



**Kementerian Pendidikan Dan Kebudayaan
Republik Indonesia
2013**



TEKNIK PEMOGRAMAN

Semester

1

Untuk SMK / MAK Kelas X

Penulis : **WELDAN KUSUF**
Editor Materi : **DJOKO SUGIONO**
Editor Bahasa :
Ilustrasi Sampul :
Desain & Ilustrasi Buku : **PPPPTK BOE MALANG**

Hak Cipta © 2013, Kementerian Pendidikan & Kebudayaan

**MILIK NEGARA
TIDAK DIPERDAGANGKAN**

Semua hak cipta dilindungi undang-undang.

Dilarang memperbanyak (merekproduksi), mendistribusikan, atau memindahkan sebagian atau seluruh isi buku teks dalam bentuk apapun atau dengan cara apapun, termasuk fotokopi, rekaman, atau melalui metode (media) elektronik atau mekanis lainnya, tanpa izin tertulis dari penerbit, kecuali dalam kasus lain, seperti diwujudkan dalam kutipan singkat atau tinjauan penulisan ilmiah dan penggunaan non-komersial tertentu lainnya diizinkan oleh perundangan hak cipta. Penggunaan untuk komersial harus mendapat izin tertulis dari Penerbit.

Hak publikasi dan penerbitan dari seluruh isi buku teks dipegang oleh Kementerian Pendidikan & Kebudayaan.

Untuk permohonan izin dapat ditujukan kepada Direktorat Pembinaan Sekolah Menengah Kejuruan, melalui alamat berikut ini:

Pusat Pengembangan & Pemberdayaan Pendidik & Tenaga Kependidikan Bidang Otomotif & Elektronika:

Jl. Teluk Mandar, Arjosari Tromol Pos 5, Malang 65102, Telp. (0341) 491239, (0341) 495849, Fax. (0341) 491342, Surel: vedcmalang@vedcmalang.or.id, Laman: www.vedcmalang.com

DISKLAIMER (*DISCLAIMER*)

Penerbit tidak menjamin kebenaran dan keakuratan isi/informasi yang tertulis di dalam buku tek ini. Kebenaran dan keakuratan isi/informasi merupakan tanggung jawab dan wewenang dari penulis.

Penerbit tidak bertanggung jawab dan tidak melayani terhadap semua komentar apapun yang ada didalam buku teks ini. Setiap komentar yang tercantum untuk tujuan perbaikan isi adalah tanggung jawab dari masing-masing penulis.

Setiap kutipan yang ada di dalam buku teks akan dicantumkan sumbernya dan penerbit tidak bertanggung jawab terhadap isi dari kutipan tersebut. Kebenaran keakuratan isi kutipan tetap menjadi tanggung jawab dan hak diberikan pada penulis dan pemilik asli. Penulis bertanggung jawab penuh terhadap setiap perawatan (perbaikan) dalam menyusun informasi dan bahan dalam buku teks ini.

Penerbit tidak bertanggung jawab atas kerugian, kerusakan atau ketidaknyamanan yang disebabkan sebagai akibat dari ketidakjelasan, ketidaktepatan atau kesalahan didalam menyusun makna kalimat didalam buku teks ini.

Kewenangan Penerbit hanya sebatas memindahkan atau menerbitkan mempublikasi, mencetak, memegang dan memproses data sesuai dengan undang-undang yang berkaitan dengan perlindungan data.

Katalog Dalam Terbitan (KDT)
Teknik Elektronika, Edisi Pertama 2013
Kementerian Pendidikan & Kebudayaan
Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan, th.
2013: Jakarta

KATA PENGANTAR

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi pembelajaran (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Studi Teknologi dan Rekayasa Teknik Elektronika.

Buku teks "Teknik Pemrograman" ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran "Teknik Pemrograman" ini disusun dengan tujuan agar supaya peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan penyelidikan ilmiah (penerapan saintifik), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks Siswa untuk Mata Pelajaran Teknik Pemrograman Kapal kelas X/Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2013
Menteri Pendidikan dan Kebudayaan

Prof. Dr. Mohammad Nuh, DEA

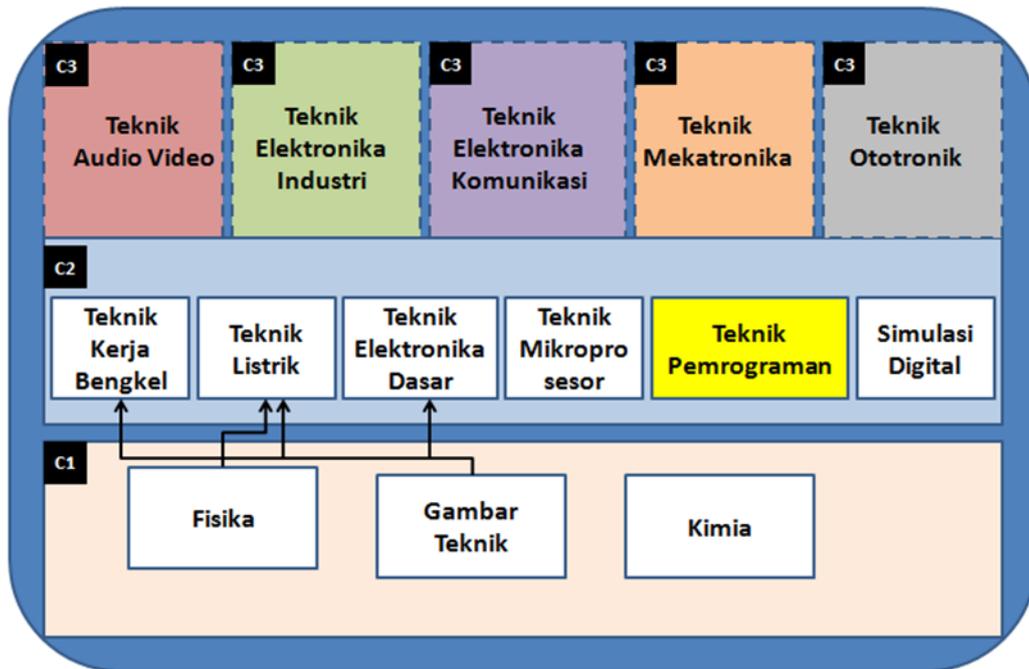
DAFTAR ISI

Hak Cipta dan Disklaimer	ii
Kata Pengantar	iv
Daftar Isi	v
PENDAHULUAN	
A. Deskripsi Materi Pembelajaran	
B. Prasyarat	
C. Petunjuk Penggunaan	
D. Tujuan Akhir	
E. Kompetensi Dasar	
F. Cek Kemampuan Awal	
KEGIATAN BELAJAR 1	
Memahami Bahasa Pemrograman Visual Basic	5
KEGIATAN 1: Mengenal <i>Integrated Development Environment Visual Basic</i>	6
1.1 Menjalankan Visual Basic 2010	6
1.2 Halaman Muka IDE Visual Basic 2010	7
1.3 Antarmuka Pemrograman Visual Basic 2010	10
1.4 Pemrograman Berorientasi Obyek (OOP) VB 2010	11
KEGIATAN 2 : Memulai Menggunakan Lingkungan Pengembang Terintegrasi (IDE) VB 2010	15
2.1 Mendesain Antarmuka Pengguna	15
2.1.1 Visual Studio Toolbox	16
2.1.2 Form Designer	19
2.1.3 Code Editor	20
2.2 Awal Membuat Aplikasi	20
2.2.1 Menambahkan Komponen	21
2.2.2 Mengatur Komponen	21
2.2.3 Menambahkan Kode Program	23
2.2.4 Mengatur Control Properties menggunakan Properties Window	24

2.2.5	Properti Default	27
2.2.6	Menjalankan Aplikasi	29
2.3	Membuat Program Visual Basic Yang Pertama	30
	KEGIATAN 3 : Membuat Program Dengan Visual Basic	35
3.1	Obyek-Obyek Standard dalam Visual Basic	35
3.2	Pemberian Awalan Nama Obyek	36
3.3	Mengunci Kontrol	36
3.4	Pembuatan Program	37
3.4.1	Membuat Antarmuka (Interface)	37
3.4.2	Mengatur Property	39
3.4.3	Menuliskan Kode di File Form	41
	KEGIATAN 4 : Memahami Konsep Dasar Visual Basic	45
4.1	Memahami Struktur Kode dari Suatu Prosedur Event (Kejadian)	45
4.2	Prosedur <i>Function</i> Visual Basic 2010	48
4.3	Variabel dan Tipe Data Pada Pemrograman Visual Basic 2010	57
4.3.1	Mendeklarasikan Variabel	57
4.3.2	Penggunaan Variabel Dalam Program	60
4.3.3	Penggunaan Variabel Untuk Input Data	62
4.3.4	Penggunaan Variabel Sebagai Output	64
4.3.5	Tipe Data	66
4.4	Operator Matematika	69
4.4.1	Empat Operator Utama	69
4.4.2	Pembagian Integer, Modulus, dan Perpangkatan	70
4.4.3	Operator Logika	70
4.4.4	Operator Kondisi	74
4.5	Fungsi Matematika dalam Visual Basic	74
4.6	Array	85
	KEGIATAN BELAJAR 2	
	Membuat Program Aplikasi Sederhana Dengan Bahasa Pemrograman Visual Basic	89

KEGIATAN 1 : Flow Chart atau Diagram Alir	90
1.1 Konsep Dasar Algoritma	90
1.2 Pengertian Dari Flowchart	91
1.3 Diagram Alir Program (Program Flowchart)	92
KEGIATAN 2 : Pengenalan Instruksi Struktur Dan Syarat Dalam Bahasa Visual Basic	97
2.1 Struktur Keputusan	97
2.2 IF.....Then	97
2.3 IF.....THEN.....ELSE	98
2.4 IF.....THEN.....ELSEIF.....ELSEIF.....ENDIF	98
2.5 SELECT CASE	101
2.6 SELECT CASE Berkondisi	102
2.7 SELECT CASE Dalam Rentang	103
2.8 IF Bertingkat	105
2.9 Menguji Data	106
KEGIATAN 3 : Pengenalan Instruksi Perulangan Dalam Bahasa Visual Basic	111
3.1 Perulangan (Looping)	111
3.2 Do.....Loop	111
3.3 While.....End While	118
3.4 For.....Next	119
3.5 Perulangan Berulang	123
3.6 Studi Kasus Tentang Penggunaan Struktur Perulangan	125
KEGIATAN 4 : Menggunakan Fungsi Timer	137
4.1 Membuat Jam Digital menggunakan Kontrol Timer	137
4.2 Menggunakan Obyek Timer Untuk Mengatur Batas Waktu	140
4.3 Menguji Program Password	143
4.4 Membuat Aplikasi Visualisasi Lampu LED Berkedip	145
KEGIATAN 5 : Membuat Program Aplikasi Dengan Visual Basic 2010	153
5.1 Program Lampu Lalu Lintas 1	153
5.2 Program Lampu Lalu Lintas 2	170
Daftar Pustaka	174

PRTA KEDUDUKAN MODUL



PENDAHULUAN

A. DESKRIPSI MATERI PEMBELAJARAN

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Meskipun awalnya program visual basic lebih banyak digunakan untuk program aplikasi bisnis, akan tetapi perkembangannya visual basic banyak juga digaga dikembangkan untuk mendukung aplikasi-aplikasi dunia teknik.

Pada buku siswa ini dibahas tentang bahasa pemrograman Visual Basic 2010, dimana pembahasan buku mencakup dari pengenalan visual basic, dasar pemrograman hingga aplikasi input-output yang berhubungan langsung dengan port serial mikrokontroler. Dengan mempelajari teknik pemrograman visual basic diharapkan siswa dapat memahami dasar teknik pemrograman dan mengaplikasikan Visual Basic sebagai fungsi aplikasi kontrol maupun sebagai antarmuka antara manusia dengan mesin atau biasa disebut HMI (*Human Machine Interface*).

B. PRASYARAT

Materi Teknik Pemrograman 1 memberikan bekal awal dalam memahami kompe-tensi teknik pemrograman pada jurusan teknik elektronika. Materi ini disampaikan pada kelas X semester 1.

C. PETUNJUK PENGGUNAAN

Buku ini disusun dengan memberikan penjelasan tentang konsep dasar pemrograman Visual Basic 2010 dengan beberapa contoh aplikasi sederhana yang berkaitan dengan dunia teknik pada umumnya dan elektronika pada khususnya. Untuk memungkinkan siswa belajar sendiri secara tuntas , maka perlu diketahui bahwa isi buku ini pada setiap kegiatan belajar umumnya terdiri atas, uraian materi, contoh-contoh aplikasi, tugas dan tes formatif serta lembar kerja, sehingga diharapkan siswa dapat belajar mandiri (*individual learning*) dan *mastery learning* (belajar tuntas) dapat tercapai.

D. TUJUAN AKHIR

Tujuan akhir yang hendak dicapai adalah agar siswa mampu:

Mengenal Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa *Visual Basic*.

Memahami menggunakan Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa *Visual Basic*.

Memahami konsep dasar bahasa *Visual Basic*.

Mampu membuat program aplikasi sederhana dengan bahasa Visual Basic

KOMPETENSI INTI (KI-3)	KOMPETENSI INTI (KI-4)
<p>Kompetensi Dasar (KD): Memahami bahasa pemrograman Visual Basic</p>	<p>Kompetensi Dasar (KD): Membuat program aplikasi sederhana dengan bahasa pemrograman Visual Basic</p>
<p>Indikator: Mengenal Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Menjelaskan bagaimana memulai menggunakan Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Memahami konsep dasar bahasa <i>Visual Basic</i>. Memahami program aplikasi sederhana dengan bahasa Visual Basic</p>	<p>Indikator: Membuat program menggunakan <i>Integrated Development Enviroment (IDE)</i> Melakukan percobaan dan menerapkan penggunaan Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Melakukan percobaan menggunakan konsep dasar bahasa pemrograman bahasa visual basic. Merencana flow chart untuk persiapan pemrograman. Membuat program sederhana dengan <i>Visual Basic</i> untuk divisualisasikan di layar monitor.</p>

KOMPETENSI INTI (KI-3)	KOMPETENSI INTI (KI-4)
<p>Kompetensi Dasar (KD): Memahami bahasa pemrograman Visual Basic</p>	<p>Kompetensi Dasar (KD): Membuat program aplikasi sederhana dengan bahasa pemrograman Visual Basic</p>
<p>Indikator: Mengenal Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Menjelaskan bagaimana memulai menggunakan Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Memahami konsep dasar bahasa <i>Visual Basic</i>. Memahami program aplikasi sederhana dengan bahasa Visual Basic</p>	<p>Indikator: Membuat program menggunakan <i>Integrated Development Enviroment (IDE)</i> Melakukan percobaan dan menerapkan penggunaan Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i> Melakukan percobaan menggunakan konsep dasar bahasa pemrograman bahasa visual basic Merencana flow chart untuk persiapan pemrograman Membuat program sederhana dengan <i>Visual Basic</i> untuk divisualisasikan di layar monitor.</p>

E. CEK KEMAMPUAN AWAL

1. Sebutkan macam-macam bahasa pemrograman yang anda ketahui!
2. Sebutkan evolusi perkembangan bahasa basic hingga versi terbaru saat ini!
3. Apa yang anda ketahui tentang Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa *Visual Basic*? Jelaskan !
4. Apa yang dimaksud dengan pemrograman berorientasi obyek (*OOP*)?
5. Jelaskan apa yang dimaksud prosedur *even* dan prosedur *function* pada Visual Basic!
6. Apa yang dimaksud dengan variabel? Jelaskan!
7. Sebutkan tipe-tipe data yang terdapat pada visual basic?
8. Apa yang dimaksud dengan operator matematika dan logika?
9. Apa yang anda ketahui tentang struktur keputusan? Jelaskan!
10. Apa yang anda ketahui tentang perulangan? Jelaskan!

KEGIATAN BELAJAR 1**Memahami Bahasa Pemrograman Visual Basic****A. Tujuan Pembelajaran**

1. Mengetahui Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa Visual Basic.
2. Menjelaskan bagaimana memulai menggunakan Lingkungan Pengembangan.
3. Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa Visual Basic.
4. Memahami konsep dasar bahasa Visual Basic.
5. Memahami program aplikasi sederhana dengan bahasa Visual Basic.

B. Uraian Materi

1. Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa Visual Basic.
2. Memulai menggunakan Lingkungan Pengembangan Terintegrasi (*Integrated Development Enviroment-IDE*) bahasa Visual Basic.
3. Membuat program dengan Visual Basic.
4. Konsep dasar bahasa Visual Basic.

C. Alokasi Waktu

16 jam pelajaran

D. Metode Pembelajaran

Teori dan Praktek

E. Media pembelajaran

- PC/Notebook
- Windows 7
- Visual Basic 2010

MEMAHAMI BAHASA PEMROGRAMAN VISUAL BASIC

KEGIATAN 1: Mengenal *Integrated Development Environment Visual Basic*

Visual Basic 2010 merupakan versi Visual Basic yang dikeluarkan oleh Microsoft pada tahun 2010. Versi ini hampir serupa dengan versi Visual Basic 2008 dengan penambahan beberapa fitur baru. Pemrograman Visual Basic telah melalui beberapa fase pengembangan sejak pertama kali dikeluarkan BASIC (Beginners All-purpose Symbolic Instruction Code) pada versi DOS. Microsoft meluncurkan Visual Basic versi 1.0 pada tahun 1991 dan perlahan-lahan Microsoft merubah semua versi DOS menjadi MS Windows yang dikenal dengan Visual Basic.

Visual Basic merupakan bahasa pemrograman prosedural dan fungsional dan versi terpopuler adalah versi Visual Basic 6. Microsoft merubahnya menjadi bahasa pemrograman berorientasi objek yang lebih baik dengan meluncurkan Visual Basic.Net, Visual Basic 2005, Visual Basic 2008, Visual Basic 2010 dan versi terakhir adalah Visual Basic 2012.

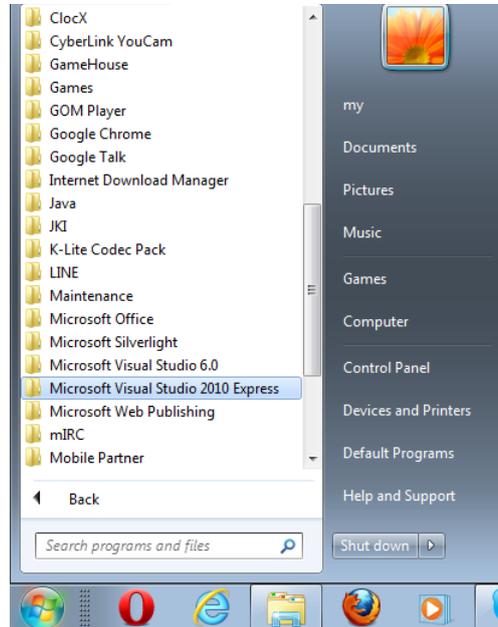
Guru dan siswa sebagian besar telah mengenal versi Visual Basic 6 akan tetapi kurang mengenal Visual Basic versi selanjutnya yang dikeluarkan oleh Microsoft. Jika kita belajar versi Visual Basic 2012 maka akan sedikit lebih membingungkan, oleh karenanya di buku ini kita akan mempelajari Visual Basic 2010 sehingga hanya memerlukan sedikit adaptasi jika kita ingin mengupdate ke Visual Basic versi 2012

1.1 Menjalankan Visual Basic 2010

Untuk menjalankan Visual Basic 2010, langkah yang harus dilakukan adalah:

1. Klik tombol Start -> All Program -> Microsoft Visual Basic 2010

2. Sehingga akan muncul jendela pembuka seperti gambar berikut



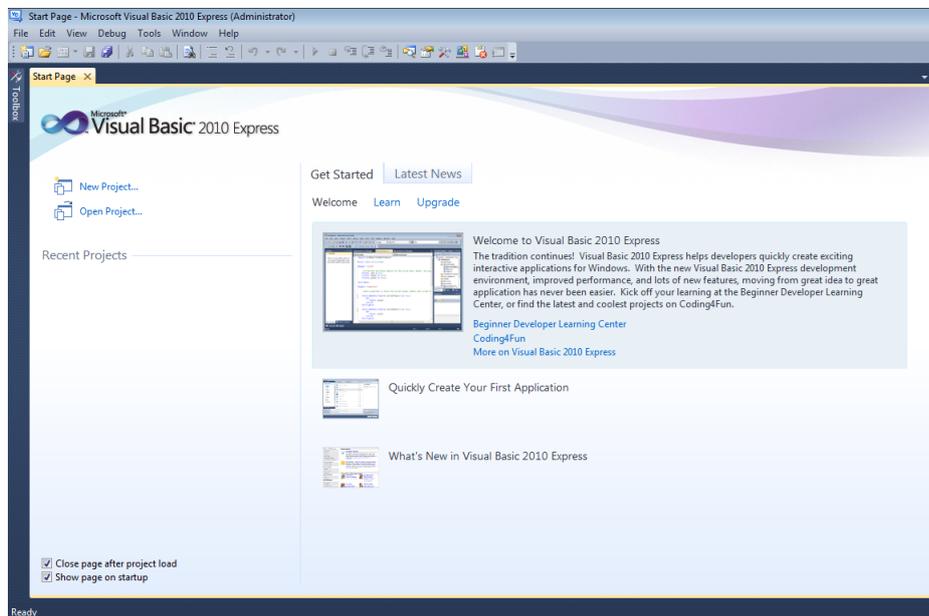
3. Tunggu sampai semua modul yang dibutuhkan untuk menjalankan Visual Basic 2010 dimuat. Apabila Anda menjalankan Visual Studio untuk pertama kali, maka akan muncul jendela yang digunakan untuk melakukan pemilihan setting standard. Pilih salah satu jenis lingkungan dari daftar yang disediakan, misalnya pilih Visual Basic Development Setting.
4. Setelah setting standard terpilih, tekan tombol Start Visual Studio untuk menerapkan settingan terpilih dan membuka Visual Studio 2010 dengan settingan yang sudah Anda atur.
5. Setelah proses loading selesai, Anda akan ditunjukkan tampilan utama dari IDE Microsoft Visual Studio 2010.

1.2 Halaman Muka IDE Visual Basic 2010

Secara umum, setiap Anda mejalankan Visual Basic pertama kali, Anda akan dihadapkan pada Start Page yang berisi berbagai macam informasi yang akan mengantarkan kepada Anda semua hal mengenai Visaual Basic 2010 secara detail. Apabila Anda terkoneksi dengan internet, maka bagian Start Page juga

akan menampilkan informasi terkini mengenai Visual Basic 2010 yang berasal dari RSS Feed yang disediakan, sehingga Anda tidak ketinggalan mengenai informasi terbaru mengenai Visual Basic 2010.

Antarmuka Visual Basic 2010 hanya sedikit berbeda dengan antarmuka Visual Basic versi sebelumnya (VB.Net, VB 2005, VB 2008) akan tetapi jauh berbeda dengan Visual Basic versi 6.0. Semenjak menggunakan .Net Framework di Visual Basic 2003, perkembangan yang ada dari Visual Basic versi terbaru tidak mengubah antarmukanya menjadi lebih baru secara signifikan, walaupun secara teknologi memiliki peningkatan yang bisa dibilang cukup signifikan.



Gambar 1.1 Jendela kerja Visual Basic 2010 sebelum dibuat sebuah project

Ketika membuka aplikasi Visual Basic 2010 akan muncul jendela antarmuka atau Integrated Development Environment (IDE) seperti gambar di atas.

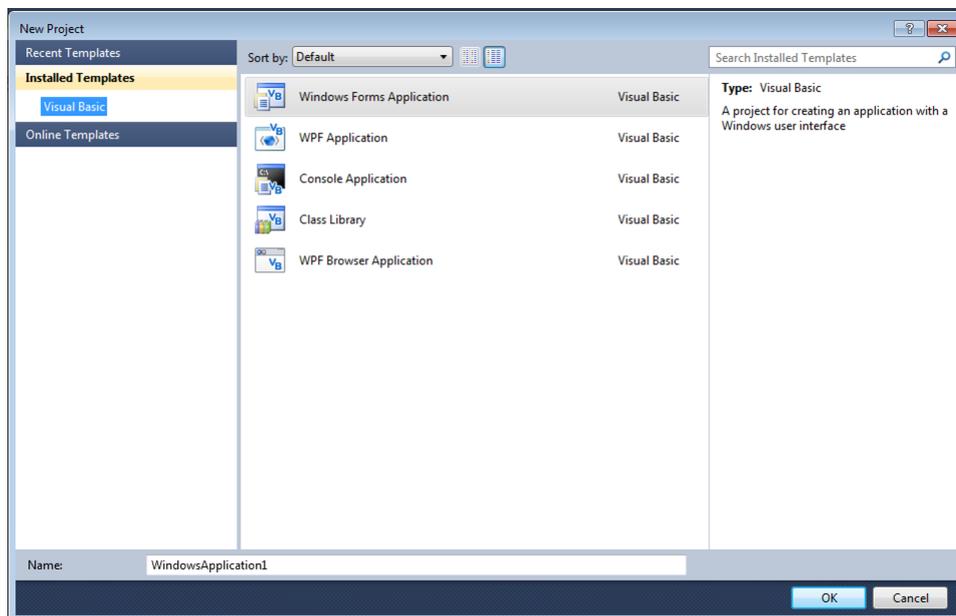
Antarmuka IDE terdiri atas beberapa panel yaitu:

- Recent Project Panel, menunjukkan project terakhir yang telah dibuat.
- Get Started Panel, menyediakan informasi dan tips untuk membuat dan mengembangkan aplikasi baru dengan cepat.
- Latest News Panel, menyediakan berita terbaru secara online tentang Visual Basic 2010.

Disamping itu, juga terdapat dua buah icon, new project dan open project.

Icon tersebut akan kita gunakan untuk membuat project baru dan membuka project lama yang telah kita simpan.

Klik pada icon **New Project**, sehingga muncul jendela New Project seperti tampak pada gambar di bawah ini



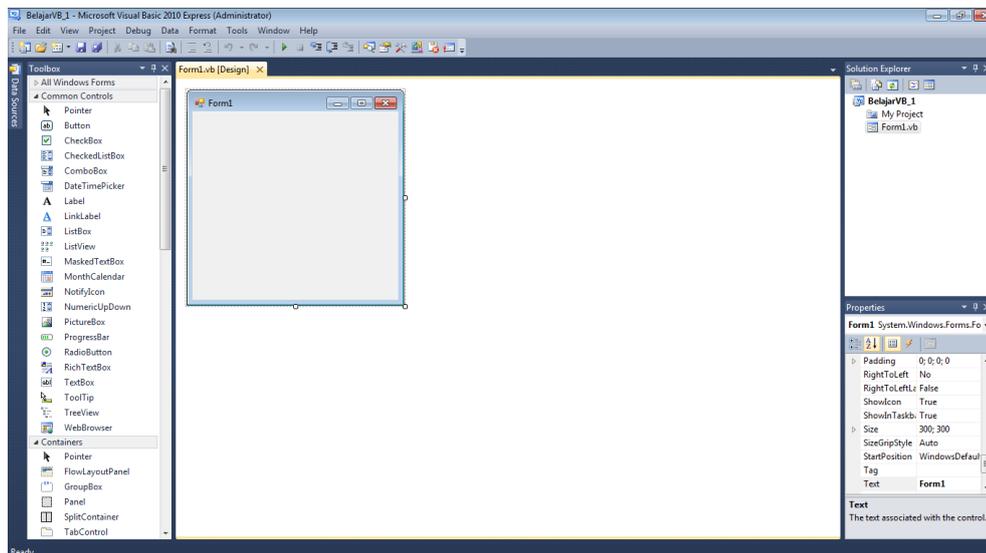
Gambar 1.2 Jendela New Project Visual basic 2010

Pada jendela tersebut terdapat lima buah tipe project yang dapat kita buat yaitu, Windows Form Application, WPF Application, Console Application, Class Library dan WPF Browser Application.

Pilih Windows Form Application dan gantilah nama default project (**WindowsApplication1**) menjadi nama project yang kita inginkan, sebagai contoh: BelajarVB_1. Setelah mengganti nama project kemudian klik OK dan akan muncul jendela kerja tempat kita mendesain dan mengembangkan program.

1.3 Antarmuka Pemrograman Visual Basic 2010

Antarmuka IDE Visual Basic 2010 tampak seperti gambar 3, antarmuka IDE terdiri atas **Menu Bar**, **Toolbar**, **Form Kosong**, **Solution Explorer**, dan **Properties Window**.



Gambar 1.3 Jendela Kerja Visual Basic 2010 ketika project sudah dibuat

Tampilan antarmuka VB 2010 di atas mungkin tidak selalu sama dengan yang muncul pada PC atau laptop Anda, tergantung bagaimana pengaturan tampilannya. Anda dapat mengatur tampilan antarmuka dengan menggeser jendela, membesarkan, mengecilkan atau membiarkan mengambang.

Jika ada tampilan jendela yang tidak terlihat sebagai contoh panel **properties**, maka Anda dapat memunculkan dengan memilih **View** pada menu bar kemudian **Other Windows** dan pilihlah **Properties Windows**. Sehingga muncul jendela Properties yang kita inginkan.

- Form, adalah tempat kita membangun dan mendesain tampilan aplikasi.
- Solution Explorer, menampilkan *list project, file*, dan komponen lainnya sehingga kita dapat dengan mudah untuk mencari dan mengaksesnya.

Sebagai contoh menampilkan project BelajarVB_1 dan Form 1.VB pada gambar 1.3

- Properties Window, tempat untuk mengatur properties karakteristik dari suatu objek yang terletak pada aplikasi yang dibuat.
- Setiap object memiliki default properties masing-masing dan dapat diatur sesuai keinginan dengan merubah setting properti dari masing-masing object.

1.4 Pemrograman Berorientasi Obyek (OOP) VB 2010

Seperti yang telah kita ketahui, bahwa Visual Basic 2010 hampir sama dengan Visual Basic 6 dalam interface/antarmuka dan dari struktur program. Perbedaan utama dari Visual Basic 2010 adalah pemrograman total berorientasi obyek, sedangkan Visual Basic 6 merupakan pemrograman berorientasi obyek akan tetapi belum menyeluruh. Untuk mengkuualifikasikan suatu pemrograman berorientasi obyek secara menyeluruh harus memiliki tiga kriteria teknologi yaitu : Encapsulation, Inheritance dan Polymorphism.

- Encapsulation

Encapsulation mengacu pada modul yang dibuat berdasar *self –contained*, dimana menghubungkan fungsi-fungsi proses dari pengolahan data. Nama dari fungsi tersebut dinamakan Class. Setiap class terdiri dari data dan metode untuk mengolah atau memanipulasi data tersebut. Setiap komponen data dari setiap *class* disebut variabel *instance*. Dimana setiap *Class* merupakan suatu obyek. Sebagai contoh, pada sistem sebuah class bernama Libraryterdapat beberapa komponen anggota data (contoh Andi dan Budi). Andi dan Budi adalah dua buah komponen anggota yang disebut instance (dua obyek) dari *classlibrary*.

- Inheritance

Pada pemrograman berorientasi obyek dibuat berdasarkan hierarki-nya. Inheritance, adalah suatu fungsi yang memungkinkan struktur dan metode yang dibuat pada suatu class dapat dipakai dan diaplikasikan pada class yang lain. Itu artinya lebih sedikit program yang dibutuhkan ketika kita menambahkan suatu fungsi pada sistem yang kompleks, sehigga menghemat waktu dan tenaga. Jika sebuah step ditambahkan pada bagian akhir dari suatu hierarki, maka hanya data yang berhubungan dengan kebutuhan step tertentu tersebut yang akan diproses. Semua yang berhubungan dengan step tersebut merupakan enherited, artinya dapat dipakai secara global di fungsi-fingsi yang lain.

- Polymorphism

Pemrograman berorientasi obyek memungkinkan suatu prosedur sebuah obyek yang dibuat tidak diketahui secara tepat tipe data yang akan digunakan sebelumnya dan baru akan diketahui ketika proses *run time* dijalankan. Sebagai contoh, Bentuk dari cursorakan berubah dari tanda panah menjadi garis tergantung dari mode program yang dijalankan. Routine yang menggerakkan cursor pada screen oleh mouse akan memberikan respon pada program dan polymorphism memungkinkan cursor tersebut berubah bentuknya ketika program dijalankan dan juga memungkinkan membuat bentuk baru secara mudah diintegrasikan.

Visual Basic 6 melakukan pemrograman basic berorientasi obyek yang tidak menyeluruh karena tidak memiliki kemampuan inheritance. Akan tetapi di Visual Basic 2010 sudah merupakan pemrograman berbasis obyek secara menyeluruh seperti OOP programmer lain (C++ dan Java). Pada Visual Basic 6 masih bisa dikatakan sistem pemrograman prosedural atau fungsional sama seperti program C, Pascal dan Fortran.

Visual Basic 2010 memungkinkan pengguna untuk menulis program dan membuatnya menjadi modul-modul class. Sebuah class juga dapat terdiri dari beberapa sub class. Sebagai contoh

Suatu class tanaman terdiri dari sub class pohon apel, Dan sub class pohon apel tersebut memiliki instance bernama buah apel.

TUGAS

1. Installah software Visual Basic 2010 pada laptop atau PC Anda. Bukalah aplikasi program tersebut dan amati tampilan halaman muka IDE Visual Basic 2010.
2. Buatlah project baru bernama "test1" sehingga tampak antarmuka pemrograman Visual Basic 2010. Amati bagian jendela kerja, properties, toolbox, dan solution explorer.

TES FORMATIF

1. Sebutkan beberapa buah panel yang terdapat pada jendela antarmuka (IDE) dan sebutkan fungsinya!
2. Apa saja fungsi dari Form, Solution Explorer, Properti Window, dan Default pada jendela Properti Windows?
3. Visual Basic 2010 merupakan Visual Basic berorientasi obyek secara menyeluruh. Sebutkan tiga syarat yang harus dimiliki pemrograman berorientasi obyek tersebut!

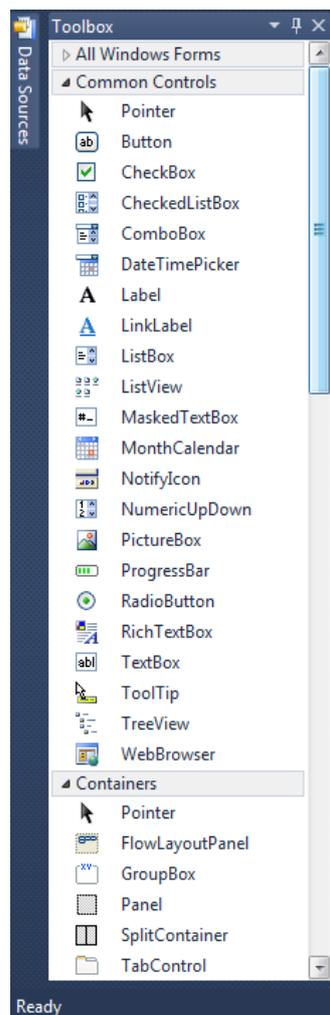


Jawab :

KEGIATAN 2 : Memulai Menggunakan Lingkungan Pengembang Terintegrasi (IDE) VB 2010

2.1 Mendesain Antarmuka Pengguna

Langkah awal membuat aplikasi project VB 2010 dimulai dengan mendesain tampilan antarmuka pengguna. Anda dapat mendesain antarmuka dengan menambahkan kontrol ke dalam form dan mengatur properties nya. Anda juga dapat menambahkan kontrol dari Toolbox, untuk menampilkan Toolbox Window dapat menggunakan short-cut Ctrl+Alt+X atau klik pada icon Toolbox yang terdiri dari banyak kontrol seperti Butto, TextBox, Label, ControlBox, CheckBox, dan lainnya seperti yang terlihat pada gambar 4.



Gambar 2.1 ToolBox Visual Basic 2010

Control Toolbox terdiri atas semua fungsi kontrol untuk mengembangkan fungsi aplikasi Visual Basic 2010. Control Toolbox secara umum dikategorikan menjadi tujuh yaitu, Common Control, Container, Menu, Tollbar, Data, Component, Printings dan Dialog.

Untuk tahap awal pembelajaran, kita akan membahas lebih banyak pada Common Control. Beberapa komponen yang sering digunakan pada Common Control adalah Button, Label, ComboBox, ListBox, PictureBox, TextBox dan lain-lain. Untuk menambahkan komponen-komponen tersebut ke dalam Form, cukup kita pilih komponen yang dibutuhkan kemudian geser dan letakkan ke dalam form.

Kita dapat merubah ukuran dan posisi setiap komponen tersebut dengan mudah serta dapat enambahkan sebanyak mungkin kontrol yang kita inginkan. Akan tetapi sebaiknya hindari form dengan jumlah komponen yang berlebihan.

2.1.1 Visual Studio Toolbox

Salah satu bagian penting pemrograman visal menggunakan Visual Basic 2010 adalah toolbox. Toolbox dapat diartikan sebagai kotak alat yang menyimpan banyak ikon yang melambangkan objek yang disebut kontrol. Kontrol pada toolbox bisa digunakan untuk mengeksekusi suatu kejadian di form aplikasi selama pemrograman.

Mengetahui nama dan fungsi kontrol di toolbox sangat penting karena berkaitan langsung dengan bagaimana kita akan mendesain rancangan aplikasi program. Berikut ini tampilan toolbox yang akan sering digunakan ketika belajar menggunakan Visual Basic 2010.

Penjelasan fungsi-fungsi dari kontrol toolbox yang ada di Visual Basic 2010 adalah sebagai berikut :

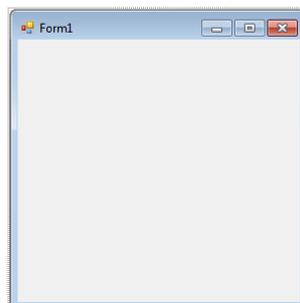
Komponen	Penjelasan/Fungsi
	Pointer Untuk memilih pointer mouse atau melepaskan dan memilih objek-objek toolbox.
	Button Untuk menerima klik dari pengguna. Jika pengguna mengkliknya maka tombol tersebut dapat dieksekusi.
	CheckBox Untuk menerima tanda centang dari pengguna. Jika pengguna mencentangnya maka item di kotak tersebut otomatis bisa dipakai untuk program.
	CheckedListBox Menampilkan item-item seperti ListBox biasa, namun dengan CheckBox disebelah kirinya sehingga satu ListBox dapat menerima dua item atau lebih.
	ComboBox Menampilkan item-item namun dalam bentuk hanya satu yang terlihat. Jika pengguna mengklik ComboBox, barulah semua item di dalamnya terlihat.
	DateTimePicker Memungkinkan pengguna menampilkan tanggal dan memilihnya. Selanjutnya tanggal dan waktu yang dipilih dapat dimodifikasi menggunakan format tertentu.
A	Label Untuk menampilkan teks dimana nantinya isi di dalamnya tidak bisa diberi masukan oleh pengguna seperti halnya TextBox.

	<p>LinkLabel</p> <p>Sama seperti Label, namun dapat memiliki link aplikasi atau ke website. Jika di klik, pengguna akan dibawa menuju ke alamat URL yang telah ditentukan. Jika alamat URL ada di internet otomatis browser akan menampilkan alamat tersebut.</p>
	<p>ListBox</p> <p>Menampilkan item-item dalam bentuk kotak dimana semua item ditampilkan di dalam daftar. Jika jumlah item lebih banyak dari ukuran ListBox maka ListBox otomatis memiliki scroll bar vertikal yang dapat dipakai untuk menampilkan semua item.</p>
	<p>ListView</p> <p>Menampilkan item-item dalam dalam satu dari lima tamplan yang berbeda</p>
	<p>MaskedTextBox</p> <p>Menerima input dari pengguna seperti halnya TextBox namun isi di dalamnya nantinya dapat di <i>masking</i> menggunakan format tertentu. Cocok untuk pengisian nomor telpon atau sejenisnya.</p>
	<p>MonthCalendar</p> <p>Menampilkan kalender bulanan dimana pengguna bisa memilih tanggal.</p>
	<p>NotifyIcon</p> <p>Menampilkan ikon di area notifikasi ketika program berjalan. Lokasi area notifikasi terletak di sebelah kanan taskbar Windows.</p>
	<p>NumericUpDown</p> <p>Menampilkan kotak teks berisi angka dimana angka-angka tersebut bisa ditambah jika tombol Up diklik dan berkurang jika tombol Down diklik.</p>
	<p>PictureBox</p> <p>Menampilkan gambar yang gambarnya bisa diatur</p>
	<p>ProgressBar</p> <p>Menampilkan sebuah bilah yang mengindikasikan kemajuan operasi pengguna tertentu.</p>

	RadioButton Memungkinkan pengguna untuk memilih satu opsi dari beberapa pilihan yang dikaitkan dengan tombol radio lainnya
	RichTextBox Menampilkan fasilitas pengeditan teks dengan pemformatan yang kaya dan fitur lain seperti pemformatan karakter dan paragraf.
	TextBox Memungkinkan pengguna memasukkan teks yang bisa lebih dari satu baris dan menyediakan kemampuan pengeditan dan <i>masking</i> kata sandi.
	ToolStrip Komponen yang menyediakan toolbar dan elemen antarmuka lain yang dapat mendukung berbagai opsi tampilan. Selain itu bisa mendukung <i>overflow</i> dan pengaturan ulang saat program berjalan
	TreeView Menampilkan koleksi hierarki itemitem yang memiliki label dimana nantinya bisa diberi gambar
	WebBrowser Memungkinkan pengguna membuka halaman web dari dalam form.

2.1.2 Form Designer

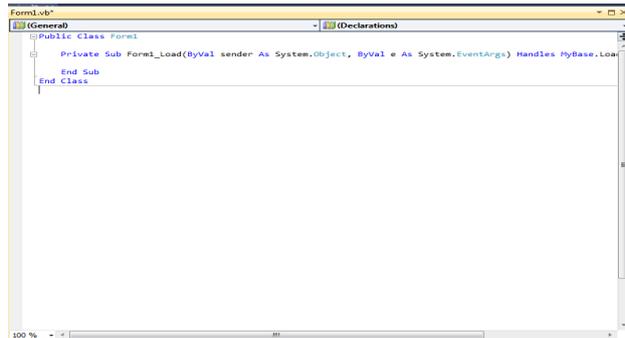
Form Designer merupakan fitur dari Visual Basic 2010 yang digunakan untuk membuat desain antarmuka atau interface dari aplikasi yang Anda kembangkan. Dengan mengadopsi fitur click-and-drop, proses penambahan komponen menjadi semakin dinamis dan mudah. Selain itu tersedia pula fitur visual guidelines yang memudahkan Anda dalam menata komponen yang terdapat pada desain form yang sedang Anda kerjakan



Gambar 2.2 Form Designer

2.1.3 Code Editor

Code Editor merupakan fitur dari Visual Basic 2010 yang digunakan untuk menambahkan kode program dari aplikasi atau project yang sedang Anda kerjakan.



Gambar 2.3 Code Editor

2.2 Awal Membuat Aplikasi

Proses pembuatan desain aplikasi bisa dikatakan proses yang paling krusial bagi para pengembang aplikasi. Menarik tidaknya suatu aplikasi yang dibuat akan sangat berpengaruh pada banyaknya user atau pengguna dari aplikasi tersebut. Tentunya dengan mengesampingkan terlebih dahulu fungsionalitas dan kompleksitas suatu aplikasi.

Bisa dikatakan pandangan seorang user atau pengguna aplikasi akan berpengaruh pada dipakai atau tidaknya suatu aplikasi yang dibuat. Selain menarik bagi pengguna suatu aplikasi juga harus didesain sedemikian sehingga pengguna merasa dibantu dan dipermudah dalam operasionalnya. Semakin informatif dan mudahnya suatu aplikasi yang dibuat akan menjadikan pengguna semakin nyaman dalam mengoperasikan aplikasi tersebut.

Ada beberapa langkah dasar yang harus Anda pahami sebelum Anda melakukan desain aplikasi.

2.2.1 Menambahkan Komponen

Komponen merupakan suatu obyek yang memiliki fungsi tertentu yang digunakan dalam desain antarmuka suatu aplikasi atau project. Untuk menambahkan suatu obyek (komponen) ke dalam form desainer, urutan langkah yang harus Anda lakukan adalah :

1. Dari aplikasi atau project yang sedang dibuka, perhatikan disebelah kiri form designer akan tampak Toolbox yang tercetak secara vertikal.
2. Perhatikan pada bagian Toolbox terlihat bahwa komponen tersusun berdasarkan kelompok kategori yang sudah diatur oleh Microsoft. Pilih salah satu komponen yang ada pada daftar dengan mengarahkan pada komponen yang diinginkan dan kliklah komponen tersebut.
3. Geser dan arahkan kursor pada desain antarmuka yang sedang dikerjakan sehingga toolbox komponen akan bergeser menutup.
4. Lakukan klik kiri pada sembarang lokasi pada form designer yang sedang dibuka.
5. Selanjutnya komponen yang telah Anda pilih akan diletakkan pada posisi yang Anda inginkan.

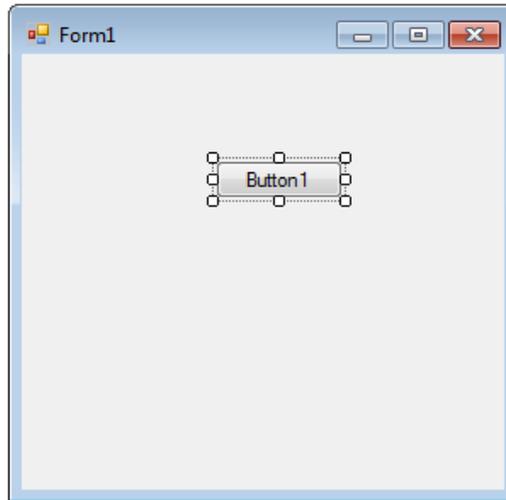
2.2.2 Mengatur Komponen

Terdapat dua cara yang digunakan untuk mengatur komponen yang sudah ditambahkan pada aplikasi yang Anda buat, yaitu mengubah ukuran komponen visual dan memindah komponen visual.

a. Mengubah Ukuran Komponen Visual

Untuk mengubah ukuran dari komponen visual yang ditambahkan pada form designer, langkah-langkah yang harus Anda lakukan adalah:

1. Pilih salah satu komponen yang terdapat pada desain form yang Anda buat.



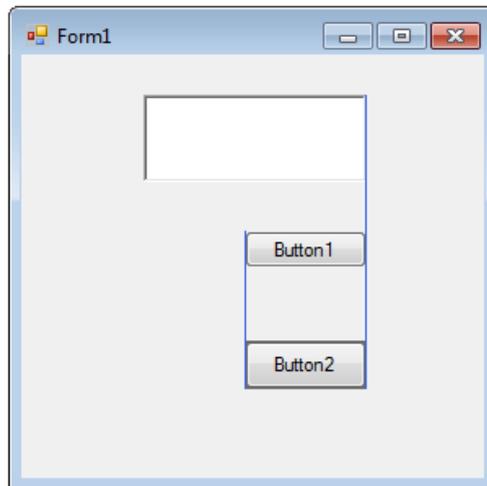
Gambar 2.4 Pemilihan Komponen

2. Akan terlihat garis bantu yang mengelilingi komponen atau obyek yang dipilih disertai dengan 8 buah node berupa lingkaran putih di sekelilingnya.

3. Klik salah satu node yang berada di sekeliling obyek, tahan lalu tarik ke arah yang Anda inginkan sehingga ukuran dari obyek terpilih akan berubah.

b. Menata Komponen Visual

Susunan komponen atau obyek yang terdapat pada aplikasi Anda awalnya akan terlihat agak tidak teratur sehingga perlu dilakukan langkah untuk menata dan merapikan obyek tersebut agar terlihat lebih rapi. Langkah yang harus Anda lakukan pada prinsipnya sama dengan memindahkan komponen di atas, namun ada satu fitur yang digunakan untuk merapikan susunan dari obyek yang ada, yaitu **visual guidelines** berupa garis bantu untuk merapikan obyek.



Gambar 2.5 Visual Guidelines

Garis samar-samar berwarna seperti pada gambar diatas merupakan visualisasi dari visual guidelines. Garis tersebut akan muncul secara otomatis apabila Anda melakukan pemindahan lokasi suatu obyek.

2.2.3 Menambahkan Kode Program

Penambahan kode program dalam suatu lingkungan pemrograman dilakukan di suatu bagian yang dinamakan **code editor**. Biasanya code editor akan memiliki banyak sekali fitur seperti code completion, code insight yang digunakan untuk memudahkan programmer dalam menuliskan kode program.

Pada Visual Basic 2010, untuk masuk menuju code editor Anda bisa menggunakan menu **View -> Code** atau bisa juga dengan menggunakan *shortcutkeyboard* yaitu dengan menekan **F7**.

Pada code editor yang tampak seperti pada gambar diatas akan terlihat tiga bagian yaitu, bagian nama obyek yang terletak di sebelah kiri atas, bagian event obyek yang terletak di sebelah kanan atas dan bagian code editor di sebelah kanan bawah.

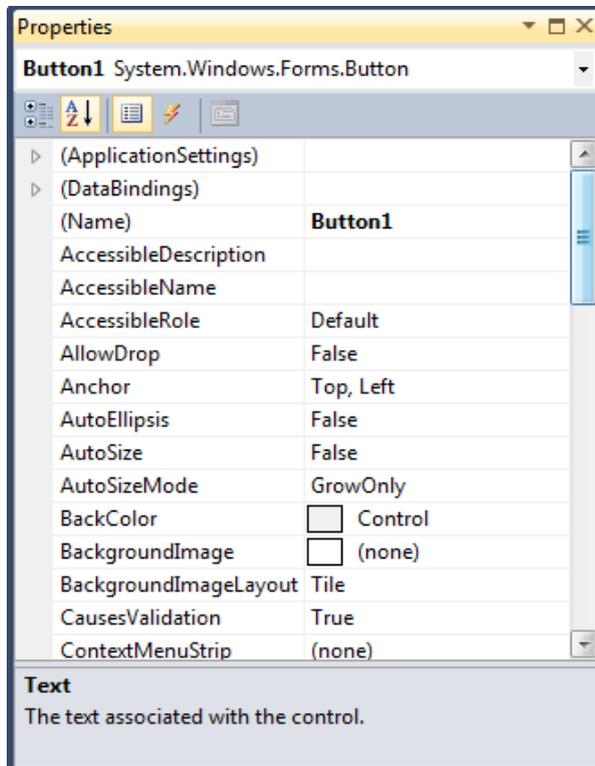
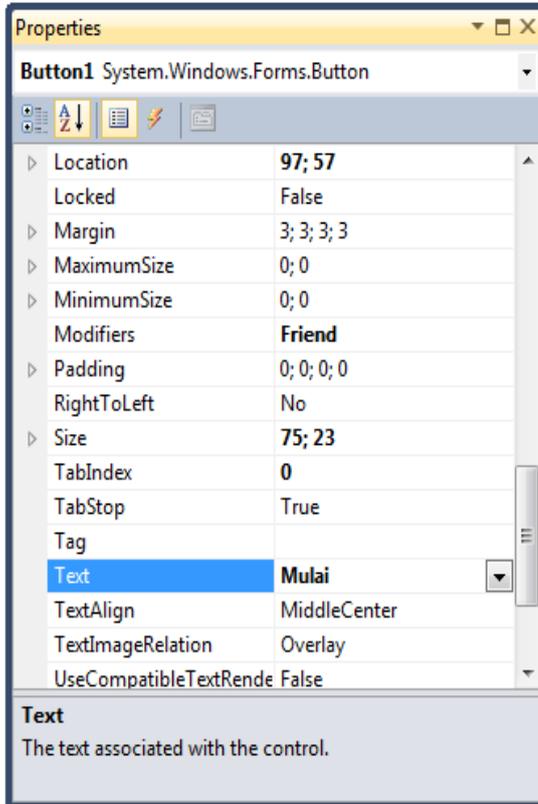
2.2.4 Mengatur Control Properties menggunakan Properties Window

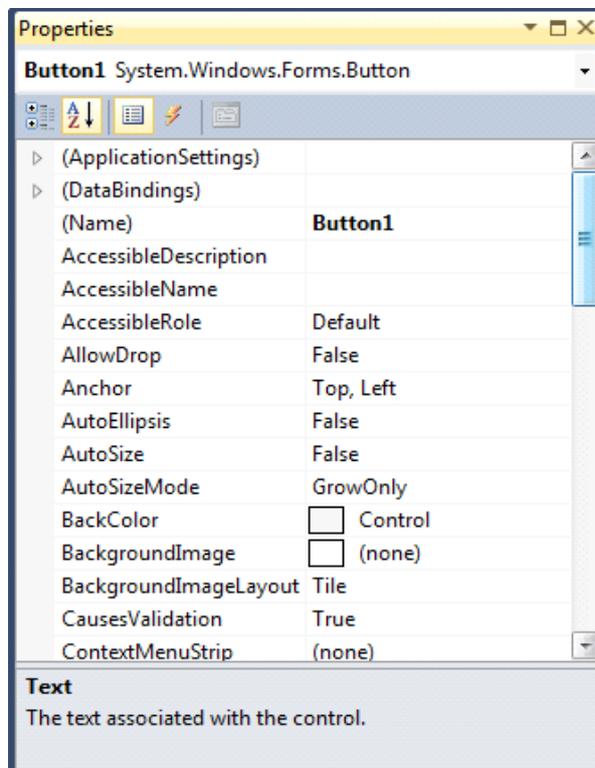
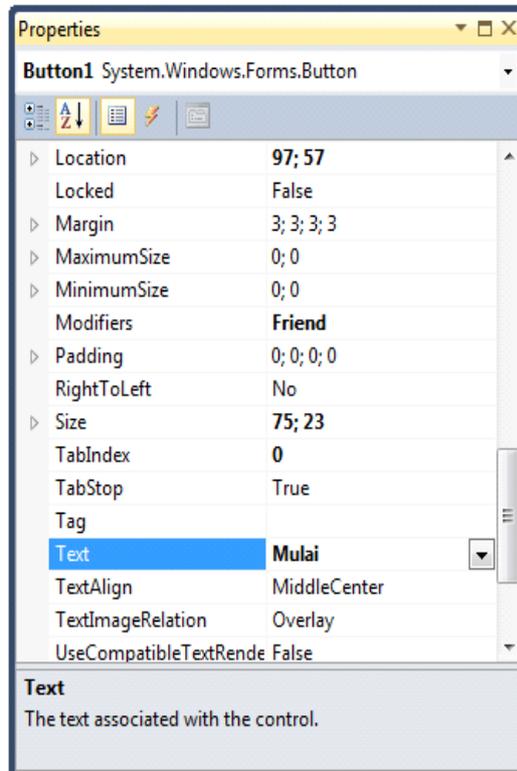
Windows Properties adalah jendela yang berisi daftar properties suatu objek yang kita pilih. Properties adalah sifat-sifat suatu objek, misalnya ukuran, judul, warna dll. Sehingga Window Properties digunakan untuk memberi atau merubah karakteristik suatu objek yang terdapat pada form.

Setiap komponen kontrol pada form aplikasi, perlu diatur karakteristiknya pada **Properties Window**. Disamping kita dapat mengatur karakteristik setiap komponen pada *Properties Window* kita juga dapat mengaturnya dengan menuliskan *script* pada program. Tetapi sebelumnya lebih baik jika belajar bagaimana mengatur karakteristik komponen kontrol pada *Properties Window* terlebih dahulu.

Pada *properties window* kita dapat melihat list detail dari tiap komponen kontrol. Sebagai contoh pada gambar 5 diletakan komponen kontrol **Button**, tampak pada *properties windows* detail karakteristik dari komponen *Button* tersebut. Terlihat pada gambar 5 *default* nama dari komponen *button* tersebut adalah **Button1** dan *text* yang tampil juga *Button1*. Kita dapat merubah tampilan *text* pada komponen *Button* tersebut dengan mengganti tulisan *text* pada *properties window* (contoh *text* = mulai).

Secara umum tampilan list pada *properties windows* terdiri dari dua buah kolom, kolom sebelah kiri menunjukkan karakteristik dari komponen dan kolom sebelah kanan menunjukkan nilai dari karakteristik tersebut. Nilai dari karakteristik tersebut dapat kita rubah dengan mengganti pilihan **option** atau dengan menuliskan secara spesifik karakteristik komponen tersebut seperti contoh di atas yaitu merubah nilai **text** dari **Button1**.



Gambar 2.6 Jendela *Properties Window*

2.2.5 Properti Default

Setiap obyek memiliki properti default, artinya properti yang otomatis akan diset dalam penulisan program, tanpa menuliskan jenis propertinya sebagai contoh :

`Text1 = "Hallo"` → maka properti yang diatur pada komponen **Text1** adalah text-nya.

Dengan mengetahui properti default, maka akan membuat penulisan program menjadi lebih efisien. Berikut ini adalah properti default untuk beberapa obyek yang sering digunakan :

`TextBox -> text`

`Label -> caption`

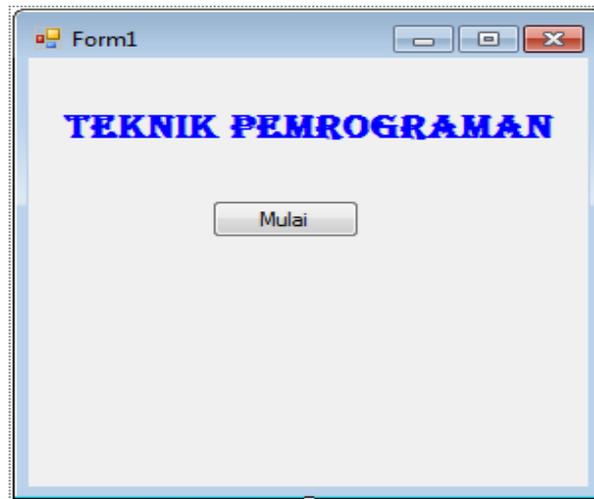
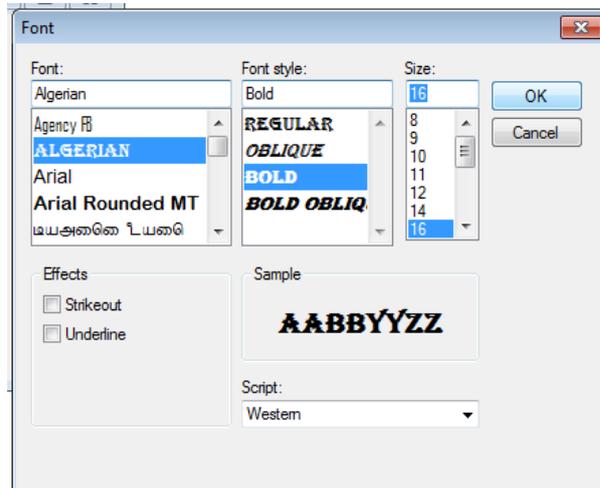
`Optionbutton -> value (true/false)`

`Checkbox -> value`

`Combo -> text`

Merubah Properties

- Klik Label pada toolbox dan letakkan pada Form
- Klik Window Properties kemudian geser scroll ke bawah hingga pada menu Text. Kemudian gantilah teks Label1 dengan tulisan "Teknik Pemrograman".
- Geser scroll pada Window Properties hingga ke menu **Font**. Tampilan *default* dari menu Font adalah Microsoft Sans Serif. Klik kotak kecil di ujung kanan sehingga muncul jendela baru dan rubahlah jenis font nya menjadi ALGERIAN, *font style* Bold, ukuran 16, seperti tampak pada gambar 6



Gambar 2.7 Mengatur Properties komponen Label

- Kemudian pilihlah menu ForeColor pada Properties Window sehingga muncul jendela baru untuk mengatur warna. Pilih Custom, dan pilihlah warna yang dikehendaki.

2.2.6 Menjalankan Aplikasi

Untuk mengetahui apakah aplikasi berjalan sesuai dengan rancangan awal, perlu dilakukan uji coba dengan menjalankan aplikasi tersebut dari awal. Proses running aplikasi dapat dilakukan dengan dua cara yaitu dengan menjalankan secara keseluruhan menggunakan menu **Debug -> Start Debugging** atau bisa juga menggunakan tombol *shortcut* **F5**.

Selain itu aplikasi ini juga bisa dijalankan per baris kode program melalui menu Debug -> Step Info atau dengan menggunakan tombol *shortcut* **F8**.

2.3 Membuat Program Visual Basic Yang Pertama

Pada sub materi ini, akan dipelajari bagaimana membuat aplikasi sederhana menggunakan komponen kontrol dari Visual Studio toolbox. Disamping itu akan dipelajari juga bagaimana mengatur *setting properties* dan mengidentifikasi penulisan kode program yang sesuai.

Langkah-langkah membuat program aplikasi :

1. Buat form yang menjadi dasar antarmuka aplikasi.
2. Gambarkan objek atau kontrol yang akan mengisi form tersebut.

Cara menggambar suatu objek atau kontrol dengan menggunakan toolbox adalah sebagai berikut:

- a. Klik kontrol yang akan digambar.
- b. Arahkan kursor mouse ke form. Kursor mouse akan berubah bentuk dari panah menjadi tanda plus.
- c. Tentukan letak kontrol pada form.
- d. Drag mouse sampai ukuran kontrol sesuai dengan keinginan (drag artinya menggeser mouse setelah tombol kiri di klik).
- e. Lepas tombol kiri mouse.

Cara lain untuk menggambar suatu kontrol adalah dengan mengklik dua kali kontrol yang akan digambar. Secara otomatis kontrol akan muncul di tengah form, selanjutnya kontrol ini dapat digeser dan disesuaikan lokasi maupun ukurannya.

Penggunaan Kontrol :

1. Command Button

Tool ini merupakan kontrol yang paling sering digunakan. Tombol ini digunakan untuk memulai, menghentikan atau mengakhiri suatu proses.

Property Command Button

Text	Menampilkan tulisan (tipe data string) di atas tombol.
Font	Mengatur <i>font type, style dan size</i> .
Enable	Menunjukkan kondisi pengaktifan setelah tombol dieksekusi atau diklik.
Back Color	Pilihan warna dari tombol.
Name	Merupakan nama dari tombol tersebut.

Event Command Button

Click	Event dipicu saat tombol dipilih baik dengan mengklik maupun dengan menekan <i>access key</i>
-------	---

2. Label Box

Label Box merupakan kontrol yang digunakan untuk menampilkan text yang tidak dapat diedit oleh *user* secara langsung. Tetapi property text dari label box dapat diubah saat run-time untuk memberi tanggapan pada suatu event.

Property Label

Text	Menampilkan tulisan (tipe data string) di dalam border.
Enable	Bila bernilai true, label akan tampil sesuai dengan pengaturan awal/aktif. Bila bernilai false, label akan tampak non aktif.
Border Style	Menentukan tipe border.
Font	Mengatur <i>font type, style, size</i> .
Visible	Bila bernilai true label akan ditampilkan pada form. Jika bernilai false label akan disembunyikan dan tidak tampak pada form aplikasi.

Event Label

Click	Event dipicu saat user mengklik label.
DbClick	Event dipicu saat user men double klik label.

Untuk program kita akan menggunakan dua kontrol di atas.

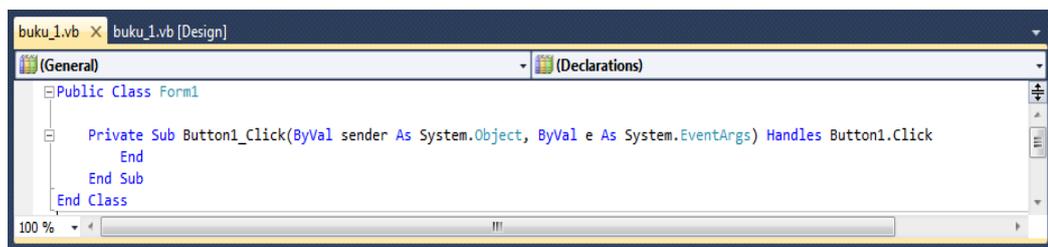
KONTROL	PROPERTIES
Form	Text : Belajar Teknik Pemrograman
Label	Text : Belajar Teknik Pemrograman Visual Basic 2010 Text allign : Middle centre Font : ALGERIAN Font style : Bold Size : 20
Command Button	Name : keluar Text : keluar



Gambar 2.8 Tampilan Awal Program

Kemudian arahkan kursor pada command button lalu klik kanan sehingga muncul Menu selanjutnya pilih View code.

Untuk memberikan respon keluar dari program setelah diklik, maka kita harus menuliskan kode untuk command button.



Gambar 2.9 Tampilan View Code

TUGAS

1. Buatlah suatu project sederhana seperti gambar 7 diatas dan perhatikan setiap langkah langkah pembuatan program secara seksama!
2. Perhatikan properties setiap komponen dan perhatikan setiap perubahan yang terjadi ketika properties dari suatu komponen di ubah!
3. Ubahlah properties dari label program tersebut dengan mengubah-ubah font, colour, size serta warna background warna form dari aplikasi tersebut!

TES FORMATIF

1. Jelaskan apa arti dan fungsi toolbox berikut :
 - Button - RichTextButton
 - Label - ProgressBar
 - ComboBox - PictureBox
2. Jelaskan pengertian dari Form Editor dan Code Editor !
3. Terangkan secara padat, jelas dan lengkap empat proses dasar pembuatan program aplikasi dengan menggunakan Visual Basic 2010!
4. Command Button dan Label Box merupakan dua buah kontrol yang sering digunakan dalam pembuatan program. Sebutkan properti yang terdapat dalam Command Button dan Label Box.



Jawab :

KEGIATAN 3 : Membuat Program Dengan Visual Basic

Sebelum kita praktekkan untuk membuat program menggunakan VB terlebih dahulu perlu diketahui beberapa komponen atau kontrol-kontrol standard yang akan sering digunakan dalam pembuatan program nantinya. Juga tentang hal-hal yang berkenaan tentang objek-objek tersebut.

3.1 Obyek-Obyek Standard dalam Visual Basic

Setiap obyek atau komponen memiliki fungsi dan karakteristik tertentu. Dalam Visual Basic, terdapat beberapa obyek yang sering digunakan dalam program. Obyek-obyek ini dapat dikatakan obyek standard Visual Basic.

Tabel berikut menunjukkan daftar obyek yang sering digunakan, fungsi masing-masing obyek dan juga properti sering dimanipulasi.

OBJEK	FUNGSI	PROPERTIES YANG SERING DIATUR
Form	Menjadi tempat mendesain program Anda.	Name, Text
Label	Menampilkan teks (informatif)	Name, Text, Font
Text Box	Menampung input untuk proses tertentu	Name, Text, Font
Frame	Mengelompokkan beberapa komponen dalam satu ruang tersendiri.	Name, Text, Font
Command Button	Tombol untuk memulai proses tertentu.	Name, Text, Font, Enabled, Default
Check Box	Menampilkan beberapa pilihan/opsi dimana dapat dipilih lebih dari satu opsi.	Name, Text, Font, Value
Open Button	Sama dengan CheckBox namun hanya ada satu pilihan yang diijinkan.	Name, Text, Font, Value

Combo Box	Menampilkan banyak data yang akan ditampilkan apabila diklik.	Name, List, Font
ListBox	Sama dengan Combo Box, namun semua data akan tampak.	Name, list, Font
Image	Menampilkan gambar.	Name, Picture,
Picture Box	Sama dengan image, hanya tidak memiliki properti stretch.	Name, Picture
Timer	Menghitung waktu (milidetik)	Name, Interval

3.2 Pemberian Awalan Nama Obyek

Dalam memberikan nama pada sebuah obyek sebaiknya konsisten sesuai dengan jenis obyek tersebut. Hal ini selain akan terlihat lebih rapi, juga akan memudahkan dalam penulisan, editing dan debugging program. Contoh, TxtBilangan1, CmdKeluar, OptUkuran, dll.

Berikut adalah contoh pemberian nama obyek yang sering digunakan

JENIS OBYEK	NAMA AWALAN OBYEK
Form	Frm
Label	Lbl
TextBox	Txt
CommandButton	Cmd
OptionButton	Opt
CheckBox	Chk
ComboBox	Cbo
ListBox	Lst
Menu	Mnu
Timer	Tim

3.3 Mengunci Kontrol

Mengunci kontrol adalah cara yang dilakukan agar terhindar dari ketidak sengaja memindahkan atau mengubah ukuran kontrol. Ketika menggambar kontrol bisa saja terjadi salah pencet yang mengakibatkan bergesernya suatu kontrol. Untuk menghindari hal-hal tersebut maka kontrol perlu dikunci agar

terhindar dari pemindahan atau pengubahan ukuran yang tidak disengaja. Penguncian ini berdampak pada seluruh kontrol dalam form yang saat ini sedang aktif. Kontrol pada form lain yang tidak aktif tidak akan dipengaruhi oleh perintah ini.

Cara untuk mengunci kontrol adalah:

1. Pilih form dimana kontrolnya akan dikunci.
2. Pilih menu Format, pilih Lock Control

3.4 Pembuatan Program

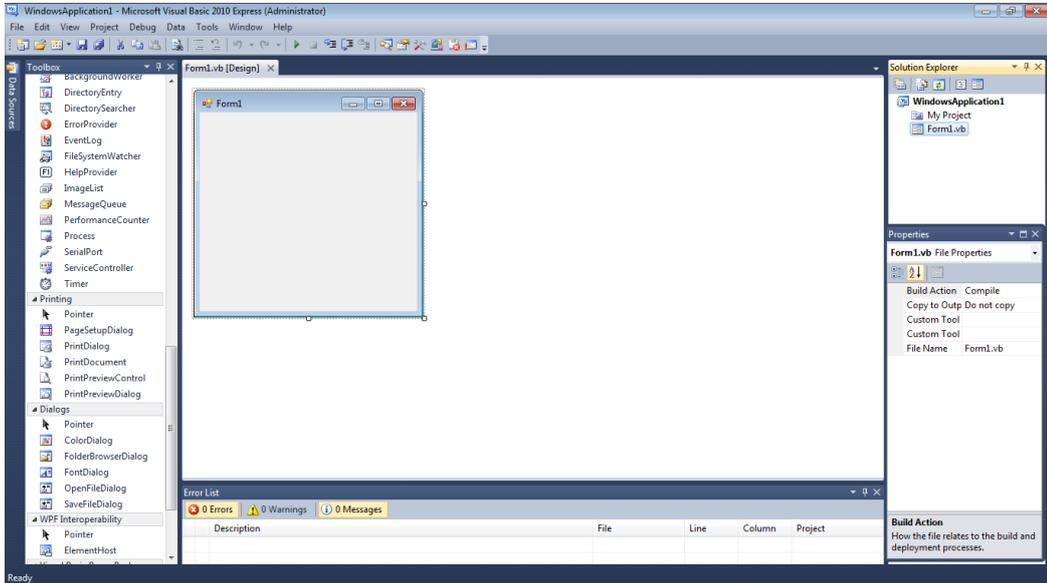
Ada beberapa tahapan untuk membuat program dalam Visual Basic yaitu:

1. Membuat antarmuka (interface)
2. Mengatur property
3. Menuliskan kode program

3.4.1 Membuat Antarmuka (Interface)

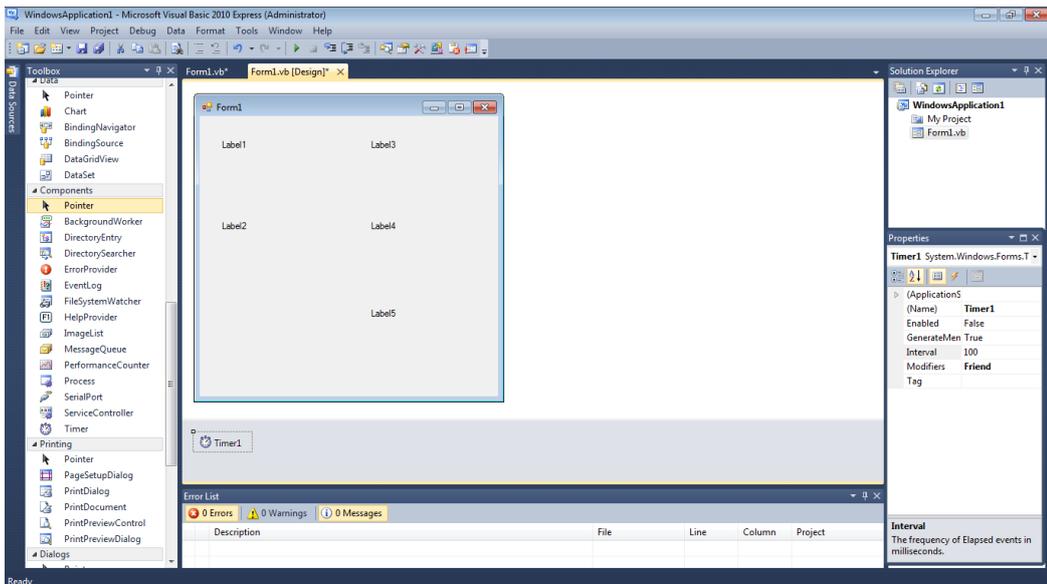
Langkah awal dalam pembuatan aplikasi Visual Basic adalah membuat form-form yang menjadi dasar antarmuka aplikasi. Kemudian menggambar oyek-obyek yang mengisi antarmuka dari form tersebut. Dalam kasus ini kita akan membuat program kalender dari *time display*. Langkah-langkahnya adalah sebagai berikut :

1. Klik Visual Basic 2010 kemudian klik New Project dan pilih Windows Form Application sehingga muncul tampilan dibawah ini



Gambar 3.1 Persiapan Membuat Antarmuka

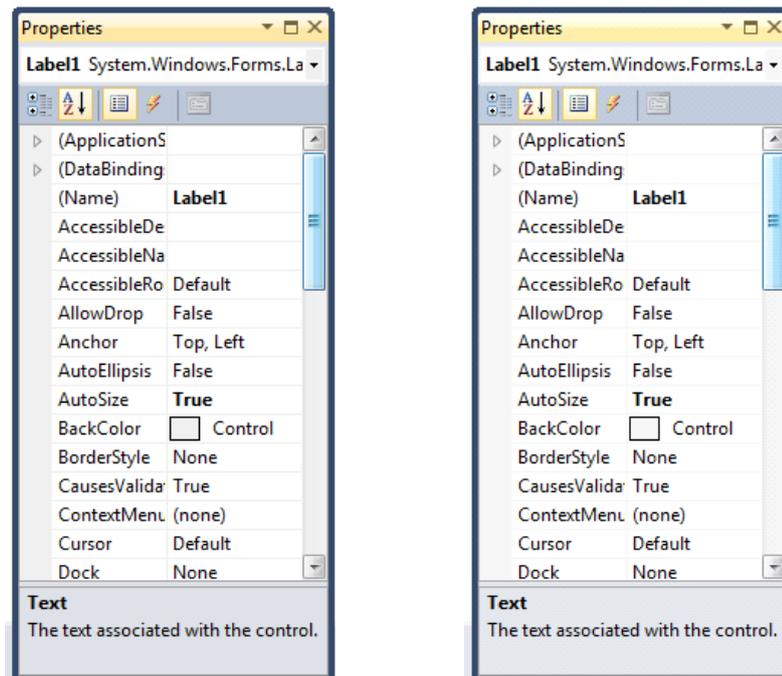
2. Tambahkan beberapa Label dan kontrol Timer ke dalam Form



Gambar 3.2 Mendesain Form

3.4.2 Mengatur Property

Langkah selanjutnya adalah mengatur property untuk obyek-obyek yang telah dibuat. Window property terdiri dari:



Gambar 3.3 Pengaturan Properties

- Daftar Obyek

Menampilkan obyek-obyek yang dapat diatur nilai propertinya. Klik anak panah (segitiga yang menghadap ke bawah) untuk menampilkan daftar keseluruhan obyek pada form. Setiap Obyek mempunyai nama dan jenis obyek, nama obyek dapat diatur dari properti Name.

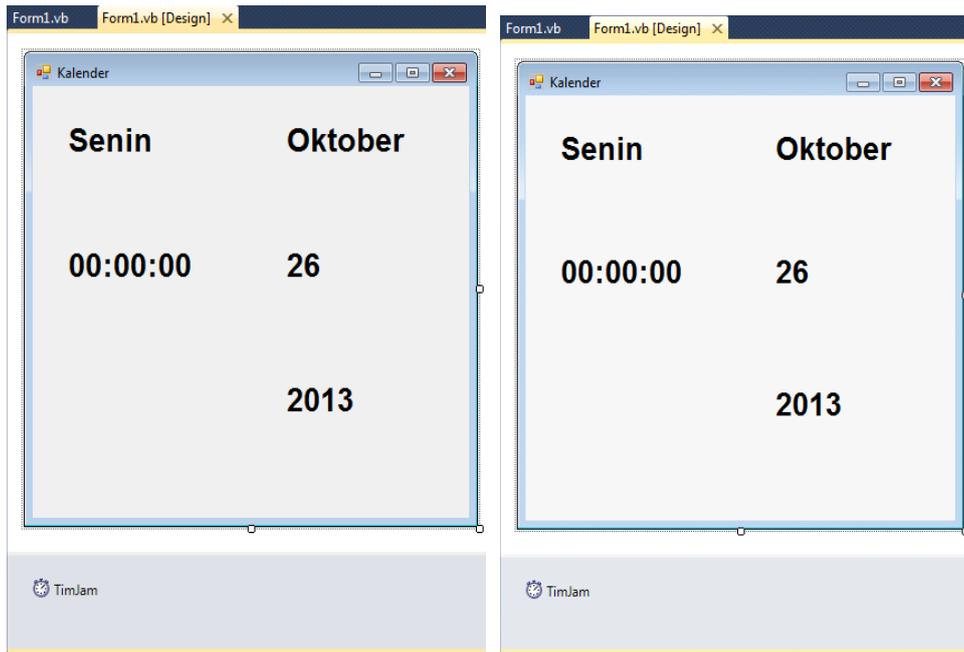
- Pengelompokkan Properti

Dalam tab Alphanumeric daftar properti ditampilkan dalam urutan abjad. Dalam tab Categories daftar property dikelompokkan menurut kriteria tertentu misal pengelompokkan berdasar appearance, behavior, font, position dll.

- Daftar Property

Kolom sebelah kiri menampilkan daftar properti dan kolom sebelah kanan berisi nilai dari properti tersebut. Nilai ini dapat diganti, ada yang bisa diisi bebas dan ada yang berupa pilihan nilai.

Pada program kalender di atas pengaturan Propert adalah sebagai berikut :



Gambar 3.4 Tampilan Form Akhir

OBJEK	PROPERTI
Form1	Name = FrmKalender Text = Kalender Border Style = 1-Fixed Single
Label1	Name = LblHari Font Style = Bold Text = Senin Font Size = 20 FontName = Arial
Label2	Name = LblHari Font Style = Bold Text = 00:00:00 Font Size = 20 FontName = Arial

Label3	Name = LblBulan Text = Oktober FontName = Arial	Font Style = Bold Font Size = 20
Label4	Name = LblTanggal Text = 26 FontName = Arial	Font Style = Bold Font Size = 20
Label5	Name = Timer1 Interval = 1000	

3.4.3 Menuliskan Kode di File Form

Ketika kita membuat aplikasi program, sebenarnya secara fisik kita membuat dua jenis file di folder project. File pertama adalah file yang bertugas menentukan spesifikasi desain antarmuka pengguna dari form tersebut dan yang kedua adalah file yang menyimpan kode-kode program yang dibuat oleh programmer.

Misal, ketika kita membuat aplikasi kalender seperti di atas ada dua file yang dibuat. File pertama disebut Form1.Desainer.vb dan yang kedua adalah Form1.vb. File pertama berisi desain antarmuka aplikasi Kalender dan file kedua berkaitan dengan kode-kode yang ditulis untuk menangani event dari program aplikasi tersebut.

Menuliskan Kode untuk Program Kalender :

- Arahkan kursor pada form, klik kanan dan pilih menu View Code.
- Akan terbuka tab menu Form1.vb. Form ini merupakan tempat untuk menuliskan kode-kode program yang berkenaan dengan aplikasi kita.
- Antara baris sub dan end sub ketiklah

```
PrivateSub TimJam_Tick(ByVal sender As
System.Object, ByVal As System.EventArgs)
Handles TimJam.Tick
    LblHari.Text = Format(Now, "dddd")
    LblWaktu.Text = Format(Now, "hh:mm:ss")
    LblBulan.Text = Format(Now, "MMMM")
    LblTahun.Text = Format(Now, "yyyy")
EndSub
```

Menyimpan Program

Setelah menyelesaikan program kalender, kita harus menyimpan dalam komputer. Visual Basic menyimpan kode dan oyek form ke dalam suatu file dan daftar dari project pada file yang lain.

Langkah-langkah untuk menyimpan pada program Visual Basic:

- Klik menu File pilih Save Form1.vb As....
- Tentukan lokasi penyimpanan file. Sebaiknya letakkan file hasil pekerjaan Anda ke dalam folder tersendiri.
- Beri nama form Anda dengan nama FrmKalender.vb
- Tutup jendela program aplikasi Visual Basic 2010 sehingga muncul jendela untuk menyimpan project yang telah dibuat.
- Pilih Save Project As dan beri nama PrjLatihanVB.vbp
- PERHATIAN : Visual Basic akan selalu menyimpan hasil pekerjaan Anda dalam bentuk project (tiap project bisa terdiri dari lebih dari satu form) sehingga langkah penyimpanan awal akan diminta untuk mengetikkan nama form dan nama project. Untuk latihan selanjutnya Anda cukup membuka file project yang telah dibuat dan menambahkan form baru dengan cara memilih menu bar project → add Windows Form. Jangan membuat New Project jika tidak perlu.

TUGAS

Setelah mempelajari bagaimana membuat program kalender, sekarang cobalah untuk membuat program Stopwatch.

Label label yang ditampilkan adalah Waktu Mulai, Waktu Berhenti, Lama Waktu. Command Buttonnya adalah Start dan Stop.

TES FORMATIF

1. Apakah tujuan dari pemberian nama pada suatu Obyek? Sebutkan beberapa nama obyek yang biasa digunakan dalam pembuatan program Visual Basic!
2. Sebutkan dan jelaskan dengan terperinci langkah-langkah dalam pembuatan program Visual Basic
3. Visual Basic memiliki beberapa komponen yang sering digunakan dalam pembuatan program, sebutkan lima komponen tersebut dan jelaskan fungsi dari masing-masing komponen beserta properti nya!

Jawab :

KEGIATAN 4 : Memahami Konsep Dasar Visual Basic

Setelah kita mempelajari bagaimana mendesain antarmuka (interface) aplikasi, menambahkan kontrol dan mengatur properties kontrol. Kita juga telah belajar menulis program secara sederhana meskipun belum memahami konsep penulisan program sebenarnya. Untuk itu sekarang kita akan mempelajari bagaimana dasar konsep pemrograman Visual Basic 2010 dan teknik menuliskan kode program.

VB 2010 adalah pemrograman berorientasi obyek dan berorientasi event (kejadian). Pada realitanya semua jendela aplikasi adalah event (kejadian) artinya pengguna menentukan apa yang harus dilakukan oleh program aplikasi, apakah mereka ingin mengklik suatu tombol, menuliskan teks pada textbox atau menutup program aplikasi atau yang lainnya. Sebuah event (kejadian) selalu dihubungkan dengan sebuah obyek dan akan menyebabkan kejadian atau aksi yang akan dijalankan oleh aplikasi.

4.1 Memahami Struktur Kode dari Suatu Prosedur Event (Kejadian)

Untuk menuliskan kode program, kita terlebih dahulu harus mengerti struktur penulisan program.

```
Private Sub...  
  
    Statements  
  
End Sub
```

Kita harus menuliskan kode program kita diantara Private Sub dan End Sub.

```
Private Sub...  
  
    Statements  
  
End Sub
```

Ada beberapa variasi struktur program

1. Public Sub

Tuliskan kode program disini

End Sub

2. Sub

Tuliskan kode program disini

End Sub

3. Function

Tuliskan kode program disini

End Sub

Coba kita tuliskan kode berikut

```
Private Sub Form1_Load(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles MyBase.Load  
  
    Me.Text="Latihan Pemrograman Visual Basic 2010"  
    Me.ForeColor = Color.Blue  
  
    Me.BackColor = Color.Cyan  
  
End Sub
```

Ketika program tersebut dijalankan dengan menekan tombol F5, maka akan muncul jendela seperti tampak pada gambar berikut :



Gambar 4.1 Tampilan Pengaturan WarnaForm

Baris pertama dari kode program akan mengganti judul dari form menjadi "Latihan Pemrograman Visual Basic 2010". Baris kedua akan merubah warna dari tulisan label1 menjadi biru. Baris ketiga akan merubahwarna latar dari form menjadi warna cyan. Tanda sama dengan pada program menunjukkan pengaturan properties dari suatu obyek, sebagai contoh warna biru obyek label1 pada form1 dan warna cyan pada obyek latar form1. **Me** adalah nama yang diberikan pada form1 class. Kita dapat juga menyebut semua baris program tersebut sebagai *statement*. Oleh karenanya aksi/eksekusi dari setiap program tergantung dari statement yang dituliskan oleh programmer.

Berikut ini contoh lain dari sebuah program sederhana :

```
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

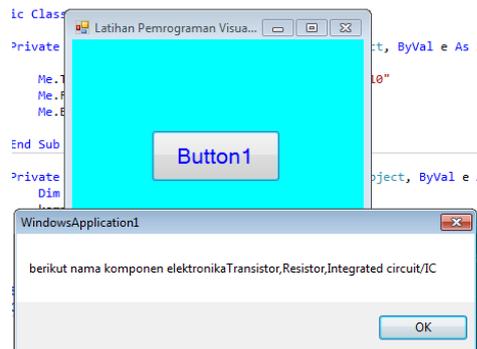
    Dim komponen1, komponen2, komponen3 As String

    komponen1 = "Transistor" komponen2 = "Resistor"
    komponen3 = "Integerated Circuit/IC"

    MsgBox ("Berikut nama komponen elektronika " & komponen1
    & " , " & komponen2 & " and " & komponen3)

End Sub
```

Pada program diatas, dimasukkan sebuah command button dengan nama Button1. Kode **Dim** berfungsi untuk mendeklarasikan variable komponen1, komponen2, komponen3 sebagai variable bertipe String, yang artinya mereka hanya bersifat sebagai variable penampung data teks (bukan angka). Fungsi **MsgBox** untuk menampilkan jendela berisi pesan teks dan isi dari variabel yang dihubungkan dengan tanda "&" seperti tampak pada gambar berikut :



Gambar 4.2 Tampilan MsgBox

4.2 Prosedur *Function* Visual Basic 2010

Function mirip dengan sebuah prosedur normal dengan tujuan utama dari sebuah fungsi adalah untuk menerima masukan tertentu dan mengembalikan nilai yang diteruskan ke program utama untuk menyelesaikan eksekusi.

Ada dua jenis fungsi dalam Visual Basic 2010 yaitu fungsi *Built In* (fungsi internal) dan fungsi-fungsi yang dibuat oleh programmer.

Syntax umum dari sebuah Function adalah

```
FunctionName (Argumen)
```

Argumen merupakan nilai yang diteruskan ke fungsi. Pada sub bab ini kita akan mempelajari dua buah fungsi internal dasar Visual Basic 2010 yaitu fungsi `MsgBox ()` dan `inputBox ()`.

a. Fungsi MsgBox ()

Kegunaan dari sebuah MsgBox adalah untuk menampilkan *message box* (kotak pesan) dan meminta pengguna untuk mengklik tombol perintah sebelum mereka dapat melanjutkan.

Sintaks dari MsgBox () adalah :

```
yourMsg = MsgBox(Prompt, Style Value, Title)
```

Pada argumen pertama yaitu Prompt, akan menampilkan pesan pada message box, Style Value akan menentukan tipe dari tombol perintah yang muncul pada message box, sedangkan Title akan menampilkan judul dari kotak pesan. Tabel dibawah ini jenis dari tombol perintah yang ditampilkan pada tampilan message box :

Style Value	Nama Konstanta	Tampilan Tombol
0	vbOkOnly	Tombol OK
1	vbOKCancel	Tombol OK dan Cancel
2	vbAbortRetryIgnore	Tombol Abort, Retry, Ignore
3	vbYesNoCancel	Tombol Yes, No dan Cancel
4	vbYesNo	Tombol Yes dan No
5	vbRetryCancel	Tombol Retry dan Cancel

Kita dapat menggunakan konstanta bernama untuk bilangan bulat pada argumen kedua agar program lebih mudah dibaca. Visual Basic 2010 secara otomatis akan muncul daftar nama konstanta dan Anda dapat memilih salah satu diantaranya.

Contoh :

```
yourMsg = MsgBox ("Klik OK Untuk Melanjutkan", 1, "Startup Menu")
```

dan

```
yourMsg = Msg ("Klik OK Untuk Melanjutkan", vbOKCancel, "Startup Menu")
```

Fungsi dari kedua potongan script program diatas adalah sama. **yourMsg** adalah variabel yang menyimpan nilai-nilai yang dikembalikan oleh fungsi MsgBox (). Nilai-nilai tersebut ditentukan oleh jenis tombol yang diklik oleh pengguna. Nilai-nilai tersebut telah dideklarasikan sebagai tipe data Integer pada suatu prosedur.

Tabel dibawah ini menunjukkan nilai-nilai, nama konstanta dan tombol :

Nilai	Nama Konstanta	Tampilan Tombol
1	vbOK	Tombol OK
2	vbCancel	Tombol Cancel
3	vbAbort	Tombol Abort
4	vbRetry	Tombol Retry
5	vbIgnore	TombolIgnore
6	vbYes	TombolYes
7	vbNo	TombolNo

Contoh program :

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Dim testmsg As Integer

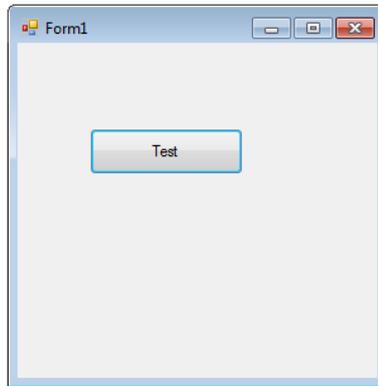
        testmsg = MsgBox("Click untuk test", 1, "Test
message")

        If testmsg = 1 Then
            MessageBox.Show("Anda telah menekan tombol
OK")
        Else
            MessageBox.Show("Anda telah menekan tombol
Cancel")
        End If

    End Sub

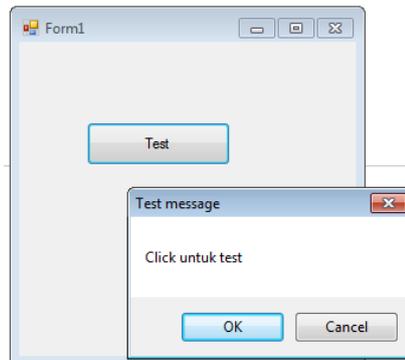
End Class
```

Buatlah project dan form baru yang terdiri dari satu buah tombol button kemudian ketiklah script program diatas. Debug atau jalankan program diatas sehingga muncul jendela aplikasi seperti berikut.



Gambar 4.3 Tampilan Form Latihan Button

Tampilan gambar diatas muncul ketika pertama kali program dijalankan. Pilih tombol button "Test" dan kliklah tombol tersebut sehingga muncul jendela baru seperti gambar dibawah ini.



Gambar 4.4 Tampilan MsgBox Latihan Button

Tampilan jendela baru diatas adalah representasi kode program baris ke tiga

```
testmsg = MsgBox("Click untuk test", 1, "Test mes-
sage")
```

Script diatas berfungsi untuk memunculkan function MsgBox() yang diberi judul **“Test message”** dan terdapat tampilan teks **“Click untuk test”**. Jika tombol **“OK”** ditekan maka nilai variabel **“testmsg”** akan bernilai **“1”** dan bernilai **“0”** jika tombol **“Cancel”** ditekan.



Gambar 4.5 Tampilan MsgBox Hasil Penekanan Tombol OK

Jendela di atas akan muncul jika kita menekan tombol **“OK”**. Tampilan jendela baru diatas adalah representasi kode program baris ke empat dan ketiga

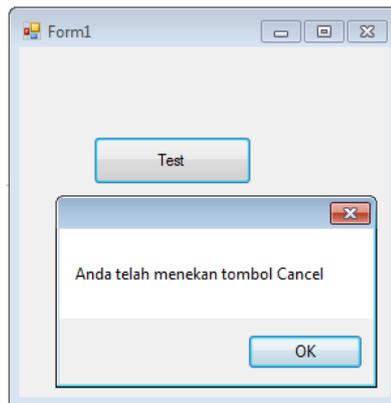
```
If testmsg = 1 Then
  MessageBox.Show("Anda telah menekan tombol OK")
```

Kode program diatas menunjukkan jika nilai variabel **“testmsg”** bernilai **“1”** maka akan memunculkan function MsgBox() yang memberi informasi **“Anda telah menekan tombol OK”**.

Jika tombol **“Cancel”** ditekan maka variabel **“testmsg”** akan bernilai **“0”** sehingga kode program berikut akan di eksekusi.

```
Else
  MessageBox.Show("Anda telah menekan tombol Cancel")
End If
```

Program diatas dieksekusi karena nilai “testmsg” tidak bernilai “1” sehingga program “Else” akan dijalankan dan akan muncul function MsgBox() yang memberi informasi “Anda telah menekan tombol Cancel” seperti tampak pada gambar dibawah ini.



Gambar 4.6 Tampilan MsgBox Hasil Penekanan Tombol Cancel

b. Fungsi InputBox ()

Fungsi InputBox () akan menampilkan sebuah kotak pesan jika pengguna dapat memasukkan sebuah nilai atau pesan dalam form dari teks. Pada Visual Basic 2010 Anda akan menemukan format seperti berikut :

```
myMessage = InputBox(Prompt, Title, default_text, x-position, y-position).
```

myMessage adalah variabel sebuah data yang dideklarasikan bertipe string, variabel ini berfungsi untuk menerima input pesan dari pengguna. Argumen dari sintak diatas dapat dijelaskan sebagai berikut :

- Prompt : pesan ditampilkan sebagai sebuah pertanyaan.
- Title : Judul dari Input Box

- Default text : Teks standar yang muncul pada kotak input dimana pengguna dapat menggunakannya sebagai sebuah masukan atau dapat juga dirubah menjadi pesan yang ingin dimasukkan oleh pengguna.
- x-position dan y-position : posisi atau koordinat dari input box.

Contoh program dari fungsi InputBox ():

```
Public Class input_box

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

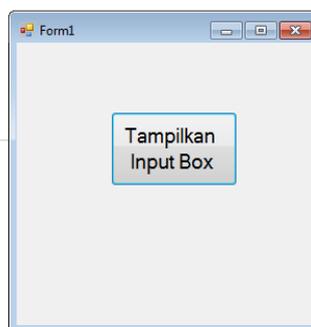
Dim Data_In As String

Data_In = Microsoft.VisualBasic.InputBox("Masukan
pesan/input anda!", "Message Entry Form", "Tulis
pesan/input anda disini", 200, 300)

If Data_In <> "" Then
    MessageBox.Show(Data_In, "input yang telah anda
masukan")
Else
    MessageBox.Show("Tidak ada pesan/input")
End If

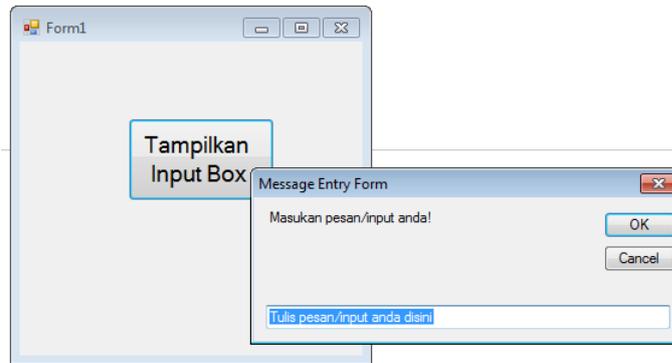
End Sub
End Class
```

Tambahkan form baru yang terdiri dari satu buah tombol button kemudian ketiklah script program diatas. Debug atau jalankan program di atas sehingga muncul jendela aplikasi seperti berikut.



Gambar 4.8 Tampilan Awal Aplikasi InputBox

Tampilan gambar diatas muncul ketika pertama kali program dijalankan. Pilih tombol button “Tampilkan input Box” dan kliklah tombol tersebut sehingga muncul jendela baru seperti gambar dibawah ini.

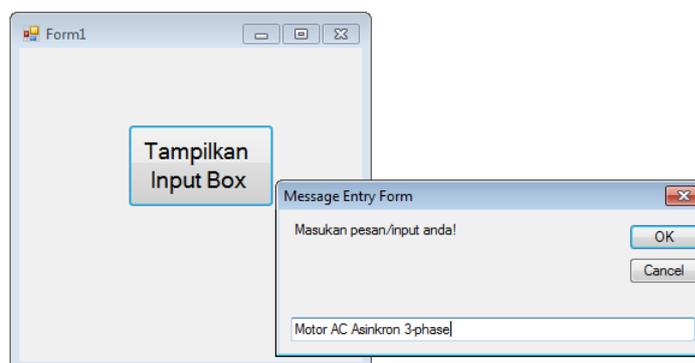


Gambar 4.9 Tampilan Aplikasi InputBox

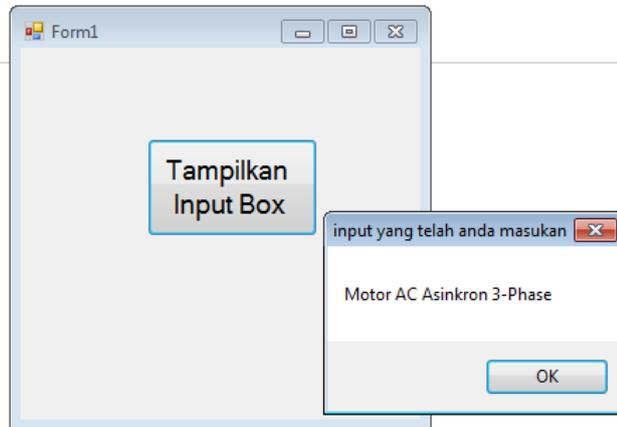
Tampilan jendela baru diatas adalah representasi kode program baris ke tiga

```
Data_In = Microsoft.VisualBasic.InputBox("Masukan pesan/input anda!", "Message Entry Form", "Tulis pesan/input anda disini", 200, 300)
```

Script diatas berfungsi untuk memunculkan function MsgBox() yang diberi judul “**Message Entry Form**” dan terdapat tampilan teks “**Masukan pesan/input anda!**”. Dibagian bawah dari jendela baru tersebut terdapat input box yang berisi tulisan “**Tulis pesan/input anda disini**”. Jika teks pada input tersebut dan kita ganti dengan teks kita sendiri (contoh : Motor AC Asinkron 3-Phase) dan tombol “**OK**” ditekan maka nilai variabel “**Data_In**” akan bernilai string teks “Motor AC Asinkron 3-Phase”.



Pada akhir kode program diatas, terdapat huruf 200,300 yang artinya tampilan jendela baru akan diletakan pada koordinat x=200 dan y=300. Sehingga ketika tombol “OK” kita tekan akan muncul jendela baru sebagai berikut:



Gambar 4.10 Tampilan Hasil Proses Aplikasi InputBox

Tampilan jendela baru MsgBox diatas merepresentasikan kode program baris ke tiga dan ke empat

```
If Data_In <> "" Then
    MessageBox.Show(Data_In, "input yang telah anda masukan")
```

Teks yang kita ketikkan ke dalam input box akan disimpan ke dalam variabel **Data_In** yang bertipe data string. Kode `If Data_In <> ""` artinya menanyakan kondisi variabel **Data_In**, jika nilai **Data_In** di isi nilai teks (tidak kosong) maka program akan mengeksekusi perintah dibawahnya, yaitu :

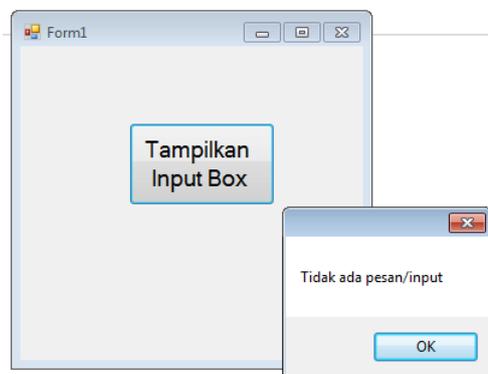
```
    MessageBox.Show(Data_In, "input yang telah anda masukan")
```

Potongan program diatas memiliki makna sebagai berikut, karena syarat terpenuhi (**Data_In** berisi teks) maka akan dimunculkan jendela baru MessageBox yang berjudul “**input yang telah anda masukan**” dan juga terdapat teks sesuai dengan **Data_In** (contoh : Motor AC Asinkron 3-Phase).

Jika input box kita kosongkan (tidak terisi satu huruf/karakter apapun) maka syarat di atas tidak akan terpenuhi. Sehingga program akan lompat ke baris enam sampai delapan.

```
Else  
  MessageBox.Show("Tidak ada pesan/input")  
End If
```

Potongan program di atas di eksekusi karena syarat dari program sebelumnya tidak terpenuhi. Sehingga akan memunculkan jendela MessageBox baru yang bertuliskan teks **“Tidak ada pesan/Input”**.



Gambar 4.11 Tampilan Tidak Ada Input Pada InputBox

4.3 Variabel dan Tipe Data Pada Pemrograman Visual Basic 2010

4.3.1 Mendeklarasikan Variabel

Variabel adalah lokasi penyimpanan data sementara dalam program kita. Kita bisa menggunakan satu atau banyak variabel pada kode program dan variabel dapat mengandung kata-kata, angka, tanggal atau properti. Variabel sangat berguna karena dengan variabel kita bisa menggunakan nama yang singkat dan mudah diingat untuk setiap data yang ingin digunakan. Variabel bisa menyimpan informasi yang dimasukkan pemakai pada saat program dijalankan, hasil dari perhitungan tertentu atau data yang ingin ditampilkan pada form kita. Jadi variabel adalah sarana yang sederhana yang bisa digunakan untuk menyimpan hampir semua jenis informasi.

Variabel harus dideklarasikan sebelum digunakan. Ketika mendeklarasikan variabel, ditentukan nama variabel dan tipe data yang akan digunakan oleh variabel tersebut. Dalam Visual Basic banyak perintah yang dipakai untuk mendeklarasikan suatu variabel. Tapi perintah yang sering digunakan adalah Dim. Dim kependekan dari Dimensi. Format standar perintah Dim adalah sebagai berikut :

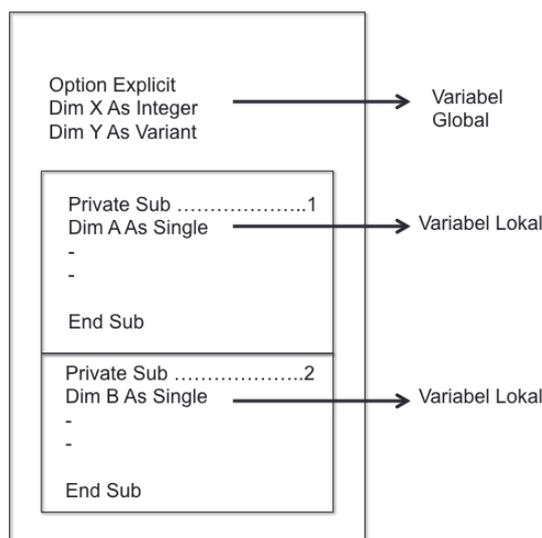
Dim NamaVar As TipeData

NamaVar adalah nama variabel yang dideklarasikan. TipeData adalah tipe data yang disimpan dalam variabel. Contoh deklarasi :

```
Dim Hasil As Integer
```

```
Dim Nama As String
```

Jika variabel diletakkan di dalam prosedur maka variabel tersebut hanya bisa digunakan oleh prosedur tersebut atau dikenal sebagai lokal variabel. Dan jika variabel dideklarasikan di awal modul (diluar prosedur tapi masih dalam area *Declaration*), maka variabel tersebut bisa digunakan oleh seluruh prosedur dalam modul tersebut. Jika deklarasi Dim diganti dengan Public, maka variabel ini bisa digunakan diseluruh bagian aplikasi (variabel-variabel semacam ini disebut variabel global).



Gambar 4.12 Bagian Penulisan Program Visual Basic

Variabel global rawan kesalahan. Bisa saja terjadi suatu variabel global mempunyai nilai tidak seperti yang diharapkan karena nilainya secara tidak sengaja telah diganti di suatu prosedur.

Nama variabel harus mengikuti aturan sebagai berikut :

- Menggunakan huruf dan angka
- Disamping angka atau huruf, karakter khusus misal “_” juga bisa dipergunakan
- Nama variabel minimum 1 huruf maksimum 255 huruf
- Tidak boleh ada spasi pada nama variabel

Jika suatu variabel langsung digunakan tanpa dideklarasikan terlebih dahulu maka akan terjadi kemungkinan :

1. Visual Basic menganggap variabel tersebut sebagai Variabel bertipe Variant, kecuali jika variabel diakhiri berikut :

Huruf Akhiran	Tipe Data	Contoh
%	Integer	Hasil%
&	Long	Penjualan&
!	Single	Netto!
#	Double	Volume#
@	Currency	Gaji@
\$	String	Nama\$

2. Penulisan beberapa variabel sekaligus dalam satu baris memungkinkan asalkan bertipe data sama, misalnya :

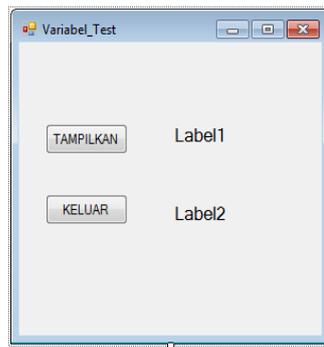
```
Dim Hasil, Bilangan1, Bilangan2 As Integer
```

Pada contoh di atas variabel, bilangan1 dan bilangan2 dideklarasikan sebagai variabel bertipe integer.

4.3.2 Penggunaan Variabel Dalam Program

Nilai pada suatu variabel bersifat sementara dan jika ada nilai baru maka nilai baru tersebut akan menggantikan nilai yang lama dari suatu operasi program. Contoh latihan dibawah ini menggambarkan bagaimana sebuah variabel dengan nama "LastName" diisi dengan suatu teks dan bagaimana teks tersebut ditampilkan.

1. Buatlah Project baru dan form baru pada Visual Basic 2010.
2. Berilah nama form1 dengan nama **Variabel_Test** dan buatlah tampilan dengan menggunakan beberapa komponen seperti tampak pada gambar berikut ini :



Gambar 4.13 Tampilan Program Aplikasi Variabel_Test

3. Double klik lah pada tombol **Tampilkan**
Even prosedur Button1_Click akan tampak pada code editor.
4. Ketiklah pernyataan dibawah ini untuk mendeklarasikan dan gunakan variabel dengan nama LastName :

```
Dim LastName As String  
  
LastName = "Hasanuddin"  
Label1.Text = LastName  
  
LastName = "Ayam Jantan dari Timur"  
Label2.Text = LastName
```

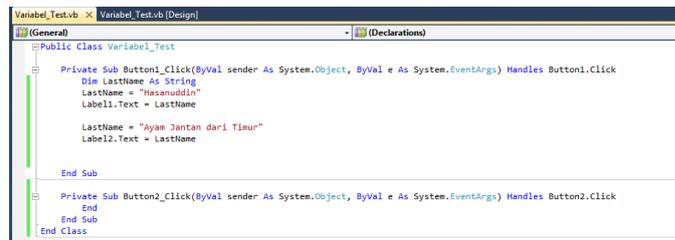
Pernyataan program diatas ditulis dalam tiga bagian, pernyataan pertama mendeklarasikan variabel LastName menggunakan pernyataan Dim dan tipe

data String. Pernyataan kedua dan ketiga menuliskan teks “Hasanuddin” pada variabel LastName, kemudian nama ini ditampilkan pada label pertama di jendela form.

Pada baris keempat, isi variabel LastName akan diganti dengan teks “Ayam Jantan dari Timur”. String pada teks kedua terlihat lebih panjang daripada string teks pertama dan berisi beberapa jarak spasi atau ruang kosong.

5. Kembalilah pada jendela Form, kemudian double klik pada tombol **Keluar**. Even prosedur Button2_Click akan muncul pada code editor. Ketiklah pernyataan “**End**” untuk menghentikan program

Maka tampilan program Anda akan terlihat seperti gambar :



```

Public Class Variabel_Test
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim LastName As String
        LastName = "Hasanuddin"
        Label1.Text = LastName

        LastName = "Ayam Jantan dari Timur"
        Label2.Text = LastName
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub
End Class

```

6. Klik tombol Save All pada toolbar untuk menyimpan program yang Anda buat.

7. Klik Start Debugging pada toolbar untuk menjalankan program.

8. Klik tombol Tampilkan

Program akan mendeklarasikan variabel dan mengisi nilai pada kedua variabel tersebut kemudian nilai dari dua variabel tersebut dikopikan pada label, oleh Label.Text akan ditampilkan seperti pada jendela aplikasi dibawah ini :



Gambar 4.14 Tampilan Hasil Aplikasi Variabel_Test

9. Untuk menghentikan program kliklah pada tombol Keluar.

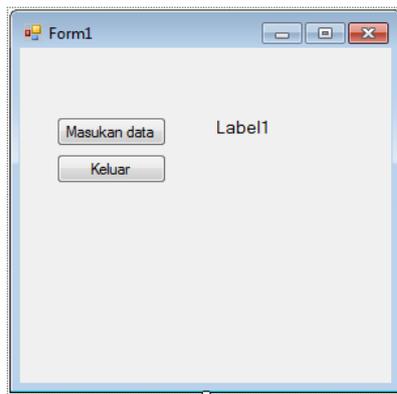
4.3.3 Penggunaan Variabel Untuk Input Data

Salah satu penggunaan dari variabel adalah untuk menyimpan informasi yang telah dimasukkan oleh pengguna. **InputBox** digunakan untuk menyimpan input data dari pengguna dan akan disimpan pada sebuah variabel.

Berikut contoh penggunaan variabel pada InputBox untuk mengolah teks yang telah diinput oleh pengguna.

Penggunaan fungsi dari InputBox :

1. Buatlah Project dan form baru pada menu File
2. Kliklah Form1.vb pada Solution Explorer dan pilih tombol View Designer.
3. Buatlah layout design interface yang terdiri dari dua buah tombol Button dan dua buah label seperti gambar dibawah ini



4. Double klik pada tombol **"Masukan data"**.
5. Ketikalah pernyataan dibawah ini untuk mendeklarasikan dua buah variabel dan memanggil fungsi dari InputBox :

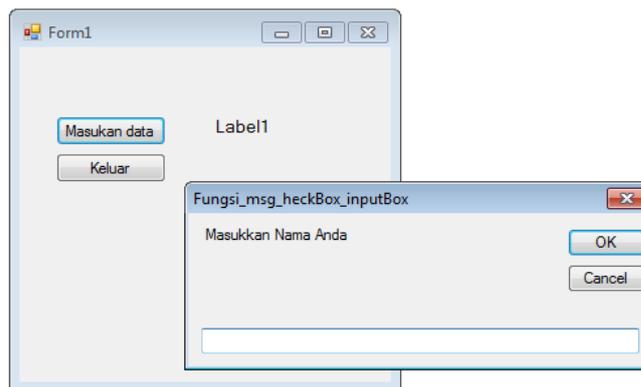
```
Dim Prompt, Nama As String  
  
Prompt = "Masukkan Nama Anda"  
  
Nama = InputBox(Prompt)  
  
Label1.Text = Nama
```

Potongan program diatas mendeklarasikan dua buah variabel dengan menggunakan pernyataan Dim yaitu Prompt dan Nama. Kedua variabel dideklarasikan menggunakan tipe data String. Baris kedua menunjukkan variabel Prompt berisi teks “**Masukan Nama Anda!**” bertipe string. Isi teks variabel prompt tersebut akan di tampilkan pada layar dialog inputBox.

Baris berikutnya disebut fungsi InputBox, InputBox merupakan fungsi dari Visual Basic yang menampilkan dialog box dan input teks pada layar. Input Teks ini nantinya akan diisikan data oleh pengguna dan data tersebut disimpan ke dalam variabel Nama. dan prompt dari pengguna untuk diinput.

Setelah data disimpan di variabel nama, kemudian program baris ke empat melakukan eksekusi mengopi isi data string yang tersimpan pada variabel nama untuk dituliskan pada properties **text** komponen **label1** yang terdapat pada form utama.

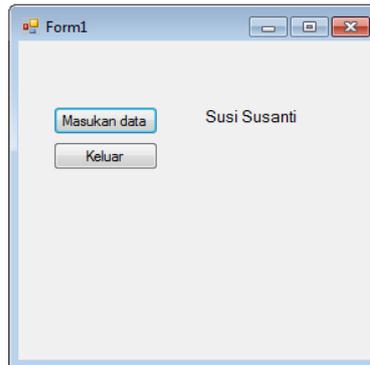
6. Simpanlah pekerjaan Anda lalu klik Start Debugging pada toolbar dan jalankan program tersebut.
7. Klik tombol Input Box. Visual Basic mengeksekusi Button1_Click pada event prosedur dan dialog box akan ditampilkan pada layar seperti pada gambar berikut :



Gambar 4.15 Tampilan InputBox Aplikasi Penggunaan Variabel

8. Ketiklah nama lengkap Anda dan klik OK

InputBox memasukkan nama Anda ke program dan menempatkannya pada variabel FullName. Program ini digunakan untuk menampilkan nama Anda di form seperti gambar dibawah ini :



Gunakan fungsi InputBox ini pada program kapanpun Anda inginkan untuk memberikan informasi kepada pengguna. Anda dapat mengkombinasikannya dengan komponen kontrol lain untuk memberikan input ataupun output pada program. Pada latihan selanjutnya Anda akan mengenal bagaimana menggunakan fungsi yang hampir serupa untuk menampilkan teks pada dialog box.

9. Klik tombol Keluar pada form untuk menghentikan atau keluar dari program.

4.3.4 Penggunaan Variabel Sebagai Output

Anda dapat menampilkan nilai dari variabel dengan menggunakan properti Text pada komponen properti lain. Contohnya dengan menggunakan komponen Label dan Rich Text, kita dapat mengecek nilai variabel tersebut dengan mengumpankan nilai variabel pada properti text dari Label atau Rich Text tadi.

Cara lain kita dapat mengumpankan nilai variabel tersebut ke sebuah argumen menggunakan Dialog Box. Salah satu kegunaan dari Dialog Box adalah untuk menampilkan output dari MsgBox. Ketika kita memanggil fungsi MsgBox, maka

64

akan muncul suatu jendela Dialog Box dengan berbagai macam variasi yang dapat kita atur lebih detail.

Sebagai contoh kita menggunakan Input Box, dimana akan diperlukan satu atau lebih suatu argumen sebagai input, dan hasil dari input tersebut akan disimpan pada suatu variabel.

Kode program untuk fungsi MsgBox adalah sebagai berikut :

```
Variabel = MsgBox(Prompt, Buttons, Title)
```

Dimana Prompt adalah teks yang akan dimunculkan pada message box, Buttons merupakan spesifikasi dari tombol, ikon, dan komponen lain untuk menampilkan message box dan Title adalah judul yang akan ditampilkan pada bagian atas jendela message box. Variabel merupakan tempat penyimpanan data teks yang di inputkan oleh pengguna.

Berikut ini contoh pembuatan program untuk menampilkan MsgBox :

1. Gunakan latihan program sebelumnya, lakukan double klik pada tombol "**Masukan Data**" pada form sehingga kursor akan menuju ke form Code Editor.
2. Pilih pernyataan pada even prosedur (baris terakhir dari program)

```
Label1.Text = Nama
```

Pernyataan ini akan menampilkan isi dari variabel Nama pada label.

3. Hapuslah pernyataan tersebut dengan menekan tombol DELETE.
4. Ketiklah pernyataan berikut ini pada code editor.

```
MsgBox(Nama, , "Hasil Masukan")
```

Ini merupakan pernyataan baru yang dinamakan fungsi MsgBox. MsgBox akan menampilkan isi dari variabel Nama pada dialog box dan menuliskan "Hasil Masukan" pada judul title bar. Code editor program Anda akan tampak seperti berikut :

```

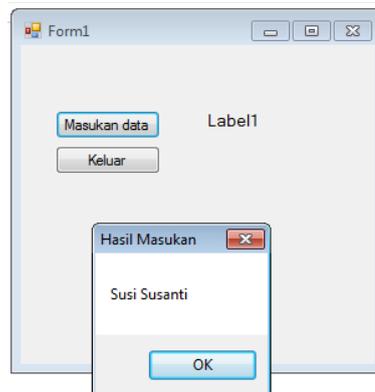
Form1.vb* - Form1.vb [Design]*
Button1 - Click
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Prompt, Nama As String
        Prompt = "Masukkan Nama Anda"
        Nama = InputBox(Prompt)
        MsgBox(Nama, , "Hasil Masukan")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub
End Class

```

5. Klik Start tombol Debugging pada toolbar untuk menjalankan program.
6. Klik tombol Input Box dan ketiklah nama Anda pada input box kemudian klik OK.

Visual Basic mengirimkan masukan tersebut pada variabel Nama dan menampilkannya pada message box. Setelah mengetik nama (contoh : Susi Susanti), maka akan tampak tampilan seperti berikut ini :



Gambar 4.16 Tampilan Hasil Aplikasi Variabel_Test

4.3.5 Tipe Data

Setiap aplikasi yang dibuat selalu berhubungan dengan data. Data itu banyak macam dan ukurannya. Ada yang berupa teks, angka, gambar, suara, video dan sebagainya. Dalam Visual Basic terdapat dua belas tipe data yaitu :

1. Tipe Data Numerik

Tipe data numerik dipakai bila kita menulis program yang berhubungan dengan perhitungan, harga, usia dan lainnya. Sehingga diperlukan pengetahuan untuk memilih tipe data yang tepat sesuai kebutuhannya.

Ada 3 macam tipe data numerik yaitu :

a. Tipe data Integer

Visual Basic membagi tipe data integer menjadi tiga macam yaitu

Tipe Data	Kebutuhan Memori	Jangkauan Nilai
Byte	1 byte	0 sampai 255
Integer	2 byte	-32.768 sampai 32.767
Long	4 byte	-2.147.483.648 sampai 2.147.483.648
Short	8 bit	-32,768...32.767
Ulong	64 bit	0... 18, 446, 774, 073, 709, 551,615
Longword	32 bit	0...4294967295

Satu byte adalah satu lokasi penyimpanan memori 8 bit

b. Tipe Data Desimal

Visual Basic mengelompokkan tipe data desimal sebagai berikut:

Tipe Data	Kebutuhan Memori	Jangkauan Nilai
Single	4 byte	-3.402823E+38 sampai -1.401298E-45 untuk negatif; 1.401298E-45 sampai 3.402823E+38 untuk positif
Double	8 byte	-1.79769313486232E +308 sampai 4.94065645841247E-324 untuk negatif ; 4.94065645841247E-324 sampai 1.79769313486232E untuk positif



Currency	8 byte	-922.337.203.685.477, 5805 sampai 922.337.203.685.477, 5805
Decimal	14 byte	±79.228.162.514.264.337.593.543.950.335 tanpa pecahan

Khusus untuk tipe data Currency, oleh Visual Basic sudah disiapkan untuk mata uang negara-negara yang disupport oleh Windows. Setting mata uang bisa dilakukan di Regional Setting di Control Panel.

2. Tipe Data Teks

Nama, alamat, dan lain lain adalah contoh data teks yang sering kita gunakan dalam program komputer. Kumpulan huruf (boleh kosong) yang dianggap sebagai satu kesatuan itu disebut String. Visual Basic menyediakan tipe data teks sebagai berikut :

Tipe Data	Kebutuhan Memori	Daya Tampung
String (panjang tetap)	1 karakter=1byte	1 sampai 65.400 karakter
String (panjang dinamis)	Pnjang karakter	0 sampai 2 milyar
Char	Terdiri dari 1 huruf / 1 angka (1 karakter)	

Literal string selalu diapit dengan tanda petik ganda ("). String dapat terdiri dari karakter apapun, contoh :

"Resistor" "4700" "Resistor ¼ watt 4k7 ohm"

3. Tipe Data Khusus

Tipe data lain yang terdapat dalam Visual Basic adalah sebagai berikut :

Tipe Data	Kebutuhan Memori	Daya Tampung
Date	8 byte	1 January 100 sampai 31 Desember 9999
Boolean	2 byte	True atau False
Object	4 byte	Sembarang obyek tempelan (embedded)
Varian (diisi angka)	16 byte	Sembarang nilai sampai Double
Variant (diisi huruf)	Panjang teks ditambah 22 byte	Sama seperti String dinamis

Tipe data variant dapat dipakai untuk mewakili sembarang tipe data kecuali String panjang tetap.

4.4 Operator Matematika

Operator	Arti
*	Perkalian
/	Pembagian pecahan
+	Penambahan
-	Pengurangan
^	Perpangkatan
\\	Pembagian integer
Mod	Modulo

Operator adalah simbol yang digunakan untuk penambahan, pengurangan, perkalian, pembagian dan kalkulasi lainnya. Tabel berikut menggambarkan ragam operator dan artinya

4.4.1 Empat Operator Utama

Empat operator utama Visual Basic yaitu *, /, + dan – bekerja seperti operator matematika biasa. Untuk perkalian tidak menggunakan lambang x tetapi lambang *. Karena penggunaan lambang x akan diartikan sebagai variabel

dengan nama x. Tabel berikut merupakan gambaran hasil operasi empat operator utama

Formula	Hasil
$5+3$	8
$9-4$	5
$2*5$	10
$6/2$	3

4.4.2 Pembagian Integer, Modulus, dan Perpangkatan

Pembagian integer digunakan untuk mendapatkan hasil bilangan bulat. Pembagian integer selalu menghasilkan bilangan bulat dan mengabaikan sisanya. Operator Mod menghasilkan modulus atau sisa pembagian dalam bentuk bilangan bulat. Sedangkan operator (^) digunakan untuk memangkatkan suatu bilangan. Bilangan di sebelah kiri tanda pangkat adalah basisnya dan bilangan disebelah kanannya adalah pangkatnya.

Tabel dibawah ini adalah gambaran hasil operasi operator di atas.

Formula	Hasil
$13 \setminus 5$	2
$27 \bmod 4$	3
5^3	125

4.4.3 Operator Logika

Jika diperlukan pengujian terhadap lebih dari satu variabel atau kontrol, dapat dilakukan dengan menggunakan operator logika And, Or, Xor atau Not. Operator logika mengkombinasikan dua atau lebih ekspresi menjadi satu ekspresi.

Tabel berikut menggambarkan bagaimana operator-operator di atas bekerja

Tabel AND

True and True	True
True and False	False
False and True	False
False and False	False

Tabel OR

True or True	True
True or False	True
False or True	True
False or False	False

Tabel XOR

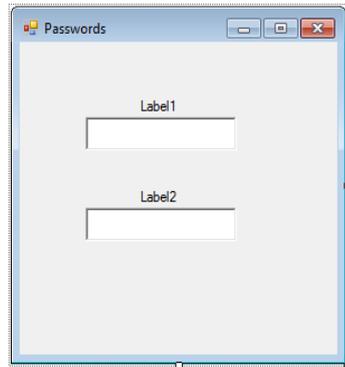
True Xor True	False
True Xor False	True
False Xor True	True
False Xor False	False

Tabel NOT

Not True	False
Not False	True

Di bawah ini merupakan contoh penggunaan operator AND untuk program penambahan password.

1. Buatlah form dengan nama Passwords, dan tambahkan dua buah komponen Label dan dua buah MaskedTextBox seperti pada gambar form dibawah.



2. Aturlah properti text label1 dengan nama “Masukan Kode Nomer Komponen” dan untuk text label2 “PIN”.



3. Klik MaskedTextBox sehingga muncul segitiga kecil di bagian atas dari komponen. Kliklah segitiga kecil tersebut untuk membuka daftar Tasks MaskedTextBox kemudian pilih Set Mask untuk menampilkan dialog box input Mask.
4. Klik pada “**Numeric (5-digits)**” pada input mask kemudian klik OK. Seperti pada kebanyakan kode PIN yang telah Anda ketahui, PIN ditampilkan dalam 5 digit, dan jika pengguna memasukkan password yang salah maka program akan langsung menolaknya.

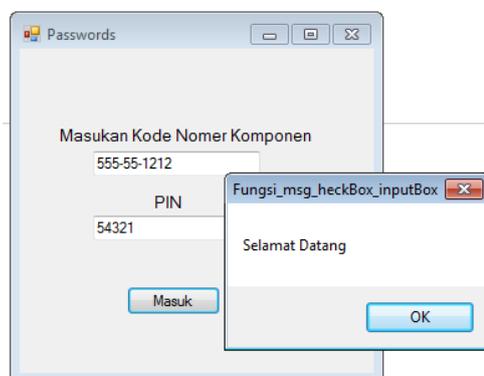
5. Double klik pada tombol “**Masuk**” untuk menampilkan even prosedur Button1_Click pada code editor kemudian ketiklah pernyataan dibawah ini :

```
If MaskedTextBox1.Text = "555-55-1212" And  
MaskedTextBox2.Text = "54321" Then  
  
    MsgBox("Selamat Datang")  
  
Else  
  
    MsgBox("Nomer Tidak Dikenali")  
  
End If
```

Pernyataan tersebut terdiri dari operator AND dimana program membutuhkan validasi PIN dari pengguna yang terkorresponden oleh nomer kode komponen untuk masuk ke dalam sistem. Pada contoh ini validasi PIN adalah “54321”, dan nomor komponen “555-55-1212”.

6. Klik tombol Start Debugging pada toolbar untuk menjalankan program.
7. Ketiklah 555-55-1212 pada masked text box Kode nomor komponen.
8. Ketiklah 54321 pada masked text box PIN.
9. Klik tombol Sign In

Pengguna masuk kedalam program ,seperti yang tampak pada gambar dibawah ini :



10. Klik OK untuk menutup message box.

4.4.4 Operator Kondisi

Operator kondisi membandingkan nilai data satu sama lain dan memberitahukan program hasil dari perbandingan tersebut. Selanjutnya program dapat melakukan pengambilan keputusan berdasarkan pada hasil perbandingan tersebut.

Tabel berikut menunjukkan operator kondisi pada Visual Basic

Urutan	Operator
=	Sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari atau sama dengan
<=	Kurang dari atau sama dengan
<>	Tidak sama dengan

4.5 Fungsi Matematika dalam Visual Basic

Dalam Visual Basic terdapat beberapa fungsi matematika yang dapat membantu kita dalam menghitung ekspresi matematika. Pada tabel berikut ini, argumen n merepresentasikan angka, variabel atau ekspresi yang ingin dievaluasi oleh fungsi.

Fungsi	Tujuan
Abs(n)	Nilai mutlak dari n
Atn(n)	Arctangent dalam radian dari n
Cos(n)	Cosinus dari sudut n
Exp(n)	Konstanta e pangkat n
Rnd(n)	Menghasilkan angka acak antara 0 dan 1
Sgn(n)	Mengembalikan -1 jika n kurang dari 0, 0
Sin(n)	Sinus sudut n
Sqr(n)	Kuadrat dari n
Str(n)	Mengubah nilai numerik menjadi string
Tan(n)	Tangen sudut n
Val(n)	Mengubah nilai string menjadi angka

Untuk menggunakan variabel dalam bahasa pemrograman, caranya sangat mudah. Anda harus mendefinikan jenis variabel di bagian deklarasi program yaitu bagian yang bertuliskan var seperti penjelasan pada sub bab diatas yaitu :

```
Dim nama_variabel As tipe_data
```

Sebagai contoh :

```
Dim nilaiX As Integer = 10
```

Maksud dari pendeklarasian variabel di atas adalah variabel yang bernama **nilaiX** dengan tipe **Integer** (bilangan bulat) dengan isi variabel angka 10.

Penggunaan Kontrol

1. Text Box

Text Box digunakan untuk menampilkan informasi yang dimasukkan saat desain time oleh usersaat run time atau diberikan melalui kode. Teks muncul dapat diedit

Properti Toolbox

BorderStyle	Menentukan tipe border
Font	Mengatur font type, style, size
MaxLength	Membatasi panjang teks yang ditampilkan (nilai 0 menunjukkan panjang teks yang tidak dibatasi)
MultiLine	Menentukan apakah text box menunjukkan baris tunggal atau rangkap
PasswordChar	Menyembunyikan teks dengan karakter tunggal
ScrollBars	Menentukan tipe tampilan scroll bar
SetLength	Panjang teks yang dipilih (saat run time)
Tag	Menyimpan ekspresi string
Text	Text yang ditampilkan

2. Radio Button

Radio Button juga merupakan salah satu komponen penting dalam Visual Basic 2010. Radio button memberikan opsi pilihan pada pengguna untuk hanya dapat memilih satu dari beberapa pilihan yang tersedia. Radio Button memberikan kemampuan untuk memilih secara eksklusif dari suatu grup pilihan. Sehingga memberikan nilai True hanya diberikan pada opsi yang dipilih saja.

Properti radio Button

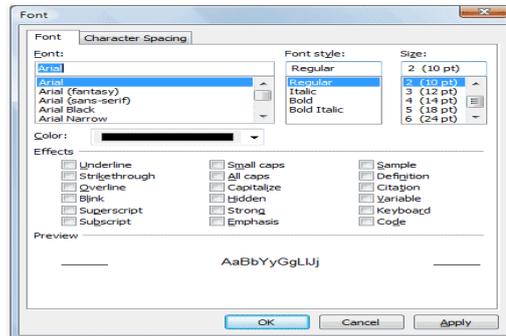
Text	Menunjukkan teks di sebelah box
Font	Mengatur font type, style, size
Value	Menunjukkan apakah option button dipilih (True) atau tidak (False). Hanya satu option button dalam satu grup yang dapat bernilai True. Satu tombol dalam setiap grup option button harus diberi nilai True saat <i>design time</i> .

Event Option Button

Click	Dipicu saat tombol diklik. Nilai properti diubah secara otomatis oleh Visual Basic.
-------	---

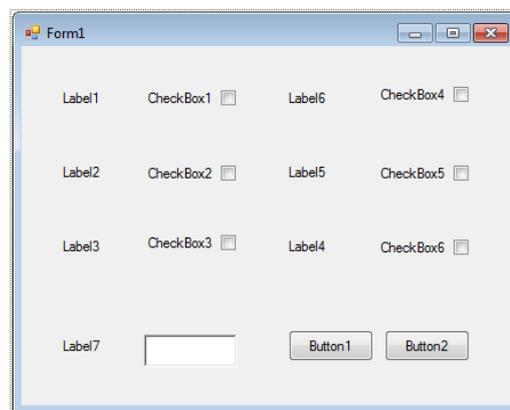
3. Check box

Check Box merupakan salah satu komponen yang penting dalam Visual Basic 2010. Check Box memungkinkan pengguna untuk memilih satu atau lebih item dengan mencentang check box tersebut. Sebagai contoh, pada kotak dialog properties **Font** dari komponen **Text**, ada banyak pilihan check box pada bagian bawah bagian Effects seperti yang terlihat pada gambar dibawah.



Gambar 4.17 Pengaturan Font Properties

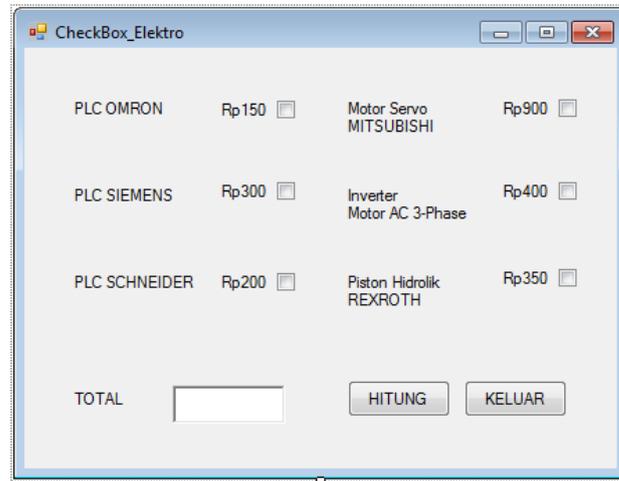
Pengguna dapat memilih underline, subscript, small caps, superscript, blink, dan lain-lain. Sebagai contoh kita akan membuat keranjang belanja toko elektronik dimana pengguna dapat mengklik check box yang sesuai dengan barang yang ingin dibeli dan total pembayaran yang dapat dihitung pada waktu yang sama seperti yang ditunjukkan pada contoh program daftar belanja komponen elektro berikut :



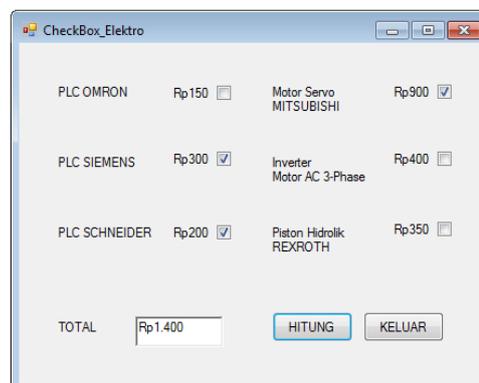
Gambar 4.18 Tampilan layout komponen program CheckBox Elektro



Komponen	Properties
Form1	Text : CheckBox_Elektro
Label1	Text : PLC OMRON
Label2	Text : PLC SIEMENS
Label3	Text : PLC SCHNEIDER
Label4	Text : Motor Servo MITSUBISHI
Label5	Text : Inverter Motor AC 3-Phase
Label6	Text : Piston Hidrolik REXROTH
Label7	Text : TOTAL
Button1	Text : Hitung
Button2	Text : KELUAR
CheckBox1	Text : Rp 150
CheckBox2	Text : Rp 300
CheckBox3	Text : Rp 200
CheckBox4	Text : Rp 900
CheckBox5	Text : Rp 400
CheckBox6	Text : Rp 350
RichTextBox	Text : (kosong)



Gambar 4.19 Tampilan default program CheckBox elektro



Gambar 4.20 Aplikasi Program CheckBox elektro

Kode program untuk CheckBox_Elektro :

```
Public Class CheckBox_elektro
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
    Const PLC_OMR As Integer = 150
```

```
    Const PLC_SMN As Integer = 300
```

```
Const PLC_SCH As Integer = 200
Const SERVO As Integer = 900
Const INVERTER As Integer = 400
Const HIDROLIK As Integer = 350

Dim pilih As Integer

If CheckBox1.Checked = True Then
    pilih += PLC_OMR
End If

If CheckBox2.Checked = True Then
    pilih += PLC_SMN
End If

If CheckBox3.Checked = True Then
    pilih += PLC_SCH
End If

If CheckBox4.Checked = True Then
    pilih += SERVO
End If

If CheckBox5.Checked = True Then
    pilih += INVERTER
End If

If CheckBox6.Checked = True Then
    pilih += HIDROLIK
End If

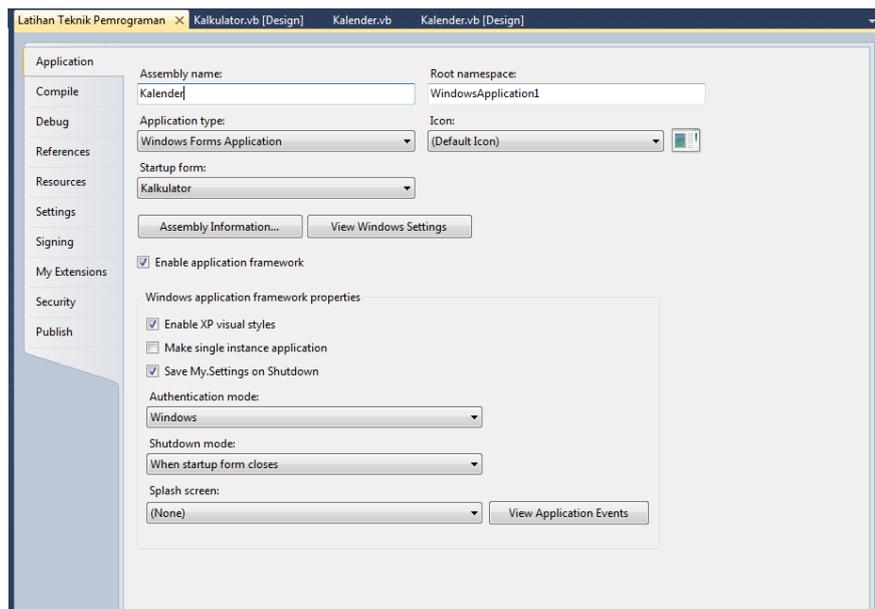
RichTextBox1.Text = pilih.ToString("c")
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click

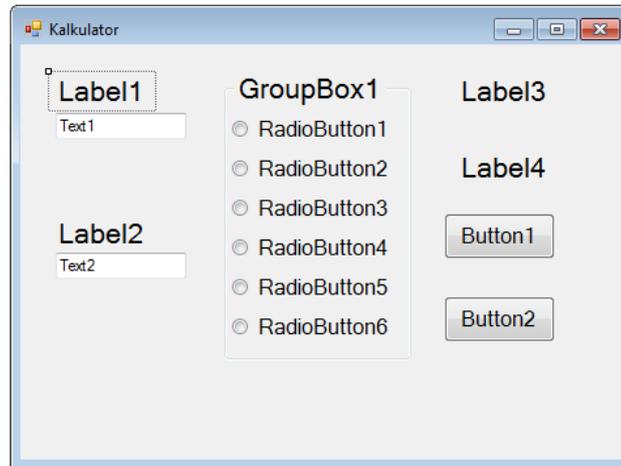
    End
End Sub
End Class
```

Latihan Membuat Program Kalkulator Sederhana

1. Pilih menu project, kemudian klik Add Windows Form...
2. Kemudian rancanglah tampilan form seperti yang ada pada gambar di bawah ini. Gunakan toolbox sesuai dengan obyek kontrol yang ingin dibuat.
3. Bila telah selesai nantinya, rename form ini dengan nama Kalkulator.frm. Tidak perlu membuat new project cukup gunakan project terdahulu untuk semua form latihan, agar semua form terkumpul menjadi satu project.
4. Untuk menjalankan aplikasi program kalkulator ini, diperlukan pengaturan Start up program menjadi Kalkulator.frm. Arahkan kursor ke Solution explorer, pilih Project dan klik dua kali hingga muncul jendela baru. Kemudian gantilah Resorce Start up form : Kalkulator.

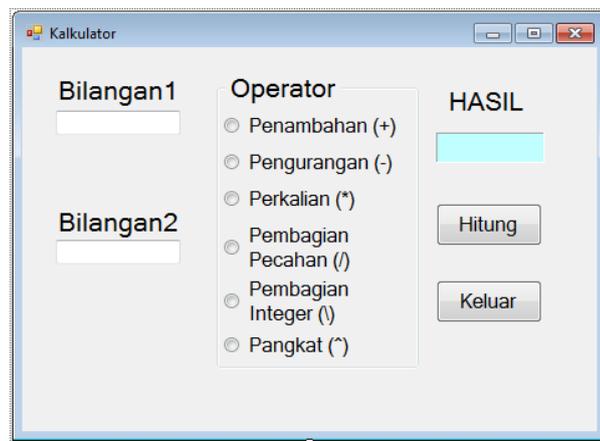


Gambar 4.21 Mengatur Start up Form



Gambar 4.22 Merancang Form Kalkulator

Berikut ini adalah contoh penggunaan dari radio button untuk program aplikasi kalkulator sederhana



Gambar 4.23. Tampilan Form Kalkulator

Kode program yang ditulis di View Code

```
Public Class Kalkulator
Private Sub BtnHitung_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles BtnHitung.Click
```

```
Dim Bil1 As Integer
Dim Bil2 As Integer
Dim msg As String
```

```
Dim judul As String
Dim style As String

Bill = Val(TxtBoxBilangan1.Text)
Bil2 = Val(TxtBoxBilangan2.Text)
judul = "Hasilnya adalah = "
style = vbOKOnly

If RadioTambah.Checked = True Then
    LblHasil.Text = Bill + Bil2
    msg = LblHasil.Text
    MsgBox(msg, style, judul)
EndIf

If RadioKurang.Checked = True Then
    LblHasil.Text = Bill - Bil2
    msg = LblHasil.Text
    MsgBox(msg, style, judul)
EndIf

If RadioKali.Checked = TrueThen
    LblHasil.Text = Bill * Bil2
    msg = LblHasil.Text
    MsgBox(msg, style, judul)
EndIf

If RadioBagiPecahan.Checked = TrueThen
    LblHasil.Text = Bill / Bil2
    msg = LblHasil.Text
    MsgBox(msg, style, judul)
EndIf

If RadioBagiInteger.Checked = TrueThen
    LblHasil.Text = Bill \ Bil2
```

```
msg = LblHasil.Text
MsgBox(msg, style, judul)
EndIf

If RadioPangkat.Checked = TrueThen
    LblHasil.Text = Bil1 ^ Bil2
    msg = LblHasil.Text
    MsgBox(msg, style, judul)
EndIf
EndSub

PrivateSub BtnKeluar_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles BtnKeluar.Click
End
EndSub
EndClass
```



Gambar 4.24 Hasil perhitungan pada program aplikasi

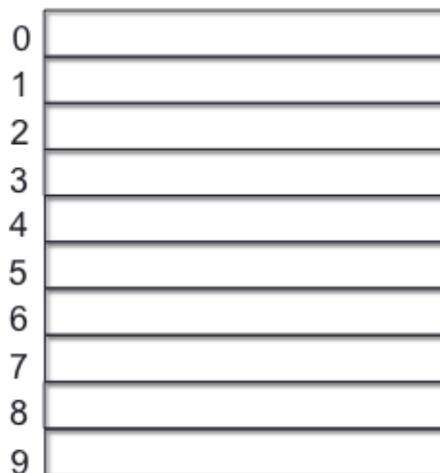
4.6 Array

Array adalah jenis variabel yang dapat menampung lebih dari satu nilai data. Secara sederhana Array bisa dianalogikan bentuknya seperti himpunan. Terdapat dua jenis array yakni :

1. Array Satu Dimensi

Array satu dimensi dapat dianalogikan seperti sebuah tabel yang terdiri dari satu kolom dan sejumlah baris. Ketika membuat array, Visual Basic akan mendesain suatu memori register (tempat penampungan memori).

Berikut ini gambar konseptual array yang terdiri dari sepuluh buah elemen register dimana elemen register tersebut memiliki label nomor mulai 0 sampai dengan 9:



Gambar 4.29. Konseptual array satu dimensi

Contoh deklarasi dan pemakaian array satu dimensi :

```
Dim Resistor (10) As String
```

Terlihat bahwa penomoran urutan variabel Resistor dalam array dimulai dari '0', bukan '1'. Array sering kali digunakan untuk meringkas satu logika (biasanya dalam kasus yang membutuhkan operasi berulang-ulang/looping)

2. Array Multidimensi

Array multi dimensi dapat dianalogikan seperti sebuah tabel yang terdiri dari sejumlah kolom dan sejumlah baris. Setiap kombinasi baris akan menyusun suatu memori register (tempat penampungan memori).

Berikut ini gambar konseptual array multi dimensi yang terdiri dari dua baris dan enam kolom elemen register dimana nomor label baris dan kolom selalu dimulai dengan indeks "0":

	0	1	2	3	4	5
0				5		
1						

Gambar 4.30 konseptual array multi dimensi

Dari tabel diatas tampak bahwa terdapat dua belas kotak memory register yang terdiri dari dua baris dan enam kolom. Pada kotak memory register baris ke-0 kolom ke-3 terisi data "5" (bisa berupa integer, string, short dan lain-lain).

Syntax umum dari Array Satu Dimensi adalah sebagai berikut :

```
Dim Nama_Array (Dim1 index, Dim2 Index, ...) As
tipe_data
```

Sehingga untuk tabel array di atas bisa kita tuliskan sebagai berikut :

```
Dim Tabel(2,6) As Integer

Tabel(0,3) =5
```

Keterangan dari syntax diatas adalah sebagai berikut :

- Dim merupakan kode deklarasi variabel **Tabel** sebagai Array
- **Tabel** adalah nama dari variabel Array dan di deklarasikan sebagai integer
- (2,6) merupakan indeks array yang berarti dua baris dan enam kolom

- Tabel(0,3) = 5 artinya menunjuk index array baris 0 kolom 3 dan diisi memori dengan data 5 bersifat integer.

TUGAS

1. Buatlah program sederhana untuk konversi temperatur dari derajat Celcius ke Fahrenheit atau Reamur dan Kelvin atau ketiga-tiganya sekaligus. Gunakan formula konversi antar derajat temperatur sebagai berikut :
$$F = (C \times 9/5) + 32$$
$$R = C \times 4/5$$
$$K = C + 273$$
2. Buatlah program sederhana untuk menghitung uang yang harus dibayarkan oleh seorang customer di sebuah supermarket jika diketahui :
Jumlah barang, Harga satuan dan PPN sebesar 10%
Kemudian kembangkan program diatas jika diketahui supermarket tersebut memberikan diskon sebesar 25%

TES FORMATIF

1. Jelaskan apa yang dimaksud dengan Variabel beserta fungsi dari penulisan variabel!
2. Ada beberapa macam tipe data yang sering digunakan dalam pendeklarasian kode program. Sebutkan tipe data tersebut serta memori yang digunakan!
3. Jelaskan tentang penggunaan dari operator matematika Pembagian Integer, Modulus dan Perpangkatan serta jelaskan pula tentang Operator kondisi!
4. Apakah yang dimaksud dengan array? Sebutkan dua jenis dari array

Jawab :

KEGIATAN BELAJAR 2

Membuat Program Aplikasi Sederhana Dengan Bahasa Pemrograman Visual Basic

A. Tujuan Pembelajaran

1. Membuat program menggunakan Integrated Development Environment (IDE).
2. Melakukan percobaan dan menerapkan penggunaan Lingkungan Pengembangan Terintegrasi (*Integrated Development Environment-IDE*) bahasa *Visual Basic*.
3. Melakukan percobaan menggunakan konsep dasar bahasa pemrograman bahasa visual basic.
4. Merencana flow chart untuk persiapan pemrograman.
5. Membuat program sederhana dengan *Visual Basic* untuk divisualisasikan di layar monitor.

B. Uraian Materi

1. Pengenalan *Flow Chart* dan aplikasi pada program.
2. Pengenalan instruksi dalam bahasa pemrograman dengan *Visual Basic*.
3. Membuat program dengan Visual Basic.

C. Alokasi Waktu

24 jam pelajaran

D. Metode Pembelajaran

Teori dan Praktek

E. Media pembelajaran

- PC/Notebook
- Windows 7
- Visual Basic 2010

Membuat Program Aplikasi Sederhana Dengan Bahasa Pemrograman Visual Basic

KEGIATAN 1 : Flow Chart atau Diagram Alir

1.1 Konsep Dasar Algoritma

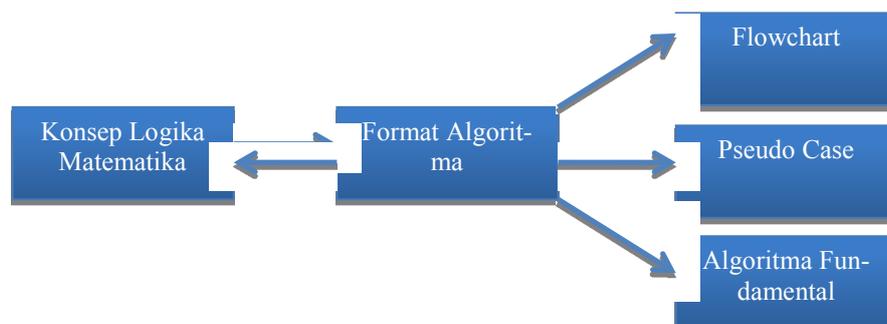
Beberapa metode untuk merancang algoritma dalam program komputer :

- Diagram Alir (Flowchart)
- Kode Semu (Pseudo Code)
- Algoritma Fundamental

Komponen yang harus ada di dalam algoritma adalah :

- Komponen input atau masukan : terdiri dari pemulihan variabel, jebis variabel, tipe ariabel, konstanta dan parameter (dalam fungsi).
- Komponen Proses : merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika maslaah, logika algoritma, struktur, metode (perbandingan, penggabungan, pengurangan, dll).
- Komponen output atau keluaran : merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah menjawab permasalahan dan tampilan yang baik.

Struktur hubungan dan jenis algoritma :



Gambar 1.1 Struktur Jenis Algoritma

Knuth (1973) menyajikan format algoritma yang dapat digunakan secara bebas untuk berbagai bahasa pemrograman, artinya dapat dengan mudah diimplementasikan menggunakan Pascal, C, Fortran, PL, atau BASIC.

1.2 Pengertian Dari Flowchart

Flowchart adalah bagian-bagian yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma. Didalam program, flowchart menunjukkan suatu prosedur sistem secara logika. Pada waktu akan menggambar suatu diagram alir, seorang programmer dapat mengikuti pedoman-pedoman sebagai berikut :

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam flowchart harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
4. Masing-masing kegiatan di dalam diagram alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan dalam flowchart harus disusun secara urut.

Ada lima jenis diagram alir yaitu :

- Diagram Alir Sistem
- Diagram Alir Dokumen
- Diagram Alir Skematik
- Diagram Alir Program
- Diagram Alir Proses

Namun dalam buku ini kita akan membahas tentang diagram alir program, sebagai dasar dari perancangan desain software yang akan kita pelajari.

1.3 Diagram Alir Program (Program Flowchart)

Diagram alir program merupakan suatu bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Flowchart ini dibuat dari derivikasi diagram alir sistem. Diagram alir program dibuat dari simbol-simbol berikut ini :

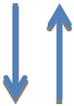
Simbol	Keterangan
	Simbol Input/Output yang digunakan untuk mewakili data input atau output.
	Simbol Proses digunakan untuk mewakili suatu proses.
	Simbol Garis Alir digunakan untuk menunjukkan arus dari proses.
	Simbol Penghubung, digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau halaman lainnya.
	Simbol Keputusan, digunakan untuk suatu penyelesaian kondisi di dalam program.
	Simbol Proses Terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
	Simbol persiapan digunakan untuk memberi nilai awal suatu besaran.
	Simbol Titik Terminal, digunakan untuk menunjukkan awal dan akhir dari suatu proses.

Diagram alir program terdiri dari dua macam, yaitu diagram alir logika program (*program logic flowchart*) dan diagram alir program komputer terinci (*detailed computer program flowchart*). Diagram alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program secara logika. Gambar berikut menunjukkan contoh diagram alir logika program untuk menghitung rata-rata :

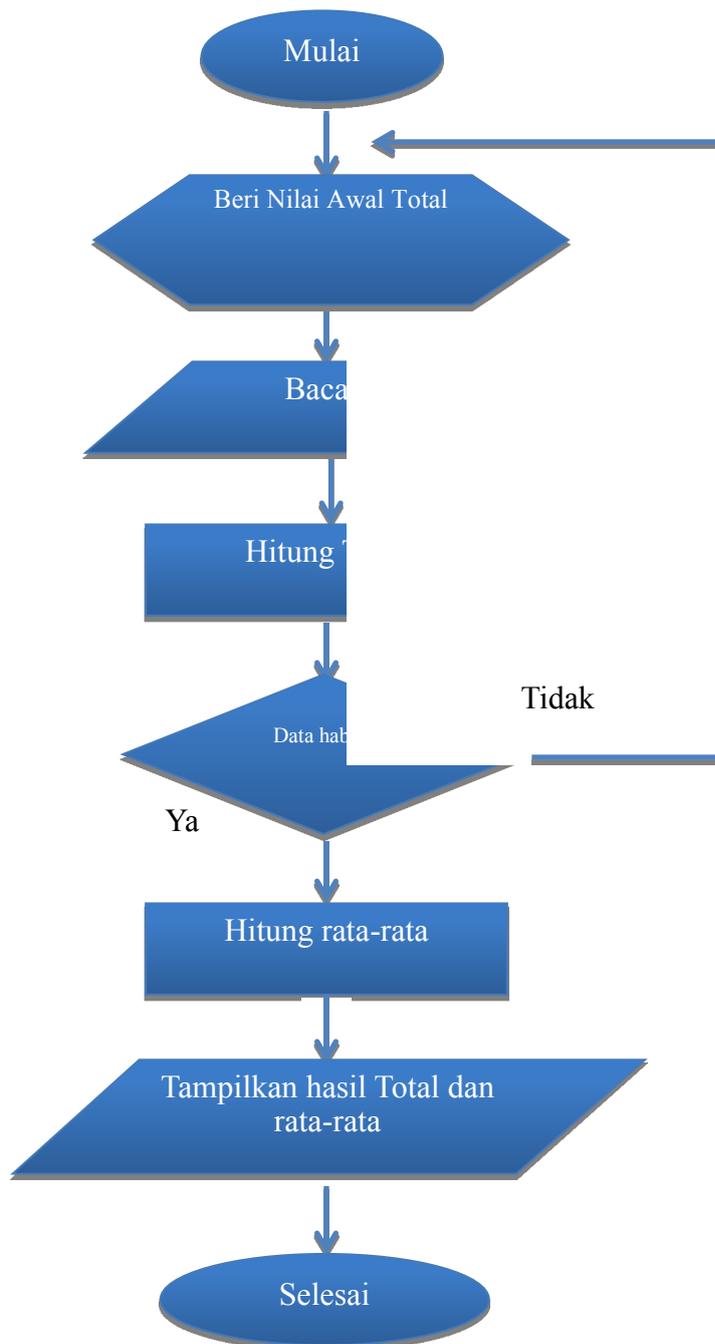
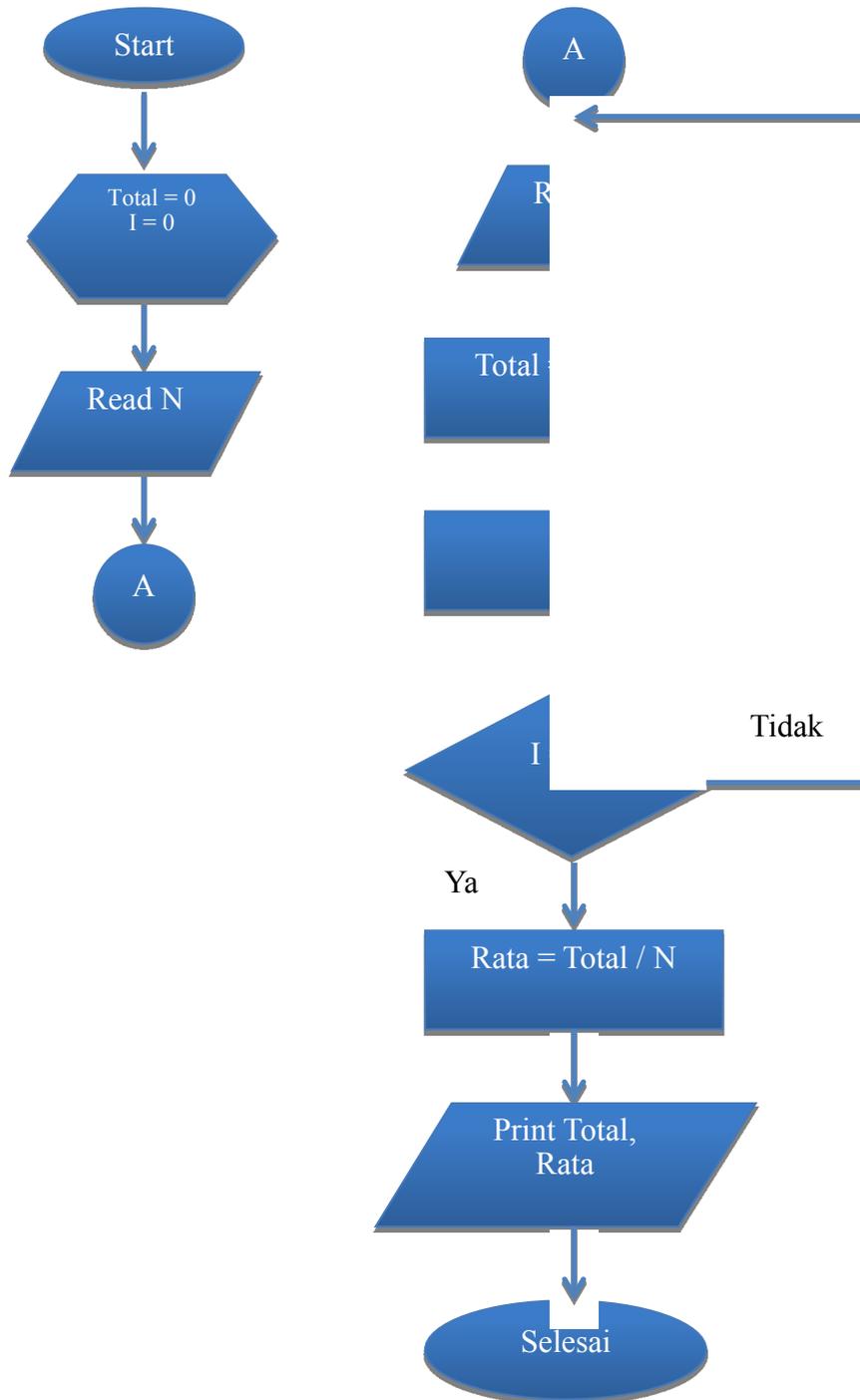


Diagram alir program komputer terinci (*detailed computer program flowchart*) digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Berikut adalah contoh diagram alir program komputer terinci menghitung rata-rata :



TUGAS

1. Pada bab sebelumnya Anda telah mengetahui tentang contoh kode program kalkulator. Buatlah flowchart dari program kalkulator tersebut sehingga mudah dimengerti dan ketika seorang programmer membaca flowchart tersebut menjadi mudah dan mengerti bagaimana harus membuat program aplikasinya!

TES FORMATIF

1. Sebutkan tiga macam metode dalam perancangan algoritma untuk membuat kode program!
2. Komponen input, proses dan output merupakan tiga komponen dalam perancangan algoritma. Jelaskan maksud dari ketiga komponen tersebut dan gambarkan struktur dari algoritma tersebut!
3. Apakah yang Anda ketahui tentang diagram alir / flowchart? Jelaskan dan sebutkan pula dua tipe dari flowchart!
4. Flowchart/diagram alir memiliki berbagai macam simbol yang selalu digunakan sebelum kita memulai merancang program. Gambarkan simbol-simbol dari flowchart beserta penjelasannya!



Jawab :

KEGIATAN 2 : Pengenalan Instruksi Struktur Dan Syarat Dalam Bahasa Visual Basic

2.1 Struktur Keputusan

Dalam kehidupan sehari-hari kita seringkali membuat berbagai keputusan. Keputusan itu dibuat karena kita dihadapkan dengan berbagai kondisi atau keadaan yang kita adapi sekarang, misalnya :

- Karena hari libur, kita tidak belajar di sekolah
- Karena akan menghadapi UN, siswa harus lebih giat belajar
- Akibat malas belajar, siswa tersebut mendapat nilai jelek

Kondisi

Karena hari libur

Karena akan menghadapi UN

Akibat malas belajar



Kondisi

Keputusan yang dibuat

Kita tidak belajar di sekolah

siswa harus lebih giat belajar

siswa tersebut mendapat nilai jelek



Keputusan yang dibuat

2.2 IF.....Then

Dalam Visual Basic terdapat fasilitas untuk pembuatan keputusan yaitu dengan menggunakan statemen **IF.....Then** yang ditulis dalam baris tunggal :

```
If Condition Then Statement
```

Dimana **condition** adalah ekspresi kondisi dan **statement** adalah pernyataan program Visual Basic. Ekspresi berkondisi digunakan pada blok pernyataan khusus yang disebut dengan struktur keputusan dan fungsinya mengendalikan

apakah pernyataan lain pada program kita dijalankan dan bagaimana urutan-urutannya.

Contoh :

```
If Nilai >= 81 Then lblNilai = 'A'
```

Pada contoh diatas ekspresi kondisi adalah Nilai >=81. Jika variabel Nilai mengandung nilai yang lebih dari atau sama dengan 81, maka Visual Basic akan menseset teks lblNilai menjadi nilai "A". Jika nilai kurang dari 81 maka Visual Basic akan melompati pernyataan penugasan tersebut dan menjalankan baris berikutnya pada *event* prosedur.

2.3 IF.....THEN.....ELSE

Perintah If....Then dapat dikembangkan menjadi format seperti berikut ini :

```
If Kondisi Then
    Satu baris atau lebih perintah Visual Basic
Else
    Satu baris atau lebih perintah Visual Basic
```

Jika kondisi bernilai **True** (benar) maka blok perintah setelah Then akan dikerjakan. Tapi jika kondisi bernilai **False** (salah), maka blok perintah setelah Else akan dikerjakan.

```
If sensor_kecepatan >= 720 And sensor_kecepatan <= 750
Then
    Msg = "Motor bekerja optimal"
    MsgBox msg, style, judul
Else
    Msg = "Motor bekerja tidak optimal"
    MsgBox msg, style, judul
End If
```

Program diatas akan memberikan informasi tentang kinerja suatu motor. Ketika motor bekerja dan terukur kecepatannya oleh sensor dalam satuan rpm (rotasi per menit). Jika motor berputar di rentang kecepatan 720 rpm hingga 750 rpm, maka aplikasi program akan memberikan informasi bahwa motor bekerja optimal. Jika kecepatan motor kurang dari 720 rpm atau lebih dari 750 rpm, maka program aplikasi akan memberikan informasi bahwa motor bekerja tidak optimal.

2.4 IF.....THEN.....ELSEIF.....ELSEIF.....ENDIF

Pengembangan lain dari perintah If adalah sebagai berikut

```
If Kondisi_1 Then
    Satu baris atau lebih perintah Visual Basic pertama
ElseIf Kondisi_2 Then
    Satu baris atau lebih perintah Visual Basic kedua
-
-
-
Else
    Satu baris atau lebih perintah Visual Basic
End If
```

Jika kondisi_1 bernilai True, maka blok perintah pertama akan dikerjakan. Jika kondisi_2 bernilai True, maka blok perintah kedua akan dikerjakan. Demikian seterusnya. Jika kondisi_1 samapai kondisi_n bernilai False, maka blok perintah setelah Else akan dikerjakan. Perintah Elseif.....Then dapat diulang sebanyak yang diperlukan.

Contoh :

```
Dim intSuhu As Integer

Dim msg As Integer `return value untuk MsgBox()

intSuhu = inputBox("Berapa suhu ruangan saat ini?")

If (intSuhu > 28) Then

    Msg = MsgBox ("suhu ruangan panas, mohon
    menyalakan Air Conditioner (AC)")

ElseIf (intSuhu > 18) then

    Msg = MsgBox ("suhu ruangan dingin,mohon matikan
    Air Conditioner (AC)")

Else

    Msg = MsgBox ("suhu ruangan terlalu dingin,mohon
    menyalakan heater")

End If
```

Program di atas menunjukkan contoh keadaan temperatur suatu ruangan dimana pengguna harus menuliskan temperatur pada input box. Jika suhu menunjukkan angka lebih dari 28 maka akan muncul perintah untuk menghidupkan AC. Sebaliknya jika suhu ruangan menunjukkan rentang angka antara 19-28 maka akan muncul perintah untuk mematikan AC. Pada saat suhu menunjukkan angka kurang dari 18 akan muncul perintah untuk menyalakan pemanas ruangan.

2.5 SELECT CASE

Struktur Select Case serupa dengan If...Then....Elseif, tetapi lebih efisien apabila percabangan tergantung pada satu variabel kunci atau test case. Penggunaan If.....Then.....Elseif dua atau tiga kali mungkin tidak membingungkan tetapi jika penggunaannya lebih dari tiga kali pasti akan membingungkan. Perintah Select Case mempunyai format sebagai berikut :

```
Select Case Variabel  
  
Case Value 1  
  
    Satu baris atau lebih perintah Visual Basic  
  
Case Value 2  
  
    Satu baris atau lebih perintah Visual Basic  
  
-  
  
-  
  
End Select
```

Struktur Select Case dimulai dengan kata Select Case dan diakhiri dengan End Select. Variabel adalah nilai kunci atau test case. Value1, Value2 bisa diganti dengan angka, string atau nilai lain yang berhubungan dengan test case. Jika salah satu nilai/Value sesuai dengan variabel, pernyataan dibawah klausa Case akan dijalankan dan Visual Basic akan melanjutkan mengeksekusi program setelah pernyataan End Select.

Contoh :

```
Private Sub pilih_komponen (Pilihan As String)  
  
    Dim pesan As String  
  
    Select Case Pilihan  
  
        Case "Resistor"  
  
            Pesan = "Anda memilih komponen pasif Resistor"  
    End Select
```

```

Case "Induktor"

Pesan = "Anda memilih komponen pasif Induktor"

Case "Kapasitor"

Pesan = "Anda memilih komponen pasif Induktor"

Case Else

Pesan = "Maaf , pilihan anda tidak tepat, silahkan
ulang"

End Select

MsgBox pesan, vbExclamation, "Pilihan Komponen
pasif elektronika"

End Sub

```

Pada program diatas menunjukkan pemilihan beberapa komponen pasif elektronika diantara resistor, induktor atau kapasitor. Setiap pilihan komponen yang benar akan muncul pesan bahwa pengguna telah memilih komponen. Jika pilihan tidak terdapat pada salah satu diantara komponen tersebut maka akan muncul pesan untuk mengulangi pilihan komponen.

2.6 SELECT CASE Berkondisi

Perintah Select Case dapat juga berkondisi dengan menggunakan operator kondisi (=, <, >, >=, <= dan <>). Tetapi perintah Select Case tidak bisa dikombinasikan dengan operator Logika (And, Or, Not, Xor). Jika ingin menggunakan operator logika harus menggunakan perintah

```
If.....Then.....ElseIf
```

Contoh program berikut menggambarkan penggunaan Select Case berkondisi

```

Private Sub ujiUsia()

'Prosedur ini menggambarkan pengelompokan power daya
rumah'

Dim intDaya As Integer

Dim msg As Integer `MsgBox1() return

```

```
intDaya 0 InputBox ("Berapa Daya Rumah Anda?")

Select Case intDaya

Case Is <= 2200

    Msg = msgBox ("Rumah anda termasuk golongan tarif
    dasar listrik R1/TR")

Case Is <= 5500

    Msg = msgBox ("Rumah anda termasuk golongan tarif
    dasar listrik R2/TR")

Case Else

    Msg = msgBox ("Rumah anda termasuk golongan tarif
    dasar listrik R3/TR")

End Select

End Sub
```

Pada program diatas pertanyaan awal akan muncul tentang daya listrik dimiliki pengguna. Jika daya listrik lebih kecil sama dengan 2200 maka pesan yang akan muncul menunjukkan golongan tarif dasar listrik R1. Jika daya listrik kurang dari sama dengan 5500 maka pesan yang muncul adalah golongan tarif dasar listrik R2. Namun jika daya listrik lebih dari 5500, maka pesan yang muncul adalah golongan tarif dasar listrik R3.

2.7 SELECT CASE Dalam Rentang

Selain itu, perintah Select Case dapat juga digunakan untuk menguji suatu nilai/ variabel apakah termasuk dalam rentang tertentu. Untuk keperluan tersebut, digunakan tambahan perintah To pada perintah Select Case seperti pada contoh program di bawah.

Contoh berikut menggambarkan perintah Select Case dalam rentang nilai tertentu :

```
Private Sub ujiUsia2 ()

    `Prosedur ini menggambarkan tingkat pendidikan
    berdasarkan usia`

    Dim intUsia As Integer

    Dim msg As Integer

    intUsia = InputBox ("Berapa Usia Anda?")

    Select Case intUsia

        Case 7 To 12

            Msg = MsgBox ("Anda sedang bersekolah di tingkat
            dasar SD")

        Case 13 To 15

            Msg = MsgBox ("Anda sedang bersekolah di tingkat
            menengah pertama SMP")

        Case 16 To 18

            Msg = MsgBox ("Anda sedang bersekolah di tingkat
            menengah kejuruan SMK")

        Case Else

            Msg = MsgBox ("Isi usia Anda dengan benar!")

    End Select

End Sub
```

2.8 IF Bertingkat

Perintah IF bertingkat artinya didalam kondisi statement IF terdapat persyaratan IF berikutnya (IF ganda/bertingkat). Perhatikan potongan program berikut untuk contoh lebih jelasnya.

```
If Nilai Ujian Nasional > 6.0 Then

    Nilai UN = nilai ujian nasional * 0,6

    Nilai raport = nilai rata-rata raport * 0.4

    Nilai total = Nilai UN + Nilai raport

If Nilai total > 8.0 Then

    `Siswa dinyatakan lulus dengan predikat
    istimewa`

ElseIf Nilai total > 7.0 Then

    `Siswa dinyatakan lulus dengan predikat
    memuaskan`

Else

    `Siswa dinyatakan lulus dengan predikat cukup`

EndIf

Else

    `Siswa dinyatakan tidak lulus`

End If
```

Penggalan program di atas akan menghitung nilai kelulusan siswa dengan beberapa syarat dan kondisi. Jika nilai ujian nasional siswa lebih dari 6.0 maka siswa dinyatakan lulus, sedangkan kurang dari 6.0 dinyatakan tidak lulus. Untuk siswa yang lulus di klasifikasikan menjadi tiga kelompok. Kelompok pertama jika nilai lebih besar dari 8.0 maka akan mendapat predikat istimewa. Kelompok kedua jika mempunyai nilai antara 7.1 sampai 8.0 maka akan mendapat predikat memuaskan. Dan kelompok terakhir memiliki nilai antara 6.1 hingga 7.0 akan mendapat predikat cukup.

2.9 Menguji Data

Menguji suatu data dapat dilakukan melalui fungsi-fungsi berikut :

- IsEmpty()
- IsNull()
- IsNumeric()

Fungsi Is...() disebut sebagai fungsi penguji data. Tabel berikut menjelaskan masing-masing kegunaan fungsi inspeksi data.

Nama Depan	Keterangan
IsDate()	Memeriksa apakah argumen fungsi ini bertipe date (atau apakah data dapat dikonversi menjadi tanggal yang valid)
IsEmpty()	Memeriksa apakah argumen fungsi ini telah diinisialisasi
IsNull()	Memeriksa apakah argumen fungsi ini menyimpan nilai Null
IsNumeric()	Memeriksa apakah argumen fungsi ini berupa angka (atau apakah data dapat dikonversi menjadi angka yang valid)

Contoh program berikut memeriksa variabel Empty

```
Dim var1, var2, var3, var4 As Variant
Dim msg As Integer 'Return value untuk MsgBox
var1 = 0           'diisi nilai nol
var2 = Null       'diisi nilai Null
var3 = ""         'diisi nilai string Null

If IsEmpty(var1) Then
    Msg = MsgBox ("var1 is empty", vbOKOnly)
End If

If IsEmpty(var2) Then
    Msg = MsgBox ("var2 is empty", vbOKOnly)
End If

If IsEmpty(var3) Then
    Msg = MsgBox ("var3 is empty", vbOKOnly)
End If

If IsEmpty(var4) Then
    Msg = MsgBox ("var4 is empty", vbOKOnly)
End If
```

Jika program tersebut dijalankan maka satu-satunya pesan yang muncul adalah :

```
var4 is empty
```

Hal ini terjadi karena fungsi IsEmpty akan bernilai True jika variabel yang menjadi argumennya belum pernah diisi suatu nilai apapun, dalam hal ini variabel yang belum pernah diisi suatu variabel apapun adalah var4.

Fungsi IsNull adalah satu-satunya cara untuk mengujij apakah suatu variabel bernilai Null. Jika suatu variabel telah diisi dengan nilai Null (misal var2 pada contoh program diatas) maka pengujian dengan perintah

```
If (var2 = Null) then.....
```

Akan selalu bernilai False. Karena itu jika akan menguji suatu variabel berisi Null harus digunakan fungsi IsNull.

Berikut adalah contoh penggunaan fungsi IsNull untuk menguji apakah suatu kontrol textbox sudah diisi atau belum.

```
If IsNull (txtJamKerja) Then  
    Msg = MsgBox ("Jam kerja belum diisi",  
vbOKOnly)  
Else  
    Msg = MsgBox (Terima kasih, jam kerja telah  
diisi", vbOKOnly)  
End If
```

TUGAS

1. Pelajari dengan seksama format penulisan perintah If....Then serta format perintah Select Case. Perhatikan perbedaan antara perintah If Then dengan Select Case
2. Buatlah flowchart dari contoh program if bertingkat yang terdapat pada penjelasan diatas!
3. Buatlah flowchart dari contoh program Select Case berentang yang terdapat pada penjelasan diatas!

TES FORMATIF

Buatlah program

Suatu perusahaan air minum kemasan membuat diskon harga untuk pembelian produknya dalam jumlah tertentu seperti yang tertera dalam tabel berikut ini :

Pembelian	Diskon	Harga Per Unit
1-10 Box	0%	Rp. 7.500,00
11-25 Box	5%	Rp. 7.500,00
26-50 Box	10%	Rp. 7.500,00
>50 Box	15%	Rp. 7.500,00

Tentukan berapa total uang yang harus dibayarkan oleh konsumen untuk pembelian tertentu!



Jawab :

KEGIATAN 3 : Pengenalan Instruksi Perulangan Dalam Bahasa Visual Basic

3.1 Perulangan (Looping)

Adakalanya dalam pemrograman dibutuhkan suatu perintah yang diulang-ulang. Dalam Visual Basic perintah tersebut dinamakan Looping. Visual Basic menyediakan beberapa perintah untuk proses Looping. Perintah Looping berjalan sampai kondisi tertentu yang menyebabkan perulangan harus dihentikan.

3.2 Do.....Loop

Ada empat macam perintah Do.....Loop yaitu :

- Do While.....Loop
- Do.....Loop While
- Do Until.....Loop
- Do.....Loop Untill

Hampir semua proses Looping dapat dikerjakan oleh keempat perintah diatas.

a. Do While.....Loop

Semua perintah perulangan mempunyai bagian kondisi/penguji untuk mengendalikan proses perulangan. Bagian penguji/kondisi pada perintah Do While sama dengan perintah If.....Then. Format dari perintah Do While adalah sebagai berikut

:

```
Do While Kondisi
```

```
    Satu blok atau lebih perintah Visual Basic
```

```
Loop
```

Perulangan akan dilakukan selama kondisi bernilai True. Perulangan akan dihentikan ketika kondisi bernilai False. Setelah perulangan berhenti, maka eksekusi program dilanjutkan ke perintah setelah Loop.

Contoh Program :

```
PublicClassForm2

    PrivateSub Form2_Load(ByVal sender As
        System.Object, ByVal e As System.EventArgs)
        Handles MyBase.Load

        Dim hitung AsInteger
            hitung = 1

        'proses perulangan
        DoWhile hitung <= 10

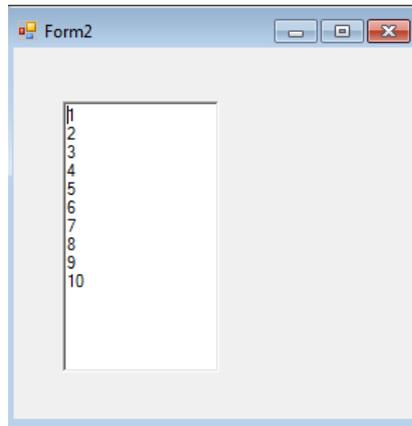
            RichTextBox1.Text = RichTextBox1.Text & hi-
                tung & vbCrLf

            hitung = hitung + 1

        Loop
    EndSub

EndClass
```

Program diatas akan mencetak angka dari 1 sampai dengan 10. Pada program ini, perulangan akan berhenti jika variabel hitung bernilai 10 yang dinyatakan dengan kondisi hitung <=10



Gambar 3.1 Hasil perulangan

b. Do.....Loop While

Perbedaan utama antara Do While.....Loop dan Do.....Loop While adalah letak kondisinya. Program berikut sama dengan contoh diatas, tetapi letak kondisinya yang berbeda. Perbedaan kondisi yang diletakkan di awal dan di akhir perulangan adalah sebagai berikut :

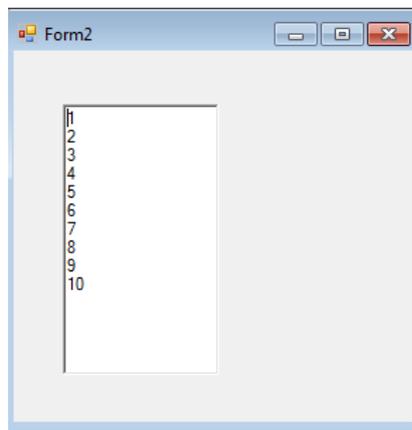
Kondisi diletakkan di awal Perulangan	Kondisi diletakkan di akhir Perulangan
Kondisi bernilai False, maka bagian yang diulang tidak akan dijalankan	Kondisi bernilai False, maka bagian yang diulang dikerjakan satu kali.

```
PublicClassForm2
```

```
PrivateSub Form2_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
```

```
Dim hitung AsInteger
hitung = 1
```

```
'proses perulangan
Do
    RichTextBox1.Text = RichTextBox1.Text & hitung
    & vbCrLf
    hitung = hitung + 1
LoopWhile hitung <= 10
EndSub
EndClass
```



Gambar 3.2 Hasil Perulangan

Jika nilai Hitung dirubah menjadi 11 maka kondisi yang diletakkan diawal looping tidak akan memproses, tapi jika kondisi terletak di akhir looping maka akan melakukan satu kali perhitungan yaitu $11+1=12$.

c. Do Until.....Loop

Pada perintah Looping sebelumnya, perintah akan dijalankan selama kondisi bernilai True dan Looping akan berhenti sampai kondisi bernilai False. Maka perintah Do Until.....Loop kebalikannya. Perintah akan dijalankan selama kondisi bernilai Falsedan berhenti jika kondisi bernilai True.

Format penulisan perintah ini adalah :

```
Do Untill kondisi
```

```
    Pernyataan
```

```
Loop
```

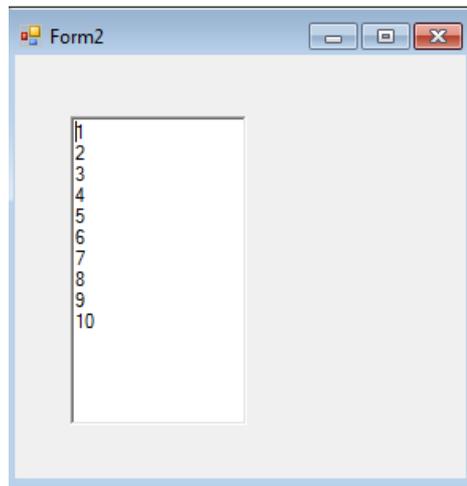
Berikut ini contoh program dengan menggunakan Do Until.....Loop

```
Public Class Form2
    Private Sub Form2_Load(ByVal sender As
        System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load

        Dim hitung As Integer
        hitung = 1
        'proses perulangan
        Do Until hitung > 10

            RichTextBox1.Text = RichTextBox1.Text &
            hitung & vbCrLf
            hitung = hitung + 1

        Loop
    EndSub
End Class
```



Gambar 3.3 Hasil Perulangan

Loop yang menggunakan Until serupa dengan loop yang menggunakan While, kecuali kondisi tes biasanya mengandung operator yang berlawanan, misalnya pada Loop While menggunakan \leq (kurang dari atau sama dengan) maka pada Loop Until menggunakan $>$ (lebih dari).

d. Do.....Loop Until

Sama dengan Do Until.....Loop, hanya letak kondisinya yang berbeda. Yang satu diawal looping yang lainnya diakhiri proses looping. Perbedaan kondisi yang diletakkan di awal dan di akhir perulangan adalah sebagai berikut :

Kondisi diletakkan di awal Perulangan	Kondisi diletakkan di akhir Perulangan
Kondisi bernilai False, ,aka bagian yang diulang tidak akan dijalankan	Kondisi bernilai False, maka bagian yang diulang dikerjakan satu kali.

```
PublicClassForm2

PrivateSub Form2_Load(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load

Dim hitung AsInteger
    hitung = 1

'proses perulangan
Do
    RichTextBox1.Text = RichTextBox1.Text &
        hitung & vbCrLf
    hitung = hitung + 1
Loopuntil hitung > 10
EndSub
EndClass
```

Keluar dari Perulangan Do.....Loop

Jika suatu saat dibutuhkan keluar dari perulangan sebelum perulangan selesai, maka hal ini dapat dilakukan dengan Exit Do. Secara umum penggunaan Exit Do ini diawali dengan perintah If.....Then. Dengan demikian dapat dibuat suatu perulangan yang dikontrol oleh kondisi Do, tetapi jika suatu kondisi tertentu terjadi (misalnya di dalam bagian yang diulang kita mengisikan sesuatu yang tidak diinginkan), maka perintah Exit Do akan memaksa Visual Basic untuk keluar dari perulangan dan melanjutkan program setelah perulangan.

Untuk lebih jelasnya perhatikan dua buah contoh program di bawah ini:

Contoh1 :

```
Private Sub Form_Activate()  
  
Dim i As Integer  
  
Dim inputName As String  
  
Do While i <=10  
  
    InputName = InputBox ("Masukkan sembarang nama atau  
    ketik Done untuk keluar")  
  
    If inputName = "Done" Then  
  
        Exit Do-----> Keluar dari perulangan  
  
    End If  
  
    Print inputName  
  
Loop  
  
End Sub
```

Contoh 2 :

```
Dim Number As Double

Do

    Number = InputBox ("Masukkan angka kuadrat. Tuliskan
    -1 untuk keluar")

    Number = Number * Number

    TextBox1.Text = Number

Loop While Number >= 0
```

Pada contoh program 2, pengguna akan memasukkan angka dan angka tersebut akan dikuadratkan oleh program. Hasil dari pengkuadratan tersebut akan ditampilkan di TextBox. Sayangnya ketika pengguna merasa cukup dengan melakukan proses pengkuadratan tersebut, pengguna tidak akan bisa keluar dari proses tersebut, kecuali dengan memasukkan angka kurang dari 0 (contoh -1, -2, -3 dst).

Ketika pengguna memasukkan nilai -1, program akan keluar dari proses looping karena ada syarat **while >= 0**. Artinya proses looping hanya akan dikerjakan jika input box bernilai lebih besar atau sama dengan 0. Ketika input box diberi masukkan -1, maka kondisi tidak terpenuhi sehingga routine (proses program) akan keluar dari fungsi looping.

Pada contoh program 1, jika input box diberi masukan "Done" maka program akan mengeksekusi perintah Exit Do. Artinya routine/proses program akan keluar dari fungsi looping Do...Loop.

3.3 While.....End While

Struktur looping ini sama dengan Do While.....Loop, hanya tidak tersedia perintah untuk keluar dari looping. Struktur ini sering digunakan dalam operasi database, yakni untuk mengambil semua data mulai dari *record* pertama sampai dengan yang terakhir dari sebuah tabel/recordset.

Struktur penulisan :

```
While (kondisi = True)
-
-
-
End While
```

3.4 For.....Next

Perulangan For....Next berfungsi mengeksekusi sekelompok pernyataan program selama beberapa kali dalam suatu *event procedure*. Pendekatan ini sangat bermanfaat apabila kita ingin melakukan beberapa perhitungan yang berhubungan, bekerja dengan elemen pada layar atau memproses beberapa input dari pemakai.

Format penulisan For....Next adalah :

```
For Variabel1 = start To end
    Pernyataan yang diulang
Next Variabel1
```

Pada format diatas, **For**, **To** dan **Next** adalah kata kunci yang dibutuhkan dan operator sama dengan “ = “ digunakan untuk menandakan fungsi syarat dari perulangan . **Start** dan **End** adalah nilai numerik yang menggambarkan titik awal dan titik akhir perulangan.

Berikut ini adalah contoh program looping dengan For.....Next

```
PublicClassForm2
```

```
PrivateSub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

End

EndSub

PrivateSub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

Dim hitung As Integer

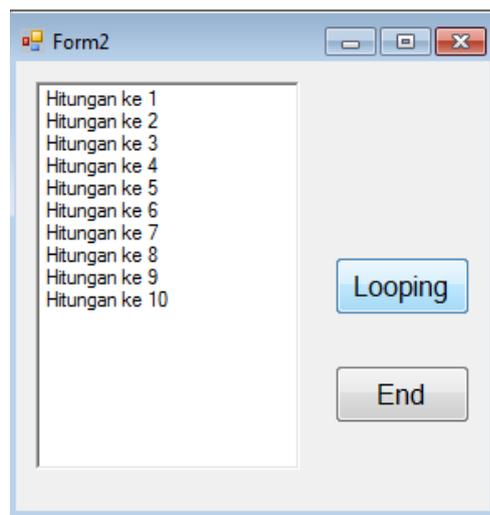
    For hitung = 1 To 10

        RichTextBox1.Text = RichTextBox1.Text &
            Hitungan ke "& hitung & vbCrLf

    Next hitung

EndSub

EndClass
```



Gambar 3.4 Hasil Perulangan

Pada program diatas, variabel yang digunakan dalam looping adalah Hitung (disebut variabel Counter). Setiap kali loop dijalankan, variabel ini bertambah satu. Pada saat pertama melakukan loop, variabel mengandung nilai 1 (nilai start) dan pada saat terakhir mengandung nilai 10 (nilai end).

Dengan imajinasi, perintah For....Next dapat digunakan untuk membuat loop dengan pola counter selain 1, 2, 3, 4 dan seterusnya. Kita bisa menentukan nilai yang berbeda untuk start pada loop lalu menggunakan kata Step untuk menaikkan counter dengan interval yang berbeda.

Contoh program :

```
PublicClassForm2

PrivateSub Button2_Click(ByVal sender As Sys-
    tem.Object, ByVal e As System.EventArgs) Handles
    Button2.Click

End
EndSub

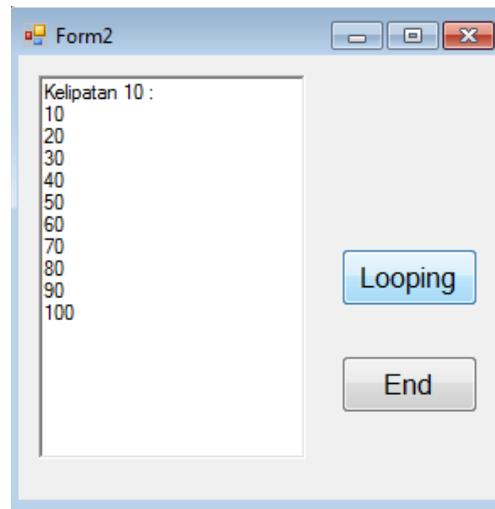
PrivateSub Button1_Click(ByVal sender As Sys-
    tem.Object, ByVal e As System.EventArgs) Handles
    Button1.Click

Dim hitung AsInteger
    RichTextBox1.Text = "Kelipatan 10 :"& vbCrLf

    For hitung = 10 To 100 step 10
        RichTextBox1.Text = RichTextBox1.Text &
            hitung & vbCrLf
    Next hitung

EndSub
EndClass
```

Script yang menandakan kelipatan 10



Gambar 3.5 Perulangan dengan kelipatan 10

Pernyataan Exit For

Pernyataan Exit For berfungsi untuk keluar dari Looping For.....Next sebelum loop selesai dieksekusi. Dengan cara ini, kita bisa merespon event tertentu yang terjadi sebelum loop selesai dijalankan sesuai dengan jumlah yang telah ditentukan.

Contoh :

```
Private Sub Form_Activate()  
  
Dim i As Integer  
  
Dim inputName As String  
  
For i = 1 To 10  
  
    inputName = InputBox ("Masukkan sembarang nama  
atau ketik Done untuk keluar")
```

```
If inputName = "Done" Then
    Exit For ----->      Keluar dari perulangan
End If
Print inputName
Next i
End Sub
```

3.5 Perulangan Berulang

Bagian yang diulang pada perintah For.....Next dapat berupa sembarang perintah, termasuk termasuk perintah For.....Next. Jika dibuat perintah looping di dalam bagian program yang diulang, maka ini disebut perulangan berulang. Secara umum perulangan berulang akan berbentuk seperti penggalan program berikut :

```
For Variabel = Start To end1
    For Variabel2 = start2 To end2
        Pernyataan
    Next Variabel2
Next Variabel1
```

Contoh program :

```
PublicClassForm2
PrivateSub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    End
EndSub
```

```

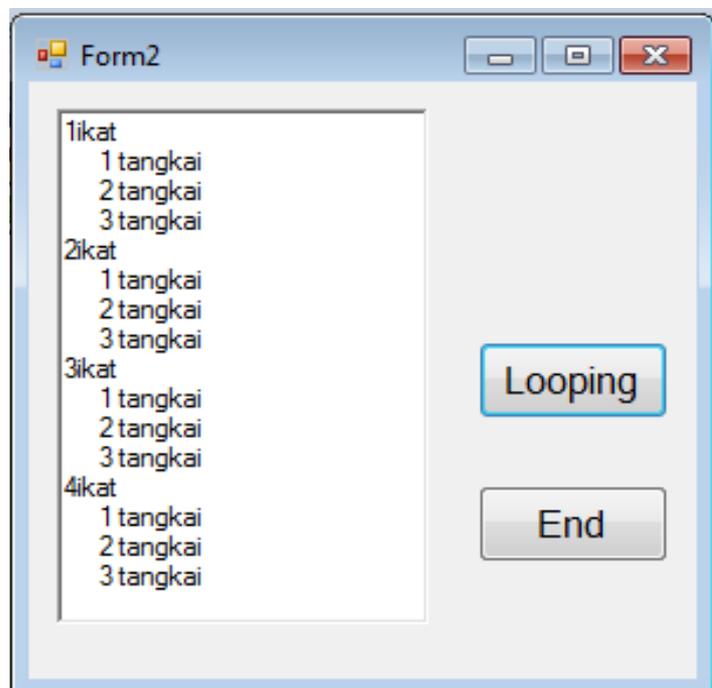
PrivateSub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim j As Integer
    Dim I As Integer
    For j = 1 To 4
        RichTextBox1.Text = RichTextBox1.Text
        & j & "ikat" & vbCrLf

        For i = 1 To 3
            RichTextBox1.Text = RichText-
                & "      " & I & " tangkai " & vbCrLf
            Next i
        Next j
    EndSub
EndClass

```

Perulangan Berulang

RichTextBox1.Text



Gambar 3.6 Hasil Perulangan Bersarang

Penggalan program diatas mengilustrasikan proses perulangan-berulang. Terdapat empat ikat bunga dimana masing-masing terdiri dari tiga buah tangkai bunga. Dari proses looping diatas, variabel “j” mewakili perulangan ikatan bunga sedangkan variabel “i” mewakili perulangan ikatan bunga. Setiap kali variabel “i” melakukan proses perulangan sebanyak tiga kali (tiga tangkai) maka akan menambahkan nilai variabel “j”.

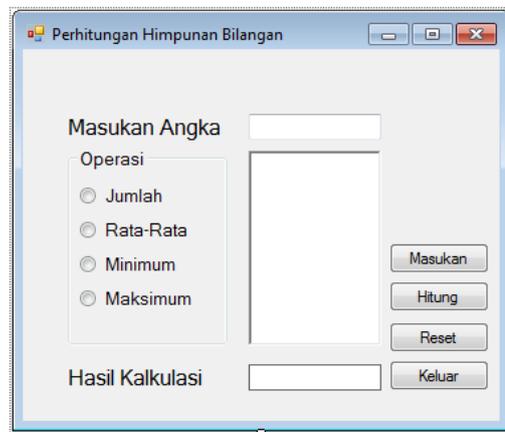
Setiap penambahan variabel “j” akan memberikan respon satu ikat telah dilaksanakan. Variabel “j” dilakukan sebanyak empat kali, sehingga total perulangan yang dilakukan adalah sebanyak “i” x “j” = 3 x 4 = 12 kali

3.6 Studi Kasus Tentang Penggunaan Struktur Perulangan

Berikut contoh program aplikasi yang membutuhkan struktur perulangan/looping dan juga struktur keputusan If maupun Select Case.

Studi Kasus 1: Perhitungan Himpunan Bilangan

Buatlah Program untuk menghitung jumlah, rata-rata, nilai maksimum dan nilai minimum dari sederetan/himpunan bilangan. Untuk keperluan ini buatlah project baru dan simpan dengan nama : **FrmHitungHimpunan**, nama project **PrjHitungHimpunan**. Ganti properti name dari form menjadi **Perhitungan Himpunan Bilangan**. Masukkan beberapa komponen ke dalam form seperti tampilan pada gambar di bawah ini :



Gambar 3.7 Tampilan Form Perhitungan Himpunan Bilangan

Skenario Program :

Tombol “**Masukan**” digunakan untuk memasukkan angka yang ditulis dalam Text Input sehingga masuk ke dalam List1. Tombol “**Hitung**” digunakan untuk menghitung hasil operasi yang ditentukan oleh opsi-opsi pada Radio Button operasi (RadioJumlah, RadioRerata, RadioMinimum dan RadioMaksimum). Hasil dari perhitungan akan dituliskan pada label Hasil kalkulasi. Tombol “**Keluar**” digunakan untuk menutup form.

Dibawah ini terdapat tabel yang merinci komponen berikut dengan properties yang dibutuhkan untuk membuat program aplikasi perhitungan himpunan bilangan

Obyek	Properti	Pengaturan
RadioButton1	Name	RadioJumlah
	Text	Jumlah
RadioButton2	Name	RadioRerata
	Text	Rata-rata
RadioButton3	Name	RadioMinimum
	Text	Minimum
RadioButton4	Name	RadioMaksimum
	Text	Maksimum
GroupBox1	Name	GroupBox1
	Text	Operasi
Button1	Name	Button1
	Text	“Masukan”
Button2	Name	Button2
	Text	“hitung”
Button3	Name	Button3
	Text	“Exit”

Button4	Name	Button4
	Text	"Reset"
TextBox1	Name	TextBox1
	Text	-
TextBox2	Name	TextBox2
	Text	-
Label1	Name	Label1
	Text	"Masukan Angka"
	Font	Microsoft Sans Serif; 12pt
Label2	Name	Label1
	Text	"Hasil Kalkulasi"
	Font	Microsoft Sans Serif; 12pt

Berikut adalah kutipan *script* untuk program diatas

```

Public Class Studi_kasus1

    Dim n As Integer

    Dim hasil As Single

    Dim data(100) As Integer

    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        n = n + 1

        data(n) = TextBox1.Text

```

```
RichTextBox1.Text = data(n) & vbCr &  
RichTextBox1.Text
```

```
TextBox1.Text = ""
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button2.Click
```

```
Dim i As Integer
```

```
    Select Case True
```

```
        Case RadioJumlah.Checked
```

```
            For i = 1 To n
```

```
                hasil = hasil + data(i)
```

```
            Next i
```

```
        Case RadioRerata.Checked
```

```
            For i = 1 To n
```

```
                hasil = hasil + data(i)
```

```
            Next i
```

```
            hasil = hasil / n
```

```
        Case RadioMaksimum.Checked
```

```
            hasil = data(1)
```

```
            For i = 1 To n
```

```
                If hasil >= data(i) Then
```

```
        hasil = hasil

    Else

        hasil = data(i)

    End If

Next i

Case RadioMinimum.Checked

    hasil = data(1)

    For i = 1 To n

        If hasil <= data(i) Then

            hasil = hasil

        Else

            hasil = data(i)

        End If

    Next i

End Select

TextBox3.Text = hasil

End Sub

Private Sub Studi_kasus1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

    n = 0

    hasil = 0
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button3.Click
```

```
End
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button4.Click
```

```
For i = 1 To n
```

```
    data(i) = 0
```

```
Next i
```

```
n = 0
```

```
hasil = 0
```

```
RichTextBox1.Text = ""
```

```
TextBox1.Text = ""
```

```
TextBox3.Text = ""
```

```
End Sub
```

```
End Class
```

Studi Kasus 2 : Perhitungan jumlah dan Rata-rata Bilangan Genap dan Bilangan Ganjil dari Range Bilangan (Aplikasi penggunaan While....When)

Pada studi kasus ini, kita akan membuat program untuk menghitung jumlah dari bilangan genap dan bilangan ganjil dari sederetan bilangan dengan range yang kita tentukan. Buatlah project baru, simpan form dengan nama :**FrmGenapGanjil** dan nama project **PrjGenapGanjil**. Ganti properti name dari form tersebut menjadi **FrmGenapGanjil**. Masukkan beberapa komponen dan sesuaikan dengan tampilan form seperti gambar berikut ini :



Gambar 3.8 Tampilan Form Genap Ganjil

Skenario program

Tombol “**Genap**” dan tombol “**Ganjil**” digunakan untuk menghitung jumlah dan rata-rata dari seluruh bilangan genap dan bilangan ganjil dari sederetan bilangan dengan range batas bawah pada text :TxtBilAwal sampai dengan batas atas text : TxtBilAkhir.

Berikut terdapat tabel yang merinci komponen dan properties yang dibutuhkan untuk membuat program aplikasi perhitungan himpunan bilangan



Obyek	Properti	Pengaturan
RadioButton1	Name	RadioJumlah
	Text	Jumlah
RadioButton2	Name	RadioRerata
	Text	Rata-rata
RadioButton3	Name	RadioMinimum
	Text	Minimum
RadioButton4	Name	RadioMaksimum
	Text	Maksimum
GroupBox1	Name	GroupBox1
	Text	Operasi
Button1	Name	Button1
	Text	"Masukan"
Button2	Name	Button2
	Text	"hitung"
Button3	Name	Button3
	Text	"Exit"
Button4	Name	Button4
	Text	"Reset"
TextBox1	Name	TextBox1
	Text	-
TextBox2	Name	TextBox2
	Text	-

Label1	Name Text Font	Label1 "Masukan Angka" Microsoft Sans Serif; 12pt
Label2	Name Text Font	Label1 "Hasil Kalkulasi" Microsoft Sans Serif; 12pt

Berikut adalah kutipan routine untuk program diatas :

```

Public Class Genap_Ganjil
Dim jumlah As Integer
Dim rata_rata As Single
Dim i As Integer
Dim hitungan As Integer

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

i = Val(TextBox1.Text)
jumlah = 0
hitungan = 0

While i < Val(TextBox2.Text)
    i = i + 1
    If i Mod 2 = 0 Then
        jumlah = jumlah + i
        hitungan = hitungan + 1
    End If
End While

rata_rata = jumlah / hitungan
TextBox3.Text = jumlah
TextBox4.Text = rata_rata

End Sub

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

i = Val(TextBox1.Text)
jumlah = 0
hitungan = 0

```

```
While i < Val(TextBox2.Text)
    i = i + 1
    If i Mod 2 <> 0 Then
        jumlah = jumlah + i
        hitungan = hitungan + 1
    End If
End While

rata_rata = jumlah / hitungan
TextBox5.Text = jumlah
TextBox6.Text = rata_rata

End Sub

Private Sub Button3_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button3.Click
    End
End Sub
End Class
```

TUGAS

1. Perhatikan perbedaan antara perintah perulangan Do....Loop, While.....When dan For.....Next. pelajari pula bagaimana caranya jika kita ingin mengakhiri proses perulangan sebelum proses tersebut selesai.

TES FORMATIF

Buatlah program

Suatu perusahaan air minum kemasan membuat diskon harga untuk pembelian produknya dalam jumlah tertentu seperti yang tertera dalam tabel berikut ini :

Pembelian	Diskon	Harga Per Unit
1-10 Box	0%	Rp. 7.500,00
11-25 Box	5%	Rp. 7.500,00
26-50 Box	10%	Rp. 7.500,00
>50 Box	15%	Rp. 7.500,00

Tentukan berapa total uang yang harus dibayarkan oleh konsumen untuk pembelian tertentu!



Jawab :

KEGIATAN 4. MENGGUNAKAN FUNGSI TIMER

Timer merupakan salah satu komponen non visual yang digunakan untuk melakukan perintah berdasarkan waktu tertentu. Fungsi timer pada Visual Basic hampir sama dengan jam yang biasa kita gunakan sehari-hari. Komponen ini akan merespon kode yang diketikkan pada sub-timer dengan memasukkan properties interval yang kita masukkan.

Seperti komponen-komponen lain, timer juga mempunyai beberapa properti. Timer memiliki properti interval yang melakukan kejadian dalam satuan milidetik, jadi apabila kita menginginkan komponen timer tersebut merespon setiap satu detik maka kita harus memberikan nilai 1000 pada kolom interval. Properti Enable digunakan untuk menjalankan (True) atau memberhentikan (False).

4.1 Membuat Jam Digital menggunakan Kontrol Timer

Salah satu penggunaan dari kontrol Timer yang paling sering digunakan adalah untuk membuat jam digital. Pada contoh latihan ini, Anda akan membuat jam digital yang akan menampilkan jam, menit dan detik. Pada contoh ini Anda akan mengatur properti dari interval sebesar 1000 yang berarti timer akan mengupdate waktu setiap 1000 milidetik atau satu kali dalam setiap detik.

Langkah-langkah membuat program Jam Digital :

1. Pada menu File, kliklah New Project dan pada Windows Form Application buatlah project baru yang dinamakan Aplikasi_Timer.
2. Ubahlah ukuran form menjadi bentuk persegi panjang kecil karena untuk membuat jam digital kita tidak membutuhkan form berukuran besar.
3. Pilihlah komponen Timer yang terdapat pada Toolbox.
4. Tambahkan komponen Label pada Toolbox dan buatlah label berukuran besar di dalam form yang nantinya akan digunakan untuk menampilkan jam digital.



Gambar 4.1 Form Awal Program Jam digital

5. Bukalah Properti window, dan atur properties dari masing-masing obyek seperti pada tabel dibawah ini.

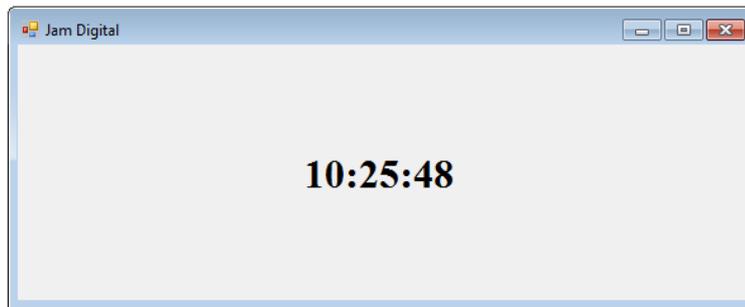
Obyek	Properti	Pengaturan
Label 1	AutoSize	False
	Font	Times New Roman, Bold, 24-point
	Text	(empty)
	TextAlign	MiddleCenter
Timer1	Enabled	True
Form1	Text	"Jam Digital"

7. Double klik ikon timer yang terdapat pada lembar kerja maka akan muncul event prosedur timer pada code editor kemudian ketiklah statement berikut ini

```
Label1.Text = TimeString
```

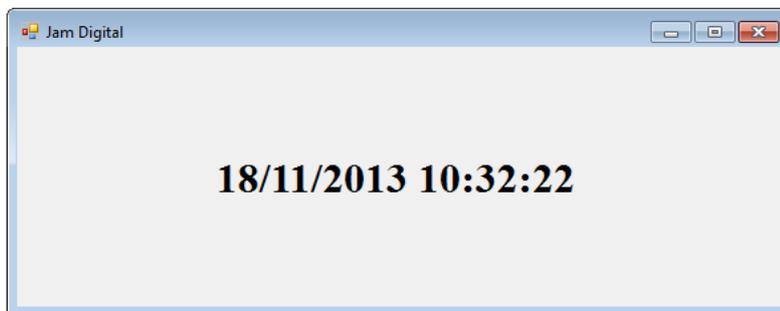
Potongan statement tersebut mempunyai arti yaitu setiap interval 1000 ms dari **Timer1** (tiap 1 detik), program akan mengupdate data perhitungan sistem waktu dari komputer dan mengirimkannya ke properti Text pada obyek **Label1**.

8. Klik Save All Button pada toolbar untuk menyimpan file Anda
9. Klik Start Debugging pada Toolbar untuk menjalankan program jam digital ini. Jam digital akan tampak seperti pada gambar dibawah ini



Gambar 4.2 Hasil Aplikasi Jam Digital

Jika Anda ingin menambah tampilan tanggal, bulan dan tahun maka kode program `TimeString` dapat diganti `System.DateTime.Now` dan hasil tampilan aplikasi akan tampak seperti seperti gambar berikut :



Gambar 4.3 Hasil Aplikasi Jam Digital Dengan Tanggal

Perhatikanlah jam digital tersebut, tampak program Visual Basic mengupdate waktu setiap detiknya. Berikut adalah kode program dari aplikasi program di atas

```
(General) | (Declarations)
Public Class Jam_Digital
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
        Label1.Text = System.DateTime.Now
    End Sub
End Class
```

4.2 Menggunakan Obyek Timer Untuk Mengatur Batas Waktu

Penggunaan obyek Timer lainnya adalah untuk mengatur jeda waktu atau waktu tunda dari suatu aplikasi. Tujuan dari mengatur waktu tersebut adalah untuk memberikan batasan waktu kepada pengguna apakah aplikasi tersebut dapat dilanjutkan atau tidak. Jika Pengguna tidak memberikan data yang benar dalam batas waktu yang ditentukan, maka program akan keluar. Contoh aplikasi yang sering menggunakan fungsi ini adalah untuk memberi batasan waktu kepada pengguna untuk memasukan password.

Aplikasi lain yang dapat dikembangkan untuk obyek timer ini antara lain untuk menampilkan pesan selamat datang atau menampilkan obyek lampu lalu lintas atau untuk mengulang suatu event pada interval tertentu seperti menyimpan sebuah file setiap 10 menit atau *memback up* sebuah data setiap pukul 20.00 pagi dan lain-lain.

Anda dapat mengatur properti interval dengan waktu tunda yang Anda inginkan kemudian Anda dapat menjalankannya dengan mengatur properti Enabled menjadi True. Selama program masih berjalan maka obyek timer Anda akan selalu aktif. Pada contoh dibawah ini akan dijelaskan langkah-langkah bagaimana membuat batas waktu dalam memasukkan password. Password dari program ini adalah "RAHASIA". Program menggunakan timer untuk menutup program jika dalam waktu 15 detik pengguna tidak memasukkan password dengan benar.

1. Tambahkan form baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama My Time Password.
2. Ubahlah ukuran form menjadi persegi panjang kecil
3. Klik komponen **TextBox1** pada Toolbox dan letakan di tengah-tengah form sebagai tempat menuliskan password pada program aplikasi.
4. Tambahkan komponen **Label1** pada form dan letakan di atas komponen **TextBox1**.
5. Klik kontrol **Button1** pada Toolbox dan letakan tombol button tersebut dibawah text box.

6. Tambahkan komponen Timer1 dengan men-double click daftar komponen tersebut di toolbox. Komponen akan otomatis terletak di bagian kiri bawah dari jendela Design.



Gambar 4.4 Form Komponen Aplikasi Timer Dengan Batas Waktu

7. Aturlah properti dari masing-masing komponen

Obyek	Properti	Pengaturan
Label1	Text	"Masukkan password Anda
TextBox1	PasswordChar	" * "
Button1	Text	"Ulangi Password"
Timer1	Enabled	True
	Interval	15000
Form1	Text	"Password"

PasswordChar mengatur display dari karakter (*) pada text box selama pengguna memasukkan password. Pengaturan properti interval sebesar 15000 dimaksudkan bahwa pengguna memiliki kesempatan selama 15 detik untuk memasukkan password dan mengklik tombol Ulangi Password. Pengaturan properti Enabled menjadi True diartikan bahwa timer bekerja jika program telah dimulai. Gambar di bawah ini merupakan tampilan form Anda :



Gambar 4.5 Tampilan Awal Aplikasi Timer Dengan Batas Waktu

8. Double klik pada obyek timer dan ketiklah pernyataan pada *event procedure* dibawah ini :

```
MsgBox ("Maaf, waktu Anda telah habis")
```

```
End
```

Pernyataan pertama menampilkan sebuah pesan bahwa waktu telah habis dan pernyataan kedua menghentikan program. Visual Basic akan mengeksekusi event prosedur ini jika interval dari timer melebihi 15 detik dan tidak memasukkan password dengan benar.

9. Pada lembar form, double klik obyek button1 dan ketiklah pernyataan dibawah ini :

```
If TextBox1.Text = "RAHASIA" Then
```

```
    Timer1.Enabled = False
```

```
    MsgBox ("Selamat Datang Di Sistem")
```

```
End
```

```
Else
```

```
    MsgBox ("Maaf Anda Tidak Dikenali")
```

```
End If
```

Program ini akan bekerja ketika password yang dimasukkan pada text box adalah “RAHASIA”. Jika benar maka timer akan non-aktif (disabled), pesan “**Selamat Datang Di Sistem**” akan muncul dan program akan berakhir (pada tahap berikutnya program akan melanjutkan ke program selanjutnya). Jika password yang dimasukkan salah pengguna akan diberi pesan pada message box “**Maaf Anda tidak Dikenali**” dan diberikan kesempatan untuk memasukkan kembali password dalam waktu 15 detik.

Klik Save All pada toolbar untuk menyimpan program tersebut.

4.3 Menguji Program Password

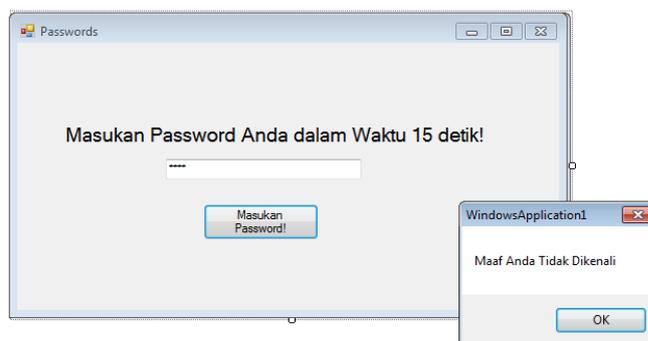
Klik tombol Start Debugging untuk menjalankan program.

1. Ketiklah **open** pada text box, seperti tampak pada gambar dibawah ini



2. Klik tombol Ulangi Password.

Pesan pada message box akan muncul di layar jika password yang Anda masukkan salah.



Gambar 4.6 Tampilan Dengan Hasil Inputan Salah Password

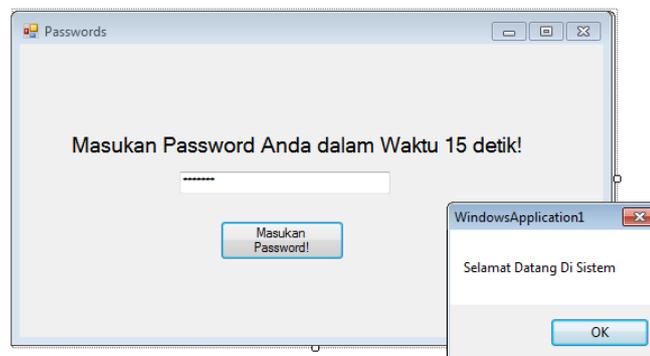
3. Klik OK kemudian tunggu sampai waktu yang telah ditentukan habis.

Program akan menampilkan pesan waktu habis pada message box



Gambar 4.7 Tampilan Dengan Hasil Inputan Waktu Terlalu Lama

4. Klik OK untuk menghentikan program.
5. Jalankan program kembali, ketiklah **“RAHASIA”** (password yang tepat) kemudian klik Ulangi Password. Maka program akan menampilkan pesan



Gambar 4.8 Tampilan Dengan Hasil Inputan Password Benar

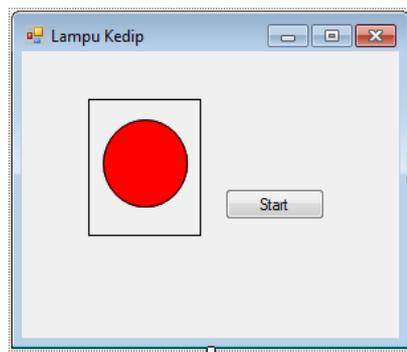
6. Klik OK untuk keluar dari program.

4.4 Membuat Aplikasi Visualisasi Lampu LED Berkedip

Setelah kita mengenal dasar penggunaan Timer, maka kita akan mencoba untuk membuat suatu program aplikasi dimana akan menampilkan visualisasi lampu LED berkedip. Secara visual, tampilan program aplikasi akan menampilkan warna merah dan hijau dari suatu obyek lampu LED secara bergantian.

Program ini merupakan dasar latihan sederhana, dimana kita akan menggunakan timer sebagai pengatur waktu transisi obyek dari warna merah menjadi hijau. Dengan mengatur timer, kita dapat menentukan saat kapan obyek harus di aktifkan (*visible = true*) dan saat kapan di non-aktifkan (*visible = false*).

Tampilan dari desain aplikasi adalah sebagai berikut :

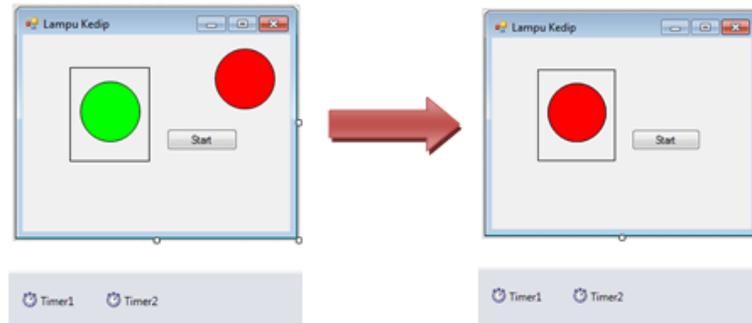


Gambar 4.8 Tampilan Aplikasi Lampu Kedip

Untuk membuat program aplikasi diatas, ikutilah langkah-langkah berikut :

1. Tambahkan form baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama Lampu Kedip.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Klik komponen **RectangleShape1** pada Toolbox dan letakan di form sebagai bingkai dari lampu LED program dari aplikasi.

4. Tambahkan komponen **OvalShape1** dan **OvalShape2** pada form dan letakan di antara **RectangleShape1**, tumpuk kedua komponen ovalshape tersebut hingga tampak hanya satu ovalshape saja.



5. Arahkan kursor ke toolbox dan tambahkan komponen **timer1** dan **timer2** dengan cara mendouble klik komponen timer pada toolbox tersebut.
6. Komponen terakhir yang diperlukan adalah **Button1**, ambil komponen Button1 tersebut dari toolbox dan atur semua properti dari tiap-tiap komponen tersebut seperti tabel dibawah ini.

Obyek	Properti	Pengaturan
RectangleShape1	Name	RectangleShape1
	Visible	True
OvalShape1	Name	OvalShape1
	BackColor	Lime
OvalShape2	Name	OvalShape2
	BackColor	Red
Button1	Name	Button1
	Text	"Start"
Timer1	Enabled	False
	Interval	1000
Timer2	Enabled	False
	Interval	1000
Form1	Text	"Lampu Kedip"

Interval timer 1000 berfungsi sebagai lama waktu komponen ovalshape akan ditampilkan. Timer1 berfungsi untuk mengatur lama waktu komponen OvalShape1 berwarna hijau ditampilkan kemudian digantikan OvalShape2 berwarna merah dengan lama waktu dari interval timer2.

7. Jalankan aplikasi program, kemudian klik lah tombol start. Akan tampak bahwa program aplikasi akan membuat lampu berkedip merah-hijau secara bergantian.

Berikut script program secara keseluruhan :

```
Public Class Lampu_kedip
    Private Sub Button1_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button1.Click
        Timer1.Enabled = True
    End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Timer1.Tick  
  
    Timer2.Enabled = False  
  
    If OvalShape1.Visible = False Then  
        OvalShape1.Visible = True  
        OvalShape2.Visible = False  
    End If  
  
    Timer2.Enabled = True  
  
End Sub  
  
Private Sub Timer2_Tick(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Timer2.Tick  
  
    Timer1.Enabled = False  
  
    If OvalShape2.Visible = False Then  
        OvalShape2.Visible = True  
        OvalShape1.Visible = False  
    End If  
  
    Timer1.Enabled = True  
  
End Sub  
  
End Class
```

Pembahasan program

Aplikasi berjalan ketika tombol “**Start**” mulai ditekan. Ketika tombol start ditekan potongan program berikut akan di eksekusi

```
Timer1.Enabled = True
```

Potongan script di atas artinya komponen Timer1 akan di jalankan dan semua script program yang ada di dalam sub timer1 akan di eksekusi.

Langkah selanjutnya kita memasuki potongan program yang ada di dalam sub timer1. Baris pertama dari program di sub timer1 adalah sebagai berikut:

```
Timer2.Enabled = False
```

Tujuan dari sub program timer1 adalah untuk menampilkan gambar lampu led berwarna hijau (Ovalshape1). Sedangkan tujuan dari sub program timer2 adalah untuk menampilkan gambar lampu led berwarna merah (OvalShape2). Karena kita ingin menampilkan kedua gambar tersebut secara bergantian, maka kita mengatur secara bergantian setiap sub timer tersebut.

Berikut potongan sub program Timer1

```
Timer2.Enabled = False

If OvalShape1.Visible = False Then
    OvalShape1.Visible = True
    OvalShape2.Visible = False
End If

Timer2.Enabled = True

End Sub
```

Potongan program di atas artinya, ketika sub program Timer1 di aktifkan maka pada baris pertama timer2 akan di non aktifkan. Kemudian baris kedua merupakan kondisi bersyarat, yang menanyakan apakah komponen OvalShape1 (Led Hijau) tidak tampak pada form aplikasi?. Jika OvalShape1 (led Hijau) tidak tampil pada form aplikasi maka program baris ketiga akan menampilkan OvalShape1(Led Hijau) pada form aplikasi, dengan cara membuat properties **visible** dari OvalShape1 menjadi **True**. Baris ke empat menunjukan untuk menon-aktifkan OvalShape2 (Led Merah), dengan membuat properties **Visible** dari Ovalshape2 menjadi **False**.

Kondisi bersyarat dari sub program timer1 di akhiri dengan *end if*, selanjutnya mengeksekusi script program

```
Timer2.Enabled = True
```

Artinya sub program Timer2 yang bertujuan untuk menampilkan Led merah akan di eksekusi. Berikut potongan sub program Timer2.

```
Timer1.Enabled = False

    If OvalShape2.Visible = False Then

        OvalShape2.Visible = True

        OvalShape1.Visible = False

    End If

Timer1.Enabled = True

End Sub
```

Sub program timer2 identik dengan sub program timer1, hanya perbedaan komponen saja. Baris pertama sub program Timer2 untuk men-non aktifkan sub program timer 1. Kemudian baris kedua merupakan kondisi bersyarat, yang menanyakan apakah komponen OvalShape2 (Led merah) non aktif?. Jika OvalShape2 (Led Merah) tidak tampil pada form aplikasi maka program baris ketiga akan menampilkan OvalShape2(Led Merah) pada form aplikasi, dengan cara membuat properties **visible** dari OvalShape2 menjadi **True**. Baris ke empat menunjukkan untuk menon-aktifkan OvalShape1 (Led Hijau), dengan membuat properties **Visible** dari Ovalshape1 menjadi **False**.

TUGAS

1. Perhatikan sifat properties timer interval dan enable kemudian jelaskan arti serta fungsi masing-masing properti tersebut!.
2. Buatlah suatu aplikasi bebas yang terdiri dari tiga buah timer, pelajari bagaimana cara untuk mengaktif dan menon-aktifkan ketiga timer tersebut secara bergantian!

TES FORMATIF

Buatlah program Digital Stopwatch yang akan menghitung lama waktu dari suatu kejadian! Program aplikasi terdiri tombol “Start”, tombol “Stop”, tombol “Reset” dan label display yang akan menampilkan tampilan angka digital. Berikut tampilan gambar aplikasi yang akan disusun.



Jawab :

KEGIATAN 5. Membuat Program Aplikasi Dengan Visual Basic 2010

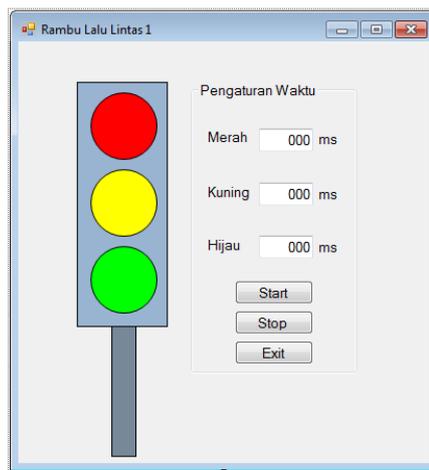
Setelah mempelajari dasar-dasar pemrograman visual basic pada bab-bab sebelumnya, sekarang kita akan belajar membuat berbagai macam program aplikasi yang akan menyatukan semua dasar teori pemrograman yang telah dipelajari.

5.1 Program Lampu Lalu Lintas 1

Setelah kita mengetahui bagaimana membuat lampu led berkedip, maka kita akan mencoba untuk membuat suatu program aplikasi dimana akan menampilkan visualisasi lampu lalu lintas yang terdiri tiga buah lampu yaitu merah, kuning dan hijau. Waktu pengaturan setiap lampu lalu-lintas tersebut dapat kita atur dengan memasukan nilai input pada input box.

Program ini menekankan bagaimana mengatur beberapa timer dalam satu aplikasi. Fungsi dari sub program timer sebagai pengatur waktu transisi obyek dari warna merah menjadi kuning kemudian menjadi hijau. Dengan mengatur timer, kita dapat menentukan saat kapan obyek harus di aktifkan (*visible = true*) dan saat kapan di non-aktifkan (*visible = false*).

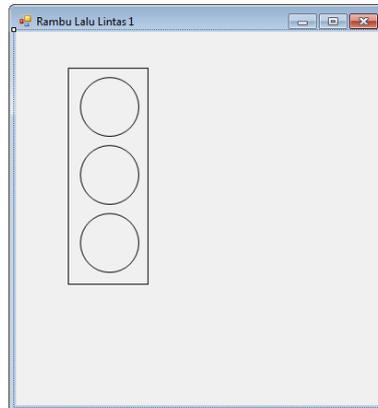
Tampilan dari desain aplikasi adalah sebagai berikut :



Gambar 5.1 Tampilan Program Aplikasi Rambu Lalu Lintas 1

Untuk membuat program aplikasi diatas, ikutilah langkah-langkah berikut :

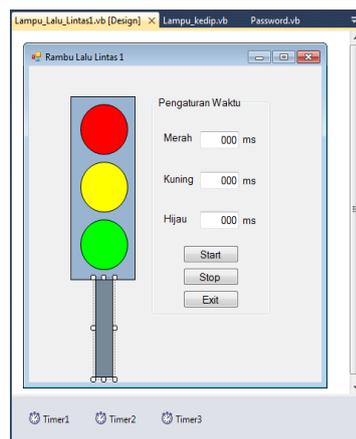
1. Tambahkan form baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama Lampu_Lalu_Lintas1.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Klik komponen **RectangleShape1** pada Toolbox dan letakan di form sebagai bingkai dari lampu-lampu lalu lintas merah, kuning dan hijau.
4. Tambahkan komponen **OvalShape1**, **OvalShape2** dan **OvalShape3** pada form dan letakan di dalam **RectangleShape1**. Atur letak ketiga OvalShape tersebut berderet secara vertikal seperti gambar berikut.



Gambar 5.2 Komponen Program Aplikasi Rambu Lalu Lintas 1

5. Tambahkan komponen **GroupBox1** dan ubah properties text komponen tersebut menjadi "Pengaturan Waktu".
6. Tambahkan komponen **TextBox1**, **TextBox2** dan **TextBox3** dan letakkan ketiga komponen tersebut di dalam komponen GroupBox1. Ubahlah properties ketiga komponen tersebut menjadi "000".

7. Tambahkan komponen **Button1**, **Button2** dan **Button3** dan letakkan di dalam GroupBox1 dan atur properties text komponen Button1 ="Start", Button2="Stop" dan Button3="Exit".
8. Tambahkan komponen **label1**, **label2**, **label3**, **label4**, **label5** dan **label6** sebagai penanda teks untuk memperjelas fungsi dari aplikasi program yang kita buat.
9. Tambahkan **Timer1**, **Timer2** dan **Timer3** dengan mendouble klik komponen Timer pada Toolbox. Ketiga komponen Timer tersebut akan otomatis berlokasi dibawah dari jendela form design yang kita buat sehingga tampak seperti gambar berikut



Gambar 5.3 Tampilan Akhir Program Aplikasi Rambu Lalu Lintas 1

10. Secara umum semua komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut.

Obyek	Properti	Pengaturan
RectangleShape1	Name	RectangleShape1
	Visible	True
	BackColor	ActiveCaption
RectangleShape2	Name	RectangleShape1
	Visible	True
	BackColor	LightSlightGray
OvalShape1	Name	OvalShape1
	BackColor	Red
	BackStyle	Opaque
OvalShape2	Name	OvalShape2
	BackColor	Yellow
	BackStyle	Opaque
OvalShape3	Name	OvalShape3
	BackColor	Lime
	BackStyle	Opaque
GroupBox1	Name	GroupBox1
	Text	"Pengaturan Waktu"
Button1	Name	Button1
	Text	"Start"
Button2	Name	Button2
	Text	"Stop"
Button3	Name	Button3
	Text	"Exit"
TextBox1	Name	TextBox1
	Text	"000"

TextBox2	Name	TextBox2
	Text	"000"
TextBox3	Name	TextBox3
	Text	"000"
Label1	Name	Label1
	Text	"Merah"
Label2	Name	Label2
	Text	"Kuning"
Label3	Name	Label3
	Text	"Hijau"
Label4	Name	Label4
	Text	"ms"
Label5	Name	Label5
	Text	"ms"
Label6	Name	Label6
	Text	"ms"
Timer1	Enabled	False
	Interval	1000
Timer2	Enabled	False
	Interval	1000
Timer3	Enabled	False
	Interval	1000
Form1	Text	"Rambu Lalu Lintas"

Berikut script program secara keseluruhan :

```
Public Class Lampu_Lalu_Lintas1

    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Timer1.Interval = TextBox3.Text

        Timer2.Interval = TextBox1.Text

        Timer3.Interval = TextBox2.Text

        Timer1.Start()

    End Sub

    Sub Timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Timer1.Tick

        If OvalShapel1.BackColor = Color.DarkGray Then

            OvalShapel1.BackColor = Color.Red

            OvalShape2.BackColor = Color.DarkGray

            OvalShape3.BackColor = Color.DarkGray

        End If

        Timer2.Start()

        Timer1.Stop()

        Timer3.Stop()

    End Sub

End Class
```

```
Private Sub Timer2_Tick(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles Timer2.Tick  
  
    If OvalShape2.BackColor = Color.DarkGray Then  
  
        OvalShape1.BackColor = Color.DarkGray  
  
        OvalShape2.BackColor = Color.Yellow  
  
        OvalShape3.BackColor = Color.DarkGray  
  
    End If  
  
    Timer3.Start()  
  
    Timer1.Stop()  
  
    Timer2.Stop()  
  
End Sub  
  
Private Sub Timer3_Tick(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Timer3.Tick  
  
    If OvalShape3.BackColor = Color.DarkGray Then  
  
        OvalShape1.BackColor = Color.DarkGray  
  
        OvalShape2.BackColor = Color.DarkGray  
  
        OvalShape3.BackColor = Color.Lime  
  
    End If  
  
    Timer1.Start()  
  
    Timer2.Stop()  
  
    Timer3.Stop()  
  
End Sub
```

```
Private Sub Button2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button2.Click
```

```
    Timer1.Enabled = False
```

```
    Timer2.Enabled = False
```

```
    Timer3.Enabled = False
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button3.Click
```

```
    End
```

```
End Sub
```

```
Private Sub Lampu_Lalu_Lintas1_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
    OvalShape1.BackColor = Color.DarkGray
```

```
    OvalShape2.BackColor = Color.DarkGray
```

```
    OvalShape3.BackColor = Color.DarkGray
```

```
End Sub
```

```
End Class
```

Pembahasan program

Tujuan dari sub program timer1 adalah untuk menampilkan gambar lampu berwarna merah (OvalShape1). Sedangkan tujuan dari sub program timer2 adalah untuk menampilkan gambar lampu berwarna kuning (OvalShape2) dan untuk sub program timer3 berfungsi untuk menampilkan lampu warna hijau (OvalShape3). Karena kita ingin menampilkan ketiga gambar tersebut secara bergantian, maka kita mengatur secara bergantian setiap sub timer tersebut dan juga mengatur berapa lama waktu tampilan dari masing-masing lampu tersebut berdasar properties interval dari setiap timer.

Aplikasi berjalan dimulai dengan memasukan nilai input timer pada InputBox1, InputBox2 dan InputBox3. Pengguna akan memasukan nilai dalam satuan mili detik, dimana nilai tersebut yang akan menentukan lama waktu penyalaan dari lampu merah, kuning maupun hijau.

ketika tombol **“Start”** ditekan potongan program berikut akan di eksekusi

```
Timer1.Interval = TextBox3.Text  
  
Timer2.Interval = TextBox1.Text  
  
Timer3.Interval = TextBox2.Text  
  
Timer1.Start()
```

Arti dari potongan program di atas adalah mengatur nilai properties interval dari timer1, timer2 dan timer3 yang didapatkan dari InputBox1, InputBox2 dan inputBox3. Jadi ketika pengguna mengetikkan nilai angka satuan dalam mili detik maka nilai-nilai tersebut yang akan masuk ke dalam interval dari masing-masing timer. Pada baris terakhir atau baris ke empat, potongan program di atas mengindikasikan memanggil atau mengaktifkan Timer1 dan mengerjakan sub program bagian yang ada di dalam Timer1.

Tahap selanjutnya adalah mengeksekusi sub program Timer1. Default program mengatur semua lampu untuk non-aktif, dengan memberikan warna abu-abu “Darkgray” pada semua properties BackColour dari lampu. Sehingga pada awal program kita membuat suatu perintah syarat “IF” jika kondisi warna OvalShape1 abu-abu, maka program akan merubahnya menjadi warna merah sedangkan

untuk lampu warna lain tetap non-aktif atau berwarna abu-abu. Berikut potongan program yang merepresentasikan syarat tersebut

```
If OvalShapel.BackColor = Color.DarkGray Then  
  
    OvalShapel.BackColor = Color.Red  
  
    OvalShape2.BackColor = Color.DarkGray  
  
    OvalShape3.BackColor = Color.DarkGray  
  
End If
```

Setelah menghidupkan lampu merah selanjutnya kita menyalakan lampu warna kuning, untuk itu kita harus mengaktifkan timer2 dan mengeksekusi perintah program di dalamnya. Untuk mengaktifkan atau memanggil sub program Timer2 maka diberikan perintah program berikut:

```
Timer2.Start()
```

Agar tidak terjadi konflik dalam program maka timer1 dan timer3 harus di non-aktifkan seperti script perintah di bawah ini

```
Timer1.Stop()
```

```
Timer3.Stop()
```

Sub program Timer2 identik dengan sub program timer1, bedanya hanyalah jika pada sub program timer1 bertujuan untuk menampilkan lampu berwarna merah, maka pada sub program ini akan menampilkan warna kuning. Seperti program sub timer1, maka program ini diawali dengan syarat jika backcolour dari OvalShape2 abu-abu maka backcolour OvalShape2 akan dirubah menjadi kuning. Berikut fungsi perintah untuk mengeksekusi situasi tersebut:

```
If OvalShape2.BackColor = Color.DarkGray Then  
    OvalShape1.BackColor = Color.DarkGray  
    OvalShape2.BackColor = Color.Yellow  
    OvalShape3.BackColor = Color.DarkGray  
End If
```

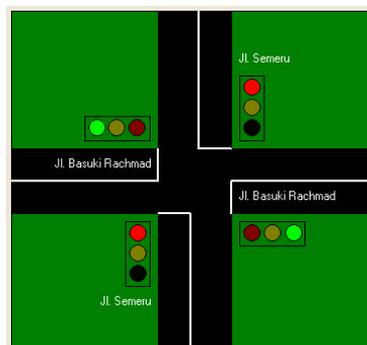
Setelah menghidupkan lampu kuning selanjutnya kita menyalakan lampu warna hijau, untuk itu timer3 harus di aktifkan sehingga semua perintah program di dalamnya sub program tersebut akan di eksekusi. Untuk mengaktifkan atau memanggil sub program Timer3 maka diberikan perintah program berikut:

```
Timer3.Start()  
Timer1.Stop()  
Timer2.Stop()
```

Sedangkan sub program timer yang lain akan di non-aktif kan agar tidak terjadi konflik pada program aplikasi seperti script yang tercetak di atas.

Berikutnya sub program timer3 digunakan untuk menampilkan lampu warna hijau, isi sub program timer3 identik dengan sub program timer1 dan sub program timer2. Dimulai dengan memberikan syarat dan merubah warna backcolour dari OvalShape3 menjadi hijau. Setelah itu kembali melakukan looping dengan mengaktifkan sub program timer1 dan seterusnya.

5.2 Program Lampu Lalu Lintas 2



Gambar 5.4. Model Persimpangan Lalu-Lintas

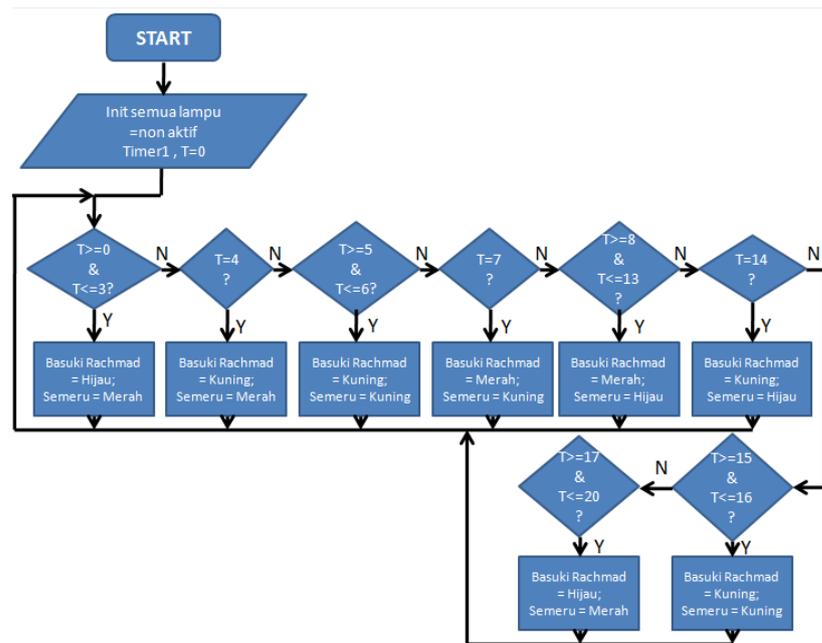


Program aplikasi selanjutnya adalah membuat program aplikasi lalu lintas pada persimpangan jalan raya seperti ilustrasi gambar di atas. Terdapat suatu pengaturan jalan lampu lalu lintas di persimpangan jalan basuki rachmad dan jalan semeru yang terdiri atas empat buah lampu lalu-lintas. Lampu lalu-lintas di persimpangan jalan raya tersebut memiliki sistem kontrol paralel, artinya lampu lalu-lintas di jalan basuki rachmad baik yang terletak di sebelah timur dan barat berjalan secara bersamaan, begitu juga dengan lampu lalu-lintas jalan semeru, baik yang di sisi utara maupun sisi selatan berjalan seiringan.

Berikut tabel kebenaran waktu penyalaan lampu lalu lintas pada persimpangan jalan basuki rachmad dan jalan semeru dalam satu siklus program. Satu siklus program terdiri dari 20 detik dan tiap satuan detik lampu lalu lintas yang aktif pada kedua jalan tersebut tampak pada tabel.

Tabel Diagram Waktu

Waktu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Jl.Basuki rachmad	Green	Green	Green	Yellow	Yellow	Yellow	Red	Red	Red	Red	Red	Red	Red	Yellow	Yellow	Yellow	Green	Green	Green	Green
Jl.Semeru	Red	Red	Red	Red	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Red	Red	Red	Red						



Gambar 5.5 Flowchart Program Lalu Lintas 2

Untuk membuat program aplikasi diatas, ikutilah langkah-langkah berikut :

1. Tambahkan form baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama Lampu_Lalu_Lintas2.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Klik form desain aplikasi, gantilah properties text menjadi "**Lampu Lalu Lintas 2**" dan ubah BackColour properties menjadi "**DarkGreen**".
4. Pertama gambarlah desain jalan semeru dan jalan basuki rachmad dengan menggunakan komponen **RectangleShape1** dan **RectangleShape2** pada kelompok **Visual Basic PowerPacks** yang terletak di Toolbox dan letakan pada form secara tegak lurus membentuk persimpangan.
5. Ubah *properties backcolour* **RectangleShape1** dan **RectangleShape2** menjadi hijau tua pada pilihan warna *custom* sehingga *properties backcolour* menjadi **0;64;0**. Kemudian ubah *properties BackStyle* menjadi **Opaque**.
6. Tambahkan garis pembatas jalan berwarna putih dengan menggunakan komponen **LineShape**.
7. Untuk membuat lampu lalu lintas, gunakan komponen **RectangleShape** sebagai bingkai dari lampu-lampu lalu lintas merah, kuning dan hijau. Ubah *properties backcolour* menjadi **Gray** kemudian ubah *properties BackStyle* menjadi **Opaque**.
8. Buatlah visualisasi lampu merah, kuning dan hijau dengan menggunakan komponen **OvalShape1**, **OvalShape2** dan **OvalShape3**. Ubah *properties backcolour* menjadi **Red**, **Yellow** dan **Red** kemudian ubah *properties BackStyle* menjadi **Opaque**. Ratakan posisi ketiga komponen tersebut dengan mengatur **Align**, **Horizontal Spacing** dan **Vertical Spacing** pada menu **Format** yang terletak di atas pada jendela aplikasi visual basic 2010.

9. Berikan keterangan label teks di atas tampilan aplikasi yang akan menunjukkan nama dari jalan semeru dan jalan basuki rachmad sehingga pengguna lebih mudah memahami.
10. Tambahkan komponen **Panel1** dari **toolbox** dan letakan di samping kanan dari jendela desain aplikasi dimana fungsinya sebagai tempat meletakkan komponen kontrol seperti tombol Button dan lain-lain.
11. Tambahkan komponen **Button1**, **Button2** dan **Button3** dan letakkan di dalam **Panel1** dan atur properties text komponen **Button1** =**"Start"**, **Button2**=**"Stop"** dan **Button3**=**"Exit"**.
12. Tambahkan **Timer1**, dengan mendouble klik komponen Timer pada Toolbox. Timer tersebut akan otomatis berlokasi dibawah dari jendela form design yang kita buat sehingga tampak seperti gambar berikut



Gambar 5.6 Tampilan Akhir Program Aplikasi Rambu Lalu Lintas 2

13. Secara umum semua komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut.

Obyek	Properti	Pengaturan
RectangleShape1	Name	RectangleShape1
	BackStyle	Opaque
	BackColour	0; 64; 0
RectangleShape2	Name	RectangleShape2
	BackStyle	Opaque
	BackColour	0; 64; 0
RectangleShape3	Name	RectangleShape3
	BackStyle	Opaque
	BackColour	Gray
RectangleShape4	Name	RectangleShape4
	BackStyle	Opaque
	BackColour	Gray
RectangleShape5	Name	RectangleShape5
	BackStyle	Opaque
	BackColour	Gray
RectangleShape6	Name	RectangleShape6
	BackStyle	Opaque
	BackColour	Gray
OvalShape1	Name	OvalShape1
	BackColour	Red
	BackStyle	Opaque
OvalShape2	Name	OvalShape2
	BackColour	Lime
	BackStyle	Opaque

OvalShape3	Name BackColour BackStyle	OvalShape3 Yellow Opaque
OvalShape4	Name BackColour BackStyle	OvalShape4 Red Opaque
OvalShape5	Name BackColour BackStyle	OvalShape5 Yellow Opaque
OvalShape6	Name BackColour BackStyle	OvalShape6 Lime Opaque
OvalShape7	Name BackColour BackStyle	OvalShape7 Red Opaque
OvalShape8	Name BackColour BackStyle	OvalShape8 Yellow Opaque
OvalShape9	Name BackColour BackStyle	OvalShape9 Lime Opaque
OvalShape10	Name BackColour BackStyle	OvalShape10 Lime Opaque

OvalShape11	Name	OvalShape11
	BackColor	Yellow
	BackStyle	Opaque
OvalShape12	Name	OvalShape12
	BackColor	Red
	BackStyle	Opaque
Panel1	Name	Panel1
Button1	Name	Button1
	Text	"Start"
Button2	Name	Button2
	Text	"Stop"
Button3	Name	Button3
	Text	"Exit"
Label1	Name	Label1
	Text	"JL. Basuki Rachmad"
Label2	Name	Label2
	Text	"JL. Basuki Rachmad"
Label3	Name	Label3
	Text	"JL. Semeru"
Label4	Name	Label4
	Text	"JL. Semeru"
Timer1	Enabled	False
	Interval	1000
Form1	Text	"Lampu Lalu Lintas 2"
	BackColor	DarkGreen

Berikut script program secara keseluruhan :

```
Public Class Lampu_Lalu_Lintas2
    Dim t As Double

    Private Sub Lampu_Lalu_Lintas2_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        OvalShape1.BackColor = Color.DarkGray
        OvalShape2.BackColor = Color.DarkGray
        OvalShape3.BackColor = Color.DarkGray
        OvalShape4.BackColor = Color.DarkGray
        OvalShape5.BackColor = Color.DarkGray
        OvalShape6.BackColor = Color.DarkGray
        OvalShape7.BackColor = Color.DarkGray
        OvalShape8.BackColor = Color.DarkGray
        OvalShape9.BackColor = Color.DarkGray
        OvalShape10.BackColor = Color.DarkGray
        OvalShape11.BackColor = Color.DarkGray
        OvalShape12.BackColor = Color.DarkGray

        Timer1.Enabled = False
        Timer1.Interval = 1000
        t = 0

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer1.Tick

        If t >= 0 And t <= 3 Then
            OvalShapel1.BackColor = Color.Red
            OvalShape2.BackColor = Color.DarkGray
            OvalShape3.BackColor = Color.DarkGray
            OvalShape4.BackColor = Color.Red
            OvalShape5.BackColor = Color.DarkGray
            OvalShape6.BackColor = Color.DarkGray
            OvalShape7.BackColor = Color.DarkGray
            OvalShape8.BackColor = Color.DarkGray
            OvalShape9.BackColor = Color.Lime
            OvalShape10.BackColor = Color.Lime
            OvalShape11.BackColor = Color.DarkGray
            OvalShape12.BackColor = Color.DarkGray

        ElseIf t = 4 Then
            OvalShapel1.BackColor = Color.Red
            OvalShape2.BackColor = Color.DarkGray
            OvalShape3.BackColor = Color.DarkGray
            OvalShape4.BackColor = Color.Red
            OvalShape5.BackColor = Color.DarkGray
            OvalShape6.BackColor = Color.DarkGray
```

```
OvalShape7.BackColor = Color.DarkGray
OvalShape8.BackColor = Color.Yellow
OvalShape9.BackColor = Color.DarkGray
OvalShape10.BackColor = Color.DarkGray
OvalShape11.BackColor = Color.Yellow
OvalShape12.BackColor = Color.DarkGray

ElseIf t >= 5 And t <= 6 Then
    OvalShape1.BackColor = Color.DarkGray
    OvalShape2.BackColor = Color.DarkGray
    OvalShape3.BackColor = Color.Yellow
    OvalShape4.BackColor = Color.DarkGray
    OvalShape5.BackColor = Color.Yellow
    OvalShape6.BackColor = Color.DarkGray
    OvalShape7.BackColor = Color.DarkGray
    OvalShape8.BackColor = Color.Yellow
    OvalShape9.BackColor = Color.DarkGray
    OvalShape10.BackColor = Color.DarkGray
    OvalShape11.BackColor = Color.Yellow
    OvalShape12.BackColor = Color.DarkGray

ElseIf t = 7 Then
    OvalShape1.BackColor = Color.DarkGray
    OvalShape2.BackColor = Color.DarkGray
    OvalShape3.BackColor = Color.Yellow
    OvalShape4.BackColor = Color.DarkGray
    OvalShape5.BackColor = Color.Yellow
    OvalShape6.BackColor = Color.DarkGray
    OvalShape7.BackColor = Color.Red
    OvalShape8.BackColor = Color.DarkGray
    OvalShape9.BackColor = Color.DarkGray
    OvalShape10.BackColor = Color.DarkGray
    OvalShape11.BackColor = Color.DarkGray
    OvalShape12.BackColor = Color.Red

ElseIf t >= 8 And t <= 13 Then
    OvalShape1.BackColor = Color.DarkGray
    OvalShape2.BackColor = Color.Lime
    OvalShape3.BackColor = Color.DarkGray
    OvalShape4.BackColor = Color.DarkGray
    OvalShape5.BackColor = Color.DarkGray
    OvalShape6.BackColor = Color.Lime
    OvalShape7.BackColor = Color.Red
    OvalShape8.BackColor = Color.DarkGray
    OvalShape9.BackColor = Color.DarkGray
    OvalShape10.BackColor = Color.DarkGray
    OvalShape11.BackColor = Color.DarkGray
    OvalShape12.BackColor = Color.Red

ElseIf t = 14 Then
    OvalShape1.BackColor = Color.DarkGray
```

```
OvalShape2.BackColor = Color.Lime
OvalShape3.BackColor = Color.DarkGray
OvalShape4.BackColor = Color.DarkGray
OvalShape5.BackColor = Color.DarkGray
OvalShape6.BackColor = Color.Lime
OvalShape7.BackColor = Color.DarkGray
OvalShape8.BackColor = Color.Yellow
OvalShape9.BackColor = Color.DarkGray
OvalShape10.BackColor = Color.DarkGray
OvalShape11.BackColor = Color.Yellow
OvalShape12.BackColor = Color.DarkGray

ElseIf t >= 15 And t <= 16 Then
    OvalShape1.BackColor = Color.DarkGray
    OvalShape2.BackColor = Color.DarkGray
    OvalShape3.BackColor = Color.Yellow
    OvalShape4.BackColor = Color.DarkGray
    OvalShape5.BackColor = Color.Yellow
    OvalShape6.BackColor = Color.DarkGray
    OvalShape7.BackColor = Color.DarkGray
    OvalShape8.BackColor = Color.Yellow
    OvalShape9.BackColor = Color.DarkGray
    OvalShape10.BackColor = Color.DarkGray
    OvalShape11.BackColor = Color.Yellow
    OvalShape12.BackColor = Color.DarkGray

ElseIf t >= 17 And t <= 20 Then
    OvalShape1.BackColor = Color.Red
    OvalShape2.BackColor = Color.DarkGray
    OvalShape3.BackColor = Color.DarkGray
    OvalShape4.BackColor = Color.Red
    OvalShape5.BackColor = Color.DarkGray
    OvalShape6.BackColor = Color.DarkGray
    OvalShape7.BackColor = Color.DarkGray
    OvalShape8.BackColor = Color.DarkGray
    OvalShape9.BackColor = Color.Lime
    OvalShape10.BackColor = Color.Lime
    OvalShape11.BackColor = Color.DarkGray
    OvalShape12.BackColor = Color.DarkGray

End If
t = t + 1
If t = 20 Then t = 0

End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    Timer1.Start()
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button2.Click  
    Timer1.Stop()  
End Sub  
  
Private Sub Button3_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button3.Click  
    End  
End Sub  
  
End Class
```

Daftar Pustaka

Halvorson, Michael. “ *Visual Basic 2010 Step By Step* ”. Microsoft Press. Redmond, Washington, 2010.

Bradley, J.C. Millspaugh, A.C. “*Programming In Visual Basic 2010*”. McGraw-Hill press. New York, America, 2011.

<http://www.vbtutor.net/index.php/visual-basic-2010-tutorial/>

Putranto, A. “*Teknik Otomasi Industri Untuk Sekolah Menengah Kejuruan*”. Direktorat Pembinaan Sekolah Menengah dan Kejuruan. Jakarta, 2008.

