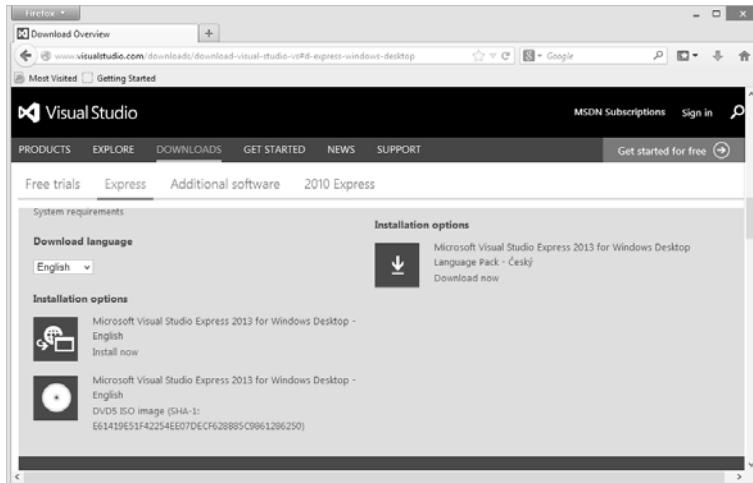
2. Gulung layar ke bawah sampai Anda menemukan pilihan **Express 2013 for Windows Desktop**.



*Pilihan 'Express 2013 for Windows Desktop'*

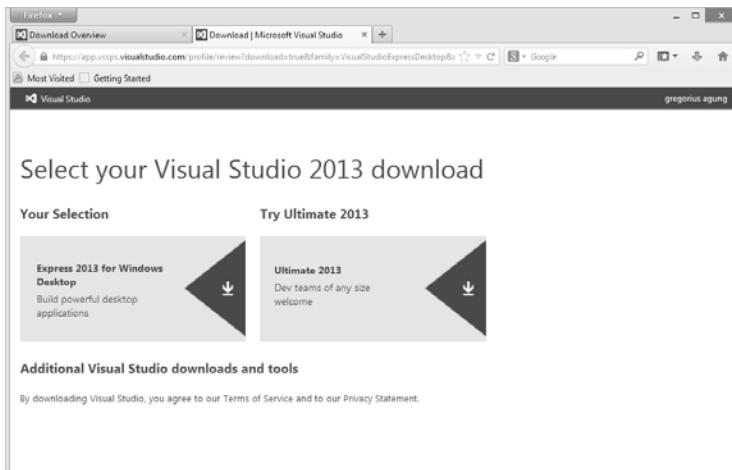
3. Klik link **Download** yang terlihat di situ.

4. Ada dua pilihan instalasi. Anda bisa memilih **Install Now** atau **DVD5 ISO Image**. Pada contoh kali ini, pilih opsi DVD5 ISO Image.



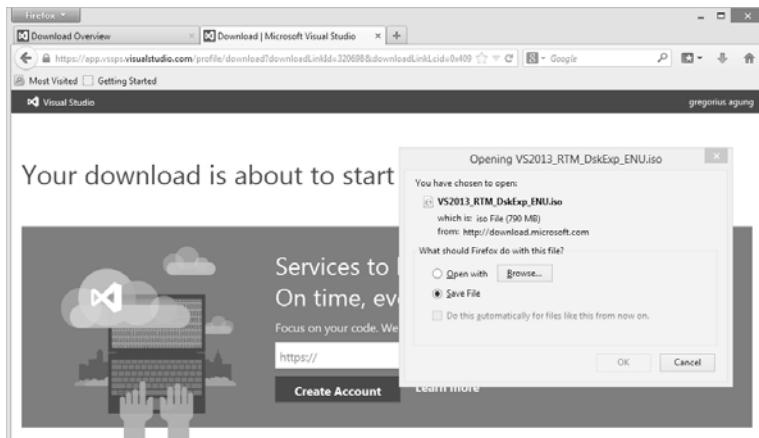
**Memilih DVD5 ISO Image**

5. Di antara pilihan, klik **Express 2013 for Windows Desktop** untuk penggunaan aplikasi bagi programmer pemula.



**Memilih opsi untuk Windows Desktop yang lebih ideal bagi programmer pemula**

6. Pilih **Save File** jika browser memunculkan kotak dialog untuk mengunduh aplikasi. Tekan tombol **OK** untuk memulai proses pengunduhan.



***Simpan installer aplikasi Visual Studio Express 2013***

Tunggu sampai proses pengunduhan selesai dilakukan. Langkah berikutnya apabila proses pengunduhan selesai adalah instalasi.

## Mengecek System Requirements

Karena di dalam buku ini menggunakan Visual Studio Express 2013, maka pastikan komputer dan sistem operasi Anda mendukung persyaratan minimum sebagai berikut:

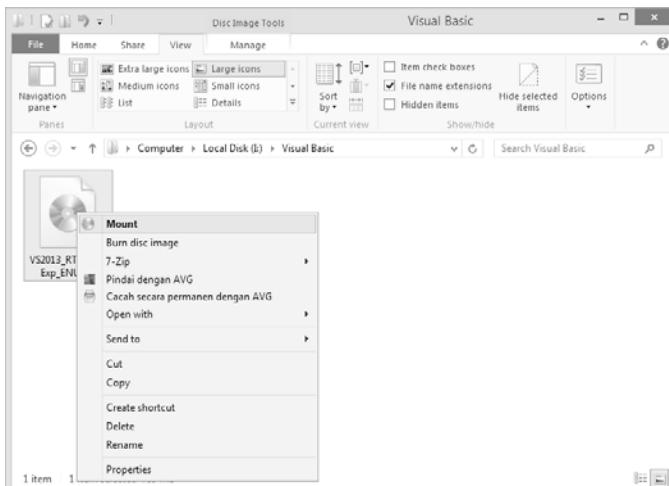
- Tidak support Windows XP dan Vista
- Windows 7 SP1 (x86 dan x64)
- Windows 8 (x86 dan x64)
- Windows 8.1 (x86 dan x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2013 (x64)
- Windows Server 2012 R2 (x64)

- Internet Explorer 10 (untuk komponen)
- Processor berkecepatan 1.6 GHz
- RAM minimal 1 Gb
- Hard disk sebesar 5 Gb
- DirectX 9 dengan resolusi minimal 1024 x 768

## Instalasi Visual Studio Express 2013

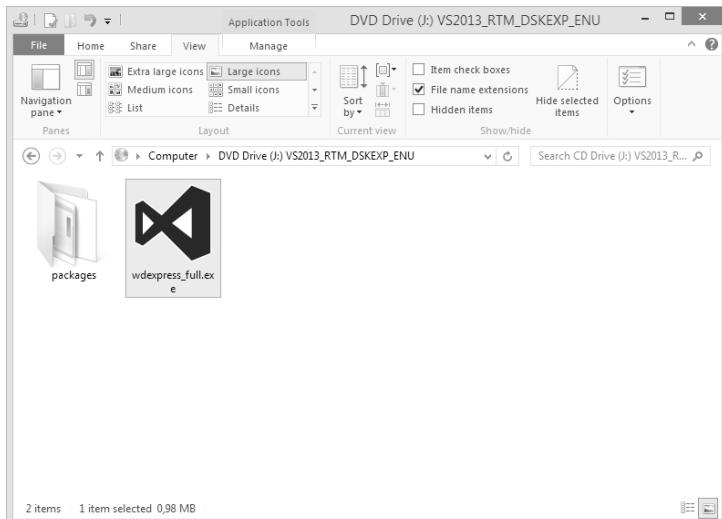
Kalau Anda sudah mengunduh DVD5 ISO Image seperti contoh langkah-langkah di atas dan memenuhi syarat system requirements seperti tertulis pada point-point sebelumnya, maka tugas berikutnya adalah melakukan instalasi. Berikut langkah-langkah melakukan instalasi terhadap Visual Studio Express 2013:

1. Buka File Explorer/Windows Explorer dan cari file yang telah diunduh tadi. Nama file tersebut semestinya: **VS2013\_RTM\_Dsk\_Exp\_ENU.iso**.
2. Klik-kanan dan pilihlah **Mount** dari context-menu yang terlihat di layar monitor.



**Pilih Mount untuk mengawali proses instalasi**

3. Selanjutnya akan terlihat ikon **wdexpress\_full.exe**. Klik-gAnda file itu untuk memulai proses instalasi.



***File wdxpress.full.exe yang bisa diklik-gAnda***

4. Aktifkan kotak cek **I agree to the License Terms and Privacy Policy**.



***Mengaktifkan I agree to the License Terms and Privacy Policy agar bisa melakukan instalasi***

5. Tekan tombol **Install** yang akan terlihat begitu mengklik opsi di atas.
6. Tunggu sampai proses instalasi selesai.



*Proses instalasi segera dimulai*

Duduklah dan tunggu proses instalasi usai. Anda bisa menyambangi pekerjaan lain, misalnya browsing internet atau melakukan kegiatan lainnya sampai proses instalasi tuntas. Jika proses instalasi berhasil, maka akan terlihat tulisan **LAUNCH**. Tekan tombol tersebut.

## Meluncurkan Visual Studio 2013

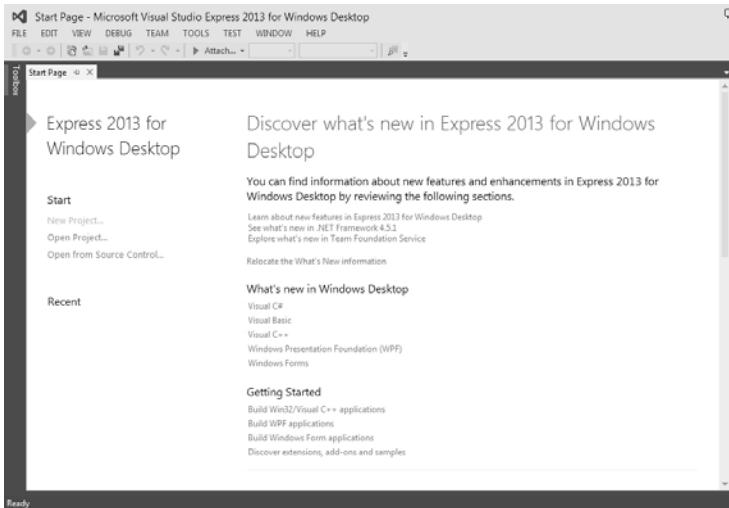
Sekarang, cobalah meluncurkan Visual Studio 2013. Jika Anda menggunakan MS Windows 8, maka prosesnya sebagai berikut:

1. Tekan tombol **Start** untuk masuk ke dalam Start Menu.
2. Pilih thumbnail **VS Express 2013 for Desktop**.



***Pilih thumbnail VS Express 2013 for Desktop***

3. Tunggu sampai Anda melihat tampilan Start Page pada aplikasi Microsoft Visual Studio Express 2013 for Windows Desktop.



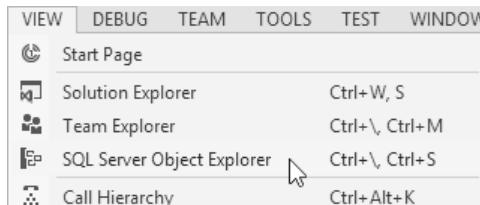
***Tampilan Visual Studio Express 2013***

4. Setelah itu, Anda sudah bisa membuat aplikasi menggunakan pemrograman Visual Basic 2013.

Kini, Anda sudah siap membuat aplikasi menggunakan pemrograman Visual Basic. Di dalam buku ini, Anda akan mempelajari tahap-tahap menggunakan Visual Basic untuk pembuatan aplikasi.

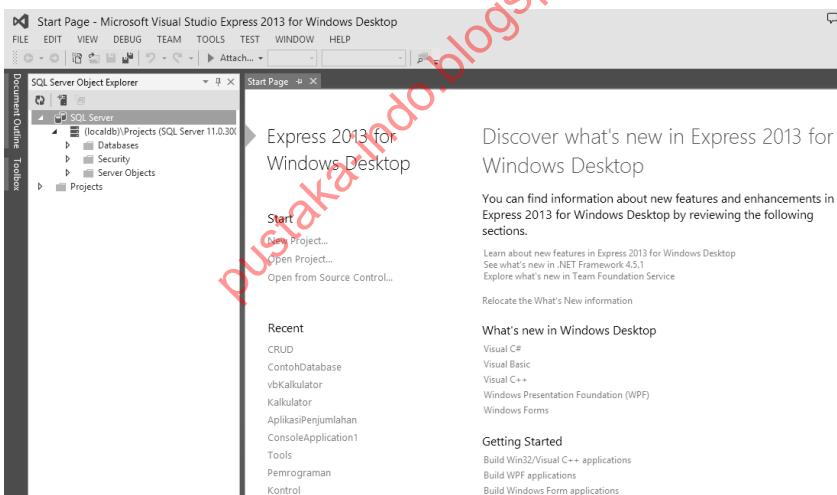
# Menggunakan SQL Server

Anda dapat mengelola database aplikasi dengan SQL Server secara langsung melalui Visual Studio.Untuk mengelola database SQL Server dalam Visual Studio, pertama-tama Anda perlu menampilkan jendela SQL Server Object Explorer. Pilih **View – SQL Server Object Explorer** untuk menampilkannya.



*Memilih opsi SQL Server Object Explorer*

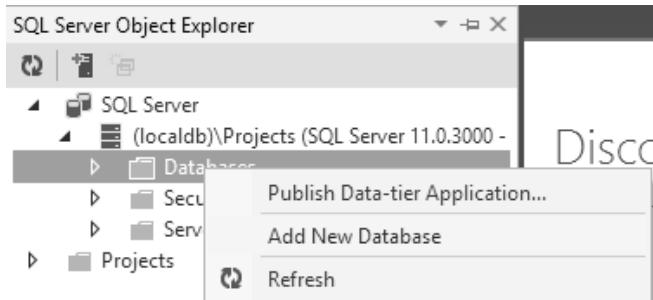
SQL Server Object Explorer kemudian akan ditampilkan di sebelah kiri Visual Studio.



*SQL Server Object Explorer ditampilkan*

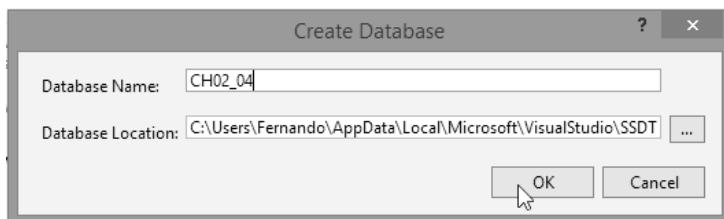
## Membuat Database

1. Pada **Object Explorer**, klik kanan folder **Database** dan pilih **Add New Database**.



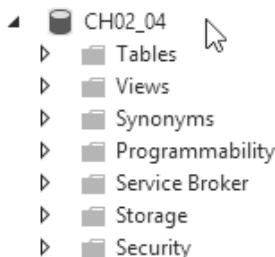
*Menambahkan database*

2. Jendela **New Database** akan muncul, masukkan nama database pada kotak **Database Name**, kemudian klik **OK**.



*Mengatur nama database*

3. Database kemudian sudah dapat digunakan.



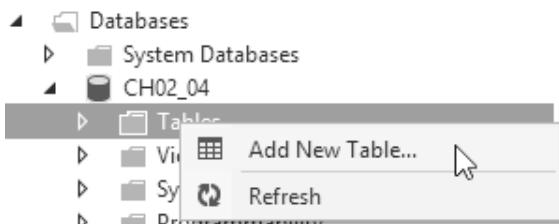
*Database berhasil dibuat*

# Menambahkan Tabel dalam Database

Dalam database, data-data disimpan dalam tabel-tabel yang saling berhubungan. Tabel terdiri dari kolom-kolom dengan tipe data tertentu dan baris-baris yang merupakan data yang akan digunakan dalam aplikasi.

Berikut ini langkah-langkah untuk menambahkan tabel dalam database:

1. Pada **Object Explorer**, perluas folder **Database**, perluas database yang akan digunakan, klik kanan folder **Tables** dan pilih **Add New Table**.



*Menambahkan tabel*

2. Atur property dari tabel yang akan dibuat, kemudian klik **Update** untuk membuat tabel.

A screenshot of the Microsoft SQL Server Management Studio (SSMS) Table Designer. The main window shows a table named 'Table' with one column 'Id' defined as 'int'. The 'Properties' pane on the right shows the table's properties: 'Name' is 'Table', 'Schema' is 'dbo', and 'Filegroup' is 'PRIMARY'. The 'Script' tab at the bottom contains the T-SQL code for creating the table:

```
CREATE TABLE [dbo].[Table]
(
    [Id] INT NOT NULL PRIMARY KEY
)
```

*Mengatur properti tabel yang akan dibuat*

3. Anda juga dapat membuat tabel dengan contoh query SQL berikut:

```
USE [CH02_03];
GO

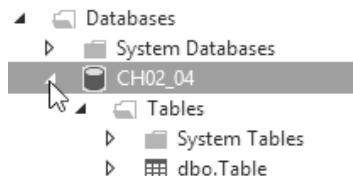
CREATE TABLE Book_Info
(
    Book_ID SMALLINT ,
    Book_Name VARCHAR(20) ,
    Description VARCHAR(30) ,
    Price [SMALLMONEY] ,
    Author_ID [int]
)

ON CH02_FG1;
GO
```

## Menambahkan Data dalam Tabel

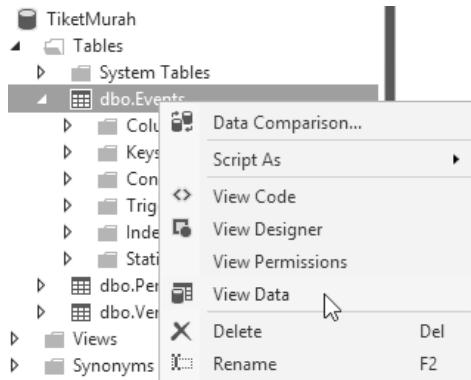
Berikut ini langkah-langkah untuk menambahkan data ke dalam tabel database:

1. Pada **Object Explorer**, perluas folder **Databases**, pilih database, dan perluas folder **Tables**.



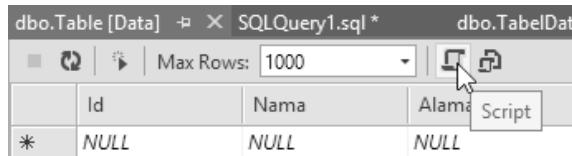
**Memperluas folder Tables**

2. Klik kanan tabel yang ingin dimodifikasi, kemudian pilih **View Data**.



**Memilih View Data**

3. Pilih **Script** di sebelah kanan atas.



**Memilih Script**

4. Masukkan kode untuk menambahkan database, kemudian klik **Execute**.

```
dbo.Table [Data] SQLQuery1.sql * dbo.TabelData [Data]
INSERT INTO [dbo].[TabelData]
Execute (Ctrl+Shift+E)
    [Id],
    [Nama],
    [Alamat],
    [Telepon]
)
```

**Memilih Execute**

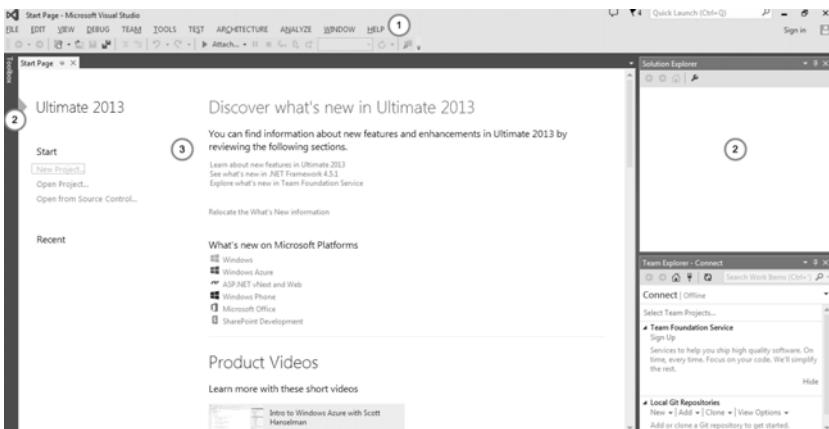
Data kemudian akan ditambahkan ke dalam tabel.

# User Interface

Pada saat Anda menjalankan aplikasi ini, maka Visual Studio 2013 akan memulainya dengan membuka sebuah halaman awal seperti yang ditunjukkan dalam gambar di bawah.

Bagian-bagian utama dari tampilan ini akan dapat dibedakan dengan mudah. Di bagian atas akan ditemukan menu, yang memberikan akses ke semua bagian maupun fungsi. Di bawah menu baru ada toolbar, di mana terdapat shortcut untuk fungsi-fungsi yang paling sering digunakan, seperti copy, paste, start debug, dan lainnya.

Quick Launch textbox bisa Anda temukan di bagian kanan atas, ataupun dengan kombinasi Ctrl+Q. Textbox ini digunakan untuk dengan cepat menuju ke suatu fungsi tanpa menggunakan rangkaian menu.



*Halaman awal <dilengkapi dengan informasi yang lebih banyak, misalnya "Halaman Awal Visual Studio 2013">*

Ruang kerja utama terletak di tengah-tengah. Pada saat sebuah solution atau project dibuka, editor, dan designer akan muncul di sini. Di bagian inilah Anda akan banyak bekerja nantinya. Toolbox tertambat di bagian kiri ruang kerja ini, dan tool window lain di sebelah kanannya. Toolbox di sebelah kiri ada dalam keadaan tertutup. Jika Anda klik judulnya, toolbox ini akan membuka. Jika fokus berpindah darinya, toolbox akan kembali menutup.

# Toolbar

Toolbar terdapat di bawah menu bar di bagian atas dari workspace.



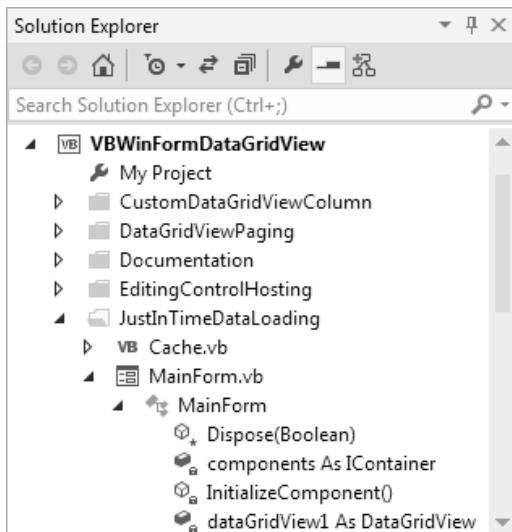
**Tampilan Toolbar Visual Studio 2013**

Icon	Keterangan
	<b>Navigate Backward.</b> Kembali ke tampilan sebelumnya.
	<b>Navigate Forward.</b> Maju ke tampilan berikut.
	<b>New Project.</b> Membuat project baru.
	<b>Open File.</b> Membuka berkas.
	<b>Save.</b> Simpan.
	<b>Save All.</b> Simpan semua berkas.
	<b>Comment out selected lines.</b> Jadikan teks yang terpilih sebagai comment.
	<b>Uncomment selected lines.</b> Hapus command dalam teks yang terpilih.
	<b>Undo.</b> Batalkan.
	<b>Redo.</b> Ulang.
	<b>Start.</b> Mulai proses (debugging).
	<b>Break All dan Stop Debugging.</b> Hentikan sementara, atau hentikan total debugging.
	<b>Step Into.</b> Masuk ke dalam sub-module.
	<b>Step Over.</b> Lompati sub-module.
	<b>Step Out.</b> Keluar dari sub-module.
	<b>Solution Configuration dan Browser Link.</b> Konfigurasi, bisa berupa Debug, Release, atau Configuration Manager.
	<b>Find in Files.</b> Cari teks dari berkas.

## Solution Explorer

Salah satu bagian dari tool window yang akan sering digunakan adalah bagian Solution Explorer. Bagian ini biasanya akan berada tertambat di docking window bagian kanan dari workspace.

Solution Explorer menyediakan tampilan yang terorganisir dari project Anda, dan termasuk di dalamnya semua item yang digunakan dalam project tersebut (lihat gambar di bawah ini).



*Panel Solution Explorer*

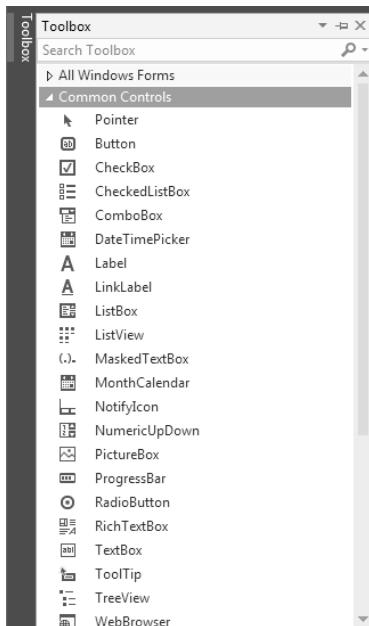
Tampilan project yang Anda buka akan dimunculkan dalam struktur tree view. View ini akan menampilkan project Anda sebagai satu kesatuan dengan nama project Anda menempati posisi atas dan item-item di dalam project itu akan ditampilkan dalam organisasi di bawah nama project tersebut.

Item-item ini bisa merupakan sebuah windows form, gambar-gambar yang dimunculkan dalam form, ataupun berupa script untuk koneksi database.

## Toolbox

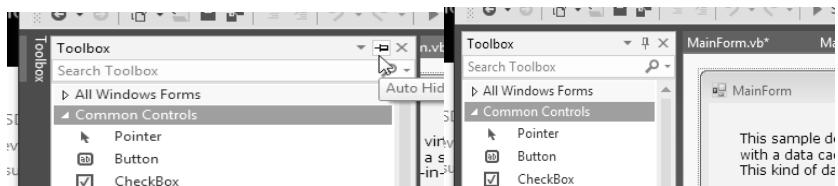
Toolbox akan berisikan tool-tool yang akan sering digunakan, tergantung dari konteks item apa yang dibuka di workspace. Misalnya Anda sedang membuka sebuah window form, maka toolbox yang dibuka akan menunjukkan kontrol window seperti Button, Combo Box, TextBox, dan lainnya.

Toolbox ini secara default akan nampak dalam keadaan tertutup, di sebelah kiri workspace. Jika Anda ingin membukanya, klik di bagian nama toolbox itu, dan toolbox akan membuka, dan akan menutup dirinya secara otomatis pada saat fokus berpindah darinya.



***Toolbox untuk design form***

Pada saat toolbox ini ditampilkan, bisa dilihat satu deret icon di bagian kanan atas window toolbox ini. Di sini Anda bisa mengubah struktur letak dari toolbox ini. Misalnya, Anda ingin membuatnya terus nampak, Anda dapat menekan icon yang berada di tengah, yang tampak seperti *pin*. Pin akan berputar 90° untuk menunjukkan bahwa window toolbox itu sekarang berada dalam keadaan tertambat.

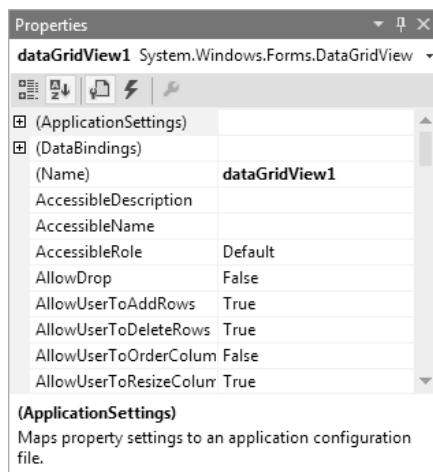


### Ketika Toolbox ditambatkan

Jika Anda perhatikan ruang kerja dalam Visual Studio 2013 ini, dapat dilihat bahwa deretan tiga icon tadi tidak hanya muncul dalam window toolbox, tapi juga muncul dalam window yang lain. Jadi, cara penambatan window tadi dapat juga diaplikasikan pada window-window yang lain, sesuai dengan kebutuhannya.

## Properties Window

Properties window biasanya ditampilkan di bagian sebelah kanan dari workspace. Isi dari properties window juga tergantung dari konteks yang terpilih dalam workspace. Jika toolbox akan memunculkan semua opsi yang dapat dipilih dari fokus di workspace, maka properties akan memunculkan data detail dari kontrol yang terpilih pada workspace.



Properties window untuk kontrol DataGridView

Dalam contoh gambar di atas, misalkan dipilih kontrol DataGridView pada workspace, maka detail dari kontrol tersebut akan dimunculkan pada properties window.

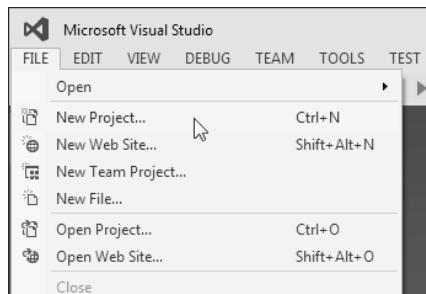
Dalam gambar tampak sebagian dari detail yang dapat diubah, misalnya namanya dan beberapa kondisi yang ditetapkan seperti AllowUserToAddRows, AllowUserToDeleteRows, dan lain-lain.

## Pembuatan Aplikasi Sederhana

Setelah Anda sedikit tahu tentang tampilan dasar dari Visual Studio 2013, maka sekarang Anda akan belajar tentang pembuatan sebuah aplikasi. Dalam aplikasi ini, Anda akan belajar mulai dari pembuatan sebuah project, design dari user interface-nya, sedikit *coding*, debugging aplikasi Anda, sampai dengan membangun versi akhir (*release version*) dari aplikasi Anda.

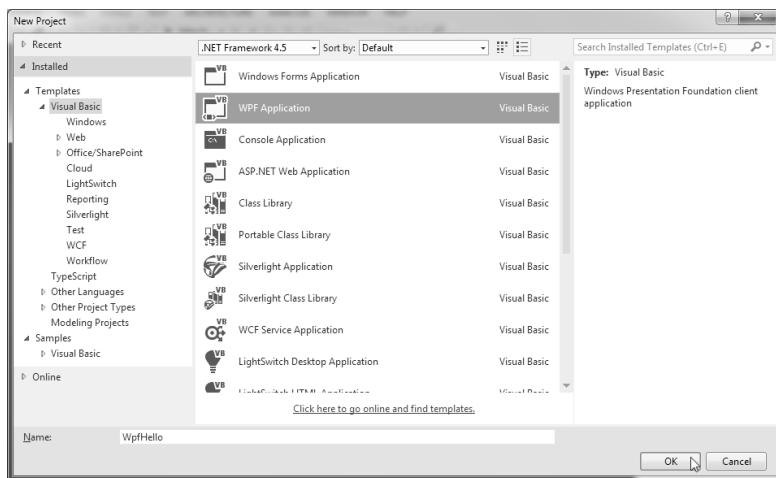
### Membuat Project Baru

Sebagai awal, Anda akan membuat sebuah project bernama Hello World. Dimulai dengan pilihan pada menu utama untuk **File, New Project**.



*Memilih opsi New Project*

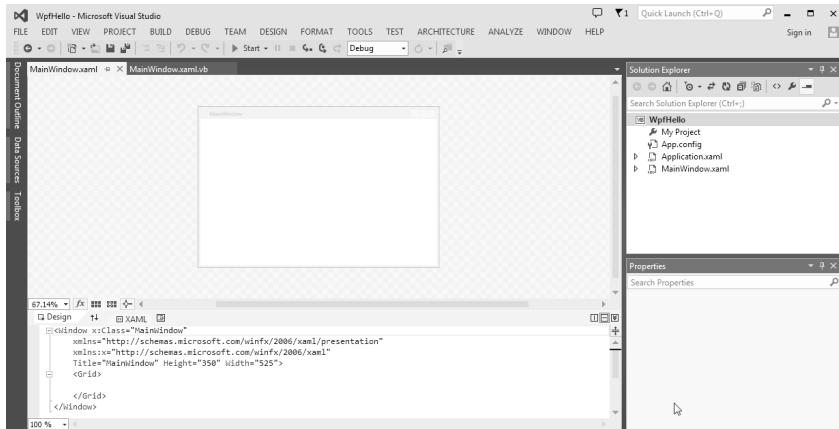
Pada dialog box berikutnya, Anda pilih untuk membuat sebuah aplikasi Windows dengan bahasa pemrograman Visual Basic dan dengan template WPF Application.



### Parameter New Project

Anda persiapkan juga untuk menamakan project itu dengan nama WpfHello pada bagian bawah dialog box tersebut.

Setelah Anda klik OK untuk membuat project tersebut, Visual Basic akan memproses pilihan Anda tersebut, dan pada akhirnya akan membuka project itu dengan sebuah halaman pembuka awal yang masih kosong, dan dengan seluruh kelengkapan project baru tersebut di dalam tool window **Solution Explorer**.



### Project baru dalam workspace

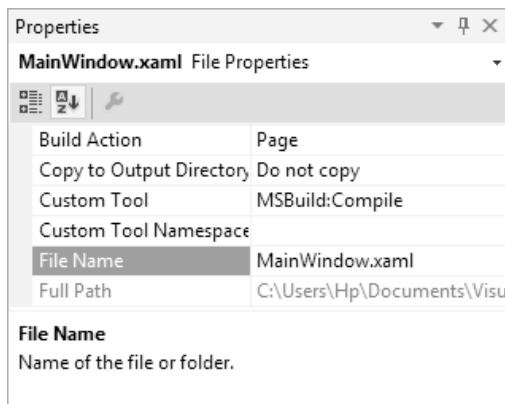
Halaman utama dari project baru WfpHello yang sudah dipersiapkan oleh Visual Basic, yaitu dalam hal ini adalah tampilan MainWindow.xaml akan dibuka di dalam workspace dalam WPF Designer di atas untuk tampilan design-nya, dan di bawah dalam tampilan XAML dalam satu layar.

## Mengganti Nama MainWindow.xaml

- Dalam proses berikut, Anda akan memberi nama yang lebih baik dari nama default MainWindow.xaml. Dalam **Solution Explorer**, pilih MainWindow.xaml. Perhatikan bahwa detail dari pilihan Anda tadi akan muncul dalam tool window **Properties** di bagian bawah **Solution Explorer**.

Jika tool window **Properties** tidak terlihat, pilihlah MainWindow.xaml dalam **Solution Explorer**, klik tombol kanan mouse, dan pilihlah **Properties**.

Gantilah properti untuk **File Name** menjadi Hello.xaml.



*Properti File Name*

Setelah penggantian tersebut, nama MainWindow.xaml yang sebelumnya ada di dalam Solution Explorer akan berganti menjadi Hello.xaml.

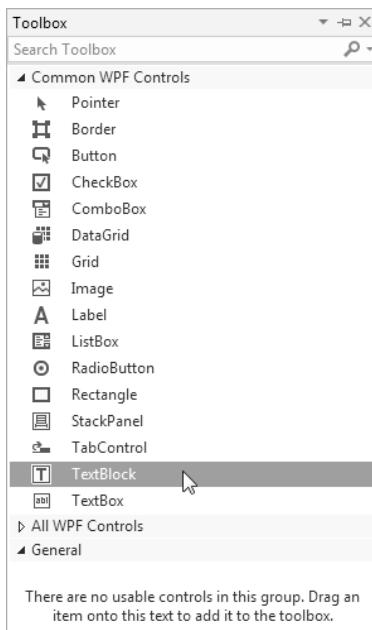
- Dalam **Solution Explorer**, bukalah Hello.xaml ke dalam designer view dan pilihlah bagian title bar dari tampilan window tersebut. Hal itu akan membuka detail baru dalam window Properties, dan carilah Title dan gantilah isinya menjadi Hello. Title bar dari window aplikasi Anda sekarang telah berganti nama menjadi Hello.

## Design User Interface

Dalam bagian ini, Anda akan menggunakan kontrol TextBlock, Radio Button, dan sebuah Button.

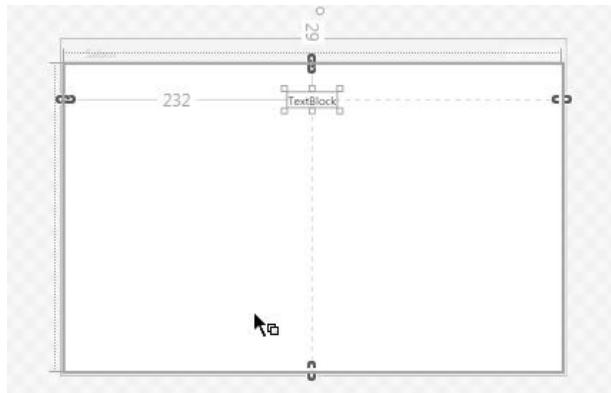
### Menambahkan Sebuah Kontrol TextBlock

- Buka window **ToolBox** dari sisi kiri workspace, dan pilih **TextBlock**.



#### *Kontrol TextBlock*

- Tarik kontrol tersebut dalam halaman kerja dalam designer view, kira-kira di bagian atas.



**Tampilan dalam mode design**

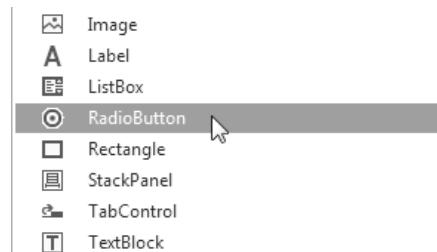
Text XAML untuk kontrol TextBlock tersebut akan terlihat seperti berikut:

```
<TextBlock HorizontalAlignment="Left" Margin="102.064,44,0,0" TextWrapping="Wrap" Text="TextBlock" VerticalAlignment="Top" Grid.Column="1" />
```

Kemudian, ubahan teks dalam **TextBlock** tersebut dari sebelumnya `Text="TextBlock"` diganti menjadi `Text="Bagaimana kabar Anda hari ini?"`.

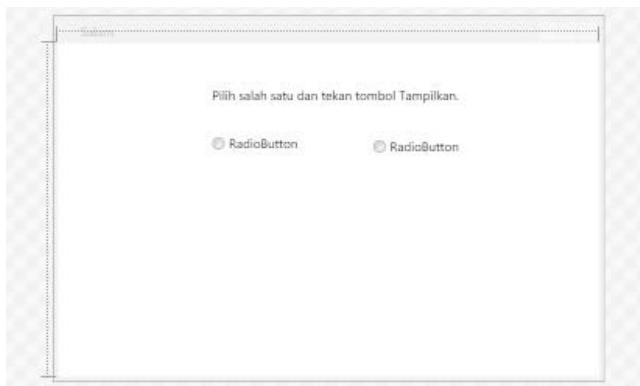
## Kontrol Radio Button

- Berikutnya, Anda akan menambahkan dua buah Radio Button dalam ruang kerja Anda. Pilih Radio Button dari dalam ToolBox.



**Radio Button**

- Ambil dan tambahkan dua buah Radio Button ke dalam ruang kerja, kira-kira di bawah TextBlock yang sudah ada.

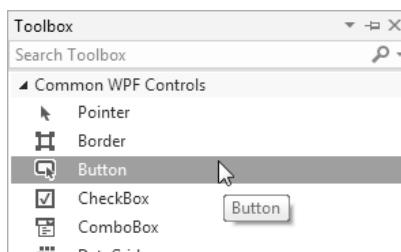


**Gambar 1-14. Radio Button dalam designer view**

- Dari window **Properties**, tambahkanlah nama untuk kedua kontrol ini, dengan cara mengubah isi dari **Name** (bagian paling atas dalam window **Properties**) menjadi `RadioButton1` dan `RadioButton2`.
- Pilih Radio Button yang sebelah kiri, kemudian klik tombol kanan mouse, pilih opsi untuk **Edit Text**, dan ganti tulisan RadioButton menjadi `Super`.
- Demikian juga yang sebelah kanan diganti menjadi `Fantastik`.

## Menambahkan Button

Tahap berikutnya adalah menambahkan sebuah Button.



**Memilih Button**

Tambahkan Button tersebut di tengah-tengah layar. Pilih **Button** tersebut, dan dari tampilan XAML di bawah, temukan teks `Content="Button"` dan gantilah menjadi `Content="Display"`.

Simpan design ini. Tampilan akhir akan menjadi seperti berikut.



*Tampilan akhir yang didapat*

## Menambahkan Program pada Button Display

Pada saat aplikasi ini berjalan akan terbuka sebuah window dengan tampilan seperti dalam designer view tadi. Seorang user akan memilih salah satu dari dua **Radio Button** yang ada dan kemudian menekan tombol **Display**. Dari pilihan user tadi, akan muncul sebuah **message box** dengan isi yang berbeda.

Untuk menciptakan kondisi seperti itu, Anda perlu menambahkannya dalam aplikasi ini.

Dalam designer view, double-click Button Display itu. Proses ini akan membuka dokumen Hello.xaml.vb ke main workspace. Perhatikan baris coding di bawah ini.

```
Private Sub Button_Click(sender As Object, e As  
RoutedEventArgs)
```

```
End Sub
```

Kemudian, tuliskan kondisi berikut di antara kedua baris tersebut:

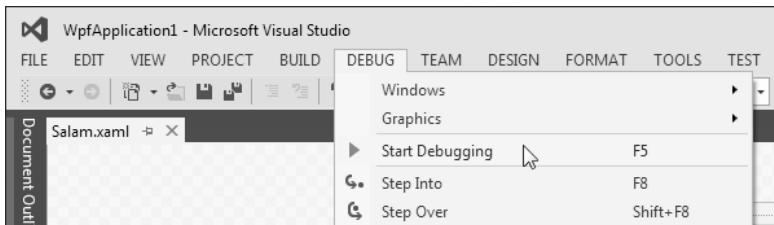
```
If RadioButton1.IsChecked = True Then  
    MessageBox.Show("Halo, manusia Super.")  
Else : RadioButton2.IsChecked = True  
    MessageBox.Show("Halo, manusia Fantastik.")  
End If
```

Simpan aplikasi ini, dan Anda siap untuk melakukan testing aplikasi Anda.

## Debugging

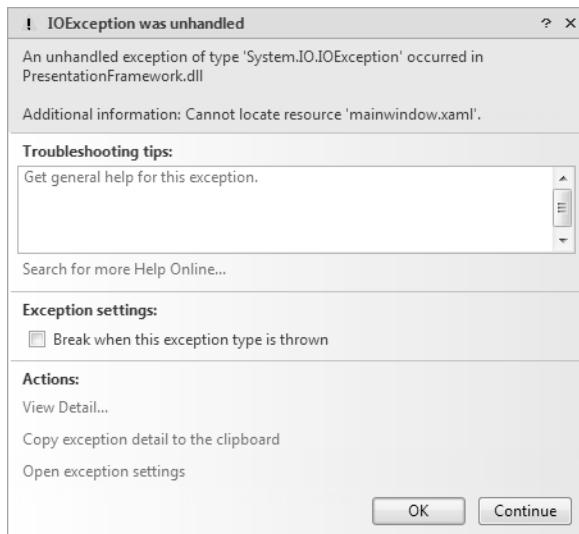
Ingat pada saat mengganti mengganti **Title** pada window utama, ada peringatan tentang kesalahan? Kita akan mencari kesalahan tersebut dalam tahap berikut ini.

Mulailah dengan menjalankan proses debugging, dengan cara memilih dari sub-menu **Debug** dan opsi **Start Debugging**.



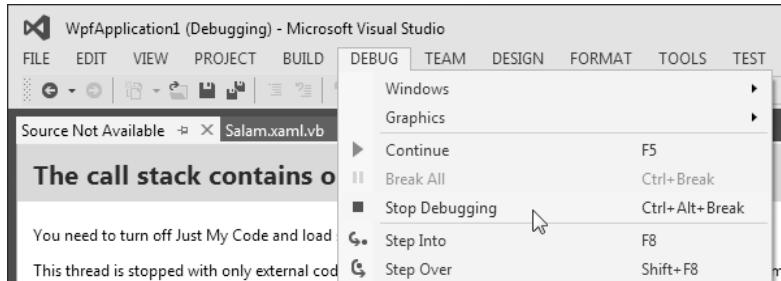
*Memilih opsi Start Debugging*

Segera setelah itu, proses akan berjalan, dan di tengah jalan akan muncul sebuah dialog box, yang berisi keterangan bahwa sebuah kesalahan telah terjadi, bahwa sistem Cannot locate resource 'mainwindow.xaml'.



### ***IOException was unhandled***

Klik pada tombol **OK** dan hentikan debugging dengan memilih **Stop Debugging** pada menu atau pada toolbar.

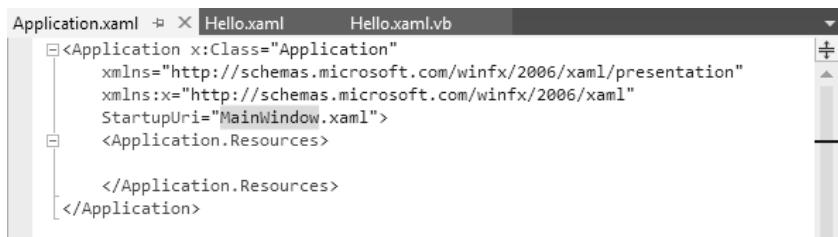


### ***Stop Debugging***

Di depan tadi Anda mengganti nama window MainWindow.xaml menjadi Hello.xaml. Namun, project itu sendiri masih mengacu pada halaman pembuka MainWindow.xaml. Karena itu, pada saat dijalankan pesan kesalahan akan terjadi.

Karena itu, Anda juga harus menginstruksikan pada project itu sendiri bahwa halaman pembuka sekarang sudah diganti menjadi Hello.xaml.

Dari Solution Explorer, bukalah berkas Application.xaml. Carilah teks yang menunjuk pada MainWindow.xaml, dan gantilah menjadi Hello.xaml.



```
Application.xaml  X  Hello.xaml    Hello.xaml.vb
```

```
<Application x:Class="Application"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    StartupUri="MainWindow.xaml">  
    <Application.Resources>  
        </Application.Resources>  
</Application>
```

***Tampilan panel Application.xaml***

Simpan perubahan ini, dan aplikasi akan berjalan dengan benar.



***Tampilan aplikasi***

## Debugging dengan Breakpoints

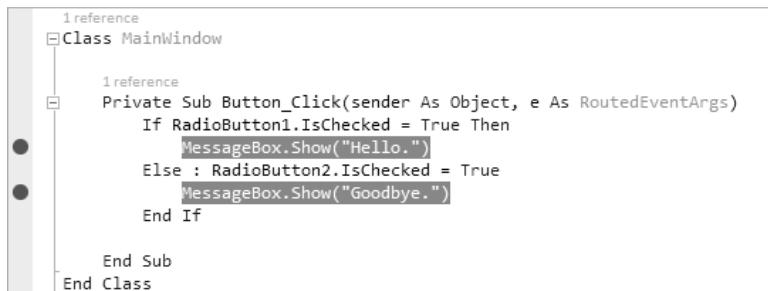
Dengan menambahkan beberapa breakpoints, Anda dapat memastikan bahwa kondisi yang Anda tuliskan berjalan dengan baik.

Bukalah berkas Hello.xaml.vb dari dalam **Solution Explorer**. Letakkan kursor pada baris yang berisi teks di bawah ini:

```
MessageBox.Show("Halo, manusia Super.")
```

Kemudian, tambahkan breakpoint pada baris itu dengan membuka dari sub-menu **Debug, Toggle Breakpoint**.

Lakukan hal yang sama untuk baris **MessageBox** yang lain.

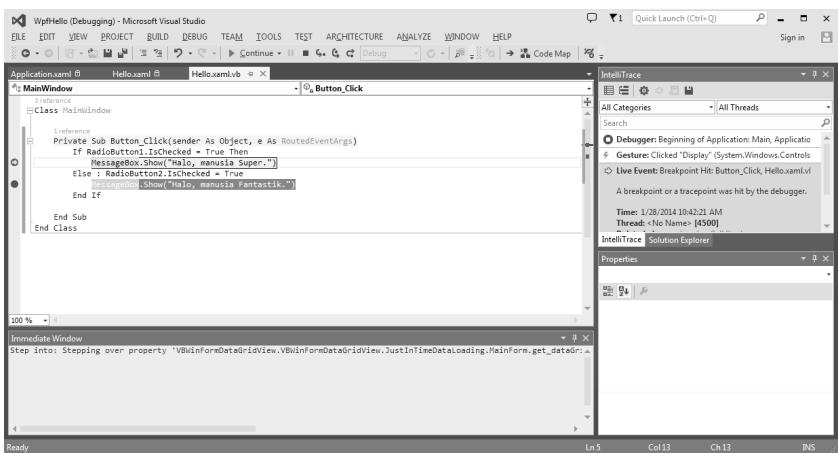


### Hasil Toggle Breakpoint

Anda akan lihat bahwa baris di mana dilakukan **Toggle Breakpoint** akan mendapatkan highlight merah, dan pada kolom sebelah kiri ada sebuah tanda lingkaran berwarna merah. Anda juga dapat menambahkan breakpoint pada baris manapun dengan cara klik pada kolom sebelah kiri tersebut.

Jalankan lagi debugging dengan cara menekan tombol **Start Debugging**. Kali ini, window yang tadi Anda design akan berhasil dimunculkan. Pilihlah salah satu dari kedua **Radio Button**, dan klik pada tombol **Display** di bawah.

Setelah itu, kontrol aplikasi akan berpindah pada tampilan *debugger* dan berhenti pada baris **MessageBox** yang sesuai dengan pilihan **Radio Button** Anda sebelumnya.



### Proses berhenti pada breakpoint

Tampilan utama layar Anda akan tampak sedikit berbeda dari tampilan sebelumnya. Inilah tampilan yang akan didapat pada saat melakukan debugging. Sepenuhnya mengenai tampilan debugging ini akan dibahas secara tersendiri pada bab-bab selanjutnya.

Dari menu **Debug**, pilih **Step Out** untuk melanjutkan proses. Aplikasi akan melanjutkan dengan membuka sebuah **message box** dengan pesan "Hallo, manusia Fantastik.". Klik pada tombol **OK**, dan pada window **Hello**, pilihlah opsi **Super**.

Sekali lagi, *debugger* akan berhenti pada baris **MessageBox**, dan kali ini pada baris yang pertama. Di sini Anda tahu bahwa aplikasi yang Anda buat ini sudah berjalan dengan baik. Dari menu **Debug**, pilih opsi untuk **Disable All Breakpoints**. Perintah ini akan menonaktifkan semua breakpoint yang sudah dibuat.

Kemudian, pilih tombol **Continue** pada toolbar atau dari sub-menu **Debug, Continue**. Di sini, Anda dapat meneruskan menjalankan aplikasi ini, atau Anda dapat menutupnya untuk menyelesaiakannya.

# Bab2

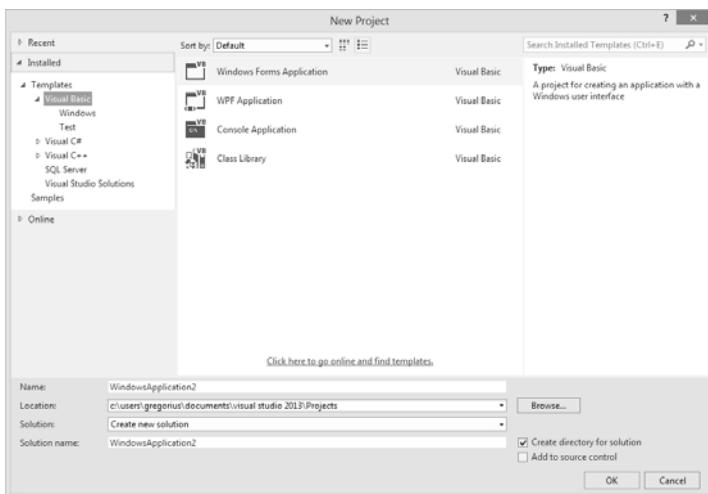
## Latihan Membuat Aplikasi-Aplikasi Sederhana

Kini Anda sudah siap mempelajari pemrograman Visual Basic 2013. Dengan semakin bertambahnya pengetahuan di bidang pemrograman ini, pada dasarnya Anda bisa menciptakan aplikasi-aplikasi apa pun.

Nah, sebelum kita masuk ke dalam teknik pemrograman yang lebih dalam, ada baiknya kita ciptakan dulu aplikasi paling sederhana yang bisa dibuat menggunakan Visual Basic 2013.

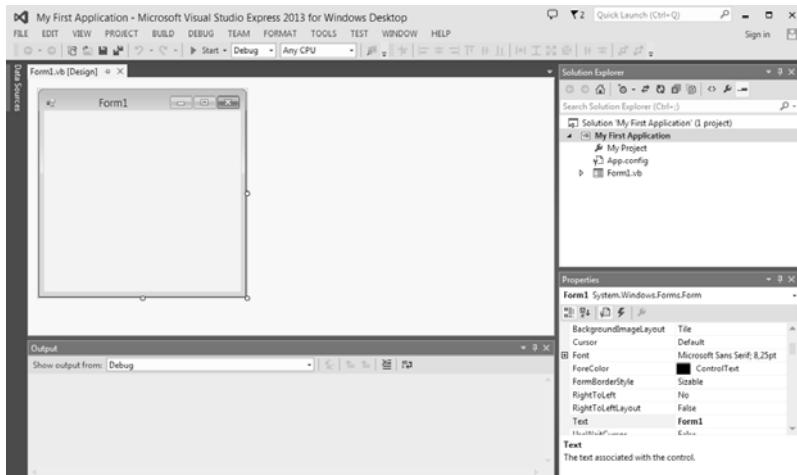
Lakukanlah langkah-langkah sebagai berikut:

1. Buatlah proyek baru dengan mengklik menu **File > New Project**.
2. Karena kita ingin membuat aplikasi yang nanti akan berbentuk window, maka pilih **Windows Forms Application**.



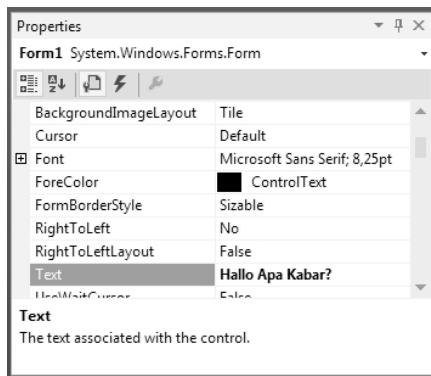
### **Membuat aplikasi berbasis window baru dengan mengklik Windows Forms Application**

3. Di bagian **Name**, gantilah nama yang ada di situ dengan nama aplikasi Anda sendiri. Misalnya: **My First Application**.
4. Tekan tombol **OK**. Tunggu beberapa saat dan Anda akan melihat jendela aplikasi kosong seperti gambar di bawah ini.



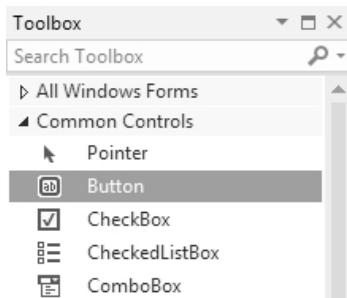
**Jendela aplikasi kosong yang siap untuk diberi script**

5. Di bagian kanan, Anda akan melihat panel **Properties**. Klik di bagian **Text** dan ubah dari **Form1** menjadi teks pilihan Anda sendiri, misalnya: **Hallo Apa Kabar?**.



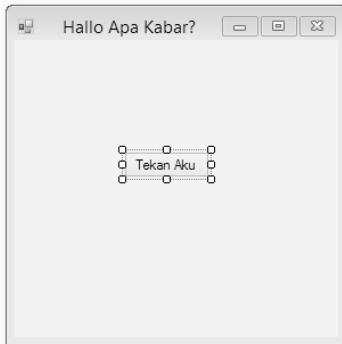
*Pengaturan teks yang nanti akan muncul  
di dalam jendela aplikasi*

6. Sekarang, kita buat sebuah tombol. Klik menu **View > Toolbox**.
7. Klik **Button** dan drag ke dalam jendela aplikasi yang ingin kita buat.



*Memilih Button untuk menghasilkan sebuah tombol  
di dalam jendela aplikasi*

8. Posisikan tombol itu di tengah-tengah jendela aplikasi dan di dalam panel **Properties**, pada opsi **Text**, ganti menjadi: **Tekan Aku**.



*Tombol yang telah berganti tulisan*

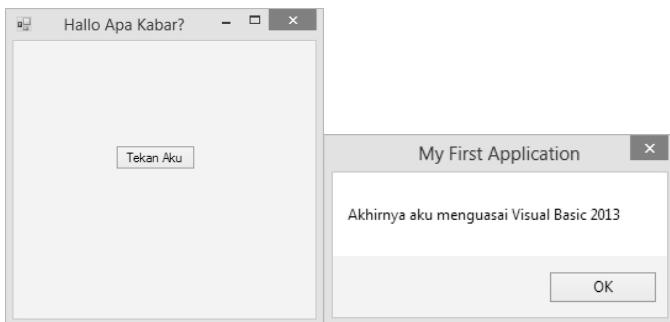
9. Kita akan “menghidupkan” tombol itu dengan memasang script. Klik-ganda pada tombol sampai muncul panel bernama: **Form1.vb**.
10. Lantas, lengkapi script yang sudah tertulis di dalam panel tersebut menjadi script berikut:

```
Public Class Form1  
  
    Private Sub Button1_Click(sender As Object, e As EventArgs)  
        Handles Button1.Click  
  
        MsgBox("Akhirnya aku menguasai Visual Basic 2013")  
  
    End Sub  
  
End Class
```



*Script yang telah lengkap*

- Untuk menguji aplikasi yang telah disisipi script itu, klik ikon **Start** untuk menjalankan aplikasi tersebut.
- Nanti Anda akan melihat jendela aplikasi seperti gambar di bawah ini yang jika tombol di tengah-tengahnya diklik, maka akan memicu munculnya kotak dialog seperti gambar kanan-bawah berikut.



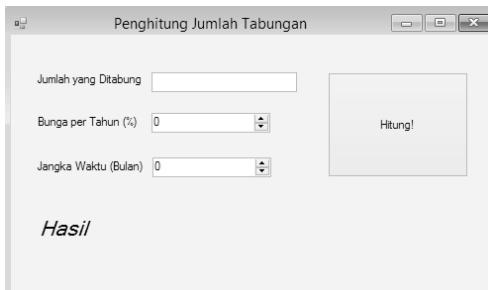
*Aplikasi sederhana yang bisa kita ciptakan*

## Aplikasi Penghitung Jumlah Tabungan

Jika Anda menabung Rp. 10.000.000,- dengan bunga per tahun sebesar 5 %, maka berapa jumlah tabungan tersebut jika disimpan selama 5 tahun (60 bulan)? Ingin tahu jawabannya? Buatlah aplikasi di bawah ini:

- Buatlah project baru dan untuk Form, namailah: **Penghitung Jumlah Tabungan**.
- Buat empat (4) buah label, yaitu:
  - Label1.** Text: Jumlah yang Ditabung
  - Label2.** Text: Bunga per Tahun(%)
  - Label3.** Text: Jangka Waktu (Bulan)
  - Label4.** Text: Hasil
- Buat satu textbox dan letakkan di samping kanan label "Jumlah yang Ditabung".

- Buat dua buah kontrol **NumericUpDown** dan letakkan masing-masing di samping kanan label “Bunga per Tahun (%)” dan “Jangka Waktu (Bulan)”.
- Terakhir, buat button dan letakkan di sisi kosong pada form. Ganti tulisan yang terlihat di atas tombol itu menjadi: **Hitung!**



### **Aplikasi untuk menghitung jumlah tabungan**

- Klik-gAnda pada tombol **Hitung!** Dan tulis script di bawah ini:

```

PrivateSub Button1_Click(sender AsObject, e AsEventArgs)
Handles Button1.Click

Dim Bunga, Tabungan, JumlahTabungan AsDecimal

Dim JangkaWaktu AsInteger

Tabungan = TextBox1.Text

Bunga = NumericUpDown1.Value / 100

JangkaWaktu = NumericUpDown2.Value

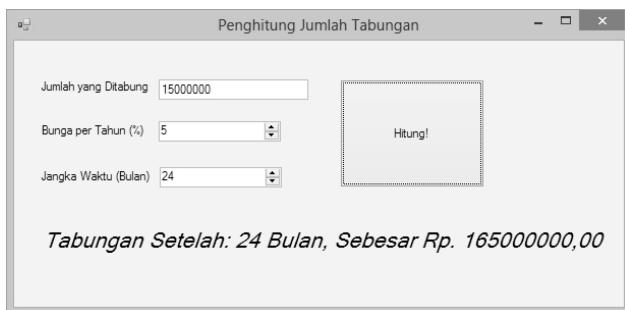
JumlahTabungan = Tabungan + (Bunga * Tabungan *
JangkaWaktu) / 12 * 100

Label4.Text = "Tabungan Setelah: "& JangkaWaktu &" Bulan,
Sebesar Rp. "& JumlahTabungan

EndSub

```

Luncurkan aplikasi di atas dengan menekan tombol **F5** dan masukkan data-data seperti yang diminta. Setelah itu, tekan tombol **Hitung!** Dan akan didapat hasil penghitungan seperti gambar berikut.



***Aplikasi untuk menghitung jumlah tabungan setelah beberapa bulan ke depan***

## **Bagaimana Script di Atas Bekerja?**

Rumus untuk mengetahui jumlah tabungan setelah beberapa bulan ke depan dengan bunga tertentu, maka gunakan rumus berikut:

```
JumlahTabungan = Tabungan + (Bunga * Tabungan * JangkaWaktu) /  
12 * 100
```

Variable Tabungan berisi data dari textBox1. Variable Bunga dari NumericUpDown1.Value, dan Variable JangkaWaktu dari NumericUpDown2.Value. Hasilnya akan ditampilkan di Label4.

## **Aplikasi untuk Menghitung Langganan PDAM**

Kalau Anda berlangganan air PDAM, terutama kantor-kantor di tengah kota, maka Anda bisa menghitung tarif langganan PDAM. Penghitungan tarif PDAM menggunakan metode progresif di mana konsumsi air yang lebih banyak akan memicu biaya yang lebih besar. Berikut contoh pengenaan tarif progresif untuk berlangganan PDAM:

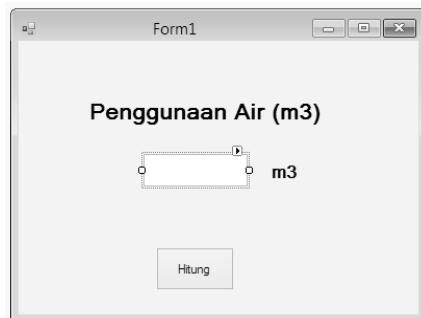
- Penggunaan 0 – 10 m<sup>3</sup> dikenakan tarif Rp. 1.852 per m<sup>3</sup>.
- Penggunaan 11 – 20 m<sup>3</sup> dikenakan tarif Rp. 2.645 per m<sup>3</sup>.
- Penggunaan 21 – 30 m<sup>3</sup> dikenakan tarif Rp. 3.439 per m<sup>3</sup>.
- Penggunaan di atas 30 m<sup>3</sup> dikenakan tarif Rp. 4.232 per m<sup>3</sup>.

Jadi, untuk penggunaan 36 m<sup>3</sup>, maka biaya yang harus dikeluarkan sebesar Rp. 104.752,-. Berikut rinciannya:

- 10 m<sup>3</sup> x Rp. 1.852 = Rp. 18.520,-
- 10 m<sup>3</sup> x Rp. 2.645 = Rp. 26.450,-
- 10 m<sup>3</sup> x Rp. 3.439 = Rp. 34.390,-
- 6 m<sup>3</sup> x Rp. 4.232 = Rp. 25.392,-

Berikut pembuatan aplikasi untuk menghitung tarif progresif PDAM:

1. Buat project baru dari awal.
2. Lantas, pasanglah dua buah label, yaitu:
  - Label1. Text: Penggunaan Air (m3)
  - Label2. Text: m<sup>3</sup>
3. Buatlah textbox untuk menginput kebutuhan air yang akan dihitung tarifnya.
4. Terakhir, buat tombol dengan tulisan **Hitung**.
5. Klik-ganda tombol itu dan isi dengan script sebagai berikut.



**Rancangan form untuk menghitung biaya langganan PDAM**

```
PublicClassForm1  
  
Function TarifPDAM(Nilai)  
  
Dim Progresif AsInteger
```

```

SelectCase Nilai

Case 1 To 10

Nilai = Nilai * 1852

Progresif = Nilai + 4000 + 4000 + 5000

Case 11 To 20

Nilai = 18520 + (Nilai * 2645)

Progresif = Nilai + 4000 + 4000 + 5000

Case 21 To 30

Nilai = 44970 + (Nilai * 3439)

Progresif = Nilai + 4000 + 4000 + 5000

CaseIs>= 31

Nilai = 79360 + ((Nilai * 30) * 4232)

Progresif = Nilai + 4000 + 4000 + 5000

EndSelect

TarifPDAM = Progresif

EndFunction

```

```

PrivateSub Button1_Click(sender AsObject, e As EventArgs)
Handles Button1.Click

Dim Air AsInteger

Dim RumusPDAM AsDecimal

Air = TextBox1.Text

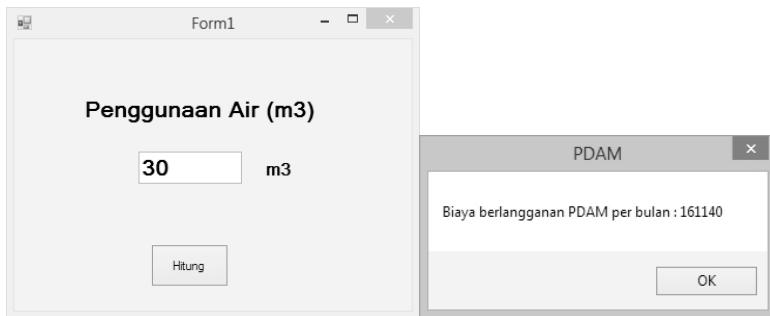
RumusPDAM = TarifPDAM(Air)

MsgBox("Biaya berlangganan PDAM per bulan : "& RumusPDAM)

EndSub

EndClass

```



*Aplikasi yang telah digunakan untuk menghitung  
biaya langganan PDAM*

## Bagaimana Script di Atas Bekerja?

Inti dari hitung-hitungan tarif progresif terdapat pada pernyataan Select Case berikut ini:

```
SelectCase Nilai
Case 1 To 10
Nilai = Nilai * 1852
Progresif = Nilai + 4000 + 4000 + 5000
Case 11 To 20
Nilai = 18520 + (Nilai * 2645)
Progresif = Nilai + 4000 + 4000 + 5000
Case 21 To 30
Nilai = 44970 + (Nilai * 3439)
Progresif = Nilai + 4000 + 4000 + 5000
Case Is >= 31
Nilai = 79360 + ((Nilai * 30) * 4232)
Progresif = Nilai + 4000 + 4000 + 5000
EndSelect
```

Lewat Select Case, script akan mendeteksi berapa angka yang dimasukkan ke dalam argument “Nilai” (nanti argument ini akan terisi angka dari textbox1.text). Jika argument berisi angka 1 sampai 10, maka script yang ada di bawah Case 1 To 10 akan dijalankan. Begitu seterusnya.

Jadi, jika angka di dalam argument “Nilai” terisi angka 1 sampai 10, maka angka tersebut akan dikalikan tarif  $m^3$  untuk tarif progresif tingkat pertama, yaitu Rp. 1.852,-. Selanjutnya, hasil Penghitungan di atas ditampilkan menggunakan pernyataan sebagai berikut:

$$\text{Progresif} = \text{Nilai} + 4000 + 4000 + 5000$$

Apabila, katakanlah, argument “Nilai” berisi angka 15, maka akan berlaku:

$$\text{Nilai} = 18520 + (\text{Nilai} * 2645)$$

Maksudnya, tarif progresif tingkat pertama sebesar Rp. 18.520,- (hasil Penghitungan dari angka 10  $m^3$  x Rp. 1.852,-) ditambahkan dengan argument “Nilai” dikalikan 10 juga. Nanti, penjumlahan antara Rp. 18.520 + Rp. 26.450 yang sebesar Rp. 44.970 ditulis dalam tarif progresif berikutnya.



# Bab 3

## Mengenal Database

Database adalah sebuah system yang digunakan untuk menyimpan data-data atau biasa disebut informasi terstruktur. Selanjutnya, informasi atau data-data tadi dapat "dipanggil" kembali bilamana dibutuhkan.

Dalam database, informasi atau data tadi disimpan dalam sebuah tempat yang biasa disebut tabel.

Setiap table menyimpan data yang berbeda sesuai kebutuhan penciptanya. Misalnya, ada tabel untuk menyimpan jumlah stok, nama supplier, alamat klien, dan sebagainya. Mengapa data-data itu disimpan ke dalam tabel? Agar data tersebut bisa dipecah dalam potongan kecil sehingga selain lebih hemat resource, pemecahan data ke dalam tabel akan memudahkan pengorganisasian terhadap data itu sendiri.

Tabel-tabel dalam database kemungkinan besar akan saling berinteraksi. Sebagai contoh, tabel untuk menyimpan nama dan stok produk bisa jadi berinteraksi dengan tabel nama-nama supplier sehingga ketika stok menipis, supplier mendapatkan informasi secara cepat. Interaksi antar tabel ini biasa disebut dengan *relationship*/hubungan antar table.

Tabel dan relationship tersebut dikelola oleh suatu program yang dikenal dengan sebutan *database management system (DBMS)*. DBMS ini akan mengatur semua informasi di dalam database menggunakan berbagai statement yang ditulis dalam bahasa pemrograman bernama *Structured Query Language (SQL)*.

Umumnya, bahasa pemrograman SQL tidak lepas kaitannya dengan SQL Server. SQL Server sendiri adalah sebuah database management system. Sebuah database SQL Server adalah sebuah database yang dikelola oleh SQL Server. Dalam praktiknya, bahasa SQL digunakan untuk mengambil record-record tertentu dari dalam *database* dan mengembalikannya pada user. Selain itu, bahasa tersebut juga bisa digunakan untuk manipulasi *record*, seperti penambahan, update, maupun penghapusan informasi dari dalam database atau lebih umum disebut dengan istilah CRUD (Create, Read, Update, dan Delete).

## SQL Server Management Studio

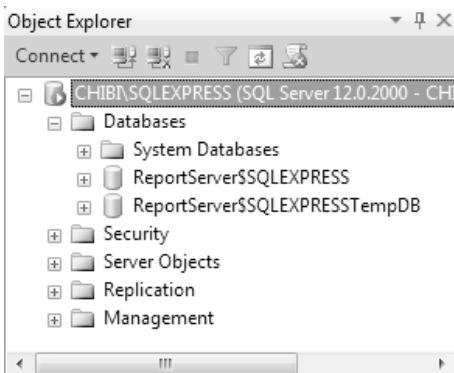
**SQL Server Management Studio (SSMS)** adalah sebuah lingkungan ter-integrasi untuk mendapat akses, konfigurasi, manajemen, administrasi, maupun mengelola seluruh komponen dalam SQL Server. Untuk menjalankan SSMS dilakukan sebagai berikut:

1. Jalankan **SSMS** dari **Start Menu** sampai terlihat jendela **Connect to Server**.
2. Masukkan akun menggunakan account **Windows** atau akun dari **account system administrator** pada **SQL Server**. Kedua pilihan tersebut ditunjukkan dalam gambar berikut ini.



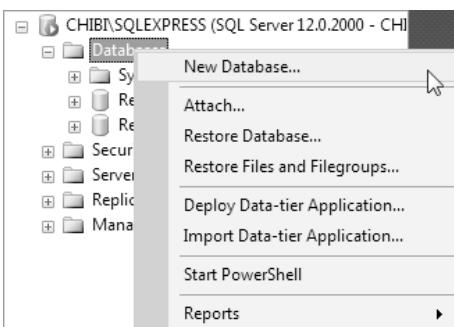
**Pilihan cara autentikasi**

- Apabila Anda memilih metode **SQL Server Authentication**, maka masukkan password untuk *account* Anda, dan tekan tombol **Connect**. Apabila Anda menggunakan cara **Windows Authentication**, maka bisa langsung menekan tombol **Connect** tersebut.
- Setelah Anda berhasil masuk, akan terlihat *tool window Object Explorer* di sebelah kiri *workspace* Anda. Di sana akan ditunjukkan semua bagian dari instalasi **SQL Server** Anda. Namun, perhatikan pada item **Database**. Lakukan klik pada ikon tanda panah di sebelah kata **Database** untuk membukanya.



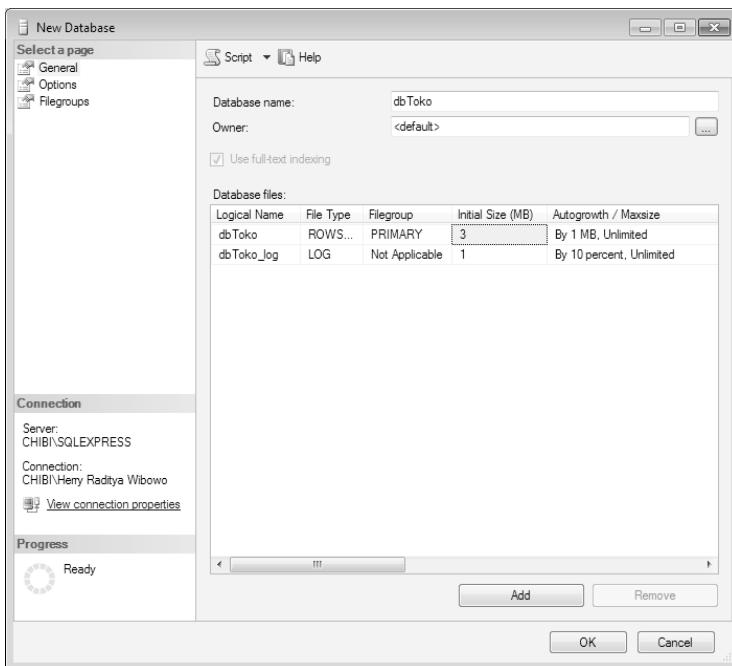
**Object Explorer**

- Buatlah database baru dengan memilih **Database**, dan klik-kanan.



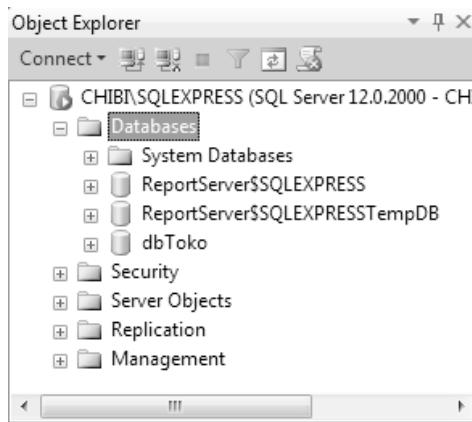
**New Database**

6. Ada beberapa pilihan di sana, namun kali ini, Anda akan membuat sebuah *database*. Karena itu, pilihlah **New Database**.



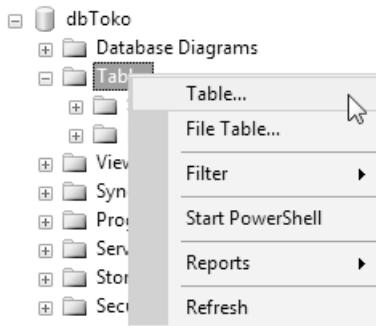
### ***Konfigurasi New Database***

7. Pakailah nilai default untuk pengaturan di atas.
8. Untuk nama database(*Database name*), buatlah nama baru dan pakailah nama yang cukup sederhana dan mencerminkan apa yang akan Anda simpan di dalamnya. Misalnya, nama **dbToko**.
9. Perhatikan bahwa setelah Anda mengganti nama menjadi **dbToko**, maka untuk *Database files* di bawahnya pun akan ikut berganti nama sesuai dengan nama *database*.
10. Tekanlah tombol **OK** untuk melanjutkan. Proses pembuatan *database* akan dijalankan. Beberapa saat kemudian, *database* baru Anda akan nampak pada **Object Explorer**.



***dbToko pada Object Explorer***

11. Anda bisa melihat, bahwa *database* baru Anda ada di sana. Bukalah *database* **dbToko** tersebut untuk melihat saja apa yang ada di dalamnya. Setelah itu, kita akan membuat tabel. Klik-kanan, dan pilihlah untuk membuat sebuah *table* baru dengan perintah **Table**.



***Membuat table baru***

12. Pada tampilan baru yang Anda dapatkan, perhatikan window *Properties* di sebelah kanan *workspace* Anda, khususnya pada bagian **(Name)**. Di sini, Anda diperkenankan untuk membuat sebuah nama untuk *table*. Gunakan nama yang sesuai dengan peruntukannya, misalnya **Barang**.

13. Biarkan opsi lain berada di nilai *default* pada window **Properties** itu, dan sekarang beralihlah pada *workspace* utama Anda.
14. Anda akan diberikan suatu lembar semacam *spreadsheet*, namun hanya terdapat tiga buah kolom, yaitu:

- **Column Name**

Di sini Anda tuliskan nama dari field. Misalnya, jika sebuah *table* digunakan untuk menyimpan informasi kontak, maka mungkin Anda akan tuliskan *Column Name* dengan nama-nama seperti **NamaDepan**, **NamaKeluarga**, **TglLahir**, **Alamat**, **NoTelp**, dan **Email** untuk menyimpan data-data dalam *database*.

- **Data Type**

Kolom ini menentukan tipe dari informasi yang akan Anda simpan dalam setiap kolom. Misalnya, **NamaDepan** seseorang adalah berupa suatu rentetan huruf, maka Anda mungkin akan menggunakan *varchar(30)*. Demikian juga untuk **TglLahir**, Anda boleh menggunakan tipe *date*.

- **Allow Nulls.**

Kolom ini hanya berupa *checkbox*. Jika Anda mengaktifkannya, *database* akan mengizinkan Anda untuk menyimpan nilai kosong (*null*) untuk kolom tersebut.

Selesaikanlah pengisian atribut untuk table pertama Anda sesuai dengan gambar berikut.

CHIBI\SQLEXPRESS....Toko - dbo.Table_1*		
Column Name	Data Type	Allow Nulls
Brg_id	nchar(10)	<input type="checkbox"/>
Brg_nama	nchar(35)	<input type="checkbox"/>
Brg_unit	nchar(5)	<input type="checkbox"/>
Brg_barcode	nchar(8)	<input type="checkbox"/>
Brg_hpp	int	<input type="checkbox"/>
► Kat_id	tinyint	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

**Atribut untuk table Barang**

Pada akhirnya, buatlah ketujuh *table* seperti detail di bawah ini.

Column Name	Data Type	Allow Nulls
Cust_id	smallint	<input type="checkbox"/>
Cust_nama	nchar(25)	<input type="checkbox"/>
Cust_email	nchar(25)	<input type="checkbox"/>
Cust_alamat	nchar(50)	<input type="checkbox"/>
Cust_kota	nchar(20)	<input type="checkbox"/>
Cust_phone	nchar(15)	<input type="checkbox"/>
Cust_reg_on	smalldatetime	<input type="checkbox"/>

**Table Users**

Column Name	Data Type	Allow Nulls
User_id	smallint	<input type="checkbox"/>
User_nama	nchar(8)	<input type="checkbox"/>
User_priv	tinyint	<input type="checkbox"/>
User_pwd	nchar(8)	<input type="checkbox"/>

**Table Kategori**

Column Name	Data Type	Allow Nulls
Kat_id	tinyint	<input type="checkbox"/>
Kat_nama	nchar(10)	<input type="checkbox"/>

**Table Jual**

Column Name	Data Type	Allow Nulls
Jual_id	nchar(10)	<input type="checkbox"/>
Jual_tgl	smalldatetime	<input type="checkbox"/>
Cust_id	smallint	<input type="checkbox"/>
Jual_note	nchar(30)	<input checked="" type="checkbox"/>
User_id	smallint	<input type="checkbox"/>

**Table Jual\_details**

Column Name	Data Type	Allow Nulls
Jual_det_id	smallint	<input type="checkbox"/>
Jual_id	nchar(10)	<input type="checkbox"/>
Brg_id	nchar(10)	<input type="checkbox"/>
Jual_det_qty	tinyint	<input type="checkbox"/>
Jual_det_harga	int	<input type="checkbox"/>
		<input type="checkbox"/>

**Table Barang**

Column Name	Data Type	Allow Nulls
Brg_id	nchar(10)	<input type="checkbox"/>
Brg_nama	nchar(35)	<input type="checkbox"/>
Brg_unit	nchar(5)	<input type="checkbox"/>
Brg_barcode	nchar(8)	<input type="checkbox"/>
Brg_hpp	int	<input type="checkbox"/>
Kat_id	tinyint	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
Brg_id	nchar(10)	<input type="checkbox"/>
Brg_det_stock	tinyint	<input type="checkbox"/>
		<input type="checkbox"/>

**Table Barang\_details**

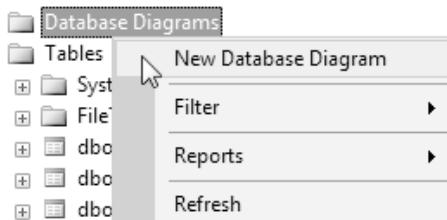
Terdapat ikon berbentuk kunci di salah satu field.Ikon itu menunjukkan bahwa *field* di mana tanda anak kunci tersebut berada merupakan Primary Key. Artinya, nilai yang ada pada *field* tersebut haruslah unik untuk keseluruhan *table*. Tidak boleh ada nilai yang kosong, dan juga tidak boleh ada nilai yang sama.

Paling tidak ada dua buah fungsi Primary Key, yaitu:

- Primary Key, umumnya akan sangat membantu dalam hal pencarian data, khususnya pada saat jumlah data dalam table tersebut terhitung banyak.
- Primary Key juga digunakan untuk hubungan antar-table. Antar-table dapat saling berhubungan (seperti contoh pada awal bab, hubungan antara table **Jual** dan table **Customer**), selama pada kedua table tersebut ada sebuah *field* yang sama.

Selanjutnya, lanjutkan ke langkah-langkah berikut:

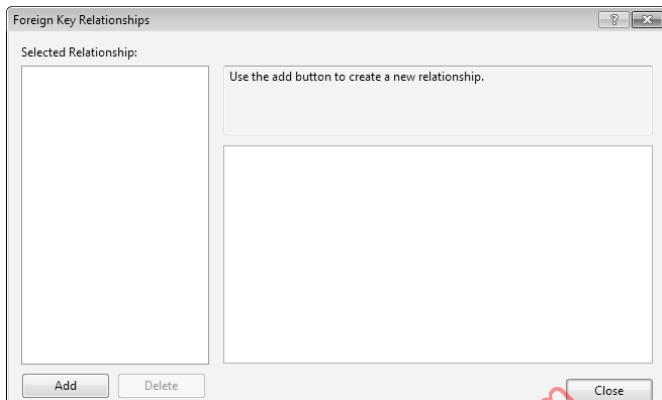
1. Kita kemudian akan menetapkan hubungan antar *table*. Untuk mempermudah penetapan ini, Anda bisa membuat sebuah **Database Diagram**.
2. Temukan *folder Database Diagram* pada window **Object Explorer**, lakukan *klik-kanan* pada mouse Anda, dan pilihlah **New Database Diagram**.



**New Database Diagram**

3. Dalam sekejap akan terbuka sebuah *workspace* baru, dengan sebuah **Add Table dialog box** ditampilkan. Pilihlah semua *table* Anda, dan tekan tombol **Add**. Kemudian tekan tombol **Close**.
4. Nanti akan tersusun sebuah diagram dasar dengan semua *table* Anda di dalamnya. Aturlah penempatannya seperti yang diinginkan, dan atur juga ukuran display, supaya lebih mudah untuk dilihat.
5. Yang paling mudah untuk dibuat hubungannya adalah *table-table* yang berupa *parent-child*, yaitu seperti **Barang** dengan **Barang\_details**, dan **Jual** dengan **Jual\_details**.

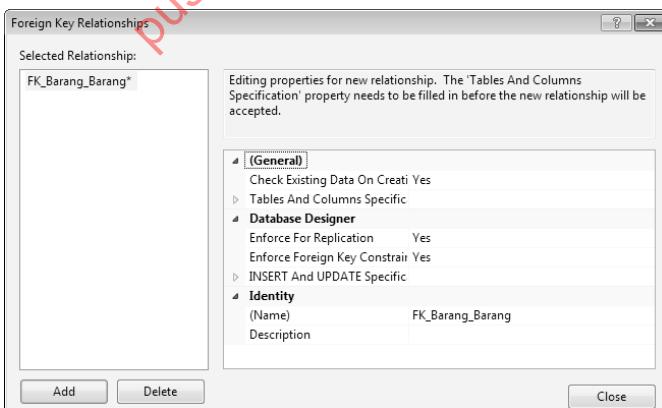
6. Pilihlah *table Barang*, tekan tombol mouse sebelah kanan, dan pilih opsi **Relationships**.



### **Foreign Key Relationships**

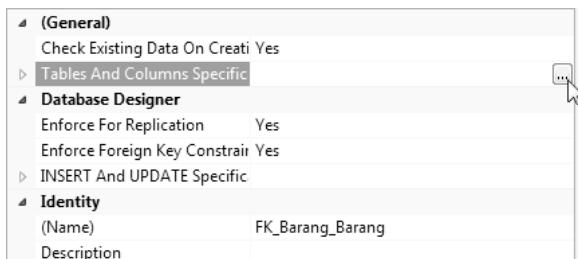
Apa itu **Foreign Key**? Istilah ini merujuk pada suatu *field* dalam sebuah *table* yang sama dengan **Primary Key** pada *table* lain. Jadi, hubungan antar *table* adalah sama dengan hubungan antara **Primary Key** suatu *table* dengan **Foreign Key** pada *table* lain.

7. Saat ini, *relationships* masih dalam keadaan kosong, karena memang belum dibuat. Tekan tombol **Add** untuk menambahkan suatu *relationship* baru.



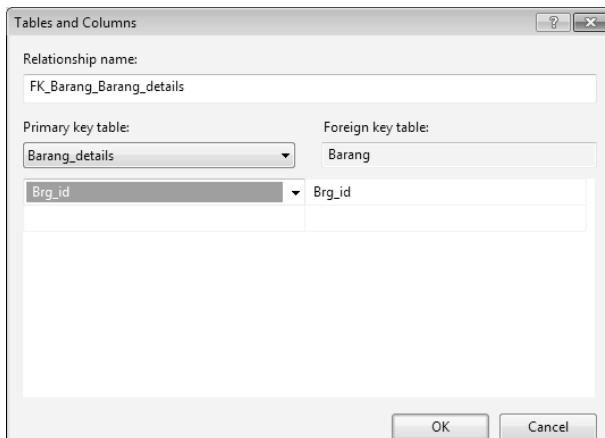
### **Penambahan suatu relationship**

- Secara *default*, system akan membuat sebuah *relationship* yang mengacu pada dirinya sendiri. Seperti yang Anda bisa lihat pada bagian sebelah kiri, kunci *relationship* tersebut diberi nama sementara, yaitu **FK\_Barang\_Barang**, yang artinya sebuah **Foreign Key** untuk *table Barang* dengan *table Barang* juga. Namun di sini Anda hendak membuat sebuah *relationship* antara *table Barang* dengan *table Barang\_details*.
- Perhatikan pada bagian sebelah kanan. Pilihlah baris pada bagian **General** yang bertuliskan **Tables and Columns Specific**. Setelah Anda pilih baris tersebut, pada bagian sebelah kanannya akan ditampilkan sebuah tombol kecil dengan isi tiga buah titik.



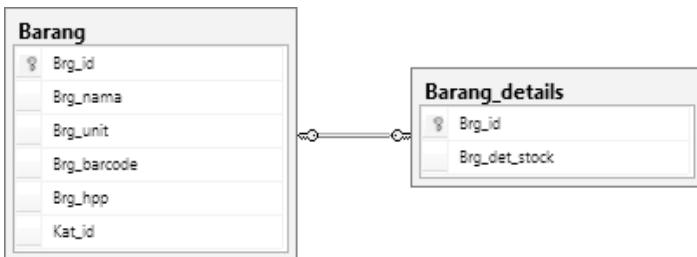
#### **Tables and Columns Specific**

- Tekan tombol tersebut. Anda akan mendapatkan sebuah *form* baru.



**Hubungan antara table Barang dengan Barang\_details**

- Dalam *form* ini hanya bagian sebelah kiri yang bisa Anda ubah nama *table*-nya. Untuk kunci *relationship* Anda yang pertama, gantilah menjadi *table* **Barang\_details**, dan kemudian ubah juga kunci *field* yang digunakan untuk mengacu pada *field* **Brg\_id**.
- Tekan **OK**. Sebagai tanda bahwa *relationship* ini sudah terbentuk, pada diagram akan ditampilkan sebuah garis penghubung antara *table* **Barang** dengan *table* **Barang\_details**.

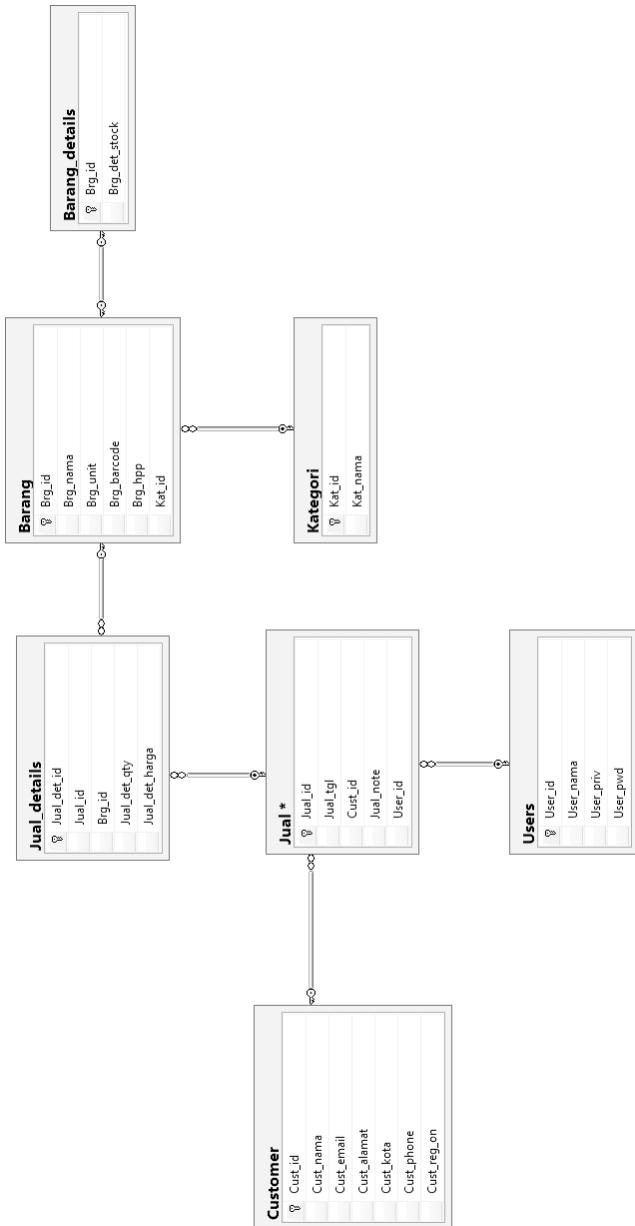


*Relationship yang pertama*

Kalau sudah begini, Anda bisa membuat semua *relationship* yang ada, misalnya dengan memanfaatkan acuan daftar berikut:

- Barang – Barang\_details, Brg\_id
- Barang – Kategori, Kat\_id
- Barang – Jual\_details, Brg\_id
- Jual – Jual\_details, Jual\_id
- Jual – Customer, Cust\_id
- Jual – Users, User\_id

Setelah semuanya dibuat, maka gambar pada diagram akan tampak sebagai berikut.



***Relationship secara penuh***

Seperti bisa dilihat pada gambar, Anda bisa menetapkan lebih dari satu hubungan untuk sebuah *table*. Simpanlah diagram tersebut, dan keluarlah dari **SSMS**.

## Mengambil Data Menggunakan Query SQL SELECT

**Statement SQL SELECT** merupakan perintah yang akan Anda gunakan untuk memilih, atau select data yang Anda butuhkan dari dalam *database* ke aplikasi Anda.

Sebuah *statement SQL SELECT* dapat dipecah dalam beberapa elemen, di mana setiap elemen diawali dengan sebuah *keyword*. Berikut elemen yang paling banyak digunakan dari sebuah *statement SELECT*, yaitu:

- **SELECT**
- **FROM**
- **WHERE**
- **ORDER BY**

### Clause SELECT ... FROM

Sebuah *statement SELECT* yang paling sederhana hanya memiliki **2** bagian, yaitu:

- (1) *field* apa yang akan Anda tampilkan.
- (2) dari *table* apa *field* tersebut berasal.

Jika Anda ingin mendapatkan semua informasi dari semua customer dari dalam *table Customer*, maka Anda bisa menggunakan tanda **asterisk (\*)**.

```
SELECT * FROM Customer
```

Jika Anda hanya ingin mendapatkan beberapa *field* saja, Anda bisa secara eksplisit menuliskannya sebagai sebuah daftar yang dipisahkan dengan tanda koma, seperti:

```
SELECT Cust_nama, Cust_email, Cust_reg_on, Cust_kota  
FROM Customer
```

Yang akan memberikan data berisi *field-field* yang Anda tuliskan dalam perintah *query*.

	Cust_nama	Cust_email	Cust_reg_on	Cust_kota
1	Cash	-	2008-12-27 00:00:00	-
2	Satu Persatu	satu_p@email.com	2013-04-14 00:00:00	Batam
3	Dua Perdua	dua_p@email.com	2013-04-22 00:00:00	Bintan
4	Tiga Pertiga	tiga_p@email.com	2013-05-02 00:00:00	Batam
5	Empat Perempat	empat_p@email.com	2013-08-22 00:00:00	Dumai
6	Lima Perlima	lima_p@email.com	2013-12-03 00:00:00	Batam

### **Hasil dari statement query**

Menuliskan *field* yang dibutuhkan secara eksplisit juga akan membantu Anda mengatur urutan-urutan dari *field* yang akan diambil, sehingga jika misalnya Anda menginginkan data **Cust\_reg\_on** ditampilkan terlebih dulu dibanding *field-field* lainnya. Anda bisa menuliskan seperti ini:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer
```

## **Clause WHERE**

Hal lain yang perlu dipelajari selanjutnya adalah membatasi, atau menggunakan filter. Dengan menambahkan sebuah **clause WHERE** pada *statement SELECT*, maka Anda sedang menambahkan sebuah kondisi yang harus dipenuhi oleh data-data yang terpilih.

Anda bisa melanjutkan dari *query* sebelumnya, dan membatasi hasilnya untuk hanya menunjukkan customer yang tinggal di Yogyakarta saja:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Kota = 'Yogyakarta'
```

Jika yang Anda inginkan adalah kebalikannya, customer yang tidak tinggal di Yogyakarta, maka Anda akan menuliskan:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Kota <> 'Yogyakarta'
```

Anda bisa menulis kondisi yang lebih rumit. Cara yang paling jelas untuk melakukan itu adalah dengan menggunakan kondisi yang lebih dari satu pada *clause WHERE*. Jika Anda ingin mengetahui customer mana yang di-registrasi setelah tanggal tertentu, dan hanya yang tinggal di Yogyakarta saja, maka Anda bisa menuliskan:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Cust_reg_on >= ('2-May-2013') AND (cust_kota =  
'Yogyakarta')
```

Perhatikan juga bahwa **SQL** memiliki *operator* spesial **BETWEEN** yang akan menguji apakah nilai berada di antara dua buah nilai lain. Misalnya:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Cust_reg_on BETWEEN '01-May-2013' AND '31-August-2013'
```

Anda juga bisa menambahkan *operator NOT*, untuk mendapatkan baris data yang *tidak* berada di antara dua tanggal:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Cust_reg_on NOT BETWEEN '01-May-2013' AND '31-August-  
2013'
```

Selanjutnya, bagaimana jika Anda hendak menguji apakah nilai dari sebuah *field* sama untuk beberapa buah nilai lain? Anda bisa menggunakan *operator OR* untuk menggabungkan kedua kondisi, seperti:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE (Cust_kota = 'Jakarta') OR (Cust_kota = 'Manado')
```

Namun, jika pada data terdapat banyak kondisi nilai yang ingin Anda bandingkan, Anda bisa menggunakan *operator IN* untuk menguji terhadap suatu kumpulan nilai. Misalnya, *query* di atas dapat ditulis ulang sebagai berikut:

```
SELECT Cust_reg_on, Cust_nama, Cust_email, Cust_kota  
FROM Customer  
WHERE Cust_kota IN ('Jakarta', 'Manado')
```

**LIKE** digunakan untuk melakukan suatu pengenalan pola dasar dengan menggunakan karakter *wildcard*. Untuk **Microsoft SQL Server**, karakter *wildcard* yang bisa digunakan adalah sebagai berikut:

Wildcard	Keterangan
_ (garis bawah)	sama dengan karakter tunggal
%	sama dengan urutan satu atau lebih karakter
[ ]	sama dengan karakter tunggal dalam range ([a-f] atau kumpulan ([abcdef]))
[^]	sama dengan karakter tunggal yang tidak dalam range ([^a-f]) atau kumpulan ([^abcdef])

Beberapa contoh akan dapat menjelaskan aturan-aturan tersebut:

- WHERE Cust\_nama LIKE 'B\_i' untuk mencari dan mendapatkan semua nama yang hanya terdiri dari tiga karakter, yang diawali dengan huruf 'B' dan diakhiri dengan huruf 'i'.
- WHERE Cust\_kota LIKE 'B%' untuk mencari dan mendapatkan semua customer yang tinggal di kota yang diawali dengan huruf 'B'.
- WHERE Cust\_nama LIKE 'B[jn]i' untuk mencari dan mendapatkan semua customer yang namanya diawali dengan huruf 'B' dan diakhiri dengan huruf 'i', namun huruf di tengahnya hanya yang berupa huruf 'j' dan 'n' saja.
- WHERE Cust\_nama LIKE 'A[^s]%' akan mendapatkan semua nama yang diawali dengan huruf 'A' di mana huruf keduanya bukan 's'.

# Bab 4

## Latihan Pemrograman Visual Basic Database: Aplikasi Gudang

Untuk membantu Anda melatih cara menggunakan Visual Basic, maka di dalam bab ini akan disajikan langkah-langkah pemrograman sebuah aplikasi. Kita akan membuat aplikasi Gudang yang memiliki beberapa form dengan fungsi spesifik, seperti menambah dan menghapus nama suplier, nama pelanggan, dan sebagainya. Aplikasi ini melibatkan database sebagai tempat penyimpanan data-data.

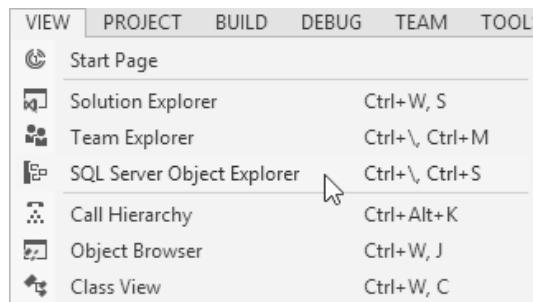
Diharapkan setelah mengikuti langkah-langkah pembuatan aplikasi di bawah ini, Anda bisa memahami alur kerja Visual Basic dari awal hingga akhir.

### Menyiapkan Database

Aplikasi gudang ini membutuhkan database sebagai penyimpan data. Oleh karena itu, tahap awal pembuatan aplikasi ini adalah menyiapkan database terlebih dulu.

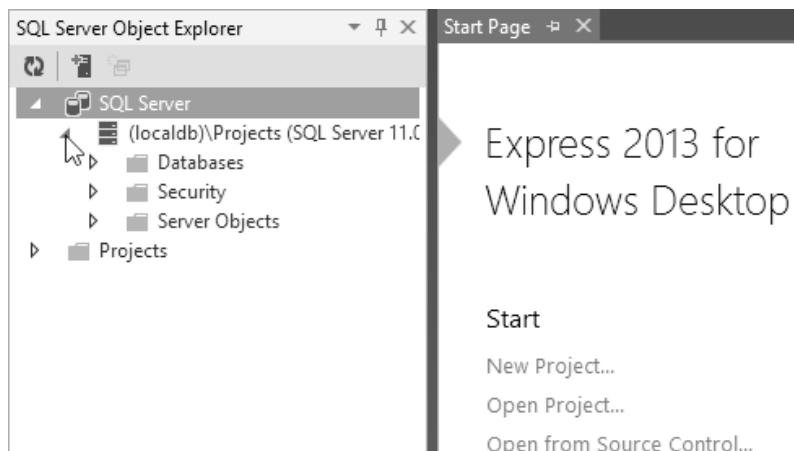
Kita akan membuat database Gudang yang akan digunakan untuk penyimpanan berbagai data yang diperlukan. Berikut langkah-langkahnya:

1. Jalankan **Visual Studio Express 2013**.
2. Buka **SQL Server Object Explorer** dengan memilih **View – SQL Server Object Explorer**.



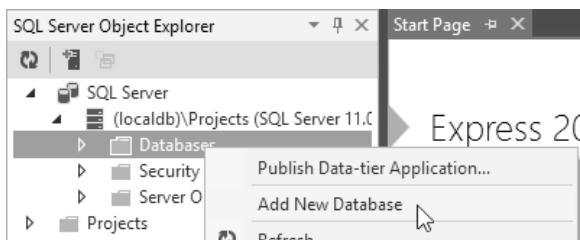
*Membuka SQL Server Object Explorer*

3. Jendela SQL Server Object Explorer akan ditampilkan di sebelah kiri layar.



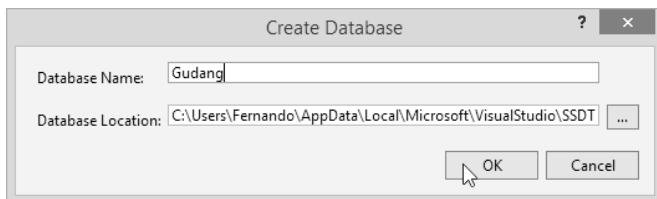
*Jendela SQL Server Object Explorer*

4. Pada jendela SQL Server Object Explorer, perluas **SQL Server**, kemudian klik kanan folder **Database** dan pilih **Add New Database** untuk menambahkan database.



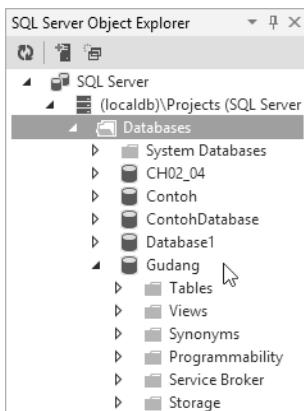
**Menambahkan database**

5. Atur nama database menjadi **Gudang** kemudian klik **OK**.



**Mengatur nama database**

6. Database kemudian akan ditambahkan.



**Database ditambahkan**

## Menambahkan Tabel Barang

Langkah selanjutnya adalah membuat tabel untuk data-data barang. Berikut langkah-langkahnya:

1. Perluas Database **Gudang** kemudian klik kanan folder **Tables** dan pilih **Add New Table...**



*Menambahkan tabel*

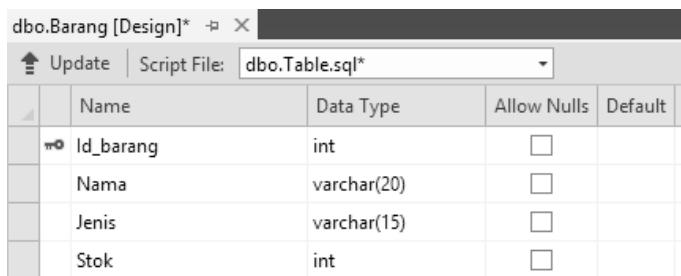
2. Pada jendela T-SQL, ubah nama tabel menjadi **Barang**.



```
CREATE TABLE [dbo].[Barang]
(
    [Id] INT NOT NULL PRIMARY KEY
)
```

*Mengubah nama tabel*

3. Pada jendela **Design**, tambahkan dan atur kolom sebagai berikut.



	Name	Data Type	Allow Nulls	Default
1	Id_barang	int	<input checked="" type="checkbox"/>	
2	Nama	varchar(20)	<input checked="" type="checkbox"/>	
3	Jenis	varchar(15)	<input checked="" type="checkbox"/>	
4	Stok	int	<input checked="" type="checkbox"/>	

*Mengatur kolom dalam tabel*

4. Anda juga dapat mengatur kolom dalam tabel dengan menggunakan kode T-SQL berikut:

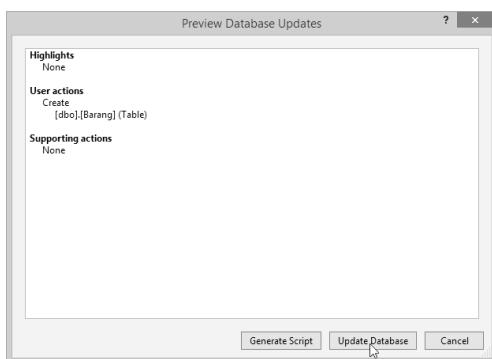
```
CREATE TABLE [dbo].[Barang]
(
    [Id_barang] INT NOT NULL PRIMARY KEY,
    [Nama] VARCHAR(20) NOT NULL,
    [Jenis] VARCHAR(15) NOT NULL,
    [Stok] INT NOT NULL
)
```

5. Setelah kolom dalam tabel diatur, klik **Update** di sebelah kiri atas untuk membuat tabel.

		Type	Allow Nulls	Default
#0	Id_barang	int	<input type="checkbox"/>	
	Nama	varchar(20)	<input type="checkbox"/>	
	Jenis	varchar(15)	<input type="checkbox"/>	
	Stok	int	<input type="checkbox"/>	

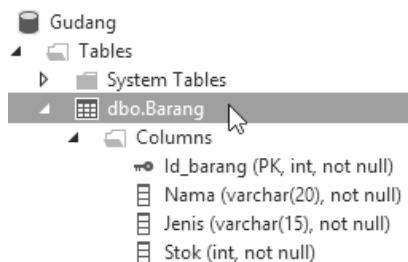
**Memilih Update**

6. Pada jendela **Preview** klik **Update Database**.



**Memilih Update Database**

7. Tabel **Barang** kemudian akan ditambahkan dalam database **Gudang**.

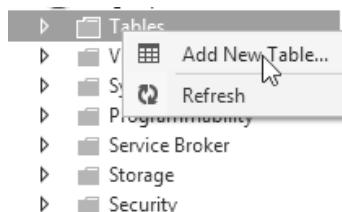


*Tabel ditambahkan dalam database*

## Menambahkan Tabel Suplier

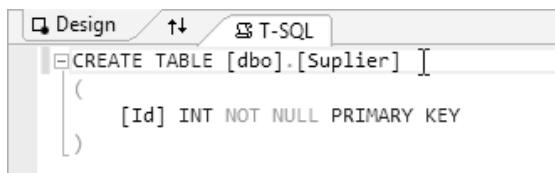
Selanjutnya, kita akan membuat tabel untuk penyimpanan data-data suplier. Berikut langkah-langkahnya:

1. Perluas Database **Gudang** kemudian klik kanan folder **Tables** dan pilih **Add New Table...**



*Menambahkan tabel*

2. Pada jendela T-SQL, ubah nama tabel menjadi **Suplier**.



```
CREATE TABLE [dbo].[Suplier]
(
    [Id] INT NOT NULL PRIMARY KEY
)
```

*Mengubah nama tabel*

3. Pada jendela Design, tambahkan dan atur kolom sebagai berikut.

	Name	Data Type	Allow Nulls	Default
1	Id_suplier	int	<input checked="" type="checkbox"/>	
2	Nama	varchar(20)	<input checked="" type="checkbox"/>	
3	Alamat	varbinary(50)	<input checked="" type="checkbox"/>	
4	Telepon	varchar(15)	<input checked="" type="checkbox"/>	

**Mengatur kolom dalam tabel**

4. Anda juga dapat mengatur kolom dalam tabel dengan menggunakan kode T-SQL berikut:

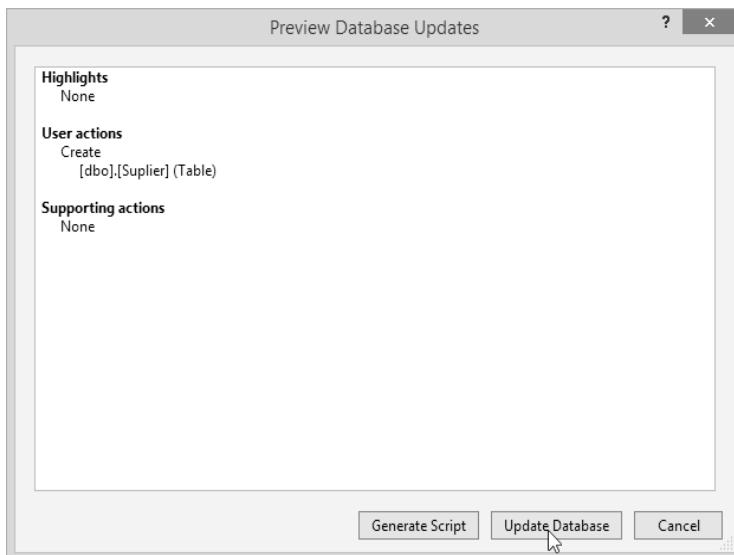
```
CREATE TABLE [dbo].[Suplier]
(
    [Id_suplier] INT NOT NULL PRIMARY KEY,
    [Nama] VARCHAR(20) NOT NULL,
    [Alamat] VARBINARY(50) NOT NULL,
    [Telepon] VARCHAR(15) NOT NULL
)
```

5. Setelah kolom dalam tabel diatur, klik **Update** di sebelah kiri atas untuk membuat tabel.

	Name	Data Type	Allow Nulls	Default
1	Id_suplier	int	<input checked="" type="checkbox"/>	
2	Nama	varchar(20)	<input checked="" type="checkbox"/>	
3	Alamat	varbinary(50)	<input checked="" type="checkbox"/>	
4	Telepon	varchar(15)	<input checked="" type="checkbox"/>	

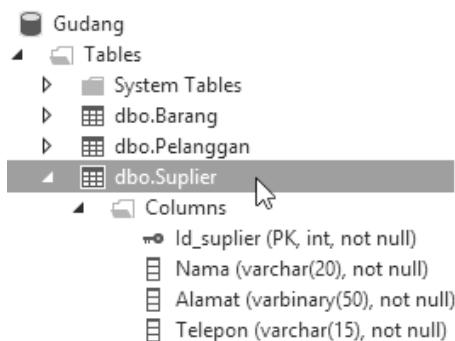
**Memilih Update**

6. Pada jendela **Preview** klik **Update Database**.



***Memilih Update Database***

7. Tabel **Suplier** kemudian akan ditambahkan dalam database **Gudang**.

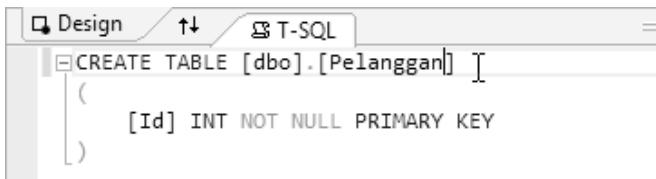


***Tabel ditambahkan dalam database***

## Menambahkan Tabel Pelanggan

Tabel berikutnya yang akan dibuat adalah tabel Pelanggan untuk menginput data-data para pelanggan. Berikut langkah-langkahnya:

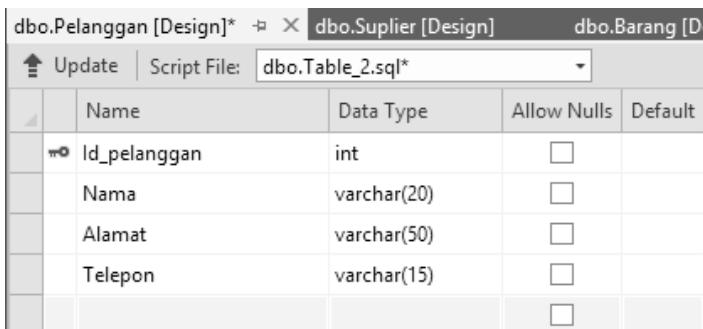
1. Perluas Database **Gudang** kemudian klik kanan folder **Tables** dan pilih **Add New Table...**
2. Pada jendela T-SQL, ubah nama tabel menjadi **Pelanggan**.



```
Design T-SQL
CREATE TABLE [dbo].[Pelanggan]
(
    [Id] INT NOT NULL PRIMARY KEY
)
```

*Mengubah nama tabel*

3. Pada jendela **Design**, tambahkan dan atur kolom sebagai berikut.



	Name	Data Type	Allow Nulls	Default
1	Id_pelanggan	int	<input type="checkbox"/>	
2	Nama	varchar(20)	<input type="checkbox"/>	
3	Alamat	varchar(50)	<input type="checkbox"/>	
4	Telepon	varchar(15)	<input type="checkbox"/>	

*Mengatur kolom dalam tabel*

4. Anda juga dapat mengatur kolom dalam tabel dengan menggunakan kode T-SQL berikut:

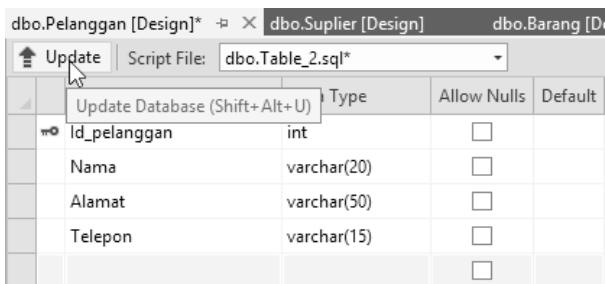
```
CREATE TABLE [dbo].[Pelanggan]
(
    [Id_pelanggan] INT NOT NULL PRIMARY KEY,
```

```

[Nama] VARCHAR(20) NOT NULL,
[Alamat] VARCHAR(50) NOT NULL,
[Telepon] VARCHAR(15) NOT NULL
)

```

- Setelah kolom dalam tabel diatur, klik **Update** di sebelah kiri atas untuk membuat tabel.

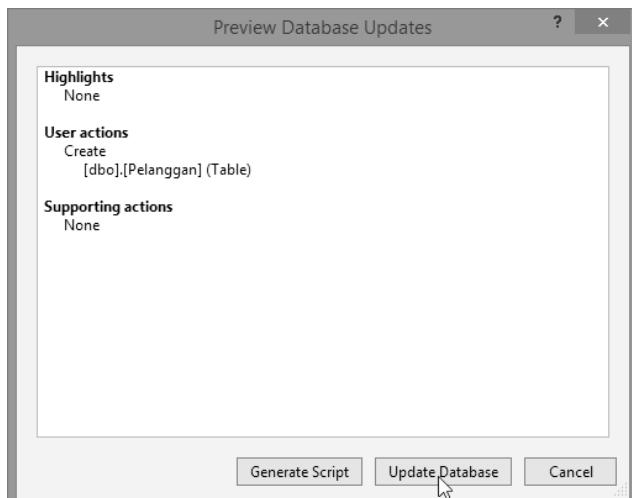


The screenshot shows the SSMS Object Explorer with three tables listed: 'dbo.Pelanggan [Design]\*', 'dbo.Suplier [Design]', and 'dbo.Barang [D...'. The 'Pelanggan' table is currently selected, indicated by a blue border around its name. The table's design view is displayed below, showing four columns: 'Id\_pelanggan' (int), 'Nama' (varchar(20)), 'Alamat' (varchar(50)), and 'Telepon' (varchar(15)). Each column has a 'Type' column header, 'Allow Nulls' checkbox, and a 'Default' column header. The 'Update' button is highlighted with a mouse cursor.

		Type	Allow Nulls	Default
Id_pelanggan	int	<input type="checkbox"/>		
Nama	varchar(20)	<input type="checkbox"/>		
Alamat	varchar(50)	<input type="checkbox"/>		
Telepon	varchar(15)	<input type="checkbox"/>		

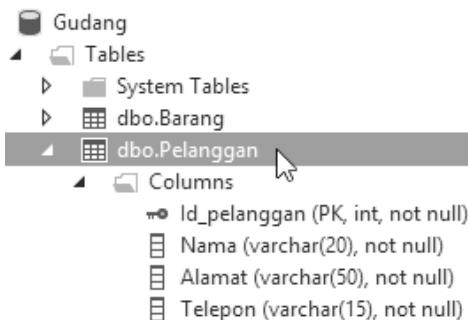
### ***Memilih Update***

- Pada jendela **Preview** klik **Update Database**.



### ***Memilih Update Database***

7. Tabel **Pelanggan** kemudian akan ditambahkan dalam database **Gudang**.



*Tabel ditambahkan dalam database*

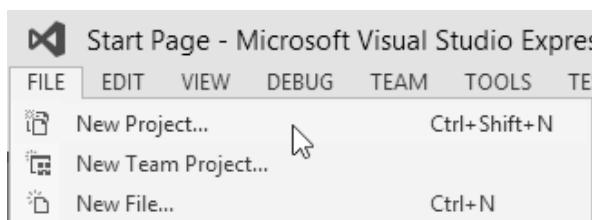
## Menyiapkan Form Aplikasi

Agar mudah digunakan, aplikasi Gudang ini akan dilengkapi dengan berbagai form. Dengan demikian, user cukup menginput data-data melalui kontrol-kontrol di dalam form tersebut.

Berikut ini langkah-langkah untuk menyiapkan form-form yang akan digunakan dalam aplikasi Gudang.

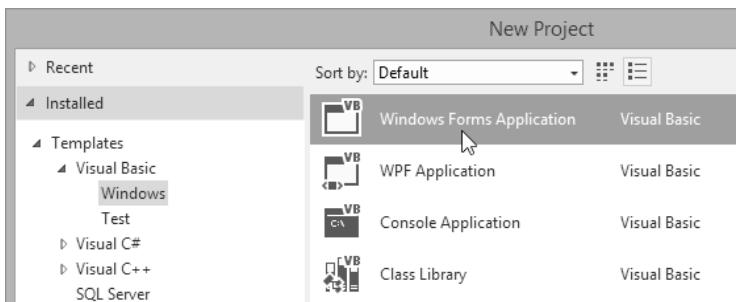
### Form Utama

1. Pilih menu **File – New Project...**



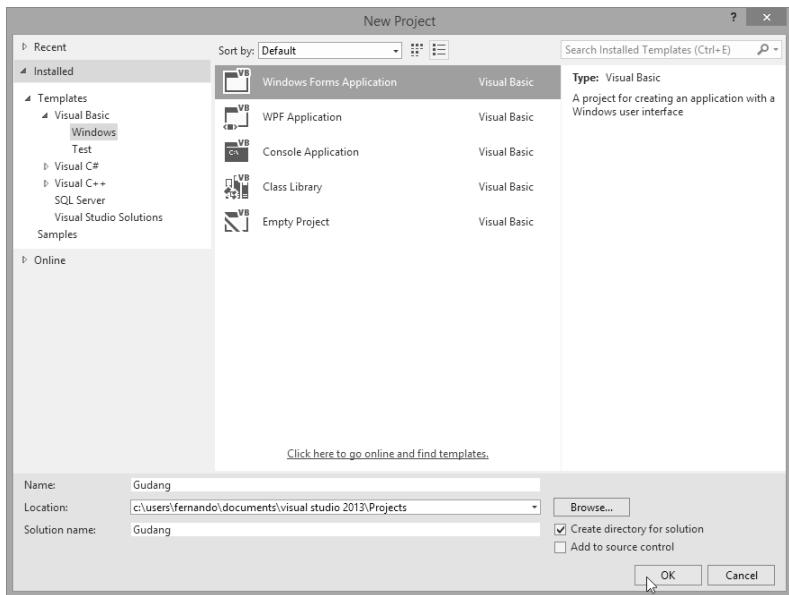
*Memilih New Project...*

2. Pada jendela **New Project**, pilih **Templates Visual Basic - Windows Forms Application**.



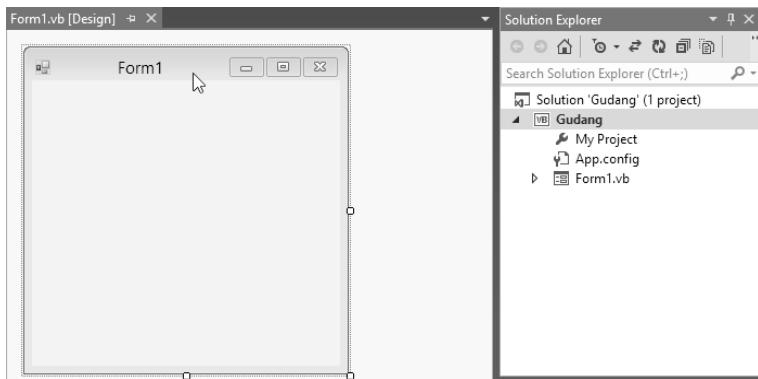
**Memilih Windows Forms Application**

3. Atur nama project menjadi **Gudang** kemudian klik **OK**.



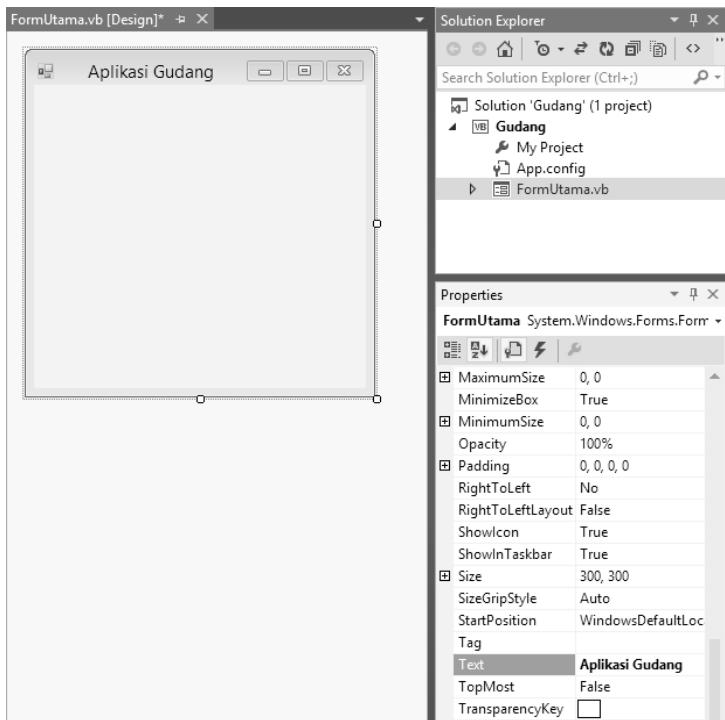
**Mengatur nama project**

4. Form baru akan ditampilkan dalam project.



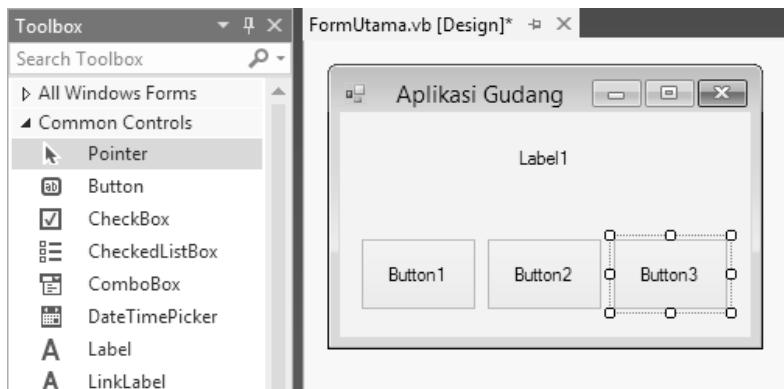
*Form ditampilkan*

5. Pilih form kemudian ubah property **Name** menjadi **FormUtama** dan property **Text** menjadi **Aplikasi Gudang** pada jendela **Properties**.



*Mengatur properti Name dan Text form*

6. Tambahkan **Label** dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



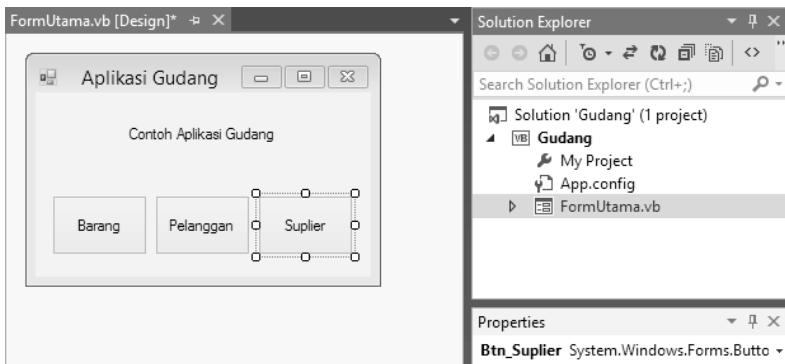
**Menambahkan kontrol**

7. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	Contoh Aplikasi Gudang
Button1	Name	Btn_Barang
	Text	Barang
Button2	Name	Btn_Pelanggan
	Text	Pelanggan
Button3	Name	Btn_Supplier
	Text	Supplier

**Tabel properti kontrol**

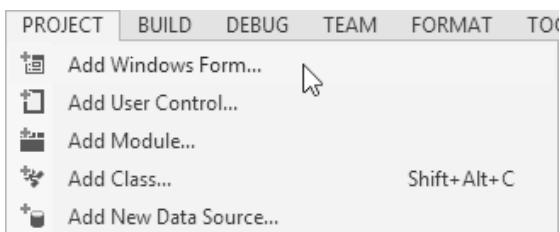
8. Form akan terlihat seperti pada gambar berikut.



**Form Utama**

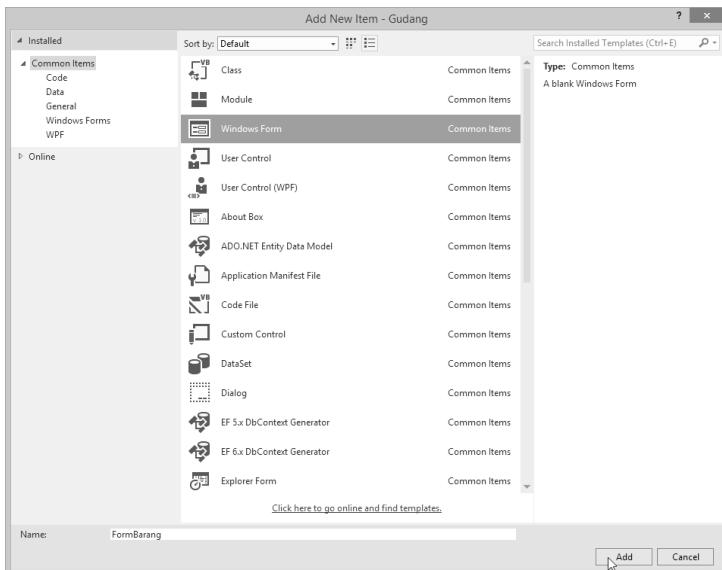
## Form Barang

1. Pilih Project – Add Windows Form.



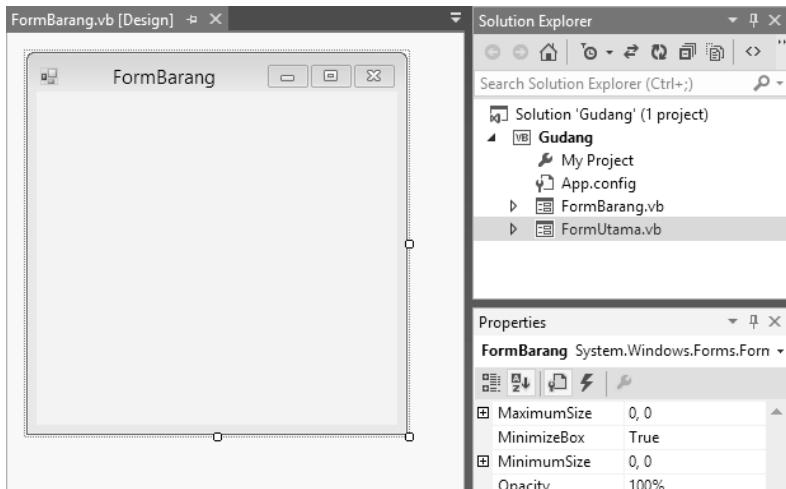
**Memilih Add Windows Form**

2. Pilih Windows Form dan ubah nama menjadi **FormBarang**, kemudian klik **Add**.



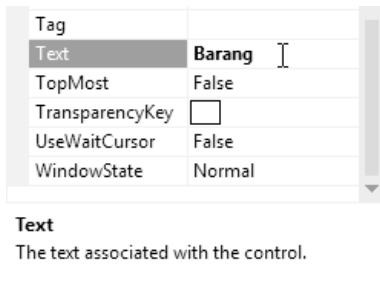
### ***Menambahkan Windows Forms***

3. FormBarang akan ditampilkan dalam project.



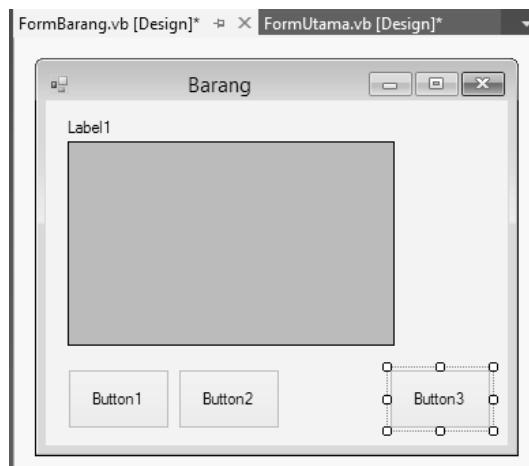
***Form ditampilkan***

4. Pilih form kemudian ubah property **Text** menjadi **Barang** pada jendela **Properties**.



**Mengatur properti Name dan Text form**

5. Tambahkan **Label**, **DataGridView**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



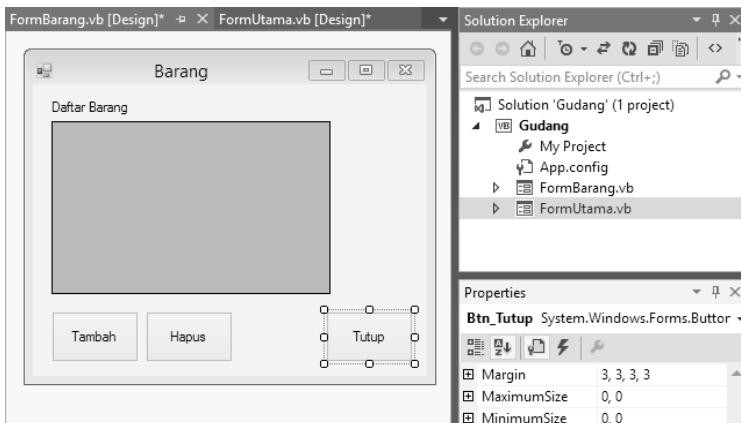
**Menambahkan kontrol**

6. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	Daftar Barang
Button1	Name	Btn_Tambah
	Text	Tambah
Button2	Name	Btn_Hapus
	Text	Hapus
Button3	Name	Btn_Tutup
	Text	Tutup

***Tabel properti kontrol***

7. Form akan terlihat seperti pada gambar berikut.

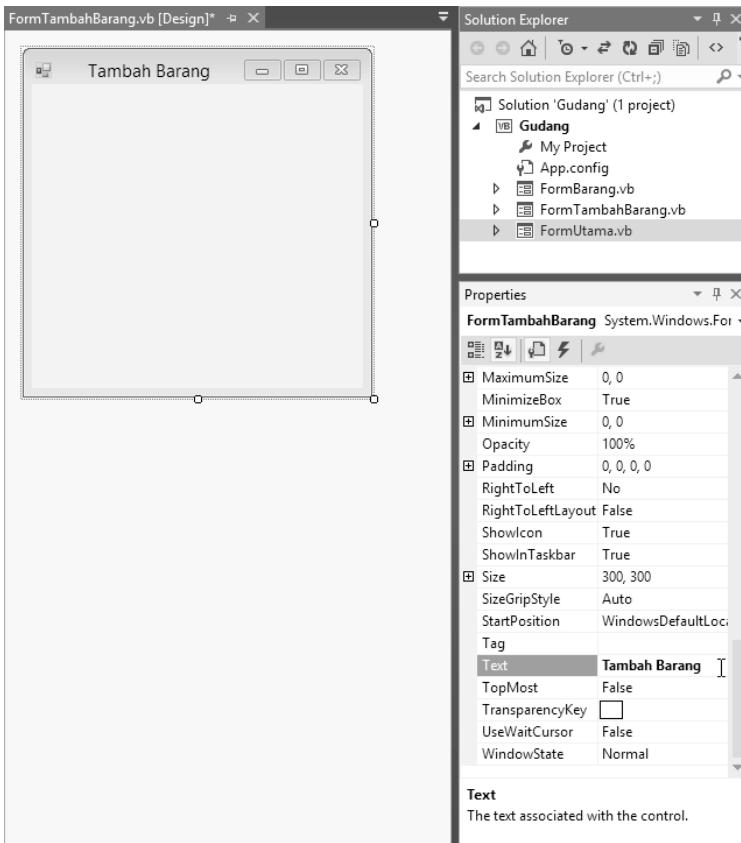


***Form Barang***

## **Form TambahBarang**

1. Pilih Project – Add Windows Form.

- Pilih **Windows Form** dan ubah nama menjadi **FormTambahBarang**, kemudian klik **Add**.
- FormTambahBarang akan ditampilkan dalam project.
- Pilih form kemudian ubah property **Text** menjadi **Tambah Barang** pada jendela **Properties**.



#### *Mengatur properti Name dan Text form*

- Tambahkan empat **Label**, empat **TextBox**, dan dua **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



### ***Menambahkan kontrol***

6. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Barang
Label2	Text	Nama
Label3	Text	Jenis
Label4	Text	Stok
TextBox1	Name	Txt_IdBarang
TextBox2	Name	Txt_Nama
TextBox3	Name	Txt_Jenis
TextBox4	Name	Txt_Stok
Button1	Name	Btn_Tambah
	Text	Tambah
Button2	Name	Btn_Batal
	Text	Batal

***Tabel properti kontrol***

7. Form akan terlihat seperti pada gambar berikut.



*Form TambahBarang*

## Form HapusBarang

1. Pilih Project – Add Windows Form.
2. Pilih **Windows Form** dan ubah nama menjadi **FormHapusBarang**, kemudian klik **Add**.
3. FormHapusBarang akan ditampilkan dalam project.
4. Pilih form, kemudian ubah property **Text** menjadi **Hapus Barang** pada jendela **Properties**.
5. Tambahkan dua **Label**, dua **TextBox**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



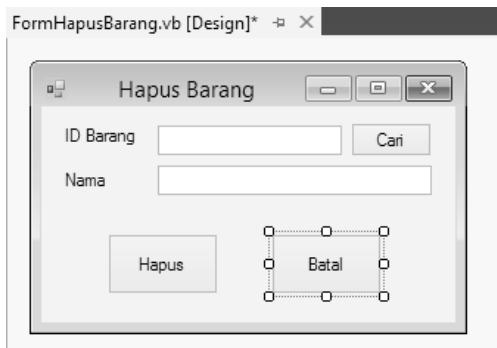
*Menambahkan kontrol*

6. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Barang
Label2	Text	Nama
TextBox1	Name	Txt_IdBarang
TextBox2	Name	Txt_Nama
Button1	Name	Btn_Hapus
	Text	Hapus
Button2	Name	Btn_Batal
	Text	Batal
Button3	Name	Btn_Cari
	Text	Cari

***Tabel properti kontrol***

7. Form akan terlihat seperti pada gambar berikut.



***Form HapusBarang***

## Form Pelanggan

1. Pilih Project – Add Windows Form.
2. Pilih **Windows Form** dan ubah nama menjadi **FormPelanggan**, kemudian klik **Add**.
3. Pilih form, kemudian ubah property **Text** menjadi **Pelanggan** pada jendela **Properties**.
4. Tambahkan **Label**, **DataGridView**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



*Menambahkan kontrol*

5. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	Daftar Pelanggan
Button1	Name	Btn_Tambah
	Text	Tambah

Button2	Name	Btn_Hapus
	Text	Hapus
Button3	Name	Btn_Tutup
	Text	Tutup

***Tabel properti kontrol***

- Form akan terlihat seperti pada gambar berikut.



***Form Pelanggan***

## **Form TambahPelanggan**

- Pilih Project – Add Windows Form.
- Pilih Windows Form dan ubah nama menjadi **FormTambahPelanggan**, kemudian klik **Add**.
- Pilih form kemudian ubah property **Text** menjadi **TambahPelanggan** pada jendela **Properties**.
- Tambahkan empat **Label**, empat **TextBox**, dan dua **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



***Menambahkan kontrol***

5. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Pelanggan
Label2	Text	Nama
Label3	Text	Alamat
Label4	Text	Telepon
TextBox1	Name	Txt_IdPelanggan
TextBox2	Name	Txt_Nama
TextBox3	Name	Txt_Alamat
TextBox4	Name	Txt_Telepon
Button1	Name	Btn_Tambah
	Text	Tambah
Button2	Name	Btn_Batal
	Text	Batal

***Tabel properti kontrol***

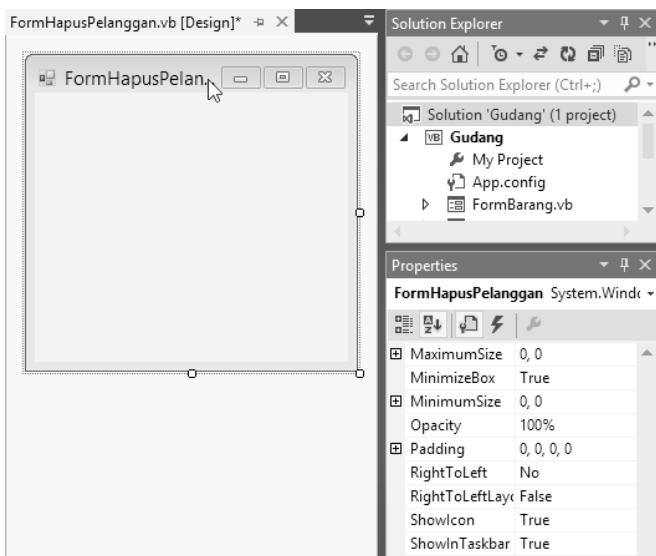
6. Form akan terlihat seperti pada gambar berikut.



**Form Tambah Pelanggan**

## Form HapusPelanggan

1. Pilih Project – Add Windows Form.
2. Pilih Windows Form dan ubah nama menjadi **FormHapusPelanggan**, kemudian klik Add.



**Form ditampilkan**

- Pilih form, kemudian ubah property **Text** menjadi **Hapus Pelanggan** pada jendela **Properties**.
- Tambahkan dua **Label**, dua **TextBox**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



**Menambahkan kontrol**

- Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Pelanggan
Label2	Text	Nama
TextBox1	Name	Txt_IdPelanggan
TextBox2	Name	Txt_Nama
Button1	Name	Btn_Hapus
	Text	Hapus
Button2	Name	Btn_Batal
	Text	Batal
Button3	Name	Btn_Cari
	Text	Cari

**Tabel properti kontrol**

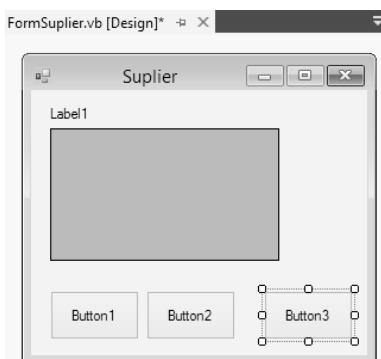
6. Form akan terlihat seperti pada gambar berikut.



**Form HapusPelanggan**

## Form Suplier

1. Pilih Project – Add Windows Form.
2. Pilih Windows Form dan ubah nama menjadi **FormSuplier**, kemudian klik **Add**.
3. Pilih form, kemudian ubah property **Text** menjadi **Suplier** pada jendela **Properties**.
4. Tambahkan **Label**, **DataGridView**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



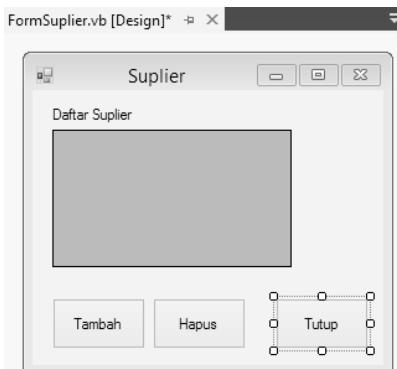
**Menambahkan kontrol**

5. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	Daftar Suplier
Button1	Name	Btn_Tambah
	Text	Tambah
Button2	Name	Btn_Hapus
	Text	Hapus
Button3	Name	Btn_Tutup
	Text	Tutup

*Tabel properti kontrol*

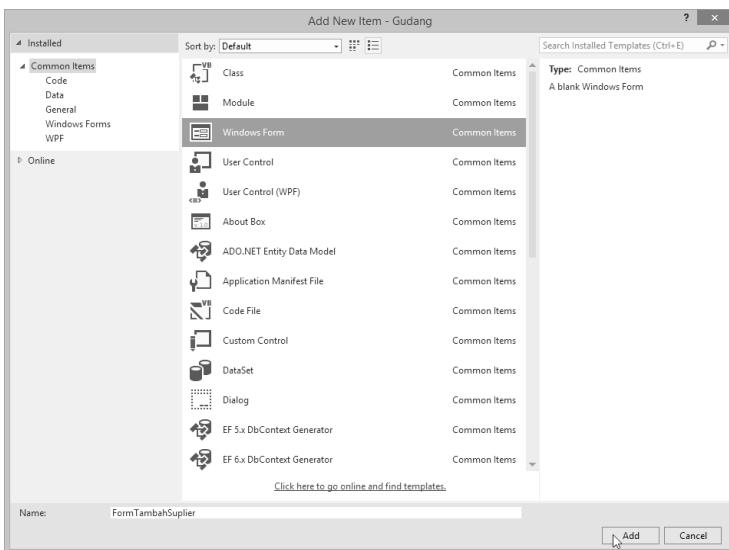
6. Form akan terlihat seperti pada gambar berikut.



*Form Suplier*

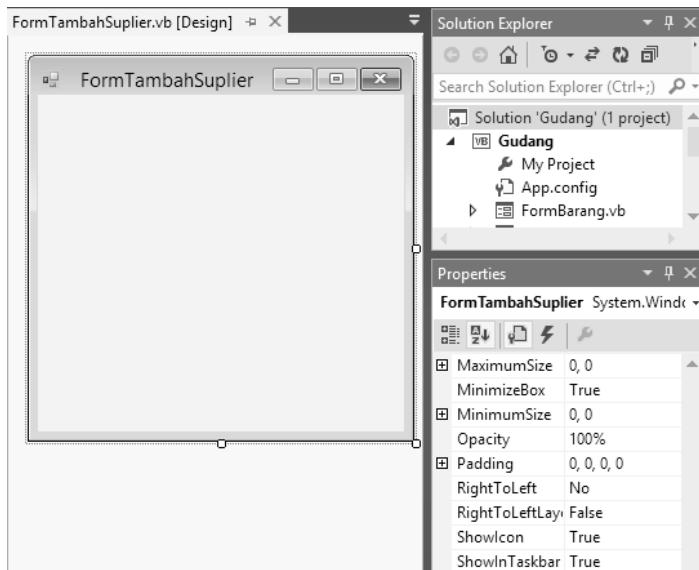
## Form TambahSuplier

1. Pilih Project – Add Windows Form.
2. Pilih Windows Form dan ubah nama menjadi **FormTambahSuplier**, kemudian klik **Add**.



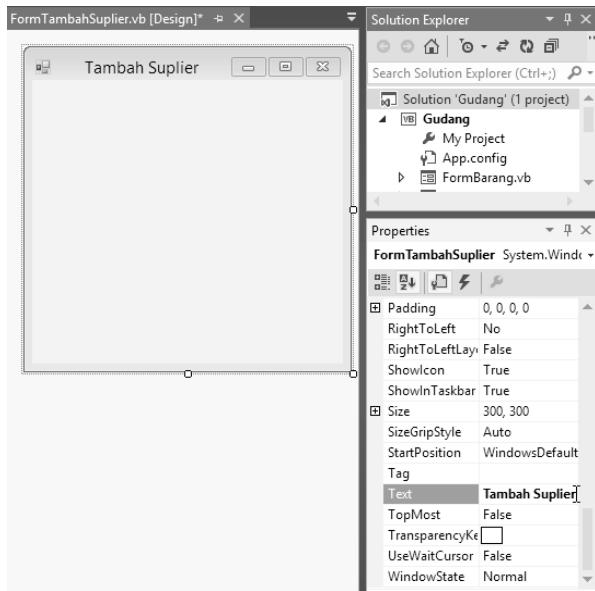
### ***Menambahkan Windows Forms***

3. FormTambahSuplier akan ditampilkan dalam project.



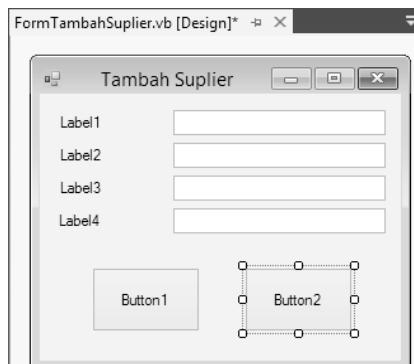
***Form ditampilkan***

4. Pilih form, kemudian ubah property **Text** menjadi Tambah Suplier pada jendela **Properties**.



*Mengatur properti Name dan Text form*

5. Tambahkan empat **Label**, empat **TextBox**, dan dua **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



*Menambahkan kontrol*

6. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Suplier
Label2	Text	Nama
Label3	Text	Alamat
Label4	Text	Telepon
TextBox1	Name	Txt_IdSuplier
TextBox2	Name	Txt_Nama
TextBox3	Name	Txt_Alamat
TextBox4	Name	Txt_Telepon
Button1	Name	Btn_Tambah
	Text	Tambah
Button2	Name	Btn_Batal
	Text	Batal

*Tabel properti kontrol*

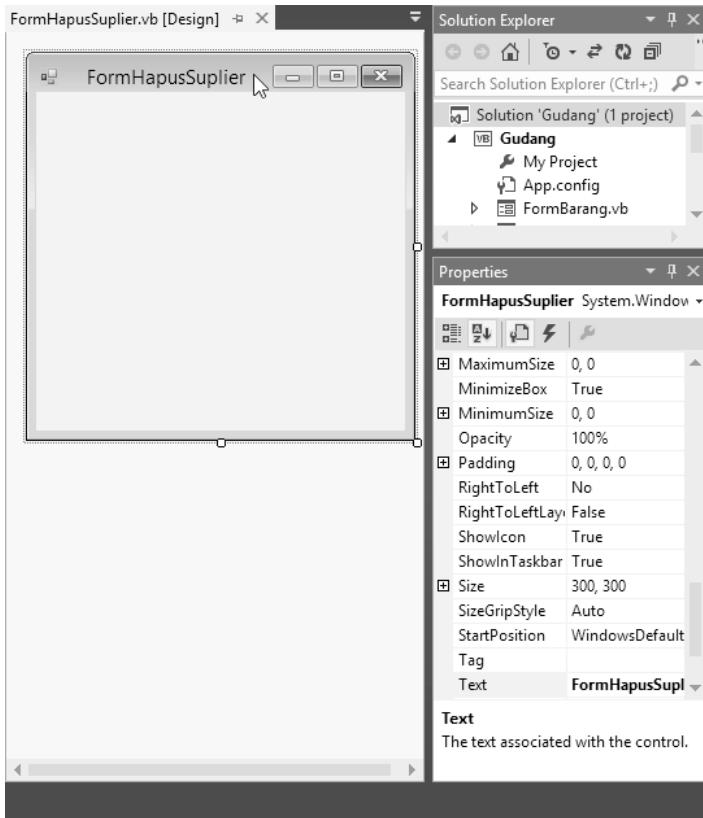
7. Form akan terlihat seperti pada gambar berikut.



*Form TambahSuplier*

## Form HapusSuplier

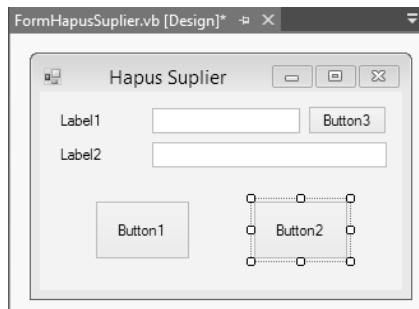
1. Pilih Project – Add Windows Form.
2. Pilih **Windows Form** dan ubah nama menjadi **FormHapusSuplier**, kemudian klik **Add**.
3. FormHapusSuplier akan ditampilkan dalam project.



*Form ditampilkan*

4. Pilih form, kemudian ubah property **Text** menjadi **Hapus Suplier** pada jendela **Properties**.

5. Tambahkan dua **Label**, dua **TextBox**, dan tiga **Button** ke dalam Form. Atur ukuran button dengan menggunakan kotak kecil di sekeliling button.



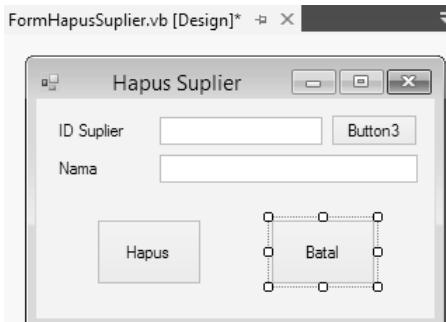
***Menambahkan kontrol***

6. Atur properti dari setiap kontrol sesuai dengan tabel berikut.

KONTROL	PROPERTI	NILAI
Label1	Text	ID Suplier
Label2	Text	Nama
TextBox1	Name	Txt_IdSuplier
TextBox2	Name	Txt_Nama
Button1	Name	Btn_Hapus
	Text	Hapus
Button2	Name	Btn_Batal
	Text	Batal
Button3	Name	Btn_Cari
	Text	Cari

***Tabel properti kontrol***

7. Form akan terlihat seperti pada gambar berikut.



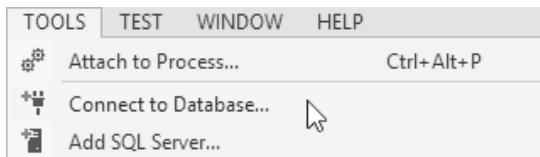
*Form HapusSuplier*

## Menghubungkan Form dengan Database

Setelah pembuatan form selesai, maka kita buat koneksi antara form dan database agar setiap input yang dilakukan oleh user dapat tersimpan dengan baik.

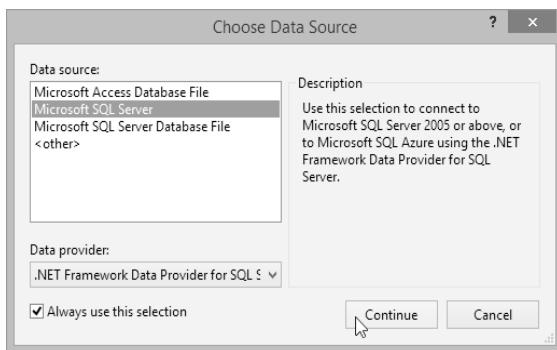
Berikut ini langkah-langkah untuk menghubungkan Form yang ada dengan Database Gudang:

1. Koneksikan aplikasi dengan database dengan memilih **Tools - Connect to Database**.



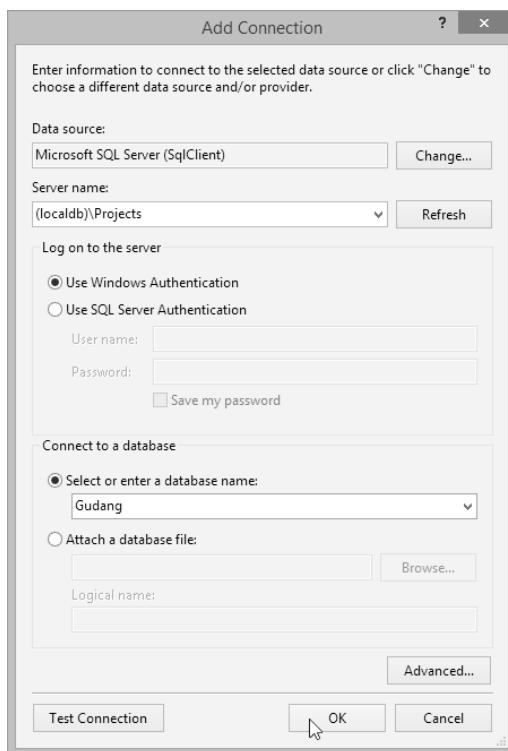
*Memilih Connect to Database*

2. Pilih **Microsoft SQL Server**, kemudian klik **Continue**.



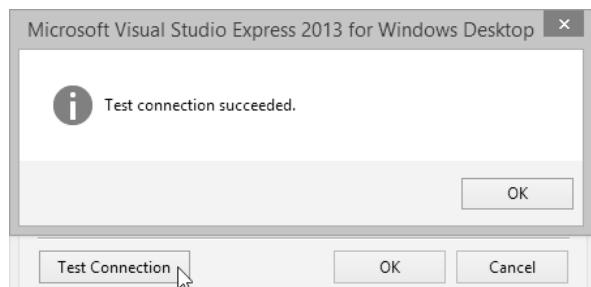
### ***Memilih SQL Server***

3. Masukkan nama server **(localdb)\Projects**, pilih database **Gudang**, kemudian klik **OK**.



### ***Menambahkan koneksi database***

4. Klik **Test Connection** untuk memeriksa apakah koneksi sudah berhasil dibuat.

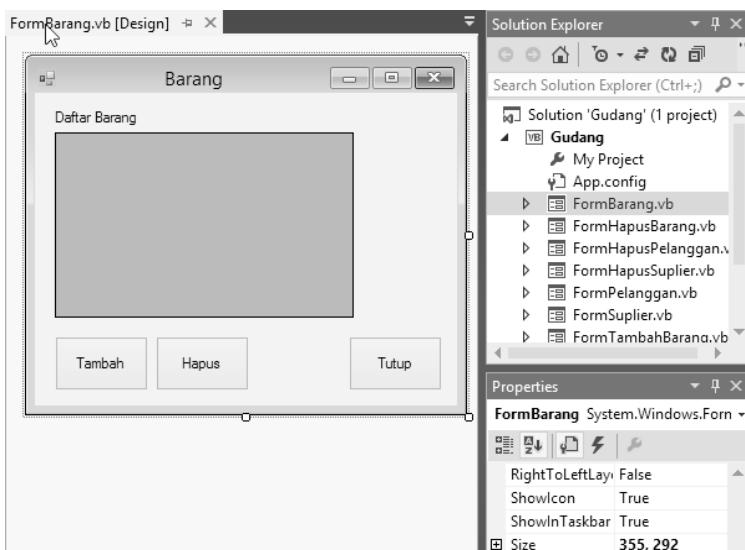


**Memeriksa koneksi**

## Form Barang dengan Tabel Barang

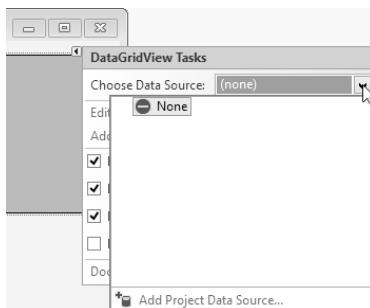
Berikut ini langkah-langkah untuk menghubungkan FormBarang dengan tabel Barang.

1. Buka FormBarang dalam project **Gudang**.



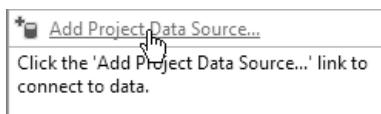
**Menampilkan FormBarang**

2. Pilih **DataGridView**, klik segitiga kecil pada **DataGridView** kemudian klik **Choose Data Source**.



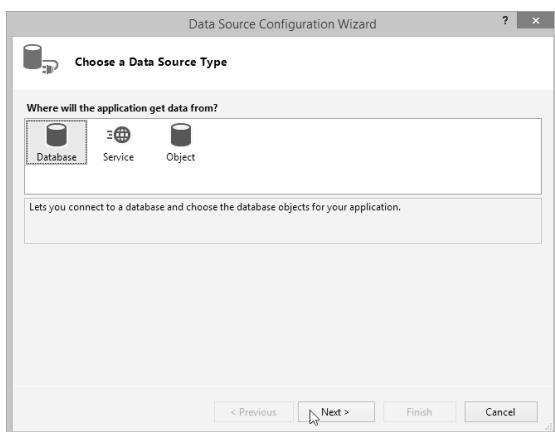
*Memilih Choose Data Source*

3. Klik **Add Project Data Source** untuk menampilkan jendela **Data Source Configuration**.



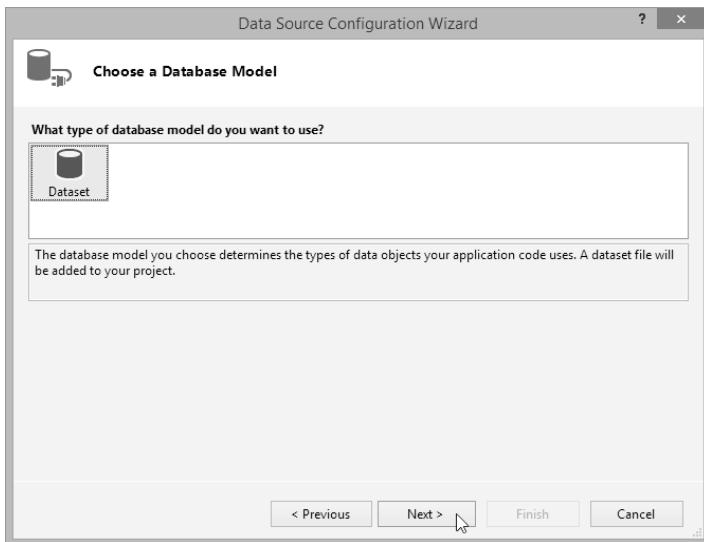
*Memilih Add Project Data Source*

4. Pilih **Database** kemudian klik **Next**.



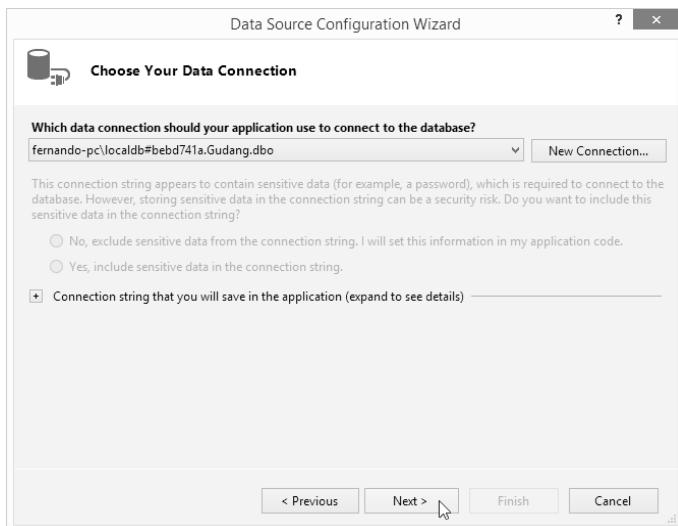
*Memilih Database*

5. Pilih **Dataset**, kemudian klik **Next**.



**Memilih Dataset**

6. Pilih koneksi yang sudah diatur, kemudian klik **Next**.



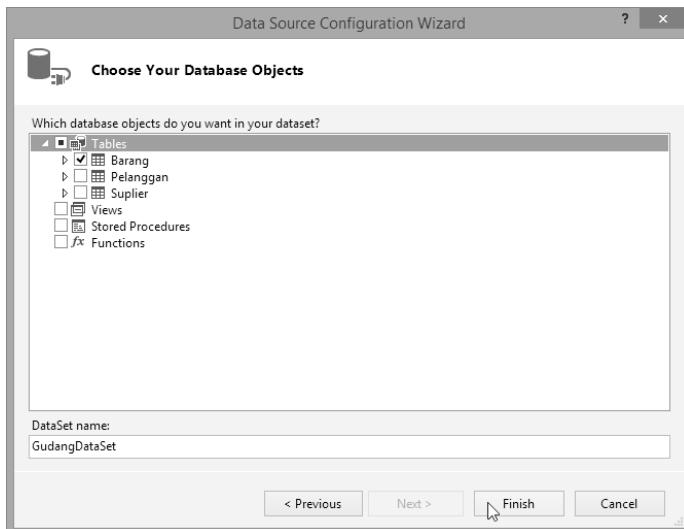
**Memilih koneksi data**

7. Centang Yes, save the connection untuk menyimpan koneksi.



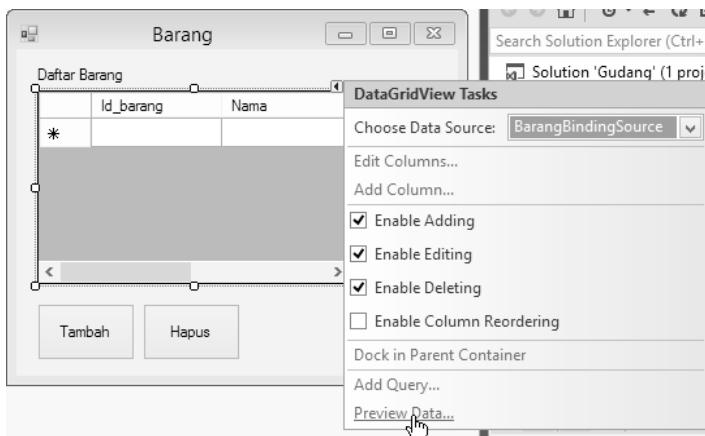
***Menyimpan koneksi***

8. Pilih tabel **Barang** yang akan digunakan, kemudian klik **Finish**.



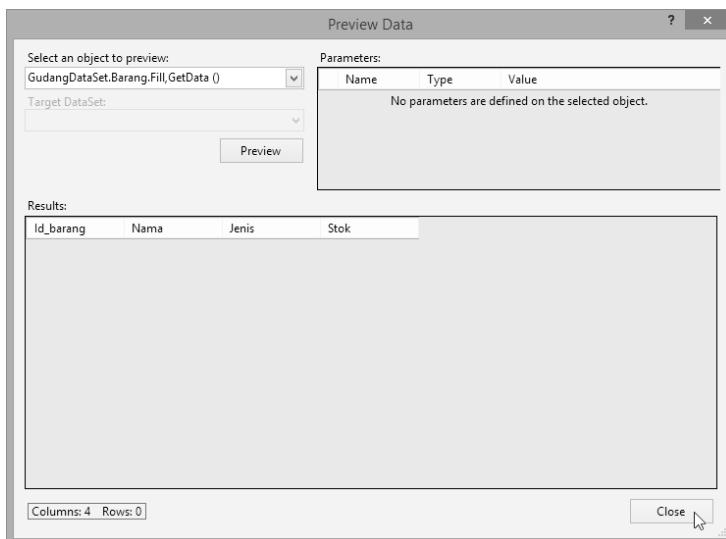
***Memilih tabel***

9. Klik segitiga kecil pada **DataGridView**, kemudian pilih **Preview Data**.



**Memilih Preview Data**

10. Klik **Preview** pada jendela **Preview Data**, kemudian pilih **Close**.



**Jendela Preview Data**

11. Tabel **Barang**, kemudian akan ditampilkan pada **DataGridView** **FormBarang**.

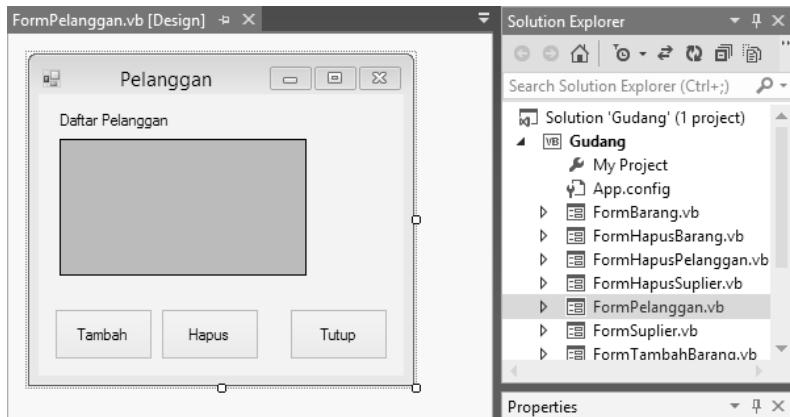


*Tabel ditampilkan dalam form aplikasi*

## Form Pelanggan dengan Tabel Pelanggan

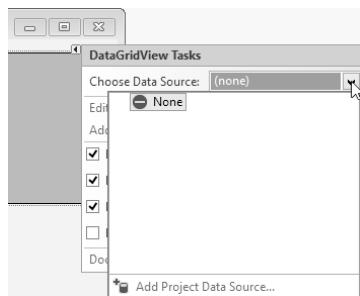
Berikut ini langkah-langkah untuk menghubungkan FormPelanggan dengan tabel Pelanggan:

1. Buka **FormPelanggan** dalam project **Gudang**.



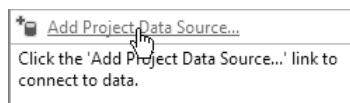
*Menampilkan FormPelanggan*

2. Pilih **DataGridView**, klik segitiga kecil pada **DataGridView** kemudian klik **Choose Data Source**.



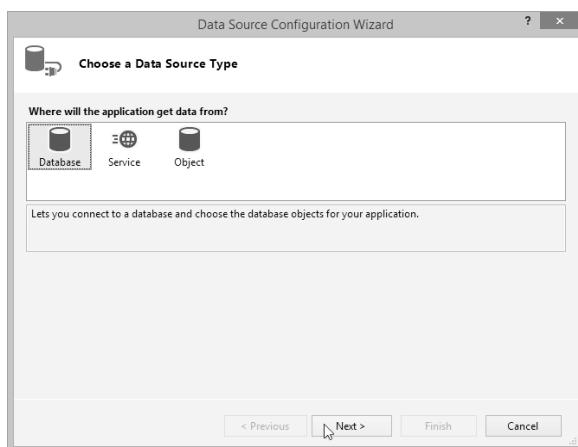
**Memilih Choose Data Source**

3. Klik **Add Project Data Source** untuk menampilkan jendela **Data Source Configuration**.



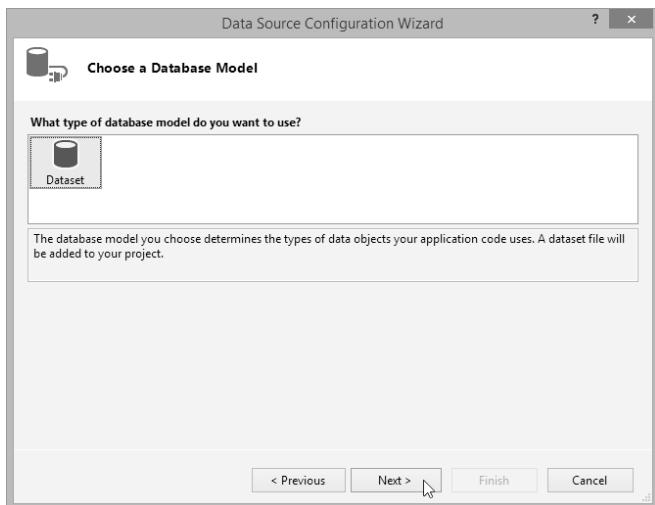
**Memilih Add Project Data Source**

4. Pilih **Database**, kemudian klik **Next**.



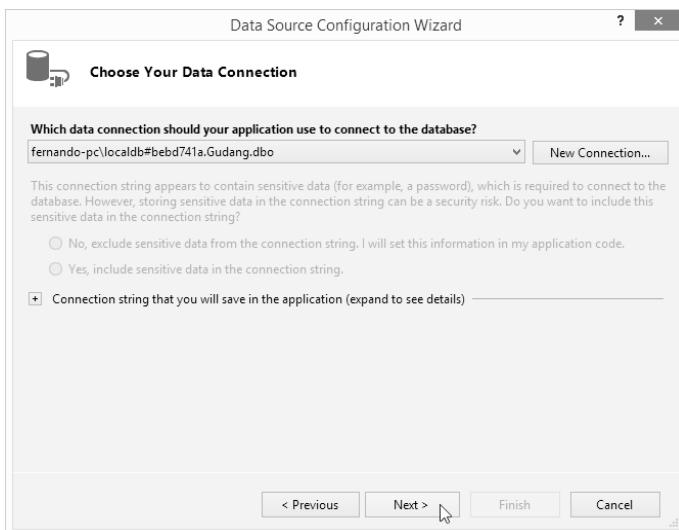
**Memilih Database**

5. Pilih **Dataset**, kemudian klik **Next**.



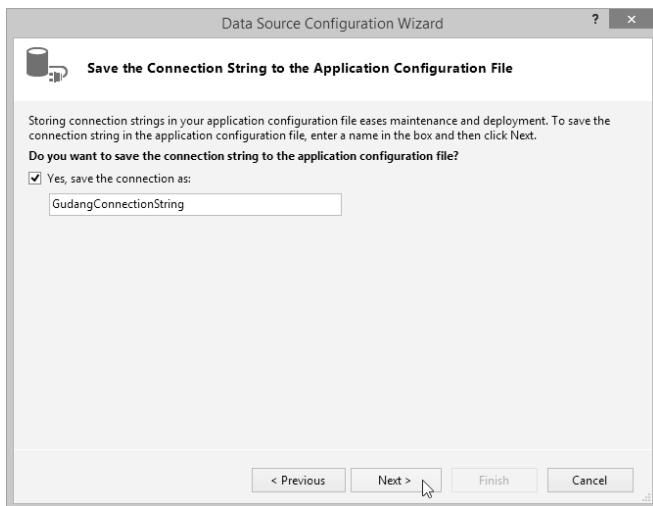
**Memilih Dataset**

6. Pilih koneksi yang sudah diatur, kemudian klik **Next**.



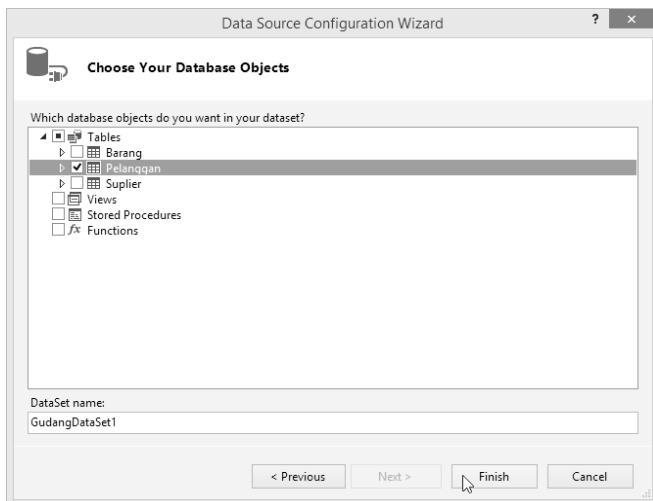
**Memilih koneksi data**

7. Centang Yes, save the connection untuk menyimpan koneksi.



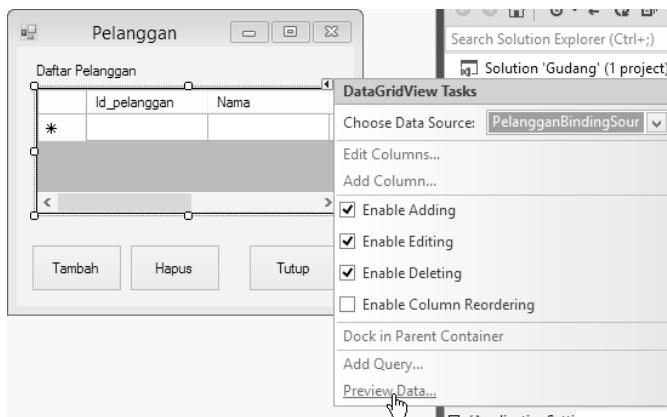
***Menyimpan koneksi***

8. Pilih tabel **Pelanggan** yang akan digunakan, kemudian klik **Finish**.



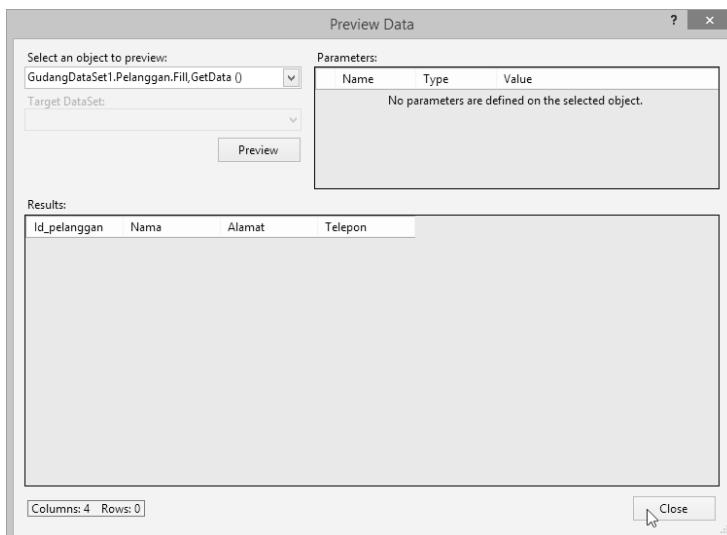
***Memilih tabel***

9. Klik segitiga kecil pada **DataGridView**, kemudian pilih **Preview Data**.



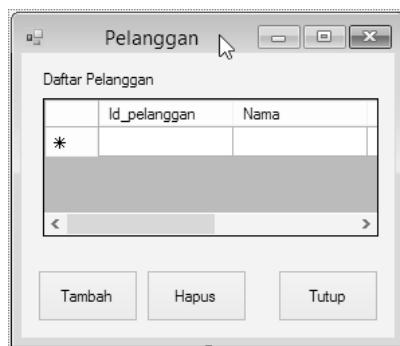
**Memilih Preview Data**

10. Klik **Preview** pada jendela **Preview Data**, kemudian pilih **Close**.



**Jendela Preview Data**

11. Tabel **Pelanggan**, kemudian akan ditampilkan pada **DataGridView** **FormPelanggan**.

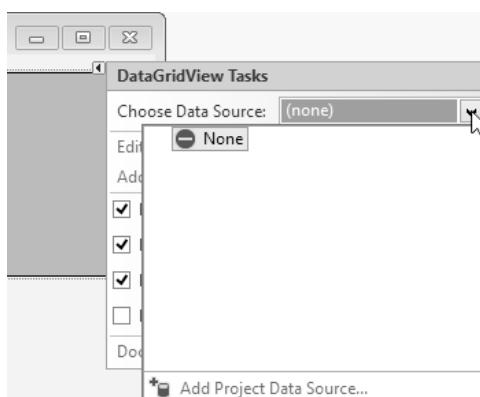


*Tabel ditampilkan dalam form aplikasi*

## Form Suplier dengan Tabel Suplier

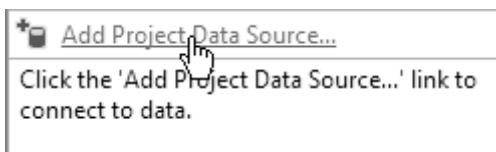
Berikut ini langkah-langkah untuk menghubungkan FormSuplier dengan tabel Suplier:

1. Buka **FormSuplier** dalam project **Gudang**.
2. Pilih **DataGridView**, klik segitiga kecil pada **DataGridView**, kemudian klik **Choose Data Source**.



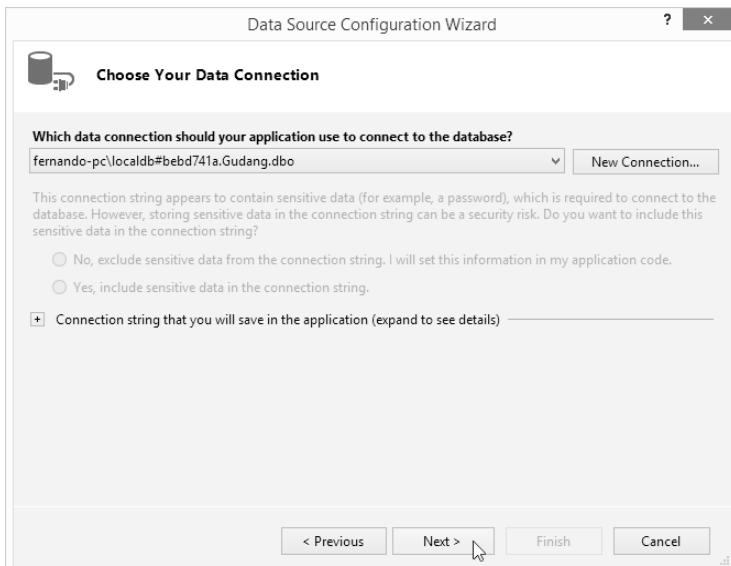
*Memilih Choose Data Source*

3. Klik **Add Project Data Source** untuk menampilkan jendela **Data Source Configuration**.



*Memilih Add Project Data Source*

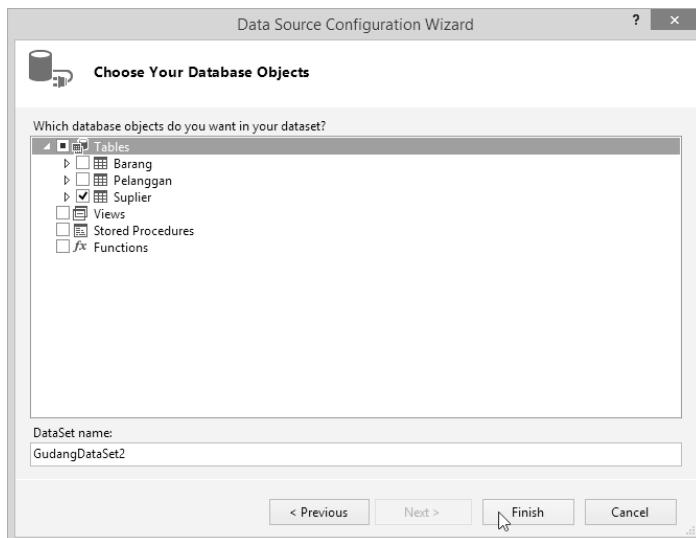
4. Pilih **Database**, kemudian klik **Next**.
5. Pilih **Dataset**, kemudian klik **Next**.
6. Pilih koneksi yang sudah diatur, kemudian klik **Next**.



*Memilih koneksi data*

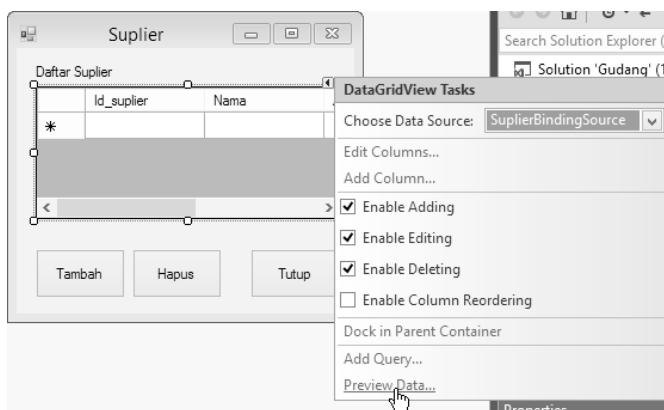
7. Centang **Yes, save the connection** untuk menyimpan koneksi.

8. Pilih tabel **Suplier** yang akan digunakan, kemudian klik **Finish**.



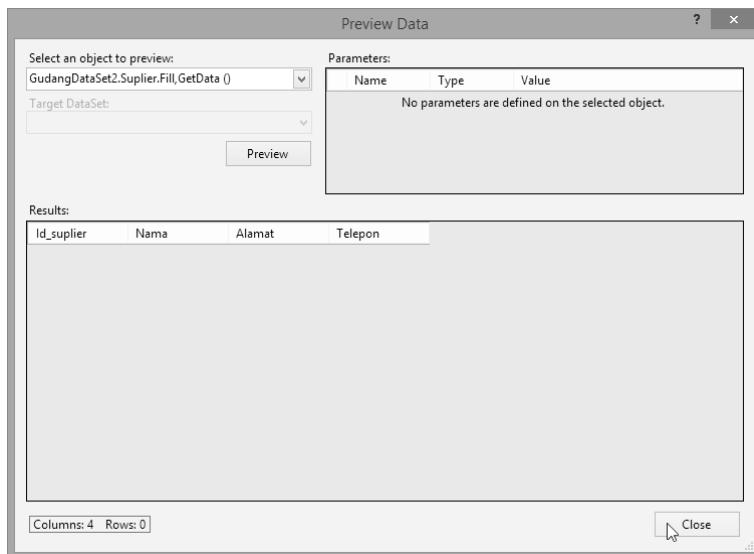
**Memilih tabel**

9. Klik segitiga kecil pada **DataGridView**, kemudian pilih **Preview Data**.



**Memilih Preview Data**

10. Klik **Preview** pada jendela **Preview Data**, kemudian pilih **Close**.



*Jendela Preview Data*

11. Tabel **Suplier**, kemudian akan ditampilkan pada **DataGridView** **FormSuplier**.



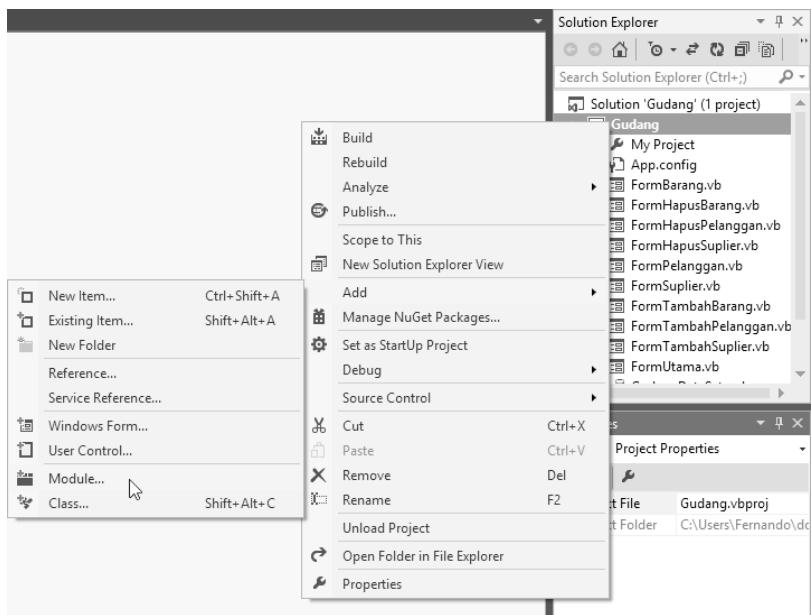
*Tabel ditampilkan dalam form aplikasi*

# Menambahkan Kode Program

Agar aplikasi ini bekerja secara keseluruhan, maka dibutuhkan script (kode program). Berikut ini langkah-langkah untuk menambahkan kode program untuk menjalankan aplikasi.

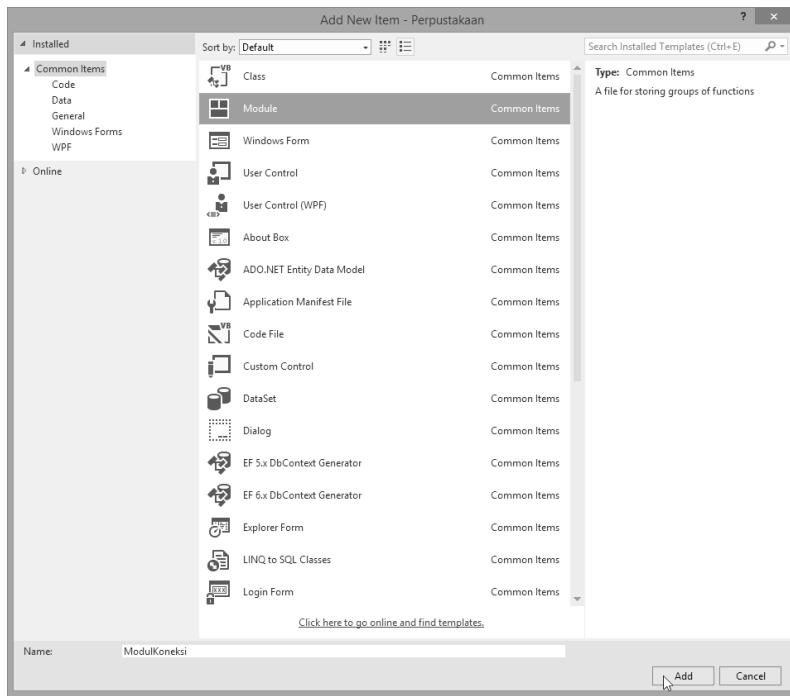
## Modul Koneksi

1. Pada **Solution Explorer** di sebelah kanan, klik kanan project **Gudang**, kemudian pilih **Add - Module**.



**Memilih Add - Module**

2. Atur nama modul menjadi **ModulKoneksi**, kemudian klik **Add**.



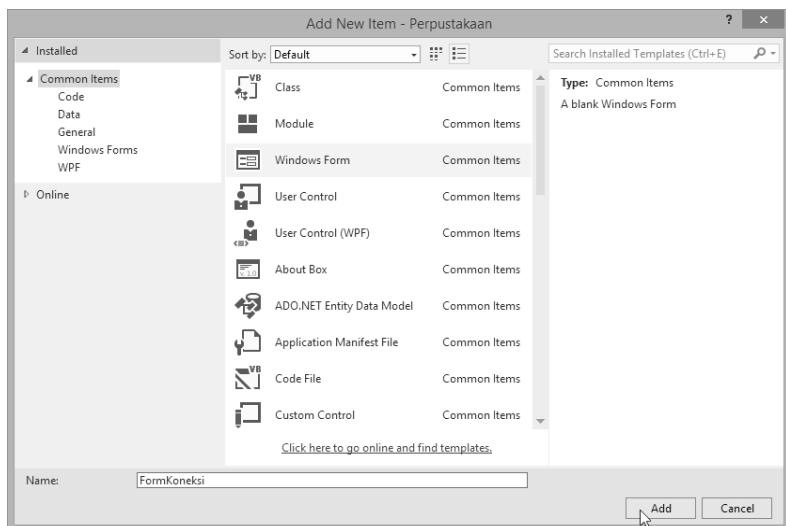
### ***Menambahkan modul***

#### **3. Tambahkan kode berikut pada modul:**

```
Imports System.Data.SqlClient  
  
Module ModulKoneksi  
  
    Public conn As SqlConnection  
  
    Public cmd As SqlCommand  
  
    Public ds As DataSet  
  
    Public adapter As SqlDataAdapter  
  
    Public rd As SqlDataReader  
  
    Public lokasidb As String  
  
    Public Sub ambilkoneksi()
```

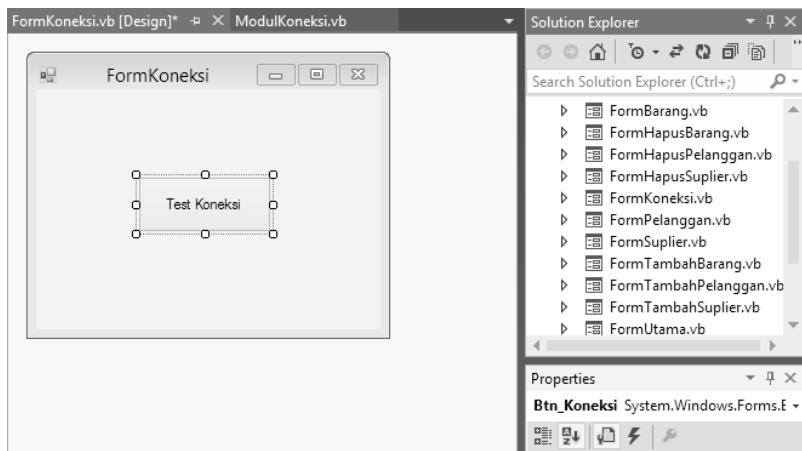
```
lokasidb = "Data Source = PC_Anda;  
Initial Catalog = nama_database;  
Persist Security Info = True;  
User ID = user_Anda;  
password = password_Anda"  
conn = New SqlConnection(lokasidb)  
  
Try  
  
If conn.State = ConnectionState.Closed Then  
conn.Open()  
  
MsgBox("Koneksi ke database berhasil",  
MsgBoxStyle.Information, "Informasi")  
  
End If  
  
Catch ex As Exception  
  
MsgBox(Err.Description, MsgBoxStyle.Critical,  
"Error")  
  
End Try  
  
End Sub  
  
End Module
```

4. Buat Form untuk mencoba modul, pilih **Project - Add Windows Form.**
5. Atur nama form menjadi **FormKoneksi**, kemudian klik **Add** untuk menambahkan form.



**Mengatur nama form**

6. Tambahkan satu button ke dalam form, kemudian atur property **Text**-nya menjadi **Test Koneksi** dan property **Name** menjadi **Btn\_Koneksi**.

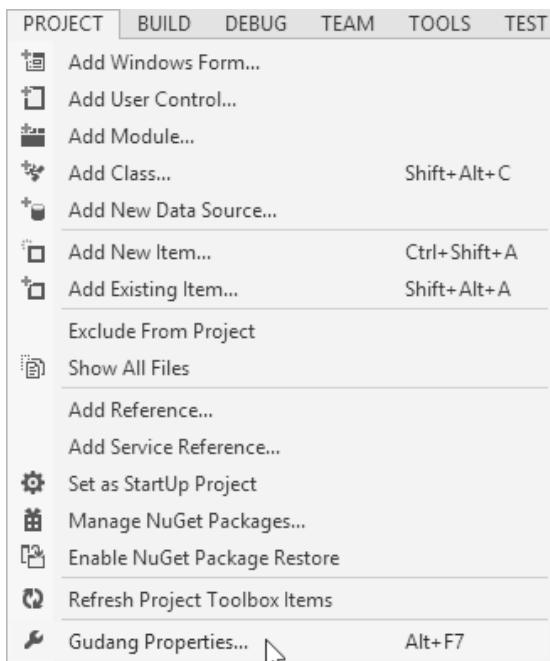


**Mengatur properti button**

7. Klik-ganda button **Test Koneksi**, kemudian masukkan kode berikut:

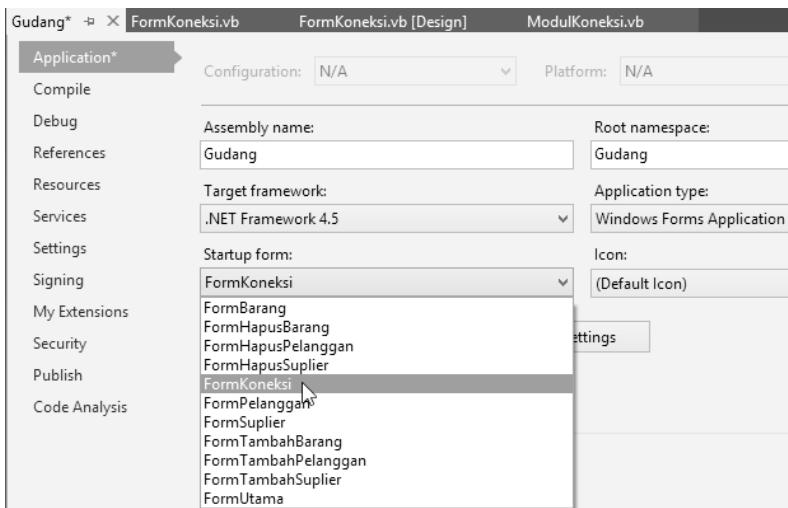
```
Imports System.Data.SqlClient  
  
Public Class FormKoneksi  
  
    Private Sub Btn_Koneksi_Click(sender As Object,  
        e As EventArgs) Handles Btn_Koneksi.Click  
  
        Call ambilkoneksi()  
  
    End Sub  
  
End Class
```

8. Untuk menjalankan **FormKoneksi**, pertama-tama Anda perlu mengaturnya sebagai form startup. Pilih **Project - Gudang Properties**.



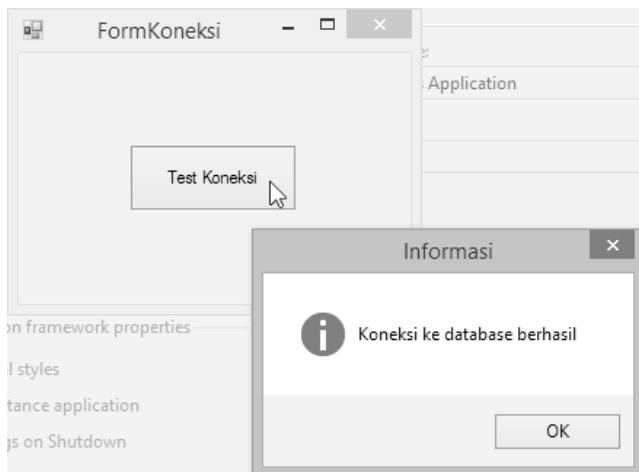
**Memilih Properties Project**

9. Pada jendela **Properties Gudang**, pilih **Startup form:** **FormKoneksi.**



*Memilih Startup form*

10. Jalankan aplikasi dengan memilih **Start**. Klik button **Test Koneksi** untuk memeriksa koneksi.

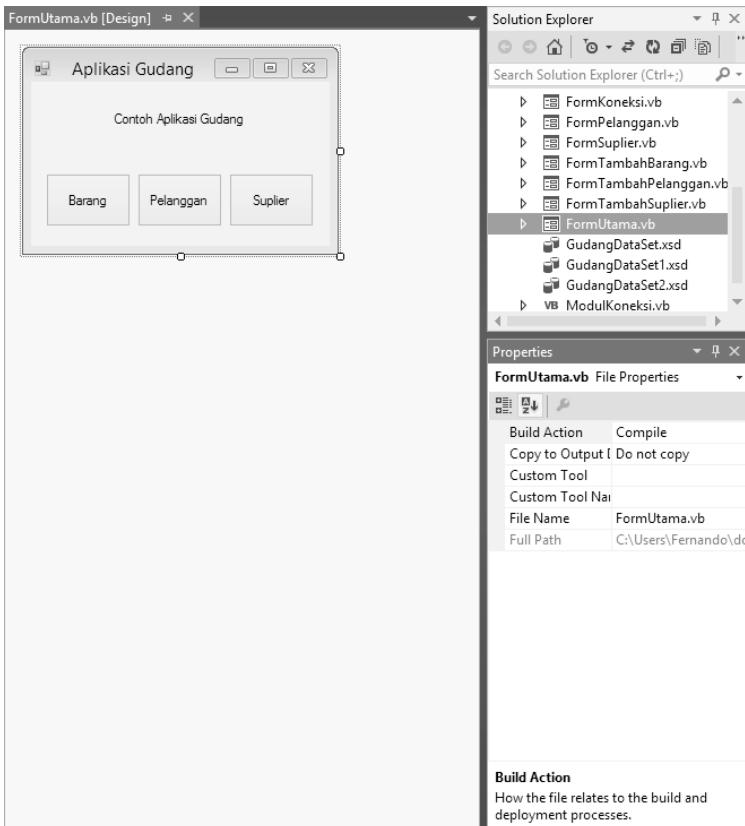


*Menjalankan aplikasi*

## Kode Form Utama

Berikut ini langkah-langkah untuk menambahkan kode program pada FormUtama:

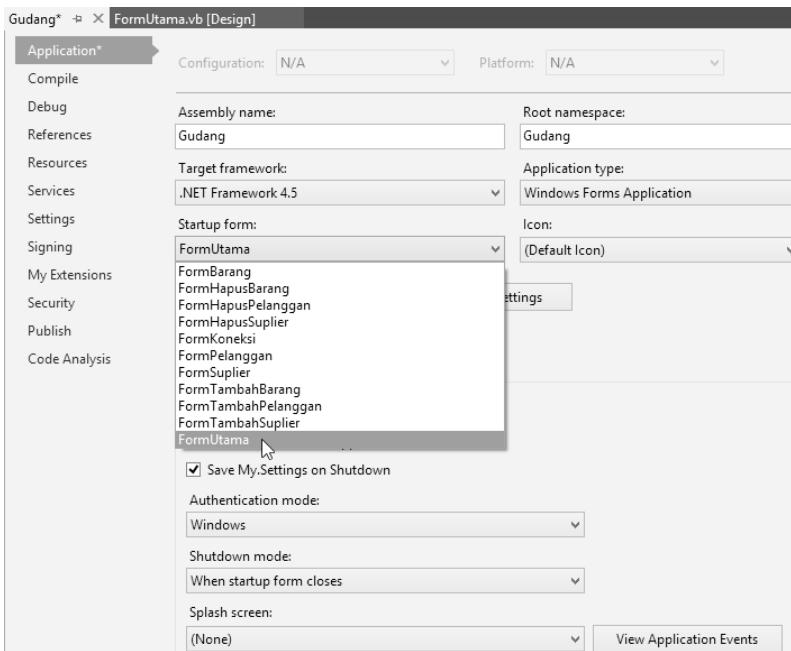
1. Buka **FormUtama** dalam project **Gudang**.



### ***Membuka Form Utama***

2. Sebelum menambahkan kode dan untuk mencoba kode yang ditambahkan dalam form, pertama-tama Anda perlu mengatur kembali form startup dengan memilih **Project - Gudang Properties**.

### 3. Atur Startup form menjadi FormUtama.



#### ***Mengatur Startup form***

### 4. Klik-ganda button **Barang**, kemudian masukkan kode berikut:

```
Private Sub Btn_Barang_Click(sender As Object,  
e As EventArgs) Handles Btn_Barang.Click  
    FormBarang.ShowDialog()  
End Sub
```

5. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Barang** untuk menampilkan **FormBarang**.

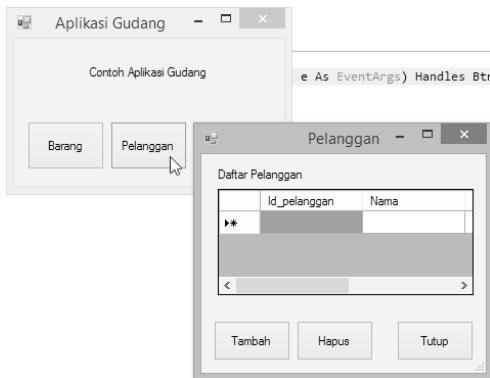


*FormBarang ditampilkan*

6. Klik-ganda button **Pelanggan**, kemudian masukkan kode berikut:

```
Private Sub Btn_Pelanggan_Click(sender As Object,
e As EventArgs) Handles Btn_Pelanggan.Click
    FormPelanggan.ShowDialog()
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Pelanggan** untuk menampilkan **FormPelanggan**.

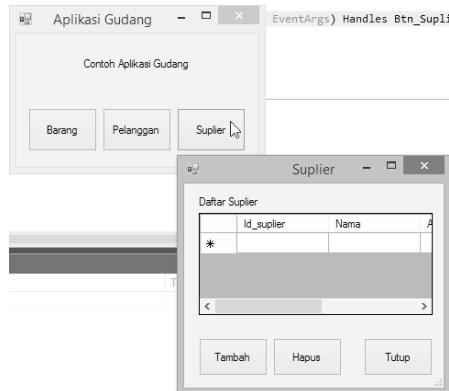


**FormPelanggan ditampilkan**

- Klik-ganda button **Suplier**, kemudian masukkan kode berikut:

```
Private Sub Btn_Suplier_Click(sender As Object,
e As EventArgs) Handles Btn_Suplier.Click
    FormSuplier.ShowDialog()
End Sub
```

- Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Suplier** untuk menampilkan **FormSuplier**.



**FormSuplier ditampilkan**

## Kode Form Barang

Berikut ini langkah-langkah untuk menambahkan kode program pada FormBarang:

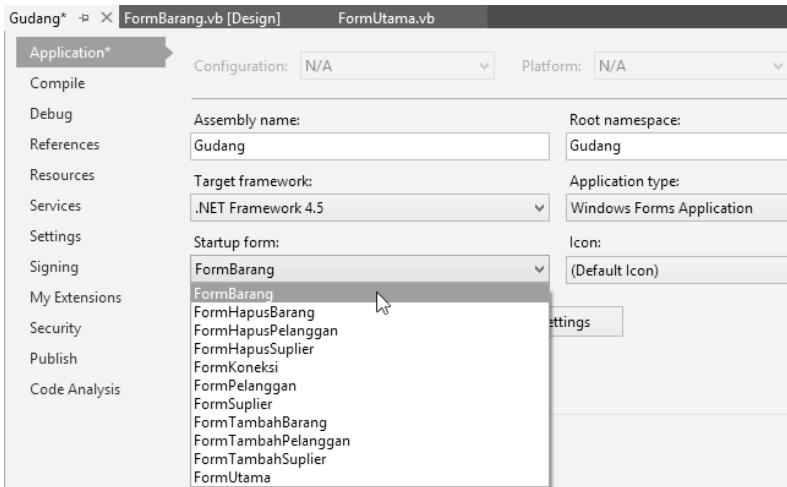
1. Buka **FormBarang** dalam project **Gudang**.



### **Membuka Form Barang**

2. Sebelum menambahkan kode dan untuk mencoba kode yang ditambahkan dalam form, pertama-tama Anda perlu mengatur kembali form startup dengan memilih **Project – Gudang Properties**.

### 3. Atur Startup form menjadi **FormBarang**.



*Mengatur Startup form*

### 4. Klik-ganda button **Tambah**, kemudian masukkan kode berikut:

```
Private Sub Btn_Tambah_Click(sender As Object,  
e As EventArgs) Handles Btn_Tambah.Click  
    FormTambahBarang.ShowDialog()  
End Sub
```

### 5. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahBarang**.

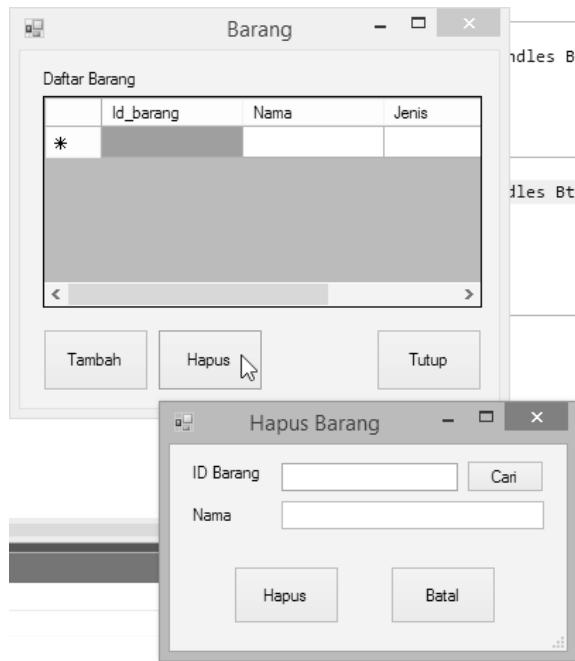


**FormTambahBarang ditampilkan**

6. Klik-ganda button **Hapus**, kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,  
e As EventArgs) Handles Btn_Hapus.Click  
FormHapusBarang.ShowDialog()  
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Hapus** untuk menampilkan **FormHapusBarang**.



***FormHapusBarang ditampilkan***

8. Klik-ganda button **Tutup** kemudian masukkan kode berikut:

```
Private Sub Btn_Tutup_Click(sender As Object,  
e As EventArgs) Handles Btn_Tutup.Click  
  
Me.Close()  
  
End Sub
```

9. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tutup** untuk menutup **FormBarang**.

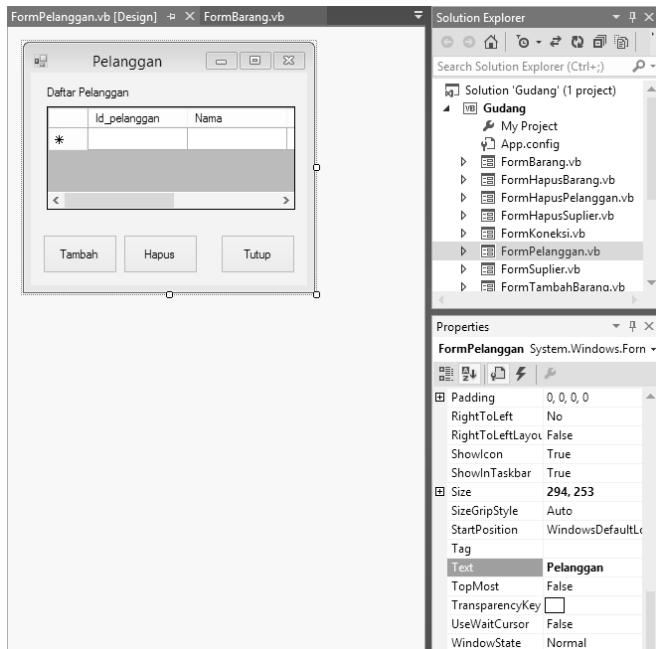


*Menutup FormBarang*

## Kode Form Pelanggan

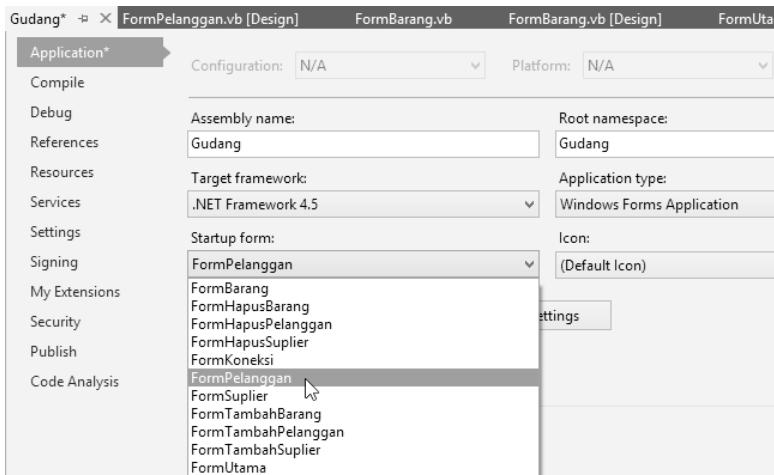
Berikut ini langkah-langkah untuk menambahkan kode program pada FormPelanggan:

1. Buka **FormPelanggan** dalam project **Gudang**.



*Membuka Form Pelanggan*

2. Anda perlu mengatur kembali form startup dengan memilih **Project - Gudang Properties**.
3. Atur **Startup form** menjadi **FormPelanggan**.

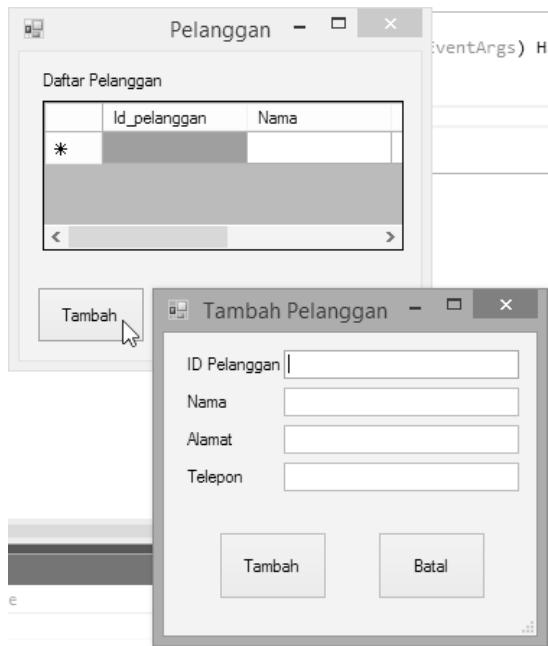


*Mengatur Startup form*

4. Klik-ganda button **Tambah** kemudian masukkan kode berikut:

```
Private Sub Btn_Tambah_Click(sender As Object,
e As EventArgs) Handles Btn_Tambah.Click
    FormTambahPelanggan.ShowDialog()
End Sub
```

5. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahPelanggan**.

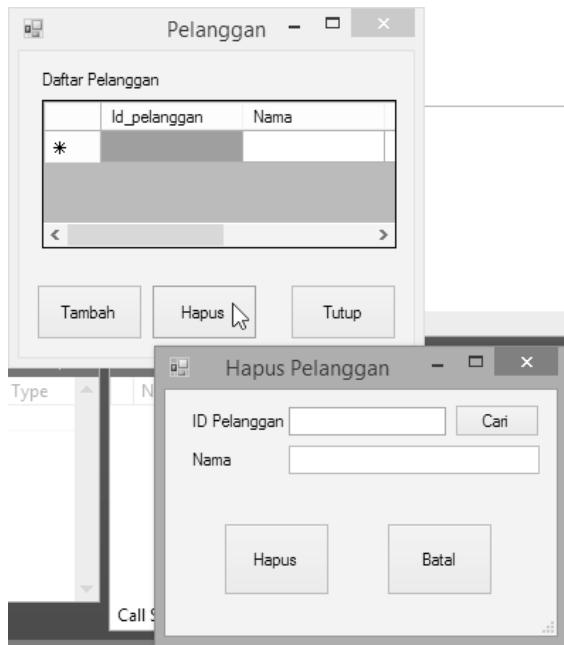


*FormTambahPelanggan ditampilkan*

6. Klik-ganda button **Hapus** kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,
e As EventArgs) Handles Btn_Hapus.Click
FormHapusPelanggan.ShowDialog()
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Hapus** untuk menampilkan **FormHapusPelanggan**.

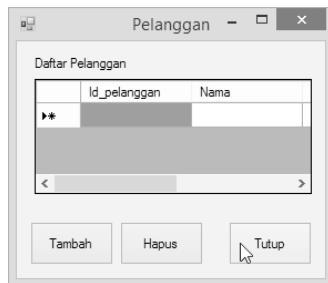


*FormHapusPelanggan ditampilkan*

8. Klik-ganda button **Tutup**, kemudian masukkan kode berikut:

```
Private Sub Btn_Tutup_Click(sender As Object,
e As EventArgs) Handles Btn_Tutup.Click
    Me.Close()
End Sub
```

9. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tutup** untuk menutup **FormPelanggan**.

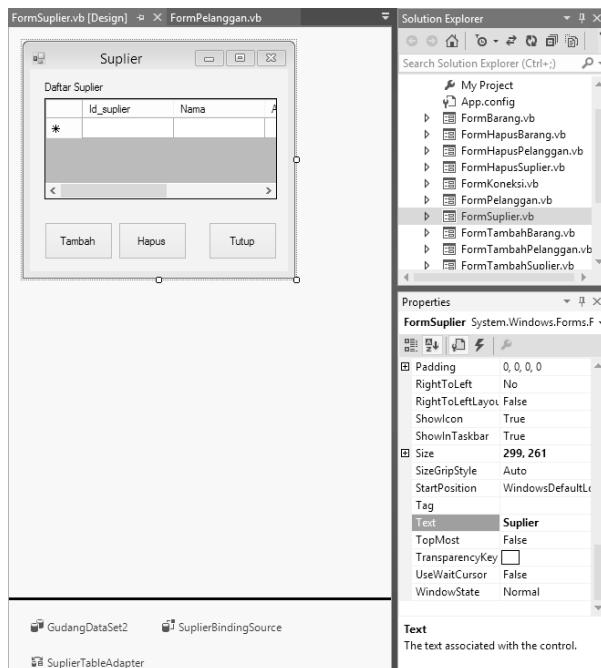


**Menutup FormPelanggan**

## Kode Form Suplier

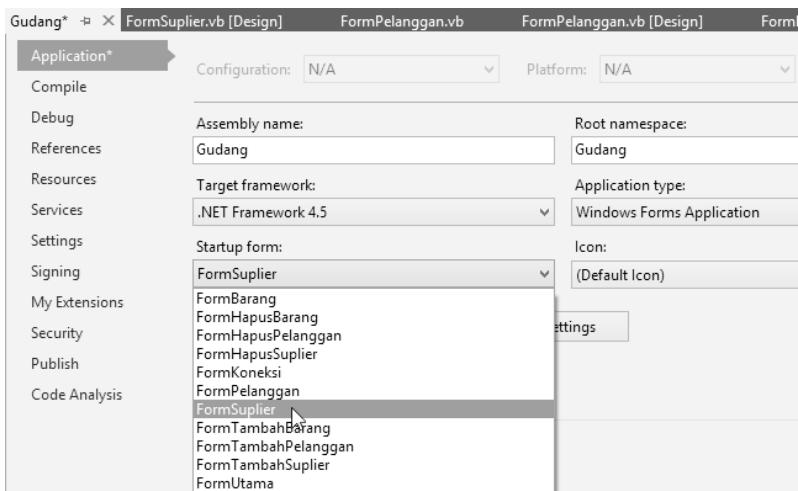
Berikut ini langkah-langkah untuk menambahkan kode program pada FormSuplier:

1. Buka **FormSuplier** dalam project Gudang.



**Membuka Form Suplier**

2. Pilih menu **Project - Gudang Properties**.
3. Atur **Startup form** menjadi **FormSuplier**.

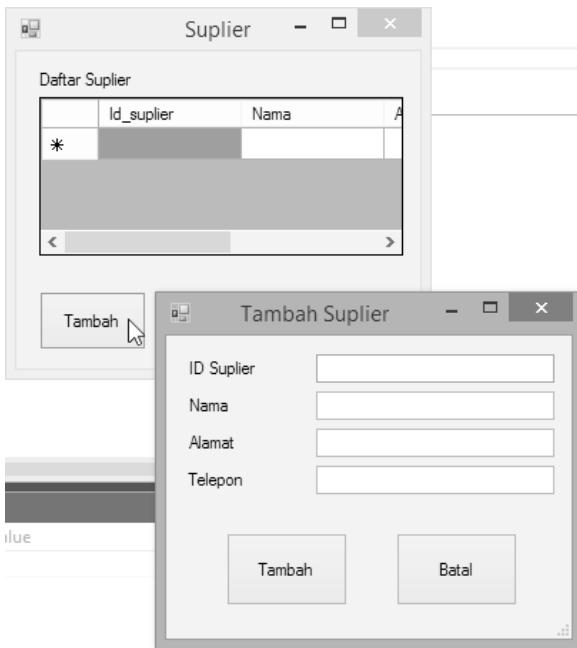


*Mengatur Startup form*

4. Klik-ganda button **Tambah**, kemudian masukkan kode berikut:

```
Private Sub Btn_Tambah_Click(sender As Object,  
e As EventArgs) Handles Btn_Tambah.Click  
    FormTambahSuplier.ShowDialog()  
End Sub
```

5. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahSuplier**.

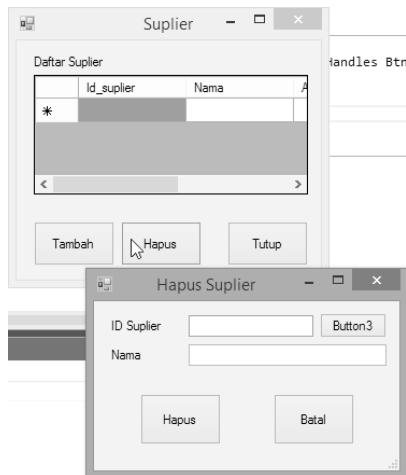


**FormTambahSuplier ditampilkan**

6. Klik-ganda button **Hapus** kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,
e As EventArgs) Handles Btn_Hapus.Click
FormHapusSuplier.ShowDialog()
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Hapus** untuk menampilkan **FormHapusSuplier**.

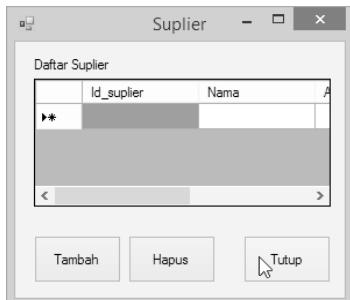


**FormHapusSuplier ditampilkan**

- Klik-ganda button **Tutup** kemudian masukkan kode berikut:

```
Private Sub Btn_Tutup_Click(sender As Object,
e As EventArgs) Handles Btn_Tutup.Click
Me.Close()
End Sub
```

- Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tutup** untuk menutup **FormSuplier**.

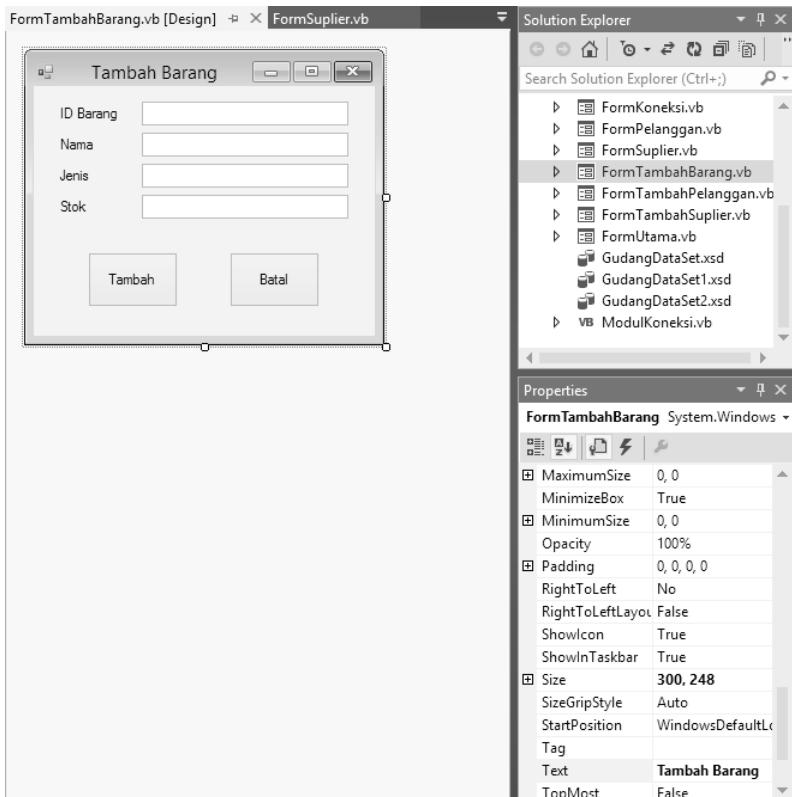


**Menutup FormSuplier**

## Kode Form Tambah Barang

Berikut ini langkah-langkah untuk menambahkan kode program pada FormTambahBarang.

1. Buka **FormTambahBarang** dalam project **Gudang**.



**Membuka Form Tambah Barang**

2. Pilih Project - **Gudang Properties**.
3. Atur **Startup form** menjadi **FormBarang**.
4. Klik-ganda **FormTambahBarang**, kemudian masukkan kode berikut:

```

Imports System.Data.SqlClient
Imports System.Xml

Public Class FormTambahBarang
    Private Sub FormTambahBarang_Load(sender As Object,
        e As EventArgs) Handles MyBase.Load
        Call ambilkoneksi()
    End Sub
End Class

```

5. Klik-ganda button **Tambah** kemudian masukkan kode berikut:

```

Private Sub Btn_Tambah_Click(sender As Object,
    e As EventArgs) Handles Btn_Tambah.Click
    If Txt_IdBarang.Text = "" Or Txt_Nama.Text = "" Or
        Txt_Jenis.Text = "" Or Txt_Stok.Text = "" Then
        MsgBox("Data belum lengkap",
            MsgBoxStyle.Information, "Informasi")
    Else
        cmd = New SqlCommand("select Id_barang from Barang
            where Id_barang = '" &
            Txt_IdBarang.Text &
            "'", conn)
        rd = cmd.ExecuteReader
        rd.Read()
        If rd.HasRows Then
            MsgBox("ID Barang Sudah digunakan",
                MsgBoxStyle.Information, "Informasi")
        End If
    End If
End Sub

```

```

rd.Close( )

Else

rd.Close( )

cmd = New SqlCommand("insert into Barang
(Id_barang, Nama, Jenis,
Stok) values ("&
Val(Txt_IdBarang.Text) &
",'"& Txt_Nama.Text &
"', '"& Txt_Jenis.Text &
"', '"& Txt_Stok.Text &
")", conn)

cmd.ExecuteNonQuery()

MsgBox("Data berhasil disimpan",
MsgBoxStyle.Information, "Informasi")

End If

End If

End Sub

```

6. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahBarang** dan menyimpan data yang dimasukkan.

The screenshot shows a Windows application window titled 'Barang'. Inside, there are two forms. The top form is titled 'Daftar Barang' and contains a table with one row. The bottom form is titled 'Tambah Barang' and has four text input fields: 'ID Barang' (2), 'Nama' (Asus Zenfone), 'Jenis' (Smartphone), and 'Stok' (20). Below these fields are two buttons: 'Tambah' and 'Batal'. A cursor is hovering over the 'Tambah' button.

### ***Menambahkan data***

7. Klik-ganda button **Batal** kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,
e As EventArgs) Handles Btn_Batal.Click
Me.Close()
End Sub
```

8. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Batal** untuk menutup **FormTambahBarang**.



*Menutup form*

## Kode Form Tambah Pelanggan

Berikut ini langkah-langkah untuk menambahkan kode program pada FormTambahPelanggan:

1. Buka **FormTambahPelanggan** dalam project **Gudang**.
2. Anda perlu mengatur kembali form startup dengan memilih **Project - Gudang Properties**.
3. Atur **Startup form** menjadi **FormPelanggan**.
4. Klik-ganda **FormTambahPelanggan** kemudian masukkan kode berikut:

```

Imports System.Data.SqlClient
Imports System.Xml

Public Class FormTambahPelanggan
    Private Sub FormTambahPelanggan_Load(
        sender As Object, e As EventArgs)
        Handles MyBase.Load
        Call ambilkoneksi()
    End Sub
End Class

```

5. Klik-ganda button **Tambah** kemudian masukkan kode berikut:

```

Private Sub Btn_Tambah_Click(sender As Object,
    e As EventArgs) Handles Btn_Tambah.Click
    If Txt_IdPelanggan.Text = "" Or Txt_Nama.Text = "" Or
        Txt_Alamat.Text = "" Or Txt_Telepon.Text = "" Then
        MsgBox("Data belum lengkap",
            MsgBoxStyle.Information, "Informasi")
    Else
        cmd = New SqlCommand("select Id_pelanggan
            from pelanggan
            where Id_pelanggan ='" &
            Txt_IdPelanggan.Text & "'",
            conn)
        rd = cmd.ExecuteReader
        rd.Read()
    End If
End Sub

```

```

If rd.HasRows Then

MsgBox( "ID Pelanggan Sudah digunakan" ,
MsgBoxStyle.Information, "Informasi" )

rd.Close()

Else

rd.Close()

cmd = New SqlCommand( "insert into Pelanggan
(Id_pelanggan, Nama,
Alamat, Telepon)

values ('" &
Val(Txt_IdPelanggan.Text)

&"', '" & Txt_Nama.Text &
" ', '" & Txt_Alamat.Text &
" ', '" & Txt_Telepon.Text

&"') ", conn)

cmd.ExecuteNonQuery()

MsgBox( "Data berhasil disimpan" ,
MsgBoxStyle.Information, "Informasi" )

End If

End If

```

6. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahPelanggan** dan menyimpan data yang dimasukkan.

### ***Menambahkan data***

7. Klik-ganda button **Batal**, kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,
e As EventArgs) Handles Btn_Batal.Click
Me.Close()
End Sub
```

8. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Batal** untuk menutup **FormTambahPelanggan**.



*Menutup form*

## Kode Form Tambah Suplier

Berikut ini langkah-langkah untuk menambahkan kode program pada FormTambahSuplier:

1. Buka Form**TambahSuplier** dalam project **Gudang**.
2. Klik menu**Project – Gudang Properties**.
3. Atur **Startup form** menjadi **FormSuplier**.
4. Klik-ganda **FormTambahSuplier**, kemudian masukkan kode berikut:

```
Imports System.Data.SqlClient
Imports System.Xml
```

```
Public Class FormTambahSuplier  
    Private Sub FormTambahSuplier_Load(  
        sender As Object, e As EventArgs)  
        Handles MyBase.Load  
        Call ambilkoneksi()  
    End Sub  
End Class
```

5. Klik-ganda button **Tambah** kemudian masukkan kode berikut:

```
Private Sub Btn_Tambah_Click(sender As Object,  
e As EventArgs) Handles Btn_Tambah.Click  
If Txt_IdSuplier.Text = "" Or Txt_Nama.Text = "" Or  
Txt_Alamat.Text = "" Or Txt_Telepon.Text = "" Then  
    MsgBox("Data belum lengkap",  
    MsgBoxStyle.Information, "Informasi")  
Else  
    cmd = New SqlCommand("select Id_suplier  
from Suplier  
where Id_suplier ='"&  
    Txt_IdSuplier.Text &"'",  
    conn)  
    rd = cmd.ExecuteReader  
    rd.Read()  
    If rd.HasRows Then
```

```

MsgBox("ID Suplier Sudah digunakan",
MsgBoxStyle.Information, "Informasi")

rd.Close()

Else

rd.Close()

cmd = New SqlCommand("insert into Suplier
(Id_suplier, Nama,
Alamat, Telepon)

values ('"&
Val(Txt_IdSuplier.Text)

&" ,'"& Txt_Nama.Text &
" ', '"& Txt_Alamat.Text

&" ', '"&
Txt_Telepon.Text &" ')",
conn)

cmd.ExecuteNonQuery()

MsgBox("Data berhasil disimpan",
MsgBoxStyle.Information, "Informasi")

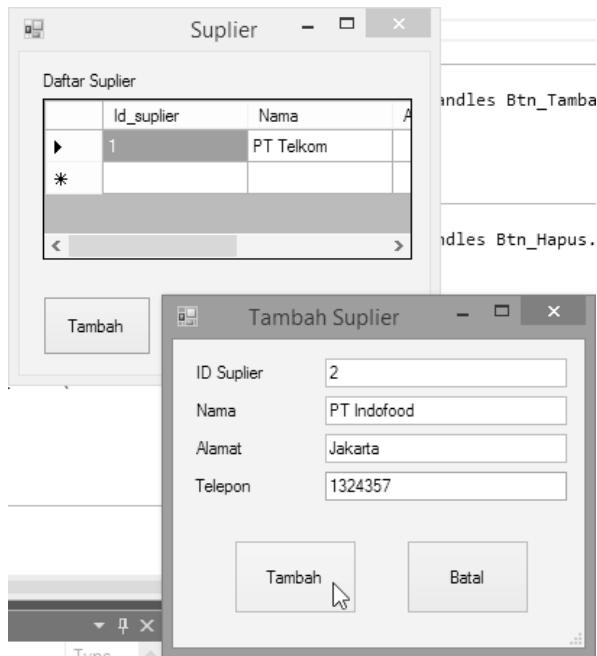
End If

End If

End Sub

```

6. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Tambah** untuk menampilkan **FormTambahSuplier** dan menyimpan data yang dimasukkan.



### **Menambahkan data**

7. Klik-ganda button **Batal** kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,  
e As EventArgs) Handles Btn_Batal.Click  
Me.Close()  
End Sub
```

8. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button Batal untuk menutup FormTambahSuplier.



*Menutup form*

## Kode Form Hapus Barang

Berikut ini langkah-langkah untuk menambahkan kode program pada FormHapusBarang:

1. Buka **FormHapusBarang** dalam project **Gudang**.
2. Pilih **Project - Gudang Properties**.
3. Atur **Startup form** menjadi **FormBarang**.
4. Klik-ganda **FormHapusBarang**, kemudian masukkan kode berikut:

```

Imports System.Data.SqlClient
Imports System.Xml
Public Class FormHapusBarang
    Private Sub FormHapusBarang_Load(sender As Object,
        e As EventArgs) Handles MyBase.Load

```

```
Call ambilkoneksi()
```

```
End Sub
```

```
End Class
```

5. Klik-ganda button **Cari**, kemudian masukkan kode berikut:

```
Private Sub Btn_Cari_Click(sender As Object,  
e As EventArgs) Handles Btn_Cari.Click  
  
If Txt_IdBarang.Text = "" Then  
  
MsgBox("Silahkan masukkan ID Barang",  
MsgBoxStyle.Information, "Informasi")  
  
Else  
  
cmd = New SqlCommand("select Nama from Barang  
where Id_barang = '" &  
Txt_IdBarang.Text & "'",  
conn)  
  
rd = cmd.ExecuteReader  
  
rd.Read()  
  
If rd.HasRows Then  
  
Txt_Nama.Text = rd.Item(0)  
  
rd.Close()  
  
Else  
  
rd.Close()  
  
MsgBox("Barang tidak tersedia",  
MsgBoxStyle.Information, "Informasi")  
  
End If  
  
End If  
  
End Sub
```

6. Klik-ganda button **Hapus**, kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,
e As EventArgs) Handles Btn_Hapus.Click
If Txt_IdBarang.Text = "" Then
    MsgBox("Silahkan masukkan ID Barang",
    MsgBoxStyle.Information, "Informasi")
Else
    If MsgBox("Apakah anda akan menghapus data ini?", 
    MsgBoxStyle.YesNo, "Informasi") =
        MsgBoxResult.Yes Then
        cmd = New SqlCommand("delete from Barang
        where Id_barang = '" &
        Txt_IdBarang.Text &
        "'", conn)
        cmd.ExecuteNonQuery()
        MsgBox("Data berhasil dihapus",
        MsgBoxStyle.Information, "Informasi")
    End If
End If
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Hapus** untuk menampilkan **FormHapusBarang** dan menghapus data.

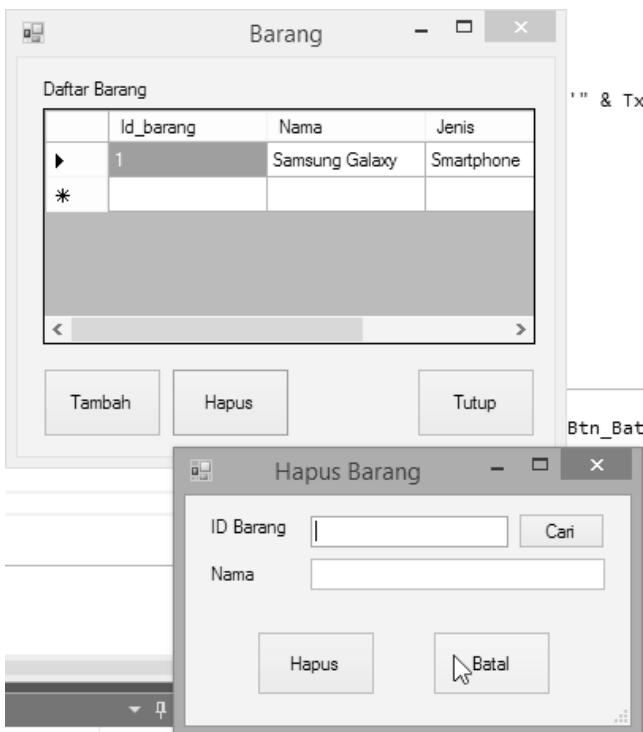


### *Menghapus data*

8. Klik-ganda button **Batal**, kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,
e As EventArgs) Handles Btn_Batal.Click
Me.Close()
End Sub
```

9. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Batal** untuk menutup **FormHapusBarang**.

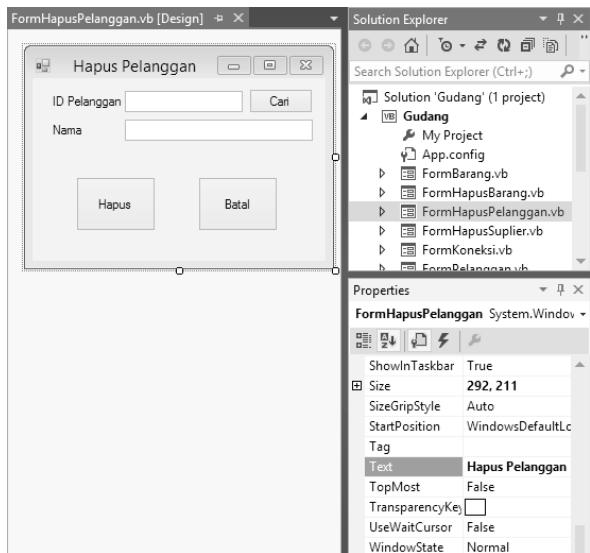


*Menutup form*

## Kode Form Hapus Pelanggan

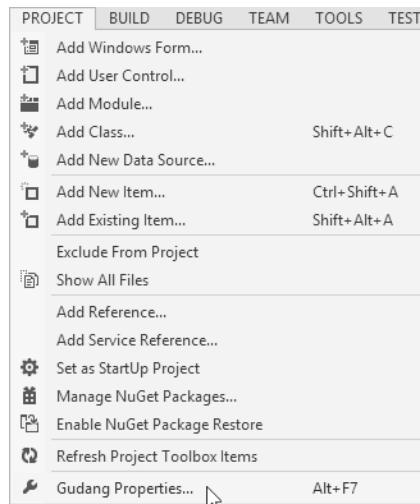
Berikut ini langkah-langkah untuk menambahkan kode program pada FormHapusPelanggan:

1. Buka FormHapusPelanggan dalam project Gudang.



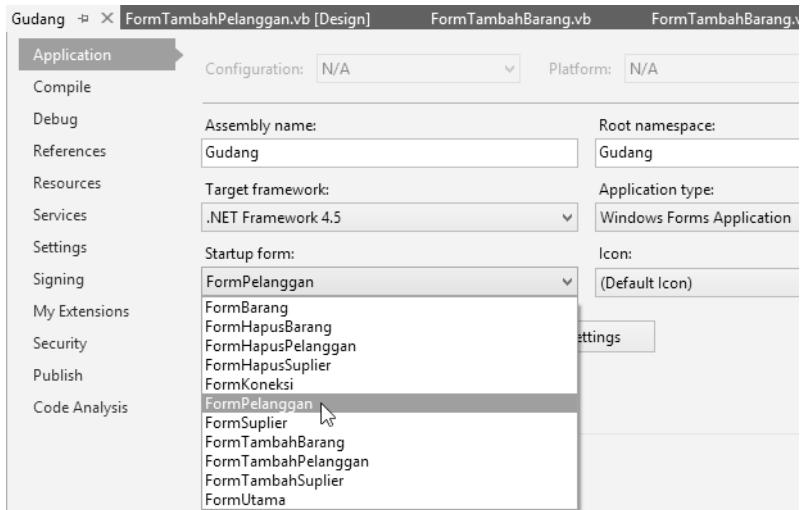
### ***Membuka Form Hapus Pelanggan***

2. Sebelum menambahkan kode dan untuk mencoba kode yang ditambahkan dalam form, pertama-tama Anda perlu mengatur kembali form startup dengan pilih Project – Gudang Properties.



### ***Mengatur Properties Project***

3. Atur Startup form menjadi FormPelanggan.



*Mengatur Startup form*

4. Klik dua kali **FormHapusPelanggan**, kemudian masukkan kode berikut:

```
Imports System.Data.SqlClient  
  
Imports System.Xml  
  
Public Class FormHapusPelanggan  
  
Private Sub FormHapusPelanggan_Load(sender As Object,  
e As EventArgs) Handles MyBase.Load  
  
Call ambilkoneksi()  
  
End Sub  
  
End Class
```

5. Klik dua kali button **Cari**, kemudian masukkan kode berikut:

```
Private Sub Btn_Cari_Click(sender As Object,
e As EventArgs) Handles Btn_Cari.Click

If Txt_IdPelanggan.Text = "" Then

MsgBox("Silahkan masukkan ID Pelanggan",
MessageBoxStyle.Information, "Informasi")

Else

cmd = New SqlCommand("select Nama from Pelanggan
where Id_pelanggan ='" &
Txt_IdPelanggan.Text &

"', conn)

rd = cmd.ExecuteReader

rd.Read()

If rd.HasRows Then

Txt_Nama.Text = rd.Item(0)

rd.Close()

Else

rd.Close()

MsgBox("Pelanggan tidak tersedia",
MessageBoxStyle.Information, "Informasi")

End If

End If

End Sub
```

6. Klik dua kali button **Hapus**, kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,
e As EventArgs) Handles Btn_Hapus.Click
If Txt_IdPelanggan.Text = "" Then
    MsgBox("Silahkan masukkan ID Pelanggan",
    MsgBoxStyle.Information, "Informasi")
Else
    If MsgBox("Apakah anda akan menghapus data ini?", 
    MsgBoxStyle.YesNo, "Informasi") =
        MsgBoxResult.Yes Then
        cmd = New SqlCommand("delete from Pelanggan
        where Id_pelanggan = '" &
        Txt_IdPelanggan.Text &
        "'", conn)
        cmd.ExecuteNonQuery()
        MsgBox("Data berhasil dihapus",
        MsgBoxStyle.Information, "Informasi")
    End If
End If
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button Hapus untuk menampilkan FormHapusPelanggan dan menghapus data.

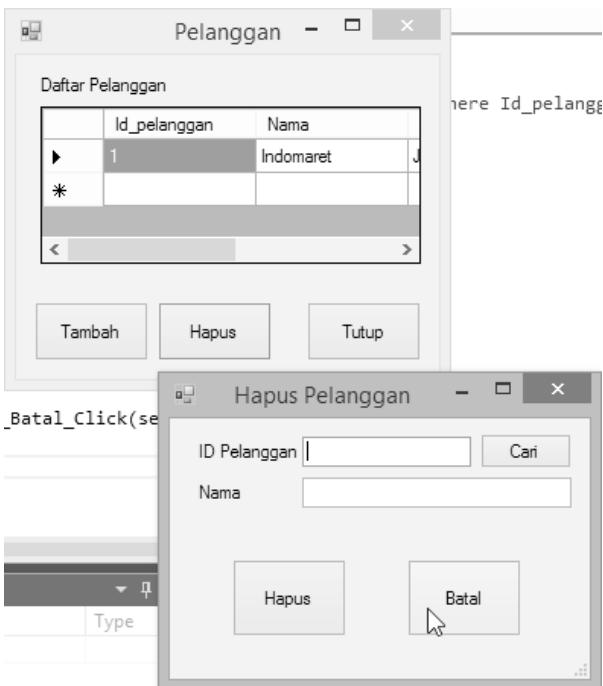


### **Menghapus data**

- Klik dua kali button **Batal**, kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,
e As EventArgs) Handles Btn_Batal.Click
    Me.Close()
End Sub
```

9. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button Batal untuk menutup FormHapusPelanggan.

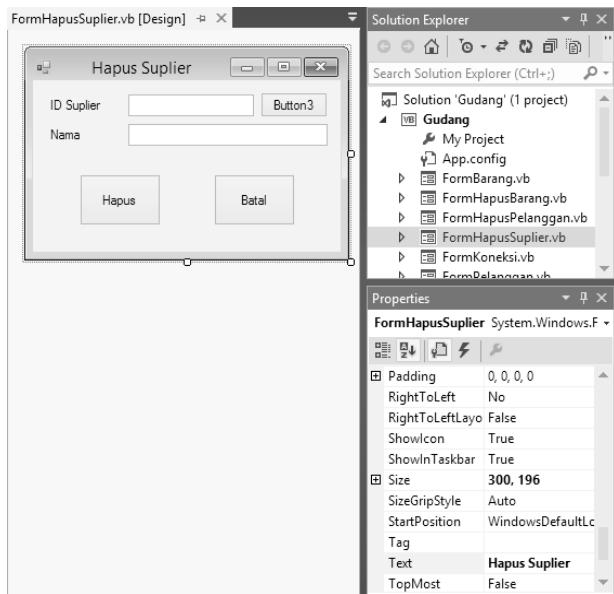


*Menutup form*

## Kode Form Hapus Suplier

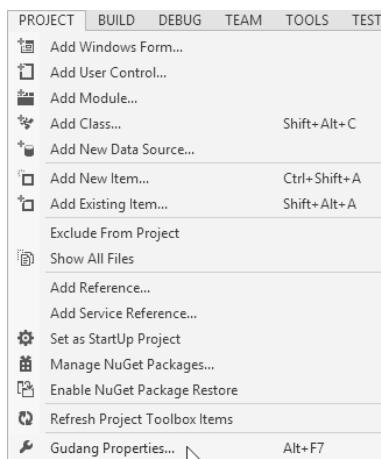
Berikut ini langkah-langkah untuk menambahkan kode program pada FormHapusSuplier:

1. Buka FormHapusSuplier dalam project Gudang.



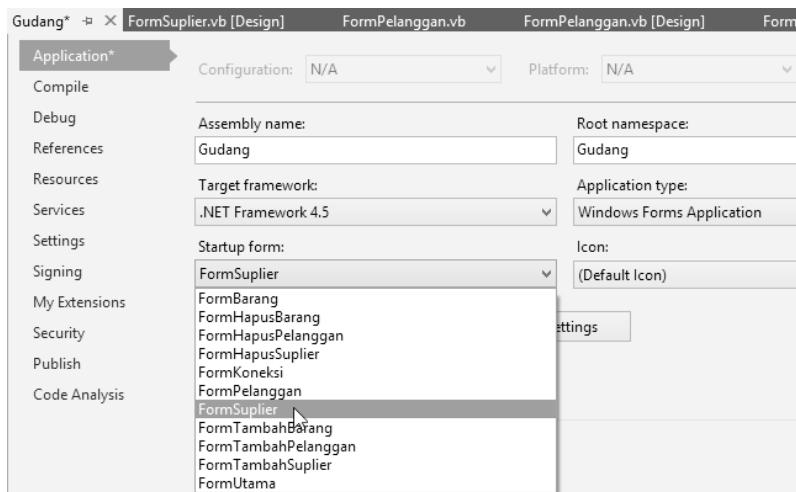
### ***Membuka Form Hapus Suplier***

2. Sebelum menambahkan kode dan untuk mencoba kode yang ditambahkan dalam form, pertama-tama Anda perlu mengatur kembali form startup dengan memilih **Project - Gudang Properties**.



### ***Mengatur Properties Project***

### 3. Atur Startup form menjadi FormSuplier.



*Mengatur Startup form*

### 4. Klik-ganda **FormHapusSuplier**, kemudian masukkan kode berikut:

```
Imports System.Data.SqlClient  
  
Imports System.Xml  
  
Public Class FormHapusSuplier  
  
    Private Sub FormHapusSuplier_Load(sender As Object,  
        e As EventArgs) Handles MyBase.Load  
  
        Call ambilkoneksi()  
  
    End Sub  
  
End Class
```

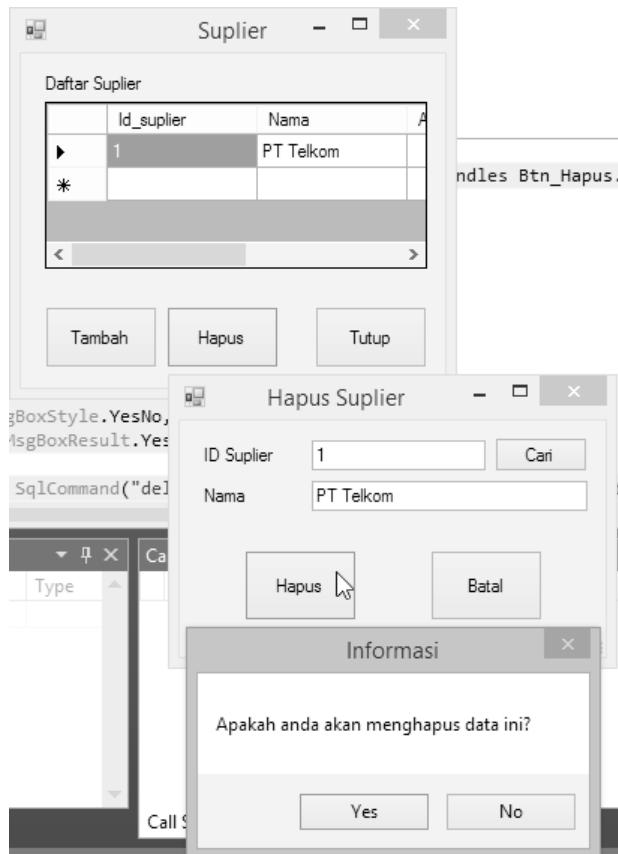
5. Klik-ganda button **Cari**, kemudian masukkan kode berikut:

```
Private Sub Btn_Cari_Click(sender As Object,
e As EventArgs) Handles Btn_Cari.Click
If Txt_IdSuplier.Text = "" Then
    MsgBox("Silahkan masukkan ID Suplier",
    MsgBoxStyle.Information, "Informasi")
Else
    cmd = New SqlCommand("select Nama from Suplier
where Id_suplier = '" &
    Txt_IdSuplier.Text & "'",
    conn)
    rd = cmd.ExecuteReader
    rd.Read()
    If rd.HasRows Then
        Txt_Nama.Text = rd.Item(0)
        rd.Close()
    Else
        rd.Close()
        MsgBox("Suplier tidak tersedia",
        MsgBoxStyle.Information, "Informasi")
    End If
End If
End Sub
```

6. Klik-ganda button **Hapus**, kemudian masukkan kode berikut:

```
Private Sub Btn_Hapus_Click(sender As Object,
e As EventArgs) Handles Btn_Hapus.Click
If Txt_IdSuplier.Text = "" Then
    MsgBox("Silahkan masukkan ID Suplier",
    MsgBoxStyle.Information, "Informasi")
Else
    If MsgBox("Apakah anda akan menghapus data ini?", 
    MsgBoxStyle.YesNo, "Informasi") =
        MsgBoxResult.Yes Then
        cmd = New SqlCommand("delete from Suplier
        where Id_suplier = '" &
        Txt_IdSuplier.Text &
        "'", conn)
        cmd.ExecuteNonQuery()
        MsgBox("Data berhasil dihapus",
        MsgBoxStyle.Information, "Informasi")
    End If
End If
End Sub
```

7. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Hapus** untuk menampilkan **FormHapusSuplier** dan menghapus data.

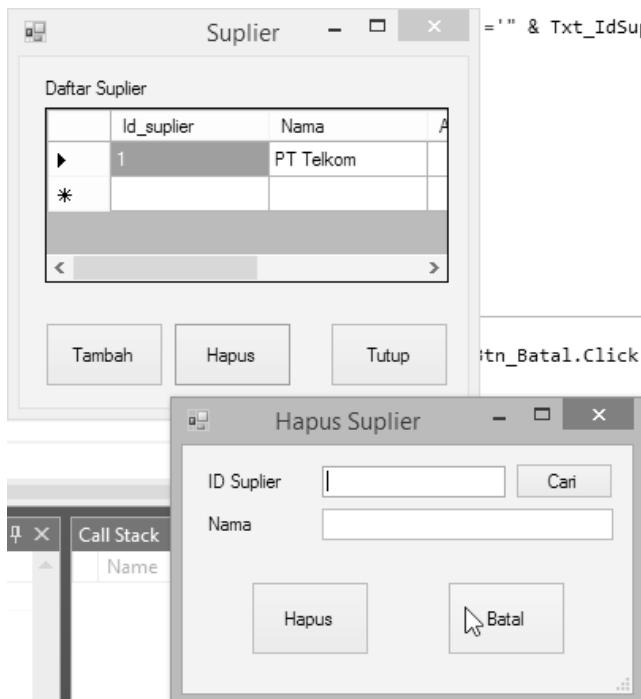


### **Menghapus data**

8. Klik-ganda button **Batal**, kemudian masukkan kode berikut:

```
Private Sub Btn_Batal_Click(sender As Object,  
e As EventArgs) Handles Btn_Batal.Click  
    Me.Close()  
End Sub
```

9. Jalankan aplikasi dengan memilih **Start**, kemudian klik Button **Batal** untuk menutup **FormHapusSuplier**.



*Menutup form*

# Tentang Penulis

## JUBILEE ENTERPRISE

Jubilee Enterprise adalah "a Creative Media Content Provider" dengan misi "Mengeksplorasi Teknologi Informasi tercanggih di dunia dan menyajikannya dalam bentuk media dengan gaya bahasa yang sederhana, mudah dicerna, dan gampang dipraktikkan oleh siapa pun".

Di Jubilee Enterprise, "Information Technology is our passion". Itulah mengapa setiap hari kami mengeksplorasi, meneliti, dan bereksperimen dengan banyak teknologi tercanggih saat ini. Hasil penelitian tersebut kami persembahkan dalam bentuk media cetak (buku) dan elektronik (blog).

Buku-buku kami, yang diterbitkan oleh PT Elex Media Komputindo (Kelompok Kompas Gramedia), telah didistribusikan ke seluruh Indonesia dan Malaysia, membantu dan menginspirasi pembaca-pembaca kami ketika menggunakan program Photoshop, CorelDraw, MS Office, Internet, Gagdet, dan lain sebagainya secara mudah dan praktis.

### **Catatan:**

Untuk melakukan pemesanan buku, hubungi  
Layanan Langsung PT Elex Media Komputindo:

### **Gramedia Direct**

Jl. Palmerah Barat No. 33, Jakarta 10270  
Telemarketing/CS: 021-53650110/111 ext: 3901/3902  
Email: **endang@gramediapublishers.com**



# Buku Latihan

# Visual Basic

## untuk Mahasiswa

Visual Basic memiliki daya tarik jika dilihat dari sisi kemudahan dalam pemrogramannya. Anda dapat membuat aplikasi hitung-hitungan sederhana sampai aplikasi yang melibatkan banyak form dan koneksi database. Tool yang digunakan adalah Visual Studio Express 2013 (gratis dan bisa diunduh langsung dari website Microsoft).

Ditujukan untuk semua pembaca, terutama mahasiswa yang ingin mencari pengetahuan tentang pemrograman Visual Basic. Dengan mempelajari Visual Basic, diharapkan para mahasiswa dapat mengerjakan tugas akhir pembuatan aplikasi yang melibatkan bahasa pemrograman tersebut. Materi meliputi pengenalan Visual Basic, database, dan pembuatan aplikasi-aplikasi sederhana.

Sebagai bahan latihan, Anda akan belajar membuat aplikasi “Gudang” yang memiliki fitur-fitur:

- **Terkoneksi dengan database.**
- **Form untuk menambah dan menghapus nama pelanggan, nama supplier, dan data-data produk/barang.**

gramedia

Penerbit PT Elex Media Komputindo  
Kompas Gramedia Building  
Jl Palmerah Barat 29-37 Jakarta 10270  
Telp. (021) 53650110, 53650111 ext. 3214  
Web Page: <http://www.elexmedia.co.id>

Kelompok
Pemrograman
Keterampilan
<input checked="" type="checkbox"/> Tingkat Pemula
<input checked="" type="checkbox"/> Tingkat Menengah
<input type="checkbox"/> Tingkat Mahir
Jenis Buku
<input checked="" type="checkbox"/> Referensi
<input checked="" type="checkbox"/> Tutorial
<input checked="" type="checkbox"/> Latihan

ISBN 978-602-02-6320-5



9 78602 0263205

121150784