



TEKNOLOGI PENGELASAN

# PEMROGRAMAN DASAR MENGUNAKAN VISUAL BASIC NET 2013

Yunianita Rachmawati, S.Kom., M.Kom  
Uce Indahyanti, M.Kom



BUKU AJAR  
UNIVERSITAS MUHAMMADIYAH SIDOARJO

# PEMROGRAMAN DASAR MENGUNAKAN VISUAL BASIC NET 2013

Yunianita Rachmawati, S.Kom., M.Kom  
Uce Indahyanti, M.Kom

ISBN 978-623-6833-91-9 (PDF)



BUKU AJAR  
UNIVERSITAS MUHAMMADIYAH SIDOARJO

**BUKU AJAR  
PEMROGRAMAN DASAR  
MENGUNAKAN VISUAL BASIC .NET  
2013**

Oleh :  
Yunianita Rahmawati, M.Kom.  
Uce Indahyanti, M.Kom.



Diterbitkan Oleh : UMSIDA Press

**UNIVERSITAS MUHAMMADIYAH SIDOARJO  
2020**

**BUKU AJAR  
PEMROGRAMAN DASAR MENGGUNAKAN VISUAL  
BASIC .NET 2013**

**Penulis:**

Yunianita Rahmawati, S.Kom., M.Kom.

Uce Indahyanti, M.Kom

**ISBN :**

978-623-6833-91-9

**Editor:**

Rohman Dijaya, S.Kom., M.Kom

**Design Sampul dan Tata Letak:**

Mochammad Nashrullah, S.Pd.

Amy Yoga Prajati, S.Kom.

**Penerbit :**

UMSIDA Press

Anggota IKAPI No. 218/ANggota Luar Biasa/JTI/2019

Anggota APPTI No. 002 018 1 09 2017

**Redaksi**

Universitas Muhammadiyah Sidoarjo

Jl. Mojopahit No 666B

Sidoarjo, Jawa Timur

Cetakan Pertama, September 2020

©Hak Cipta dilindungi undang undang

Dilarang memperbanyak karya tulis ini dengan sengaja, tanpa ijin  
tertulis dari penerbit.

## KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Allah SWT, atas rahmat dan karunia-Nya Buku Ajar Berpikir Analisis melalui Fluida dapat diselesaikan dengan baik dan tanpa halangan yang berarti. Shalawat dan salam selalu kami sampaikan kepada junjungan Nabi Muhammad SAW.

Tim penulis mengucapkan terimakasih kepada:

1. Sri Mukhodim Faridah Hanum., SST., MM., M.Kes., Dekan Fakultas Ilmu Kesehatan yang memberikan arahan dan motivasi kepada penulis dalam menyelesaikan buku ajar ini.
2. Cholifah.,S.ST., M. Kes, Kaprodi Manajemen Informasi Kesehatan yang telah memberikan dukungan untuk menyusun buku ajar ini.
3. Rekan-rekan dosen pengampu Mata Kuliah Pemrograman Dasar di prodi Manajemen Informasi Kesehatan yang telah berbagi pengalaman dalam mengampu mata kuliah tersebut.

Saran dan kritik sangat penulis harapkan untuk mewujudkan buku ajar Pemrograman Dasar Menggunakan Visual Basic .Net 2013 yang lebih baik dan tentunya sesuai dengan amanat peraturan yang berlaku. Terimakasih.

**Tim Penulis**

## DAFTAR ISI

### HALAMAN SAMPUL

### KATA PENGANTAR

### DAFTAR ISI

### PENDAHULUAN

### BAB I PENGENALAN VISUAL BASIC .NET

1.1. Pengantar Visual Basic .NET .....	1
1.1.1. Apa itu Visual Studio .Net?.....	1
1.1.2. Sekilas Tentang Visual Basic .Net .....	2
1.1.3. Mengenal .Net Framework .....	4
1.1.4. Sejarah Singkat Visual Basic .Net.....	9
1.1.5. Kelebihan Visual Basic .Net.....	12
1.2. Cara Instalasi dan Uninstall Visual Basic .Net 2013	13
1.2.1. Cara Instalasi Visual Studio 2013 .....	14
1.2.2. Cara Uninstalasi Visual Studio 2013 .....	16
1.3. Memulai Visual Studio .Net 2013 dan Membuat Project Baru .....	17
1.4. Berkenalan dengan Visual Basic .Net 2013 .....	19
1.4.1. IDE (Integrated Development Environment) Visual Basic .Net 2013. ....	20
1.4.2. Kode Editor Visual Studio .Net 2013 .....	39
1.4.3. Output Window dan Error List Window....	46
1.5. Menyimpan Project, Menjalankan Project, Menambahkan Form Baru, dan Membuka Project	47
1.5.1. Menyimpan Project.....	48
1.5.2. Menjalankan Project .....	49
1.5.3. Menambahkan Form Baru .....	50
1.5.4. Membuka Project.....	52

1.6. Latihan Program .....	53
1.6.1. Latihan Program 1 .....	53
1.6.2. Latihan Program 2 .....	57

## **BAB II ELEMEN-ELEMEN PROGRAM**

2.1. Pendahuluan.....	64
2.2. Variabel.....	64
2.2.1. Syarat Penulisan Variabel.....	64
2.2.2. Deklarasi Variabel.....	65
2.2.3. Isi Variabel .....	67
2.2.4. Aksesibilitas Variabel.....	68
2.2.5. Ruang Lingkup Variabel .....	69
2.3. Tipe Data.....	71
2.3.1. Tipe Data Variant.....	71
2.3.2. Tipe Data Lain.....	73
2.3.3. Tipe Data Buatan Sendiri.....	74
2.3.4. Konversi Tipe Data .....	75
2.4. Konstanta.....	77
2.4.1. Penulisan Konstanta.....	77
2.4.2. Sumber Konstanta .....	77
2.5. Operator dan Operand .....	79
2.6. Latihan Program .....	84

## **BAB III STRUKTUR PERCABANGAN**

3.1. Pengantar Struktur Percabangan .....	91
3.2. Struktur IF – THEN .....	93
3.3. Struktur IF – THEN – Else .....	96
3.4. Struktur Nested IF.....	98
3.5. Struktur Nested IF / IF Bersaran .....	100
3.6. Struktur IIF .....	103
3.7. Struktur Select Case.....	104

3.8. Latihan Program .....	107
<b>BAB IV STRUKTUR PENGULANGAN</b>	
4.1. Pengantar Struktur Pengulangan .....	111
4.2. Struktur For - Next.....	112
4.3. Struktur For Each ... Next .....	114
4.4. Struktur While ... End While .....	114
4.5. Struktur Do While ... Loop .....	115
4.6. Struktur Do Until ... Loop .....	117
4.7. Struktur Do ... Loop While .....	118
4.8. Struktur Do ... Loop Until .....	119
4.9. Latihan Program .....	120
<b>BAB V ARRAY</b>	
5.1. Deklarasi dan Pengisian Array.....	125
5.2. Array Multi Dimensi.....	125
5.3. Array Dinamis .....	128
5.4. Latihan Program .....	129
5.4.1. Latihan Program 1 .....	129
5.4.2. Latihan Program 2 .....	131
5.4.3. Latihan Program 3 .....	135
<b>BAB VI PROSEDUR DAN FUNGSI</b>	
6.1. Prosedur .....	137
6.2. Fungsi.....	140
6.3. Fungsi-Fungsi Built-In dalam VB .Net .....	142
6.3.1. Fungsi Kotak Dialog Pesan.....	143
6.3.2. Fungsi Kotak Dialog Input.....	149
6.3.3. Fungsi Identifikasi Tipe Data .....	151
6.3.4. Fungsi String .....	151
6.3.5. Fungsi Matematika.....	153
6.3.6. Fungsi Statistik.....	155

6.3.7. Fungsi Tanggal dan Waktu .....	155
6.3.8. Fungsi Format.....	158
6.3.9. Fungsi Manipulasi File dan Folder .....	160
6.4. Latihan Program .....	163

## **BAB VII PEMROGRAMAN DATABASE**

7.1. Pengertian, Jenis, dan Manfaat Database .....	173
7.2. Perintah SQL .....	176
7.3. ADO. NET .....	181
7.3.1. Net Data Provider .....	185
7.3.2. Menampilkan Data Pada Tabel .....	187
7.3.3. Metode Pencarian Data pada DataTable .....	188
7.3.4. Metode Pencarian Data pada DataView .....	190
7.4. Membuat Database Microsoft Access 2013.....	193
7.4.1. Membuat Database Baru Pada Microsoft Access 2013 .....	193
7.4.2. Membuat Tabel Baru Pada Microsoft Access 2013.....	196
7.4.3. Membuat Raltionship Antar Tabel pada Microsoft Access 2013 .....	199
7.5. Menghubungkan Visual Basic dengan Microsoft Access 2013 .....	203
7.6. Latihan Program .....	206

## **BAB VIII MEMBUAT LAPORAN DENGAN CRYSTAL REPORT**

8.1. Instalasi Crystal Report .....	214
8.2. Membuat Laporan Baru .....	214
8.3. Crystal Reports Designer .....	223
8.3.1. Reports Designer .....	224
8.3.2. Field Explorer Crystal Report.....	225
8.3.3. Toolbox Crystal Report.....	228



8.4. Menampilkan Laporan Crystal Reports Pada Form .....	229
8.5. Menampilkan Laporan Crystal Reports Pada Form dengan Menambahkan Parameter.....	230
8.6. Model Implementasi Sumber Data .....	230
<b>BAB IX MENU, MDI, PEMBUATAN SETUP, TOOLTIP</b>	
9.1. Menu .....	232
9.2. MDI .....	235
9.3. Pembuatan Setup .....	236
9.3.1. Mencoba Instalasi Program Yang Telah Dibuat .....	244
9.3.2. Uninstal Program Hasil Instalasi Program Yang Telah Dibuat .....	244
9.4. Instalasi Microsoft Visual Studio 2013 Installer	245
9.5. Pembuatan Tooltip .....	246
<b>BAB X MENANGANI KESALAHAN PROGRAM</b>	
10.1. Menggunakan On Error .....	250
10.2. Komponen Error Provider.....	255
10.3. Menggunakan Try-Catch-Finally.....	258
10.4. Menggunakan Debugger .....	262
10.5. Melakukan Proses Debug .....	263
<b>DAFTAR PUSTAKA</b>	
<b>BIODATA PENULIS</b>	

**BATANG TUBUH DAN  
SUB-CAPAIAN PEMBELAJARAN MATA KULIAH**

<b>BAB</b>	<b>Sub-Capaian Pembelajaran Mata Kuliah</b>
BAB I PENGENALAN VISUAL BASIC .NET	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami .Net Framework, sejarah visual basic .Net, kelebihan visual basic .Net, dan cara instalasi serta uninstall visual basic .Net.</li> <li>2. Mahasiswa mampu memahami lingkungan visual visual basic .net.</li> <li>3. Mahasiswa mampu membuat program visual basic .net sebagai pengenalan toolbox, properti, dan membuat kode program sederhana.</li> </ol>
BAB II ELEMEN- ELEMEN PROGRAM	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami variabel, tipe data, konstanta, serta operator dan operand.</li> <li>2. Mahasiswa mampu membuat program tentang variabel, tipe data, konstanta, serta operator dan operand.</li> </ol>
BAB III STRUKTUR PERCABANGAN	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami struktur percabangan.</li> <li>2. Mahasiswa mampu membuat program tentang struktur percabangan.</li> </ol>
BAB IV STRUKTUR PENGULANGAN	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami struktur pengulangan.</li> <li>2. Mahasiswa mampu membuat program tentang struktur pengulangan.</li> </ol>
BAB V ARRAY	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami array.</li> <li>2. Mahasiswa mampu membuat program tentang array.</li> </ol>

BAB VI PROSEDUR DAN FUNGSI	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami prosedur dan fungsi.</li> <li>2. Mahasiswa mampu membuat program tentang prosedur dan fungsi.</li> </ol>
BAB VII PEMROGRAMAN DATABASE	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami pemrograman database.</li> <li>2. Mahasiswa mampu membuat program tentang pemrograman database.</li> </ol>
BAB VIII MEMBUAT LAPORAN DENGAN CRYSTAL REPORT	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami pembuatan laporan menggunakan crystal report.</li> <li>2. Mahasiswa mampu membuat program tentang pembuatan laporan menggunakan crystal report.</li> </ol>
BAB IX MENU, MDI, PEMBUATAN SETUP, TOOLTIP	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami Menu, Mdi, Pembuatan Setup, Tooltip.</li> <li>2. Mahasiswa mampu membuat program disertai Menu, Mdi, Pembuatan Setup, Tooltip.</li> </ol>
BAB X MENANGANI KESALAHAN PROGRAM	<ol style="list-style-type: none"> <li>1. Mahasiswa mampu memahami menangani penanganan kesalahan program dalam membuat suatu aplikasi/program.</li> <li>2. Mahasiswa mampu membuat program yang disertai dengan penanganan kesalahan program agar alur program berjalan normal.</li> </ol>

## **BAB I**

### **PENGENALAN VISUAL STUDIO .NET**

Pada bab 1 ini menjelaskan tentang pengantar Visual Basic .Net, cara instalasi dan uninstall Visual Basic .Net 2013, Memulai Visual Studio .Net dan Membuat Project Baru, Berkenalan dengan Visual Basic .Net 2013, Menyimpan Project, Menjalankan Project, menambahkan Form Baru, dan Membuka Project, serta latihan dan tugas.

#### **1.1. Pengantar Visual Basic .Net**

Visual basic selain disebut sebagai bahasa pemrograman (*language program*) juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows.

**Microsoft Visual Basic .NET** adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi command-line.

##### **1.1.1. Apa itu Visual Studio .Net?**

**Visual Studio .Net** adalah salah satu bentuk *Integrated Development Environment (IDE)* terbaru dan tercanggih yang dimiliki oleh Microsoft saat ini. IDE tersebut **mampu mendukung dan menggabungkan banyak tipe bahasa**, seperti Visual Basic .Net, C# .Net, J# .Net, C++ .Net

serta bahasa .Net lainnya menjadi satu dalam lingkungan dot net framework. Hasil aplikasi program yang dibangun menggunakan Visual Studio .NET dapat dijalankan pada berbagai platform operating system seperti Linux, Unix, Solaris, Windows, dan OS lainnya dengan menggunakan mono yang dihasilkan oleh perusahaan Symbian. Berikut gambaran singkat dari uraian di atas ditampilkan pada gambar 1.1 (Yung, 2005).

Vb.Net	C#.Net	C++.Net	J#.Net	.net lainnya
Dot net Framework		Mono		
Windows	Linux	Unix	Solaris	Operatin g system lainnya

Gambar 1.1. Bagan Visual Studio .Net

### 1.1.2. Sekilas Tentang Visual Basic .Net

**Visual basic .Net** adalah salah satu bahasa pemrograman yang ada di dalam Visual Studio .Net. beberapa fitur yang ada pada Visual basic .Net adalah tampilan control, mendukung penuh OOP (*Object Oriented Programming*), fasilitas GUI (Graphic Universal Interface), koneksi database (Yuswanto, 2008).

Pada modul ini Visual Studio yang digunakan adalah Visual Studio 2013 karena alasan kecepatan, produk Visual Studio 2013 termasuk produk keluaran

terbaru, dan fitur pada Visual Studio 2013 sudah mencukupi untuk menyelesaikan kasus-kasus pada modul ini. Menurut Microsoft (2018), untuk **spesifikasi komputer** yang digunakan untuk Visual Studio 2013 yaitu :

- 1.6 GHz atau prosesor
- 1,5 GB RAM
- 5 GB hard disk yang tersedia
- Hard drive 5400 RPM
- Video card berkemampuan DirectX 9 berjalan pada resolusi layar 1024 x 768 atau lebih tinggi

Menurut Yuswanto (2006), **Visual Studio .Net terdiri dari empat edisi** yang dimulai dari edisi yang paling dasar sampai edisi yang paling lengkap, diantaranya adalah :

- **Visual Studio .Net Professional**

Pada edisi ini terdapat beberapa fitur yaitu **Visual Basic .Net, C++ .Net, C# .Net, J# .Net**, dan layanan web **XML** (*Extensible Markup Language*). Layanan XML ini digunakan untuk membangun aplikasi web, windows, mobile menggunakan WML pada telepon seluler berbasis WAP dan HTML untuk PC, palm, dan peger. Selain itu disertakan Microsoft SQL Server Desktop Engine yang kompatibel dengan database SQL Server.

- **Visual Studio .Net Academic**

Fitur yang ada pada edisi ini sama dengan fitur yang ada pada edisi professional. Pada edisi ini terdapat fitur

tambahan yaitu **fitur instruksional yang didesain untuk menyederhanakan pengelola mata kuliah atau tugas-tugas mahasiswa yang dapat diakses melalui Server Web atau situs FTP (*File Transfer Protocol*)**.

- **Visual Studio .Net Enterprise Developer**

Pada edisi ini terdapat fitur yang ada pada edisi professional ditambah dengan **kemampuan pengembangan aplikasi enterprise**, seperti :

- Tim pengembang enterprise dapat bekerja sama untuk aplikasi windows dan web.
- Terdapat tool dan teknologi untuk membangun aplikasi enterprise.
- Mempunyai template proyek enterprise dan framework yang berisi petunjuk arsitektur untuk membuat aplikasi.

- **Visual Studio .Net Enterprise Architec**

Fitur yang ada pada edisi sama dengan fitur yang ada pada edisi enterprise developer juga ditambah dengan **kemampuan untuk mendesain, menentukan, dan mengkomunikasikan arsitektur dan fungsionalitas aplikasi**. Beberapa fitur tambahannya adalah :

- Dapat digunakan untuk mendesain web XML dan aplikasi secara visual.
- Terdapat Modeling Visio berbasis UML (Unified Modelling Language) untuk membuat aplikasi arsitektur bisnis dan desain database.

- Terdapat fasilitas BizTalkServer Developer Edition untuk proses bisnis.
- Terdapat Enterprise template dan frameworks dengan menggunakan Template Description Language.

### 1.1.3. Mengenal .Net Framework

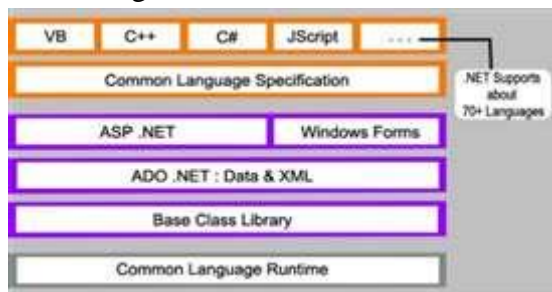
Menurut Yuswanto (2008), **.Net Framework** adalah lingkungan komputasi baru / platform tunggal yang menyederhanakan proses pembuatan aplikasi pada lingkungan terdistribusi internet (sistem komputer mandiri yang terhubung dalam suatu jaringan).

Dengan demikian .Net Framework didesain untuk memenuhi tujuan berikut :

- Menyediakan lingkungan pemrograman berorientasi obyek yang konsisten meskipun kode obyek disimpan dan dijalankan secara lokal. Meskipun kode obyek dijalankan secara lokal tetapi disebarkan melalui internet atau dijalankan secara remote.
- Menyediakan lingkungan yang menjalankan kode dengan meminimalkan konflik saat software disebarkan / deployment dan tentang versi / versioning.
- Menyediakan lingkungan untuk menjalankan kode yang menjamin keamanan saat kode dijalankan, termasuk kode yang dijalankan oleh pihak ketiga yang tidak diketahui atau kurang dipercaya.



- Menyediakan lingkungan untuk menjalankan kode yang dapat menghilangkan masalah performa dari lingkungan scripted dan interpreted.
- Menyediakan lingkungan berbasis standar produksi yang tinggi untuk memadukan investasi yang ada dengan aplikasi dan service generasi mendatang.
- Membuat pengembang memiliki pengalaman yang konsisten diberbagai tipe aplikasi seperti aplikasi berbasis windows dan aplikasi berbasis web.
- Membangun komunikasi standar industri guna memastikan kode berbasis .Net Framework dapat disatukan dengan kode-kode lain.



Gambar 1.2. Struktur .Net Framework  
[Hidayatullah:2015]

Gambar 1.2 menunjukkan stuktur .Net Framework. Pada struktur .Net Framework terdapat banyak class yang dapat digunakan untuk mengembangkan aplikasi client, seperti aplikasi berbasis web. Kumpulan dari class (pustaka class) pada .Net Framework dibagi menjadi empat area yaitu :

- ASP.NET dan Service Web  
ASP .NET menyediakan inti dari infrastruktur form-form web untuk pengembangan berbasis UI (User Interface) untuk antarmuka pengguna. Service web digunakan untuk pengembangan antarmuka secara terprogram.
- Form-Form Windows  
Form-form windows digunakan untuk pengembangan antarmuka pengguna pada platform windows.
- ADO.NET dan XML  
ADO.NET dan XML digunakan untuk mengakses database.
- Inti dari pustaka base class adalah menyediakan infrastruktur layanan seperti keamanan, pengolahan transaksi, dsb.

.Net Framework terdiri dari dua bagian utama yaitu Common Language Runtime (CLR) dan gabungan kelas library termasuk juga ASP.NET untuk aplikasi web dan XML Web Service, dan Windows Form untuk aplikasi klien ADO.NET.

#### **A. Common Language Runtime (CLR)**

**CLR atau runtime** merupakan dasar dari .Net Framework. Runtime merupakan *engine yang menjalankan aplikasi .Net Framework*. Layanan inti yang dapat ditangani oleh runtime adalah :

- Manajemen memori suatu aplikasi.
- Mengolah kode program saat loading dan run.

- Mengolah keamanan kode akses.
- Memastikan pendefinisian tipe data secara ketat.
- Melakukan konversi dari IL (*Intermediate Language*) ke kode-kode asli.
- Menangani kejadian yang tidak diharapkan yang muncul saat kode eksekusi.
- Otomatisasi tampilan obyek.
- Mendukung layanan pengembangan seperti *profiling*, *debugging*, dan sebagainya.

**Prinsip dasar runtime** adalah konsep pengolahan kode. Kode program yang dijalankan oleh runtime disebut **kode terkelola** (*Managed Code*), sedangkan kode program yang tidak dijalankan oleh runtime disebut **kode tidak terkelola** (*Unmanaged Code*).

**Visual Basic .Net** menghasilkan **kode terkelola yang dapat dijalankan apabila pada sistem komputer terdapat runtime**. Artinya, suatu aplikasi yang dihasilkan oleh Visual Basic .Net tidak dapat dijalankan sebelum pada sistem komputer diinstal terlebih dahulu runtime (CLR). Jadi keberadaan CLR sangat menentukan kapan suatu obyek digunakan dan dilepas. Kondisi semacam ini disebut dengan *Managed Code*. Seperti *Unmanaged Kode* seperti yang digunakan pada Visual Basic versi awal, terjadi pada program yang dikompilasi dengan COM dan dideklarasikan oleh Win32API.

DOT NET Framework dapat dihosted oleh control *unmanaged Kode* yang memanggil runtime (CLR) ke prosesnya dan menjalankan managed Kode sehingga tercipta suatu lingkungan software yang menggunakan fitur managed dan unmanaged. Untuk melakukan runtime, dibutuhkan compiler bahasa yang mengikuti CLS (*Common Language Specification*). Tujuannya untuk menerangkan suatu subset tipe data yang didukung oleh runtime dan umumnya digunakan oleh semua bahasa dalam lingkup .Net.

## B. Dot Net Framework Class Library

**Dot NET Framework merupakan koleksi dari tipe data yang digunakan secara berulang-ulang dan terintegrasi dengan CLR. Class library sifatnya berorientasi obyek, menyediakan tipe data dimana *managed Kode* dapat mewariskan fungsionalitasnya. Konsep pewarisan (*inheritance*) menyediakan cara bagi pengembang untuk memakai kode-kode dari suatu class sehingga kode-kode yang terbentuk dalam suatu fungsi tersebut dapat dipakai berulang-ulang tanpa harus menulis kembali kodenya.** Hal ini selain membuat tipe data dari .Net Framework mudah dipakai tetapi juga mengurangi waktu belajar dari fasilitas .Net Framework.

Dot Net Framework memberikan suatu fasilitas untuk menyelesaikan sejumlah tugas pemrograman umum, termasuk pengolahan string, konektivitas database, dan akses file. Class library juga memiliki tipe-

tipe data yang mendukung variasi scenario pengembangan terspesialisasi. Sebagai contoh, .Net Framework dapat digunakan untuk mengembangkan tipe aplikasi dan layanan diantaranya aplikasi console, aplikasi scripting atau hosted, aplikasi windows GUI (Windows Form), aplikasi ASP .Net, XML Web Service, dan Windows Service.

Teknologi .Net Framework merupakan teknologi yang mampu mendukung 20 lebih bahasa pemrograman, termasuk Visual Basic .Net. Saat ini .Net framework juga dikembangkan untuk mobile dan embedded devices yang disebut **Compact .Net Framework**. Jadi dengan menguasai salah satu bahasa pemrograman yang mendukung teknologi .Net Framework, seseorang dapat menghasilkan berbagai tipe aplikasi maupun target aplikasi seperti pada server, desktop, atau smartp device seperti PDA.

#### 1.1.4. Sejarah Singkat Visual Basic .Net

Menurut Yuswanto (2006), cikal bakal bahasa pemrograman Visual Basic adalah **BASIC** (*Beginner's All-purpose Symbolic Instruction Code*) yang diciptakan sekitar tahun **1964** oleh **Profesor John Kennedy dan Thomas Kurtz**. Basic ditujukan sebagai bahasa yang paling sederhana bagi orang yang tidak terlalu familiar dengan dunia pemrograman.

Pada tahun **1975**, Microsoft mengeluarkan produk pertamanya dan sukses yaitu *Quick Basic /*

**QBasic**, yang banyak digunakan pada komputer pribadi / PC (*Personal Computer*). Kemudian Microsoft mengeluarkan produk terbaru yang sangat revolusioner yaitu **Visual Basic 1.0** yang lebih dikenal dengan istilah **GUI / Graphical User Interface**.

Pada tahun **1993**, **Microsoft Visual Basic 3.0** dirilis yang ditandai dengan adanya **kontrol data standar** yang menyediakan **akses database dalam aplikasi dengan pengkodean minimal**. Aplikasi ini disertai dengan mesin **database Jet versi 1.1 (*Jet Engine*) yang digunakan untuk konektivitas database melalui DAO (*Data Access Objects / kontrol data*)**. Keberadaan mesin Jet ini diteruskan hingga **Jet versi 4.0**.

Pada tahun **1996**, **Visual Basic 4.0** dirilis yang ditandai dengan adanya teknologi **OLE (*Object Linking and Embedding*) dan kemampuan untuk membuat obyek**. Versi ini mendukung **kontrol custom 32-bit** yang disebut **OCX**. Selain itu terdapat fasilitas **pemrograman database RDO (*Remote Data Objects*) dan RDC (*Remote Data Control*)**. **RDO** adalah sebuah metode akses data yang dirancang dari bawah ke atas untuk menggantikan DAO. Pustaka ActiveX 32-bit ini lebih cepat dan lebih kecil dari DAO, dan dirancang bersama hierarki obyek yang sama dengan ODBC API.

Pada tahun **1997**, **Visual Basic 5.0** dirilis dengan **mendukung standar COM Microsoft dan memungkinkan pembuatan kontrol-kontrol ActiveX**. Versi ini mempunyai **kemampuan membuat bahasa**

**interpreter / compile dengan performa yang lebih baik.** Versi ini merupakan **terobosan nyata** karena **pengembang program bisa membuat kontrol DLL sendiri.**

Pada tahun **1998, Visual Basic 6.0** dirilis dengan kemampuan berinteraksi dengan **database SQL Server.** Salah satu fasilitas terbaru adalah **ADO (ActiveX Data Objects versi 2.0).** ADO dirancang bagi kinerja perusahaan besar dan aplikasi internet serta merupakan salah satu komponen utama dalam UDA (*Universal Data Access*) versi terbaru ADO adalah 2.5, yang disertakan dalam sistem operasi Windows 2000.

Perkembangan Microsoft .NET pertama kali dipaparkan oleh Microsoft pada bulan juli 2000 dalam PDC (*Professional Developers Conference*) di Orlando, Amerika Serikat. Pada tahun **2002,** Microsoft resmi merilis **Visual Studio .Net 2002** yang terdiri dari **Visual Basic .Net, Visual C++ .Net, Visual C# .Net, dan Visual J# .Net.**

Pada tahun **2003, Visual Studio .Net 2003** dirilis **dengan diperbaikinya performa dan aspek keamanan dari Visual Basic .Net 2002.** Pada versi ini disertakan **.Net Framework 1.1.** Kemudian Microsoft terus memperbaiki Visual Studio .Net untuk menutupi kekurangan dari versi sebelumnya. Berikut perkembangan Visual Basic .Net [6] :

- Visual Studio .Net 2005
- Visual Studio .Net 2008

- Visual Studio .Net 2010
- Visual Studio .Net 2012
- Visual Studio .Net 2013
- Visual Studio .Net 2015
- Visual Studio .Net 2017
- Visual Studio .Net 2019

#### 1.1.5. Kelebihan Visual Basic .Net

Suatu aplikasi memiliki kelebihan dan kelemahan, tetapi yang dibahas pada modul ini hanya kelebihan saja. Menurut Priyanto Hidayatullah (2015), Berikut **kelebihan** yang dimiliki oleh **Visual Basic .Net** yaitu :

- Sederhana dan mudah dipahami  
Seperti pada VB, bahasa yang digunakan pada VB.Net sangat sederhana sehingga lebih mudah dipahami bagi yang masih awam di dunia pemrograman.
- Mendukung GUI  
VB .Net bisa membuat software dengan antarmuka grafis yang lebih user friendly.
- Menyederhanakan deployment  
VB .Net mengatasi masalah deployment dari aplikasi berbasis windows yaitu DLL Hell dan registrasi COM (*Component Object Model*). Selain itu tersedia wizard yang memudahkan dalam pembuatan file setup.
- Menyederhanakan pengembangan perangkat lunak



Hal-hal yang membuat pengembangan perangkat lunak menjadi lebih lunak adalah terjadi kesalahan penulisan kode dari sisi sintaks (bahasa), maka VB.Net langsung menuliskan kesalahannya pada bagian *Message Windows* sehingga *programmer* dapat memperbaiki kode dengan lebih cepat. Editor menu bersifat WYSIWYG (*What You See Is What You Get*). Adanya berbagai wizard yang memandu programmer dalam membuat software. Tersedianya Crystal Reports (CR) untuk membuat laporan (pada Visual Studi versi wal, crystal reports diinstal terpisah). Adanya Kode Snippets yaitu fitur yang menyisipkan kode-kode koleksi yang pernah dibuat pada program yang sedang dibuat.

- Mendukung penuh OOP  
Memiliki fitur bahasa pemrograman berorientasi obyek seperti *inheritance* (pewarisan), *encapsulation* (pembungkusan), dan *polymorphism* (banyak bentuk).
- Mempermudah pengembangan aplikasi berbasis web  
Disediakan designer form web. Selain itu disediakan layanan Web XML sehingga memungkinkan suatu aplikasi “berkomunikasi” dengan aplikasi lainnya dari berbagai platform menggunakan protocol internet terbuka.
- Migrasi ke VB Net dapat dilakukan dengan mudah  
Jika programmer telah membuat aplikasi di VB, maka konversi ke VB.Net dapat dijalankan dengan mudah.

- Banyak digunakan oleh programmer-programmer di seluruh dunia  
Salah satu keuntungannya adalah jika programmer mengalami masalah/pertanyaan, maka bisa bisa bertanya ke programmer lain diseluruh dunia melalui forum-forum di internet.

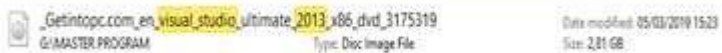
## 1.2. Cara Instalasi dan Uninstall Visual Basic .Net 2013

Seerti yang telah dijelaskan pada penjelasan sebelumnya, pada modul ini program Visual Studio .Net yang digunakan adalah program Visual Studio .Net 2013 karena alasan kecepatan, produk Visual Studio 2013 termasuk produk keluaran terbaru, dan fitur pada Visual Studio 2013 sudah mencukupi untuk menyelesaikan kasus-kasus pada modul ini. Pada subbab ini akan dijelaskan cara instalasi dan uninstall Visual Studio 2013.

### 1.2.1. Cara Instalasi Visual Studio 2013

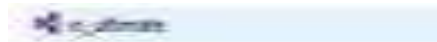
Untuk instalasi Visual Studio .Net 2013, lakukan beberapa langkah berikut ini yaitu :

1. Bukalah file instalasi Visual Studio .Net 2013 di komputer, seperti gambar 1.3.



Gambar 1.3. File Instalasi Visual Studio .Net 2013

2. Klik **setup.exe** atau file vs\_ultimete pada installer Visual Studio .Net 2013, seperti gambar 1.4.



Gambar 1.4. File Vs\_Ultimate pada Visual Studio .Net 2013

3. Akan muncul kotak dialog seperti gambar 1.5 dibawah ini. Pilihlah letak penyimpanan file yang akan diinstalasi, defaultnya akan teretak di folder C:\, jadi tidak perlu diganti. Lalu **centanglah I agree**. Kemudian **klik Next**. Sebagai catatan jika di folder C:\ tidak mencukupi untuk tempat penyimpanan instalasi Visual Studio .Net 2013, maka tempat penyimpanan bisa diganti di folder lain, dengan cara mengganti di bagian **Setup Require** kemudian **klik kotak yang bergambar titik-titik**.



Gambar 1.5. Kotak Dialog Awal Pada Instalasi Visual Studio .Net 2013

4. Lalu akan muncul kotak dialog kedua yang ditampilkan pada gambar 1.6. Kotak dialog ini menampilkan file atau program apa saja yang akan terinstall pada Visual Studio .Net 2013. Secara default, file yang akan terinstal oleh Visual Studio .Net 2013 akan tercentang dengan sendirinya, tetapi jika pengguna meu mengubah file apa saja yang akan terinstal pada Visual Studio .Net 2013 maka pengguna bisa mengubahnya dengan mencentang atau tidak pada file yang akan terinstall atau **klik select all** untuk



menginstall semua file yang ada. Tetapi untuk pemula, **tidak perlu merubah apa pun** jadi langsung **Klik Install**.

- Gambar 1.6. Kotak Dialog Kedua Pada Visual Studio .Net 2013
5. Proses instalasi Visual Studio .Net 2013 dimulai. Tunggulah hingga proses selesai, kira-kira 30 hingga 45 menit. Jika **proses instalasi selesai** maka akan muncul tampilan seperti gambar 1.7.



Gambar 1.7. Proses Instalasi Visual Studio .Net 2013 telah selesai

### 1.2.2. Cara Uninstalasi Visual Studio 2013

Untuk menghapus program Visual Studio .Net 2013 yang telah terinstal di komputer yaitu dengan cara : **Start > Control Panel > Apps > Memilih Visual Studio .Net 2013 Ultimate > Modify > Uninstall > Tunggu hingga proses selesai.** Berikut gambar untuk menu uninstall yang ditampilkan pada gambar 1.8.



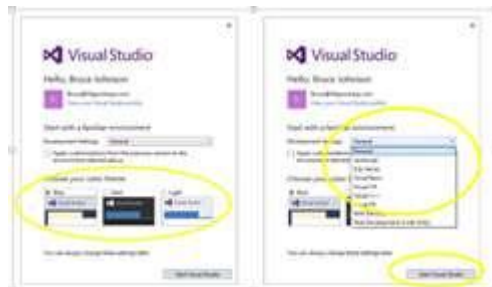
Gambar 1.8. Uninstal Visual Studio .Net 2013

### 1.3. Memulai Visual Studio .Net 2013 dan Membuat Project Baru

Pada subbab ini akan dijelaskan cara memulai Visual Studio .Net 2013 dan membuat project baru pada Visual Studio .Net 2013.

Untuk memulai Visual Studio .Net 2013 dan membuat project baru pada Visual Studio .Net 2013 yaitu dengan cara : **Start Windows → Visual Studio 2013**. Jika pengguna pertama kali menjalankan Visual Studio .Net 2013 maka akan muncul tampilan seperti gambar 8. Tampilan pada gambar 1.8 digunakan untuk memilih warna tampilan dan memilih program apa yang akan digunakan. Berikut langkah-langkahnya :

- **Untuk memilih warna (gambar sebelah kiri), perhatikan pilihan pada Choose Your Colour Theme → lalu pilih salah satu warna pada menu tersebut, apakah Blue, Dark, atau Light.**
- **Untuk memilih program apa yang akan digunakan (gambar sebelah kanan), perhatikan pilihan pada Development Setting → lalu pilih Visual Basic.**
- **Klik Start Visual Studio.**



Gambar 1.8. Tampilan Awal Menjalankan Visual Studio 2013

- Kemudian muncul tampilan seperti gambar 1.9. Pada gambar 1.9 ini digunakan untuk memulai aktivitas pemakai. Misalnya untuk **membuat project baru** maka pilih **New Project**, untuk **membuka project** yang telah dibuat maka pilih **Open Project**. Lalu ada menu **Recent** yang menampilkan program apa yang telah dibuka oleh pemakai. Untuk modul ini, pengguna akan membuat project baru, maka **klik New Project**.



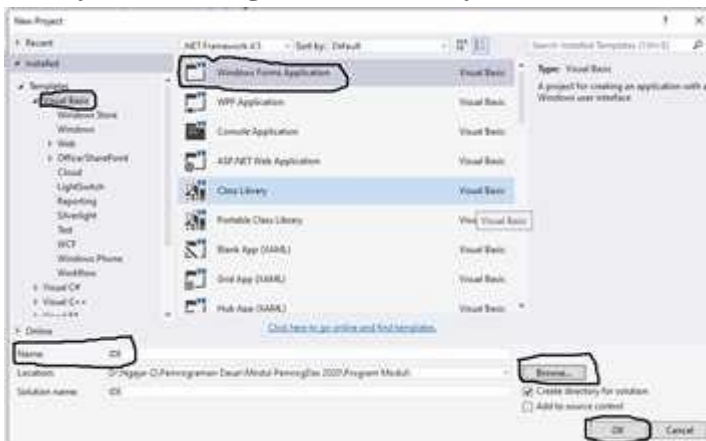
Gambar 1.9. Pemilihan Project Visual Studio 2013

Untuk membuat project baru, maka pengguna harus membuka Visual Studio .Net 2013 hingga muncul tampilan seperti gambar 1.9.

- Kemudian **klik New Project**, akan muncul tampilan seperti gambar 10 berikut. Pada gambar 1.10 ini pengguna diminta untuk memilih bahasa pemrograman apa yang akan digunakan. Karena modul ini membahas Visual Basic, maka **pilih Visual Basic** lalu **pilih Windows Form Application**, lalu **Ubah nama file pada isian Name**, lalu untuk merubah lokasi penyimpanan klik Browse pada Location, **lalu pilih folder mana yang akan digunakan untuk tempat**

penyimpanan, lalu **klik OK**. Perlu diketahui, untuk nama Solution akan sama dengan nama file yang ditulis pada isian Name.

Sebagai contoh, pengguna membuat project dengan nama IDE dan project disimpan di tempat penyimpanan lain. Maka, pada isian **Name** diisi **IDE** dan isian **Location** diisi secara otomatis oleh sistem sesuai dengan pilihan pengguna menjadi **D:\Ngajar Q\Pemrograman Dasar\Modul PemrogDas 2020\Program Modul**. Untuk isian **Solution Name** isinya sama dengan isian Name yaitu **IDE**.



Gambar 1.10. Tampilan New Project Visual Basic

#### 1.4. Berkenalan dengan Visual Basic .Net 2013

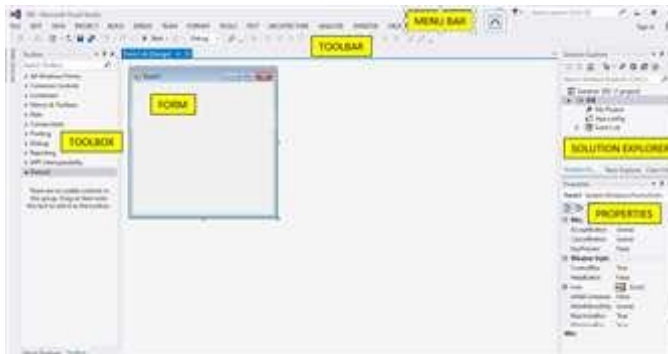
Pada subbab ini akan dijelaskan tentang lingkungan Visual Basic .Net 2013 diantaranya adalah IDE (*Integrated Development Environment*) Visual



Basic .Net 2013, Kode Editor, serta Output Window dan Error List Window.

#### 1.4.1. IDE (Integrated Development Environment) Visual Basic .Net 2013

Tampilan IDE Visual Basic .Net 2013 ditampilkan pada gambar 1.11. Pada gambar 1.11 terdapat bagian-bagian dari Visual Basic .Net 2013 diantaranya adalah form, menu bar, toolbar, toolbox, properties, dan solution explorer.



Gambar 1.11. Tampilan IDE Visual Studio .Net 2013

##### A. Form Visual Studio .Net 2013

Form adalah bagian IDE Visual Studio .Net 2013 yang berada di bagian paling tengah-tengah antara bagian toolbox dan bagian solution explorer. Bagian form merupakan tempat dimana kontrol / tool diletakkan. User dapat melakukan drag dan drop tool yang akan digunakan dan dipasang di bagian tengah form. Form berfungsi sebagai tempat untuk pembuatan tampilan / antarmuka / *user interface* dari sebuah

aplikasi Windows. Dengan menggunakan form, pengguna lebih mudah untuk mengatur tampilan dari aplikasi yang akan dibuat sehingga pengguna dapat lebih banyak berkonsentrasi padajalannya aplikasi tanpa membuang banyak waktu untuk mendesain tampilan program.

Secara default, pada waktu pengguna membuat proyek *Windows Application*, maka telah terdapat sebuah form bernama form1 pada proyek tersebut, yang terletak di sebelah kiri atas form. Selain nama form1, di form tersebut terdapat control box, yang berada di sebelah kanan atas form. Control box ini berisi minimizebox (tombol yang digunakan untuk mengecilkan form), maximizebox (tombol yang digunakan untuk memperbesar form), dan close (tombol yang digunakan untuk menutup form). Form ini dapat diperbesar dan diperkecil sesuai keinginan dengan cara mengarahkan form pada titik yang ada di form kemudian menarik maju atau mundur / resize. Form bagian tengah berfungsi untuk meletakkan tool yang akan digunakan dalam membuat aplikasi. Tampilan form ini dapat dilihat pada gambar 1.12.



Gambar 1.12. Form Visual Studio .Net 2013

Pada gambar 1.12, terdapat tiga titik pada form tersebut yaitu titik yang terletak di samping kanan, titik yang terletak di bagian bawah, dan titik yang terletak di pojok kanan bawah. Jika kursor diarahkan pada titik yang terletak disamping kanan maka form dapat ditarik ke kanan atau ke kiri, jika kursor diarahkan pada titik yang terletak di bagian bawah maka form dapat ditarik ke bawah atau ke atas, sedangkan jika kursor diarahkan pada titik yang terletak di pojok kanan bawah maka form dapat ditarik secara diagonal ke bawah atau ke atas.

## **B. Menu Bar Visual Studio .Net 2013**

Menu bar adalah bagian IDE Visual Studio .Net 2013 yang berada di bagian paling atas. Menu bar adalah bagian IDE yang terdiri dari perintah-perintah untuk mengatur IDE, mengedit kode, dan mengeksekusi program. Di dalam menu bar, perintah-perintah dikelompokkan ke dalam beberapa bagian sesuai dengan jenis perintah tersebut. Tampilan menu bar Visual Studio .Net 2013 ditampilkan pada gambar 1.13.



Gambar 1.13. Menu Bar Visual Studio .Net 2013

Untuk menggunakan menu bar atau pilihan pada menu bar, pengguna tinggal meng-klik menu atau pilihan yang akan dijalankan. Contoh jika pengguna ingin membuat proyek baru, maka pilih menu File > New > Project. Pada IDE Visual Studio .Net 2013 terdapat 13 menu utama, antara lain :

1. Menu **File** berisi perintah-perintah untuk membuat proyek, membuka proyek, menutup proyek, mencetak data dari proyek, dll.
2. Menu **Edit** berisi perintah-perintah untuk undo, cut, paste, dll.
3. Menu **View** berisi perintah-perintah untuk menampilkan window-window dari IDE dan toolbar.
4. Menu **Project** berisi perintah-perintah untuk mengatur proyek dan file-file.
5. Menu **Build** berisi perintah-perintah untuk meng-compile program.
6. Menu **Debug** berisi perintah-perintah untuk men-debug dan menjalankan program.
7. Menu **Team** berisi perintah untuk berhubungan dengan *Team Foundation Server*.
8. Menu **Tools** berisi perintah-perintah untuk mengakses komponen IDE tambahan dengan mengubah IDE.
9. Menu **Test**, mengelola pengujian yang akan dilakukan pada proyek.
10. Menu **Architecture** berisi tools-tools pemodelan data menggunakan ER-Diagram, UML, dsb.
11. Menu **Analyze**, menganalisis kode untuk mengetahui performansinya.
12. Menu **Window** berisi perintah-perintah untuk mengatur dan menampilkan Windows.
13. Menu **Help** berisi perintah-perintah untuk mengakses fasilitas bantuan.

### C. Toolbar Visual Studio .Net 2013

Toolbar adalah bagian IDE Visual Studio .Net 2013 yang berada di atas dibawah menu bar. Toolbar fungsinya sama seperti menu bar. Bedanya pada toolbar pilihan-pilihan berbentuk icon. Untuk menggunakan toolbar, klik salah satu icon yang diinginkan, misalnya jika ingin menyimpan file yang telah dibuat maka klik salah satu icon berbentuk disket



. Icon-icon pada toolbar adalah pilihan-pilihan pada menu bar yang sering digunakan dalam membuat program aplikasi. Toolbar memudahkan pengguna untuk memilih proses yang sering dilakukan tanpa harus memilihnya pada menu bar. Tampilan toolbar ditampilkan pada gambar 1.14.



Gambar 1.14. Toolbar Visual Studio .Net 2013


### D. ToolBox Visual Studio .Net 2013



Toolbox adalah bagian IDE Visual Studio .Net 2013 yang berada di sebelah kiri form. Toolbox adalah letak tool dan komponen-komponen. Tampilan toolbox dapat dilihat pada gambar 1.15.




Gambar 1.15. Toolbox Visual Studio .Net 2013

Tool dan komponen yang ada pada toolbox dipakai dalam pembuatan program aplikasi. Untuk **menempatkan tool atau komponen pada form program aplikasi yaitu dengan cara drag dan drop dari toolbox ke form program aplikasi**. Cara **menampilkan jendela toolbox** yaitu pilih menu **View > Toolbox** atau tekan **Ctrl + Alt + X**.

Pada toolbox yang ditampilkan pada gambar 1.15, terdapat panah disebelah kiri kategori toolbox **▷ All Windows Forms**, panah ini digunakan untuk melihat isi masing-masing tool yang terletak pada masing-masing toolbox. Caranya dengan **menekan panah disebelah kiri kategori toolbox ▷ All Windows Forms**. Disebelah atas toolbox terdapat icon panah yang menunjuk ke bawah , panah ini disebut **Window Position**. Panah ini digunakan untuk **mengatur letak toolbox**. Letak toolbox dapat dipindahkan letaknya, tidak selalu harus terletak di sebelah kiri form. Di sebelah kanan Window Position, terdapat icon


**Auto Hide** , yang digunakan untuk menampilkan atau menyembunyikan toolbox. Di sebelah kanan Auto Hide, terdapat icon **Close** , yang digunakan untuk menutup toolbox.

Tool dan komponen-komponen yang terletak dalam toolbox dibagi dalam beberapa kategori yaitu :

1. **All Windows Forms**, , berisi seluruh tool dan komponen yang ada pada toolbox. Berikut contoh tampilan beberapa toolbox yang berada pada kategori All Windows Forms.



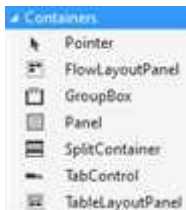
Gambar 1.16. Toolbox pada Kategori All Windows Forms

2. **Common Controls**, , berisi tool umum yang sering digunakan seperti label, textbox, button, dsb. Berikut contoh tampilan beberapa toolbox yang berada pada kategori Common Controls.



Gambar 1.17. Toolbox pada Kategori Common Controls

3. **Containers**, ▸ **Containers** , berisi tool yang mengelompokkan tool lainnya seperti panel, groupbox, tabcontrol, dsb. Berikut contoh tampilan toolbox yang berada pada kategori Containers.



Gambar 1.18. Toolbox pada Kategori Containers

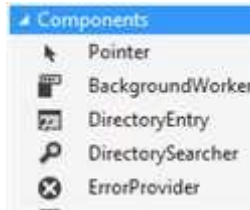
4. **Menus & Toolbars**, ▸ **Menus & Toolbars** , berisi tool dan komponen menu, context menu, dan toolbar. Berikut contoh tampilan toolbox yang berada pada kategori Menus & Toolbars.



Gambar 1.19. Toolbox pada Kategori Menus & Toolbars



5. **Components**, ▸ **Components**, berisi komponen seperti timer, imagelist, dsb. Berikut contoh tampilan beberapa toolbox yang berada pada kategori Components.



Gambar 1.19. Toolbox pada Kategori Components

6. **Printing**, ▸ **Printing**, berisi komponen untuk mencetak dokumen. Berikut contoh tampilan toolbox yang berada pada kategori Printing.

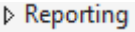


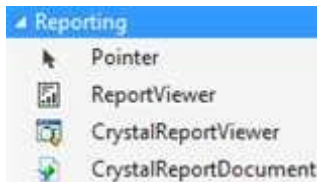
Gambar 1.20. Toolbox pada Kategori Printing

7. **Dialogs**, ▸ **Dialogs**, berisi komponen untuk berinteraksi dengan pengguna dalam hal membuka file, menyimpan file, membuka folder, dll. Berikut contoh tampilan toolbox yang berada pada kategori Dialogs.

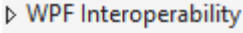


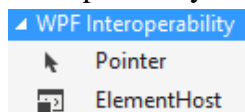
Gambar 1.21. Toolbox pada Kategori Dialogs

8. **Reporting**,  **Reporting**, berisi tool untuk membuat laporan pada Visual Basic .Net 2013. Berikut contoh tampilan toolbox yang berada pada kategori Reporting.

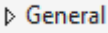


Gambar 1.22. Toolbox pada Kategori Reporting

9. **WPF Interoperability**,  **WPF Interoperability**, berisi komponen untuk *Windows Presentation Foundation*. Berikut contoh tampilan toolbox yang berada pada kategori WPF Interoperability.



Gambar 1.23. Toolbox pada Kategori WPF Interoperability







10. **General**,  **General**, jika ada tool atau komponen yang mau ditambahkan dan belum memiliki kategori yang jelas, maka bisa dimasukkan ke kategori ini.

Berikut contoh tampilan toolbox yang berada pada kategori General.





















Gambar 1.24. Toolbox pada Kategori General








Penjelasan fungsi-fungsi dari tool pada toolbox yang ada di Visual Basic .Net 2013 :

- **Pointer**,  Pointer : untuk memilih pointer mouse atau melepaskan dari objek-objek toolbox.
- **BackgroundWorker**,  BackgroundWorker : mengeksekusi operasi tertentu di thread lain.
- **BindingNavigator**,  BindingNavigator : menyediakan antarmuka untuk navigasi dan manipulasi data yang dikaitkan dengan form.
- **BindingSource**,  BindingSource : mengenkapsulasi sumber data untuk sebuah form dan menyediakan fasilitas update, navigasi, filter, dan sorting.
- **Button**,  Button : untuk menerima klik dari pengguna. Jika pengguna mengkliknya maka kode tertentu bisa dieksekusi.
- **CheckBox**,  CheckBox : untuk menerima tanda centang dari pengguna. Jika pengguna mencentangnya


maka item di kotak tersebut otomatis bisa dipakai untuk program.










- **CheckedListBox**,  `CheckedListBox` : menampilkan item-item seperti lisbox biasa, namun dengan checkbox di sebelah kirinya sehingga satu listbox bisa menerima dua atau lebih item.
- **ColorDialog**,  `ColorDialog` : menampilkan warna-warna untuk dipilih.
- **ComboBox**,  `ComboBox` : menampilkan item-item namun dalam bentuk hanya satu terlihat. Jika pengguna mengklik ComboBox barulah semua item di dalamnya akan kelihatan.
- **ContextMenuStrip**,  `ContextMenuStrip` : menampilkan menu shourcut jika pengguna mengklik kanan pada item.
- **DataGridView**,  `DataGridView` : menampilkan data dalam bentuk tabel yang bisa dimodifikasi.
- **DataSet**,  `DataSet` : merupakan representasi data dari sumber data di memori.
- **DateTimePicker**,  `DateTimePicker` : memungkinkan pengguna menampilkan tanggal dan memilihnya. Selanjutnya tanggal dan waktu yang dipilih dapat dimodifikasi menggunakan format tertentu.
- **DirectoryEntry**,  `DirectoryEntry` : mengenkapsulasi node atau objek di hierarki Active Directory (AD) di windows server.

- **DirectorySearcher**,  DirectorySearcher : melakukan query terhadap Active Directory.
- **DomainUpDown**,  DomainUpDown : menampilkan nilai string yang ditampilkan pengguna dari daftar item dengan cara mengklik tombol Up dan Down.
- **ErrorProvider**,  ErrorProvider : menentukan dukungan untuk inisialisasi transaksi.
- **EventLog**,  EventLog : menyediakan interaksi dengan EventLog dari windows.
- **FileSystemWatcher**,  FileSystemWatcher : memonitor perubahan sistem file dan mengeksekusi event ketika ada perubahan direktori atau file.
- **FlowLayoutPanel**,  FlowLayoutPanel : menangani layout dan komponennya, kemudian mengaturnya dalam layout yang otomatis.
- **FolderBrowserDialog**,  FolderBrowserDialog : menampilkan kotak dialog yang meminta pengguna untuk memilih folder yang ada.
- **FontDialog**,  FontDialog : menampilkan kotak dialog yang meminta pengguna untuk memilih font yang ada.
- **GroupBox**,  GroupBox : mengelompokkan kontrol dalam satu kotak untuk mempermudah pengaturan dan manajemen antarmuka.
- **HelpProvider**,  HelpProvider : menyediakan bantuan popup atau online untuk sebuah kontrol.









- **HscrollBar**,  `HScrollBar` : memungkinkan komponen induk untuk menggulirkan isinya secara horizontal. Kontrol ini tidak diperlukan jika komponen induknya telah memiliki atribut Autoscroll atau atribut tersebut sudah dalam keadaan enabled.
- **ImageList**,  `ImageList` : menampilkan daftar gambar. Banyak gambar yang bisa disimpan di ImageList yang nantinya bisa digunakan oleh kontrol lain seperti ListView, TreeView, dan ToolStrip.
- **Label**,  `Label` : untuk menampilkan teks dimana nantinya isi didalamnya tidak bisa diberimasukan oleh pengguna seperti halnya textbox.
- **LinkLabel**,  `LinkLabel` : sama seperti label, namun bisa memiliki link ke aplikasi atau WWW. Jika diklik, pengguna akan dibawa ke alamat URL yang ditentukannya. Jika alamat URL ada di internet, otomatis browser akan menampilkan alamat tersebut.
- **ListBox**,  `ListBox` : menampilkan item-item dalam bentuk kotak di mana semua item ditampilkan di dalam daftar. Jika jumlah item lebih banyak dari ukuran ListBox maka ListBox otomatis memiliki scrollbar vertikal yang dapat dipakai untuk menampilkan semua item tersebut.
- **ListView**,  `ListView` : menampilkan item-item satu dari lima tampilan yang berbeda.
- **MaskedTextBox**,  `MaskedTextBox` : menerima input dari pengguna seperti halnya TextBox, namun isi di








dalamnya nantinya dapat di-masking menggunakan format tertentu. Ini cocok untuk pengisian nomor telepon atau sejenisnya dimana jika dimasukkan 0245347125023198 akan otomatis diganti menjadi (024) 329718257012798.




- **MenuStrip**,  `MenuStrip` : menampilkan perintah dan aplikasi yang dikelompokkan menurut fungsinya.
- **MessageQueue**,  `MessageQueue` : menyediakan akses ke queue di Message Queue Server.
- **MonthCalendar**,  `MonthCalendar` : menampilkan kalender bulanan di mana pengguna bisa memilih tanggal.
- **NotifyIcon**,  `NotifyIcon` : menampilkan icon di area notifikasi ketika program berjalan. Lokasi area notifikasi ada di sebelah kanan taksbar windows.
- **NumericUpDown**,  `NumericUpDown` : menampilkan daftar angka dimana pengguna dapat menggulung menggunakan tombol up dan down
- **OpenFileDialog**,  `OpenFileDialog` : menampilkan jendela dialog yang meminta pengguna untuk memilih file tertentu untuk dibuka.
- **PageSetupDialog**,  `PageSetupDialog` : menampilkan jendela dialog yang memungkinkan pengguna mengubah pengaturan yang berkaitan dengan halaman, seperti margin dan orientasi kertas.

- **PictureBox**,  `PictureBox` : menampilkan gambar yang tampilannya bisa diatur sedemikian rupa.
- **PrintDialog**,  `PrintDialog` : menampilkan kotak dialog yang berkaitan dengan pengaturan pencetakan.
- **PrintDocument**,  `PrintDocument` : mendefinisikan objek yang dikirim ke printer.
- **PrintPreviewControl**,  `PrintPreviewControl` : merepresentasikan bagian dari print preview yang menampilkan dokumen yang sedang di-preview. Kontrol ini tidak menampilkan berbagai dialog ataupun tombol.
- **PrintPreviewDialog**,  `PrintPreviewDialog` : menampilkan kotak dialog yang menampilkan preview kepada pengguna berupa apa yang akan dicetak.
- **Process**,  `Process` : menyediakan akses ke proses lokal atau remote dan memungkinkan pengguna menghentikan atau menginisialisasi proses-proses lokal.
- **ProgressBar**,  `ProgressBar` : menampilkan sebuah bilah yang mengindikasikan kemajuan operasi pengguna tertentu.
- **PropertyGrid**,  `PropertyGrid` : menampilkan antarmuka untuk melihat properti dari objek-objek tertentu.
- **RadioButton**,  `RadioButton` : memungkinkan pengguna untuk memilih satu opsi dari beberapa pilihan yang dikaitkan dengan tombol lainnya.



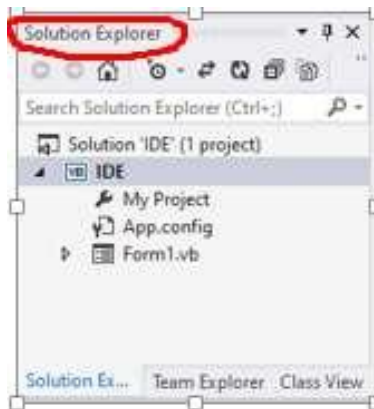
- **RichTextBox**,  **RichTextBox** : menampilkan fasilitas pengeditan teks dengan pemformatan yang kaya dan fitur lain seperti pemformatan karakter dan paragraf.
- **SaveFileDialog**,  **SaveFileDialog** : menampilkan kotak dialog yang meminta pengguna memilih lokasi tertentu untuk menyimpan file.
- **SerialPort**,  **SerialPort** : merepresentasikan sumber daya serial port.
- **ServiceController**,  **ServiceController** : menyediakan kemampuan untuk berkomunikasi, query, manipulasi, dan menghentikan layanan windows service.
- **SplitContainer**,  **SplitContainer** : membagi area display dari kontainer menjadi dua panel yang bisa diatur ukurannya dimana Anda bisa menambahkan kontrol ke dalamnya.
- **Splitter**,  **Splitter** : membagi area display dari container menjadi dua panel yang bisa diatur ulang ukurannya.
- **StatusStrip**,  **StatusStrip** : menampilkan informasi ke pengguna tentang objek yang sedang dilihat, komponen-komponen objek atau operasi objek.
- **TabControl**,  **TabControl** : mengatur dan menampilkan kumpulan tab dimana tiap tab bisa memiliki kontrol dan komponen.

- **TableLayoutPanel**,  `TableLayoutPanel` : menangani layout dari komponen-komponen, kemudian mengaturnya dalam format tabel secara otomatis.
- **TextBox**,  `TextBox` : memungkinkan pengguna memasukkan teks yang bisa lebih dari satu baris dan menyediakan kemampuan pengeditan dan masking kata sandi.
- **Timer**,  `Timer` : komponen yang mengeksekusi event pada interval-interval yang dapat ditentukan oleh pengguna.
- **ToolStrip**,  `ToolStrip` : komponen yang menyediakan toolbar dan elemen antarmuka lain yang dapat mendukung berbagai opsi tampilan. Selain itu bisa mendukung overflow dan pengaturan ulang saat program berjalan.
- **ToolStripContainer**,  `ToolStripContainer` : menampilkan panel di setiap sisi form yang bisa mengandung satu atau lebih komponen seperti ToolStrip, MenuStrip, atau StatusStrip.
- **ToolTip**,  `ToolTip` : menampilkan informasi ketika pengguna menggerakkan pointer di atas kontrol tertentu.
- **TrackBar**,  `TrackBar` : memungkinkan pengguna memilih antara kisaran nilai dengan cara menggerakkan penanda di bilah tertentu.

- **TreeView**,  **TreeView** : menampilkan koleksi hierarki item-item yang memiliki label yang nantinya bisa diberi gambar.
- **VscrollBar**,  **VScrollBar** : memungkinkan komponen induknya untuk menggulirkan isi di dalamnya secara vertikal. Komponen ini tidak diperlukan jika komponen induknya sudah memiliki atribut Autoscroll yang aktif.
- **WebBrowser**,  **WebBrowser** : memungkinkan pengguna membuka halaman web dari dalam form.





#### E. Solution Explorer Visual Studio .Net 2013

Solution explorer adalah bagian IDE Visual Studio .Net 2013 yang berada di sebelah kanan atas. Solution explorer memberikan tampilan daftar file-file dari proyek yang sedang dibuat. Pada jendela solution explorer terdapat beberapa tombol tree yang berisi daftar dari file-file yang digunakan dalam proyek. Jika ingin **menampilkan solution explore** caranya dengan pilih menu **View > Solution Explorer** atau tekan **Ctrl + Alt + L**. Tampilan Solution Explorer dapat dilihat pada gambar 1.25.



Gambar 1.25. Solution Explorer Visual Studio .Net 2013

## F. Properties Window Visual Studio .Net 2013

Properties windows adalah bagian IDE Visual Studio .Net 2013 yang berada di sebelah kanan bawah. Properties window adalah tempat menyimpan properti dari setiap obyek tool dan komponen. Properties window juga digunakan untuk mengubah property dari obyek tool dan komponen yang sedang dipakai dalam form. Pengguna dapat mengurutkan property berdasarkan kategori (**klik icon Categorized pada Properties Window, **) dan alphabet (**klik icon Alphabetical pada Properties Window, **). Di bagian properties window, pengguna dapat menampilkan property (**klik icon Property pada Properties Window, **) dan event (**klik icon Event pada Properties Window, **) dari suatu tool dan komponen. Cara menampilkan Properties Window yaitu dengan cara pilih menu **View > Properties Window** atau tekan

**F4.** Tampilan Properties Windows dapat dilihat pada gambar 1.26.



Gambar 1.26. Properties Window Visual Studio .Net 2013

Ada properti untuk beberapa tool yang sama, seperti Name, Autosize, dan sebagainya, namun banyak pula properti yang khusus dimiliki oleh tool tertentu saja. Berikut ini merupakan contoh tampilan beberapa **property windows** untuk **tool form** dan arti dari nilai propertinya :

- ✚ **BackColor**, BackColor  : menentukan warna background dari komponen.
- ✚ **Font**, Font Microsoft Sans Se : menentukan jenis dan ukuran font yang akan ditampilkan di kontrol.
- ✚ **ForeColor**, ForeColor  : menentukan warna foreground / depan dari komponen yang digunakan untuk menampilkan teks di komponen tersebut.

- ✚ **MaximizeBox**, `MaximizeBox True` : menentukan apakah form memiliki kotak Maximize di bagian kanan atas dari caption bar
- ✚ **MaximumSize**, `MaximumSize 0; 0` : menentukan ukuran maksimal dari form yang bisa di-resize
- ✚ **MinimizeBox**, `MinimizeBox True` : menentukan apakah form memiliki kotak Minimize di bagian kanan atas dari caption bar.
- ✚ **MinimumSize**, `MinimumSize 0; 0` : menentukan ukuran minimal dari form yang bisa di-resize.

#### 1.4.2. Kode Editor Visual Studio .Net 2013

Kode Editor adalah tempat untuk menuliskan kode program dari aplikasi program yang sedang dibuat. Tampilan Kode Editor dapat dilihat pada gambar 1.27.



Gambar 1.27. Kode Editor

Pada gambar 1.27 terlihat bahwa Kode Editor memiliki tiga bagian yaitu bagian letak penulisan kode program, obyek / tool yang berisi tool yang berada di form atau tool yang sedang digunakan saat membuat aplikasi

program, dan event yang berisi event / kejadian dari masing-masing tool. Pada obyek / tool dan event dapat dipilih oleh pengguna sesuai dengan keinginan, defaultnya saat pengguna akan memberikan Kode program pada suatu tool dengan cara double klik pada tool tersebut maka Visual Basic .Net 2013 akan memberikan event secara default sesuai dengan sistem. Misalnya untuk tool Button maka event yang diberikan adalah klik. Pada letak penulisan Kode program terdapat awal Kode program dan akhir Kode program. Satu prosedur Kode program dinyatakan sebagai awal Kode program hingga akhir Kode program. Untuk menampilkan Kode Editor yaitu dengan cara pilih menu **View > Kode** atau tekan **F7** atau **double klik pada salah satu tool di form yang akan diberi Kode program**. Untuk kembali pada tampilan designer yaitu dengan cara **View > Designer**.

**Event** adalah suatu kejadian yang diterima oleh suatu kontrol, misalnya click, dblclick, keypress, dan sebagainya. Berikut ini event yang sering digunakan oleh pemakai program yaitu :

### **1. Event untuk mouse**

- **Klik** : event ini terjadi bila tombol kiri mouse ditekan dan dilepas dengan cepat saat posisi pointer berada di atas obyek.
- **DblClick** : event ini terjadi bila tombol kiri mouse ditekan dan dilepas dengan cepat sebanyak dua kali saat posisi pointer berada di atas obyek.

- **DragDrop** : event ini terjadi bila tombol kiri mouse ditekan dan ditahan kemudian menyeret/menggeser obyek dari satu tempat ke tempat lain, kemudian melepas tombol kiri mouse tersebut.
- **DragOver** : hampir sama dengan DragDrop tetapi DragOver ini biasanya digunakan untuk mengubah bentuk tampilan pointer mouse saat obyek diseret.
- **MouseDown** : event ini terjadi bila tombol kiri mouse ditekan dan ditahan.
- **MouseUp** : event ini terjadi bila bila tombol kiri mouse dilepas sehabis ditekan.
- **MouseMove** : event ini terjadi bila mouse dipindah posisinya ke tempat lain tanpa menekan tombol mouse tersebut.

## 2. Event untuk keyboard

- **KeyPress** : event ini terjadi bila sebuah tombol keyboard ditekan. Nilai parameter dari KeyPress adalah kode ASCII untuk menyatakan jenis tombol keyboard yang ditekan.
- **KeyDown** : event ini terjadi bila Anda menekan dan menahan sebuah tombol keyboard.
- **KeyUp** : event ini terjadi bila Anda melepaskan sebuah tombol keyboard.

## 3. Event untuk perubahan

- **Activate** : event ini terjadi bila sebuah form menjadi window yang aktif.
- **Deactive** : event ini terjadi ketika Anda berpindah dari satu form ke form yang lain.



- **GotFocus** : event ini terjadi bila sebuah obyek menjadi satu-satunya fokus. Sebuah form dapat membuat obyek fokus bila form tersebut menjadi window aktif.
- **LostFocus** : event ini terjadi bila sebuah obyek kehilangan fokus karena ada obyek lain yang mendapatkan fokus.
- **Load** : event ini terjadi bila sebuah form dipanggil atau dibuka.
- **Unload** : event ini terjadi bila sebuah form ditutup.
- **Initialize** : event ini terjadi bila semua referensi untuk form atau class dihapus dari memori komputer. Pada obyek form, event ini terjadi setelah event unload.
- **Paint** : event ini terjadi bila sebuah form perlu digambar ulang. Biasanya dilakukan saat sebuah form dipindahkan dari form lain yang menutupinya.
- **Resize** : event ini terjadi bila sebuah form diubah ukurannya.
- **Change** : event ini terjadi bila isi dari sebuah kontrol diubah.
- **QueryUnload** : event ini terjadi apabila suatu aplikasi ditutup. Biasanya digunakan untuk memastikan bahwa semua yang berhubungan dengan aplikasi juga telah ditutup, atau memastikan bahwa data telah disimpan sebelum aplikasi ditutup.

#### 4. Event-event lainnya

- **Timer** : event ini terjadi ketika event lain dijalankan sebelum event timer ini. Event ini hanya berhubungan dengan kontrol time.
  - **Scroll** : event ini terjadi saat Anda menyeret kotak kecil pada scroll bar. Event ini hanya berhubungan dengan kontrol scrol bar.
  - **PathChange** : event ini terjadi pada saat standar path untuk mencari suatu file diubah. Event ini hanya ada pada kontrol file listbox.
  - **PatterChange** : event ini terjadi apabila kode pencarian daftar file diubah, misalnya \*.\* diubah menjadi \*.jpg.
  - **Validate** : event ini digunakan untuk memastikan data yang telah ditangani dengan baik.
  - **SelChange** : event ini terjadi saat kontrol grid sedang digunakan.
5. **Event form** yang sering dijumpai adalah :
- ❖ **Activated** : event yang dipicu ketika form diaktifkan.
  - ❖ **Closed** : event yang dipicu ketika form sudah ditutup.
  - ❖ **Closing** : event yang dipicu ketika form akan ditutup.
  - ❖ **Deactivate** : event yang dipicu ketika form tidak diaktifkan.
  - ❖ **InputLanguageChanged** : event yang dipicu ketika input bahasa dan form akan diubah.

- ❖ **InputLanguageChanging** : event yang dipicu ketika input bahasa dari form akan diubah
- ❖ **Load** : event yang dipicu ketika form load pertama kali.
- ❖ **MaximizedBoundsChanged** : event yang dipicu ketika nilai dari properti MaximizedBounds sebuah form diubah.
- ❖ **MaximizedSizeChanged** : event yang dipicu ketika nilai properti MaximizedSize yang mendefinisikan ukuran maksimal diubah.
- ❖ **MdiChildActive** : event yang dipicu ketika jendela MDI anak diaktifkan.
- ❖ **MenuComplete** : event yang dipicu ketika proses seleksi menu telah sempurna.
- ❖ **MenuStart** : event yang dipicu ketika menu ditampilkan
- ❖ **MinimumSizeChanged** : event yang dipicu ketika nilai dari properti MinimumSize dari form diubah.

#### A. Penulisan Kode Program

Untuk menuliskan Kode program yang akan ditampilkan pada tool tertentu formatnya adalah **Nama\_ToolBox.Properties\_ToolBox = Diisi Apa**. Contohnya, apabila ingin menampilkan isian data dari text yang ada tool lain maka penulisannya sebagai berikut : **Nama\_ToolBox.Text = Nama\_ToolBox.Text**. Contoh : `TextBox1.text = Combobox1.text`. Untuk menghapus isian data yang berada pada suatu toolbox maka penulisannya adalah :

**Nama\_ToolBox.Properties\_ Toolbox = ""**. Contoh :  
 TextBox1.text = ""

Untuk mengisi isian menggunakan data bertipe **String** maka gunakan tanda petik dua (""), jika ingin mengisi isian menggunakan data bertipe **Integer** maka tidak perlu menambahkan apapun jika ingin mengisi isian menggunakan data bertipe **Date** maka gunakan tanda kres (##). Contoh :

TextBox1.text = "Pemrograman Dasar" → Pengisian data bertipe String

TextBox1.text = 1234 → Pengisian data bertipe Integer

TextBox1.text = ## → Pengisian data bertipe Date

## B. Penulisan komentar

Komentar biasanya ditulis dengan tujuan untuk memberikan penjelasan mengenai kode program yang ditulis agar orang lain atau programmer lain mengetahui maksud dari jalannya kode program yang ditulis jika program kita dikembangkan oleh orang lain. Tulisan komentar ini tidak akan dieksekusi oleh Visual Basic .Net 2013. Ada 2 cara dalam penulisan program, cara Penulisan Komentar yaitu :

Cara **pertama**, Komentar adalah penjelasan dari suatu kode program. Komentar ini merupakan sebuah kata atau kalimat yang tidak akan dijalankan oleh VB.Net. Untuk penulisan **komentar** ditandai dengan **tanda petik tunggal ( ' )**. Jadi kata atau kalimat setelah tanda petik tunggal akan diabaikan oleh program VB.Net atau tidak akan dijalankan oleh program VB.Net.

Cara **kedua**, dengan menekan tombol shourcut **Comment** yang ada di menu bar untuk menambahkan komentar (menambahkan petik satu) , dan tombol shourcut **Uncomment** untuk menghilangkan komentar

(menghilangkan petik satu) . Kedua tombol ini hanya akan muncul saat pengguna berada pada tampilan Kode editor.

#### 1.4.3. Output Window dan Error List Window

Output window ini menunjukkan aktivitas saat Visual Basic .Net 2013 dalam mengompilasi aplikasi. Jadi, output window ini akan terlihat jika pengguna sedang menjalankan programnya, sesaat sebelum form muncul. Letak output window ini berada di bawah Kode editor. Tampilan output window dapat dilihat pada gambar 1.28. Pada gambar 1.28 dapat dilihat isi dari output window ini adalah keterangan dari program aplikasi yang telah dibuat yaitu tempat program aplikasi ini disimpan. Isi dari output window ini beragam tergantung dari kompleksitas dari program aplikasi yang dibuat.



Gambar 1.28. Output Window

Jika pengguna dalam menuliskan Kode program mengalami kesalahan, maka Visual Basic .Net 2013 akan menunjukkan kesalahan penulisan Kode dengan cara memberi tanda garis bawah bergelombang (biasa disebut “cacing”) pada Kode program yang salah. Apabila ini terjadi maka pengguna harus mengubah Kode program yang salah menjadi benar agar “cacing”-nya hilang. Berikut contoh penulisan Kode program yang salah sehingga muncul “cacing” pada Kode program tersebut.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load  
    yy()  
End Sub
```

Apabila pengguna menjalankan program aplikasi yang dibuat saat terjadi kesalahan penulisan Kode program, maka kesalahan tersebut akan muncul di Error List Window. Error List Window ini berisi daftar kesalahan pengguna dikarenakan kesalahan penulisan Kode program. Untuk mencari letak kesalahan penulisan Kode program, maka pengguna dapat melakukan double-click pada salah satu daftar kesalahan pada Error List Window dan secara otomatis Visual Basic .Net 2013 akan menunjukkan letak kesalahan tersebut. Tampilan Error List Window ditampilkan pada gambar 1.29. Letak Error List Window berada di bawah Kode editor.



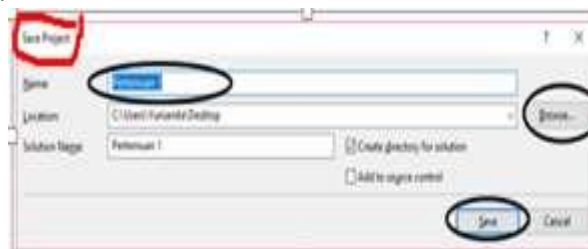
Gambar 1.29. Error List Window

## 1.5. Menyimpan Project, Menjalankan Project, Menambahkan Form Baru, dan Membuka Project

Pada subbab ini akan dijelaskan tentang menyimpan project, menjalankan project, menambahkan form baru, dan membuka project.

### 1.5.1. Menyimpan Project

Untuk menyimpan seluruh project yang telah dibuat, klik **File** → **Save All**. Lalu atur nama pada isian **Name** dan tempat penyimpanan dengan cara klik **Browse** lalu atur tempat penyimpanannya → klik **Save**. Atau tekan tombol **Save All** di menu bar. Lalu akan muncul kotak dialog seperti yang terlihat pada gambar 1.30. Pada kotak dialog tersebut, isikan **Name** yaitu nama Project, **Location** yaitu tempat penyimpanan project tersebut tersimpan (pengguna bisa mencari letak penyimpanan project dengan cara menekan tombol **Browse**), lalu klik **Save**. Untuk diketahui bahwa isian pada Solution Name akan terisi otomatis saat pengguna mengetikkan isian Name dan isi pada Solution Name sama dengan isi pada Name. Sebagai contoh, pada gambar 1.30, sebuah project disimpan dengan nama “Pertemuan 1” dan disimpan di folder C:\.



Gambar 1.30. Menyimpan Project

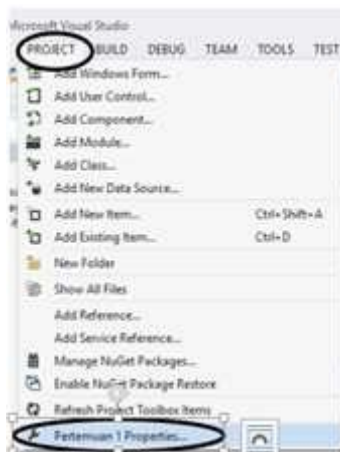
Untuk menyimpan project per form, **klik File → Save Form Form1.vb**. Atau tekan **ctrl + S**. Atau tekan tombol Save di menu bar.

### 1.5.2. Menjalankan Project

Untuk menjalankan project yang telah dibuat, klik **Start** pada menu bar (gambar panah dengan warna hijau, ) atau tekan **F5**.

### A. Mengganti Form Yang Akan Dijalankan

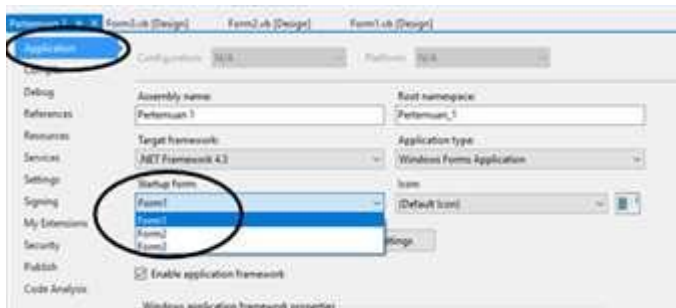
Pada visual basic, saat programmer menjalankan project, form yang dijalankan pertama kali adalah form 1. Jika ingin merubah form yang dijalankan selain form 1, maka caranya adalah klik **Project → klik Pertemuan 1 Properties** (nama projectnya apa). Pastikan **semua project telah disimpan** sebelum melakukan langkah ini. Ikuti langkah seperti gambar 1.31 dan 1.32.





Gambar 1.31. Langkah 1 Mengganti Form yang Dijalankan

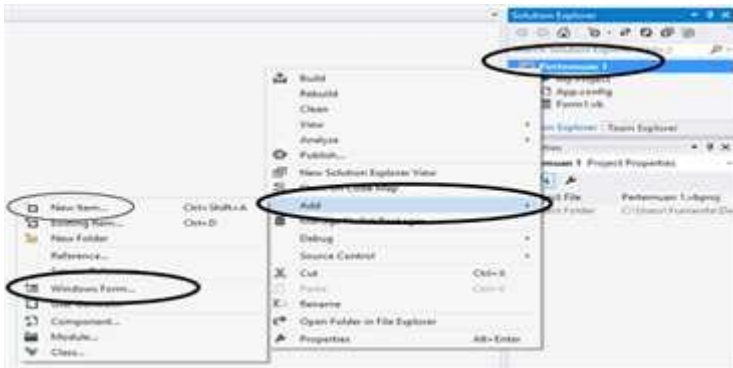
Lalu muncul kotak dialog berikutnya. Pada tampilan tersebut, Pilih **Application** → pada menu **Statup Form** pilih form lain yang akan dijalankan pertama kali → klik **Save All**.



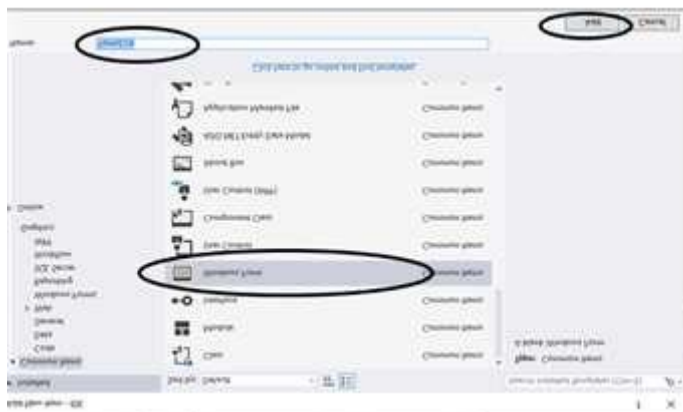
Gambar 1.32. Langkah 2 Mengganti Form yang Akan Dijalankan

### 1.5.3. Menambahkan Form Baru

Untuk menambahkan form, ada dua cara, cara pertama yaitu dengan cara **klik kanan pada project yang dibuat yang terletak di solution explorer** → **klik Add** → **klik Windows Form** atau **New Item**. Langkah menambahkan form baru cara pertama ditampilkan pada gambar 1.33. Lalu pilih **Windows Form** dan **ketikkan nama** form yang akan dibuat pada isian **Name**. Lalu klik **Add**. Langkah selanjutnya ditampilkan pada gambar 1.34.

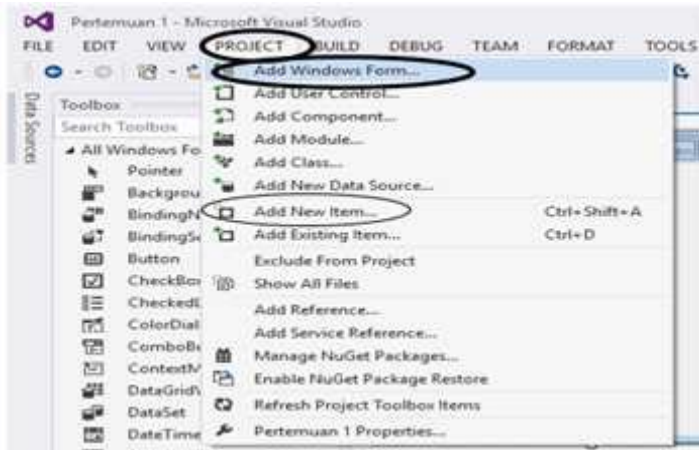


Gambar 1.33. Menambahkan Form Baru Cara 1  
 Untuk menambahkan form cara kedua yaitu dengan cara klik **Project** → **Add Windows Form** atau **Add New Item**.



Gambar 1.34. Langkah Selanjutnya Menambahkan Form Cara 1 dan 2

Langkah menambahkan form baru cara kedua ditampilkan pada gambar 1.35.



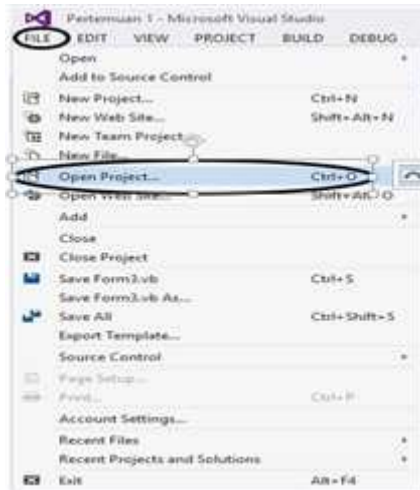
Gambar 1.35. Menambahkan Form Baru Cara 2

Lalu muncul kotak dialog penambahan item. Pada tampilan tersebut, pilih **Windows Form** → **ketik nama** form pada isian **Name** → klik **Add**. Langkah selanjutnya ditampilkan pada gambar 1.34.

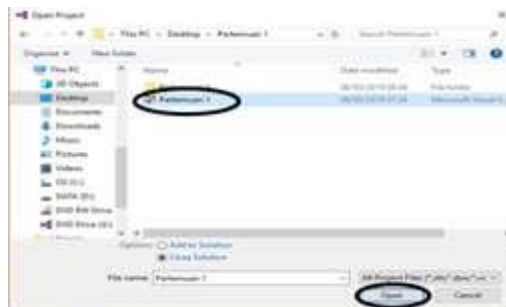
#### 1.5.4. Membuka Project

Langkah untuk membuka project yang telah dibuat sebelumnya, maka caranya adalah **klik File** → **Open Project** atau tekan **Ctrl + O**. Langkah ini ditampilkan pada gambar 1.36.

Lalu muncul kotak dialog untuk memilih folder mana yang akan dibuka. Lalu folder yang berisi project yang akan dibuka, lalu **klik icon yang bergambar visual studio 2013** → klik **Open**. Langkah ini ditampilkan pada gambar 1.37.



Gambar 1.36. Langkah 1 Membuka Project



Gambar 1.37. Langkah 2 Membuka Project

## 1.6. Latihan Program

Pada latihan program ini terdapat dua latihan yaitu latihan 1 dan latihan 2.

### 1.6.1. Latihan Program 1

Gambar 1.38. Desain Form Latihan 1

Pada latihan 1 ini diminta untuk membuat form data diri. User memasukkan data diri pada isian data diri, lalu user menekan tombol OK maka data pada isian data diri ditampilkan pada hasil data diri. Tombol bersil digunakan untuk memberihkan data diri, baik data diri yang ada pada isian data diri maupun pada hasil data diri. Berikut tampilan form pada latihan 1 ditampilkan pada gambar 1.38. Berikut tool pada form latihan 1 yang diatur ditampilkan di tabel 1.1.

Tabel 1.1. Tabel Tool pada Form Latihan 1

<b>Tool/Control</b>	<b>Properties</b>	<b>Setting</b>
Form 1	BackColor	192; 255; 255
	Text	Data Diri
GroupBox 1	Text	Isi Data Diri
GroupBox 2	Text	Hasil Data Diri
Label 1	Text	DATA DIRI
	Font	Magneto, 14.25 Bold
	ForeColor	White
Label 2	Text	<i>Nim</i>

Label 3	Text	Nama
Label 4	Text	Alamat
Label 5	Text	Jenis Kelamin
Label 6	Text	Jurusan
Label 7	Text	No HP
Label 8	Text	Tanggal Lahir
Label 9	Text	Nim
Label 10	Text	Nama
Label 11	Text	Alamat
Label 12	Text	Jenis Kelamin
Label 13	Text	Jurusan
Label 14	Text	No HP
Label 15	Text	Tanggal Lahir
Label 16	Text	Nim
	Name	lblNim
Label 17	Text	Nama
	Name	lblNama
<b>Tool/Control</b>	<b>Properties</b>	<b>Setting</b>
Label 18	Text	Alamat
	Name	lblAlamat
Label 19	Text	Jenis Kelamin
	Name	lblJk
Label 20	Text	Jurusan
	Name	lblJurusan
Label 21	Text	No HP
	Name	lblHp
Label 22	Text	Tanggal Lahir
	Name	lblTglLhr
TextBox 1	Name	txtNim
TextBox 2	Name	txtNama
TextBox 3	Name	txtAlamat
	Multiline	True
RadioButton 1	Name	rdbLaki
	Text	Laki-Laki
RadioButton 2	Name	rdbPr
	Text	Perempuan
ComboBox 1	Name	cbJurusan
	Items	MIK

		Hukum Bidan
	Text	Pilih Jurusan
MaskedTextBox x 1	Name	txtHp
	Mask	Custom, diisi 0000- 0000-0000
DateTimePicker	Name	dtpTglLhr
Button 1	Name	btnOk
	Text	OK
Button 2	Name	btnBersih
	Text	Bersih

Berikut kode program yang harus ditambahkan agar program berjalan sesuai dengan yang diinginkan adalah :

- Tambahkan Kode program berikut pada button Ok dengan nama btnOk. Kode program ini digunakan untuk menampilkan semua isian ke tool lain yaitu label.

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles
btnOk.Click
    lblNim.Text = txtNim.Text
    lblNama.Text = txtNama.Text
    lblAlamat.Text = txtAlamat.Text
    lblJurusan.Text = cbJurusan.Text
    lblHp.Text = txtHp.Text
    lblTglLhr.Text = dtpTglLahir.Text
    If rdbLaki.Checked = True Then
        lblJk.Text = rdbLaki.Text
    Else
        lblJk.Text = rdbPr.Text
    End If
End Sub
```

- Tambahkan Kode program berikut pada button Bersih dengan nama btnBersih. Kode program ini digunakan

untuk membersihkan isian pada semua textbox, label, dan mengosongkan check pada radiobutton.

```
Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles  
btnBersih.Click  
    lblNim.Text = ""  
    lblNama.Text = ""  
    lblAlamat.Text = ""  
    lblJurusan.Text = ""  
    lblHp.Text = ""  
    lblTglLhr.Text = ""  
    lblJk.Text = ""  
    txtNim.Text = ""  
    txtNama.Text = ""  
    txtAlamat.Text = ""  
    cbJurusan.Text = "Pilih Jurusan"  
    txtHp.Text = ""  
    dtpTglLahir.Text = Now  
    rdbLaki.Checked = False  
    rdbPr.Checked = False  
End Sub
```



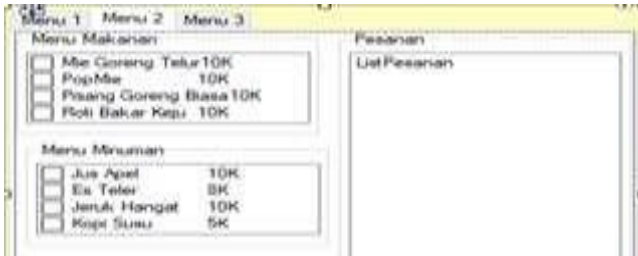
### 1.6.2. Latihan Program 2

Pada latihan 2 ini menampilkan menu makanan dan minuman dengan menggunakan beberapa macam tool diantaranya menu makanan dan minuman yang ditampilkan pada tool checkbox, checkedListBox, dan listBox. Menu makanan dan minuman ini nantinya akan dipilih oleh user dan kemudian akan ditampilkan pada menu pesanan, menu pesanan ini ditampilkan menggunakan tool textbox dan listBox. Berikut tampilan form pada latihan 2 dapat dilihat pada gambar 1.39, gambar 1.40, dan gambar 1.41.

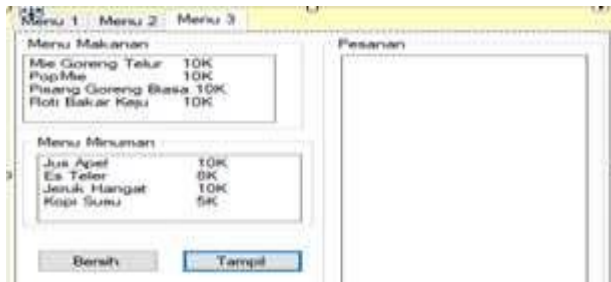


Gambar 1.39. latihan 2 Tab 1 (Menu 1)

Tampilan form pada latihan 2 tab 1 (Menu 1) ditampilkan pada gambar 1.39, tampilan form pada latihan 2 tab 2 (Menu 2) ditampilkan pada gambar 1.40, dan tampilan form pada latihan 2 tab 3 (Menu 3) ditampilkan pada gambar 1.41.



Gambar 1.40. Gambar latihan 2 Tab 2 (Menu 2)



Gambar 1.41. latihan 3 Tab 3 (Menu 3)

Berikut tool pada form latihan 2 yang harus diatur :

Tabel 1.2. Tabel Tool pada Form Latihan 2 tab 1, 2, 3.

Tool/Control	Propertie s	Setting
Form 1	BackColor	255; 255; 192
	Text	Menu Pesanan Makanan dan Minuman
PictureBox 1	Image	Pilih gambar sendiri
	SizeMode	StretchImage
Label 1	Text	Menu Makanan dan Minuman Barcelona 2
	Font	Palatino Linotype; 14,25pt; style=Bold
TabControl 1 Klik <b>OK</b> , jika sudah menambahkan TabPage	TabPage 1	Text : Menu 1
	TabPage 2	Text : Menu 2
	TabPage 3	Text : Menu 3
		NB : untuk <b>menambahk</b> <b>an</b> TabPage klik <b>Add</b> dan untuk <b>menghapus</b> klik <b>Remove</b>

<b>Tool pada Tab 1 / Menu 1 yaitu :</b>		
GroupBox 1	Text	Menu Makanan
GroupBox 2	Text	Menu Minuman
GroupBox 3	Text	Menu Pesanan
CheckBox 1	Name	cbMie
	Text	Mie Goreng Telur 10K
CheckBox 2	Name	cbPop
	Text	PopMie 10K
CheckBox 3	Name	cbPisang
	Text	Pisang Goreng Biasa10K
CheckBox 4	Name	cbRoti
	Text	Roti Bakar Keju 10K
CheckBox 5	Name	cbJus
	Text	Jus Apel 10K
CheckBox 6	Name	cbEs
	Text	Es Teler 8K
CheckBox 7	Name	cbJeruk
	Text	Jeruk Hangat 10K
CheckBox 8	Name	cbKopi
	Text	Kopi Susu 5K
TextBox 1	Name	txtPesanan
Button 1	Name	btnTampil
	Text	Tampil
Button 2	Name	btnPesan
	Text	Tampil Pesan
Button 3	Name	btnBersih
	Text	Bersih
<b>Tool pada Tab 2 / Menu 2 yaitu :</b>		
GroupBox 4	Text	Menu Makanan
GroupBox 5	Text	Menu Minuman
GroupBox 6	Text	Menu Pesanan
CheckedListBox 1	Name	clMakanan
	Items	Mie Goreng Telur 10K PopMie 10K Pisang Goreng Biasa 10K Roti Bakar Keju 10K
	Name	clMinuman
	Items	Jus Apel 10K Es Teler 8K Jeruk Hangat 10K

		Kopi Susu 5K
ListBox 1	Name	ListPesanan
<b>Tool pada Tab 3 / Menu 3 yaitu :</b>		
GroupBox 7	Text	Menu Makanan
GroupBox 8	Text	Menu Minuman
GroupBox 9	Text	Menu Pesanan
ListBox 2	Name	listMakanan
	Selection Mode	MultiExtended
ListBox 3	Name	listMinuman
	Selection Mode	MultiExtended
TextBox 2	Name	txtPesanan2
Button 4	Name	btnTampil2
	Text	Tampil
Button 5	Name	btnBersih2
	Text	Bersih

Kode program yang harus ditambahkan agar program berjalan sesuai dengan yang diinginkan adalah :

- Tambahkan Kode program berikut pada CheckedListBox Minuman dengan nama clMinuman. Kode program ini digunakan untuk menampilkan menu minuman berupa CheckedListBox ke menu pesanan berupa listbox. **PILIH EVENT ITEMCHECK.**

```
Private Sub clMinuman_ItemCheck(sender As Object, e As ItemCheckEventArgs)
Handles clMinuman.ItemCheck
    'untuk menambahkan dan menghilangkan data checkBoxListBox makanan pada
    textbox pesanan
    Dim itemMinuman As String = clMinuman.SelectedItem
    If e.NewValue Then
        ListPesanan.Items.Add(itemMinuman)
    Else
        ListPesanan.Items.Remove(itemMinuman)
    End If
End Sub
```

- Tambahkan Kode program berikut pada CheckedListBox Makanan dengan nama clMakanan. Kode program ini

digunakan untuk menampilkan menu makanan berupa CheckedListBox ke menu pesanan berupa listBox. PILIH

### **EVENT ITEMCHECK.**

```
Private Sub cIMakanan_ItemCheck1(sender As Object, e As ItemCheckEventArgs)
Handles cIMakanan.ItemCheck
    Dim itemMakanan As String = cIMakanan.SelectedItem
    If e.NewValue Then
        ListPesanan.Items.Add(itemMakanan)
    Else
        ListPesanan.Items.Remove(itemMakanan)
    End If
End Sub
```

- Tambahkan Kode program berikut pada button Tampil dengan nama btnTampil. Kode program ini digunakan untuk menampilkan menu makanan dan minuman berupa checkbox ke menu pesanan berupa textbox.

```
Private Sub btnTampil_Click(sender As Object, e As EventArgs) Handles
btnTampil.Click
    If cbMie.Checked = True Then
        txtPesanan.Text = txtPesanan.Text & cbMie.Text & vbCrLf
    End If
    If cbPop.Checked = True Then
        txtPesanan.Text = txtPesanan.Text & cbPop.Text & vbCrLf
    End If
    If cbPisang.Checked = True Then
        txtPesanan.Text = txtPesanan.Text & cbPisang.Text & vbCrLf
    End If
    If cbRoti.Checked = True Then
        txtPesanan.Text = txtPesanan.Text & cbRoti.Text & vbCrLf
    End If
    If cbJus.Checked = True Then
        txtPesanan.Text += cbJus.Text & vbCrLf
    End If
    If cbEs.Checked = True Then
        txtPesanan.Text += cbEs.Text & vbCrLf
    End If
    If cbJeruk.Checked = True Then
        txtPesanan.Text += cbJeruk.Text & vbCrLf
    End If
    If cbKopi.Checked = True Then
        txtPesanan.Text += cbKopi.Text & vbCrLf
    End If
End Sub
```

End Sub

- Tambahkan Kode program berikut pada button Tampil Pesan dengan nama btnPesan. Kode program ini digunakan untuk menampilkan menu makanan dan minuman berupa checkbox ke menu pesanan berupa kotak dialog pesan /

message box.

```
Private Sub btnPesan_Click(sender As Object, e As EventArgs) Handles
btnPesan.Click
    'penamaan variabel
    Dim pesanan As String
    'variabel pesanan diisi kosong
    pesanan = ""
    'untuk menambahkan checkbox makanan dan minuman pada pesan
    If cbMie.Checked = True Then
        pesanan = pesanan & cbMie.Text & vbCrLf
    End If
    If cbPop.Checked = True Then
        pesanan = pesanan & cbPop.Text & vbCrLf
    End If
    If cbPisang.Checked = True Then
        pesanan = pesanan & cbPisang.Text & vbCrLf
    End If
    If cbRoti.Checked = True Then
        pesanan = pesanan & cbRoti.Text & vbCrLf
    End If
    If cbJus.Checked = True Then
        pesanan += cbJus.Text & vbCrLf
    End If
    If cbEs.Checked = True Then
        pesanan += cbEs.Text & vbCrLf
    End If
    If cbJeruk.Checked = True Then
        pesanan += cbJeruk.Text & vbCrLf
    End If
    If cbKopi.Checked = True Then
        pesanan += cbKopi.Text & vbCrLf
    End If
    'Menampilkan makanan dan minuman dalam bentuk pesan
    MsgBox("=====Menu Pesanan======" & vbCrLf & pesanan)
End Sub
```

End Sub

- Tambahkan Kode program berikut pada button Bersih dengan nama btnBersih. Kode program ini digunakan untuk

menghilangkan menu makanan dan minuman di menu pesanan berupa textbox dan menghilangkan centang pada tool checkbox

```
Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles btnBersih.Click
    'menghilangkan menu makanan dan minuman pada menu pesanan
    txtPesanan.Text = ""
    'menghilangkan centang pada checkbox
    cbMie.Checked = False
    cbPop.Checked = False
    cbPisang.Checked = False
    cbRoti.Checked = False
    cbJus.Checked = False
    cbEs.Checked = False
    cbJeruk.Checked = False
    cbKopi.Checked = False
End Sub
```

- Tambahkan Kode program berikut pada button Tampil dengan nama btnTampil2. Kode program ini digunakan untuk menampilkan menu makanan dan minuman berupa listbox di menu pesanan berupa textbox.

```
Private Sub btnTampil2_Click(sender As Object, e As EventArgs) Handles btnTampil2.Click
    Dim pesanan As String
    Dim i, j As Integer
    'untuk menampilkan list makanan pada textbox pesanan
    For i = 1 To listMakanan.SelectedItems.Count
        pesanan = pesanan + listMakanan.SelectedItems.Item(i - 1) + vbCrLf
    Next
    'untuk menampilkan list minuman pada textbox pesanan
    For j = 1 To ListMinuman.SelectedItems.Count
        pesanan = pesanan + ListMinuman.SelectedItems.Item(j - 1) + vbCrLf
    Next
    txtPesanan2.Text = pesanan
End Sub
```

- Tambahkan Kode program berikut pada button Bersih dengan nama btnBersih2. Kode program ini digunakan untuk menghilangkan menu makanan dan minuman di menu pesanan berupa textbox dan menghilangkan select pada tool listbox.

```
Private Sub btnBersih2_Click(sender As Object, e As EventArgs) Handles btnBersih2.Click
    txtPesanan2.Text = ""
```

```
'Menghilangkan pilihan pada listbox makanan dan minuman
listMakanan.SelectionMode = SelectionMode.None
ListMinuman.SelectionMode = SelectionMode.None
End Sub
```

## BAB II ELEMEN-ELEMEN PROGRAM

Pada bab 2 ini menjelaskan tentang variabel, tipe data, konstanta, operator dan operand, latihan program, dan tugas.

### 2.1. Pendahuluan

Suatu data sangat terkait dengan program, karena **data merupakan nilai yang dibutuhkan oleh aplikasi**, sedangkan **program merupakan instruksi yang digunakan untuk mengolah data tersebut**. Jadi, data dan program merupakan unsur-unsur utama dalam membuat suatu program aplikasi yang sempurna, perhatikan gambar 2.1 berikut.



Gambar 2.1. Keterkaitan Data dan Program [Yung:2005]

### 2.2. Variabel

**Variabel adalah suatu tempat dalam memori komputer yang digunakan untuk menyimpan sementara data-data yang dioperasikan. Variabel ini mempunyai nama serta disesuaikan dengan tipe data yang disimpan** [Yung:2005].

#### 2.2.1. Syarat Penulisan Variabel

Ada beberapa **syarat** yang harus dipenuhi dalam **menuliskan nama suatu variabel**, yaitu [Yung:2005] :



- Harus dimulai dengan huruf (abjad).
- Nama variabel harus unik atau tidak boleh sama dalam sebuah lingkup variabel.
- Tidak boleh menggunakan kata kunci, objek, properti, dan method visual basic.
- Tidak boleh ada spasi.
- Yang diperbolehkan dalam penamaan variabel adalah huruf, angka, dan garis bawah ( \_ ).
- Panjang nama variabel tidak boleh lebih dari 255 karakter.

### 2.2.2. Deklarasi Variabel

Suatu variabel yang akan digunakan harus dideklarasikan terlebih dahulu. Pada saat mendeklarasikan variabel, user “meminta” kepada sistem operasi untuk mengalokasikan tempat di memori sesuai dengan kapasitas tipe data yang bersangkutan.

Sesuai dengan data yang ditampung, suatu variabel harus mempunyai tipe data yang sesuai dengan isinya. Perintah umum yang digunakan dalam pendeklarasian variabel adalah **dim** atau **dim** diikuti **as**. Penulisan umum dalam pendeklarasian variabel adalah :

**Dim nama\_variabel** → contoh : dim nama

**Dim nama\_variabel As Tipe\_Data** → contoh : dim nilai as integer

Jika **beberapa variabel mempunyai tipe data yang sama**, bisa dituliskan dalam satu baris dengan menggunakan tanda **koma** ( , ) sebagai pemisah antara satu variabel dengan variabel lain. Contoh :

**Dim nama, alamat**

**Dim nama, alamat as string**

Ada 2 cara penulisan dalam mendeklarasikan nama variabel dan tipe data variabel yang akan digunakan [1] :

### 1. Deklarasi Eksplisit

Suatu **deklarasi** yang diperuntukkan pada nama **variabel** dan tipe data variabel dilakukan pada **awal prosedur** dengan menggunakan perintah **Dim**. Dengan perintah ini, visual basic akan mengalokasikan memori sesuai dengan nama dan tipe data variabel yang dikehendaki.

Disamping harus menggunakan perintah Dim, pendeklarasian ini harus menambahkan kata Option Eksplisit pada tiap awal pembuatan sebuah proyek. **Pendeklarasian ini dilakukan pada bagian [General]** (diletakkan pada bagian atas kode program atau sebelum penulisan kode program) **dan prosedur [declaration]** (diletakkan pada awal prosedur). Deklarasi ini sangat dianjurkan karena lebih terencana dan terkontrol. Contoh :

Dim jenis kelamin → penulisan variabel tanpa tipe data.

Dim nama as String → penulisan satu variabel yaitu nama dengan tipe string.

Dim nama, buku as string → penulisan dua variabel yaitu nama dan buku dengan tipe string. Antara nama variabel yang satu dengan yang lain **dipisahkan** tanda **koma** ( , ).

Dim nilai as integer → penulisan satu variabel yaitu nilai dengan tipe integer.

Dim total\_uang as double → penulisan satu variabel yaitu total\_uang dengan tipe double.

Dim kelulusan as boolean → penulisan satu variabel yaitu kelulusan dengan tipe boolean.

### 2. Deklarasi Implisit

**Deklarasi** ini akan membuat visual basic mengalokasikan **nama** variabel dan **tipe** data variabel serta langsung mengisi **nilai dari variabelnya**. Contoh :

Dim nama as string = "yunianita" → variabel nama bertipe **string** mempunyai nilai / isi yaitu yunianita. Untuk mengisi variabel yang bertipe string diikuti tanda **petik dua** (" ").

Dim nilai as integer = 0 → variabel nilai bertipe integer mempunyai nilai / isi yaitu 0. Untuk mengisi variabel yang bertipe integer tanpa diikuti tanda petik dua.

Dim kelulusan as boolean	}	pendeklarasian variabel dan pemberian isi variabel ditulis dalam dua baris.
Kelulusan = true		

Pada pendeklarasian variabel secara implisit, pendeklarasian variabel dan pemberian isi variabel dapat ditulis dalam satu baris atau dalam dua baris secara berurutan. Contoh pendeklarasian variabel dan pemberian isi variabel yang ditulis dalam satu baris yaitu "Dim nama as string = "yunianita"". Sedangkan contoh untuk pendeklarasian variabel dan pemberian isi variabel yang ditulis dalam dua baris secara berurutan terdapat pada ilustrasi di atas. Pada ilustrasi di atas, pendeklarasian variabel ditulis pada baris pertama yaitu "dim kelulusan as boolean". Lalu baris berikutnya ditulis kode program untuk memberikan pada variabel kelulusan yaitu "kelulusan=true".

### 2.2.3. Isi Variabel

Isi variabel merupakan informasi yang disimpan dalam variabel. Isi / nilai variabel bisa berubah-ubah sesuai dengan operasi yang dilakukan pada variabel tersebut tetapi tipe datanya harus

tetap. Berikut cara penulisan perintah untuk mengisi nilai ke dalam variabel yaitu :

**<nama\_variabel> = <nilai> atau Dim <nama\_variabel> As  
<Tipe\_Data>=<nilai>**

Contoh :

Index = 10 (anggap variabel index sudah dideklarasikan sebelumnya)

Index = txtBil.text (anggap variabel index sudah dideklarasikan sebelumnya dan isinya diambil dari textbox yang bernama txtBil.text)

Dim index as Integer = 100

#### 2.2.4. Aksesibilitas Variabel

Selain Dim terdapat atribut variabel lainnya seperti public, private, dll, yang menunjukkan aksesibilitas variabel. Aksesibilitas variabel menentukan apakah variabel tersebut dapat diakses dari modul luar (atau aplikasi) tempat variabel tersebut dideklarasikan. Atribut berikut bersifat opsional artinya boleh digunakan atau tidak, yang macamnya adalah :

- **Dim**

Variabel yang didelarasikan dengan kata kunci Dim memiliki akses terbatas. Jika dideklarasikan pada prosedur maka jangkauannya hanya di prosedur saja. Jika prosedur selesai dijalankan maka nilai variabel dim akan hilang.

- **Public**

Variabel yang didelarasikan dengan kata kunci Public memiliki akses bebas. Tidak áda batasan untuk mengaksesnya. Kata kunci Pubcic dapat digunakan pada ruang lingkup modul, namespace, dan file tetapi tidak bisa digunakan pada prosedur.

- **Private**

Variabel yang didelarasikan dengan kata kunci Private memiliki akses terbatas. Yang dapat mengaksesnya hanya dari dalam isi deklarasi. Private dapat digunakan pada level modul, class atau structure, tetapi tidak dapat digunakan pada level namespace, file, dan procedure.

- **Protected**

Variabel yang didelarasikan dengan kata kunci Protected memiliki akses proteksi. Yang dapat mengaksesnya adalah class yang sama atau yang mewarisi class tersebut. Protected hanya dapat digunakan pada level class untuk mendeklarasikan anggota dari class, protected ini tidak dapat digunakan pada level prosedur, modul, namespace, dan file.

- **Friend**

Friend dapat digunakan pada level modul, namespace, file, dan class atau structure. Tetapi tidak bisa digunakan pada level prosedur.

- **Protected Friend**

Variabel yang dideklarasikan dengan menggunakan kata kunci Protected Friend memiliki akses gabungan antara protected dengan friend. Variabel ini dapat digunakan oleh kode dimanapun pada assembly yang sama, pada kode class itu sendiri, dan pada kode yang diwarisi dari suatu class. Protected Friend digunakan untuk mendeklarasikan anggota dari class. Protected Friend hanya dapat digunakan pada level class, tetapi tidak bisa digunakan pada level prosedur, modul, namespace, dan file.

### 2.2.5. Ruang Lingkup Variabel

Ruang lingkup variabel (*variabel scope*) adalah suatu daerah dimana dimana variabel yang dibuat dapat dikenal (dipakai). Ruang lingkup variabel biasanya berhubungan dengan waktu hidup variabel. Ada variabel yang mempunyai ruang lingkup kecil, sedang, maupun besar sehingga dikenal dimana-mana. Adapun macam-macam ruang lingkup variabel adalah sebagai berikut :

- **Ruang lingkup pada sebuah blok**

Suatu variabel bisa dideklarasikan dalam suatu blok pernyataan seperti pernyataan `for...next`, `if...then`, dll. Variabel tersebut mempunyai ruang lingkup hanya di dalam blok pernyataan tersebut. Contoh :

```
For i as integer = 0 to 10
```

```
    i=i+1
```

```
next
```

- **Ruang lingkup pada sebuah prosedur**

Pada ruang lingkup ini, suatu **variabel hanya boleh digunakan pada sebuah prosedur di tempat di mana variabel dideklarasikan**. Biasanya jenis variabel ini digunakan dalam **perhitungan sementara (*temporary*)**. Dalam mendeklarasikan variabel tersebut digunakan perintah **static** atau **dim**. **Perbedaan perintah static atau dim adalah :**

- Deklarasi dengan menggunakan perintah `dim` menyebabkan variabel hanya aktif saat prosedur dijalankan.
- Deklarasi dengan menggunakan `static` menyebabkan variabel tetap aktif meskipun prosedur telah selesai dijalankan, tetapi tidak bisa dipanggil oleh prosedur lain.

Contoh :

Static nama\_peg as string

Dim gaji\_peg as single

- **Ruang lingkup pada sebuah modul**

Pada ruang lingkup ini, suatu **variabel hanya dapat digunakan pada semua prosedur dalam sebuah modul dimana variabel tersebut dideklarasikan**. Dalam mendeklarasikan variabel ini digunakan perintah **private** atau **dim** di bagian [declaration] dari modul. Dari kedua perintah ini tidak ada perbedaan penggunaa, hanya saja perintah private lebih dianjurkan untuk membedakannya dengan pendeklarasian dalam prosedur.

Contoh :

Private nama as string

Dim nama as string

- **Ruang lingkup pada seluruh modul atau tingkat NameSpace**

Pada ruang lingkup ini, suatu **variabel akan dikenal baik pada semua prosedur maupun pada semua modul dimana variabel tersebut di deklarasikan**. Dalam mendeklarasikan variabel ini digunakan perintah **public** atau **Friend**, tetapi perintah ini tidak bisa digunakan untuk prosedur saja. Pendeklarasian ini dilakukan pada bagian [declaration] pada salah satu modul.

Contoh :

Public nama as string

### 2.3. Tipe Data

**Tipe data merupakan** suatu bentuk penggolongan jenis data berdasarkan kategori data, ukuran, dan kegunaan data yang

ditampung dalam suatu variabel. **Tipe data ini digunakan untuk mendeklarasikan variabel, dengan tipe data ini programmer bisa menentukan nilai yang dapat dimasukkan dalam variabel.** Dalam merancang suatu program aplikasi hendaknya diperhatikan jangan sampai menggunakan tipe data yang salah atau tidak tepat karena program yang dihasilkan tidak akan berjalan sesuai dengan keinginan. Misalnya, data untuk perhitungan, digunakan tipe data string dioperasikan dengan data dengan tipe integer, maka bisa dipastikan operasinya akan “error” [1].

### 2.3.1. Tipe Data Variant

Tipe data variant merupakan tipe data yang umum dimana semua data dapat ditampung ke dalam variabel **tanpa** menggunakan perintah **As**, maka default dari tipe data yang digunakan adalah **variant**. Contoh : Dim harga, jumlah.

Variabel dengan tipe data variant dapat diisi dengan data bertipe angka, string, waktu atau tanggal, atau diisi data dari tool lain. Tanda petik dua (“”) digunakan untuk mengisi data abjad atau angka. Pada data waktu atau tanggal menggunakan tanda pagar (#) untuk mengisi datanya. Contoh :

Dim nama, harga, jumlah, total, tanggal\_lahir

Nama = “dion”

Harga = “2600”

Jumlah = 3

Tanggal\_lahir = #08/07/2000#

Total = textbox1.text → mengisi variabel total yang diambil dari data yang berada pada tool textbox1.

Suatu **variabel** dengan tipe data variant dapat **digabungkan** dengan variabel lainnya dengan menggunakan tanda **&** atau tanda **+**. Ada perbedaan antara kedua tanda tersebut saat



digunakan untuk menggabungkan beberapa variabel, perhatikan gambar 2.2 dan gambar 2.3, diantaranya adalah :

No	Deklarasi dan operasi	Hasil Proses
1	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = "234" Nilai_2 = 5 Hasil = Nilai_1 & Nilai_2	2345
2	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = "xxxx" Nilai_2 = 5 Hasil = Nilai_1 & Nilai_2	xxxx5
3	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = 234 Nilai_2 = 5 Hasil = Nilai_1 & Nilai_2	2345

Gambar 2.2. Penggabungan Variabel dengan Tanda & [Yung:2005]

No	Deklarasi dan operasi	Hasil Proses
1	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = "234" Nilai_2 = 5 Hasil = Nilai_1 + Nilai_2	239
2	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = "xxxx" Nilai_2 = 5 Hasil = Nilai_1 + Nilai_2	Run-Time Error '13' Type Mismatch
3	Dim Hasil, Nilai_1, Nilai_2 Nilai_1 = 234 Nilai_2 = 5 Hasil = Nilai_1 + Nilai_2	239

Gambar 2.3. Penggabungan Variabel dengan Tanda + [Yung:2005]

### 2.3.2. Tipe Data Lain

Secara umum tipe data dibagi menjadi 5 kelompok yaitu :

- Numerik → tipe data numerik digunakan untuk menyimpan data angka (number).
- String → tipe data string digunakan untuk menyimpan rangkaian data karakter.

- Boolean → tipe data boolean digunakan untuk menyimpan data dengan nilai benar atau salah.
- Date → tipe data date digunakan untuk menyimpan data berupa tanggal dan waktu
- Object → tipe data object dipakai untuk menyimpan data berupa obyek.

Mendeklarasikan suatu variabel dalam visual basic untuk tipe data yang bukan varian harus menggunakan perintah **private**, **dim**, **public**, atau **static**. Contoh :

Private hasil as integer

Dim nama as string

Public jumlah as double

Static kota as string

Berikut ini tipe data beserta jangkauannya yang dapat digunakan pada program Visual Basic .Net, ditampilkan pada gambar 2.4.

Tipe Data Visual Basic	Struktur Runtime	Ukuran	Jangkauan
Boolean	<u>System.Boolean</u>	2 bytes	Benar atau Salah / True atau False
Byte	<u>System.Byte</u>	1 bytes	0 sampai 255 (unsigned / tidak ditandai / bilangan positif)
Char	<u>System.Char</u>	2 bytes	0 sampai 65.535 (unsigned / tidak ditandai)
Date	<u>System.DateTime</u>	8 bytes	0:00:00 pada 1 Januari, 0001 sampai 11:59:59 PM pada 31 Desember, 9999
Decimal	<u>System.Decimal</u>	16 bytes	0 sampai +/- 79.228.162.514.264.337.593.543.950.335 tanpa nilai decimal; 0 sampai +/- 7,9228162.514264337593543950335 dengan 28 angka disebelah koma decimal; +/- 0,000000000000000000000000000000001 (+/-1E-28) untuk nilai yang lebih kecil dari d
Double (presisi ganda - nilai pecahan)	<u>System.Double</u>	8 bytes	-1,79769313486231570E+308 sampai -4,94065645841246544E-324 untuk nilai negatif; 4,94065645841246544E-324 sampai 1,79769313486231570E+308 untuk nilai positif
Integer (bilangan bulat)	<u>System.Int32</u>	4 bytes	-2.147.438.648 sampai 2.147.438.647
Long (Long Integer) (bilangan bulat)	<u>System.Int64</u>	8 bytes	-9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807
Object	<u>System.Object</u>	4 bytes	Tipe apapun dapat disimpan dalam variabel tipe objek
Short (bilangan bulat)	<u>System.Int16</u>	2 bytes	-32.768 sampai 32.767
Single (presisi tunggal - nilai pecahan)	<u>System.Single</u>	4 bytes	-3,4028235E+38 sampai -1,401298E-45 untuk nilai negatif; 1,401298E-45 sampai 3,4028235E+38 untuk nilai positif
String	<u>System.String</u>	10 bytes	0 sampai 2 juta karakter
Tipe Data Buatan User	<u>System.ValueType</u>	Berbeda-beda	Tergantung platform yang digunakan

Gambar 2.4. Tipe Data Pada Visual Basic.Net

### 2.3.3. Tipe Data Buatan Sendiri

Selain tipe data yang telah disediakan oleh Visual Basic .Net, user dapat membuat tipe data baru sendiri dengan mengambil salah satu tipe data standar yang telah disediakan. Dalam mendeklarasikan tipe data buatan sendiri, gunakan perintah **type** setelah kata public atau private. Penggunaan kata kunci public atau private tergantung pada ruang lingkup variable yang diinginkan. Secara umum penulisan dari tipe data buatan sendiri sebagai berikut :

**Private Type <nama\_tipe\_data>**

....  
....

**End Type**

Atau

**Public Type <nama\_tipe\_data>**

....

....

**End Type**

Sebagai contoh, dalam tabel SISWA terdiri dari field NIS, NAMA, ALAMAT, maka untuk membuat tipe data sendiri yaitu :

**Private Type Siswa**

**NIS as string \* 5**

**NAMA as string \* 30**

**ALAMAT as string \* 50**

**End Type**

Sedangkan untuk memasukkan atau mengambil data dengan tipe data buatan sendiri dapat langsung diisikan lewat program atau lewat variabel. Berikut contoh pengisian yang dilakukan lewat program yaitu :

**Sub isi\_data\_siswa()**

**NIS = "222"**

**NAMA = "AA"**

**ALAMAT = "SDA"**

**End sub**

Berikut contoh pengisian yang dilakukan lewat variabel :

**Sub isi\_data\_siswa()**

**NIS = txtNis.text**

**NAMA = txtNama.text**

**ALAMAT = txtAlamat.text**

**End sub**

#### 2.3.4. Konversi Tipe Data (*Casting*)

**Konversi Tipe data adalah** sebuah fungsi yang digunakan untuk mengkonversi (mengubah tipe dari satu tipe ke tipe data yang lain). Konversi tipe data pada visual basic dapat dilakukan secara implicit maupun explicit. Konversi secara implisit dilakukan tanpa

menggunakan fungsi bantu casting tertentu, misalnya Cbool, Cbyte, Cchar, Cdate, Cdac, Cdbl, Cint, Clng, Cobj, Cshort, Cstr, dan Csng. Contoh :



Macam konversi tipe data secara eksplisit dapat dilihat pada gambar 2.5.

Konversi	Tipe Data	Metode
Boolean	Boolean	Boolean
Byte	Byte	Byte
Char	Char	Char
Date	Date	Date
Dbl	Dbl	Dbl
Integer	Integer	Integer
Long	Long	Long
Object	Object	Object
Short	Short	Short
String	String	String
Single	Single	Single
Sng	Sng	Sng

Gambar 2.5. Konversi Tipe Data Pada Visual Basic.Net (Dok. Internet)

#### 2.4. Konstanta

**Konstanta atau literal adalah** suatu variabel yang mempunyai nilai tetap. Konstanta diperlukan jika dalam suatu operasi memerlukan suatu sebuah nilai tetap yang senantiasa digunakan padad banyak bagian dari suatu rutin, atau untuk mengingat bilangan yang sulit [1].

#### 2.4.1. Penulisan Konstanta

Mengingat tipe data bentuknya bermacam-macam maka penulisan konstanta mempunyai aturan tersendiri, yaitu:

- **Konstanta Numerik** → Konstanta numerik (angka) seperti byte, integer, single, double, dan lainnya ditulis langsung apa adanya. Contoh : 2750, 123E+5, -540.53
- **Konstanta String** → Konstanta string ditulis dengan dibatasi oleh tanda kutip (“ ”). Contoh : “2500”, “Jl. Mawar 25 Surabaya”, “Visual Basic.Net”
- **Konstanta Date** → Konstanta date digunakan untuk data tanggal dan jam dimasa penulisannya dibatasi oleh tanda pagar # #. Contoh : #19/09/2019#, #10:45#, #5:25 PM#.
- **Konstanta Boolean** → Konstanta boolean hanya berisikan dua nilai yaitu true dan false.

#### 2.4.2. Sumber Konstanta

Menurut sumbe datanya, konstanta dibagi menjadi 2 macam yaitu :

- **Intrinsic**

Sumber konstanta intrinsic telah didefinisikan oleh visual basic.net, misalnya saja Visual Basic form Application, Visual Basic, dan Data Access Object (DAO). Konstanta ini disebut **konstanta internal**. Macam-macam konstanta internal dapat dilihat di Object Libraries di Object Browser. Pemanggilan konstanta internal harus didahului oleh sebuah awalan yaitu vb untuk visual basic dan visual form applicatin, sedangkan db untuk Data Access Object (DAO). Contoh : cbsizeall, vbokcancel, dbrelationshipuniqu, xldialogborder.

Disamping nilai konstantanya diketik langsung pada program, nilai konstanta bisa diisikan lewat jendela properties. Misalnya untuk mendeklarasikan **ukuran jendela (window)**. **Properties WindowState** ini mempunyai nilai konstanta yang tidak dapat diubah lagi. Nilai-nilai tersebut antara lain **0 – normal**, **1 – minimized**, dan **2 – maximized**. Jika nilai-nilai tersebut diterjemahkan, tujuannya sebagai berikut : **0 – normal** untuk menjadikan ukuran jendela form saat program dijalankan sesuai dengan ukuran saat desain form, **1 – minimized** untuk menjadikan ukuran jendela form saat program dijalankan dibuat sekecil mungkin sehingga bisa menempati taksbar dari windows, dan **2 – maximized** untuk menjadikan ukuran jendela form saat program dijalankan dibuat seluas layar monitor.

- **Symbolic**

Sumber konstanta symbolic didefinisikan sendiri oleh pembuat programnya. Standar penulisan untuk mendeklarasikan konstanta symbolic tersebut adalah :

**[Public | Private] Const nama\_Konstants [As Tipe]**  
**= Ekspresi**

Contoh :

Const nilai\_phi = 3.14

Public cons nama = “Langit”

Cons tgl\_lahir = #24-5-2008#

Dengan melihat sumber data konstanta, maka disimpulkan **ruang lingkup konstanta** yaitu :

- **Pada sebuah prosedur**

Nilai konstanta hanya dikenal pada prosedur yang bersangkutan jika pendeklarasian konstanta dilakukan dalam prosedur tersebut.

- **Pada semua prosedur dalam suatu modul**

Nilai konstanta akan dikenal pada semua prosedur dalam satu modul jika pendeklarasiannya dilakukan pada bagian [declaration] modul tersebut.

- **Pada seluruh bagian aplikasi**

Nilai konstanta akan dikenal pada seluruh aplikasi yang dibuat jika pendeklarasiannya menggunakan kata `public` sebelum kata `const`.

## 2.5. Operator dan Operand

Operator dan operand merupakan dua elemen dasar pada proses operasi. Dengan kata lain keberadaan operator dalam suatu operasi senantiasa diikuti oleh operand. **Operator** adalah suatu tanda yang digunakan untuk menghubungkan satu variabel atau konstanta dengan variabel atau konstanta lain dengan tujuan melakukan berbagai manipulasi dan pengolahan data. Sedangkan **operand** adalah suatu data atau variabel yang dikenai operasi tersebut. Berikut operator yang ada di visual basic.net yaitu :

- **Operator Penugasan**

**Operator penugasan (Assignment)** merupakan operator yang berfungsi untuk **memasukkan suatu data ke dalam suatu variabel**. Simbol dari operator penugasan adalah tanda sama dengan (=). Contoh :

Dim harga, jumlah, total as integer

Harga = 2500

Jumlah = 5

Total = harga \* jumlah

- **Operator Aritmatika**

**Operator aritmatika** adalah operator yang digunakan untuk **melakukan perhitungan matematis aritmatika**,



seperti penjumlahan, pengurangan, dan sebagainya. **Operator aritmatika mempunyai hierarki paling tinggi dibandingkan dengan operator pembanding dan operator logika.** Berikut macam-macam operator aritmatika menurut hirarki paling tinggi ke hirarki paling rendah, ditampilkan pada tabel 2.1.

Tabel 2.1. Tabel Operator Aritmatika

Opera tor	Operasi	Keterangan
()	Pengelompokan	Tanda kurung buka dan kurung tutup digunakan untuk mengelompokkan operasi yang akan dikerjakan lebih dulu
^	Pemangkatan / Eksponensial	Digunakan untuk operasi pemangkatan
+, -	Tanda/Nilai	Biasanya tanda positif tidak dituliskan lagi
*	Perkalian	digunakan untuk operasi perkalian
/	Pembagian	digunakan untuk operasi pembagian dengan hasil sampai dengan decimal
\	Pembagian Integer	digunakan untuk operasi pembagian dengan hasil integer
Mod	Modulus	digunakan untuk operasi pembagian yang hasilnya merupakan sisa hasil bagi
+	Penjumlahan	digunakan untuk operasi penjumlahan
-	Pengurangan	digunakan untuk operasi pengurangan

Dalam suatu operasi, program akan menjalankan operator yang paling tinggi hingga ke operator yang paling rendah. Urutannya adalah operator pengelompokan, pangkat, tanda negatif, perkalian dan pembagian, pembagi integer, modulus, diikuti penambahan dan pengurangan.

Jika terdapat operator yang mempunyai urutan yang sama dalam suatu operasi, misalnya terdapat operator penjumlahan dan pengurangan, maka operasi yang dikerjakan terlebih dahulu adalah operasi yang berada di paling kiri.

- **Operator Pembanding**

**Operator pembanding (relasi)** adalah operator yang digunakan untuk **membandingkan suatu data/ekspresi dengan data/ekspresi lain dan menghasilkan nilai logika (boolean) benar atau salah**. Syarat yang harus dipenuhi agar kedua data bisa dibandingkan yaitu harus mempunyai **tipe data yang sama**. Bentuk operator pembanding, diperlihatkan pada tabel 2.2.

Tabel 2.2. Tabel Operator Pembanding

Operator	Operasi
=	Sama dengan
<>	Tidak sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan

- **Operator Logika**

**Operator logika** merupakan operator yang digunakan untuk **mengekspresikan satu atau lebih data (ekspresi) logika (boolean) yang menghasilkan data logika baru**. Bentuk operator logika, diperlihatkan pada tabel 2.3.

Tabel 2.3. Tabel Operator Logika

Operator	Keterangan
And	Hasilnya true jika kedua operand bernilai true. Selain itu hasilnya false. Tabel operator And dapat dilihat pada tabel 2.4.
Or	Hasilnya true jika salah satu atau kedua operand bernilai true. Selain itu hasilnya false. Tabel operator Or dapat dilihat pada tabel 2.5.
Xor	Hasilnya true jika salah satu operand bernilai true. Selain itu hasilnya false. Tabel operator Xor dapat dilihat pada tabel 2.6.
AndAlso	Hasilnya true jika kedua operand bernilai true. Selain itu hasilnya false. Operasi ini hanya mengevaluasi jika ekspresi 1 bernilai true, jika ekspresi 1 bernilai false maka tidak akan dievaluasi dan mempunyai hasil false. Tabel operator AndAlso dapat dilihat pada tabel 2.7.

OrElse	Hasilnya true jika salah satu atau kedua operand bernilai true. Selain itu hasilnya false. Operasi ini hanya mengevaluasi jika ekspresi 1 bernilai false, jika ekspresi 1 bernilai true maka tidak akan dievaluasi dan mempunyai hasil true. Tabel operator OrElse dapat dilihat pada tabel 2.8.
Not	Merupakan operator unary. Hasilnya true jika operand false. Hasilnya false jika operand true. Tabel operator Not dapat dilihat pada tabel 2.9.
Implikasi	operator logika yang menghasilkan nilai false (0) jika nilai pertama yang diinputkan adalah nilai true (1) dan nilai kedua yang diinputkan adalah nilai false (0). Tabel operator Not dapat dilihat pada tabel 2.10.

Penjelasan tentang rincian masing-masing operator logika dijelaskan pada tabel 2.4 untuk operator and, tabel 2.5 untuk operator or, tabel 2.6 untuk operator Xor, tabel 2.7 untuk operator andalso, tabel 2.8 untuk operator orelse, dan tabel 2.9 untuk operator not.

Tabel 2.4. Tabel Operator And

Ekspresi 1	Ekspresi 2	Hasil
True	True	True
True	False	False
False	True	False
False	False	False

Tabel 2.5. Tabel Operator Or

Ekspresi 1	Ekspresi 2	Hasil
True	True	True
True	False	True
False	True	True
False	False	False

Tabel 2.6. Tabel Operator Xor

Ekspresi 1	Ekspresi 2	Hasil
True	True	False
True	False	True
False	True	True
False	False	False

Tabel 2.7. Tabel Operator AndAlso

Ekspresi 1	Ekspresi 2	Hasil
True	True	True
True	False	False
False	Tidak dievaluasi	True

Tabel 2.8. Tabel Operator OrElse

Ekspresi 1	Ekspresi 2	Hasil
True	Tidak dievaluasi	True
False	True	True
False	False	False

Tabel 2.9. Tabel Operator Not

Ekspresi 1	Hasil
True	False
False	True

Tabel 2.10 Tabel Kebenaran Operator Implikasi

Bilangan 1	Bilangan 2	And
1	1	1
0	1	1
1	0	0
0	0	1

Operator **AndAlso** dan **OrElse** disebut evaluasi sirkuit pendek karena operasi tersebut ketika melakukan evaluasi hanya sampai mengetahui benar atau salah saja.

- **Operator Bitwise**

Guna operator bitwise adalah memanipulasi bit secara individu untuk membuat nilai dari tipe integer (byte, short, integer, dan long). Bentuk operator bitwise, diperlihatkan pada tabel 2.11.

Tabel 2.11. Tabel Operator Bitwise

Operator	Kependekan Dari	Artinya	Contoh	Hasil
And	Bitwise And	Mempunyai hasil 1, jika baik sisi kanan maupun kiri dari operator adalah 1	0 And 0	0
			0 And 1	0
			1 And 0	0
			1 And 1	1
Or			0 And 0	0

	Inklusif Bitwise Or	Memiliki hasil 1, jika salah satu atau kedua sisi dari operator adalah 1	0 And 1	1
			1 And 0	1
			1 And 1	1
Xor	Eksklusif f Bitwise Or	Memiliki hasil 1, jika salah satu dari sisi kanan atau kiri dari operator adalah 1, tetapi tidak keduanya	0 And 0	0
			0 And 1	1
			1 And 0	1
			1 And 1	0

Operator bitwise bekerja dengan menerapkan pemberian operasi Boolean pada tiap bit yang berhubungan dalam dua operand.

- **Operator String**

Operator string merupakan operator yang digunakan untuk **menggabungkan dua buah string atau lebih. Simbol** dari operasi string adalah + **dan &**.

Operator string merupakan operator yang berdiri sendiri dan bukan bagian dari operator aritmatika, meskipun hirarkinya persis di bawah operator aritmatika dan di atas operator perbandingan.

Ada perbedaan antara operator + dengan operator & dalam mengoperasikan data / variabel (dapat dilihat pada gambar 2 dan gambar 3).

## 2.6. Latihan Program

Latihan program yang dibahas berkaitan dengan materi yang telah dibahas pada bab 2.

Program/Form 1 ini bertujuan untuk mengenalkan operator aritmatika yang ada pada visual basic.Net yaitu perpangkatan (^), perkalian (\*), pembagian biasa (/) dan pembagian integer (\), modulus (mod), penambahan (+), dan pengurangan (-). Hasil dari operator aritmatika ini adalah nilai angka yang bertipe integer atau double. Berikut desain, tool yang harus diatur, dan

Kode programnya. Tampilan program/form 1 dapat dilihat pada gambar 2.5.



Gambar 2.5. Desain Form 1 (Form Operator Aritmatika)

Berikut tool dan properties pada program/form 1 yang diatur, ditampilkan pada tabel 2.12.

Tabel 2.12. Tabel Tool dan Properties Form 1 (Form Aritmatika)

Tool	Properties	Setting
Form 1	Text	Operator Aritmatika
	BackColor	Lemon Chiffon
	Font	TimesNewRoman; 12
GroupBox 1	Text	Data Bilangan 1
GroupBox 2	Text	Data Bilangan 2
GroupBox 3	Text	Operator Aritmatika
Button 1	Text	^
	Name	btnPangkat
Button 2	Text	*
	Name	btnKali
Button 3	Text	/
	Name	btnBagi
Button 4	Text	\
	Name	btnBagi2
Button 5	Text	Mod
	Name	btnMod
	Text	+

Button 6	Name	btnTambah
Button 7	Text	-
	Name	btnKurang
Button 8	Text	Bersih
	Name	btnBersih
Button 9	Text	Form Utama
	Name	btnFrmUtama
Label 1	Text	Operator Aritmatika
	Font	TimesNewRoman; Bold; 24
Label 2	Text	Bilangan 1
Label 3	Text	Bilangan 2
Label 4	Text	Hasil
Label 5	Text	Bilangan 1
Label 6	Text	Bilangan 2
Label 7	Text	Hasil
TextBox 1	Name	txtBil1
TextBox 2	Name	txtBil2
TextBox 3	Name	txtHasil
	Enabled	False
TextBox 4	Name	txtBil12
TextBox 5	Name	txtBil12
TextBox 6	Name	txtHasil2
	Enabled	False

Kode program yang harus ditambahkan agar program berjalan sesuai dengan yang diinginkan adalah :

- Tambahkan kode program di posisi paling atas. Kode program ini untuk pengenalan variabel yaitu :

```
Dim bil1, bil2, hasil2 As Double
```

- Tambahkan kode program di form. Kode program ini berfungsi agar angka pada bilangan 1 dan bilangan 2 pada grup data bilangan 2 tampil pada saat form dijalankan yaitu

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    bil1 = 10.666
    bil2 = 4.5
    txtBil12.Text = bil1
    txtBil22.Text = bil2
End Sub
```

End Sub

- Tambahkan kode berikut pada tombol ^ (pangkat). Kode program ini digunakan untuk melakukan operasi perpangkatan.

```
Private Sub btnPangkat_Click(sender As Object, e As EventArgs) Handles
btnPangkat.Click
    'perhitungan pada grup 2
    bil1 = 10.666
    bil2 = 4.5
    txtBil12.Text = bil1
    txtBil22.Text = bil2
    txtHasil2.Text = bil1 ^ bil2
    'perhitungan pada grup 1
    txtHasil.Text = txtBil1.Text ^ txtBil2.Text
End Sub
```

- Tambahkan kode pada tombol \* (kali). Kode program ini digunakan untuk melakukan operasi perkalian.

```
Private Sub btnKali_Click(sender As Object, e As EventArgs) Handles
btnKali.Click
    'perhitungan pada grup 2
    bil1 = 10.666
    bil2 = 4.5
    txtBil12.Text = bil1
    txtBil22.Text = bil2
    txtHasil2.Text = bil1 * bil2
    'perhitungan pada grup 1
    txtHasil.Text = txtBil1.Text * txtBil2.Text
End Sub
```

- Tambahkan kode pada tombol / (bagi) untuk pembagian. Kode program ini digunakan untuk melakukan operasi pembagian.

```
Private Sub btnBagi1_Click(sender As Object, e As EventArgs) Handles
btnBagi1.Click
```



```

'perhitungan pada grup 2
  bil1 = 1 0.666
  bil2 = 4.5
  txtBil12.Text = bil1
  txtBil22.Text = bil2
  txtHasil2.Text = bil1 / bil2
  'perhitungan pada grup 1
  txtHasil.Text = txtBil1.Text / txtBil2.Text

```

End Sub

- Tambahkan kode pada tombol \ (pembagian integer). Kode program ini digunakan untuk melakukan operasi pembagian.

```

Private Sub btnBagi2_Click(sender As Object, e As EventArgs) Handles
btnBagi2.Click
  'perhitungan pada grup 2
  bil1 = 10.666
  bil2 = 4.5
  txtBil12.Text = bil1
  txtBil22.Text = bil2
  txtHasil2.Text = bil1 \ bil2
  txtHasil.Text = txtBil1.Text \ txtBil2.Text

```

End Sub

- Tambahkan kode pada tombol mod. Kode program ini digunakan untuk melakukan operasi mod.

```

Private Sub btnMod_Click(sender As Object, e As EventArgs) Handles
btnMod.Click
  'perhitungan pada grup 2
  bil1 = 10.666
  bil2 = 4.5
  txtBil12.Text = bil1
  txtBil22.Text = bil2
  txtHasil2.Text = bil1 Mod bil2
  'perhitungan pada grup 1
  txtHasil.Text = txtBil1.Text Mod txtBil2.Text

```

End Sub

- Tambahkan kode pada tombol + (tambah). Kode program ini digunakan untuk melakukan operasi penambahan.

```

Private Sub btntambah_Click(sender As Object, e As EventArgs) Handles
btntambah.Click

```

```

'perhitungan pada grup 2
  bil1 = 10.666
  bil2 = 4.5
  txtBil12.Text = bil1
  txtBil22.Text = bil2
  txtHasil2.Text = CDbI(bil1) + CDbI(bil2)
  'perhitungan pada grup 1
  txtHasil.Text = CInt(txtBil1.Text) + CInt(txtBil2.Text)

```

End Sub

- Tambahkan kode pada tombol – (kurang). Kode program ini digunakan untuk melakukan operasi pengurangan.

```

Private Sub btnKurang_Click(sender As Object, e As EventArgs) Handles
btnKurang.Click
  'perhitungan pada grup 2
  bil1 = 10.666
  bil2 = 4.5
  txtBil12.Text = bil1
  txtBil22.Text = bil2
  txtHasil2.Text = bil1 - bil2
  'perhitungan pada grup 1
  txtHasil.Text = txtBil1.Text - txtBil2.Text

```

End Sub

- Tambahkan kode pada tombol bersih. Kode program ini digunakan untuk melakukan penghapusan data yang ada di tool textbox.

```

Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles
btnBersih.Click
  txtBil1.Text = ""
  txtBil2.Text = ""
  txtHasil.Text = ""
  txtBil1.Focus() 'Agar kursor berada di textbox bil1

```

End Sub

- Tambahkan kode pada tombol Form Utama. Kode program berungsi untuk memanggil form 2 dan menyembunyikan form operator aritmatika ini.

```

Private Sub btnFrmUtama_Click(sender As Object, e As EventArgs) Handles
btnFrmUtama.Click
  Form2.Show()
  Me.Hide()

```

End Sub

- Tambahkan kode program pada textbox 1 dengan event keypress. Kode program ini berfungsi memindahkan kursor antar textbox menggunakan enter dan data yang dimasukkan hanya data angka.

```
Private Sub txtBil1_KeyPress(sender As Object, e As KeyPressEventArgs)
Handles txtBil1.KeyPress
    'memindahkan kursor dari textbox1 ke yang lain dengan enter
    If e.KeyChar = Chr(13) Then
        txtBil2.Focus()
    End If
    'agar masukan textbox hanya angka dan delete
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9") Or e.KeyChar = vbBack
    Then
        e.Handled = True
    End If
End Sub
```

- Tambahkan kode program pada textbox 2 dengan event keypress. Kode program berfungsi untuk memindahkan kursor antar textbox menggunakan enter dan data yang dimasukkan hanya data angka.

```
Private Sub txtBil2_KeyPress(sender As Object, e As KeyPressEventArgs)
Handles txtBil2.KeyPress
    If e.KeyChar = Chr(13) Then
        btnPangkat.Focus()
    End If
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9") Or e.KeyChar = vbBack
    Then
        e.Handled = True
    End If
End Sub
```

### BAB III STRUKTUR PERCABANGAN

Bab 3 ini menjelaskan tentang Pengantar Struktur Percabangan, Struktur IF – THEN, Struktur IF – THEN – Else, Struktur Nested IF, Struktur IIF, Struktur Select Case, dan tugas. Struktur Select Case merupakan bentuk lain dari penulisan IF-THEN.

### 3.1. Pengantar Struktur Percabangan

Pada program-program yang telah dibuat sejauh ini, statemen-statemen dieksekusi secara sekuensial, yang berarti statemen-statemen dieksekusi secara berurutan, satu demi satu. Contoh program yang statemen-statemennya dieksekusi secara sekuensial dapat dilihat pada gambar 3.1.

```
Langkah 1 --> lblNim.Text = txtNim.Text  
langkah 2 --> lblNama.Text = txtNama.Text  
langkah 3 --> lblAlamat.Text = txtAlamat.Text
```

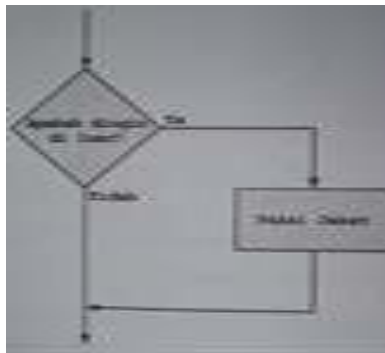
Gambar 3.1. Statemen-Statemen Sekuensial

Tipe kode program seperti gambar 3.1 di atas dinamakan dengan **struktur runtun** karena statemen-statemen dieksekusi secara beruntun, tanpa percabangan ke arah lain. Pada praktiknya, program sering memerlukan **lebih dari satu alur eksekusi** karena banyak algoritma memerlukan sebuah program untuk mengeksekusi sejumlah statemen-statemen dengan kondisi-kondisi tertentu. Ini dapat dilakukan dengan **struktur percabangan**. **Struktur Percabangan atau Penyeleksian kondisi atau pernyataan kondisi (Conditional Statement)** merupakan suatu pernyataan yang menganalisa suatu keadaan dan mengambil keputusan berdasarkan hasil analisa itu.

Dalam format paling sederhana dari struktur percabangan, ekspresi diuji apakah nilainya **true** atau **false**. Jika ekspresi bernilai **true**, sebuah aksi akan dilakukan. Jika ekspresi bernilai **false**, aksi tidak akan dilakukan. Atau, Hasil dari

penyeleksian, **jika benar maka akan dikerjakan instruksi tertentu**. Sedang **kondisi salah, maka akan dikerjakan instruksi lain**. Struktur percabangan sederhana jika digambarkan pada diagram alir dilihat pada gambar 3.2.

Pada gambar 3.2 menunjukkan diagram alir struktur percabangan sederhana. Pada gambar tersebut, terdapat symbol diamond yang merepresentasikan pertanyaan ya/tidak, atau ekspresi true/false. Jika jawaban terhadap pertanyaan itu adalah ya (atau jika ekspresi bernilai **true**) maka program akan menjalankan satu alur/statemen. Jika jawaban terhadap pertanyaan itu adalah tidak (atau jika ekspresi bernilai **false**) maka program akan menjalankan satu alur/statemen lain.



Gambar 3.2. Diagram Alir Struktur Percabangan Sederhana  
[yuswanto:2006]

Pada diagram alir, aksi **pakai jaket** dilakukan hanya ketika di luardingin. Jika di luar tidak dingin, aksi ini akan dilompati. Aksi dieksekusi secara kondisional (dengan syarat) karena ia adilakukan hanya ketika kondisi atau syarat tertentu terpenuhi.

Struktur percabangan dapat pula diekspresikan dengan pseudocode. Statemen ditulis diantara **if...then** dan **end if**, statemen ini dijalankan jika syarat terpenuhi. Statemen **end if** menandakan akhir dari struktur percabangan dalam pseudocode.

Contoh :

```

If dingin di luar then
    Pakai jaket
End if

```

Untuk struktur percabangan yang digunakan dalam pemrograman ini menggunakan statement **IF ... THEN...** dan **SELECT....CASE**.

### 3.2. Struktur IF – THEN

Struktur **IF – THEN** disebut juga dengan **branch structure** merupakan **struktur percabangan dimana suatu ekspresi akan dikerjakan bila kondisinya terpenuhi**. Tetapi jika kondisinya tidak terpenuhi maka ekspresi di dalam struktur if tidak akan dijalankan dan blok if akan dilompati serta program akan melakukan tindakan berikutnya. Bentuk If-Then sederhana dapat ditampilkan dalam bentuk diagram alir pada gambar 3.3.



Gambar 3.3. Bentuk IF - THEN

Pada gambar 3.3 dijelaskan alur If-Then sederhana. Dimulai dengan adanya suatu syarat, jika syarat tersebut terpenuhi, maka akan menjalankan suatu instruksi. Tetapi jika syarat tersebut tidak terpenuhi, maka instruksi tersebut tidak dijalankan dan program akan selesai.

#### **Bentuk penulisan dari struktur if – then sebagai berikut :**

```

If kondisi Then
    .....ekspresi.....

```

```

End If

```

Keterangan :

- Kondisi  
Berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda <, >, =, <> dari suatu keadaan yang akan menghasilkan nilai “Benar” atau “Salah”.
- Ekspresi  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisinya bernilai “benar” (memenuhi syarat).

Contoh Program 1 → kode program di bawah ini digunakan untuk memindahkan kursor dari satu tool ke tool lain dengan menggunakan enter. Pada kode ini memanfaatkan kode ASCII enter yaitu 13. Jadi, jika user menekan enter, maka kursor akan berpindah ke tool lain. Misalnya, posisi kursor user terlewat di textbox1, setelah user mengisikan data di textbox1 dan user menekan enter, maka kursor akan berpindah ke textbox lain yaitu txtBil2 tanpa menggunakan mouse.



Contoh Program 2 → kode program di bawah ini digunakan untuk memasukkan data angka saja pada textbox. Pada kode ini memanfaatkan kode ASCII angka 0 yaitu 48, angka 9 yaitu 57, dan backspace yaitu vbback. Jadi, user hanya diijinkan untuk menekan angka 0 hingga 9 dan menekan tombol backspace untuk menghapus data jika ada kesalahan. Selain karakter tadi, karakter lain tidak dapat dimasukkan ke textbox.

Ekspresi yang diuji di dalam sebuah statemen If-Then dapat bernilai **true** atau **false**. Ekspresi semacam itu dinamakan **ekspresi Boolean**. Operator yang digunakan untuk menghasilkan ekspresi Boolean adalah **operator relasional**. Berikut macam-macam operator relasional, ditampilkan pada tabel 3.1.

Tabel 3.1. Tabel Operator Relasional

Operator Relasional	Arti	Ekspresi Relasional	arti
>	Lebih besar dari	$x > y$	Apakah x lebih besar dari y?
<	Lebih kecil dari	$x < y$	Apakah x kurang besar dari y?
=	Sama dengan	$x = y$	Apakah x sama dengan y?
<>	Tidak sama dengan	$x < > y$	Apakah x tidak sama dengan y?
>=	Lebih besar dari atau sama dengan	$x > = y$	Apakah x lebih besar atau sama dengan y?
<=	Lebih kecil dari atau sama dengan	$x < = y$	Apakah x kurang besar atau sama dengan y?

Terdapat beberapa aturan yang perlu diingat pada statemen if-then multi-baris yaitu :

- Sebuah ekspresi Boolean harus diberikan antara If-Then.
- Kata **If-Then** ditempatkan di baris yang sama.
- Tidak ada selain komentar yang bisa ditempatkan setelah kata kunci **Then**, pada baris pertama
- Statemen **End If** harus berada baris yang sama. Hanya komentar yang bisa ditempatkan setelahnya pada baris yang sama.



### 3.3. Struktur IF – THEN – ELSE

Struktur **IF – THEN – ELSE** disebut juga dengan selection strukture merupakan struktur percabangan dimana suatu ekspresi akan dikerjakan bila kondisinya terpenuhi. Tetapi jika kondisinya tidak terpenuhi maka ekspresi lainnya akan dikerjakan. Bentuk If-Then-Else dapat ditampilkan dalam bentuk diagram alir pada gambar 3.4.

Pada gambar 3.4 dijelaskan bahwa terdapat suatu kondisi, jika kondisi bernilai true maka program akan menjalankan statemen IF. Jika kondisi bernilai false maka program akan menjalankan ELSE statemen.



Gambar 3.4. Bentuk If-Then-Else

#### Bentuk penulisan dari struktur if – then – else :

**If** kondisi **Then**

.....ekspresi 1....

**ELSE**

.....ekspresi 2....

**End If**

#### Keterangan :

- Kondisi

Berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda  $<$ ,  $>$ ,  $=$ ,  $<>$  dari suatu keadaan yang akan menghasilkan nilai “**True/Benar**” atau “**False/Salah**”.

- Ekspresi 1

Rangkaian kode program (blok program) yang akan dijalankan jika kondisinya bernilai “**true/benar**” (memenuhi syarat).

- Ekspresi 2

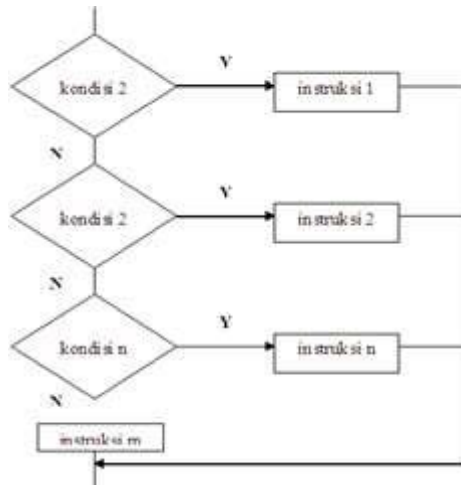
Rangkaian kode program (blok program) yang akan dijalankan jika kondisinya bernilai “**false/salah**” (tidak memenuhi syarat).

Contoh Program → program di bawah ini digunakan untuk menampilkan teks yang ada pada tool RadioButton pada tool TextBox. Tool RadioButton ini untuk memberikan pilihan kepada user berupa jenis kelamin yaitu laki-laki dan perempuan. Tool RadioButton ini mempunyai nama yaitu rdbLaki dan rdbPr. Sedangkan tool Textbox jenis kelamin mempunyai nama jk. Jika user menekan centang pada pilihan laki-laki pada tool radiobutton laki-laki (rdbLaki) maka teks pada radiobutton laki-laki (rdbLaki) akan ditampilkan pada textbox jenis kelamin (jk). Sebaliknya, Jika user menekan centang pada pilihan perempuan pada tool radiobutton perempuan (rdbPr) maka teks pada radiobutton perempuan (rdbPr) akan ditampilkan pada textbox jenis kelamin (jk).



### 3.4. Struktur If-Then-Else If

Statemen ini digunakan untuk membuat sejumlah keputusan dengan menetapkan beberapa aturan yang berbeda. Misalnya Jika Nilai Akhir = 90 maka Nilai Huruf = A, Jika Nilai Akhir = 70 maka Nilai Huruf = B, Jika Nilai Akhir = 50 maka Nilai Huruf = C, dan seterusnya. Statemen yang digunakan untuk menuliskan kode program tersebut yaitu menggunakan **If-Then-Elseif**. Bentuk If-Then-Elseif dapat ditampilkan dalam bentuk diagram alir pada gambar 3.5.



Gambar 3.5. Bentuk If-Then-Elseif

Pada gambar 3.5 dijelaskan bahwa terdapat beberapa kondisi, jika kondisi 1 terpenuhi maka program akan menjalankan instruksi 1. Jika kondisi 1 tidak terpenuhi maka program akan

memeriksa kondisi 2, jika kondisi 2 terpenuhi maka program akan menjalankan instruksi 2. Jika kondisi 2 tidak terpenuhi maka program akan memeriksa kondisi n, jika kondisi n terpenuhi maka program akan menjalankan instruksi n. jika kondisi n tidak terpenuhi maka program akan menjalankan instruksi m.

### **Bentuk penulisan dari struktur if – then – else :**

**If** kondisi 1 **Then**

.....ekspresi 1....

**ELSEIF** kondisi 2 **Then**

.....ekspresi 2....

**ELSE**

.....ekspresi lain....

**End If**

### **Keterangan :**

- **Kondisi 1**  
Suatu kondisi 1 yang berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda <, >, =, <> dari suatu keadaan yang akan menghasilkan nilai **“True/Benar”** atau **“False/Salah”**.
- **Kondisi 2**  
Suatu kondisi 2 yang berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda <, >, =, <> dari suatu keadaan yang akan menghasilkan nilai **“True/Benar”** atau **“False/Salah”**.
- **Ekspresi 1**  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 1 bernilai **“true/benar”** (memenuhi syarat).

- Ekspresi 2  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 2 bernilai “**true/benar**” (memenuhi syarat).
- Ekspresi lain  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 1 dan kondisi 2 tidak memenuhi syarat atau jika kondisi 1 dan kondisi 2 bernilai “**false/salah**” (tidak memenuhi syarat).

Contoh program → program dibawah ini digunakan untuk menampilkan nilai huruf berdasarkan rata-rata nilai yang dihasilkan.

```

*menampilkan nilai huruf
IF <math>NilaiRata < 60</math> Then
  LabelNilai.Text = "g"
ElseIF <math>NilaiRata < 70</math> Then
  LabelNilai.Text = "f"
ElseIF <math>NilaiRata < 80</math> Then
  LabelNilai.Text = "c"
ElseIF <math>NilaiRata < 90</math> Then
  LabelNilai.Text = "b"
ElseIF <math>NilaiRata < 100</math> Then
  LabelNilai.Text = "A"
Else
  LabelNilai.Text = "Huruf tak valid"
End IF

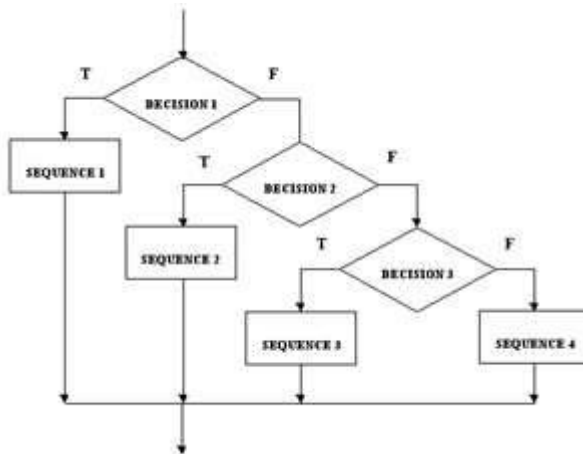
```

### 3.5. Struktur Nested IF / IF Bersarang

Pada kondisi tertentu di dalam **struktur if – then – else** bisa ditempatkan **struktur if – then – else** yang lain. Bentuk semacam ini disebut dengan **IF bersarang** atau **nested IF**. Bentuk If Bersarang dapat ditampilkan dalam bentuk diagram alir pada gambar 3.6.

Pada gambar 3.6 dijelaskan bahwa terdapat beberapa kondisi / decision. Jika decision 1 terpenuhi maka program

akan menjalankan sequence/urutan 1. Jika decision 1 tidak terpenuhi maka program akan memeriksa decision 2, jika decision 2 terpenuhi maka program menjalankan sequence/urutan 2. Jika decision 2 tidak terpenuhi maka program akan memeriksa decision 3, jika decision 2 terpenuhi maka program menjalankan sequence/urutan 2. Jika decision 3 tidak terpenuhi maka program akan menjalankan sequence/urutan 4.



Gambar 3.6. Struktur If Bersarang

**Bentuk penulisan dari struktur if bersarang sebagai berikut :**

**If** kondisi **Then**

**If** kondisi 2 **Then**

.....ekspresi 1....

**ELSE**

.....ekspresi 2....

**End If**

**End If**

Atau

**If** kondisi 1 **Then**

**If** kondisi 2 **Then**

        .....ekspresi 1....

**ELSE**

        .....ekspresi 2....

**End If**

**Else**

**If** kondisi 3 **Then**

        .....ekspresi 3....

**ELSE**

        .....ekspresi 4....

**End If**

**End If**

Keterangan :

- Kondisi 1  
Berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda <, >, =, <> dari suatu keadaan yang akan menghasilkan nilai “Benar” atau “Salah”.
- Kondisi 2  
Kondisi 2 dijalankan jika kondisi 2 adalah “benar ” (terpenuhi).
- Kondisi 3  
Kondisi 3 dijalankan jika kondisi 3 adalah “benar ” (terpenuhi).
- Ekspresi 1  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 1 bernilai “benar” (memenuhi syarat).
- Ekspresi 3

Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 3 bernilai “benar” (memenuhi syarat).

- Ekspresi 4

Rangkaian kode program (blok program) yang akan dijalankan jika kondisi 4 bernilai “benar” (memenuhi syarat).

Contoh Program → program dibawah ini digunakan untuk menampilkan harga tiket pesawat berdasarkan pesawat yang digunakan dan tujuannya.

```

Private Sub tujuan_SelectedIndexChanged(ByVal sender
  IF maskapai.Text = "Garuda" Then
    IF tujuan.Text = "Medan-Padang" Then
      harga.Text = 400000
    ElseIf tujuan.Text = "Medan-Jakarta" Then
      harga.Text = 300000
    ElseIf tujuan.Text = "Medan-Bali" Then
      harga.Text = 1000000
    End IF
  ElseIf maskapai.Text = "Lion" Then
    IF tujuan.Text = "Medan-Padang" Then
      harga.Text = 300000
    ElseIf tujuan.Text = "Medan-Jakarta" Then
      harga.Text = 400000
    ElseIf tujuan.Text = "Medan-Bali" Then
      harga.Text = 300000
    End IF
  ElseIf maskapai.Text = "Sriwijaya" Then
    IF tujuan.Text = "Medan-Padang" Then
      harga.Text = 400000
    ElseIf tujuan.Text = "Medan-Jakarta" Then
      harga.Text = 400000
    ElseIf tujuan.Text = "Medan-Bali" Then
      harga.Text = 300000
    End IF
  End IF
End Sub

```

### 3.6. Struktur IIF

Bentuk penulisan dari **if – then – else** adalah tegak atau **vertikal**. Bentuk semacam ini tentu banyak memakan baris atau terlalu panjang, apalagi jika bentuknya nested if. Untuk mengatasi kekurangan tersebut, bisa digunakan



**struktur IIF** yaitu **penulisan struktur if – then – else yang bentuknya datar atau horisontal**. Bentuk penulisan IIF membutuhkan variabel untuk menampung jawabannya. Dari segi kecepatan struktur IIF lebih cepat prosesnya dibandingkan dengan struktur if – then – else.

**Bentuk penulisan dari struktur IIF sebagai berikut :**

**Hasil = IIF (kondisi, jawaban 1, jawaban 2)**

Keterangan :

- **Kondisi**  
Berisi perbandingan antara satu variabel/konstanta dengan variabel/konstanta lain dengan menggunakan tanda <, >, =, <> dari suatu keadaan yang akan menghasilkan nilai “Benar” atau “Salah”.
- **Jawaban 1**  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisinya bernilai “benar” (memenuhi syarat).
- **Jawaban 2**  
Rangkaian kode program (blok program) yang akan dijalankan jika kondisinya bernilai “salah” (tidak memenuhi syarat).

Berikut terdapat kode program untuk menampilkan teks jenis kelamin laki-laki atau perempuan dari pilihan user, kode program ditulis dengan bentuk If-Then-Else.



Contoh Program :

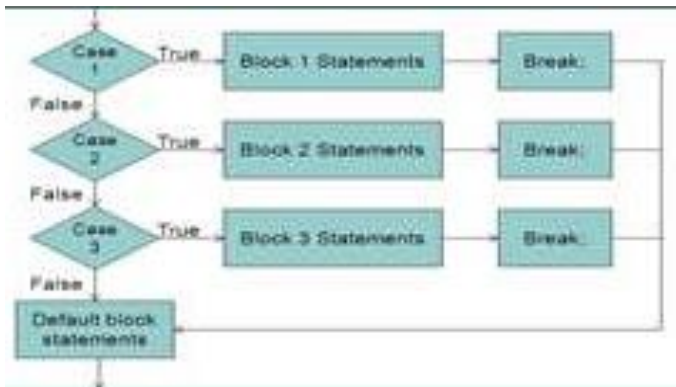
Jika kode program dalam bentuk If-Then-Else di atas diubah penulisannya menjadi bentuk IIF maka bentuknya sebagai berikut.

```
Private Sub Form_Click()
    Dim jenis as string
    Dim status as string
    Jenis = inputbox("jenis kelamin (P / W)")
    Status = IIF (jenis = "P", "Pria", IIF (jenis = "W", "Wanita", ""))
    Print "Jenis Kelamin : " & status
End Sub
```

### 3.7. Struktur Select Case

Struktur **select case** digunakan untuk menjalankan satu **blok perintah yang jumlahnya banyak** atau **bertingkat-tingkat**. Struktur ini mirip dengan struktur if – then – else, tetapi select case mempunyai penulisan yang lebih mudah sehingga penulisan programnya lebih efisien dan mudah dibaca oleh pembuat program.

Perbedaan antara select case dengan struktur if – then – else adalah bila struktur if – then – else menyeleksi suatu kondisi dan terpenuhi, selanjutnya proses penyeleksian masih dilakukan terhadap struktur if – then – else berikutnya. Sedangkan pada stuktur select case bila suatu kondisi sudah terpenuhi dan blok program yang telah diproses, selanjutnya blok program lainnya dalam lingkungan select case tidak akan diseleksi lagi. Bentuk Select-Case dapat ditampilkan dalam bentuk diagram alir pada gambar 3.7.



Gambar 3.7. Struktur Select Case

Pada gambar 3.7 dijelaskan bahwa terdapat beberapa kondisi. Jika kondisi 1 benar maka program akan menjalankan blok statemen 1. Jika kondisi 1 salah maka program akan memeriksa kondisi 2, jika kondisi 2 benar maka program akan menjalankan blok statemen 2. Jika kondisi 2 salah maka program akan memeriksa kondisi 3, jika kondisi 3 benar maka program akan menjalankan blok statemen 3. Jika kondisi 3 salah maka program akan menjalankan blok statemen default.

**Bentuk penulisan dari struktur select case :**

**Select Case** tes\_ekspresi atau diketik letak penulisan tes\_ekspresi

**Case** Ekspresi 1

.....Blok Perintah 1.....

**Case** Ekspresi 2

.....Blok Perintah 2.....

**Case Else**

.....Blok Perintah 3.....

**End Select**

Keterangan :

- Tes\_ekspresi  
Tes\_ekspresi ini bisa bernilai numerik maupun string.
- Ekspresi 1  
Nilai kedua proses seleksi dari tes\_ekspresi.
- Ekspresi 2  
Nilai kedua proses seleksi dari tes\_ekspresi.  
Penulisan nilai pada ekspresi 1 maupun ekspresi 2 bisa langsung berbentuk konstanta atau melalui operator perbandingan.
- Blok perintah 1  
Suatu rangkaian perintah yang dikerjakan bila kondisi ekspresi 1 terpenuhi.
- Blok perintah 2  
Suatu rangkaian perintah yang dikerjakan bila kondisi ekspresi 2 terpenuhi.
- Blok perintah 3  
Suatu rangkaian perintah yang dikerjakan bila kondisinya tidak ada yang memenuhi syarat.

Contoh program → program dibawah ini digunakan untuk menampilkan harga tiket pesawat berdasarkan pesawat yang digunakan dan tujuannya dalam bentuk Select-Case.

```

Private Sub tujuan_SelectedIndexChanged(ByVal m
.Select Case (tujuan.Text)
  Case "Sarada"
    Select Case (tujuan.Text)
      Case "Medan-Padang"
        harga.Text = 600000
      Case "Medan-Jakarta"
        harga.Text = 800000
      Case "Medan-Bali"
        harga.Text = 900000
    End Select
  Case "Lion"
    Select Case (tujuan.Text)
      Case "Medan-Padang"
        harga.Text = 300000
      Case "Medan-Jakarta"
        harga.Text = 400000
      Case "Medan-Bali"
        harga.Text = 700000
    End Select
  Case "Pulaujaya"
    Select Case (tujuan.Text)
      Case "Medan-Padang"
        harga.Text = 400000
      Case "Medan-Jakarta"
        harga.Text = 700000
      Case "Medan-Bali"
        harga.Text = 800000
    End Select
End Select

```

### 3.8. Latihan Program

Program ini mengenai perhitungan nilai, yang diberi nama form perhitungan nilai. Form perhitungan nilai ini menjelaskan tentang menghitung nilai akhir mata kuliah yaitu nilai akhir dan nilai huruf berdasarkan nilai tugas, nilai uts, dan nilai uas. Untuk menghitung nilai akhir didapatkan dari total dari nilai tugas 40%, nilai uts 30%, dan nilai uas 30%. Sedangkan untuk nilai huruf mempunyai ketentuan sebagai berikut :

- Nilai huruf A : 80 – 100
- Nilai huruf B : 70 - 79
- Nilai huruf C : 50 - 69
- Nilai huruf D : 40 - 49
- Nilai huruf E : 0 – 39

Tampilan form perhitungan nilai dapat dilihat pada gambar 3.9.

Gambar 3.8. Desain Form Perhitungan Nilai

Berikut aturan masing-masing tool dengan properties pada form perhitungan nilai.

Tabel 3.2. Tabel Properties Form Perhitungan Nilai

Tools	Properties	Settings
Form 2	BackColor	255; 255; 192
	Font	Times New Roman; Bold; 12
	Text	Hitung Nilai
Label 1	Text	Perhitungan Nilai
	Font	Times New Roman; 24
Label 2	Text	Nilai Tugas
Label 3	Text	Nilai UTS
Label 4	Text	Nilai UAS
Label 5	Text	Total Nilai
Label 6	Text	Nilai Huruf
Textbox 1	Name	txtTugas
Textbox 2	Name	txtUts
Textbox 3	Name	txtUas
Textbox 4	Name	txtTotNilai
	Enabled	False (agar textbox tidak diisi user)
Textbox 5	Name	txtNH
	Enabled	False
Button 1	Text	OK
	Name	btnOk
Button 2	Text	Bersih
	Name	btnBersih
Button 3	Text	Cetak

	Name	btnCetak
Button 4	Text	Keluar
	Name	btnKeluar

Kode program yang harus ditambahkan agar program berjalan sesuai dengan yang diinginkan adalah :

- Tambahkan kode program berikut pada **tombol ok (btnOk)**. Kode program ini berfungsi untuk melakukan proses perhitungan nilai total dan proses perhitungan nilai huruf.

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles btnOk.Click
    'perhitungan nilai total
    txtTotNilai.Text = (0.4 * txtTugas.Text) + (0.3 * txtUts.Text) + (0.3 * txtUas.Text)
    'perhitungan nilai huruf
    If txtTotNilai.Text >= 80 And txtTotNilai.Text <= 100 Then
        txtNH.Text = "A"
    ElseIf txtTotNilai.Text >= 70 And txtTotNilai.Text < 80 Then
        txtNH.Text = "B"
    ElseIf txtTotNilai.Text >= 50 And txtTotNilai.Text < 70 Then
        txtNH.Text = "C"
    ElseIf txtTotNilai.Text >= 40 And txtTotNilai.Text < 50 Then
        txtNH.Text = "D"
    ElseIf txtTotNilai.Text = 0 And txtTotNilai.Text < 40 Then
        txtNH.Text = "E"
    End If
End Sub
```

- Tambahkan kode program berikut pada **tombol bersih (btnBersih)**. Kode program ini berfungsi untuk melakukan proses membersihkan semua textbox dan meletakkan kursor di textbox nilai tugas (txtTugas).

```
Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles btnBersih.Click
    'mengosongkan semua textbox
    txtTugas.Text = ""
    txtUts.Text = ""
    txtUas.Text = ""
    txtTotNilai.Text = ""
    txtNH.Text = ""
    txtTugas.Focus()
End Sub
```

- Tambahkan kode program berikut pada **tombol cetak (btnCetak)**. Kode program ini berfungsi untuk melakukan

proses menampilkan semua data pada pesan (*messagebox*) yaitu data data nilai tugas, nilai uts, nilai uas, total nilai, dan nilai huruf.

```
Private Sub btnCetak_Click(sender As Object, e As EventArgs) Handles
btnCetak.Click
    MsgBox("Nilai Tugas = " & txtTugas.Text & vbCrLf & "Nilai UTS = " &
        txtUts.Text & vbCrLf & "Nilai UAS = " & txtUas.Text & vbCrLf & "Total
        Nilai = " & txtTotNilai.Text & vbCrLf & "Nilai Huruf = " & txtNH.Text)
End Sub
```

- Tambahkan kode program berikut pada **tombol keluar (btnKeluar)**. Kode program ini berfungsi untuk melakukan menampilkan form login dan menyembunyikan form perhitungan nilai.

```
Private Sub btnKeluar_Click(sender As Object, e As EventArgs) Handles
btnKeluar.Click
    Form1.Show()
    Me.Hide()
End Sub
```



## BAB IV STRUKTUR PENGULANGAN

Bab 4 ini menjelaskan tentang Pengantar Struktur Perulangan, **For ... Next For Each ... Next, While ... End While, Do While ... Loop, Do Until ... Loop, Do ... Loop While, Do ... Loop Until**, dan latihan program.

### 4.1. Pengantar Struktur Pengulangan

**Pengulangan** adalah proses mengeksekusi sebuah atau beberapa pernyataan secara berulang-ulang sampai kondisi yang membatasi pengulangan tersebut terpenuhi. Apabila suatu pengulangan tidak mempunyai kondisi yang membatasi proses pengulangannya, maka akan terjadi pengulangan abadi (*infinite looping*).

**Struktur pengulangan atau loop atau looping** digunakan untuk mengulang suatu blok perintah sampai kondisi tertentu. Proses pengulangan ini dapat dikendalikan jumlahnya oleh aplikasi yang dibuat pada kondisi tertentu.

Proses pengulangan akan terus dikerjakan selama kondisi yang dibandingkan oleh proses tersebut masih bernilai **“benar” (true)**. Jika kondisi yang dibandingkan bernilai **“salah” (false)**, proses pengulangan akan berhenti dan jalannya program akan dilanjutkan setelah proses pengulangan.

Bentuk pengenalan pada visual basic .Net adalah :

- a. **For ... Next**
- b. **For Each ... Next**
- c. **While ... End While**
- d. **Do While ... Loop**
- e. **Do Until ... Loop**

- f. **Do ... Loop While**
- g. **Do ... Loop Until**

#### 4.2. Struktur For ... Next

**Struktur For – Next** digunakan untuk mengulang blok perintah dalam jumlah yang telah ditentukan atau mengulang blok perintah apabila diketahui sampai berapa kali pengulangan akan dilakukan. Itu karena pada pengulangan bentuk ini, harus ditentukan nilai awal dan nilai akhir dari pengulangan yang akan dilakukan. Nilai awal dan nilai akhir variabel penghitung inilah yang menjadi kondisi pada pengulangan bentuk For...Next. Nilai variabel penghitung ini akan secara otomatis bertambah atau berkurang setiap kali suatu pengulangan dikerjakan. **Bentuk penulisan dari struktur For – Next yaitu:**

**For counter = awal to akhir step pertambahan**  
 .... ekspresi ....

**Next counter**

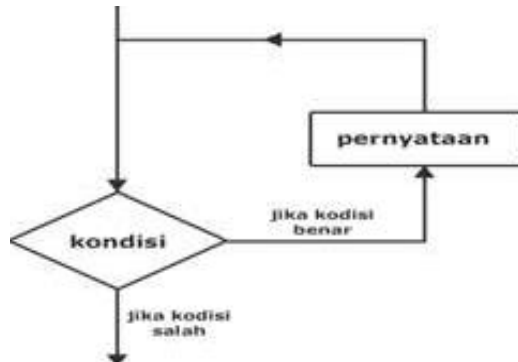
Keterangan :

- Counter → nama variabel integer yang digunakan untuk melakukan proses pengulangan.
- Awal → nilai suatu variabel integer untuk menentukan harga awal suatu pengulangan.
- Akhir → nilai suatu variabel integer untuk menentukan harga akhir suatu pengulangan.
- Pertambahan → besarnya nilai perubahan dari nilai awal hingga nilai akhir. Jika pengulangannya menurun yaitu dari nilai besar menuju nilai kecil, maka nilai perubahannya harus negatif. Nilai standar visual basic

untuk nilai pertambahan adalah 1, kecuali jika didefinisikan sendiri.

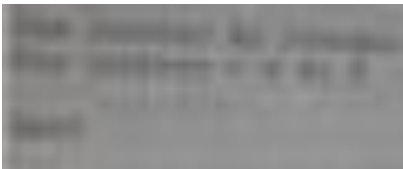
- Ekspresi → suatu blok perintah yang akan dikerjakan jika kondisi dari proses pengulangan memenuhi syarat.

Struktur perulangan For...Next digambarkan pada gambar 4.1 di bawah ini. Pada gambar 4.1 dijelaskan langkah perulangan pada struktur For...Next, dimana jika kondisi bernilai benar maka pernyataan / ekspresi yang berisi blok program akan dikerjakan. Sebaliknya, jika kondisi bernilai salah maka pernyataan / ekspresi yang berisi blok program tidak akan dikerjakan.



Gambar 4.1. Struktur Perulangan For...Next

Contoh bentuk pengulangan For...Next dengan pencacah naik atau positif yaitu :



Contoh bentuk pengulangan For...Next dengan pencacah turun atau negatif yaitu :



#### 4.3. Struktur For Each ... Next

Pengulangan For Each...Next sama dengan pernyataan For...Next, kecuali variabel pengulangan yang diperlukan bukan numerik melainkan array atau collection. Pengulangan For Each akan mengulang untuk **setiap elemen dalam array atau collection**. Bentuk penulisan perintah For Each sebagai berikut :

**For Each** <elemen> **In** <Grup>

.... ekspresi ....

**Next** <nama\_variabel>

<Elemen> harus sama tipe datanya dengan tiap-tiap elemen dalam array atau collection. <Grup> adalah array atau collection. Langkah pengulangan secara otomatis melalui tiap-tiap elemen dalam array atau collection dan keluar dari pengulangan setelah sampai pada akhir array atau collection. Contoh bentuk pengulangan For Each yaitu :



#### 4.4. Struktur While ... End While

Bentuk pengulangan **While ... End While berhenti jika kondisi telah bernilai salah (false)**. Kondisi dari bentuk pengulangan ini diletakkan di awal. Bentuk penulisan perintah **While ... End While** sebagai berikut :

**While** <kondisi>

.... ekspresi ....

### End While

Pada bentuk pengulangan **While ... End While** bagian pengulangan ada kemungkinan tidak dieksekusi. Itu karena jika kondisi pengulangan tidak terpenuhi, maka ekspresi-ekspresi tidak akan dieksekusi atau dijalankan. Contoh pengulangan bentuk pengulangan **While ... End While** dalam program yaitu :



#### 4.5. Struktur Do While ... Loop

Pengulangan **Do While ... Loop** merupakan loop yang sangat mirip dengan pengulangan **While ... End While**. Struktur Do While – Loop digunakan untuk melakukan pengulangan terus-menerus selama kondisi memenuhi syarat (bernilai **True**). Pengulangan ini akan berhenti selama kondisinya tidak memenuhi syarat (bernilai **False**).

Struktur penulisan dari proses pengulangan ini diawali dengan kata **Do While** dan diakhiri dengan kata **Loop**. Jika visual basic menemukan struktur Do While – Loop maka yang pertama kali dikerjakan adalah menguji kondisi. Jika kondisinya terpenuhi maka blok program antara Do While – Loop akan dikerjakan, sedangkan jika sebaliknya, visual basic akan mengabaikan blok perintah Do While – Loop dan meneruskan mengeksekusi rutin/perintah di bawah kata **Loop**. Bentuk pengulangan **Do While ... Loop** adalah sebagai berikut :

```

Do While <kondisi>
    .... ekspresi ....
Exit Do
    .... ekspresi ....
  
```

## Loop

Contoh pengulangan bentuk pengulangan **Do While**

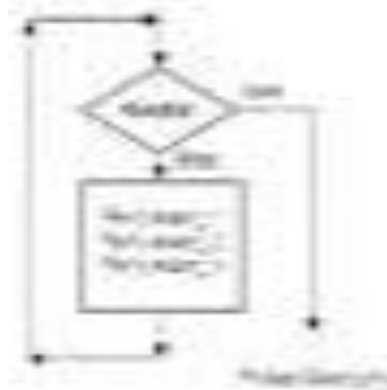
... **Loop** dalam program yaitu :

```

Dim counter As Integer
counter = 0

'kondisi pengulangan
Do While counter <= 10

    counter = counter + 1
Loop
  
```



Gambar 4.2. Struktur Perulangan Do While – Loop

Struktur perulangan Do While – Loop digambarkan pada gambar 4.2 di bawah ini. Pada gambar 4.2 dijelaskan langkah perulangan pada struktur Do While – Loop, dimana jika kondisi bernilai benar maka pernyataan / ekspresi yang berisi blok program akan dikerjakan. Sebaliknya, jika kondisi bernilai salah maka pernyataan / ekspresi yang berisi blok program tidak akan dikerjakan dan akan mengerjakan proses selanjutnya.

Yang membedakan **Do While ... Loop** dengan **While ... End While** adalah perintah **Exit Do**. Perintah tersebut adalah perintah untuk memaksa program menghentikan pengulangan sebelum

pengulangan berakhir pada bentuk pengulangan **Do While ... Loop**. Perintah ini digunakan pada kondisi tertentu saja. jika menggunakan perintah ini harus ada kondisi yang mendahuluinya. Contoh :

```

DIM counter AS Integer
counter = 0

Do While counter <= 10 'kondisi pengulangan
    LE counter = 5 Then Exit Do 'kondisi exit do
    counter = counter + 1
Loop
  
```

Pada contoh tersebut walaupun pengulangan harus dilakukan sampai sepuluh kali, tetapi dengan menggunakan perintah **Exit Do**, program memaksa pengulangan hanya dilakukan sampai lima kali.

Perbedaan syarat dan pemakaian antara Do While – Loop dengan For – Next dijelaskan pada tabel 4.1.

Tabel 4.1. Tabel Perbedaan Do While – Loop dengan For – Next

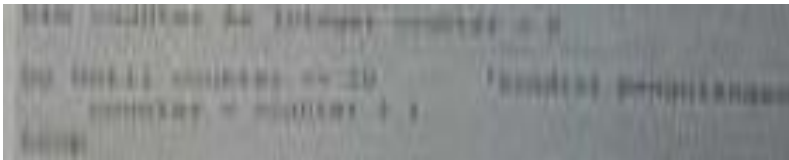
No	Do While – Loop	For - Next
1	Nilai pada kondisi bisa bertipe number atau string	Nilai pada kondisi bisa bertipe number
2	Batasan perulangannya bisa terbatas atau tidak berhingga	Batasan perulangannya bisa terbatas
3	Variabel yang dipakai pada kondisi harus diberi harga awal	Variabel yang dipakai pada kondisi tidak harus diberi harga awal
4	Perubahan nilai pada kondisi yang bertipe number harus dilakukan secara manual	Perubahan nilai pada kondisi dilakukan secara otomatis
5	Suatu blok perintah dalam Do While – Loop bisa dikerjakan tanpa harus menyeleksi nilai kondisi terlebih dahulu	Suatu blok perintah dalam For – Next tidak bisa dikerjakan sebelum kondisi terseleksi

#### 4.6. Struktur Do Until ... Loop

Bentuk pengulangan **Do Until ... Loop** hampir sama dengan bentuk pengulangan **While ... End While** dan **Do While ... Loop**. Perbedaannya adalah kalau pada pengulangan **While ... End While** dan **Do While ... Loop** pengulangan dihentikan pada waktu kondisi bernilai salah (*false*), sedangkan pada bentuk **Do Until ... Loop** pengulangan dihentikan pada waktu kondisi bernilai benar (*true*). Bentuk pengulangan **Do Until ... Loop** adalah :

```
Do Until <kondisi>
.... ekspresi ....
Loop
```

Contoh pengulangan bentuk pengulangan **Do Until ... Loop** dalam program yaitu :



Pada bentuk pengulangan **Do Until ... Loop** bagian pengulangan ada kemungkinan tidak dieksekusi sama sekali jika kondisi berhentinya terpenuhi. Itu juga terjadi pada bentuk pengulangan **While ... End While** dan **Do While ... Loop**.

#### 4.7. Struktur Do ... Loop While

Bentuk pengulangan **Do ... Loop While** berhenti jika kondisi telah bernilai salah (*false*). Jadi selama kondisi masih bernilai benar (*true*), pengulangan akan terus dilakukan. Bentuk pengulangan **Do ... Loop While** merupakan pengulangan yang kondisinya terletak di akhir pengulangan. Bentuk pengulangan **Do ... Loop While** adalah :

```
Do
.... ekspresi ....
Loop While
```



### Loop While <kondisi>

Contoh pengulangan bentuk pengulangan **Do ... Loop While** dalam program yaitu :

```
Dim counter As Integer
counter = 0
Do
    counter = counter + 1
Loop While counter < 10 'kondisi pengulangan
```

Pada bentuk pengulangan **Do ... Loop While** bagian pengulangan **minimal akan dieksekusi sebanyak satu kali**. Itu karena bagian pengulangan dijalankan terlebih dahulu baru kemudian bagian kondisi pengulangan diperiksa. Contoh :

```
Dim counter As Integer
counter = 10
Do
    counter = counter + 1
Loop While counter < 10 'kondisi pengulangan
```

Terlihat pada contoh tersebut, walaupun kondisi tidak terpenuhi tetapi bagian pengulangan dijalankan sekali. Hal itu, karena bagian pengulangan dijalankan terlebih dahulu baru kemudian di bagian akhir diadakan pemeriksaan kondisi.

#### 4.8. Struktur Do ... Loop Until

Bentuk pengulangan **Do ... Loop Until** hampir sama dengan bentuk pengulangan **Do ... Loop While**. Struktur Do – Loop Until juga merupakan struktur pengulangan yang metodenya kebalikan dari struktur Do While – Loop. Pengulangan berhenti jika kondisi telah bernilai benar (*true*). Jadi, selama kondisi bernilai salah (*false*), pengulangan akan terus dilakukan. Bentuk pengulangan **Do**

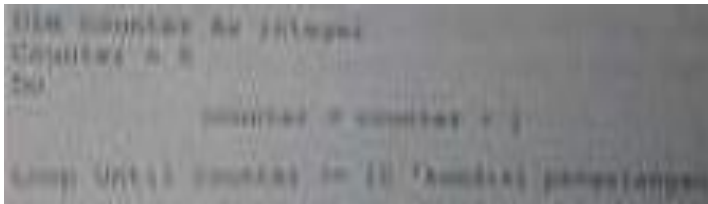
... **Loop Until** merupakan pengulangan yang pemeriksaan kondisinya terletak di akhir pengulangan sehingga pengulangan minimal dilakukan satu kali (mirip dengan **Do ... Loop While**). Bentuk pengulangan **Do ... Loop Until** adalah sebagai berikut :

**Do**

.... ekspresi ....

**Loop Until <kondisi>**

Contoh pengulangan bentuk pengulangan **Do ... Loop Until** dalam program yaitu :



#### 4.9. Latihan Program

Form 1 ini digunakan untuk menampilkan angka 1-10. Untuk menampilkan angka 1-10 ini digunakan struktur pengulangan dengan berbagai macam penulisan yaitu dengan for-next, do-while, dan do loop-until. Untuk penulisan pengulangan dengan for-next diberikan dua contoh yaitu menampilkan angka yang bertambah naik dan menampilkan angka yang turun.



Gambar 4.3. Desain Form 1

Berikut aturan masing-masing tool pada form 1 dengan properties yang ditampilkan pada tabel 4.2.

Tabel 4.2. Tabel Tool dan Properties pada Form 1

Tool	Properties	Setting
Form 1	Text	Menampilkan angka 1-10
Label 1	Text	Angka 1-10
Button 1	Text	Struktur For –Next Tambah
	Name	btnFor
Button 2	Text	Struktur For –Next Kurang
	Name	btnForKurang
Button 3	Text	Struktur Do While - Loop
	Name	btnDoWhile
Button 4	Text	Struktur Do Loop - Until
	Name	btnDoLoop
Listbox 1	-	-

Tambahkan kode program form 1 berikut agar program dapat berjalan sesuai dengan keinginan.

- Tambahkan kode program di posisi paling atas. Kode program ini untuk pengenalan variabel.

```
Dim bil1, bil2, bil3, bil4 As Integer
```

- Tambahkan kode program berikut pada button Struktur For-Next untuk menampilkan angka 1-10 dengan kenaikan angka sebanyak 1.

```
Private Sub btnFor_Click(sender As Object, e As EventArgs) Handles
btnFor.Click
    ListBox1.Items.Clear()
    For bil1 = 1 To 10
        ListBox1.Items.Add(bil1)
    Next
End Sub
```

- Tambahkan kode program berikut pada button Struktur For-Next Kurang untuk menampilkan angka 1-10 dengan penurunan angka sebanyak 1.

```
Private Sub btnForKurang_Click(sender As Object, e As EventArgs) Handles
btnForKurang.Click
    ListBox1.Items.Clear()
    For bil2 = 10 To 1 Step -1
        ListBox1.Items.Add(bil2)
    Next
End Sub
```

- Tambahkan kode program berikut pada button Struktur Do While-Loop untuk menampilkan angka 1-10 dengan kenaikan angka sebanyak 1.

```
Private Sub btnDoWhile_Click(sender As Object, e As EventArgs) Handles
btnDoWhile.Click
    ListBox1.Items.Clear()
    bil3 = 1
    Do While bil3 <= 10
        ListBox1.Items.Add(bil3)
        bil3 = bil3 + 1
    Loop
End Sub
```

- Tambahkan kode program berikut pada button Struktur Do Loop-Until untuk menampilkan angka 1-10 dengan kenaikan angka sebanyak 1.

```
Private Sub btnDoLoop_Click(sender As Object, e As EventArgs) Handles
btnDoLoop.Click
    ListBox1.Items.Clear()
```

```
bil4 = 1
Do Until bil4 > 10
    ListBox1.Items.Add(bil4)
    bil4 = bil4 + 1
Loop
End Sub
```

## BAB V

### ARRAY

Pada bab 5 ini akan dijelaskan tentang array diantaranya pendeklarasian dan pengisian array, array multidimensi, dan array dinamis.

Array adalah sekelompok lokasi memori yang bertetangga yang memiliki nama sama dan tipe sama. Untuk merujuk ke lokasi tertentu dalam memori atau sebuah elemen di dalam suatu array, perlu menspesifikasikan nama array dan nomor posisi elemen array yang ditunjuk. Nomor posisi adalah nilai yang mengindikasikan lokasi spesifik di dalam array. (Sianipar, 2014). Nomor posisi disebut indeks / subscript / subscript. Indeks adalah sebuah integer atau ekspresi integer.

Array merupakan tipe terstruktur yang terdiri dari sejumlah komponen yang mempunyai tipe sama. Komponen-komponen ini disebut dengan tipe komponen (*component type*) atau tipe basis (*base type*). Banyaknya komponen dalam suatu array ditunjukkan oleh suatu indeks yang disebut dengan tipe indeks (*indeks type*). (Subari & Yuswanto, 2008).

Gambaran array dapat dilihat pada gambar 5.1. Pada gambar 5.1 menampilkan sebuah array integer bernama arrayAngka. Array ini memuat 12 elemen, yang semuanya dapat

dirujuk dengan memberikan nama diikuti dengan nomor posisi elemen (diapit dengan tanda kurung). **Elemen pertama di dalam sebuah array adalah elemen ke nol**, jadi, elemen pertama pada array arrayAngka dirujuk sebagai arrayAngka(0), elemen kedua sebagai arrayAngka(1), dan seterusnya. Elemen ke-i arrayAngka dinyatakan sebagai arrayAngka(i-1).

Nama array  
 (perhatikan bahwa nama  
 elemen array  
 memiliki nama  
 sama, arrayAngka)

arrayAngka (0)	-65
arrayAngka (1)	6
arrayAngka (2)	0
arrayAngka (3)	72
arrayAngka (4)	1943
arrayAngka (5)	-89
arrayAngka (6)	0
arrayAngka (7)	62
arrayAngka (8)	-3
arrayAngka (9)	1
arrayAngka (10)	6453
arrayAngka (11)	78

Nomor posisi elemen  
 di dalam array  
 arrayAngka.

Gambar 5.1. Array dengan 12 Elemen [Sianipar:2014]

Pada suatu waktu perlu menyimpan sejumlah nilai dengan tipe yang sama ke dalam beberapa variabel. Akan tetapi, jika variabel yang dibutuhkan sangat banyak, gunakan array untuk menggantikan variabel-variabel tersebut.

Sebagai contoh, kode program biasa yang digunakan untuk mendeklarasikan beberapa variabel dengan tipe yang sama adalah :

```
Dim hari1, hari2, hari3 As String
```

Atau

```
Dim hari1 As String  
Dim hari2 As String
```

```
Dim hari3 As String
```

Untuk menyederhanakannya gunakan array. Dengan menggunakan array pendeklarasian menjadi lebih efisien.

```
Dim hr(2) As String
hr(0) = "senin"
hr(1) = "selasa"
hr(2) = "Rabu"
```

### 5.1. Deklarasi dan Pengisian Array

Suatu array yang akan digunakan harus dideklarasikan terlebih dahulu. Deklarasi pada array hampir sama dengan deklarasi variabel biasa, hanya pada array harus ditambahkan nilai indeks (jumlah elemen) yang diletakkan di dalam tanda kurung (). Berikut cara mendeklarasikan array yaitu :

**[dim|public|private|friend] <nama\_array> (ukuran\_array) AS <tipe\_data>**

Contoh : 

```
Dim hr(2) As String
```

Pada contoh program array di atas bernama hr dan memiliki 3 data dengan indeks 0 hingga 2. Batas bawah array adalah bernilai 0 (dapat dilihat dengan fungsi **LBound** (<nama\_array>) atau Lower Bound, contoh : txtHari.Text = LBound(hr). sedangkan untuk melihat batas atas dari suatu array adalah dengan menggunakan fungsi **Ubound** (<nama\_array>) atau Upper Bound, contoh : UBound(hr).

Untuk mengisi elemen array dilakukan dengan cara :

**<nama\_array> (<indeks\_ke->)=<isi\_elemen\_array>**

Contoh :

```
hr(0) = "senin"
hr(1) = "selasa"
hr(2) = "Rabu"
```

Pada program di atas, array hr diberi isian data berdasarkan urutan elemen. Elemen 0 diisi senin, elemen 1 diisi selasa, dan elemen 3 diisi rabu.

## 5.2. Array multidimensi

Pada pembahasan sebelumnya, dijelaskan tentang array yang mempunyai sebuah indeks (subscript) hanya sebuah. Bentuk array semacam ini disebut **Array Dimensi Satu** (*One Dimensional Array*). Di samping itu, ada juga array yang mempunyai banyak dimensi yang disebut dengan **Array Multi Dimensi** atau **Array Berdimensi Banyak** (*Multidimensional Array*). Pemakaian array multi dimensi (dimensi 2, dimensi 3, dst.) bertujuan untuk meningkatkan daya tampung data dan data akan lebih mudah dikelola. (Subari & Yuswanto, 2008). Gambaran array dimensi 2 dapat dilihat pada gambar 5.2.

The image shows three tables representing arrays. The first two are 1D arrays, and the third is a 2D array.

	Nama(3)	Umur(3)	NamaUmur(3,1)	
0	Raka	17	Mario	17
1	Mario	25	Fatih	25
2	Fatih	18	Fari	18
3	Fari	12	Raka	12

Gambar 5.2. Array 1 dan 2 Dimensi [Hidayatulloh : 2015]

Pada gambar 5.2. terdapat dua buah array berdimensi satu yaitu array nama dengan indeks sebanyak 3 dan array umur dengan indeks sebanyak 3. Juga terdapat satu buah array berdimensi dua yang bernama array nama\_umur dengan indeks baris sebanyak 3 dan indeks kolom sebanyak 1. Pada array dua dimensi ini membentuk **tabel** yang mempunyai dua indeks yaitu **baris dan kolom**.

Untuk mendeklarasikan array dua dimensi atau lebih, dapat dilakukan dengan dua cara yaitu :

**Cara pertama pendeklarasian array dua dimensi :**



```
[dim|public|private|friend] <nama_array>
(ukuran_array_dimesi1, ukuran_array_dimesi2, ...) AS
<tipe_data>
```

```
Contoh : Dim array2(1, 1) As String
```

Pada contoh di atas, dideklarasikan array dua dimensi dengan nama array2 yang mempunyai 2 baris dan 2 kolom.

Untuk menampilkannya array pada array2 dengan

cara :

```
array2(0, 0) = "Nama"
array2(0, 1) = "Jenis"
```

Maksud dari contoh di atas, array2 pada posisi baris 0 dan kolom 0 diisi dengan data “Nama”, serta array2 pada posisi baris 0 dan kolom 1 diisi dengan data “Jenis”.

Untuk memberikan isian array pada array2 dengan

cara :

```
Dim isiarray(1) As String
isiarray(0) = txtNama.Text
isiarray(1) = cbJenis.Text
```

Pada saat melakukan pengisian array dua dimensi tetap mendeklarasikan array satu dimensi yang dimaksudkan untuk menyimpan banyaknya data dalam array. Terdapat array satu dimensi dengan nama isiarray yang mempunyai ukuran sebanyak dua. Hal ini berarti bahwa array isiarray dapat menyimpan dua data yaitu data pertama yang berindeks 0 berisi nama yang datanya diambil dari txtNama.Text. Selanjutnya, data pertama yang berindeks 1 berisi jenis yang datanya diambil dari cbJenis.Text .

### **Cara kedua pendeklarasian array dua dimensi :**

**Dengan menyisipkan koma (,) didalam tanda kurung.** Tanda koma ini diberikan untuk membedakan antara baris dan kolom. Diberikan juga isian langsung pada deklarasi tersebut dengan diikuti tanda sama dengan (=) dan data ditulis dalam tanda kurung kurawal ({}).

yang menjelaskan per item. Untuk membedakan item satu dengan yang lain dipisahkan dengan tanda koma (.). Berikut penulisannya :  
**[dim|public|private|friend] <nama\_array> (,) AS <type\_data> = {{item1}, {item2},...,{item-n}}**

Contoh :

```
Dim MH(,) As String = {"Katak", "Hewan"}, {"Padi", "Tumbuhan"}, {"Sapi", "Hewan"}}
```

Contoh di atas merupakan pendeklarasian array dua dimensi dengan nama MH dengan baris dan kolom tidak dideskripsikan. Array MH mempunyai tiga data yang terbentuk dalam 3 baris dan 2 kolom. Data array MH dapat dilihat di tabel 5.1.

Tabel 5.1. Tabel Data Array MH

	Nama (kolom 1)	Jenis (kolom 2)	
<b>Baris 1</b>	Katak	Hewan	→ Data 1
<b>Baris 2</b>	Padi	Tumbuhan	→ Data 2
<b>Baris 3</b>	Sapi	Hewan	→ Data 3

### 5.3. Array Dinamis

Saat mendeklarasikan suatu array, jumlah elemen yang dibutuhkan telah ditetapkan dari awal. Penentuan nilai elemen seperti ini membawa konsekuensi tersendiri yaitu jika nilai elemen terlalu kecil, maka array tidak akan menampung semua data. Sebaliknya, jika nilai elemen array terlalu besar, maka akan ada tempat dalam array yang tidak dimanfaatkan dan memakan banyak memori. Untuk itu, perlu mendeklarasikan array secara dinamis. Langkah-langkah dalam mendeklarasikan array dinamis adalah :

1. Deklarasikan array dengan ukuran kosong.

Contoh: `Dim angka() As Integer`

2. Deklarasikan ulang variabel array dengan perintah **redim** dan tuliskan ukuran array yang diinginkan tanpa menulis ulang tipe datanya. Contoh : **ReDim** angka(5)

Kelemahan penggunaan **redim** adalah tiap kali perintah dipanggil, semua nilai pada elemen sebelumnya akan hilang. Untuk mengatasinya, VB membuat perintah **preserve** untuk menambah elemen array tanpa menghapus nilai sebelumnya. Contoh : **ReDim Preserve** angka(9)

3. Deklarasikan array tersebut secara berulang-ulang dengan ukuran yang berbeda-beda sesuai kebutuhan.

Array dinamis dapat juga diterapkan pada array multidimensi seperti contoh berikut :

```
Dim matriks () as double
... 'kode
Redim matriks (12, 12, 12)
```

## 5.4. Latihan Program

Pada latihan program ini terdapat 3 latihan program mengenai array.

### 5.4.1. Latihan Program 1

Pada latihan program 1 menampilkan nama hari dengan menggunakan array 1 dimensi. Latihan program 1 ini membahas pendeklarasin array dengan dua cara, pengisian data array yang dilakukan dengan dua cara yaitu datanya diisikan secara statis dan dinamis (melalui textbox nama hari), dan cara menampilkan array pada tool listview. Tampilan program dapat dilihat pada gambar 5.3. Tool dan properties yang digunakan pada latihan program 1 dilihat pada tabel 5.2.



Gambar 5.3. Desain Form 1

Tabel 5.2. Tabel Tool dan Properties Latihan Program 1

Tool	Properties	Setting
Form1	Text	Array 1 Dimensi
Label 1	Text	Nama Hari
TextBox 1	Name	txtHari
	Multiline	True
Button 1	Text	OK1
	Name	btnOk
Button 2	Text	OK2
	Name	btnOk2
List View 1	Name	LVHari
Label 2	Text	Batas Array
Label 3	Text	-
	Nama	lblBatasArray

Berikut kode program yang ditambahkan pada latihan program 1 agar program dapat berjalan sesuai keinginan.

- Tambahkan kode program berikut di tool form dengan event load. Kode program ini mengatur properties listview dan memberikan nama kolom dan lebar kolom pada listview.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'agar kursor fokus pada textbox txthari
    txtHari.Focus()
    'pengaturan propertyies pada tool list view
    LVHari.View = View.Details
    LVHari.GridLines = True
```

'yang ada di dalam () adl judul dan lebar kolom pada listview  
LVHari.Columns.Add("Nama Hari ", 125)

End Sub

- Tambahkan kode program berikut di tool button ok 1. Kode program ini untuk menampilkan nama hari pada listview dengan menggunakan array yang mana data array satu dimensi diambil dari textbox txthari.

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles
btnOk.Click
    'Deklarasi array cara kedua
    Dim hari(0) As String
    'Pengisian array cara kedua yang diambil dari textbox
    hari(0) = txtHari.Text
    'Cara menampilkan array pada listview
    Dim data As New ListViewItem
    data = LVHari.Items.Add(hari(0))
    txtHari.Text = ""
    txtHari.Focus()
    'Melihat batas bawah dan atas array hari
    lblBatasArray.Text = "Batas Bawah Array hari = " & LBound(hari) & vbCrLf
    & "Batas Atas Array hari = " & UBound(hari)
End Sub
```

- Tambahkan kode program berikut di tool button ok 2. Kode program ini untuk menampilkan nama hari pada listview dengan menggunakan array satu dimensi yang mana data array diisikan melalui kode program

```
Private Sub btnOk2_Click(sender As Object, e As EventArgs) Handles
btnOk2.Click
    'Deklarasi array cara pertama
    Dim hr(2) As String
    'pengisian array cara pertama
    hr(0) = "senin"
    hr(1) = "selasa"
    hr(2) = "rabu"
    'Cara menampilkan array pada listview
    Dim data As New ListViewItem
    data = LVHari.Items.Add(hr(0))
    data = LVHari.Items.Add(hr(1))
    data = LVHari.Items.Add(hr(2))
    'Melihat batas bawah dan atas array hr
```

```

lblBatasArray.Text = "Batas Bawah Array hr = " & LBound(hr) & vbCrLf &
"Batas Atas Array hr = " & UBound(hr)
End Sub

```

### 5.4.2. Latihan Program 2

Pada latihan program 2 menampilkan nama makhluk hidup dan jenisnya dengan menggunakan array 2 dimensi. Latihan program 2 ini membahas pendeklarasin array dengan dua cara, pengisian data array yang dilakukan dengan dua cara yaitu datanya diisikan secara statis dan dinamis (melalui textbox nama makhluk hidup dan combobox jenis makhluk hidup), dan cara menampilkan array pada tool listview. Tampilan program dapat dilihat pada gambar 5.4. Tool dan properties yang digunakan pada latihan program 2 dilihat pada tabel 5.3.



Gambar 5.4. Desain Form 2

Tabel 5.3. Tabel Tool dan Properties Latihan Program 2

Tool	Properties	Setting
Form2	Text	Array 2 Dimensi
Label 1	Text	Nama Makhluk Hidup
Label 2	Text	Jenis Makhluk Hidup
TextBox 1	Name	txtNama

ComboBox 1	Name	cbJenis
	Items	Hewan Tumbuhan
Button 1	Text	OK1
	Name	btnOk
Button 2	Text	OK2
	Name	btnOk2
Button 3	Text	btnBersih
	Name	Bersih
List View 1	Name	LvMh

Berikut kode program yang ditambahkan pada latihan program 2 agar program dapat berjalan sesuai keinginan.

- Buatlah prosedur tampil. Prosedur ini digunakan untuk mendeklarasikan array dua dimensi dan menampilkannya di tool listview.

```
Public Sub tampil()
    'Mendeklarasikan array dua dimensi cara pertama
    Dim array2(1, 1) As String
    'array2(0, 0) = "No"
    'array2(0, 1) = "Nama"
    'array2(1, 0) = "Jenis"
    'Menampilkan array dua dimensi
    array2(0, 0) = "Nama"
    array2(0, 1) = "Jenis"
    'agar kursor fokus pada textbox txtnama
    txtNama.Focus()
    'pengaturan propertyies pada tool list view
    LVMh.View = View.Details
    LVMh.GridLines = True
    'Menampilkan baris dan kolom pada listview yang telah
    'dideklarasikan pada array dua dimensi beserta ukuran baris dan 'kolom
    For baris = 0 To 1
        For kolom = 0 To 1
            LVMh.Columns.Add(array2(baris, kolom), 70)
        Next
    Next
Next
End Sub
```

- Tambahkan kode program berikut di tool form dengan event load. Kode program ini mengatur properties listview dan memberikan nama kolom dan lebar kolom pada list view.

```
Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'memanggil prosedur tampil
    tampil()
End Sub
```

- Tambahkan kode program berikut di tool button ok 1. Kode program ini untuk menampilkan nama makhluk hidup dan jenis makhluk hidup pada listview dengan menggunakan array dua dimensi yang mana data array diambil dari textbox txtnama dan combobox cbjenis.

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles btnOk.Click
    'pengisian array dua dimensi cara pertama
    Dim isiarray(1) As String
    'data diambil dari textbox dan combobox
    isiarray(0) = txtNama.Text
    isiarray(1) = cbJenis.Text
    'menampilkan data di listview
    Dim deret As New ListViewItem
    deret = LVMh.Items.Add(isiarray(0))
    deret.SubItems.Add(isiarray(1))
End Sub
```

- Tambahkan kode program berikut di tool button ok 2. Kode program ini untuk menampilkan nama makhluk hidup dan jenis makhluk hidup pada listview dengan menggunakan array dua dimensi yang mana data array diisikan melalui coding.

```
Private Sub btnOk2_Click(sender As Object, e As EventArgs) Handles btnOk2.Click
    'Mendeklarasikan array dan Mengisi data dua dimensi cara kedua
```



```

Dim MH(,) As String = {"Katak", "Hewan"}, {"Padi", "Tumbuhan"}, {"Sapi",
    "Hewan"}}
'Menampilkan data array
Dim deret As New ListViewItem
deret = LVMh.Items.Add(MH(0, 0))
deret.SubItems.Add(MH(0, 1))
deret = LVMh.Items.Add(MH(1, 0))
deret.SubItems.Add(MH(1, 1))
deret = LVMh.Items.Add(MH(2, 0))
deret.SubItems.Add(MH(2, 1))
btnOk2.Enabled = False
End Sub

```

- Tambahkan kode program berikut di tool button bersih. Kode program ini digunakan untuk membersihkan data yang ada di tool listview, textbox txtnama, mengaktifkan kembali button ok2, combobox cbjenis menjadi “Pilih Jenis MH”, dan mengatur kembali listview.

```

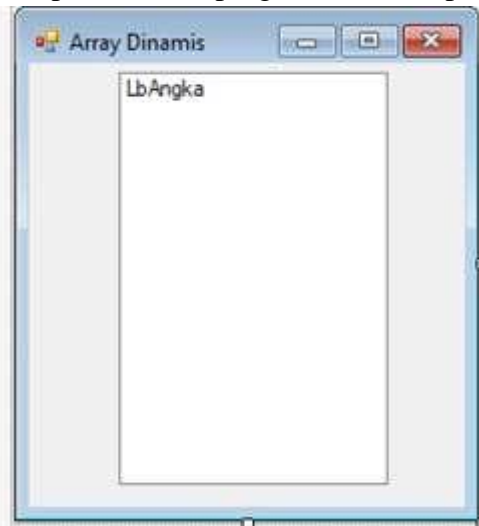
Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles
    btnBersih.Click
    txtNama.Text = ""
    'Untuk mengembalikan text di combobox cbjenis
    cbJenis.Text = "Pilih Jenis MH"
    'memberihkan data di listview
    LVMh.Clear()
    'memanggil prosedur tampil
    tampil()
    'agar tombol ok2 tidak berfungsi
    btnOk2.Enabled = True
End Sub

```

### 5.4.3. Latihan Program 3

Pada latihan program 3 menampilkan data angka dengan menggunakan array dinamis. Latihan program 3 ini membahas pendeklarasin array dinamis, pengisian data array, dan cara menampilkan array pada tool listbos. Tampilan

program dapat dilihat pada gambar 5.5. Tool dan properties yang digunakan pada latihan program 2 dilihat pada tabel 5.4.



Gambar 5.5. Desain Form 3

Tabel 5.3. Tabel Tool dan Properties Latihan Program 3

Tool	Properties	Setting
Form3	Text	Array Dinamis
ListBox 1	Name	LbAngka

Berikut kode program yang ditambahkan pada latihan program 3 agar program dapat berjalan sesuai keinginan.

- Tambahkan kode program berikut di tool form dengan event load. Kode program ini untuk menampilkan deretan angka pada tool listbox dengan mendeklarasikan array dinamis dan penggunaan keyword `redim` dan `preserve`

```
Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'Pendeklarasian array dinamis
    Dim angka() As Integer
    'menggunakan redim untuk menentukan jumlah elemen
    ReDim angka(5)
    LbAngka.Items.Add("Angka X adalah : ")
    'Jumlah batas atas variabel pada FOR TIDAK BOLEH melebihi
```

```

'jumlah array yang disediakan Batas atas For <= nilai array
For x = 1 To 5
    angka(x) = x
    LbAngka.Items.Add("==>" & angka(x))
Next
LbAngka.Items.Add(vbCrLf)
LbAngka.Items.Add("Angka Y adalah : ")
'menggunakan preserve agar data array sebelumnya
'tidak terhapus saat ditampilkan
ReDim Preserve angka(8)
For y = 6 To 8
    angka(y) = y * 100
    LbAngka.Items.Add("==>" & angka(y))
Next
LbAngka.Items.Add(vbCrLf)
LbAngka.Items.Add("Angka Z adalah : ")
For z = 1 To 8
    LbAngka.Items.Add("==>" & angka(z))
Next
End Sub

```

## BAB VI PROSEDUR DAN FUNGSI

Pada bab 6 ini dijelaskan tentang prosedur, fungsi, dan fungsi-fungsi built-in dalam vb.net. Prosedur dan fungsi (rutin) merupakan sebuah blok program yang dapat dipanggil atau digunakan untuk penerapan tugas yang diulang-ulang seperti perhitungan yang sering digunakan, (Priyanto, 2015). Keuntungan menggunakan prosedur dan fungsi adalah :

- Program lebih terstruktur sehingga lebih mudah men-debug jika terjadi kesalahan/error. Karena program terbagi-bagi menjadi komponen-komponen yang mandiri.
- Tidak harus menulis berulang kali body fungsi sehingga lebih efisien dalam bekerja.
- Prosedur yang terletak dalam suatu program dapat dipakai berulang-ulang sesuai dengan kebutuhan.

## 6.1. Prosedur

Prosedur adalah suatu blok perintah yang dijalankan sebagai tanggapan atas terbentuknya kejadian (*event*) dan tidak akan mengembalikan nilai. Pada prosedur ini **tidak ada nilai output** (*return value*). Sebuah prosedur dapat memiliki parameter ataupun tidak. Parameter pada prosedur dapat berjumlah lebih dari satu parameter. **Parameter** berfungsi untuk memberikan nilai masukan (*input*) ke prosedur. Pernyataan yang terdapat di dalam prosedur akan dieksekusi apabila prosedur dipanggil.

Sebuah prosedur didefinisikan dengan kata kunci dengan kata kunci Sub dan End Sub. Prosedur dapat ditulis pada module, class, maupun structure. Pada prosedur terdapat penentu akses seperti private, public, protected, dan friend. Secara default, prosedur mempunyai penentu akses public, artinya prosedur dapat dipanggil dari mana saja pada proyek tersebut. Berikut bentuk prosedur yaitu :

```
[Public|Private|Friend|Protected] Sub (nama_parameter)  
.....blok perintah...  
End Sub
```

Contoh :

```
Private Sub bersih()  
    txtNamaDosen.Text = ""  
    txtKotaDosen.Text = ""  
End Sub
```

Contoh program di atas merupakan contoh prosedur tanpa parameter dengan nama prosedur bersih.

Untuk memanggil prosedurnya adalah :

```
Call nama_prosedur (parameter1, parameter2)
```

Atau

```
nama_prosedur (parameter1, parameter2)
```

Contoh untuk memanggil prosedur bersih di atas adalah :  
call bersih() atau bersih().

Untuk parameter dari prosedur/fungsi terdapat dua jenis yaitu ByVal dan Byref. Pada parameter ByVal, semua perubahan yang dibuat ke nilai parameter tersebut hanya efektif dalam prosedur/fungsi yang memanggil. Nilai asli sekarang dalam prosedur/fungsi yang dipanggil tidak terpengaruh. Sedangkan parameter yang di-passing by reference (ByRef), jika parameter di dalam prosedur/fungsi berubah, perubahan juga akan dirasakan di luar prosedur/ fungsi. Contoh penulisan prosedur menggunakan parameter ByVal dan ByRef yaitu :

```
[Public|Private|Friend|Protected] Sub ([ByVal|ByRef]  
Nama_Variabel As Tipe_Data)  
.....blok perintah...  
End Sub
```

Berikut contoh program prosedur dengan parameter ByVal yaitu :

```
Sub Tampil(ByVal nilai As Integer)  
    nilai = nilai + 1  
    MsgBox(nilai)  
End Sub
```

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles btnOk.Click  
    Dim X As Integer  
    X = 20  
    'Memanggil prosedur tampil  
    Call Tampil(X)  
    'Menampilkan bilangan X  
    MsgBox(X)  
End Sub
```

Ketika prosedur Tampil dipanggil, prosedur tampil membawa parameter nilai yang ditambahkan dengan angka 1. Lalu variabel X dipanggil yang digabungkan dengan prosedur Tampil maka menghasilkan isi nilai X ditambahkan 1 yaitu 20+1=21. Kemudian saat nilai X dipanggil lagi tanpa digabungkan dengan prosedur Tampil maka menghasilkan isi nilai saja yaitu 20. Hal ini membuktikan bahwa pada parameter yang di-passing ByVal, jika

parameter di dalam prosedur berubah maka perubahan tidak akan terasa di luar prosedur.

Berikut contoh program prosedur dengan parameter

ByRef yaitu :

```
Sub Tampil2(ByRef nilai As Integer)
    nilai = nilai + 1
    MsgBox(nilai)
End Sub
```

```
Private Sub btnOk2_Click(sender As Object, e As EventArgs) Handles btnOk2.Click
    Dim X As Integer
    X = 20
    'Memanggil prosedur tampil2
    Call Tampil2(X)
    'Memanggil prosedur bilangan X
    MsgBox(X)
End Sub
```

Ketika prosedur Tampil dipanggil, prosedur tampil membawa parameter nilai yang ditambahkan dengan angka 1. Lalu variabel X dipanggil yang digabungkan dengan prosedur Tampil maka menghasilkan isi nilai X ditambahkan 1 yaitu  $20+1=21$ . Kemudian saat nilai X dipanggil lagi tanpa digabungkan dengan prosedur Tampil maka menghasilkan isi nilai X ditambahkan 1 yaitu  $20+1=21$ . Hal ini membuktikan bahwa pada parameter yang di-passing ByRef, jika parameter di dalam prosedur berubah maka perubahan akan terasa di luar prosedur.

Pada prosedur, apabila ingin keluar secara paksa dari prosedur tanpa harus mengeksekusi sampai akhir prosedur atau End Sub, maka dapat menggunakan perintah **Exit Sub**. Contoh :

```
Private Sub btnOk3_Click(sender As Object, e As EventArgs) Handles btnOk3.Click
    For Y = 1 To 8
        If Y <> 4 Then
            MsgBox(Y)
        Else
            Exit Sub
        End If
    Next
End Sub
```

Pada contoh program di atas, program menampilkan pesan yang berisi angka 1 hingga 8 kecuali angka 4. Tetapi jika disertakan kode program `Exit Sub` maka program hanya menampilkan pesan yang berisi angka 1 hingga 3 saja dan program akan keluar.

## 6.2. Fungsi

Fungsi / function adalah suatu prosedur yang mempunyai nilai output (*return value*). Jenis prosedur ini lebih banyak digunakan pada rutin program untuk berhitung. Nilai output ini disimpan dalam suatu variabel atau pada properti dari objek. Bentuk penulisan sebuah fungsi adalah :

**[Private | Public | Friend | Protected] Function nama\_fungsi (Parameter) [As Tipe\_Data]**

**...blok perintah**

**End Function**

Sedangkan untuk mengakses atau memanggil sebuah fungsi caranya sebagai berikut :

**Call nama\_fungsi (parameter)**

**Atau**

**nama\_fungsi (parameter)**

**Atau**

**Nama\_variabel\_penampung = nama\_fungsi (parameter)**

Contoh program menggunakan fungsi yaitu :

```
Function kuadrat(angka As Integer) As Integer
    kuadrat = angka * angka
End Function
```

```
Private Sub btnFungsi_Click(sender As Object, e As EventArgs) Handles
    btnFungsi.Click
    Dim tamp As Integer
    'Memanggil fungsi
    tamp = kuadrat(10)
```

MsgBox(tamp)

End Sub

Pada program di atas, terdapat fungsi kuadrat bertipe integer. Untuk memanggil fungsi kuadrat, fungsi tersebut ditampung pada variabel tamp yang bertipe integer juga dan juga fungsi kuadrat diberikan nilai yang akan dikuadratkan yaitu nilai 10. Terdapat tiga perbedaan antara prosedur dan fungsi yang ditampilkan pada tabel 6.1.

Tabel 6.1. Tabel Perbedaan Prosedur dan Fungsi

Prosedur	Fungsi
Tidak memiliki tipe data	Memiliki data dengan tipe data tertentu seperti variabel
Nilai yang diberikan pada prosedur akan digunakan untuk prosedur itu sendiri	Nilai yang diberikan pada fungsi akan diolah dan dihasilkan nilai baru untuk dipakai oleh ekspresi di luar fungsi
Karena prosedur tidak memiliki nilai output, maka tidak ada parameter yang perlu ditulis	Suatu fungsi dipanggil dengan menuliskan pada sisi kanan operator penugasan (=). Sedangkan pada sisi kiri bisa dituliskan ekspresi yang sesuai dengan kode yang digunakan.

### 6.3. Fungsi-Fungsi Built-In dalam VB .Net

Fungsi built-in adalah fungsi-fungsi yang sudah terdefinisi dalam VB.Net. Fungsi ini bisa dipakai langsung tanpa harus membuatnya terlebih dahulu.

Fungsi adalah suatu program yang dapat menerima berbagai nilai dan memberikan umpan balik tertentu. Nilai yang dimasukkan di dalam suatu fungsi disebut parameter, yang berupa angka, string, tanggal, dsb.

Manfaat pemakaian fungsi dalam program adalah menghemat waktu dan tenaga. Suatu fungsi digunakan untuk melakukan beberapa hal yaitu :



- Memanipulasi teks/string yaitu melakukan kegiatan berbagai operasi string, misalnya menghitung panjang string, mengambil bagian string, dan sebagainya.
- Memanipulasi tanggal, jam, dan data yaitu melakukan operasi tanggal, jam, dan data.
- Perhitungan matematik yaitu melakukan proses perhitungan yang berkaitan dengan data nilai/angka, misalnya menghitung nilai sinus, absolut, dan sebagainya.

Cara kerja suatu fungsi dapat diibaratkan sebagai suatu rumus yang sudah disiapkan oleh visual basic. Pemakai dapat memasukkan nilai yang dihitung, lalu hasilnya akan segera didapat. Gambaran cara kerja fungsi dapat dilihat pada gambar 6.1.



Gambar 6.1. Alur Kerja Suatu Fungsi [Yuswanto:2006]

Pada gambar 6.1 terlihat bahwa variabel A berisi 2.9 yang selanjutnya variabel A dikenakan fungsi Int(). Hasil dari pengolahan fungsi tersebut adalah 2.

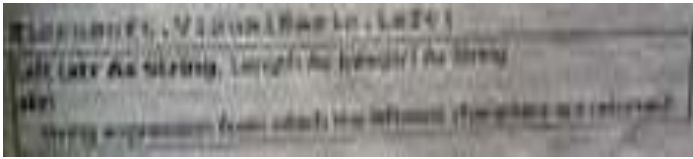
Bentuk penulisan dari fungsi adalah nama fungsi ditulis dengan tanda kurung yang dapat berisi parameter dari fungsi tersebut. Contoh :

➤ Int(3.14)

➤ Microsoft.VisualBasic.Left("komputer",3)

Beberapa fungsi pada visual basic telah dilengkapi dengan bantuan yang berupa tooltipstext, yang menunjukkan sintaks yang tepat dari fungsi tersebut. Umumnya tooltipstext

ini ditempatkan di bawah objek yang bersangkutan dalam suatu frame (kotak) dengan warna dasar kuning muda.



Gambar 6.2. Tooltiptext dari Fungsi Left [Yuswanto:2006]

### 6.3.1. Fungsi Kotak Dialog Pesan

Visual Basic .Net menyediakan kotak dialog yang berisi pesan teks dan menunggu pemakai untuk menutup kotak dialog tersebut dengan cara melakukan klik pada suatu tombol. Kotak dialog pesan ini hanya akan terlihat saat aplikasi yang menerapkannya berjalan.

Vb.Net menyediakan dua model kotak dialog pesan yaitu `MessageBox.Show` dan `MsgBox`. Class `MessageBox.Show` baru diperkenalkan pada vb.Net sedangkan `MsgBox` merupakan bawaan dari vb 6.0.

#### A. Membuat Kotak Dialog Pesan

Tujuan membuat kotak dialog pesan adalah menampilkan pesan pada kotak dialog pesan atau mengambil nilai dari kotak dialog pesan. Visual basic .Net menyediakan dua model kotak dialog pesan yaitu :

##### 1. `MessageBox.show()`

Kotak dialog pesan ini harus dijalankan dengan metode **show** dari class **System.Windows.Forms.MessageBox** yang disediakan dalam .Net Framework. Bentuk penulisan dari `MessageBox.show` sebagai berikut :

**`MessageBox.show(prompt [, judul][, tombol])`**

Keterangan :

- **Prompt**  
Merupakan teks yang akan ditampilkan sebagai pesan dalam kotak dialog pesan.  
Teks atau kalimat yang ingin ditampilkan pada kotak dialog pesan ditulis dalam tanda petik dua (“”). Panjang pesan/teks maksimum yang dapat ditampilkan adalah 1024 karakter dan masih tergantung pula pada jenis dan ukuran huruf yang digunakan.  
Jika prompt dibuat lebih dari satu baris, dapat ditambahkan dengan pemisah baris menggunakan karakter pindah baris (*carriage return*) yaitu **chr(13)**, karakter *linefeed* yaitu **chr(10)** atau perintah **vbCrLf**.
- **Judul**  
Pemakaian judul bersifat optional, berisikan teks yang ditampilkan pada baris judul kotak dialog pesan (*title*).  
Jika tidak diisi, visual basic .Net akan menggunakan nama proyek (*project*) sebagai judulnya.
- **Tombol**  
Pemakaian tombol bersifat optional, berisikan angka yang menunjukkan nilai dari tombol yang ditampilkan, model icon yang digunakan, juga untuk menentukan tombol default dan modalitas dari kotak pesan.

Contoh :

```
Private Sub btnPesan1_Click(sender As Object, e As EventArgs) Handles
btnPesan1.Click
    MessageBox.Show("Selamat Belajar VB.Net 2013", "Ucapan",
    MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub
```

Contoh program di atas menampilkan kotak dialog pesan dengan menggunakan model `MessageBox.show`. Kotak dialog pesan ini menampilkan isi pesan “Selamat Belajar

VB .Net 2013” dengan judul pesan “Ucapan”. Icon yang digunakan adalah information (tanda i) dan menggunakan tombol ok saja.

## 2. MsgBox()

Kotak dialog ini dapat digunakan sebagai input atau output dari aplikasi yang dibuat. Pesan ini biasanya berupa peringatan, pesan error, pesan untuk memastikan tindakan, dan lainnya. Sintaks dari fungsi MsgBox adalah :

**MsgBox(prompt[, tombol][, judul])**

Untuk keterangan sintaks dapat dilihat di penjelasan `MessageBox.show()`.

Terdapat perbedaan dalam memasang icon pada fungsi MsgBox yaitu setelah memasang tombol ikuti dengan tanda plus “+” dan ketikkan model icon yang akan ditampilkan.

Contoh :

```
Private Sub btnPesan2_Click(sender As Object, e As EventArgs) Handles
    btnPesan2.Click
        MsgBox("SEMANGAT" & vbCrLf & "SEMANGAT", MsgBoxStyle.OkCancel
            + MsgBoxStyle.Critical, "Berjuang")
    End Sub
```

Pada contoh program di atas, menampilkan pesan dengan menggunakan fungsi MsgBox. Isi pesan berupa “semangat” dan judul pesan berupa “Berjuang”. Untuk icon yang digunakan adalah critical (tanda X) sedangkan tombol yang digunakan adalah OK dan Cancel.

## B. Parameter

Pada parameter ini ditentukan jenis tombol dan icon yang akan ditampilkan di kotak dialog pesan. Visual basic .Net memberikan standar nilai tombol jika parameter tidak diisi





adalah tombol OK. Berikut macam-macam konstanta yang digunakan pada pengaturan tombol pada kotak dialog pesan.

Tabel 6.2. Tabel Konstanta Pada Tombol Pesan

Konstanta	Nilai	Keterangan
OKOnly	0	Tombol OK saja
OKCancel	1	Tombol OK dan Cancel
ABortRetryIgnore	2	Tombol Abort, Retry, dan Cancel
YesNoCancel	3	Tombol Yes, No, dan Cancel
YesNo	4	Tombol Yes dan No
RetryCancel	5	Tombol Retry dan Cancel

Selain tombol, terdapat jenis icon yang ditampilkan pada kotak dialog pesan. Fungsi icon adalah untuk membedakan jenis kotak dialog pesan yang ditampilkan. Berikut jenis icon yang digunakan pada kotak dialog pesan.

Tabel 6.3. Tabel Icon Pada Tombol Pesan

Konstanta	Nilai	Keterangan	Gambar
Critical	16	Menampilkan icon pesan kritis	
Question	32	Menampilkan icon pesan pertanyaan	
Exclamation	48	Menampilkan icon pesan peringatan	
Information	64	Menampilkan icon pesan informasi	

Default penekanan tombol pada kotak dialog pesan juga dapat ditentukan dengan memberikan nilai konstanta atau nilai numerik. Berikut default tombol pada kotak dialog pesan.

Tabel 6.3. Tabel Default Tombol Pada Kotak Dialog Pesan

Konstanta	Nilai	Keterangan
DefaultButton1	0	Tombol pertama sebagai default
DefaultButton2	256	Tombol kedua sebagai default

DefaultButton3	512	Tombol ketiga sebagai default
----------------	-----	-------------------------------

Khusus untuk class `MessageBox.show` terdapat fasilitas untuk penempatan teks (prompt dan judul), icon, dan tanda “X” (*close*). Berikut macam-macam perataan teks yang ditampilkan pada kotak dialog pesan.

Tabel 6.4. Tabel Perataan Teks Pada Kotak Dialog Pesan

Konstanta	Keterangan
<code>DefaultDesktopOnly</code>	Posisi teks dan icon rata kiri (default)
<code>RightAlign</code>	Posisi teks rata kanan pada kotak dialog pesan
<code>RtlReading</code>	Posisi teks dan icon ditukar pada kotak pesan
<code>ServiceNotification</code>	Posisi teks dan icon rata kiri

Visual basic menyediakan konstanta lain yang berhubungan dengan respon pemakai program terhadap kotak dialog pesan yaitu :

Tabel 6.5. Tabel Konstanta Lain Pada Kotak Dialog Pesan

Konstanta	Nilai	Keterangan
<code>ApplicationModul</code>	0	Pengguna program harus merespon kotakdialog pesan jika ingin melanjutkan pekerjaan dengan aplikasi
<code>SystemModal</code>	4096	Semua aplikasi yang sedang dijalankan pada sistem komputer akan dihentikan sampai pemakai program merespon kotak dialog pesan

Jenis parameter `SystemModal` biasanya digunakan agar kesalahan yang terjadi tidak merambat ke aplikasi lain yang sedang dijalankan.

### C. Mengembalikan Nilai

Sebuah kotak dialog pesan yang direspon akan mengembalikan sebuah nilai bertipe integer yang berasosiasi dengan tombol yang ditekan. Nilai inilah yang akan

menentukan tindakan apa selanjutnya yang akan dilakukan oleh aplikasi program. Berikut nilai-nilai yang akan dikembalikan oleh fungsi kotak dialog pesan. Nilai ini akan dimasukkan dalam variabel dimana fungsi kotak dialog pesan dibuat.

Tabel 6.6. Tabel Pengembalian Nilai Pada Kotak Dialog Pesan

Konstanta	Nilai	Keterangan
vbOk	1	Ok
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Metode-metode yang nilainya akan dikembalikan oleh class `MessageBox.show`. Untuk menambahkan metode tersebut, tambahkan kata kunci “**DialogResult**” yang diikuti oleh tanda titik.

Tabel 6.7. Tabel Pengembalian Metode Pada Kotak Dialog Pesan

Metode Selengkapnya	Tombol yang Dipilih
<code>DialogResult.Ok</code>	Ok
<code>DialogResult.Cancel</code>	Cancel
<code>DialogResult.Abort</code>	Abort
<code>DialogResult.Retry</code>	Retry
<code>DialogResult.Ignore</code>	Ignore
<code>DialogResult.Yes</code>	Yes
<code>DialogResult.No</code>	No

### 6.3.2. Fungsi Kotak Dialog Input

Kotak dialog input (*inputbox*) merupakan fungsi yang menghasilkan kotak yang meminta masukan tertentu dari pemakai program. `Inputbox` menampilkan kotak dialog yang berisi kotak teks yang bisa diisi oleh nilai tertentu (berupa string atau numerik) oleh pemakai program.

## A. Membuat Kotak Input

Dalam membuat inputbox harus disiapkan variabel yang dipakai untuk menampung nilai masukan. Bentuk penulisan dari inputbox seperti berikut ini :

**Nama\_variabel = InputBox (prompt [, judul][, default][, Xpos][, Ypos])**

Keterangan :

- Nama\_variabel yaitu untuk menyimpan hasil yang dikembalikan oleh fungsi inputbox.
- Prompt yaitu untuk teks atau kalimat yang ingin ditampilkan pada kotak pesan (ditulis dengan diapit tanda petik dua (“”)), untuk menerangkan data apa yang akan dimasukkan pada inputbox.
- Judul yaitu teks yang ditampilkan pada baris teratas kotak input (*title bar*) dan ditulis dengan diapit tanda petik dua (“”). Jika string ini tidak disertakan, judul pada kotak dialog pesan akan bertuliskan nama proyek aplikasi yang dibuat.
- Default adalah nilai standar yang akan ditulis pada kotak pengisian (jawaban yang sudah disediakan) dan ditulis di dalam tanda petik dua (“”)
- Xpos yaitu lokasi (nilai dengan tipe integer) pada sumbu X dimana kotak input akan muncul di layar. Jika tidak diisi maka inputbox akan diletakkan pada posisi default.
- Ypos yaitu lokasi (nilai dengan tipe integer) pada sumbu Y dimana kotak input akan muncul di layar. Jika tidak diisi maka inputbox akan diletakkan pada posisi default.

Contoh :

```
Private Sub btnInput1_Click(sender As Object, e As EventArgs) Handles
    btnInput1.Click
        Dim teks As String
```



```
teks = InputBox("Teks Prompt", "Judul", "Isi Default", 0, 450)
```

End Sub

Contoh program di atas menjelaskan tentang cara menampilkan inputbox dengan nama variabel “teks”, isi “teks prompt”, judul “judul”, isian “isi default”, lokasi X = 0 dan lokasi Y = 450.

## B. Menggunakan Kotak Input

Dalam menggunakan fungsi inputbox, pemakai mengetikkan suatu nilai pada kotak inputan dengan tipe data **string**. Saat pemakai melakukan klik pada tombol OK, nilai ini akan disimpan ke dalam variabel. Sedang jika tombol cancel di klik maka proses memasukkan data ke variabel dibatalkan. Nilai inilah yang nantinya akan dilakukan oleh aplikasi yang dibuat (contoh program ada di subbab latihan program).

### 6.3.3. Fungsi Identifikasi Tipe Data

Macam-macam fungsi untuk mengidentifikasi tipe data

:

Tabel 6.8. Tabel Fungsi Identifikasi Tipe Data

Nama Fungsi	Keterangan
IsArray (variabel)	Hasilnya true jika parameternya adalah array.
IsDate (ekspresi)	Hasilnya true jika ekspresi merupakan date yang benar.
IsDBNull (ekspresi)	Hasilnya true jika ekspresi adalah DBNull. Nilai DBNull adalah nilai nonvalid dan berbeda dengan nilai Nothing.
IsNothing (ekspresi)	Hasilnya adalah Boolean yang menunjukkan apakah ekspresi mewakili suatu variabel dari objek yang sekarang belum diberi nilai.
IsNumeric (ekspresi)	Hasilnya adalah true jika ekspresi adalah angka yang valid. Fungsi ini untuk mengecek validitas dari string yang berisi data numerik.

IsReference (ekspresi)	Hasilnya adalah Boolean yang menunjukkan apakah ekspresi mewakili variabel objek.
TypeName (nama_variabel)	Hasilnya adalah string yang menunjukkan tipe variabel

### 6.3.4. Fungsi String

Tipe data string/karakter merupakan tipe data yang paling sering digunakan karena tipe data ini dapat dipakai untuk data AlphaNumeric (alfabetic dan numeric). Jadi, dengan tipe data ini pemakai dapat memasukkan data abjad, angka, tanggal, dan sebagainya.

Secara umum dikatakan bahwa suatu data memiliki tipe data string jika data yang dimaksud dilingkup dengan tanda petik dua “”. Contoh data yang bertipe string yaitu “Halo”, “Halo2”, “123”, ”04/06/2015”, dan sebagainya. Berikut macam-macam fungsi string yaitu :

Tabel 6.9. Tabel Fungsi String

Nama Fungsi	Keterangan
Operator &	Digunakan untuk menggabungkan data, baik data string, numerik, maupun tanggal dan jam. Contoh : Label1.Text = "Semangat " & "Belajar"
Asc (karakter / string)	Digunakan untuk mengubah nilai string menjadi nilai angka kode karakter ASCII yang berhubungan dengan argumen string. Contoh : Label1.Text = "Nilai ASCII dari A : " & Asc("A")
Chr (angka / number)	Digunakan untuk mengubah nilai kode karakter ASCII menjadi nilai string. Contoh : Label1.Text = "Nilai String dari 97 : " & chr(97)
LCase (string)	Digunakan untuk mengubah suatu karakter / string menjadi huruf kecil. Contoh : Label1.Text = LCase("HALO")
UCase (string)	Digunakan untuk mengubah suatu karakter / string menjadi huruf besar. Contoh :

	Label1.Text = UCASE("hawai")
Val (string)	Digunakan untuk mengubah nilai suatu string menjadi numerik/number. Contoh : Label1.Text = Val("2006") + 4
Len (string)	Digunakan untuk menghitung jumlah atau banyaknya karakter pada suatu string. Contoh : Label1.Text = Len ("Selamat Belajar")
InStr ([PosisiStart,], String1, String2[, Pembanding])	Digunakan untuk mencari posisi suatu string pada suatu teks atau kalimat. Argumen PosisiStart digunakan untuk menentukan posisi awal pencarian. Nilai default pencarian adalah Case Sensitive (membedakan huruf besar dan huruf kecil). Contoh : Dim kata As String = "mang" Dim kal As String = "semangat belajar" Label1.Text = InStr(1, kal, kata)
Str (number)	Digunakan untuk mengubah nilai numerik menjadi string. Contoh : Dim thn As Integer = 2020 Dim kata As String = Str(thn) Label1.Text = "Tahun " & thn
StrReverse (string)	Digunakan untuk membalikkan urutan suatu karakter pada suatu kata atau kalimat. Contoh : Dim kata As String = "Sidoarjo" Label1.Text = StrReverse(kata)
Left (string, angka/N)	Digunakan untuk mendapatkan sejumlah karakter string dari sebelah kiri sebanyak N karakter. Contoh : Label1.Text = Microsoft.VisualBasic.Left("Selamat Belajar", 4)
Right (string, angka/N)	Digunakan untuk mendapatkan sejumlah karakter string dari sebelah kanan sebanyak N karakter. Contoh : Label1.Text = Microsoft.VisualBasic.Right("Selamat Belajar", 4)
Mid (string, start/P, panjang/N)	Digunakan untuk mendapatkan sejumlah karakter string mulai dari posisi P sebanyak N karakter. Contoh : Label1.Text = Mid("Selamat Belajar", 4,6)
Trim (string)	Digunakan untuk menghiangkan spasi kosong (blank) di sebelah kiri dan kanan data string. Contoh :

	Dim teks As String = " belajar " Label1.Text = Trim(teks)
LTrim (string)	Digunakan untuk menghiangkan spasi kosong (blank) di sebelah kiri data string. Contoh :
	Dim teks As String = " belajar " Label1.Text = LTrim(teks)
RTrim (string)	Digunakan untuk menghiangkan spasi kosong (blank) di sebelah kanan data string. Contoh :
	Dim teks As String = " belajar " Label1.Text = RTrim(teks)

### 6.3.5. Fungsi Matematika

Fungsi matemetika adalah suatu fungsi yang digunakan untuk melakukan perhitungan matematik tertentu pada data dengan tipe data numerik (angka). Cara mengaktifkan fungsi matematika adalah :

1. Mengimpor class Math yaitu dengan cara menetikkan perintah **Imports System.Math** pada general atau di luar class utama.
2. Menggunakan awalan Math pada setiap fungsi matematika yang akan digunakan. Misalnya Math.Sin, Math.Abs, dll.

Macam-macam fungsi matematika adalah :

Tabel 6.10. Tabel Fungsi Matematika

Nama Fungsi	Keterangan
Abs (number)	Untuk menghasilkan nilai absolut. Contoh :
	Label1.Text = Math.Abs (-28.9)
Atan (number)	Untuk menghasilkan nilai arc tangen (dalam radian). Contoh :
	Label1.Text = Math.Atan (3.14)
Tan (number)	Untuk menghasilkan nilai tangen pada suatu sudut. Contoh :
	Label1.Text = Math.Tan (30*(Math.Pi/180))
Cos (number)	Untuk menghasilkan nilai kosinus (dalam radian). Contoh :
	Label1.Text = Math.Cos (3.14)
Sin (number)	Untuk menghasilkan nilai sinus. Nilai number yang diberikan harus dalam bentuk radian bukan derajat.
	Untuk mengubah nilai dari derajat ke radian maka nilai derajat harus dikalikan dengan 3.14/180. Sebaliknya, jika

	ingin mengubah nilai radian ke derajat maka nilai radian harus dikalikan dengan $180/3.14$ . Contoh : Label1.Text = Math.Sin (90*(3.14/180))
Exp (number)	Untuk menghasilkan nilai basis logaritma atau bilangan eksponensial. Contoh : Label1.Text = Math.Exp (100)
Log (number)	Untuk menghasilkan bilangan logaritma natural basis e dari ekspresi angka. Contoh : Label1.Text = Math.Log (100)
Log10 (number)	Untuk menghasilkan bilangan logaritma natural basis 10 dari ekspresi angka. Contoh : Label1.Text = Math.Log10 (100)
Oct (number)	Untuk menghasilkan nilai octal dari ekspresi angka decimal. Contoh : Label1.Text = Oct (18)
Hex (number)	Untuk menghasilkan nilai hexadecimal dari ekspresi angka decimal. Contoh : Label1.Text = Hex (49)
Ceiling (number)	Untuk membualtkan nilai pecahan (desimal) menjadi pembulatan ke atas. Contoh : Label1.Text = Math.Ceiling (18.1)
Floor (number)	Untuk membualtkan nilai pecahan (desimal) menjadi pembulatan ke bawah. Contoh : Label1.Text = Math.Floor (18.1)
Round (ekspresi[, jumlahdesimal])	Untuk membulatkan angka sampai decimal tertentu. Contoh : Label1.Text = Math.Round (10.5, 3)
Int (number)	Untuk menghasilkan bilangan integer pada suatu number. Contoh : Label1.Text = Int (18.1)
Sqrt (number)	Untuk menentukan akar kuadrat. Contoh : Label1.Text = Math.Sqrt (10)
Mod (Number Mod Number)	Untuk menghasilkan sisa hasil bagi dari suatu bilangan yang dibagi dengan bilangan lainnya. Contoh : Label1.Text = (10 mod 3)

### 6.3.6. Fungsi Statistik

Fungsi statistik merupakan suatu fungsi yang digunakan untuk melakukan perhitungan data statistik. Berikut fungsi statistik yang disediakan oleh visual basic.

Tabel 6.11. Tabel Fungsi Statistik

Nama Fungsi	Keterangan
Avg (field)	Untuk menghitung nilai rata-rata pada perintah query. Contoh : <b>Select Gaji From Pegawai</b>
Count (field)	Untuk menghitung banyaknya data pada perintah query. Contoh : <b>Select Count (Nama) As Jumlah From Barang</b>
Max (field)	Untuk menghasilkan nilai terbesar pada perintah query. Contoh : <b>Select Max(Gaji) From Pegawai</b>
Min (field)	Untuk menghasilkan nilai terkecil pada perintah query. Contoh : <b>Select Min(Gaji) From Pegawai</b>
Sum (field)	Untuk digunakan untuk menjumlah suatu nilai ( <i>value</i> ) pada perintah query. Contoh : <b>Select Sum(Gaji) As Total From Pegawai</b>

### 6.3.7. Fungsi Tanggal dan Waktu

Tipe data jenis tanggal (*date*) dan jam (*time*) merupakan tipe data yang khusus diperuntukkan bagi operasi tanggal dan jam. Untuk pengoperasian data string agar bisa dibaca sebagai data tanggal maupun data jam/waktu, tambahkan tanda “#” pada awal dan akhir program tersebut. Contoh :

```
Dim tgl As Date = #8/20/2020#
Dim jam As Date = #1:22:50 PM#
```

Beriku beberapa karakter yang digunakan untuk argumen pengaturan tanggal dan waktu yaitu :

Tabel 6.12. Tabel Karakter Pada Tanggal dan Waktu

Karakter	Keterangan
/	Pemisah nilai tanggal
:	Pemisah nilai jam / waktu
d	Menampilkan hari dalam bentuk angka 1 s/d 31
dd	Menampilkan dalam bentuk angka yang didahului dengan angka 0 seperti 01 s/d 31
ddd	Menampilkan nama hari dengan singkatan
dddd	Menampilkan nama hari secara penuh
w	Menampilkan hari dalam minggu sebagai angka, seperti 1 untuk minggu dan 7 untuk sabtu
ww	Menampilkan minggu dalam satu tahun
M	Menampilkan bulan sebagai angka dalam satu tahun, seperti 1 s/d 12
MM	Menampilkan bulan sebagai angka dalam satu tahun yang di dahului dengan angka 0, seperti 01 s/d 12

MMM	Menampilkan nama bulan dengan singkatan, seperti Jan, Des
MMMM	Menampilkan nama bulan secara penuh, seperti Januari
y	Menampilkan hari dalam satu tahun sebagai angka, seperti 1 s/d 366
yy	Menampilkan angka sebagai angka 2 digit, seperti 00 s/d 99
yyyy	Menampilkan angka sebagai angka 4 digit, seperti 0000 s/d 9999
h	Menampilkan jam sebagai angka tanpa diawali 0, seperti 0 s/d 12
hh	Menampilkan jam sebagai angka tanpa diawali 0, seperti 00 s/d 12
H	Menampilkan jam sebagai angka tanpa diawali 0 dengan format 24 jam, seperti 0 s/d 24
HH	Menampilkan jam sebagai angka tanpa diawali 0 dengan format 24 jam, seperti 00 s/d 24
m	Menampilkan menit sebagai angka tanpa diawali 0, seperti 0 s/d 59
mm	Menampilkan menit sebagai angka yang diawali 0, seperti 0 s/d 59
<b>Karakter</b>	<b>Keterangan</b>
s	Menampilkan detik sebagai angka tanpa diawali 0, seperti 0 s/d 59
ss	Menampilkan detik sebagai angka yang diawali 0, seperti 0 s/d 59
AM/PM	Menggunakan format 12 jam dan menampilkan indikasi jam AM atau PM

Beberapa fungsi yang diperuntukkan khusus bagi data tipe tanggal dan jam, diantaranya adalah :

Tabel 6.13. Tabel Fungsi Tanggal dan Jam

<b>Nama Fungsi</b>	<b>Keterangan</b>
Now ()	Untuk menghasilkan tanggal dan waktusistem komputer saat ini. Contoh : Label1.Text = now()
Year (Date)	Untuk menghasilkan angka yang mewakili tahun dari tanggal yang ditentukan dalam argumen. Contoh : Label1.Text = Microsoft.VisualBasic.Year(#9/20/2020#)
Month (Date)	Untuk menghasilkan angka yang mewakili bulan dari tanggal yang ditentukan dalam argument. Contoh : Label1.Text = Microsoft.VisualBasic.Month(#9/20/2020#)
Weekday (Date)	Untuk menghasilkan angka hari ke dalam satu minggu. Hari minggu 0, senin 1, dst. Contoh : Label1.Text = Microsoft.VisualBasic.Weekday(#9/20/2020#)
Day ()	Untuk menghasilkan nilai angka hari dari tanggal yang ditentukan dalam argumen. Nilainya diantaranya 1 s/d 31. Contoh : Label1.Text = Microsoft.VisualBasic.Day(#9/20/2020#)

Hour (time)	Digunakan untuk menampilkan nilai jam dari jam sistem komputer dengan angka 0 sampai 24 yang diwakili oleh argumen time. Contoh : Label1.Text = Hour (#12:25:05 AM#)
Minute (time)	Digunakan untuk menampilkan nilai menit dari jam sistem komputer dengan angka 0 sampai 60 yang diwakili oleh argumen time. Contoh : Label1.Text = Second (#12:25:05 AM#)
Second (time)	Digunakan untuk menampilkan nilai detik dari jam sistem komputer dengan angka 0 sampai 60 yang diwakili oleh argumen time. Contoh : Label1.Text = Hour (#12:25:05 AM#)
DateAdd(Interval, N, Date)	Untuk menambah nilai tanggal dengan variabel interval. Interval adalah unit waktu (jam, hari, minggu). N adalah nilai dari interval yang ditambahkan ke date. Date adalah nilai dari tanggal awal. Contoh : Label1.Text = DateAdd(DateInterval.Day, 30, #9/20/2020#)
DateValue (String)	Untuk mengubah data teks menjadi data tanggal. Contoh : Label1.Text = DateValue("15/03/2020")
TimeValue (String)	Untuk mengubah data teks menjadi data jam/waktu. Contoh : Label1.Text = TimeValue("12 : 30 : 09 AM")
DateDiff (Interval, Date1, Date2 [,FirstDayOfWeek[, FirstWeekOfYear]])	Untuk selisih diantara dua tanggal sebesar nilai interval. Interval adalah format dimana bagian dari tanggal akan ditampilkan. Date1 adalah tanggal yang akan mengurangi Date2. Date2 adalah tanggal yang akan dikurangi date1. FirstDayOfWeek [optional] untuk menentukan hari pertama dalam satu minggu. FirstWeekOfYear [optional] untuk menentukan minggu pertama dalam satu tahun. Contoh : Label1.Text = DateDiff(DateInterval.DayOfYear, #9/20/2020#, #10/20/2020#)
DatePart (Interval, Date [,FirstDayOfWeek[, FirstWeekOfYear]])	Untuk mengambil bagian dari tanggal yang diuji. Interval adalah format dimana bagian dari tanggal akan ditampilkan.date adalah nilai tanggal yang akan diuji. FirstDayOfWeek [optional] untuk menentukan hari pertama dalam satu minggu. FirstWeekOfYear [optional] untuk menentukan minggu pertama dalam satu tahun. Contoh : Label1.Text = DatePart(DateInterval.Year, #9/20/2020#)

Sebagai tambahan, nilai interval yang dikenakan pada fungsi DateDiff dan DatePart adalah d, h, M, m s, w, ww, y, dan yyyy (lihat keterangan pada tabel 6.12).



### 6.3.8. Fungsi Format

Fungsi format digunakan untuk menampilkan suatu nilai dalam bentuk tampilan tertentu. Biasanya fungsi-fungsi ini digunakan untuk menampilkan angka, tanggal, dan jam. Berikut ini beberapa fungsi format yaitu :

Tabel 6.14. Tabel Fungsi Format

Nama Fungsi	Keterangan
# Bentuk : <b>format</b> ( <b>number</b> , " <b>#,#.00</b> ")	Untuk memformat tampilan angka dalam bentuk ribuan, jutaan, dan bentuk decimal. Contoh : Label1.Text = Format(1500000.25, " <b>#,#.00</b> ")
Currency Bentuk : <b>format</b> ( <b>number</b> , " <b>Currency</b> ")	Digunakan untuk menampilkan tanda mata uang di depan nilai serta dipisah dalam kelipatan ribuan. Untuk mengubah tanda mata uang lakukan klik pada tombol <b>Start, Settings, Control Panel</b> . Selanjutnya, lakukan double klik pada <b>Regional Settings</b> , Pilih Tab <b>Currency</b> , dan ganti mata uang pada kotak isian tombol. Contoh : Label1.Text = Format(1500000.25, " <b>Currency</b> ")
Fixed Bentuk : <b>format</b> ( <b>number</b> , " <b>Fixed</b> ")	Untuk menampilkan nilai dengan maksimal 2 desimal. Contoh : Label1.Text = Format(1500000.25, " <b>Fixed</b> ")
General Number Bentuk : <b>format</b> ( <b>number</b> , " <b>General Number</b> ")	Untuk menampilkan nilai dengan jumlah bilangan decimal yang bebas. Contoh : Label1.Text = Format(1500000.25, " <b>General Number</b> ")
Percent Bentuk : <b>format</b> ( <b>number</b> , " <b>Percent</b> ")	Untuk menampilkan nilai yang dikalikan 100 dan ditambahi tanda persen %. Contoh : Label1.Text = Format(12, " <b>Percent</b> ")
Scientific Bentuk : <b>format</b> ( <b>number</b> , " <b>Scientific</b> ")	Untuk menampilkan nilai dalam format ilmiah / pangkat. Contoh : Label1.Text = Format(1500000.25123, " <b>Scientific</b> ")
Short Date Bentuk : <b>format</b> ( <b>Date</b> , " <b>Short Date</b> ")	Untuk menampilkan tanggal dalam format pendek yaitu dd/mm/yyyy. Contoh : Label1.Text = Format(#3/17/2020#, " <b>Short Date</b> ")
Long Date Bentuk : <b>format</b> ( <b>Date</b> , " <b>Long Date</b> ")	Untuk menampilkan tanggal dalam format panjang yaitu dd/mmm/yyyy. Contoh : Label1.Text = Format(#3/17/2020#, " <b>Long Date</b> ")
ddd	Untuk menampilkan nama hari dari data tanggal. Contoh :

Bentuk : <b>format (Date, "ddd")</b>	Label1.Text = Format(#3/17/2020#, "ddd")
Short Time Bentuk : <b>format (Time, "Short Time")</b>	Untuk menampilkan waktu dalam format jam dan menit saja. Contoh : Label1.Text = Format(#10:15:05 PM#, "Short Time")
Long Time Bentuk : <b>format (Time, "Long Time")</b>	Untuk menampilkan waktu dalam format waktu dalam format 24 jam. Contoh : Label1.Text = Format(#10:15:05 PM#, "Long Time")
On/Off Bentuk : <b>format (Number, "On/Off")</b>	Untuk menampilkan nilai <b>Off</b> jika nilainya 0 dan <b>On</b> jika nilainya 1. Contoh : Label1.Text = Format (1, "On/Off")
True/False Bentuk : <b>format (Number, "True/False")</b>	Untuk menampilkan nilai <b>False</b> jika nilainya 0 dan <b>True</b> jika nilainya 1. Contoh : Label1.Text = Format (1, "True/False")
Yes/No Bentuk : <b>format (Number, "Yes/No")</b>	Untuk menampilkan nilai <b>No</b> jika nilainya 0 dan <b>Yes</b> jika nilainya 1. Contoh : Label1.Text = Format (1, "Yes/No")

### 6.3.9. Fungsi Manipulasi File dan Folder

Fungsi manipulasi file dan folder merupakan fungsi-fungsi yang berhubungan dengan operasi file dan folder. Beberapa bentuk operasi yang berkaitan dengan file antara lain mengcopy, mengubah, dan menghapus suatu file. Sedangkan fungsi yang berkaitan dengan manipulasi folder antara lain mengcopy, mengubah, dan menghapus suatu folder. Beberapa fungsi manipulasi file dan folder yaitu :

Tabel 6.15. Tabel Fungsi Manipulasi File dan Folder

Nama Fungsi	Keterangan
Filecopy (Source, Destination)	Untuk membuat duplikat file dari suatu tempat ke tempat lainnya. Contoh :  <code>Private Sub btnFileCopy_Click(sender As Object, e As EventArgs) Handles btnFileCopy.Click</code>

	<pre> Dim asal As String = "C:\Users\CLIENT\Downloads\Documents\02-Persyaratan s3 2020-2021b.pdf" Dim tujuan As String = "C:\Users\CLIENT\Downloads\Programs\02-Persyaratan s3 2020-2021b.pdf" FileCopy(asal, tujuan) MsgBox("OK") End Sub </pre>
Rename (OldPath, NewPath)	Untuk mengubah nama suatu file. Contoh :
	<pre> Private Sub btnRename_Click(sender As Object, e As EventArgs) Handles btnRename.Click Dim lama As String = "C:\Users\CLIENT\Downloads\Programs\02-Persyaratan s3 2020-2021b.pdf" Dim baru As String = "C:\Users\CLIENT\Downloads\Programs\Syarat1.pdf" Rename(lama, baru) MsgBox("OK") End Sub </pre>
Kill (PathName)	Untuk menghapus suatu file dari hard disk. Contoh :
	<pre> Private Sub btnKill_Click(sender As Object, e As EventArgs) Handles btnKill.Click Kill("C:\Users\CLIENT\Downloads\Programs\Syarat1.pdf") MsgBox("OK") End Sub </pre>
Mkdir (PathName)	Untuk membuat folder baru. Contoh :
	<pre> Private Sub btnMkdir_Click(sender As Object, e As EventArgs) Handles btnMkdir.Click Mkdir("C:\Users\CLIENT\Downloads\Programs\cb") MsgBox("OK") End Sub </pre>
ChDir (PathName)	Untuk mengubah folder yang telah ditentukan oleh argumen. Contoh :
	<pre> Private Sub btnChDir_Click(sender As Object, e As EventArgs) Handles btnChDir.Click ChDir("C:\Users\CLIENT\Downloads\Programs\cb") MsgBox("OK") End Sub </pre>
Rmdir (PathName)	Untuk menghapus folder yang telah ditentukan oleh argumen.

	Untuk menghapus folder, folder harus dalam keadaan kosong atau folder tidak boleh ada sub folder di dalamnya. Contoh :
	Private Sub BbtnRmDirutton1_Click(sender As Object, e As EventArgs) Handles BbtnRmDirutton1.Click Rmdir("C:\Users\CLIENT\Downloads\Programs\cb") MsgBox("OK, folder sudah terhapus") End Sub

### 6.3.10. Fungsi Finansial

Fungsi finansial/keuangan merupakan suatu fungsi yang digunakan untuk membuat perhitungan bisnis yang umum. Berikut macam-macam fungsi finansial yaitu :

Tabel 6.16. Tabel Fungsi Finansial

Nama Fungsi	Keterangan
Pmt() Bentuk : <b>pmt(SukuBunga, JumlahPeriode, NilaiSekarang, NilaiKemudian, Tipe)</b>	Untuk menghitung nilai angsuran secara periodik untuk melunasi nilai tunai pinjaman berdasarkan periode pembayaran yang konstan dan nilai bunga yang konstan pula. Contoh : Label5.Text = Pmt((16 / 100) / 12, 3 * 12, 6000000)
Suku bunga yaitu besar bunga bank setiap bulan. Jumlah Periode yaitu lama pinjaman dalam satuan bulan. Nilai sekarang yaitu besar nilai uang saat peminjaman. Nilai kemudian yaitu besar uang yang harus dikembalikan. Tipe yaitu kapan pembayaran dilakukan. Hasil perhitungan negatif karena itu merupakan angsuran yang harus dibayarkan. Gunakan <b>DueDate.End-ofPeriod</b> jika pembayaran di akhir periode. Gunakan <b>DueDate.Beg-ofPeriod</b> jika pembayaran di awal periode. Defaultnya di kahir periode.	
IPmt() Bentuk : <b>Ipmt(SukuBunga, Periode, JumlahPeriode, NilaiSekarang, NilaiKemudian, Tipe)</b>	Untuk menghitung nilai bunga dari pembayaran yang diperlukan untuk membayar kembali nilai pinjaman dalam jangka waktu yang ditentukan, dengan periode pembayaran dan suku bunga yang konstan. Contoh : Label5.Text = IPmt((16 / 100) / 12, 1, 3 * 12, 6000000)
PPmt () IPmt() Bentuk : <b>Ppmt(SukuBunga, Periode, JumlahPeriode,</b>	Untuk menghitung nilai pokok dari pembayaran yang dilakukan untuk membayar kembali suatu nilai pinjaman dalam jangka waktu yang ditentukan, dengan periode dan suku bunga yang konstan. Contoh : Label5.Text = PPmt((16 / 100) / 12, 1, 3 * 12, 6000000)

<b>NilaiSekarang, NilaiKemudian, Tipe)</b>	
NPer () Bentuknya :	<i>Number of Periode</i> untuk menghitung jumlah periode untuk mengembalikan suatu investasi berdasarkan jumlah pembayaran dan bunga yang tetap. Contoh :
<b>NPer(SukuBunga, Pembayaran, NilaiSekarang, NilaiKemudian, Tipe)</b>	Label5.Text = NPer((16 / 100) / 12, -210942.2, 6000000) Tanda minus mewakili jumlah uang yang harus dibayarkan.
Rate() Bentuknya :	Untuk menghitung laju pengembalian investasi yang menghasilkan serangkaian pembayaran yang sama setiap periode atau pembayaran sekaligus dalam jumlah besar. Nilai perkiraan berkisar 0 hingga 1. Contoh :
<b>Rate(JumlahPeriode, Pembayaran, NilaiSekarang, NilaiKemudian, Tipe, Terkaan)</b>	Label5.Text = Rate(5, 1500000, -5000000) Tanda minus mewakili jumlah uang yang harus dibayarkan.
DDB() Bentuknya :	Untuk menghitung nilai penyusutan dari suatu hak milik untuk suatu periode tertentu dengan menggunakan <i>Double Declining Balance Method</i> atau metode saldo menurun ganda. Metode ini menyebabkan nilai penyusutan tidak merata pada setiap tahunnya, sehingga penyusutan lebih besar pada awal pemakaian dibandingkan pada akhir pemakaian. Contoh :
<b>DDB(pokok, residu, umur, periode)</b>	Label5.Text = DDB(25000000, 2500000,5,3)
Pokok adalah biaya awal asset. Umur adalah jangka waktu pemakaian asset. Periode adalah rentang waktu masing-masing yang akan dihitung. Residu adalah nilai akhir asset setelah dipotong penurunan (penyusutan).	
SYD() Bentuknya :	Untuk menghitung nilai penyusutan dari suatu hak milik untuk suatu periode tertentu dengan menggunakan <i>Sum Of Years Digits</i> atau Metode Jumlah Angka Tahunan. Metode ini menyebabkan nilai penyusutan akan besar pada awal tahunnya dan menurun kemudian. Nilai terkecil setelah penyusutan adalah nilai residunya. Contoh :
<b>SYD(pokok, residu, umur, periode)</b>	Label5.Text = SYD(25000000, 2500000,5,3)
SLN () Bentuknya :	Untuk menghitung nilai penyusutan dari suatu hak milik untuk suatu periode tertentu dengan menggunakan <i>Straight Line Method</i> atau Metode Garis Lurus. Metode ini menyebabkan nilai penyusutan akan merata setiap tahunnya. Contoh :
<b>SLN(pokok, residu, umur)</b>	Label5.Text = SLN(25000000, 2500000,5)

## 6.4. Latihan Program

Latihan program ini membahas mengenai penjelasan yang ada di bab 6 ini. Dimulai dari pembuatan prosedur, fungsi, pesan, inputbox, hingga fungsi built-in yang ada di visual basic. Tampilan design program Prosedur dan FUNgsi dapat dilihat pada gambar 6.3.



Gambar 6.3. Design Form Prosedur dan Fungsi

Dari tampilan program tersebut, tool dan property yang digunakan adalah :

Tabel 6.17. Tabel Tool Form Prosedur dan Fungsi

Tool	Properties	Keterangan
Label 1	Text	Fungsi String
Label 2	Text	Fungsi Matematika
Label 3	Text	Fungsi Tanggal dan Waktu
Label 4	Text	Fungsi Format
Label 5	Text	Fungsi Finansial
Button 1	Name	Btnok
	Text	Ok Byval
Button 2	Name	Btnok2
	Text	Ok ByRef
Button 3	Name	btnOk3
	Text	Ok Exit Sub

Button 4	Name	btnFungsi
	Text	Fungsi
Button 5	Name	btnPesan1
	Text	Pesan MessageBox
Button 6	Name	btnPesan2
	Text	Pesan MsgBox
Button 7	Name	btnInput1
	Text	Tampil InputBox
Button 8	Name	btnInput2
	Text	Praktek InputBox
Button 9	Name	btnTampil
	Text	Contoh Beberapa Fungsi
Button 10	Name	btnFileCopy
	Text	Contoh FileCopy
Button 11	Name	btnRename
	Text	Contoh Rename
Button 12	Name	btnKill
	Text	Contoh Kill
Button 13	Name	btnMkDir
	Text	Contoh MkDir
Button 14	Name	BtnChDir
	Text	Contoh ChDir
Button 15	Name	btnRmDir
	Text	Contoh RmDir
Button 16	Name	btnFiannsial
	Text	Contoh Finansial

Lalu tambahkan kode program pada :

- Buatlah beberapa sub prosedur dan fungsi yaitu prosedur tampil, prosedur tampil2, dan fungsi kuadrat.

```
Sub Tampil(ByVal nilai As Integer)
```

```
    nilai = nilai + 1
```

```
    MsgBox(nilai)
```

```
End Sub
```

```
Sub Tampil2(ByRef nilai As Integer)
```

```
    nilai = nilai + 1
```

```
    MsgBox(nilai)
```

```
End Sub
```

```
Function kuadrat(angka As Integer) As Integer
```

```
    kuadrat = angka * angka
```

### End Function

- Ketikkan kode program berikut pada tombol Ok ByVal. Kode program untuk menampilkan bilangan X dengan menggunakan prosedur ByVal.

```
Private Sub btnOk_Click(sender As Object, e As EventArgs) Handles
btnOk.Click
    Dim X As Integer
    X = 20
    'Memanggil p rosedur tampil
    Call Tampil(X)
    'Menampilkan bilangan X
    MsgBox(X)
End Sub
```

- Ketikkan kode program berikut pada tombol Ok ByRef. Kode program untuk menampilkan bilangan X dengan menggunakan prosedur ByRef.

```
Private Sub btnOk2_Click(sender As Object, e As EventArgs) Handles
btnOk2.Click
    Dim X As Integer
    X = 20
    'Memanggil prosedur tampil2
    Call Tampil2(X)
    'Memanggil prosedur bilangan X
    MsgBox(X)
End Sub
```

- Ketikkan kode program berikut pada tombol Ok Exit Sub. Kode program ini untuk menampilkan angka 1 hingga 8 yang ditampilkan pada kotak dialog pesan dan menghentikan program secara paksa menggunakan Exit Sub.

```
Private Sub btnOk3_Click(sender As Object, e As EventArgs) Handles
btnOk3.Click
    For Y = 1 To 8
        If Y <> 4 Then
            MsgBox(Y)
        End If
    Next Y
End Sub
```



```

Else
    'memaksa program berhenti
Exit Sub
End If
Next
End Sub

```

- Ketikkan kode program berikut pada tombol Fungsi. Kode program ini untuk menampilkan angka yang dikuadratkan dengan menggunakan fungsi.

```

Private Sub btnFungsi_Click(sender As Object, e As EventArgs) Handles
btnFungsi.Click
    Dim tamp As Integer
    'Memanggil fungsi
    tamp = kuadrat(10)
    MsgBox(tamp)
End Sub

```

- Ketikkan kode program berikut pada tombol Pesan MessageBox. Kode program ini untuk menampilkan suatu pesan pada kotak dialog pesan dengan menggunakan class MessageBox.

```

Private Sub btnPesan1_Click(sender As Object, e As EventArgs) Handles
btnPesan1.Click
    'Menampilkan Pesan MessageBox
    MessageBox.Show("Selamat Belajar VB.Net 2013", "Ucapan",
    MessageBoxButtons.OK, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button1,
    MessageBoxOptions.RightAlign)
End Sub

```

- Ketikkan kode program berikut pada tombol Pesan MessageBox. Kode program ini untuk menampilkan suatu pesan pada kotak dialog pesan dengan menggunakan class

```

MsgBox.
Private Sub btnPesan2_Click(sender As Object, e As EventArgs) Handles
btnPesan2.Click
    'penggunaan kotak pesan dengan MsgBox

```

```
MsgBox("SEMANGAT" & vbCrLf & "SEMANGAT", MsgBoxStyle.OkCancel +
MsgBoxStyle.Exclamation, "Berjuang")
```

```
End Sub
```

- Ketikkan kode program berikut pada tombol Tampil InputBox. Kode program ini hanya untuk menampilkan InputBox tanpa ada langkah selanjutnya.

```
Private Sub btnInput1_Click(sender As Object, e As EventArgs) Handles
btnInput1.Click
    Dim teks As String
    'penggunaan inputbox
    teks = InputBox("Teks Prompt", "Judul", "Isi Default", 0, 450)
End Sub
```

- Ketikkan kode program berikut pada tombol Contoh ChDir. Kode program ini digunakan untuk mengubah posisi pemakai dari suatu folder ke folder lain.

```
Private Sub btnChDir_Click(sender As Object, e As EventArgs) Handles
btnChDir.Click
    'Folder telah diubah, sesuaikan pada folder komputer Anda
    Dir("C:\Users\CLIENT\Downloads\Programs\cb")
    MsgBox("OK")
End Sub
```

- Ketikkan kode program berikut pada tombol Praktek InputBox. Kode program ini digunakan menampilkan bilangan ganjil, dengan penentuan bilangan awal dan akhir menggunakan inputbox.

```
Private Sub btnInput2_Click(sender As Object, e As EventArgs)
Handles btnInput2.Click
    Dim awal, akhir, hasil As String
    'penggunaan inputbox
    awal = InputBox("Batas Awal Nilai ", "Batas Awal", "1")
    akhir = InputBox("Batas Akhir Nilai ", "Batas Akhir")
    hasil = ""
    If awal <= akhir Then
        For A As Integer = awal To akhir
            If (A Mod 2) = 1 Then
                hasil &= A & " "
            End If
        Next
    End If
End Sub
```

```

        MsgBox(hasil)
    Else
        MessageBox.Show("Nilai akhir tidak boleh lebih kecil dari
        nilai awal", "PERHATIAN", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End If
End Sub

```

- Ketikkan kode program berikut pada tombol Contoh FileCopy. Kode program ini digunakan untuk mengcopy

file.

```

Private Sub btnFileCopy_Click(sender As Object, e As EventArgs) Handles
btnFileCopy.Click
    'Mengcopy file, sesuaikan pada folder komputer Anda
    Dim asal As String = "C:\Users\CLIENT\Downloads\Documents\02-
    Persyaratan s3 2020-2021b.pdf"
    Dim tujuan As String = "C:\Users\CLIENT\Downloads\Programs\02-
    Persyaratan s3 2020-2021b.pdf"
    FileCopy(asal, tujuan)
    MsgBox("OK, file telah dicopy")
End Sub

```

- Ketikkan kode program berikut pada tombol Contoh Kill. Kode program ini digunakan untuk menghapus file.

```

Private Sub btnKill_Click(sender As Object, e As EventArgs) Handles
btnKill.Click 'sesuaikan pada folder komputer Anda
    Kill("C:\Users\CLIENT\Downloads\Programs\Syarat1.pdf")
    MsgBox("OK, file sudah terhapus")
End Sub

```

- Ketikkan kode program berikut pada tombol Contoh Rename. Kode program ini digunakan untuk mengubah

nama file.

```

Private Sub btnRename_Click(sender As Object, e As EventArgs) Handles
btnRename.Click
    'Mengubah nama file,sesuaikan pada folder komputer Anda
    Dim lama As String = "C:\Users\CLIENT\Downloads\Programs\02-
    Persyaratan s3 2020-2021b.pdf"
    Dim baru As String =
    "C:\Users\CLIENT\Downloads\Programs\Syarat1.pdf"
    Rename(lama, baru)
    MsgBox("OK, file sudah diubah namanya")
End Sub

```

- Ketikkan kode program berikut pada tombol Contoh Mkdir. Kode program ini digunakan untuk membuat folder baru.

```
Private Sub btnMkdir_Click(sender As Object, e As EventArgs) Handles
btnMkdir.Click
    'Membuat folder, sesuaikan pada folder komputer Anda
    Mkdir("C:\Users\CLIENT\Downloads\Programs\cb")
    MsgBox("OK, folder telah dibuat")
End Sub
```

- Ketikkan kode program berikut pada tombol Contoh Rmdir. Kode program ini digunakan untuk menghapus folder.

```
Private Sub BbtnRmdirutton1_Click(sender As Object, e As EventArgs)
Handles BbtnRmdirutton1.Click
    'Menghapus folder, sesuaikan pada folder komputer Anda
    Rmdir("C:\Users\CLIENT\Downloads\Programs\cb")
    MsgBox("OK, folder sudah terhapus")
End Sub
```

- Ketikkan kode program berikut pada tombol Contoh Fungsi Finansial. Kode program ini digunakan untuk mempraktekkan fungsi finansial yang ada visual basic.

```
Private Sub btnFinansial_Click(sender As Object, e As EventArgs) Handles
btnFinansial.Click
    'Untuk soal kasus Fungsi Finansial ini lihat di Buku Yuswanto, 2006
    Label5.Text = "Hasil Program Fungsi Finansial : " & vbCrLf & "Fungsi Pmt :
" & Pmt((16 / 100) / 12, 3 * 12, 6000000) & vbCrLf & _
    "Fungsi IPmt : " & IPmt((16 / 100) / 12, 1, 3 * 12, 6000000) & vbCrLf & _
    "Fungsi PPmt : " & PPmt((16 / 100) / 12, 1, 3 * 12, 6000000) & vbCrLf & _
    "Fungsi Nper : " & NPer((16 / 100) / 12, -210942.2, 6000000) & vbCrLf & _
    "Fungsi Rate : " & Rate(5, 1500000, -5000000) & vbCrLf & _
    "Fungsi DDB : " & DDB(25000000, 2500000, 5, 3) & vbCrLf & _
    "Fungsi SYD : " & SYD(25000000, 2500000, 5, 3) & vbCrLf & _
    "Fungsi SLN : " & SLN(25000000, 2500000, 5)
End Sub
```

- Ketikkan kode program berikut pada tombol Contoh Beberapa Fungsi. Kode program ini digunakan untuk mempraktekkan fungsi-fungsi yang ada di visual basic yaitu fungsi string, matematika, tanggal dan waktu, serta format.

```
Private Sub btnTampil_Click(sender As Object, e As EventArgs) Handles
btnTampil.Click
```

```

Dim kata As String = "mang"
Dim kal As String = "semangat belajar"
Dim kata1 As String = "Sidoarjo"
Dim thn As Integer = 2020
Dim kata2 As String = Str(thn)
Dim teks As String = " belajar "
'Keterangan kode program lihat di modul
Label1.Text = "Hasil Program Fungsi String : " & vbCrLf & "Operator && : "
& "Semangat " & "Belajar" & vbCrLf & _

```

```

"Ex Fungsi Asc : " & "Nilai ASCII dari A = " & Asc("A") & vbCrLf & "Ex Fungsi
Chr : " & "Nilai String dari 97 : " & _ Chr(97) & vbCrLf & "Ex Fungsi LCase : "
& LCase("HALO") & _ vbCrLf & "Ex Fungsi UCase : " & UCase("hawai") &
vbCrLf & _
"Ex Fungsi Val : " & Val("2006") + 4 & vbCrLf & _
"Ex Fungsi Len : " & Len("Selamat Belajar") & vbCrLf & _
"Ex Fungsi Instr : " & Instr(1, kal, kata) & vbCrLf & _
"Ex Fungsi Str : " & "Tahun " & thn & vbCrLf & _
"Ex Fungsi StrReverse : " & StrReverse(kata1) & vbCrLf & _
"Ex Fungsi Left : " & Microsoft.VisualBasic.Left("Selamat Belajar", 4) & vbCrLf
& "Ex Fungsi Right : " & _ Microsoft.VisualBasic.Right("Selamat Belajar", 4)
& vbCrLf & _
"Ex Fungsi Mid : " & Mid("Selamat Belajar", 4, 6) & vbCrLf & _
"Ex Fungsi Trim : " & Trim(teks) & vbCrLf & _
"Ex Fungsi LTrim : " & LTrim(teks) & vbCrLf & _
"Ex Fungsi RTrim : " & RTrim(teks)

```

```

Label2.Text = "Hasil Program Fungsi Matematika : " & vbCrLf & "Ex Fungsi Abs
: " & Math.Abs(-28.9) & vbCrLf & _
"Ex Fungsi Atan : " & Math.Atan(3.14) & vbCrLf & _
"Ex Fungsi Tan : " & Math.Tan(30 * (Math.Pi / 180)) & vbCrLf & _ & "Ex Fungsi
Cos : " & Math.Cos(3.14) & vbCrLf & _
"Ex Fungsi Sin : " & Math.Sin(90 * (3.14 / 180)) & vbCrLf & _
"Ex Fungsi exp : " & Math.Exp(100) & vbCrLf & _
"Ex Fungsi Log : " & Math.Log(100) & vbCrLf & _
"Ex Fungsi Log10 : " & Math.Log10(100) & vbCrLf & _
"Ex Fungsi Oct : " & Oct(18) & vbCrLf & _
"Ex Fungsi Hex : " & Hex(49) & vbCrLf & _
"Ex Fungsi Ceiling : " & Math.Ceiling(18.1) & vbCrLf & _
"Ex Fungsi Floor : " & Math.Floor(18.1) & vbCrLf & _
"Ex Fungsi Round : " & Math.Round(10.5, 3) & vbCrLf & _
"Ex Fungsi Int : " & Int(18.1) & vbCrLf & _
"Ex Fungsi Sqrt : " & Math.Sqrt(10) & vbCrLf & _
"Ex Fungsi Mod : " & (10 Mod 3)

```

```

Label3.Text = "Hasil Program Fungsi Tanggal dan Waktu : " & vbCrLf & "Ex
Fungsi Now : " & Now() & vbCrLf & _
"Ex Fungsi Year : " & Microsoft.VisualBasic.Year(#9/20/2020#) & vbCrLf & "Ex
Fungsi Month : " & _ Microsoft.VisualBasic.Month(#9/20/2020#) & vbCrLf &
_
"Ex Fungsi Weekday : " & _ Microsoft.VisualBasic.Weekday(#9/20/2020#) &
vbCrLf & _
"Ex Fungsi Day : " & Microsoft.VisualBasic.Day(#9/20/2020#) & vbCrLf & "Ex
Fungsi Hour : " & Hour(#12:25:05 AM#) & vbCrLf & _
& "Ex Fungsi Minute : " & Minute(#12:25:05 AM#) & vbCrLf & _
"Ex Fungsi Second : " & Second(#12:25:05 AM#) & vbCrLf & _
"Ex Fungsi DateAdd : " & DateAdd(DateInterval.Day, 30, #9/20/2020#) &
vbCrLf & "Ex Fungsi Date Value : " & _ DateValue("15/03/2020") & vbCrLf &
"Ex Fungsi Time Value:" & _ TimeValue("12 : 30 : 09 AM") & vbCrLf & "Ex
Fungsi DateDiff : " & DateDiff(DateInterval.DayOfYear, #9/20/2020#,
#10/20/2020#) & vbCrLf & "Ex DatePart : " & DatePart(DateInterval.Year,
#9/20/2020#)

```

```

Label4.Text = "Hasil Program Fungsi Format : " & vbCrLf & "Ex Format # : " &
Format(1500000.25, "#,#.00") & vbCrLf & _
"Ex Fungsi Currency : " & Format(1500000.25, "Currency") & vbCrLf & "Ex
Fungsi Fixed : " & Format(1500000.25, "Fixed") & vbCrLf & "Ex Fungsi
General Number : " & Format(1500000.25, "General Number") & vbCrLf &
"Ex Fungsi Percent : " & Format(12, "Percent") & vbCrLf & "Ex Fungsi
Scientific : " & Format(1500000.25123, "Scientific") & vbCrLf & "Ex Fungsi
Short Date : " & Format(#3/17/2020#, "Short Date") & vbCrLf &
"Ex Fungsi Long Date : " & Format(#3/17/2020#, "Long Date") & vbCrLf &
"Ex Fungsi ddd : " & Format(#3/17/2020#, "ddd") & vbCrLf & "Ex Fungsi
Short Time : " & Format(#10:15:05 PM#, "Short Time") & vbCrLf & "Ex
Fungsi Long Time : " & Format(#10:15:05 PM#, "Long Time") & vbCrLf & _
"Ex Fungsi On/Off dari 1 : " & Format(1, "On/Off") & vbCrLf &
"Ex Fungsi True/False dari 1 : " & Format(1, "True/False") & vbCrLf & "Ex
Fungsi Yes/No dari 1 : " & Format(1, "Yes/No") & vbCrLf

```

End Sub

## BAB VII PEMROGRAMAN DATABASE

Program yang dibuat pada bab 7 ini menggunakan database Microsoft Access 2013 yang mempunyai ekstensi .accdb.

### 7.1. Pengertian, Jenis, dan Manfaat Database

Menurut Jogiyanto pengertian **database** adalah kumpulan informasi dan data yang saling berhubungan satu sama lain, dimana data tersebut tersimpan di simpanan luar komputer dan diperlukan software tertentu untuk memanipulasinya.

**Basis data atau database** adalah sekumpulan data yang terorganisir dalam bentuk skema, tabel, *query*, laporan, *view*, dan objek lainnya.

**Sistem manajemen basis data (DBMS)** adalah perangkat lunak komputer yang berinteraksi dengan Pengguna, aplikasi lain, dan basis data untuk mengambil dan menganalisis data. DBMS memungkinkan definisi, pembuatan, *query*, update, dan administrasi basis data.

Berikut ini adalah beberapa **fungsi database**:

- Mengelompokkan data dan informasi sehingga lebih mudah dimengerti
- Mencegah terjadinya duplikat data maupun inkonsistensi data

- Mempermudah proses penyimpanan, akses, pembaharuan, dan menghapus data.
- Menjaga kualitas data dan informasi yang diakses sesuai dengan yang di-input.
- Membantu proses penyimpanan data yang besar
- Membantu meningkatkan kinerja aplikasi yang membutuhkan penyimpanan data

### **Manfaat database :**

1. Tidak Terjadi Redudansi Basis Data → Seperti yang sudah disinggung pada pengertian database sebelumnya, database bisa membantu meminimalkan redudansi data. Redudansi sendiri merupakan terjadinya data-data ganda dalam berkas-berkas yang berbeda.
2. Integritas Data Terjaga → Database memastikan integritas data yang tinggi dimana database akan memastikan keakuratan, aksesibilitas, konsistensi dan juga kualitas tinggi pada suatu data.
3. Independensi Data Terjaga → Database menjaga independensi data dimana orang lain tidak dapat merubah data meskipun data bisa diakses.
4. Kemudahan Berbagi Data → Menggunakan perangkat lunak database bisa digunakan untuk berbagi data atau informasi dengan sesama pengguna lainnya.
5. Menjaga Keamanan Data → Database menjamin keamanan suatu informasi dan data, dimana Anda bisa menyisipkan kode akses untuk data-data tertentu yang tidak bisa diakses bersama.



6. Kemudahan Akses Data → Dengan database bisa memudahkan untuk mengakses dan mendapatkan data karena semua data terorganisir dengan baik.

Beberapa **jenis database** yang digunakan adalah :

1. **Microsoft Access**

Salah satu software database ini adalah yang paling sering digunakan. Microsoft access sangat cocok digunakan untuk sebagian besar komputer yang relasional. Selain itu, jika Anda berbisnis dalam skala rumahan, bisa memilih sistem database ini karena sangat ringan digunakan dan format datanya sangat umum sehingga memudahkan pembacaan.

2. **Oracle**

Salah satu software database ini sangat mampu untuk menyimpan data dengan ukuran yang maksimum hingga terabyte. Oracle paling banyak digunakan pada perusahaan-perusahaan terutama yang sedang berkembang karena memang untuk mengaksesnya tersedia secara gratis.

3. **Microsoft SQL Server**

Software database ini merupakan manajemen basis data yang umum digunakan pada Microsoft dengan bahasa pemrograman yang digunakan adalah Transact-SQL. Tipe data yang digunakan cukup banyak sehingga sangat efektif untuk mendukung kinerja Anda.

4. **MySQL**

Salah satu software database yang open access untuk umum dan kompatibel pada sistem operasi Windows maupun Linux. Keunggulan yang bisa Anda gunakan dengan menggunakan program MySQL adalah bisa digunakan untuk multi user. Kelebihan lainnya dari MySQL yaitu tersedia gratis, query data yang cepat dan berlisensi resmi.

## 5. Firebird

Bisa dibilang software database ini memiliki fitur sistem yang standar dan ringan yaitu fitur ANS SQL-99 dan SQL – 2003. Kompatibel untuk digunakan pada sistem operasi Windows, Linux maupun Unix.

## 6. Postgre SQL

Menawarkan sistem database opensource dengan lisensi GPL/ General Public License. Software ini menggunakan bahasa pemrograman C++, C, SQL, PHP dan lainnya. Jika digunakan untuk pekerjaan pribadi, maka software ini sangat recommended digunakan.

## 7.2. Perintah SQL

**SQL (Structured Query Language)** adalah bahasa yang di gunakan dalam pemrograman dan di rancang untuk mengelola data yang tersimpan dalam Sistem Manajemen Database Relasional (*Relational Database Management System*, RDBMS), atau untuk *stream processing* dalam Relational Data Stream Management System (RDSMS).

Salah satu dari fungsi database adalah menyimpan dan memelihara serta mengelola data, oleh karena itu penting bagi kita untuk mengetahui perintah dasar SQL yang dapat di gunakan untuk menyimpan dan memelihara serta mengelola data dalam database tersebut baik secara langsung maupun menggunakan macam-macam bahasa pemrograman.

Beberapa perintah dasar SQL yaitu :

### 1. Perintah SELECT

Perintah SELECT merupakan perintah dasar SQL yang di gunakan untuk memilih data dari database. Data yang di kembalikan di simpan dalam tabel yang di sebut result-set.

Sintaks **SELECT** :

**SELECT kolom1, kolom2, ... FROM nama\_tabel;**

Perintah pertama di atas di gunakan untuk menyeleksi kolom NIS, NAMA, dan ALAMAT dari tabel SISWA.

atau

**SELECT \* FROM nama\_tabel;**

sedangkan perintah kedua di gunakan untuk menyeleksi semua kolom dari tabel SISWA.

## 2. Perintah SELECT DISTINCT

Perintah SELECT DISTINCT merupakan perintah dasar SQL yang di gunakan untuk mengembalikan hanya nilai yang berbeda dari dalam sebuah tabel, dengan kata lain semua record duplikat (record dengan nilai yang sama) yang terdapat pada tabel akan di anggap sebagai satu record/nilai.

Sintaks SELECT DISTINCT :

**SELECT DISTINCT kolom1, kolom2, ... FROM nama\_tabel;**

Perintah di atas di gunakan untuk menampilkan kolom ALAMAT dari tabel SISWA dengan mengabaikan nilai yang duplikat, misalnya terdapat 10 siswa dengan alamat 'Jakarta', 15 siswa dengan alamat 'Bandung', 20 siswa dengan alamat 'Depok', dan seterusnya, result-set hanya menampilkan daftar alamat di antaranya 'Jakarta', 'Bandung', dan 'Depok' masing-masing 1 record.

## 3. Perintah WHERE

Perintah WHERE merupakan perintah dasar SQL yang di gunakan untuk mem-filter hasil SELECT dengan mengekstrak record yang memenuhi persyaratan tertentu.

Sintaks :

```
SELECT kolom1, kolom2, ... FROM nama_tabel  
WHERE kondisi;
```

Perintah di atas di gunakan untuk mengekstraksi data (NIS dan NAMA) dari tabel SISWA dengan kondisi “field ALAMAT berisi nilai JAKARTA”. Perintah di atas menggunakan operator sama dengan (=), untuk operator lain yang di dukung perintah WHERE, lihat tabel di bawah ini.

Tabel 7.1. Tabel Operator pada Perintah SQL Where

Operator	Deskripsi
=	Sama dengan
<>	Tidak sama dengan, pada beberapa versi SQL, operator yang di gunakan adalah !=
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
BETWEEN	Antara rentang inklusif
LIKE	Cari pola yang ‘seperti’ parameter
IN	Menentukan kemungkinan nilai dari beberapa kolom

#### 4. Perintah (operator) AND, OR dan NOT

Operator AND, OR dan NOT merupakan perintah dasar SQL yang biasanya di kombinasikan dengan perintah WHERE. Ketiganya di gunakan untuk mem-filter record berdasarkan suatu kondisi, operator AND akan menampilkan record apabila semua kondisi bernilai TRUE, operator OR akan menampilkan record apabila salah satu kondisi bernilai TRUE, sedangkan operator NOT akan menampilkan record apabila semua kondisi bernilai FALSE.

Sintaks AND :

```
SELECT kolom1, kolom2, ... FROM nama_tabel  
WHERE kondisi1 AND kondisi2 AND kondisi3 ...;
```

Perintah di atas akan menampilkan record NIS dan NAMA dari tabel SISWA dengan ALAMAT di JAKARTA dan TAHUN\_LAHIR “2000”, artinya record siswa yang lahir di tahun “2000” namun tidak beralamat di “Jakarta: atau siswa yang beralamat di “Jakarta” namaun lahir bukan pada tahun “2000” tidak akan di tampilkan.

Sintaks OR :

```
SELECT kolom1, kolom2, ... FROM nama_tabel  
WHERE kondisi1 OR kondisi2 OR kondisi3 ...;
```

Perintah di atas akan menampilkan record NIS dan NAMA dari tabel SISWA dengan ALAMAT di “JAKARTA” atau di “Bandung”, artinya record siswa yang beralamat di “Jakarta” dan di “Bandung” saja yang akan di tampilkan.

Sintaks NOT :

```
SELECT kolom1, kolom2, ... FROM nama_tabel WHERE  
NOT kondisi;
```

Perintah di atas akan menampilkan semua record NIS dan NAMA dari tabel SISWA kecuali record siswa yang beralamat di ‘Jakarta’.

## 5. Perintah ORDER BY

Perintah ORDER BY merupakan perintah dasar SQL yang di gunakan untuk mengurutkan result-set dalam pengurutan ‘ascending’ atau ‘descending’. Secara default perintah ORDER BY menampilkan record dalam pengurutan ‘ascending’ (‘ASC’). Untuk mengurutkan ‘descending’, gunakan kata kunci ‘DESC’.

```
SELECT kolom1, kolom2, ... FROM nama_tabel  
ORDER BY column DESC;
```

Perintah di atas akan menampilkan result-set berupa field NIS dan NAMA dari tabel SISWA dengan di urutkan berdasarkan TAHUN\_LAHIR secara descending, artinya tahun lahir akan di tampilkan mulai dari yang terbesar (siswa termuda) hingga terkecil (siswa termuda).

## 6. Perintah INSERT INTO

Dalam SQL, perintah INSERT INTO merupakan perintah dasar SQL bagian dari perintah untuk DML (Data Manipulation Language) Saya asumsikan Anda telah faham perbedaan DDL, DCL, dan DML. Perintah INSERT INTO dapat di gunakan untuk menambahkan record baru ke dalam tabel.

Sintaks INSERT INTO :

```
INSERT INTO nama_tabel VALUES (nilai1, nilai2, nilai3,  
...);
```

Perintah di atas di gunakan untuk menambahkan nilai '12345', 'Abdul' dan 'Jakarta' pada tabel SISWA, pastikan urutan nilai ('values') dalam urutan yang sama seperti kolom dalam tabel. Jika urutan nilai tidak sama dengan urutan kolom pada tabel, maka sintaks INSERT INTO yang di gunakan adalah sebagai berikut:

```
INSERT INTO nama_tabel (kolom1, kolom2) VALUES  
(nilai1, nilai2);
```

Perintah di atas di gunakan untuk menambahkan nilai '12345' pada kolom NIM, dan nilai 'Abdul' pada kolom NAMA dengan mengabaikan kolom lain yang tidak di isi, misalnya kolom ALAMAT.

## 7. Perintah UPDATE

Perintah UPDATE merupakan perintah dasar SQL yang di gunakan untuk memperbaiki atau mengubah nilai suatu record berdasarkan kriteria tertentu.

Sintaks UPDATE :

**UPDATE nama\_tabel SET kolom1 = nilai1, kolom2 = nilai2, ... WHERE kondisi;**

Perintah di atas di gunakan untuk memperbaiki kolom NAMA menjadi 'Ahmad' dan kolom ALAMAT menjadi 'Jakarta' pada record dengan NIM '12345' (ingat sebelumnya NIM '12345' di gunakan oleh siswa bernama 'Abdul' yang beralamat di 'Jakarta', sebut saja data tersebut keliru dan harus di perbarui).

## 8. Perintah DELETE

Hampir sama dengan perintah UPDATE, perintah DELETE juga merupakan perintah dasar SQL yang di gunakan untuk menghapus nilai suatu record berdasarkan kriteria tertentu.

Sintaks DELETE :

**DELETE FROM table\_name WHERE condition;**

Perintah di atas di gunakan untuk menghapus record dengan NIM '12345', ingat bahwa NIM tersebut di gunakan oleh siswa bernama 'Ahmad' (sebelumnya bernama 'Abdul') dan dengan di eksekusinya perintah DELETE ini maka record tersebut akan terhapus.

## 7.3. ADO. NET

ADO (Active X Data Object untuk .Net Framework) merupakan kumpulan class yang berisi komponen yang berfungsi menghubungkan antara aplikasi visual basic dan basis data. Dengan menggunakan ADO.Net, dapat dilakukan koneksi (hubungan) dengan basis data untuk memasukkan data, mengambil data,

memanipulasi data, maupun memperbarui data. Class ADO.Net terdapat pada file System.Data.

Visual basic .Net menyediakan beberapa namespace dalam .Net Framework yaitu :

- System.Data → berisi class untuk menangani obyek utama seperti DataTable, DataColumn, DataView, dan Constraint.
- System.Data.Common → untuk penyedia data (data provider) yang berisi sekumpulan fungsi-fungsi untuk mengakses data seperti DataAdapter.
- System.Data.OleDb → berisi class-class untuk menangani data provider yang sumber data dari OLEDB (Access).
- System.Data.SqlClient → untuk melakukan koneksi ke sumber data SQL Server.
- System.Data.SqlTypes → menyediakan class untuk tipe data native dalam SQ Server.
- System.Data.Odbc → menyediakan class untuk bekerja sama dengan berbagai macam driver ODBC(*Open Database Connectivity*).

Pada ADO.Net mempunyai dua buah komponen pokok yaitu :

### 1. Data Provider

Data provider merupakan komponen yang berisi nama piranti/driver yang akan digunakan oleh model obyek ADO .NET untuk mengakses sumber data.

Data provider mempunyai empat obyek utama untuk koneksi ke sumber data yaitu :

- **Object Connection** yang menyediakan koneksi ke database.
- **Object Command** yang digunakan untuk mengeksekusi perintah.



Objek Command menyediakan tiga metode yang digunakan untuk mengeksekusi perintah pada database:

- a. `ExecuteNonQuery` : Melaksanakan perintah yang memiliki nilai yang tidak kembali seperti `INSERT`, `UPDATE`, atau `DELETE`.
  - b. `ExecuteScalar` : Mengembalikan nilai tunggal dari query database.
  - c. `ExecuteReader` : Mengembalikan hasil set dengan cara objek `dataReader`
- **Objek `DataReader`** yang menyediakan `forward-only`, `read only`, `recordset` terhubung.
  - **Objek `DataAdapter`** yang populasikan `DataSet` terputus dengan data dan melakukan pembaruan.  
`DataAdapter` menyediakan empat sifat yang mewakili perintah database yaitu `SelectCommand`, `InsertCommand`, `DeleteCommand`, dan `UpdateCommand`.

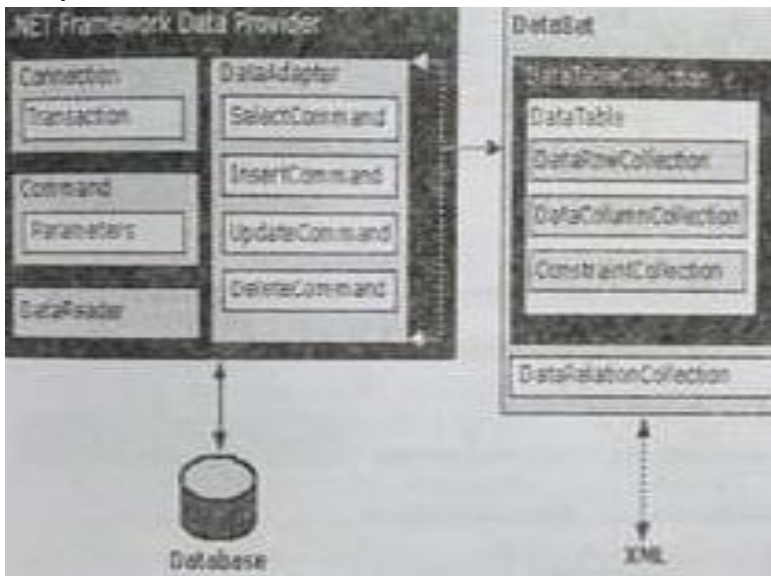
## 2. `DataSet`

`Dataset` merupakan obyek yang terpisah dari sumber data dan dapat digambarkan sebagai sebuah `recordset` yang terpisah. Pada `dataset` juga terdapat tabel, baris, kolom, relasi, `constraint`, dan `view`.

Semua tipe data, baik yang flat, hirarki atau relasional dapat disimpan di `dataset`. Karena sesuai dengan obyek dasar dari `dataset` yaitu XML, semua model penyimpanan data dapat bekerja dengan `dataset`.

Secara ringkas cara mengakses data dengan menggunakan ADO .Net dapat dijelaskan sebagai berikut (dapat dilihat pada gambar 7.1) :

- a. Lakukan koneksi ke sumber data dengan menggunakan obyek connection.
- b. Obyek command digunakan untuk mengeksekusi perintah dengan parameter tabel, SQL, atau stored procedure pada sumber data.
- c. Jika obyek command menghasilkan lebih dari sebuah nilai, obyek command akan menghasilkan sebuah DataReader untuk menghasilkan sekumpulan data tersebut.
- d. DataAdapter juga dapat digunakan untuk membangun obyek dataset.



Gambar 7.1. Arsitektur ADO .Net [Yuswanto:2007]

Teknologi ini dibangun dengan aplikasi terkoneksi dan aplikasi terputus. **Aplikasi terkoneksi** adalah suatu aplikasi dimana

pemakai aplikasi secara terus-menerus melakukan koneksi ke suatu sumber data sepanjang aplikasi tersebut dijalankan. **Aplikasi terputus** adalah suatu aplikasi dimana pemakai tidak secara kontinyu melakukan koneksi ke suatu sumber data, melainkan koneksi hanya dibuka saat mengambil atau menyimpan perubahan data ke suatu sumber data. Sedangkan saat melakukan proses yang lain, koneksi dalam keadaan tertutup, dapat dilihat pada gambar 7.2.



Gambar 7.2. Aplikasi Terputus dan Terkoneksi [Yuswanto:2007]

### 7.3.1..Net Data Provider

.Net data provider adalah sekumpulan class yang digunakan untuk melakukan koneksi ke dalam sumber data, serta untuk mengambil dan mengubah data.

Visual basic .Net mengklasifikasikan beberapa tipe dalam .Net data provider berdasarkan jenis sumber data yang digunakan yaitu :

- **SQL Server .Net Data Provider** → provider ini untuk

database SQL Server 7.0 dan versi atasnya.

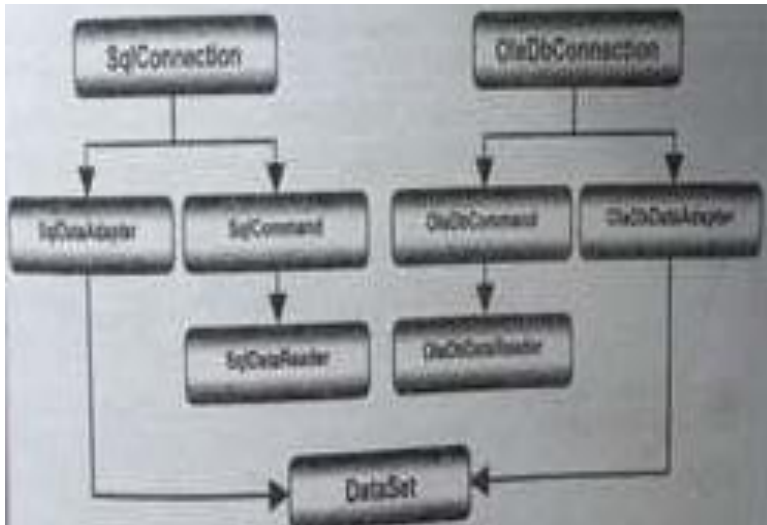
- **OleDb.Net Data Provider** → provider ini untuk database SQL Server 6.5, Microsoft Access, dan database lain yang mempunyai provider OLEDB.
- **ORACLE .Net Data Provider** → provider ini untuk database Oracle
- **ODBC .Net Data Provider** → provider ini untuk database yang hanya mempunyai driver ODBC.

.Net data provider menyediakan bermacam-macam class untuk membuat aplikasi terkoneksi yaitu :

- **Class xxxConnection** → untuk melakukan koneksi ke database.
- **Class xxxCommand** → untuk menjalankan perintah yang mengandung parameter tabel, SQL, dan stored procedure pada sumber data.
- **Class xxxDataReader** → merupakan class yang sangat efisien. Untuk melakukan pembacaan data pada database dengan sifat *forward-only* dan *read-only*.
- **Class xxxDataAdapter** → untuk menghubungkan antara database dengan dataset.

Catatan : Awalan xxx pada class-class di atas, nantinya harus diganti dengan jenis database yang digunakan.

Gambaran umum tentang fungsi dari masing-masing class dapat dilihat pada gambar 7.3.



Gambar 7.3. Class-Clas ADO .Net Pada Aplikasi Terkoneksi  
[Yuswanto:2007]

ADO .Net mempunyai beberapa komponen yang berada di toolbox visual basic. Komponen ini digunakan untuk menghubungkan aplikasi visual basic dengan database. Selain menggunakan toolbox untuk mengkoneksikan dengan database, dapat digunakan dengan menggunakan coding. Beberapa komponen ADO .Net tersebut yaitu :

- Dataset → untuk menyimpan dan memindahkan data dari banyak tabel ke dalam memori.
- DataView → untuk membuat suatu view yang dapat menentukan pilihan filter dan sorting dalam obyek dataset.
- OleDbConnection → untuk menangani koneksi ke database yang menggunakan provider OLEDB.
- OleDbDataAdapter → untuk melakukan komunikasi antara sumber data dengan Dataset yang menggunakan provider OLEDB. Melalui DataAdapter dapat dijalankan pernyataan SQL atau procedure.

- OleDbCommand → untuk melakukan konfigurasi suatu pernyataan SQL tunggal / stored procedure yang dapat dijalankan secara langsung. Menggunakan provider OLEDB.
- SqlConnection → connection yang menggunakan provider SQL.
- SqlDataAdapter → dataadapter yang menggunakan provider SQL.
- SqlCommand → command data yang menggunakan provider SQL.

### 7.3.2. Menampilkan Data Pada Tabel

Tool yang digunakan untuk menampilkan data pada tabel yang aplikasinya terkoneksi terhubung database yaitu :

1. **Datagrid** → merupakan sebuah tempat yang digunakan untuk menampung dan memperlihatkan isi data dalam bentuk tabel yang rapi. Data-data yang ada disetiap field ditempatkan dalam kolom-kolom yang ada di datagrid.
2. **Listview** → merupakan obyek yang digunakan untuk menampilkan item dengan salah satu bentuk tampilan yaitu tampilan teks saja, teks dengan icon kecil, teks dengan icon besar, dan tampilan report. Listview sangat cocok untuk menampilkan item yang berisi beberapa potongan informasi dari sekumpulan informasi yang berbentuk kolom.
3. **DatagridView** → untuk menampilkan data berbentuk grid, sama dengan datagrid, yang membedakan komponen yang terkandung di dalamnya.

### 7.3.3. Metode Pencarian Data pada DataTable

Pada pemrograman database, proses pencarian data

umumnya sangat dibutuhkan agar data yang akan dimanipulasi sesuai dengan yang diinginkan. Kesalahan dalam proses pencarian bisa berakibat fatal jika manipulasi datanya berbentuk update atau delete. Berikut beberapa metode yang digunakan untuk pencarian data pada DataTable yaitu :

### 1. Metode Find

Metode find digunakan untuk menentukan lokasi record yang memenuhi syarat/kriteria tertentu. Kriteria yang dimaksud adalah nilai pembanding dari field kunci atau mirip dengan statement WHERE pada SQL. Syarat pemakaian metode Find adalah field kunci harus dibuat primary key. Bentuk penulisan yaitu :

**DataSet.Table().Find(Kriteria)**

### 2. Metode Contains

Metode contains digunakan untuk menentukan lokasi record yang memenuhi syarat/kriteria tertentu. Perbedaan antara pencarian data dengan metode find dan contains adalah jika metode find menghasilkan DataRow sedangkan metode contains menghasilkan Boolean. Syarat dari pemakaian metode contains adalah field kunci harus dibuat primary key. Bentuk penulisan yaitu :

**DataSet.Table().Contains(Kriteria)**

### 3. Metode Select

Metode select digunakan untuk mencari data dari nilai-nilai yang ada (kriteria) pada kolom/field tetapi tidak harus dibuat primary key. Hasil pencarian datanya adalah sekumpulan datarow. Beda metode pencarian ini dengan dua metode sebelumnya adalah variabel harus berbentuk array.

Misalkan data disimpan pada variabel `dataku`, maka **Dim Dataku() as DataRow**. Bentuk penulisan yaitu :

**DataSet.Table().Select(Kriteria)**

Pada metode `select`, dapat juga dilakukan pencarian data dengan menggunakan kata kunci **LIKE** dari perintah SQL dengan menambah karakter `%`. Tujuannya untuk mengabaikan karakter tertentu, misalnya “Kode LIKE ‘MK%’”, artinya mencari data yang kodenya berawalan “MK” apapun karakter berikutnya.

Gunakan kata kunci **AND** untuk mendata data dengan kata kunci yang berbeda.

#### 4. Pencarian dengan Dua Kata Kunci

Dalam proses pencarian data tidak menutup kemungkinan data yang dicari harus menggunakan dua kata kunci atau lebih. Hal semacam ini membutuhkan sedikit ketelitian apalagi jika kedua kunci tersebut mempunyai tipe data yang berbeda, misalnya tipe pertama tipe datanya String dan kunci kedua tipe datanya tanggal.

- **Metode Find**

Syarat utama dari proses pencarian data dengan menggunakan metode `find` adalah nilai-nilai yang menjadi kriteria harus berada pada kolom/field yang merupakan primary key dari tabel tersebut.

Prinsip dasar lainnya adalah menyiapkan variabel array dengan tipe data obyek sebagai tempat untuk menampung nilai-nilai yang akan dicari.

- **Metode Contains**

Metode `contains` mempunyai syarat yang sama dengan



metode Find, yaitu field kuncinya harus dibuat primary key.

Prinsip dasarnya juga sama yaitu menyiapkan variabel array dengan tipe data obyek sebagai tempat untuk menampung nilai-nilai yang akan dicari.

- **Metode Select**

Pada prinsipnya, pencarian dengan dua kata kunci atau satu kata kunci dengan metode select adalah sama. Hanya pada pencarian dengan dua kunci, penulisan keuda kriterianya menggunakan kata kunci “AND”.

#### **7.3.4. Metode Pencarian Data pada DataView**

DataView merupakan sebuah obyek yang digunakan untuk menampilkan data yang ada pada DataTable. Pada prakteknya, sebuah DataView hanya dapat mengambil data berdasarkan DataTable beserta seluruh kolomnya.

Terdapat perbedaan dalam melakukan pencarian dan pengurutan data dengan menggunakan DataView dibandingkan dengan menggunakan perintah “select” pada DataTable yaitu :

- Pada metode Select di DataTable, tidak dapat langsung melakukan binding data tetapi pada DataView, dengan mudah melakukannya.

Data Binding adalah sebuah fasilitas yang tersedia dalam .NETFramework untuk menghubungkan kontrol yang ada pada form dengan data source, seperti pada DataSet, DataTable, DataView atau Array.

- Proses pencarian data pada DataView didasarkan pada nilai di kolom/field mana pun, tidak harus pada kolom/field yang

mempunyai primary key. Sedangkan pada DataTable, proses pencarian data didasarkan pada nilai di kolom/field yang sudah dijasikan primary key.

Berikut ini sintak penulisan untuk membuat sebuah DataView yaitu :

```
Dim <nama_DataView> As New DataView  
<nama_DataView> = <nama_DataSet>.DefaultView
```

Berikut beberapa metode pencari Data pada DataView yaitu :

- **Metode RowFilter**

Metode ini bertujuan untuk mencari/memfilter data berdasarkan kriteria tertentu. Metode ini mirip dengan metode Select sehingga dapat menambahkan kata kunci “**LIKE**”. Juga dapat disertakan metode **Sort** yang bertujuan untuk mengurutkan data secara **Ascending** atau **Descending**. Selain itu, juga bisa dibuat kriteria dengan mengabaikan karakter tertentu, seperti pada SQL yaitu dengan tanda “%”. Contoh, ingin memfilter field Nama yang berawalan S, penulisannya yaitu **DV.Brg.RowFilter="Nama='S%'"**.

Konsep yang ditawarkan oleh metode RowFilter ADO .NET akan menyembunyikan baris data yang tidak sesuai dengan kriteria yang dicari pada DataTable. DataTable yang dimaksud adalah DataTable yang telah berasosiasi dengan DataView yang melakukan metode RowFilter tersebut.

Obyek DataView mempunyai properti **RowStateFilter** yang bertujuan untuk menentukan **State** pada row yang akan digunakan. Bentuk penulisan dari properti adalah :

**DV\_Barang.RowFilter** =  
**DataViewRowState.CurrentRowState.**

Berikut macam-macam nilai dari RowStateFilter yaitu :

- ❖ CurrentRows → semua record baru dan yang mengalami perubahan atau tidak.
- ❖ Deleted → semua record yang telah terhapus.
- ❖ ModifiedCurrent → semua record yang telah diubah.
- ❖ ModifiedOriginal → nilai awal pada record yang telah diubah.
- ❖ New → semua record baru.
- ❖ OriginalRows → semua record dengan nilai awalnya, termasuk record yang tidak mengalami perubahan dan yang telah dihapus.
- ❖ Unchanged → semua record yang tidak mengalami perubahan nilai.

- **Metode Find**

Metode Find bertujuan untuk mencari data berdasarkan kriteria tertentu, tetapi nilai yang dihasilkan berupa nilai index dari posisi baris datanya. Pencarian data dikatakan gagal jika nilai indexnya -1.

Syarat dari pemakaian metode ini adalah datanya harus diurutkan, yaitu dengan mengatur properti Sort pada field/kolom yang dijadikan kriteria/kunci. Misalnya, ingin mencari barang dengan kode “MK789” maka :

**DV\_BRG.Sort=”kode”**

**Dim idx As Integer = DV\_BRG.Find(“MK789”)**

- **Metode FindRows**

Metode ini bertujuan untuk memfilter data sehingga

dihasilkan sekumpulan DataRow. Sebelum datanya dicari, datanya harus diurutkan terlebih dahulu (sort) berdasarkan field kunci. Misalnya, ingin mencari barang dengan kode “MK789” maka :

**DV\_BRG.Sort=”kode”**

**Dim cari As DataRowView**

**cari = DV\_BRG.FindRows(“MK789”)**

## 7.4. Membuat Database Microsoft Access 2013

Pada subbab ini dijelaskan cara membuat database pada Microsoft Access 2013.

### 7.4.1. Membuat Database Baru Pada Microsoft Access 2013

Berikut cara untuk membuat database baru di Microsoft Access 2013 yaitu :

1. Bukalah aplikasi Micosoft Access database (Microsoft Access 2013). Untuk membuka aplikasi Micosoft Access 2013 dapat dilakukan dengan cara double klik pada icon Microsoft Access. Atau dengan cara klik Start → Microsoft Office → Microsoft Access 2013.



Gambar 7.4. Cara Membuka Microsoft Access 2013

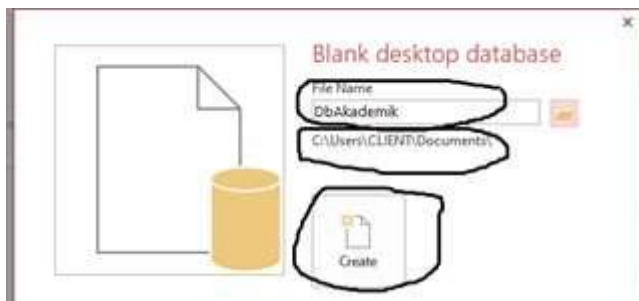
2. Lalu pilih **blank desktop**



Gambar 7.5. Memilih Blank Database

3. Kemudian muncul kotak dialog seperti pada gambar 7.6 lalu tulis nama file pada isian File Name kemudian pilih **“create”**. Nama file ini nantinya disebut nama database yang akan digunakan pada mengkoneksikan database. Perhatikan juga alamat penyimpanan database yang terletak di bawah File Name.

Nb : untuk mengisikan nama database pastikan tidak ada spasi dan jangan menggunakan nama keyword (kata kunci) dari Microsoft Access dan Visual Basic.



Gambar 7.6. Membuat Database Baru

4. Setelah ditekan create seperti pada langkah 3 di atas, muncullah datasheet. Ini merupakan tampilan awal saat membuka Microsoft Access 2013. Secara otomatis terdapat table1.



Gambar 7.7. Datasheet

4. Table1 di atas dapat digunakan oleh pengguna dengan cara mengedit table1. Atau user juga dapat menghapus table1 ini jika tidak ingin menggunakannya dengan cara menutup tabel1 ini dan jangan disimpan. Cara menutup tabel pada table1 ini yaitu **klik kanan pada tabel1 → close** atau dengan cara **menekan tombol silang yang berada di kanan kata table1** (lihat bagian yang dilingkari pada gambar 7.8).

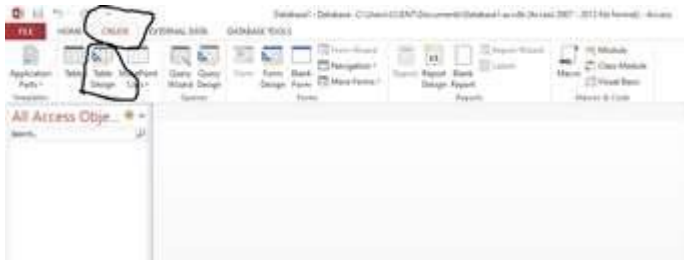


Gambar 7.8. Tampilan Table1

## 7.4.2. Membuat Tabel Baru Pada Microsoft Access 2013

Berikut cara untuk membuat tabel baru pada Microsoft Access 2013 yaitu :

1. Kemudian untuk menambahkan tabel baru yaitu dengan cara pilih menu **CREATE** → **table design**.



Gambar 7.9 Tampilan Table Design

Lalu muncullah kotak dialog seperti pada gambar 7.10. terdapat Field Name dan Data Type. Field name merupakan nama kolom pada suatu tabel. Jadi, nama kolom yang menyusun suatu tabel dituliskan di field name ini. Sedangkan data type merupakan tipe data pada suatu kolom. Suatu kolom dapat menyimpan data tipe teks, gambar, angka, tanggal/waktu, dan lain-lain. Untuk mengisikan tipe data, user diberikan beberapa pilihan tipe data yang ada di Microsoft Access 2013, sehingga user dapat memilih salah satu tipe data yang cocok untuk kolom tersebut, dapat dilihat pada gambar 7.10.



Gambar 7.10. Tampilan Awal Table Design

Beberapa tipe data yang ada pada Microsoft Access 2013 yaitu :

- **Auto Number.** Nilai atau angka yang terisi secara otomatis.
- **Text / Short Text** digunakan untuk data alfanumerik (huruf, angka, simbol) panjang karakter maksimal 255. Di Microsoft Access versi 2013 sampai 2016 namanya short text tetapi di microsoft access 2010 dibawah namanya text.
- **Long Text / Memo.** kegunaan type data ini sama seperti short text hanya saja kapasitasnya lebih besar, type data long text ini bisa menampung kalimat dan paragraf maksimal 64.000 karakter. Di Microsoft Access versi 2013 ke atas namanya Long Text tetapi di Microsoft Access 2010 dibawah namanya Memo.
- **Number** digunakan untuk menyimpan type data angka atau Numeric.
- **Date/Time** digunakan untuk menyimpan data berupa tanggal, bulan, tahun.
- **Currency** digunakan untuk menyimpan data keuangan atau nilai nominal.
- **Ole Object** digunakan untuk memasukkan data berupa gambar atau foto.
- **Hyperlink** digunakan untuk membuat sebuah link yang dapat memanggil suatu situs atau file atau dokumen kemudian menampilkannya.
- **Attachment** digunakan untuk melampirkan suatu file berupa gambar, dokumen, musik mp3, atau video.
- **Calculated** digunakan Untuk menampilkan data hasil perhitungan antara beberapa field, ketika memilih data type calculated maka akan Secara otomatis terbuka



expression builder disitu kita bisa menuliskan sebuah rumus penjumlahan.

- **Lookup Wizard** digunakan untuk membuat sebuah drop box yang menampilkan data disuatu field dan tabel yang berbeda dan bisa juga membuat data pilihan sendiri.
2. Lalu user diminta untuk mengisikan field name dan memilih data type yang sesuai dengan data yang disimpan dalam suatu kolom. Isikan hingga beberapa kolom sesuai dengan kebutuhan.



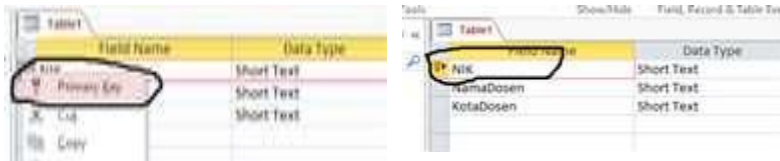
Gambar 7.11. Isian Field Name dan Data Type yang Diisikan Oleh User

3. Setelah semua kolom telah ditulis beserta tipe datanya kemudian langkah selanjutnya yaitu memberikan primary key pada suatu kolom.

Langkah 3 ini dilakukan jika suatu tabel diperlukan untuk menambahkan primary key pada suatu kolom. Ada beberapa tabel yang tidak memerlukan primary key contohnya yaitu tabel transaksi.

Primary key ini merupakan kunci/identitas sehingga kolom yang menjadi primary key ini akan menjadi suatu identitas/kunci dari tabel tersebut.

Kolom yang menjadi primary key mempunyai syarat yaitu data yang diisi harus unik atau tidak boleh sama dengan data lain, dan data yang diisikan tidak boleh kosong atau data harus terisi.

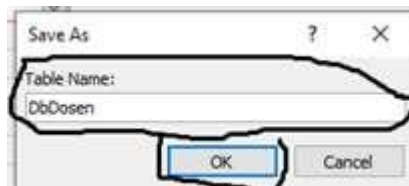


Gambar 7.12. Langkah 3 Menambahkan Primary Key (Kiri) dan Hasil Penambahan Primary Key (Kanan)

4. Langkah terakhir, **menyimpan tabel** dengan cara menekan **ctrl + S** lalu **isikan nama tabel** pada isian Table Name lalu tekan **OK**.

Nb : untuk mengisikan nama tabel pastikan tidak ada spasi dan jangan menggunakan nama keyword (kata kunci) dari Microsoft Access dan Visual Basic.

Pastikan tabel dalam keadaan tertutup saat menjalankan program vb, sebab akan terjadi error saat menjalankan vb jika tabel dalam keadaan terbuka



Gambar 7.13. Menyimpan Tabel

### 7.4.3. Membuat Relationship Antar Tabel pada Microsoft Access 2013

Berikut cara untuk membuat relationship antar tabel pada Microsoft Access 2013 yaitu :

1. Untuk membuat relationship antar tabel pastikan memiliki lebih dari 1 tabel agar bisa dibuat hubungan antara tabel satu dengan yang lain.
2. Hubungkan antar tabel. Pertama, Double klik pada salah satu tabel yang akandihubungkan. Lalu pilih menu **TABLE > relationship** seperti yang terlihat pada gambar 7.12.



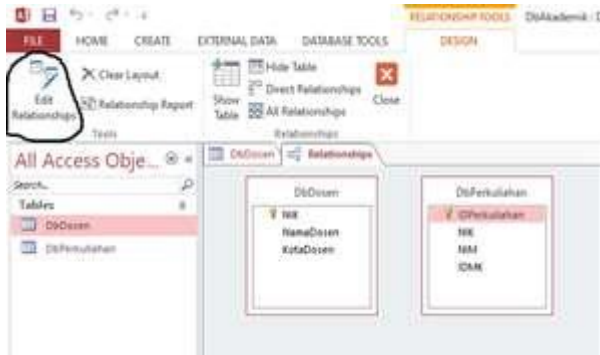
Gambar 7.14. Langkah Awal Relationship Antar Tabel

Lalu muncul kotak dialog yang terlihat pada gambar 7.14. lalu pilih menu **show table > pilih tabel yang akan dihubungkan > klik Add.**



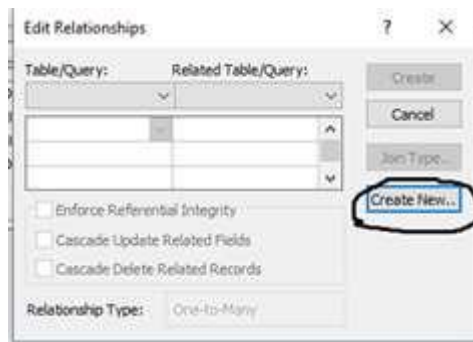
Gambar 7.15. Kotak Dialog Penambahan Tabel

3. Lalu muncul tabel yang dipilih yang ditampilkan pada gambar 7.16. Kemudian pilih menu **Edit relationship.**



Gambar 7.16. Edit relationship dan Tabel yang Ditambahkan untuk Relationship

4. Lalu muncul kotak dialog seperti gambar 7.17. Klik **create new**. Nanti akan muncul kotak dialog yang ditampilkan pada gambar 7.18.

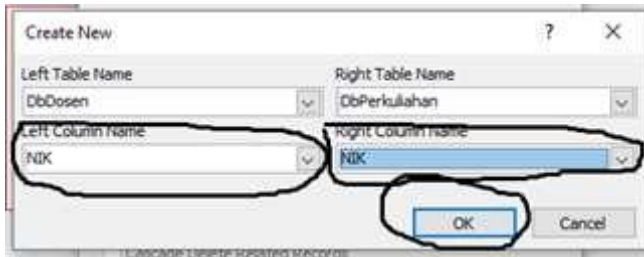


Gambar 7.17. Kotak Dialog Edit Relationship

5. Pilih tabel yang diinginkan dengan primary key-nya. Hubungkan tabel satu yang memiliki primary key dengan tabel lain yaitu dengan memasang nama kolom yang sama dengan nama kolom yang berada pada tabel awal.  
Contoh tabel kiri bernama tabel DbDosen dengan kolom NIK (memiliki primary key) dipasangkan dengan tabel lain yaitu tabel

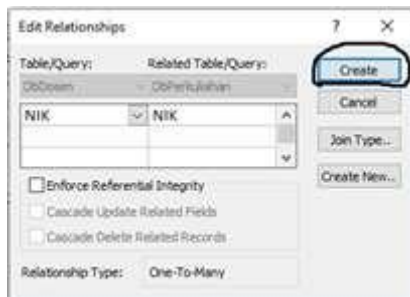
DbPerkuliahan dengan kolom NIK. Jika dipasangkan dengan kolom yang lain maka akan error.

Lalu klik **OK**.



Gambar 7.18. Kotak Dialog Relationship Antar Tabel

6. Maka tampilan akan seperti pada gambar 7.19. Lalu pilih **create**, maka kedua tabel tersebut sudah terhubung.



Gambar 7.19. Edit Relationship Setelah Ditambahkan Tabel dan Kolom

7. Tampilan akhir pada Relationship yaitu tabel yang satu terhubung dengan tabel yang lain.



Gambar 7.20. Relationship dengan Tabel Saling Terhubung

## 7.5. Menghubungkan Visual Basic dengan Microsoft Access 2013

Untuk menkoneksi visual basic ke Microsoft Acces 2013, maka langkah yang dilakukan adalah :

### A. Membuat koneksi ke database

Berikut langkah untuk membuat koneksi ke database

:

1. Tambahkan module pada project yaitu dengan **cara klik kanan – Add – module – lalu ketikkan nama module.**



Gambar 7.21. Cara Menambahkan Module Koneksi

2. Lalu ketikkan kode program di bawah ini. Ketikkan kode program berikut dibagian paling atas.

Kode program ini digunakan untuk mengkoneksi visual basic ke database yaitu Microsoft Access.

```
Imports System.Data.OleDb
```

Jika database yang digunakan adalah SQL Server maka perintah yang digunakan adalah :

```
Imports system.data.sqlclient
```

Jika database yang digunakan MySQL maka perintah yang digunakan adalah :

```
Imports MySql.data.MySqlclient
```

Jika database yang digunakan adalah Oracle maka perintah yang digunakan adalah :

```
Imports Oracle.dataaccess.client
atau
imports System.Data.OracleClient
```

3. Tambahkan kode program di dalam module yaitu :

```
Public conn As OleDbConnection
Public daData As New OleDbDataAdapter
Public dsData As New DataSet
Public query As String
```

4. Tambahkan sebuah prosedur, sebagai contoh prosedur koneksiAkademik (penulisan tanpa spasi). Penulisannya adalah `Public Sub KoneksiAkademik()`. Dengan menggunakan keyword `public sub` sehingga dapat diakses oleh modul atau form lain. Dan diakhiri dengan `End Sub`.
5. Di dalam prosedur tambahkan kode program berikut.

```
conn = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=D:\Ngajar Q\DbAkademik.accdb;")
```

Kode program di atas digunakan untuk memanggil database yang digunakan. Kode program `"Provider=Microsoft.ACE.OLEDB.12.0;"` digunakan jika

menggunakan database Microsoft 2010 ke atas (yang mempunyai ekstensi .accdb). Kode program “`Data Source=D:\Ngajar Q\DbAkademik.accdb;`” menunjukkan alamat database disimpan dan nama database beserta ekstensinya.

## B. Membuat Kode program Program Ke Database Pada Form

Module koneksi telah selesai, langkah selanjutnya adalah mendesain tampilan dan membuat program agar program berjalan sesuai kebutuhan. Mengingat module koneksi yang telah dibuat tidak bisa dijalankan jika tidak dibuat aplikasi programnya (form dan kode program form). Di dalam kode program form inilah module koneksi dipanggil agar dapat menjalankan fungsinya yaitu menghubungkan visual basic ke database yang digunakan. Berikut beberapa kode program pada form yang berkaitan dengan koneksi database diantaranya adalah :

1. Ketikkan kode program berikut ini. Kode program ini diketik di bagian paling atas form yaitu :

```
Imports System.Data.OleDb
```

kode program ini sama dengan kode program yang ada pada module koneksi.

2. Pada form\_load ketikkan kode program berikut.

```
Try
    KoneksiAkademik()
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
```

Kode program “koneksiAkademik” diambil dari prosedur koneksiAkademik yang ada pada module koneksi.

Penulisan kode program “koneksiAkademik” dituliskan diantara fungsi try...catch... bertujuan untuk “menangkap



error” yang disebabkan oleh kegagalan koneksi. Error ini ditampilkan pada kotak dialog pesan. Jika tidak menggunakan fungsi try...catch... maka program akan berhenti dengan sendirinya jika ada error tanpa mengetahui kesalahannya apa.

## 7.6. Latihan Program

Untuk latihan program ini yaitu membuat koneksi database sederhana dari visual basic ke Microsoft Access 2013 disertai dengan input, update, dan delete data. Untuk data yang digunakan hanya satu tabel yaitu tabel dosen saja.

Nah, untuk membuat program database sederhana ini dimulai dengan membuat database pada Microsoft Access 2013. Buatlah database dengan nama **DbAkademik** yang di dalamnya terdapat tabel dosen yang bernama **DbDosen**. Berikut nama kolom dan tipe data yang ada pada tabel DbDosen yang dapat dilihat pada gambar 7.20. Field Name / kolom “**NIK**” dijadikan **primary key**.



Field Name	Data Type
NIK	Short Text
NamaDosen	Short Text
KotaDosen	Short Text

Gambar 7.22. Isi Tabel DbDosen (Nama Kolom dan Tipe Data)

Setelah selesai membuat database, lalu buatlah program pada visual basic. Langkah awal dalam membuat program pada visual basic adalah membuat module koneksi. Module koneksi inilah yang nantinya akan dipanggil pada form-form yang akan terhubung dengan database.

Buatlah module koneksi dengan nama **KoneksiDbAkademik**. Ketikkan kode program berikut pada

module koneksi, dapat dilihat pada gambar 7.23 (cara pembuatan module dan keterangan kode program yang ada di module koneksi telah dijelaskan di atas pada subbab 7.5 bagian A).

```
Imports System.Data.OleDb

References:
Module KoneksiDbAkademik
    Public conn As OleDbConnection
    Public daData As New OleDbDataAdapter
    Public dsData As New DataSet
    Public query As String
End Module

References:
Public Sub KoneksiAkademik()
    conn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source=0:\DbAkademik.accdb;")
End Sub

End Module
```

Gambar 7.23. Kode program Module Koneksi

Programmer haruslah memperhatikan isi pada Data Source (lihat gambar 7.23). Isi data source merupakan alamat dimana database disimpan dan nama database serta ekstensi database. Di contoh program di atas bahwa database disimpan pada folder D:\ dan nama database adalah DbAkademik.accdb. Jadi, untuk isi data source masing-masing programmer berbeda bergantung pada letak penyimpanan database.



Gambar 7.24. Desain Form Dosen

Setelah membuat module koneksi, langkah selanjutnya adalah membuat design form. Form inilah yang akan dihubungkan ke database. Berikut design form Dosen (sebagai catatan bahwa untuk menampilkan data pada form ini menggunakan tool ListView).

Berikut keterangan properties dari form Dosen :

Tabel 7.2. Tabel Tool Form Dosen

<b>Tool</b>	<b>Properties</b>	<b>Keterangan</b>
Label 1	Text	NIK
Label 2	Text	Nama Dosen
Label 3	Text	Kota Dosen
TextBox 1	Name	txtNIK
TextBox 2	Name	txtNamaDosen
TextBox 3	Name	txtKotaDosen
Button 1	Text	Simpan
	Name	btnSimpan
Button 2	Text	Ubah
	Name	btnUbah
Button 3	Text	Hapus
	Name	btnHapus
Button 4	Text	Bersih

	Name	btnBersih
ListView1	FullRowSelect	True
	GridLines	True
	MultiSelect	False
	View	Details

Lalu tambahkan kode program pada :

- Ketikkan kode program berikut ini. Kode program ini diketik di bagian paling atas form.

```
Imports System.Data.OleDb. coding
```

- Form dengan event load :

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    txtNik.Enabled = False
    txtNamaDosen.Focus()
    btnUbah.Enabled = False
    btnHapus.Enabled = False
    Try
        KoneksiAkademik()
        PosisiList()
        IsiList()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
```

- Buatlah procedure posisi list → untuk mengatur posisi kolom pada listview1.

```
'Membuat kolom pada ListView
Private Sub PosisiList()
    Try
        With ListView1.Columns
            .Add("NIK", 100)
            .Add("Nama Dosen", 100)
            .Add("Kota Dosen", 100)
        End With
    Catch ex As Exception
```

```

        MessageBox.Show(ex.Message)
    End Try
End Sub

```

- Buatlah procedure mengambil data dari list → untuk mengambil data dari listview1 dan ditampilkan ke textbox.

```

'Mengambil data dari ListView
Private Sub AmbilDataListView()
    With ListView1.SelectedItems
        Try
            txtNik.Text = .Item(0).SubItems(0).Text
            txtNamaDosen.Text = .Item(0).SubItems(1).Text
            txtKotaDosen.Text = .Item(0).SubItems(2).Text
        Catch ex As Exception
        End Try
    End With
End Sub

```

- Buatlah procedure isi list → untuk mengisi data pada listview1.

```

Private Sub IsiList() 'Menampilkan data pada ListView
    Try
        query = "SELECT * FROM DbDosen ORDER BY NIK"
        daData = New OleDbDataAdapter(query, conn)
        dsData = New DataSet
        daData.Fill(dsData)
        ListView1.Items.Clear()
        For a = 0 To dsData.Tables(0).Rows.Count - 1
            With ListView1
                .Items.Add(dsData.Tables(0).Rows(a).Item(0))
                .Items(a).SubItems.Add(dsData.Tables(0).Rows(a).Item(1))
                .Items(a).SubItems.Add(dsData.Tables(0).Rows(a).Item(2))
                txtNik.Text = "DS-" & Now.Year & "-" & Now.Month & "-" &
                Now.Day & "-" & a + 1
            End With
        Next
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub

```

- Buatlah procedure bersih → untuk membersihkan textbox.

```

Private Sub bersih()
    IsiList()

```

```
txtNamaDosen.Text = ""
txtKotaDosen.Text = ""
```

End Sub

- Tulislah kode program pada tombol bersih dengan cara memanggil sub bersih, digunakan untuk memberihkan ada yang ada pada seluruh isian textbox dan meletakkan kursor pada isian Nama Dosen.

```
Private Sub btnBersih_Click(sender As Object, e As EventArgs) Handles
btnBersih.Click
    txtNik.Enabled = False
    txtNamaDosen.Focus()
    bersih()
End Sub
```

- Tulislah kode program pada ListView1, yang digunakan untuk menekan data yang tampil pada listview dan data yang sedang diklik akan tampil di textbox untuk diedit atau

dihapus.

```
Private Sub ListView1_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles ListView1.SelectedIndexChanged
    txtNik.Enabled = False
    btnUbah.Enabled = True
    btnHapus.Enabled = True
    btnSimpan.Enabled = False
    txtNamaDosen.Focus()
    Try
        AmbilDataListView()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
```

End Sub

- Tulislah kode program pada tombol simpan, yang digunakan untuk menyimpan data pada database.

```
Private Sub btnSimpan_Click(sender As Object, e As EventArgs) Handles
btnSimpan.Click
    Dim x As Object = MessageBox.Show("Data TERSIMPAN", "SIMPAN",
    MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
    Try
        query = "INSERT INTO DbDosen VALUES ('" & txtNik.Text & "', '" &
        txtNamaDosen.Text & "', '" & txtKotaDosen.Text & "')"
    End Try
End Sub
```

```

daData = New OleDbDataAdapter(query, conn)
dsData = New DataSet
daData.Fill(dsData)
'MsgBox("DATA TERSIMPAN", MsgBoxStyle.Information, "SIMPAN")
IsiList()
bersih()
If x = Windows.Forms.DialogResult.OK Then
    IsiList()
    txtNamaDosen.Focus()
End If
Catch ex As Exception
    MsgBox("NO NIK TIDAK Boleh SAMA", MsgBoxStyle.Critical, "Error")
End Try
End Sub

```

- Tulislah kode program pada tombol ubah, digunakan untuk mengubah data yang ada pada database berdasarkan NIK.

```

Private Sub btnUbah_Click(sender As Object, e As EventArgs) Handles btnUbah.Click
    Dim x As Object = MessageBox.Show("Data Berhasil DIUBAH", "UBAH",
    MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
    Try
        query = "UPDATE DbDosen SET NamaDosen = '" & txtNamaDosen.Text & "',
        KotaDosen = '" & txtKotaDosen.Text & "' WHERE NIK='" & txtNik.Text & """"
        daData = New OleDbDataAdapter(query, conn)
        dsData = New DataSet
        daData.Fill(dsData)
        IsiList()
        bersih()
        btnUbah.Enabled = False
        btnSimpan.Enabled = True
        btnHapus.Enabled = False
        txtNik.Enabled = False
        If x = Windows.Forms.DialogResult.OK Then
            IsiList()
            txtNamaDosen.Focus()
        End If
    Catch ex As Exception
    End Try
End Sub

```

- Tulislah kode program pada tombol hapus, digunakan untuk menghapus data yang ada pada database berdasarkan NIK.

```

Private Sub btnHapus_Click(sender As Object, e As EventArgs) Handles btnHapus.Click
    Dim x As Object = MessageBox.Show("Apakah Data akan DIHAPUS?", "HAPUS",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk)
    If x = Windows.Forms.DialogResult.OK Then
        query = "DELETE FROM DbDosen WHERE NIK='" & txtNik.Text & """"
    End If
End Sub

```

```

daData = New OleDbDataAdapter(query, conn)
dsData = New DataSet
daData.Fill(dsData)
MsgBox("DATA BERHASIL DIHAPUS", MsgBoxStyle.Information, "HAPUS")
IsiList()
bersih()
txtNik.Focus()
Elseif x = Windows.Forms.DialogResult.Cancel Then
bersih()
btnUbah.Enabled = False
btnHapus.Enabled = False
txtNik.Enabled = True
txtNik.Focus()
btnSimpan.Enabled = True
End If
End Sub

```

## BAB VIII

### MEMBUAT LAPORAN DENGAN CRYSTAL REPORT

Pada bab 8 ini dijelaskan cara download dan instalasi crystal report, cara membuat laporan dengan crystal report (saat membuat laporan dengan crystal report, programmer dapat menampilkan laporan sederhana maupun menggunakan chart), menampilkan laporan crystal report pada form aplikasi, serta menampilkan laporan berdasarkan kriteria tertentu menggunakan kode program.

Mencetak laporan umumnya dijadikan sebagai hasil akhir dari suatu rangkaian pengolahan data. Oleh karena itu proses yang dilakukan membutuhkan teknik tersendiri agar hasil laporannya mudah dibaca dan dipahami. Banyak paket program yang ditawarkan dan salah satunya adalah Crystal Reports.

Crystal Reports merupakan salah satu paket program yang digunakan untuk membuat, menganalisa, dan menterjemahkan informasi yang terkandung dalam database ke dalam berbagai jenis laporan. Crystal Reports dirancang untuk membuat laporan yang dapat digunakan dengan berbagai bahasa pemrograman berbasis



Windows, seperti Visual Basic, Visual C/C++, Visual Interdev, dan Borland Delphi.

Beberapa kelebihan yang dimiliki program Crystal Reports, yaitu :

- ❖ Pembuatan laporan dengan Crystal Reports tidak terlalu sulit dan tidak banyak melibatkan kode program.
- ❖ Program Crystal Reports banyak digunakan karena mudah terintegrasi dengan bahasa pemrograman lain.
- ❖ Fasilitas impor hasil laporan yang mendukung format-format paket program lain seperti Microsoft Office, Adobe Acrobat, HTML, dan sebagainya.

### **8.1. Instalasi Crystal Report**

Pada visual studio 2013, crystal reports diinstal terpisah. Untuk mendownload aplikasi crystal reports dapat dikunjungi situs <https://getintopc.com/software/office-tools/download-crystal-reports-developer-for-visual-studio/>.

Aplikasi crystal reports yang digunakan adalah crystal reports for visual studio 2013. Berikut langkah instalasinya yaitu :

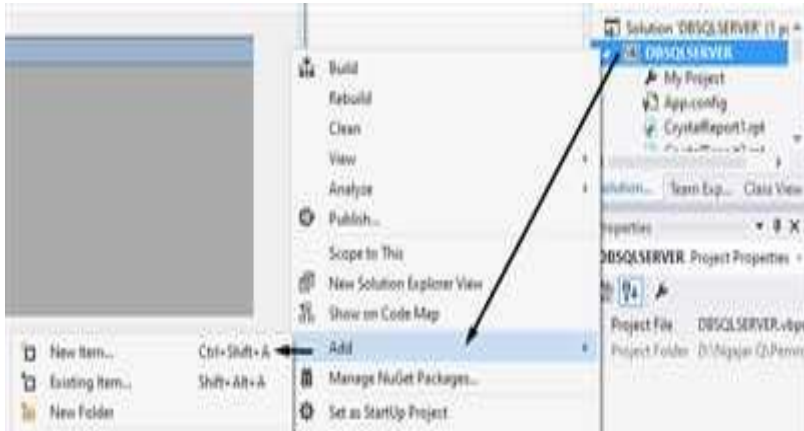
1. Buka file hasil download dengan cara double klik pada file tersebut dan memasukkan password nya yaitu 123.
2. Lalu double klik pada file Application (dalam bentuk .rar), kemudian ikuti langkah instalasi hingga selesai.
3. Pastikan visual studio 2013 dalam keadaan tertutup saat instalasi crystal report.

### **8.2. Membuat Laporan Baru**

Sebelum menampilkan laporan pada form, terlebih dahulu harus membuat laporan baru pada crystal reports dan

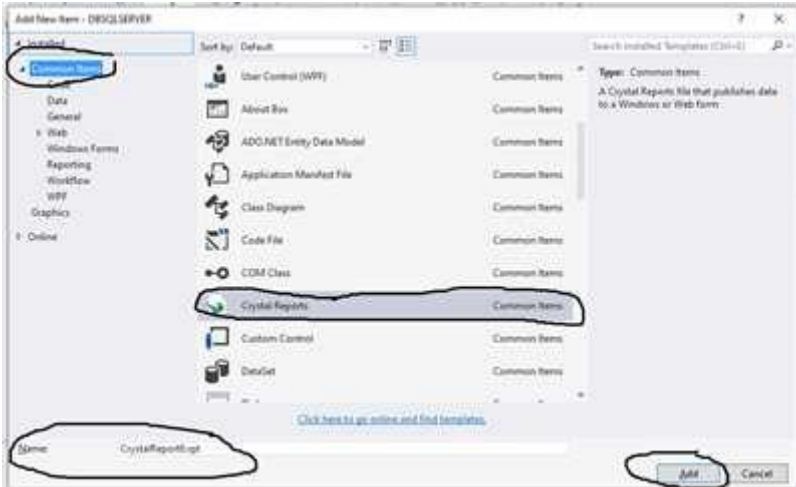
mendesainnya. Berikut langkah untuk membuat laporan baru pada crystal reports yaitu :

1. Buka kembali program yang akan dibuat laporannya.
2. Pada jendela **Solution Explorer**, klik kanan pada **nama\_proyek**, pilih **Add > Add New Item**, lihat gambar 8.1.



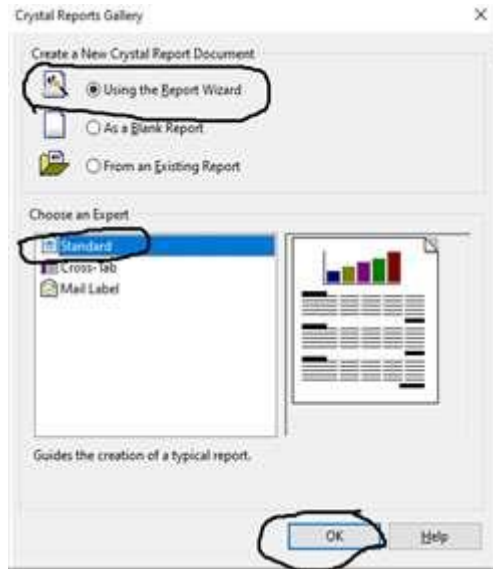
Gambar 8.1. Menambah Item Baru

3. Pada Jendela **Add New Item**, pilih **Common Items > Crystal Report**. Gantilah nama laporan jika perlu pada bagian **Name**. kemudian klik **Add**.



Gambar 8.2. Kotak Dialog Add New Item

4. Lalu muncul kotak dialog **Crystal Reports Gallery**. Pilihlah jenis laporan yang akan ditampilkan. Untuk contoh, pilih **Using Reports Wizard** dan **Standard** klik **OK**. (penjelasan lebih lanjut ada di akhir subbab).



Gambar 8.3. Crystal Reports Gallery

5. Muncul kotak dialog **Standard Reports Creation Wizard**.

Pada kotak dialog ini, user diminta untuk memilih koneksi database yang akan digunakan yaitu Project Data berisi seluruh data yang ada di proyek yang sedang dibuat, MyConnections berisi database yang pernah digunakan, atau Create New Connection berisi pilihan untuk membuat koneksi database baru.

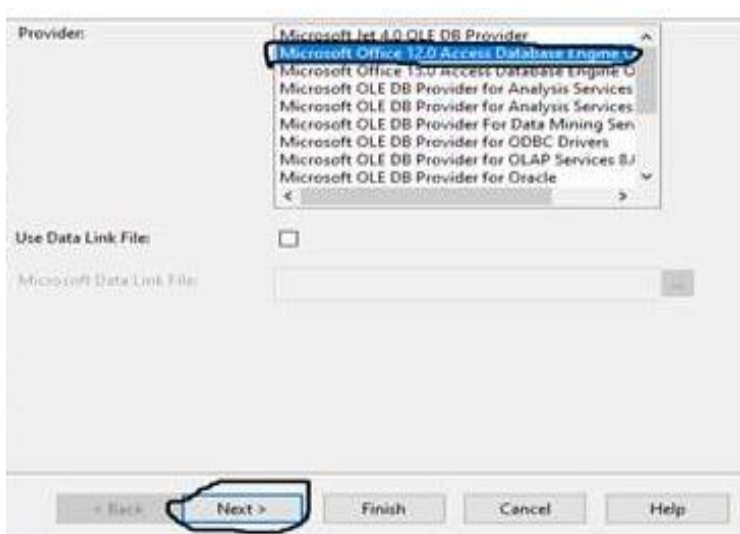
Untuk **membuat koneksi database baru**, maka pilih **Create New Connection > Klik Kanan pada Make New Connection > Klik Make New Connection lagi**.

Pilihan ini digunakan jika menggunakan database Access / SQLServer dan menggunakan teknologi ADO .Net maka pilih **OLEDB ADO**.



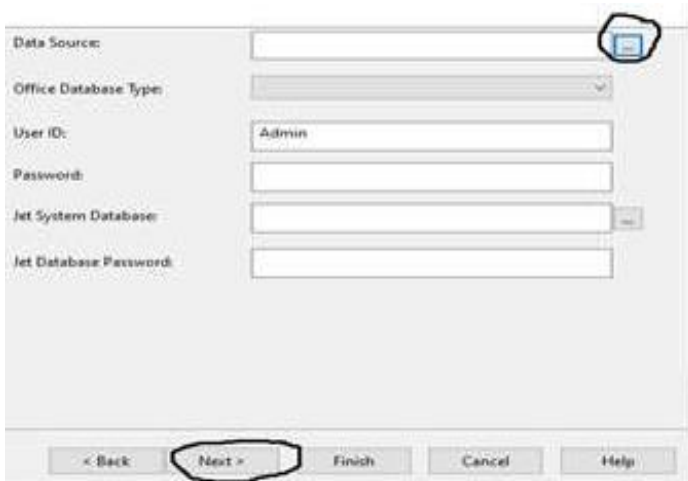
Gambar 8.4. Memilih Sumber Data dari OLE DB (ADO)

6. Pada kotak dialog **OLE DB (ADO)**, tentukan provider dari database yang akan digunakan. Pada contoh menggunakan **Microsoft Office 12.0 Access Database Engine**, karena menggunakan database Miceosoft Access 2013 yang berekstensi .accdb. Lalu tekan **Next**.



Gambar 8.5. Memilih Provider

7. Lalu pilihlah **Data Source** dengan **menekan tiga titik** disebelah kanan > **Carilah Database** yang akan digunakan > **Tekan Next.**



The screenshot shows a dialog box for selecting a data source. It contains several input fields: 'Data Source' (empty), 'Office Database Type' (dropdown), 'User ID' (containing 'Admin'), 'Password' (empty), 'Jet System Database' (empty), and 'Jet Database Password' (empty). A small icon with three dots is circled in red in the top right corner of the 'Data Source' field. At the bottom, the 'Next >' button is circled in red.

Gambar 8.6. Memilih Data Source

8. Muncul kotak dialog **Advance Information** > **Tekan Finish.**



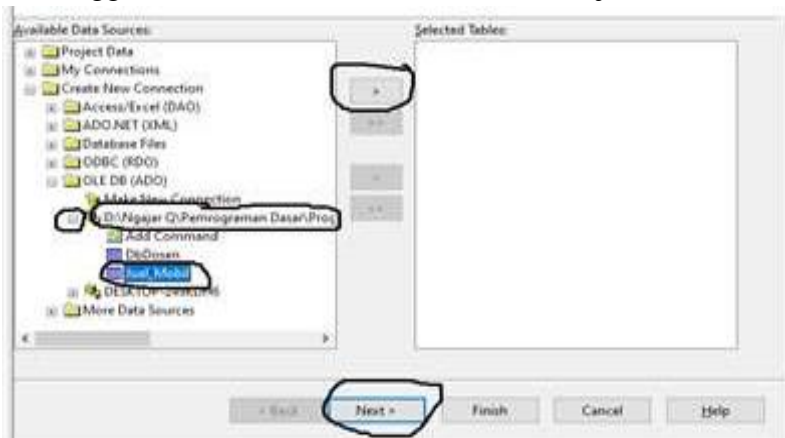
The screenshot shows the 'Advance Information' dialog box. It features a table with two columns: 'Name' and 'Value'. The table contains two entries: 'Locale Identifiers' with a value of '1033' and 'OLE DB Services' with a value of '-8'. Below the table are three buttons: 'Add Property ...', 'Edit Value ...', and 'Remove Property ...'. At the bottom, the 'Finish' button is circled in red.

Name	Value
Locale Identifiers	1033
OLE DB Services	-8

Gambar 8.7. Advance Information

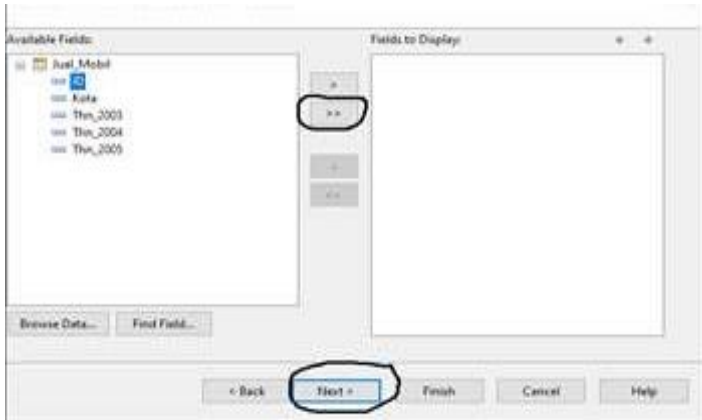
9. Tampilan kotak dialog kembali ke kotak dialog **Standard Reports Creation Wizard**. Koneksi database yang telah berhasil ditambahkan akan muncul dibawah “Make New Connection”

Lalu **pilih database dan tabel** yang akan digunakan. Untuk memilih tabel klik tanda **plus “+”** di samping database > **tekan tanda panah “>”** > **tekan Next**. Pada contoh : menggunakan database akademik dan tabel jual\_mobil.



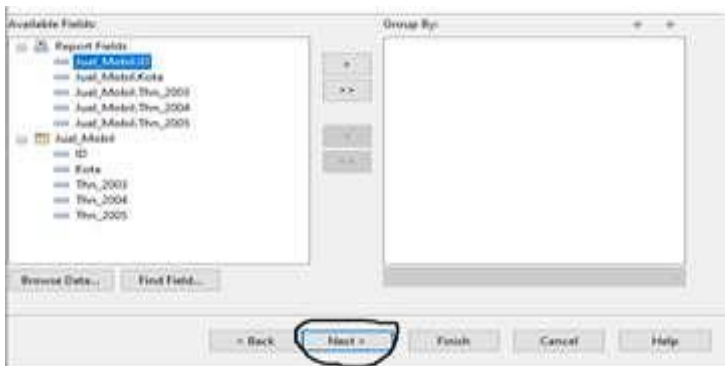
Gambar 8.8. Memilih Database dan Tabel

10. Pada kotak dialog **Filed**, **pilih kolom** yang akan ditampilkan pada laporan. Jika **seluruh kolom** ditampilkan di laporan lalu tekan tanda “>>” lalu klik **Next**. Contoh : seluruh field ditampilkan.



Gambar 8.9. Memilih Kolom/Field

11. Pada kotak dialog **Grouping**, memilih data yang ditampilkan dapat dikelompokkan berdasarkan kolom apa, misalnya kolom ID. Jika data tidak ingin digrupkan maka kotak dialog ini tidak perlu diisi, jika data ingin digrupkan maka **pilih salah satu atau semua semua kolom** lalu tekan tanda “>” atau “>>” lalu tekan **Next**.



Gambar 8.10. Memilih Grouping Field

12. Pada kotak dialog **Record Selection**, memilih data hasil pencarian diurutkan berdasarkan kolom apa. misalnya kolom ID. Jika data tidak ingin diurutkan maka kotak dialog

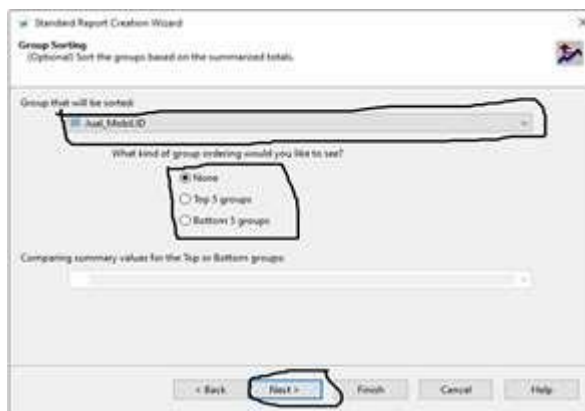


ini tidak perlu diisi, jika data ingin diurutkan maka **pilih salah satu atau semua semua kolom** lalu tekan tanda “>” atau “>>” lalu tekan **Next**.



Gambar 8.11. Memilih Record Selection

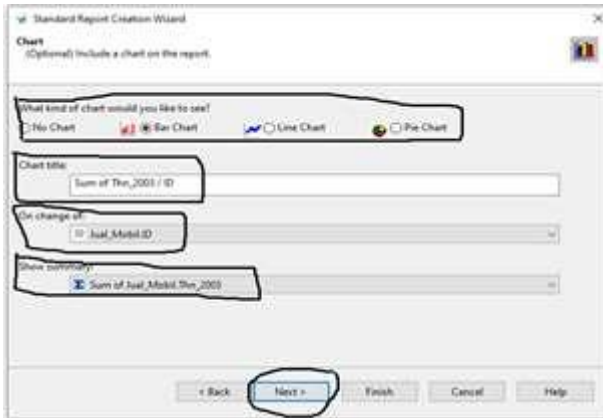
13. Pada kotak dialog **Group Sorting**, memilih pengurutan grup berdasarkan kolom apa, lalu atur jenis pengelompokan grup diatur (terdapat pilihan tidak diatur, kelompok 5 teratas, atau kelompok 5 terbawah), dan membandingkan nilai akhir. Lalu **klik Next**.



Gambar 8.12. Memilih Group Sorting

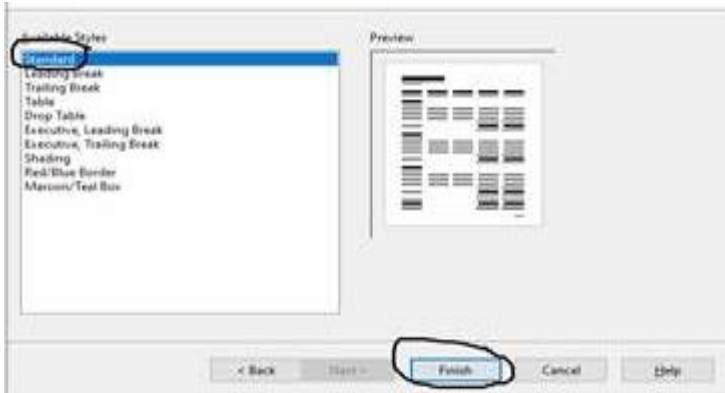
14. Pada kotak dialog **Chart**, memilih apakah laporan menggunakan grafik atau tidak, judul grafik, grafik

ditampilkan berdasarkan apa, dan ringkasan dari grafik yang ditampilkan. Lalu **klik Next**.



Gambar 8.13. Memilih Grafik

15. Muncul kotak dialog Record Selection, seperti pada gambar 8.11. Pilih salah satu kolom atau tidak sama sekali. Lalu **klik Next**.
16. Pada kotak dialog **Reports Style**, pilih model laporan yang akan ditampilkan. Pada contoh, pilih **Standard** lalu klik **Finish**.



Gambar 8.14. Memilih Record Style

17. Hasil laporan baru yang telah dibuat dapat dilihat. Jendela ini dapat dikatakan sebagai **Crystal Reports Designer**. Pada jendela ini dapat diatur tampilan dari laporan yang dibuat.



Gambar 8.15. Jendela Crystal Reports Designer

Pada jendela Crystal Reports Gallery disediakan tiga cara untuk mendesain dokumen reports baru yaitu :

#### a. Reports Wizard

Pilihan ini bertujuan untuk membuat dokumen reports baru yang kosong. Setiap langkah dalam membuat laporan akan dipandu dengan tampilan jendela yang interaktif. Terdapat beberapa pilihan template laporan, yaitu :

- **Standard.** Laporan yang dibentuk adalah laporan standar atau laporan umum.
- **Cross-Tab.** Laporan yang dibentuk adalah laporan berbentuk tabel grid. Laporan ini biasanya untuk ringkasan data.

- **Mail Label.** Laporan yang dibentuk adalah laporan dengan banyak kolom. Biasanya digunakan untuk membuat sheet dari label daftar alamat.

#### **b. Blank Report**

Pilihan ini bertujuan untuk membuat dokumen reports baru dan kosong.

#### **c. Existing Report**

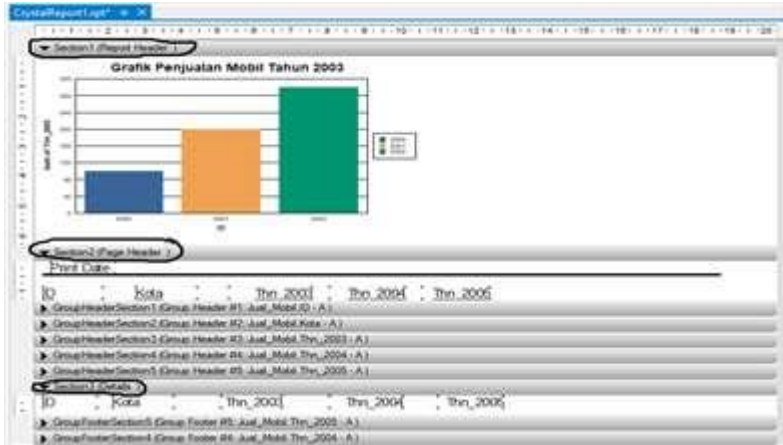
Pilihan ini bertujuan untuk membuat dokumen reports baru dengan menggunakan reports yang sudah ada/dibuat sebelumnya.

### **8.3. Crystal Reports Designer**

Crystal Reports Designer merupakan fasilitas pada visual basic .Net yang digunakan untuk menentukan sumber data laporan, memilih dan mengelompokkan record data yang ingin ditampilkan, dan untuk membuat format obyek serta layout laporan. Pada Crystal Reports Designer, terdapat bagian utama yang mendukung proses pembuatan laporan yaitu Reports Designer, Field Explorer, Toolbox, dan Properties.

#### **8.3.1. Reports Designer**

Reports designer digunakan untuk mendesain laporan Crystal Report. Tampilan Reports designer ditampilkan pada gambar 8.16.



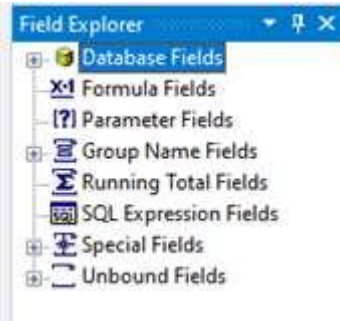
Gambar 8.16. Reports Designer

Pada Crystal Reports terdapat bagian-bagian yang membentuk laporan yaitu :

- **Reports header.** Bagian ini umumnya berisi judul laporan, logo perusahaan, dan informasi yang tercetak hanya pada awal laporan. Informasi di bagian ini hanya sekali ditampilkan pada bagian awal laporan.
- **Page header.** Bagian ini tampil di bagian atas (*header*) setiap halaman. Berisi informasi yang ingin dicetak pada bagian atas tiap halaman seperti judul kolom atau tanggal.
- **Details.** Bagian ini merupakan inti dari laporan / data yang akan ditampilkan. Berisi record-record yang ditampilkan.
- **Reports footer.** Bagian ini ditampilkan hanya sekali yaitu di bagian akhir (*footer*) suatu laporan. Bagian ini berisi informasi no halaman, nama pembuat laporan.
- **Page footer.** Bagian ini ditampilkan di bagian bawah di seluruh halaman. Berisi informasi yang hanya tercetak sekali pada bagian akhir laporan seperti grand total, summary.

- **Bagian reports tambahan.** Default dari laporan yang dibuat dengan Crystal Reports tidak menampilkan bagian reports tambahan. Bagian ini akan ditampilkan jika pada laporan ditambahkan pada suatu group, summary, atau sub total. Crystal Reports akan membuat dua bagian yaitu Group Header yang terletak di atas bagian Details dan Group Footer yang terletak di bawah bagian Details.

### 8.3.2. Field Explorer Crystal Report



Gambar 8.17. Field Explorer Crystal Report

Field explorer merupakan jendela yang menampilkan daftar semua field, antara lain :

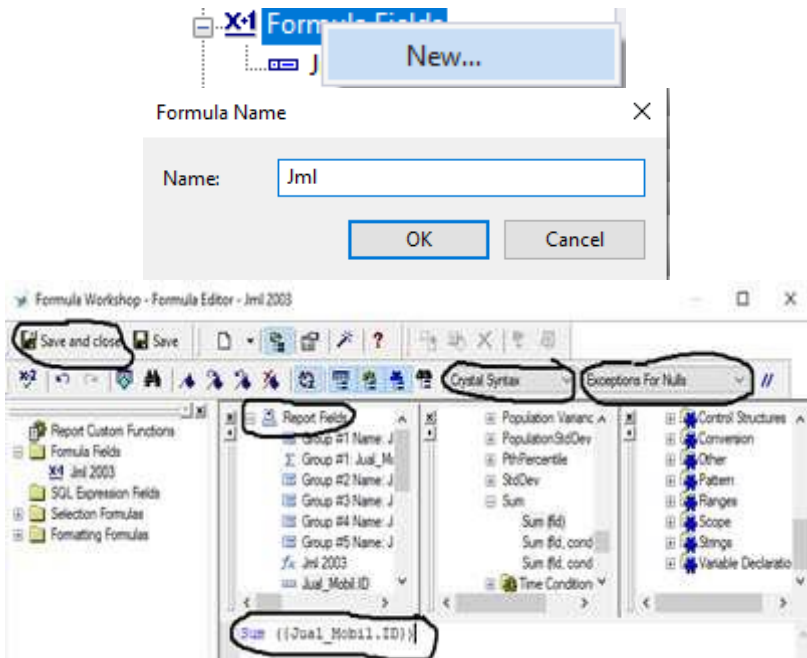
- **Database Fields**, pada bagian ini terdapat tanel-tanel yang akan digunakan pada laporan.

Untuk melihat nama-nama tabel yang aktif, lakukan klik ganda pada Database Fields atau klik pada tanda positif (+) yang ada pada Database Fields. Begitu juga jika ingin melihat field dari masing-masing tabel.

Pada bagian ini juga dapat menambahkan database baru yaitu dengan cara **klik kanan pada Database Fields dan pilih Add/Remove Database**. Lalu pilih data source yang

akan digunakan, caranya sama dengan saat menambahkan database baru pada menambahkan laporan baru.

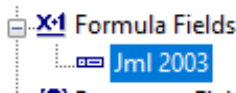
- **Formula Fields**, digunakan untuk menuliskan rumusan sesuai dengan kebutuhan. Contoh, membuat perhitungan. Untuk menambahkan formula yaitu dengan cara **klik kanan Formula Field > New > Isikan nama Formula > lalu isikan formula** dengan menambahkan crystal syntax, Expression, dan Report Fields. Formula yang ditambahkan akan ditampilkan di bagian bawah. Jika sudah selesai tekan **Save And Close**.



Gambar 8.18. Tambah Formula

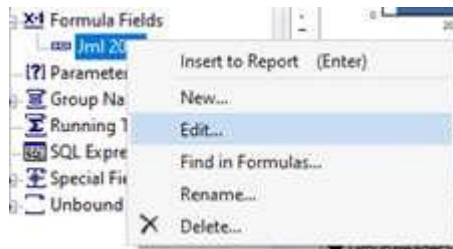
Hasil formula yang telah dibuat akan muncul di Field Explorer tepatnya di bawah Formula Fields. Untuk menggunakannya,

lakukan drag and drop formula dan letakkan Crystal Reports Designer.



Gambar 8.19. Hasil Formula Baru

Jika ingin mengedit formula yang telah dibuat maka tekan **klik kanan pada Formula Fields > Edit**. Maka akan muncul kptak dialog formula seperti gambar 8.18.



Gambar 8.20. Edit Formula

- **Parameter Fields**, digunakan untuk menampung nilai dari user yang digunakan pada laporan. Contoh, menampilkan teks sebagai judul laporan, pemilihan data tertentu (*filtering*), dan sebagainya.
- **Group Name Fields**, digunakan untuk pengelompokan data berdasarkan field tertentu.
- **Running Total Fields**, digunakan untuk menyelesaikan perhitungan yang datanya tidak semua berasal dari record, tetapi berdasarkan kondisi tertentu.
- **SQL Expression Fields**, digunakan untuk mengisi pernyataan SQL yang di tulis di SQL Expression editor
- **Special Fields**, digunakan untuk menampilkan informasi yang biasanya ada di dalam sebuah laporan, seperti tanggal, jam, no halaman, dan sebagainya.

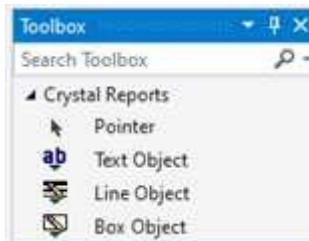


- **Unbound Fields**, digunakan untuk menyimpan data Boolean, numeric, string, atau yang tidak berhubungan dengan database manapun.

### 8.3.3. Toolbox Crystal Report

Crystal reports juga mempunyai toolbox seperti pada form, hanya saja disini tool yang disediakan hanya sedikit karena tool hanya digunakan untuk melengkapi desain laporan. Toolbox crystal reports dapat dilihat pada gambar 8.18. Jika ingin menggunakan tool pada toolbox yaitu dengan cara drag-and-drop tool yang akan digunakan.

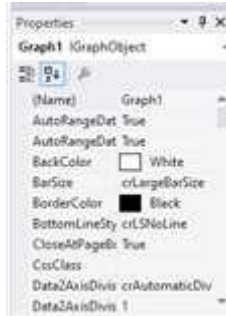
Berikut macam-macam toolbox crystal reports diantaranya Tool Text Object untuk menambahkan teks pada laporan, Tool Line Object untuk menambahkan garis pada laporan, dan Tool Box Object untuk menambahkan kotak pada laporan.



Gambar 8.21. Toolbox Crystal Reports

### 8.3.4. Properties Crystal Report

Properties crystal reports digunakan untuk mengatur masing-masing tool yang ada pada crystal reports, dapat dilihat pada gambar 8.22.

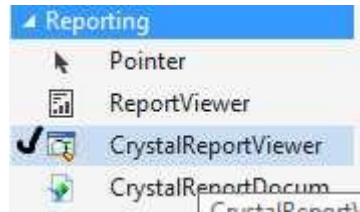


Gambar 8.22. Properties Crystal Reports

#### 8.4. Menampilkan Laporan Crystal Reports Pada Form

Laporan yang telah dibuat di crystal reports masih belum terlihat bentuk tampilan datanya. Untuk itu dibutuhkan kontrol lain yang dipasang di form yaitu **CrystalReportViewer**. Langkah-langkah yang dilakukan untuk menambahkan laporan pada form adalah :

1. Tambahkan tool CrystalReportViewer yang ada pada toolbox pada grup Reporting, lihat gambar 8.23.



Gambar 8.23. CrystalReportViewer pada Toolbox

2. Lalu atur properties CrystalReportViewer yaitu :
  - ReportSource → pilih nama crystal report yang akan digunakan. Contoh, pilih Crystal Report 1.
  - Klik **panah yang terletak di pojok kanan atas**, pada **Choose a Tool Panel View** pilih **None**. Hal ini akan membuat tampilan crystal reports menjadi lebar.

## 8.5. Menampilkan Laporan Crystal Reports Pada Form dengan Menambahkan Parameter

Jika ingin menampilkan laporan berdasarkan parameter tertentu maka harus ditambahkan tool untuk menampung parameter tersebut serta menambahkan kode program untuk menampilkan hasil laporan berdasarkan parameter yang diinginkan. Parameter dapat diatur dari Crystal Reports maupun dari form sendiri. Tetapi yang dibahas di subbab ini hanya menambahkan parameter dari form saja.

Sebagai contoh, dari laporan yang telah dibuat, ingin ditambahkan parameter kota. Jadi dapat ditampilkan laporan berdasarkan kota yang diketikkan user pada isian Kota. Untuk membuat itu, tambahkan tool label, textbox, dan button. Lalu tambahkan kode program pada tombol button yaitu :

```
'Untuk menampilkan form6 baru
Dim report As New Form6
'Variabel kota untuk menampung data yang dimasukkan user dari
textbox kota
Dim kota As String = txtKota.Text
'Crystal Report Formula : mengambil data dari field Kota dan
diseleksi dari textbox kota
report.CrystalReportViewer1.SelectionFormula = "{Jual_Mobil.Kota} =
'" & kota & "'"
'Menampilkan report baru di form baru
report.Show()
```

## 8.6. Model Implementasi Sumber Data

Dalam membuat suatu laporan, data merupakan komponen yang sangat vital dan mutlak disediakan. Umumnya data-data tersebut disimpan dalam sebuah database. Terdapat dua model untuk mengambil data yang ada dalam database yaitu :

- **Pull Mode**

Pada model ini proses yang terjadi adalah driver akan melakukan koneksi ke database dan menarik data yang ada di dalam database tersebut sesuai dengan permintaan. Dengan model ini, diantara koneksi dalam database dengan perintah SQL akan menghasilkan data yang ditangani oleh Crystal Reports. Umumnya model ini digunakan untuk laporan yang pengambilan datanya berasal dari sebuah database yang koneksinya tidak mengalami perubahan atau tidak memerlukan pengkodean. Contoh : koneksi database melalui OLE DB (ADO) yang ada di jendela Database Expert.



Gambar 8.24. Proses Pada Pull Mode [Yuswanto:2007]

- **Push Mode**

Pada model ini koneksi ke database digunakan untuk mengambil data dan mengisikan data tersebut ke dalam dataset. Data yang berada dalam dataset selanjutnya ditampilkan pada laporan. Push Mode merupakan model yang lebih fleksibel, karena perubahan koneksi database dapat dilakukan melalui kode-kode program. Dengan metode ini memungkinkan untuk membangun koneksi yang terbagi (*sharing*) ke dalam aplikasi dan membagi data sebelum Crystal Reports menerimanya. Contoh, koneksi database melalui ADO .Net

DataSets pada Project Data yang ada di jendela Database Expert.



Gambar 8.25. Proses Pada Push Mode [Yuswanto:2007]

## BAB IX

### MENU, MDI, PEMBUATAN SETUP, TOOLTIP

Pada bab 9 ini dijelaskan tentang cara membuat menu, MDI, dan setup. Pembuatan menu, MDI, dan Setup ini dibuat saat seluruh form pada suatu aplikasi telah selesai dibuat (sebagai *finishing*).

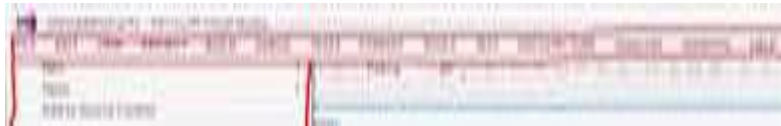
Tujuan dari pembuatan menu yaitu agar mudah mengakses form demi form yang ada pada aplikasi. Pembuatan MDI bertujuan agar user dapat membuka banyak form. Selain itu, form tetap terbuka (tidak hilang) walaupun user mengakses form lain. Pembuatan setup mempunyai tujuan agar aplikasi dapat diinstal/dipakai di berbagai komputer tanpa harus membuka visual basic terlebih dahulu. Sehingga aplikasi yang dibuat *user friendly*. Sedangkan pembuatan tooltip berfungsi untuk memberikan informasi suatu tool ke user jika pointer diarahkan pada tool tersebut.

#### 9.1. Menu

Menu digunakan untuk menampilkan pilihan-pilihan pada sebuah form. Keuntungan penggunaan menu untuk menghemat tempat dalam menampilkan pilihan. Sebagai gambaran menu, lihatlah menu yang ada di visual basic yaitu

terdapat menu file, edit, view, dan sebagainya. Masing-masing menu bisa mempunyai sub menu atau tidak. Misalnya pada menu file mempunyai sub menu new, open, dan lain-lain. Tampilan menu pada visual basic dapat dilihat pada gambar 9.1. berikut.

Pada gambar 9.1. terlihat tulisan di bagian atas yang diberi garis kotak berwarna merah dan berbentuk horisontal menunjukkan menu, sedangkan tulisan yang tersusun ke bawah dan diberi kotak warna merah dan berbentuk vertikal menunjukkan sub menu. Pada masing-masing menu dan sub menu terdapat shourcut.



Gambar 9.1. Menu pada Visual Basic

Menu yang terlihat pada gambar 9.1 dapat dibuat sendiri oleh pemakai dan diterapkan pada aplikasi yang dibuatnya. Untuk membuat menu, sub menu, dan shourcut pada suatu aplikasi yaitu dengan cara menyesuaikan form-form yang ada pada aplikasi dan kelompokkan form berdasarkan kriteria yang sesuai. Misalkan pada suatu aplikasi terdapat kumpulan form-form yang jika digolongkan menjadi beberapa kriteria yaitu master, transaksi, dan laporan. Maka dari penggolongan ini, jadikan kriteria ini sebagai menu. Kemudian untuk sub menu isilah dengan form-form yang ada pada aplikasi tersebut. Contoh menu master terdiri dari form anggota, form buku, dll. Menu transaksi terdiri dari form peminjaman, form pengembalian, dll. Menu laporan terdiri dari form laporan peminjaman, form laporan pengembalian.

Dalam pembuatan menu, selain menunya dapat menyesuaikan dengan aplikasi yang dibuat, menu juga bisa dibuat seperti yang ada pada menu visual basic (meniru menu yang ada pada suatu aplikasi). Hal ini diperbolehkan, dengan catatan bahwa letak menu, sub menu, dan shourcut harus sama dengan aplikasi lain atau standar pemakaian dalam berkomputer harus dipenuhi. Misalkan, pemakai ingin menggunakan menu file pada aplikasinya maka menu file ini harus berada di sebelah kiri dari menu yang lain, mempunyai shourcut alt+F, dan mempunyai sub menu New, Open, dll. Jika ini diganti, maka user akan bingung dan aplikasi yang dibuat kurang diminati oleh user karena menyebabkan kebingungan pada user.

Contoh pembuatan menu, sub menu, dan shourcut pada suatu aplikasi (misalkan aplikasi perpustakaan) yaitu :

1. Tambahkan form baru dan jadikan form baru ini sebagai form utama.
2. Tambahkan tool MenuStrip pada form Utama.
3. Tuliskan kata pada **Type Here** (lihat gambar 9.2). Letak tulisan Type Here yang horizontal untuk menunjukkan menu sedangkan letak tulisan Type Here yang vertikal untuk menunjukkan sub menu.

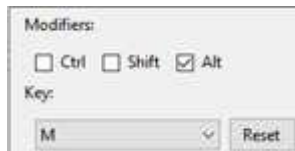


Gambar 9.2. Menu Suatu Aplikasi

4. Untuk memberi shourcut pada masing-masing menu dan sub menu yaitu dengan cara klik menu atau sub menu yang

akan diberi shourcut lalu modifikasi isian shourcut pada propertis menu atau sub menu yaitu properties ShourcutKeys (**ShourcutKeys**).

Pada properties ShourcutKeys, terdapat pilihan modifier (berisi Ctrl, Shift, dan Alt) dan keys (berisi huruf, angka, symbol, lain) yang dipilih oleh programmer. Misalkan programmer memberikan shourcut pada menu Master dan dan modifikasi isian properties ShourcutKeys yaitu dengan memilih modifier Alt dan Key "M". Maka pada saat program dijalankan dan user menekan Alt+M maka aplikasi akan menampilkan sub menu dari menu Master secara dropdownlist atau akan menjalankan perintah sesuai dengan saat user menekan tombol menu.



Gambar 9.3. Properties ShourcutKeys

5. Isikan menu dan sub menu sesuai dengan kebutuhan aplikasi.
6. Setiap menu dan sub menu ini juga dapat diberi kode program sesuai degan keinginan. Misalkan untuk sub menu Form Anggota diberikan perintah untuk memanggil form

lain.

```
Private Sub FormAnggotaToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles FormAnggotaToolStripMenuItem.Click
    Form2.Show()
End Sub
```

## 9.2. MDI



Pada aplikasi MDI (*Multiple Document Interface*) terdapat dua jenis form. Yang pertama adalah form MDI (form induk) dan yang kedua adalah form anak. Aplikasi MDI adalah aplikasi yang dapat menggunakan beberapa form sekaligus yang menginduk pada suatu form induk. Berikut contoh membuat form induk yaitu :

1. Bukalah form yang dijadikan form utama.
2. Pada properties form (form utama), ubah nilai **IsMdiContainer** menjadi **True**.
3. Isikan kode program pada semua menu atau sub menu yang diberikan perintah untuk memanggil form lain. Misalkan sub menu Form Anggota (lihat gambar 9.2) diberikan perintah untuk memanggil form 2, maka codingnya yaitu :

```
Private Sub FormAnggotaToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles FormAnggotaToolStripMenuItem.Click
    Form2.MdiParent = Me
    Form2.Show()
End Sub
```

Fungsi kode program ini yaitu agar saat user mengakses beberapa form yang berbeda, form yang tidak diakses tetap muncul (tidak hilang atau sembunyi).

4. Lakukan langkah no 3 hingga menu atau sub menu telah diberi coding. Kode program yang diberikan antar menu dan sub menu hamper sama, yang membedakan adalah form yang dipanggil itu form apa atau form berapa.
5. Jadikan form utama ini menjadi form yang pertama kali dijalankan caranya yaitu klik menu **Project > nama\_project Properties**. Gantilah **Starup Form** menjadi form yang dijadikan form utama.

### 9.3. Pembuatan Setup

Pembuatan setup aplikasi disebut dengan **deployment**. **Deployment** adalah cara untuk memaketkan aplikasi yang telah dibuat sehingga mudah untuk mendistribusikannya.

Pada VB.Net menggunakan project deployment untuk membuat file instalasi (.msi) dari aplikasi yang telah dibuat. Installer pada VB.Net menggunakan teknologi Microsoft Windows Installer untuk mengelola instalasi. Sebagai contoh, installer secara otomatis akan melakukan instalasi rolls back (membatalkan instalasi dan menghapus file yang sudah diinstal sehingga sama seperti sebelum instalasi dimulai) jika terjadi kesalahan menginstal.

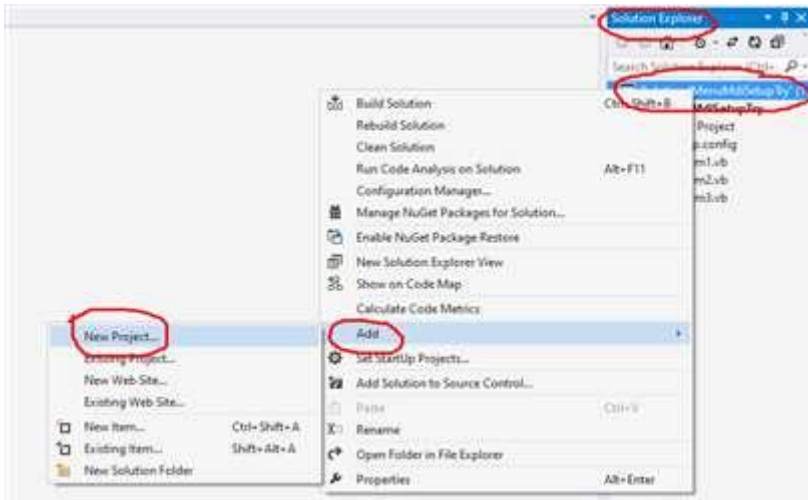
Sebelum melakukan deployment, siapkan beberapa hal berikut :

1. Lakukan instalasi Visual Studio Installer (subbab 9.3.3).
2. Siapkan file aplikasi VB.Net yang akan dibuat installer-nya. Pastikan program tidak ada error dan berjalan dengan baik dan lancar.
3. Buat file yang berisi informasi mengenai aplikasi, cara penginstalan aplikasi, cara penggunaan aplikasi, dan sebagainya. Biasanya file ini bernama Readme.txt atau **install.txt**. File ini berfungsi untuk memberi penjelasan kepada user jika akan melakukan penginstalan program. Letakkan file ini di folder debug, agar ikut dalam proses setup. File ini juga dicopy-kan ke folder hasil file setup (karena file ini tidak terlihat pada folder hasil setup).
4. File-file lain yang sekiranya dibutuhkan seperti database. Letakkan file ini di folder debug, agar ikut dalam proses setup.

Berikut contoh pembuatan deployment :

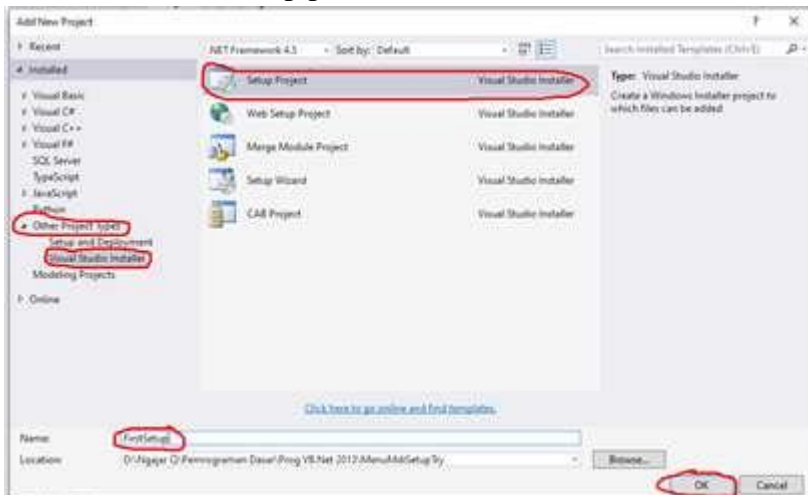
1. Buka file program yang akan dibuat instalasinya.

2. Klik kanan pada Solution Explorer lalu klik Add > New Project. Lihat gambar 9.4.



Gambar 9.4. Menambah Proyek Setup

3. Pada kotak dialog **Add New Project**, pilih **Other Project Types > Visual Studio Installer** dan pilih **Setup Project** lalu beri nama setup pada **Name** dan klik **OK**.



Gambar 9.5. Memilih Tipe Setup

4. Kemudian akan muncul kotak dialog **File System** (**nama\_Setup**) yang dapat dilihat pada gambar 9.6.



Gambar 9.6. File System

Pada kotak dialog **File System** tersebut terdapat tiga folder yaitu Application Folder, User Desktop, dan User's Programs Menu. Lalu atur isi dari ketiga folder tersebut.

- Pertama, atur isi dari folder Folder Application Folder. Isi dari folder ini berasal dari file yang berada di folder Bin > Debug pada program yang akan dibuat instalasinya. Caranya yaitu :  
Klik folder Application Folder yang berada di sebelah kiri kotak dialog (yang diberi lingkaran warna merah) yang dapat dilihat pada gambar 9.6.



Gambar 9.6. Petunjuk Klik Folder Application Folder

Setelah itu akan muncul tampilan seperti gambar 9.7. Area sebelah kiri berisi ketiga forlse yang telah disebutkan (pada gambar 9.7 ditunjukkan dengan no 1) dan area sebelah kanan Kotak dialog menjadi kosong (no 2).



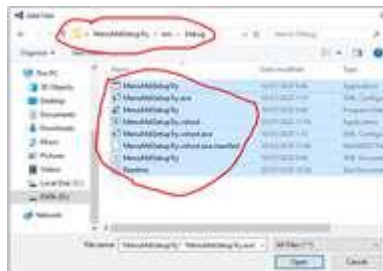
Gambar 9.7. Folder Application Folder Setelah Ditekan

Untuk mengisi Folder Application Folder, maka lakukan klik kanan pada area kosong Folder Application Folder yang berada di sebelah kanan (**klik kanan pada no 2**, lihat gambar 9.7) lalu pilih **File**.



Gambar 9.8. Langkah Pertama Mengisi Folder Application Folder

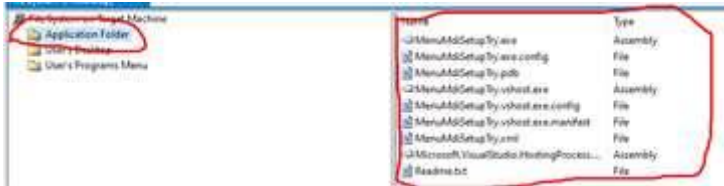
Lalu muncul kotak dialog seperti dilihat gambar 9.9.



Gambar 9.9. Kotak Dialog Add File Folder Application Folder

Pada kotak dialog tersebut, carilah folder yang menyimpan program yang akan dibuat instalernya. Lalu buka folder **Bin > Debug** (lihat coretan atas pada gambar 9.9). Lalu **blok semua file yang ada di folder debug > klik OK** (lihat coretan bawah pada gambar 9.9).

Hasil dari proses tersebut dapat dilihat pada gambar 9.10. Pada gambar tersebut isi dari Folder Application Folder merupakan file yang diambil dari folder Bin > Debug pada folder program.



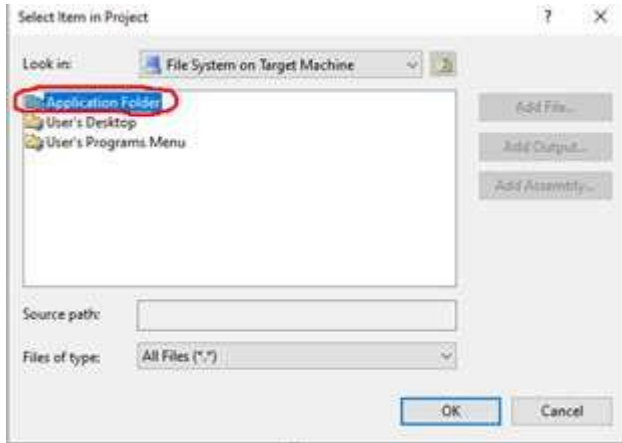
Gambar 9.10. Isi Folder Application Folder

- Selanjutnya, atur isi dari Folder User's Desktop. Isi dari folder ini mengambil dari folder Application Folder yaitu file yang berekstensi file .exe. Caranya adalah :  
**Klik folder Folder User's Desktop** (lihat lingkaran merah gambar 9.11 sebelah kiri) lalu **klik kanan di area sebelah kanan** lalu pilih **Create New Shourcut** (lihat lingkaran merah gambar 9.11 sebelah kanan).



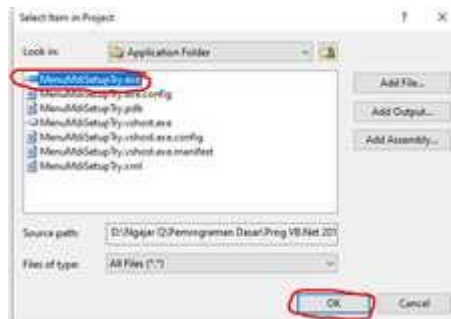
Gambar 9.11. Application Folder User's Desktop

Lalu muncul kotak dialog seperti gambar 9.12. Lalu **double klik pada Application Folder**.



Gambar 9.12. Kotak Dialog Select Item 1

Lalu pilih file dengan **ekstensi .exe**. Lalu pilih **OK**. Lihat gambar 9.13.



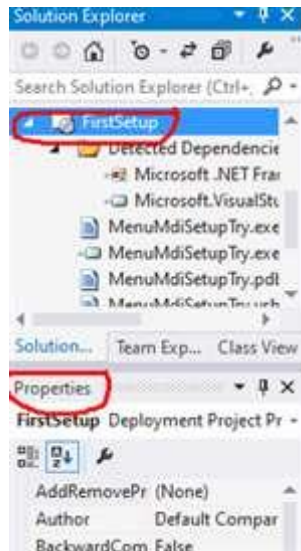
Gambar 9.13. Kotak Dialog Select Item 2

Hasil dari langkah di atas ditampilkan pada gambar 9.14. Pada gambar tersebut terlihat bahwa terdapat satu file .exe pada Folder User's Desktop.



Gambar 9.14. Isi Folder User's Desktop

- Terakhir, untuk folder User's Programs Menu, boleh diisi foldernya atau tidak. Jika diisi, langkahnya sama seperti mengisi folder User's Desktop.
5. Atur properties pada File Setupnya. Caranya klik pada file setup yang telah dibuat yang terletak pada Solution Explorer. Contoh, klik FirstSetup pada Solution Explorer. Lihat gambar 9.15.



Gambar 9.15. Properties FisrtSetup

Beberapa properties FirstSetup yang perlu diatur dapat dilihat pada tabel 9.1.

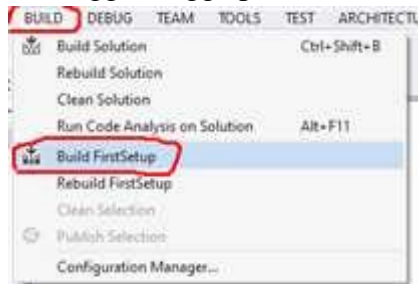
Tabel 9.1. Tabel Properties FirstSetup

Properties	Setting
Author	Isikan dengan pembuat program
Description	Isikan dengan deskripsi program
Manufacturer	Isikan dengan pembuat program
Target Platform	Isikan default : x86 atau diubah
Title	Isikan judul program



Version	Isikan default : 1.0.0 atau diubah
---------	------------------------------------

6. Kemudian klik menu **Build** > pilih **Build nama\_setup** (lihat gambar 9.16). Tunggu hingga proses selesai.



Gambar 9.16. Petunjuk Klik **Build** > **Build nama\_setup**

7. Jika sudah selesai proses setup maka akan tampil pesan sukses seperti pada gambar 9.17.



Gambar 9.17. Pesan Proses Berhasil

8. Proses setup telah selesai, lalu cek di folder program aplikasi. Terdapat folder baru bernama file setup yang dibuat. Contoh, FirstSetup. **Double klik folder tersebut > klik folder debug**. Dalam folder debug terdapat dua file setup yaitu file dengan ekstensi **.exe** dan **.msi**. Lihat gambar 9.18.



Gambar 9.18. Hasil Pembuatan File Setup

Jika kursor didekatkan ke file dengan ekstensi .exe maka akan muncul tooltip yang berisi hal-hal yang diatur pada properti file setup, lihat gambar 9.19. Pada gambar tersebut tooltip berisi author, description, dan sebagainya.



Gambar 9.19. Tooltip Pada File .Exe

### 9.3.1. Mencoba Instalasi Program Yang Telah Dibuat

Untuk mencoba hasil instalasi dari program yang telah dibuat yaitu dengan cara :

1. Buka Windows Explorer dan buka folder tempat penyimpanan proyek development (terletak pada folder program yang dibuat file setup-nya). Lalu **double klik pada folder Debug**. Double klik pada file yang berekstensi **.exe**. lalu mulai menginstal. Lalu klik **Next** hingga proses menginstal. Tunggu hingga proses menginstal selesai. Lalu klik **Close**.
2. Lihatlah **desktop**, maka akan muncul shortcut dari program yang diinstal dengan ekstensi .exe dengan **nama proyek** dari program yang dibuat.
3. Jika ditekan double klik pada shortcut tersebut maka program akan berjalan sesuai dengan program yang dibuat.

4. Jika shortcut bisa diinstal maka proses membuat file setup telah berhasil.

### 9.3.2. Uninstal Program Hasil Instalasi Program Yang Telah Dibuat

Untuk uninstall program yang telah dibuat dan diinstal, sama dengan uninstall program lainnya. Yaitu dengan cara ke **control panel** lalu cari file dengan **nama file setup** (bukan nama proyek) lalu **double klik pada file tersebut lalu klik uninstall**.

### 9.4. Instalasi Microsoft Visual Studio 2013 Installer

Untuk membuat setup pada visual basic .Net 2013 terdapat dua cara yaitu dengan berbayar (menggunakan InstallShield) dan gratis (menggunakan Visual Studio Installer). Cara yang dilakukan di atas menggunakan yang gratis / Visual Studio Installer.

Pada visual studio basic .Net 2013, Visual Studio Installer tidak langsung terinstal secara otomatis. Jadi harus mencari Visual Studio Installer 2013 di internet lalu mendownloadnya. Berikut disertakan alamat web untuk mendownload Visual Studio Installer (gambar 9.20). Lalu tekan download dan tunggu hingga proses selesai.

<https://marketplace.visualstudio.com/items?itemName=UnniRaviNdranathan-MSFT.MicrosoftVisualStudio2013InstallerProjects>.



Gambar 9.20. Situs Visual Studio Installer

Setelah selesai mendownload, bukalah hasil download tersebut, lalu instalasi hingga berhasil. Pada saat melakukan instalasi program visual basic .Net 2013 harus ditutup. Jika sudah berhasil proses instalasinya maka akan ada pesan “succes”. Lalu cek di visual basic (seperti gambar 9.5), barulah terdapat program Visual Studio Installer.

## 9.5. Pembuatan Tooltip

Tooltip merupakan sebuah tool yang disediakan pada visual basic .Net. Fungsi dibuatnya tooltip adalah untuk memberikan informasi kepada user mengenai suatu tool. Cara untuk melihat informasi yang ditampilkan pada tooltip yaitu dengan mengarahkan pointer ke tool yang akan dilihat informasinya, maka akan terlihat informasi yang ditampilkan di tooltip.

Tooltip ini merupakan salah satu tool penting yang harus ada pada program aplikasi apabila programmer hendak mendistribusikan program yang dibuatnya. Hal ini agar user dapat mengerti fungsi dari tombol atau tool yang ada di program aplikasi tersebut. Berikut tampilan tooltip pada windows explorer.



Gambar 9.20. Contoh Tooltip

Pada gambar 9.20 terlihat bahwa terdapat tooltip dari suatu folder “Program”. Tooltip tersebut berisi kapan pembuatan folder tersebut, ukuran folder, dan keterangan file apa saja yang berada pada folder tersebut.

Berikut contoh pembuatan tool pada visual basic. Bukalah Microsof Visual Basic dan buatlah satu form. Berikut design yang ditampilkan.



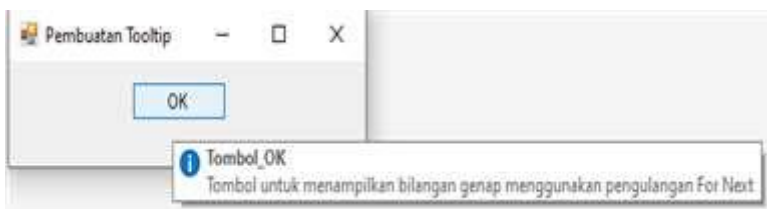
Gambar 9.21. Design Form Pembuatan Tooltip

Tool dan properties yang digunakan pada form tersebut adalah :

Tabel 9.2. Tabel Tool dan Properties Form Pembuatan Tooltip

Tool	Properties	Keterangan
Button 1	Text	OK
	Tooltip On ToolTip1	Tombol untuk menampilkan bilangan genap menggunakan pengulangan For Next
ToolTip1	ToolTipIcon	Info
	ToolTipTitle	Tombol_OK (berisi judul dari tool yang diberi tooltip). (jangan menggunakan spasi, ganti dengan symbol garis bawah ( _ ))

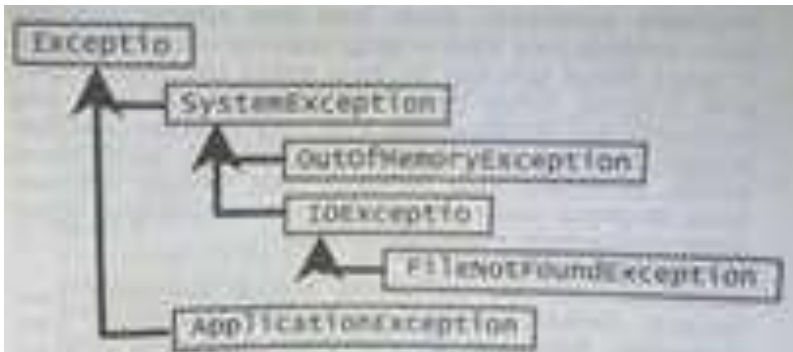
Berikut hasil program pembuatan tooltip ditampilkan pada gambar 9.21. Pada gambar tersebut terdapat tooltip jika pointer diarahkan ke tombol OK.



Gambar 9.21. Hasil Form Pembuatan Tooltip

## BAB X MENANGANI KESALAHAN PROGRAM

Sebelum membahas penanganan kesalahan, perlu dijelaskan exception. Exception merupakan kesalahan kondisi atau perilaku tak terduga yang terjadi sepanjang pelaksanaan dari suatu program dan memerlukan penanganan dari luar agar alur programnya normal.



Gambar 10.1. Gambar Menu Obyek Exception[Yuswanto:2006]

Pada Microsoft .Net Framework, exception adalah object yang menerima warisan dari kelas dasar exception. Kemudian .Net Framework menggambarkan bidang kelas exception. Microsoft .Net Framework menyediakan obyek exception, seperti pada gambar 9.1 (Yuswanto, 2006).

Visual basic .Net memiliki cara penanganan kesalahan yang terstruktur saat program dijalankan. Istilah penanganannya

sering disebut dengan Structure Exception Handling. Dengan kemampuan yang dimiliki oleh visual basic .Net, seorang pembuat program akan dapat dengan mudah menemukan letak kesalahan yang terjadi dan sesegera mungkin untuk memperbaikinya.

Visual basic .Net mempunyai tiga macam kesalahan yang mungkin muncul ketika mengembangkan suatu aplikasi. Ketiga kesalahan tersebut antara lain :

### **1. Kesalahan sintaks**

Kesalahan sintaks (kesalahan tata tulis) merupakan kesalahan yang terjadi akibat melanggar aturan penulisan yang sudah ditetapkan oleh visual basic .Net. Kesalahan tersebut antara lain salah ejaan pada kata kunci atau nama variabel, kesalahan meletakkan tanda baca, dan adanya pasangan suatu perintah yang tidak lengkap (misalnya penulisan fungsi IF tanpa diikuti dengan End IF).

Pada waktu menuliskan kode-kode program, visual basic .Net akan mengecek kode program yang telah diketik dan jika ditemukan kode yang memiliki kesalahan sintaks, yang ditandai dengan garis gelombang berwarna merah atau biru (biasa disebut cacing). Dengan demikian penulis program langsung mengetahui adanya kesalahan dan diharapkan segera dapat memperbaiki kesalahan tersebut.

### **2. Kesalahan run time**

Kesalahan run time merupakan bentuk kesalahan yang muncul ketika kode-kode program dikompilasi atau dijalankan. Kesalahan ini disebabkan kode-kode program yang kelihatannya benar oleh komputer, tetapi tidak benar saat dijalankan. Misalnya terjadi pembagian bilangan dengan angka nol.

### 3. Kesalahan logika

Kesalahan logika merupakan kesalahan pemrograman yang memberikan hasil yang tidak sesuai dengan yang diharapkan atau tidak terduga. Bentuk kesalahan seperti ini akan sulit dilacak dan diperbaiki karena kesalahan ini tidak menghentikan jalannya program. Misalnya, jika nilai akhir (NA) lebih besar atau sama dengan 85 maka nilai hurufnya "A". kode program berikut ini salah satunya yaitu **If NA 85 Then** seharusnya **If NA >= 85 Then**.

#### 10.1. Menggunakan On Error

Percabangan **On Error** digunakan untuk penanganan kesalahan (error) yang rutin dan menetapkan hasil kesalahan tersebut dalam suatu prosedur. Bentuk umum dari penulisan perintah On Error seperti berikut :

**On Error {GoTo [line | 0 | -1] | Resume Next}**

Keterangan :

- **GoTo Line**

Pernyataan GoTo yang diikuti oleh baris, menyebabkan jika terjadi kesalahan rutin maka eksekusi program akan diarahkan ke nomor baris yang tertera setelah perintah GoTo. Argumentasi dari baris adalah nama dari garis label atau garis nomor.

- **GoTo 0**

Pernyataan GoTo yang diikuti oleh angka 0, menyebabkan kesalahan yang terjadi akan diabaikan dan membuang nilai kesalahan yang ada.



- **GoTo -1**

Pernyataan GoTo yang diikuti oleh angka -1, menyebabkan munculnya suatu exception, tetapi langsung diabaikan dan membuang nilai kesalahan yang ada.

- **Resume Next**

Pernyataan Resume Next menyebabkan kesalahan pada program diabaikan dan proses eksekusi program dilanjutkan ke baris berikutnya.

Visual basic .Net menyediakan tiga macam bentuk penulisan dari **On Error** yaitu :

- **On Error Resume Next**

Pemasangan percabangan ini menyebabkan jika terjadi kesalahan, program akan mengabaikan kesalahan tersebut dan melanjutkan ke perintah di bawah perintah yang salah. Bentuk percabangan ini disarankan pada saat program mengakses suatu obyek program.

Contoh :

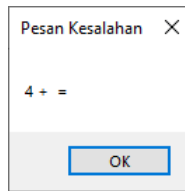
Buatlah suatu tombol Button dan berikan perintah tombol tersebut agar menjumlahkan dua variabel (variabel C dan D) dan hasilnya juga ditampung pada suatu variabel (variabel hasil). Variabel C berisi angka dan variabel D berisi string kosong. Variabel hasil berisi penjumlahan antara variabel C dan D. tambahkan juga pesan untuk menampilkan variabel C, D, dan hasil. Tambahkan juga kode program “On Error Resume Next” untuk pengecekan kesalahan.

Kode programnya :

```

Private Sub btnOnErrorResumeNext_Click(sender As Object, e As EventArgs)
Handles btnOnErrorResumeNext.Click
    Dim C, D, hasil
    On Error Resume Next
    C = 4
    D = ""
    hasil = C + D
    MessageBox.Show(C & " + " & D & " = " & hasil, "Pesan Kesalahan")
End Sub

```

Hasil program :

Gambar 10.2. Hasil Program On Error Resume Next

Pada gambar tersebut terlihat bahwa angka 4 ditambahkan dengan string kosong hasilnya tidak ada. Hal ini karena angka 4 merupakan data integer dan string kosong berupa data string. Sehingga data integer ditambahkan data string akan menghasilkan error. Tetapi karena dalam program disertakan kode program “On Error Resume Next” maka program mengabaikan kode yang error tersebut sehingga hasilnya kosong.

- **On Error GoTo (Baris)**

Pemasangan percabangan ini menyebabkan penanganan kesalahan (*Error Handling*) menjadi aktif, sehingga bila terjadi kesalahan maka program akan mencabang ke suatu baris atau label. Letak dari baris atau label yang dituju harus dalam prosedur yang sama.

### Contoh :

Buatlah suatu tombol Button dan berikan perintah tombol tersebut agar menjumlahkan dua variabel (variabel C dan D) dan hasilnya juga ditampung pada suatu variabel (variabel hasil). Variabel C berisi angka dan variabel D berisi string kosong. Variabel hasil berisi penjumlahan antara variabel C dan D. tambahkan juga pesan untuk menampilkan variabel C, D, dan hasil. Tambahkan juga kode program “On Error GoTo “label”” untuk pengecekan kesalahan. Untuk penulisan penanganan menggunakan On Error GoTo <Baris> yaitu dengan cara menambahkan baris dengan nama variabel yang tertulis di belakang kode program “On Error GoTo ” dan diikuti dengan tanda titik dua (:). sebagai contoh, “SALAH : ”. Penulisan ini ditulis di akhir program dan masih di dalam prosedur tombol button. Kemudian diikuti dengan pesan yang merupakan aksi selanjutnya dalam menangani kesalahan program yang berisi pesan kesalahan program, contoh isi pesan yaitu “prosesnya salah”. Lakukan proses tersebut pada tool Button dengan event click.

### Kode program :

```
Private Sub btnOnErrorGoTo_Click(sender As Object, e As EventArgs) Handles
btnOnErrorGoTo.Click
    Dim C, D, hasil
    On Error GoTo SALAH
    C = 4
    D = ""
    hasil = C + D
    MessageBox.Show(C & " + " & D & " = " & hasil, "Pesan Kesalahan")
    SALAH:
    MessageBox.Show("Prosesnya Salah", "Pesan Kesalahan")
End Sub
```

Hasil program :



Gambar 10.3. Hasil Program On Error GoTo <Baris>

Pada gambar tersebut terlihat bahwa angka 4 ditambahkan dengan string kosong hasilnya tidak ada. Hal ini karena angka 4 merupakan data integer dan string kosong berupa data string. Sehingga data integer ditambahkan data string akan menghasilkan error. Tetapi karena dalam program disertakan kode program “On Error GoTo <Baris>” maka program mengabaikan kode yang error tersebut sehingga hasilnya kosong.

- **On Error GoTo 0 Atau On Error GoTo -1**

Pemasangan percabangan ini menyebabkan perangkat kesalahan yang sudah dipasang menjadi tidak aktif (membatalkan penanganan kesalahan).

Contoh :

Buatlah suatu tombol Button dan berikan perintah tombol tersebut agar menjumlahkan dua variabel (variabel C dan D) dan hasilnya juga ditampilkan pada suatu variabel (variabel hasil). Variabel C berisi angka dan variabel D berisi string kosong. Variabel hasil berisi penjumlahan antara variabel C dan D. tambahkan juga pesan untuk menampilkan variabel C, D, dan hasil. Tambahkan juga kode program “On Error

GoTo 0” atau “On Error GoTo -1” untuk pengecekan kesalahan. Lakukan proses tersebut pada tool Button dengan event click.

Kode program :

```
Private Sub btnOnErrorGoTo0_Click(sender As Object, e As EventArgs) Handles
btnOnErrorGoTo0.Click
    Dim C, D, hasil
    On Error GoTo 0
    C = 4
    D = ""
    hasil = C + D
    MessageBox.Show(C & " + " & D & " = " & hasil, "Pesan Kesalahan")
End Sub
```

Hasil program :



Gambar 10.4. Hasil Program On Error GoTo 0/-1

Pada gambar tersebut untuk menampilkan kesalahan error pada program. Tampilan pesan yang ditampilkan berbeda dengan tampilan pesan biasanya yang memanggil class MessageBox atau MsgBox.

## 10.2. Komponen Error Provider

Pemakaian komponen error provider sebenarnya bukan untuk melakukan kontrol kesalahan tetapi untuk validasi inputan.

Namun, karena efek yang dihasilkan bertujuan untuk mencegah kesalahan, maka hal ini dibahas di bab ini.

Visual basic .Net menyediakan Komponen Error Provider bertujuan untuk mengatur pesan kesalahan pada setiap kontrol dalam form dengan cara menampilkan tooltip ketika isia tidak valid. Validitas dari inputan ditunjukkan Komponen Error Provider dengan cara menampilkan icon kecil berwarna merah berbentuk tanda seru (!) di dalamnya. Tooltip yang dimaksud akan tampil saat pointer mouse diarahkan ke icon tersebut. Letak dari icon tersebut berada di sebelah kanan dari tool yang di cek validasinya.

**Event** yang digunakan adalah **validating** yaitu **untuk melakukan pengecekan (validasi) data saat pointer meninggalkan kontrol isian**. Sedang bentuk pemasangan komentar suatu tool dan isi dari tooltip sebagai berikut.

**ErrorProvider1.SetError(kontrol, "komentar").**

Beberapa langkah untuk menggunakan tool Error Provider yaitu :

1. Tambahkan tool Error Provider di design form.
2. Tambahkan kode program di tool yang akan diberikan validasi data (tool yang biasa digunakan yaitu tool yang berupa tool isian / tool yang digunakan untuk memasukkan data, misalnya, tool textbox).
3. Ganti **event** tool tersebut menjadi **validating** (contoh : Private Sub textbox1\_validating).
4. Dalam kode program tersebut buatlah percabangan sederhana **If...Then...Else** (percabangan dapat ditingkatkan sesuai kebutuhan).

Pada kondisi di percabangan isilah dengan pengecekan tool, misalnya data yang diisikan tidak boleh kosong dan inputan yang dimasukkan harus berupa data numerik. Contoh :

```
If Trim(TextBox1.Text) = "" Or Not
IsNumeric(TextBox1.Text) Then
```

Pada rumus menggunakan fungsi trim karena untuk menghilangkan spasi diantara data yang dimasukkan.

Pada ekspresi pertama, tuliskan kode pemasangan komentar suatu tool dan isi dari tooltip (lihat bentuk pada penjelasan sebelumnya). Contoh :

```
ErrorProvider1.SetError(TextBox1, "Input tidak boleh
kosong dan harus angka")
```

Lalu di bawah kode program di atas tuliskan kode program agar sistem komputer tidak bekerja sebelum kesalahan dibetulkan. Berikut codingnya :

```
e.Cancel = True
```

Tulis kode program menormalkan isi dari tooltip. Berikut contoh codingnya :

```
ErrorProvider1.SetError(TextBox1, "")
```

### Contoh program :

Buatlah program sederhana yang berisi tool label dengan text “Jumlah Bilangan”, textbox1, Button1 dengan text “OK”, dan ErrorProvider1.

Program ini sebagai latihan untuk menampilkan validasi data pada tool textbox1. Tool textbox1 harus diisi data numerik dan tidak boleh kosong, jika textbox1 tidak maka akan muncul validasi data. Lakukan proses tersebut pada tool Textbox dengan event validating.

### Kode program :

Lalu tuliskan kode pada tool **textbox1** dengan **event validating**, sebaga berikut :

```
Private Sub TextBox1_Validating(sender As Object, e As
System.ComponentModel.CancelEventArgs) Handles TextBox1.Validating
If Trim(TextBox1.Text) = "" Or Not IsNumeric(TextBox1.Text) Then
    ErrorProvider1.SetError(TextBox1, "Input tidak boleh kosong dan harus
angka")
    e.Cancel = True
Else
    ErrorProvider1.SetError(TextBox1, "")
End If
End Sub
```

Hasil program :

Hasil program ditampilkan pada gambar 9.5.



Gambar 10.5. Validasi Inputan Angka dengan ErrorProvider

### 10.3. Menggunakan Try-Catch-Finally

Visual basic .Net memperkenalkan sebuah bentuk dari struktur pernyataan untuk menangani kesalahan yang dikenal dengan nama **Try-Catch -Finally – End Try**. Pernyataan ini ditujukan untuk mengeliminasi penggunaan pernyataan On Error Go To.

Penggunaan Try-Catch -Finally – End Try bertujuan untuk melindungi blok kode program yang mempunyai potensi untuk terjadinya suatu kesalahan. Bentuk umum dari penulisan pernyataan Try-Catch yaitu :

**Try**

**<Program Logic>**

**Catch <Exception Handling>**

**<Message>**



## End Try

### Keterangan :

- Program logic berisi kode-kode program yang diseleksi dan jika terjadi kesalahan langsung menuju ke catch.
- Exception handling merupakan kode-kode program yang menangani kesalahan yang terjadi.
- Message merupakan pesan yang dimunculkan jika terdapat kesalahan yang ditangkap oleh Exception Handling.

Proses yang dilakukan oleh pernyataan Try-Catch adalah jika terjadi kesalahan, pernyataan ini akan mengembalikan pesan kesalahan yang berasal dari Exception Class dan turunannya. Dari information Exception Class ini dapat dievaluasi jenis kesalahan yang lebih terperinci. Jika ingin mengabaikan jenis kesalahan yang muncul, tuliskan pernyataan pada Catch berikut :

### **Catch When Err.Number<>0**

Selanjutnya tampilkan pesan kesalahan berikut ini.

### **MessageBos.show(Err.Description)**

Visual basic .Net menyediakan dua kategori untuk Exception Class yaitu :

1. SystemException yang berisi Exception yang telah tersedia dari awalnya (*Pre-defined Common Language Runtime Exception Classes*).
2. ApplicationException yang merupakan jenis penanganan yang dibuat oleh pembuat program

Pernyataan Exception Class dengan penanganan kesalahan Try-Catch lebih sesuai diterapkan pada pemrograman berorientasi obyek (OOP), karena memberi pesan kesalahan berupa baris pernyataan yang mudah dimengerti. Selain itu proses penanganan Exception Class dapat mengurutkan proses kesalahan

sampai ke titik awal proses kesalahan sampai ke titik awal proses dijalankan (*StackTrace*).

### Contoh 1 :

Lakukan pengecekan dari suatu kesalahan dari proses pembagian suatu bilangan dengan nol. Gunakan pernyataan kesalahan dan pesan yang ditampilkan secara umum. Lakukan proses tersebut pada toll Button dengan event click.

### Kode program 1 :

```
Private Sub btnTryCatch1_Click(sender As Object, e As EventArgs) Handles  
btnTryCatch1.Click  
    Dim X As Integer  
    Dim Y As Integer = 3  
    Try  
        X = Y / 0  
    Catch When Err.Number  
        MessageBox.Show(Err.Description)  
    End Try  
End Sub
```

### Hasil program 1 :



Gambar 10.6. Pesan dari Hasil Pengecekan Kesalahan

Suatu bilangan tidak akan bisa dibagi dengan angka nol, hasilnya akan error. Jika kode program menggunakan Try-Catch dalam menangani kesalahan maka pesan error ditampilkan pada pesan yang ditampilkan pada gambar 9.6. Jika tidak menggunakan Try-Catch maka program akan berhenti dengan sendirinya dan juga

terdapat pesan error dari visual basic. Hanya saja, jika pesan error ditampilkan pada kotak pesan, maka hal ini tidak akan membuat user bingung dikarenakan program berhenti dengan sendirinya.

Pada pernyataan Try-Cath dapat diikuti oleh pernyataan Finally. Setelah pernyataan Finally akan dijalankan kode program yang terjadi setelah pernyataan Catch selesai dijalankan. Dengan kata lain kode program setelah pernyataan Finally akan dikerjakan tanpa melihat kondisi pada pernyataan Try, benar atau salah.

### Contoh 2 :

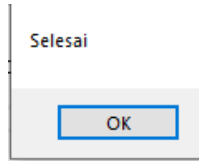
Lakukan pengecekan dari suatu proses perhitungan Aritmatika antara variabel dengan tipe data numerik dengan string. Pesan kesalahan Catch diambil dari sistem komputer. Pada pernyataan Finally, tambahkan pesan "Finish".

### Kode program 2 :

```
Private Sub btnTryCatch2_Click(sender As Object, e As EventArgs) Handles
btnTryCatch2.Click
    Dim X As Integer
    Dim Y As Integer = 3
    Try
        X = Y / 0
    Catch When Err.Number
        MessageBox.Show(Err.Description)
    Finally
        MessageBox.Show("Selesai")
    End Try
End Sub
```

### Hasil program 2 :





Gambar 10.7. Pesan dari Pernyataan Cath dan Finally

Jika menggunakan pernyataan Try-Catch-Finally terdapat dua pesan errornya yaitu pesan error dari pernyataan Catch dan pesan error dari pernyataan Finally, seperti gambar 10.7.

### Contoh 3 :

Lakukan pengecekan file "Data.txt" sedang dibuka atau tidak. Pada pernyataan Catch tambahkan Exception "FileNotFoundException" dengan pesan "File Data.txt tidak ada". Lakukan proses tersebut pada tool button dan event click.

### Kode program 3 :

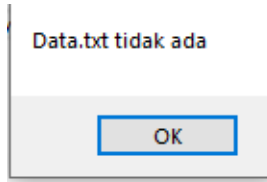
Tuliskan kode program ini di bagian atas (di luar prosedur form) :

```
Imports System.IO
```

Lalu tuliskan kode program berikut di tool button dan event click :

```
Private Sub btnTryCatch3_Click(sender As Object, e As EventArgs) Handles
    btnTryCatch3.Click
    Dim fs As FileStream
    Try
        fs = New FileStream("Data.txt", FileMode.Open)
    Catch ex As FileNotFoundException
        MessageBox.Show("Data.txt tidak ada")
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
```

### Hasil program 3 :



Gambar 10.8. Pesan Dari Kondisi File Tidak Ada


#### 10.4. Menggunakan Debugger

Fasilitas debugging akan membantu programmer untuk mendiagnosa kesalahan program dengan menguji aliran program saat dijalankan. Untuk mendukung kegiatan tersebut, visual basic .Net menyediakan suatu toolbar yang bernama **Debug**. bentuk dari toolbar dapat dilihat pada gambar.

Tabel berikut ini menggambarkan rincian dari bagian-bagian dan fungsi dari toolbar debug, yaitu :

Tabel 10.3. Tabel Bagian Toolbar Debug

Tombol	Nama	Fungsi
	<b>Start / Continue (F5)</b>	Untuk menjalankan aplikasi form (Sub Main). Pada mode Break, akan diubah menjadi Continue.
	<b>Break All (Ctrl + Alt + Break)</b>	Untuk menghentikan seluruh proses pengujian aliran program, saat dilakukan proses Debug.
	<b>Stop Debugging (Shift + F5)</b>	Untuk menghentikan eksekusi program dan kembali ke model design.
	<b>Restart (Ctrl + Shift + F5)</b>	Untuk mengulang proses pengujian aliran program mulai dari awal lagi yaitu pengecekan kode pada region.
	<b>Step Into (F11)</b>	Untuk memeriksa kode program selanjutnya. Jika jalannya program harus membaca prosedur, pengecekan dilakukan pada baris kode program di prosedur dan hasilnya ditampilkan.
	<b>Step Over (F10)</b>	Untuk memeriksa kode program selanjutnya. Jika jalannya program harus membaca prosedur, prosedur tersebut tidak dibaca tetapi hasilnya ditampilkan.

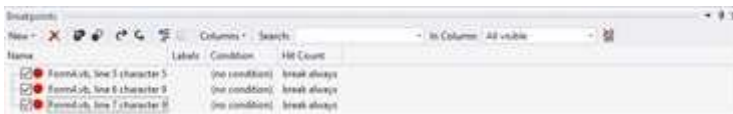
	<b>BreakPoints (Ctrl + Alt + B)</b>	Untuk mengaktifkan jendela Breakpoint. Pada jendela ini terdapat banyak menu lainnya.
---	-------------------------------------	---

## 10.5. Melakukan Proses Debug

Sebelum melakukan proses debug pastikan bahwa kode-kode programnya sudah diberi tanda **Breakpoint**. Jika belum maka proses debug akan dilakukan secara umum yaitu kontrol atau obyek yang dapat diproses dan mempunyai TabIndex 0 (nol).



Gambar 10.9. Pemasangan Breakpoint

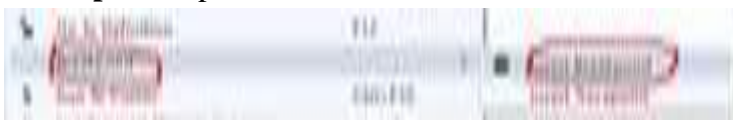


Gambar 10.10. Informasi Kode program Breakpoint

Breakpoint merupakan symbol titik yang tebal (.) pada kode program yang menandakan awal dari kode program yang akan didebug. Sedang posisi yang sedang didebug ditandai dengan *Execution Indicator* (|) berwarna kuning.

Terdapat tiga macam cara untuk mengaktifkan dan mematikan Breakpoint yaitu :

1. Lakukan **klik kanan** pada jendela editor visual basic .Net tepat dimana baris kode program akan di debug > pilih **Breakpoint** > pilih **Insert BreakPoint**.



Gambar 10.11. Menu Shourcut Insert Breakpoint

2. Tempatkan pointer pada baris kode program yang akan di debug dan ditekan tombol **F9** dari tombol kunci keyboard.
3. Lakukan klik kanan mouse pada baris kode program dan klik menu **Insert Breakpoint**, seperti gambar 9.11.

Langkah-langkah untuk melakukan proses debug pada kode-kode program visual basic .Net, sebagai berikut :

1. Lakukan breakpoint pada pada kode program sebagai awal baris yang akan di debug.
2. Selanjutnya tekan tombol kunci **F5** untuk melalui proses debug. Jika tidak hafal dengan tombol kunci, aktifkan toolbar debug, klik **Debug**. Pada toolbar Debug klik tombol **Start**. Kode program yang pertama kali dilakukan debug adalah kode program pada #Region.
3. Klik tombol **Step Into** atau tekan tombol kunci **F11** untuk melakukan debug kode program pada baris berikutnya.
4. Jika proses debug dilakukan terus-menerus sampai seluruh kode program dikerjakan, proses akan berakhir dengan menampilkan hasil (*output*) dari kode program tersebut.

**DAFTAR PUSTAKA**

- Hadi, Rahardian. 2004. *Membuat Laporan dengan Crystal Reports 8.5 dan Visual Basic 6.0*. Jakarta : Elex Media Komputindo.
- Hidayatullah, Priyanto. 2015. *Visual Basic .Net Membuat Aplikasi Database dan Program Kreatif*. Bandung : Informatika.
- Kusumo, Ario Suryo. 2000. *Buku Latihan Microsoft Visual Basic 6.0*. Jakarta : Elex Media Komputindo.
- Komputer, Wahana. 2005. *Pemrograman Visual Basic .Net 2005*. Yogyakarta : Andi.
- Microsoft. 2018. *Visual Studio 2013 System Requiement*.  
<https://docs.microsoft.com/en-us/visualstudio/productinfo/vs2013-sysrequirements-vs#--visual-studio-2013>.
- Rusmawan, Uus. 2012. *Koleksi Program VB.Net untuk Tugas Akhir dan Skripsi*. Jakarta : Elex Media Komputindo.
- Sianipar, R.H. 2008. *Pemrograman Visual Basic .Net untuk Pemula*, Jakarta : Cerdas Pustaka.
- Sianipar, R.H. 2014. *Pemrograman Visual Basic .Net*. Bandung : Informatika.
- Sianipar, R.H. 2017. *Pemrograman Visual Basic .Net untuk Pemula*. Bandung : Informatika.
- Subari dan Yuswanto. 2008. *Panduan Lengkap Pemrograman Basic 6.0*, Jakarta : Cerdas Pustaka.



Suhendra, Egar. 2017. *Pengenalan .Net Framework*.  
<https://garicfalo.blogspot.com/2017/11/pengenalan-net-framework.html>.

Suryantara, I Gusti Ngurah. 2017. *Merancang Aplikasi Rental dengan VB.Net*. Yogyakarta : Andi.

Yung, Kok. 2005. *Membangun Aplikasi Database dengan Visual Basic .Net 2005 dan Perintah SQL*. Jakarta : PT Elex Media Komputindo.

Yuswanto. 2006. *Pemrograman Dasar Visual Basic.Net*. Jakarta : Prestasi Pustaka Pustaka.

Yuswanto dan Subari. 2007. *Pemrograman Database Visual Basic.Net*. Jakarta : Prestasi Pustaka Pustaka.

**BIODATA PENULIS**

**Yunianita Rahmawati, S.Kom., M.Kom,** lahir di Magetan – Jawa Timur, pada tanggal 1 Juni 1985. Penulis menamatkan pendidikan S1 di STIKOM Surabaya Jurusan Sistem Informasi (2003-2008), dan menamatkan S2 Jurusan Teknologi Informasi di STTS Surabaya (2011-2017). Saat ini aktif berkarya sebagai dosen tetap Prodi Informatika Fakultas Saintek UMSIDA.

**Uce Indahyanti, M.Kom,** lahir di Situbondo – Jawa Timur, pada tanggal 11 Mei 1971. Penulis menamatkan pendidikan S1 di STIKOM Surabaya Jurusan Manajemen Informatika (1990 - 1996), dan menamatkan S2 Jurusan Sistem Informasi di ITS Surabaya (2010-2012). Saat ini aktif berkarya sebagai



dosen tetap Prodi Informatika Fakultas Saintek UMSIDA. Pernah bergabung dalam sebuah proyek nasional terkait pengembangan e-learning yang dibiayai oleh lembaga nirlaba IGI-GTZ. Penelitian atau publikasi yang pernah dilakukan seputar bidang sistem informasi dan sistem e-learning.