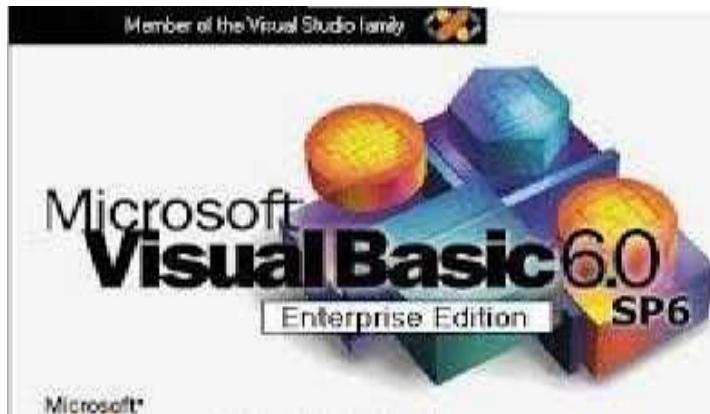


Modul

Bahasa & Pemograman I



Yahfizham, ST. M.Cs

**PROGRAM STUDI
PENDIDIKAN MATEMATIKA
FAKULTAS ILMU TARBIYAH DAN KEGURUAN
INSTITUT AGAMA ISLAM NEGERI
SUMATERA UTARA
2014**



Modul

Bahasa & Pemograman I



YAHFIZHAM, ST, M.Cs

PROGRAM STUDI
PENDIDIKAN MATEMATIKA
FAKULTAS ILMU TARBIYAH DAN KEGURUAN
INSTITUT AGAMA ISLAM NEGERI
SUMATERA UTARA
2014

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PRAKATA

Alhamdulillah Hirrobil Alamin, puja dan puji syukur kepada Allah SWT yang senantiasa memberikan kesehatan, petunjuk, rezeki dan lain sebagainya, sehingga sampai saat ini dan seterusnya pemberian tersebut dapat dirasakan oleh Penulis, Pembaca dan Kita Semua. Sholawat dan salam selalu diberikan kepada junjungan kita Nabi Besar Muhammad SAW, semoga kiranya kita dapat berkumpul bersama beliau kelak, dihari kemudian.

Sifat dinamis dari bidang teknologi, informasi, komunikasi dan sistem informasi mengharuskan para tenaga pendidik dilingkungan tinggi, baik formal dan non formal, mengetahui teknologi-teknologi terbaru maupun yang sedang berkembang untuk dapat tetap eksis dan *up-to-date*. Mahasiswa akan berhadapan dengan subjek-subjek dan objek-objek yang kompleks dan membutuhkan penjelasan yang jelas, singkat untuk dapat memahami dan menggunakan konsep-konsep didalam karir mereka.

Modul ini dibuat dan dikembangkan sebagai bahan ajar untuk mahasiswa Prodi Pendidikan Matematika, Fakultas Ilmu Tarbiyah dan Keguruan, dilingkungan IAIN Sumatera Utara pada mata kuliah *Bahasa & Pemograman I, Pemograman Berbasis Visual dan Dasar-Dasar Pemograman Visual Basic 6.0*. Disampaikan dalam bentuk yang mudah untuk dipahami, sehingga memungkinkan mahasiswa menjadi partisipan yang aktif dalam proses belajar mengajar. Disusun berdasarkan desain pembelajaran yang akan diberikan selama satu semester dengan harapan dapat lebih memudahkan proses belajar-mengajar dan mampu meningkatkan kualitas pembelajaran mahasiswa serta kelak diimplementasikan dilingkungan dunia kerja nantinya.

Terlalu jauh dari kesempurnaan akan isi, penjelasan, penyajian, tata bahasa dan hal-hal lain yang terkait, maka Penulis berharap semoga saja banyak pihak yang mau perduli dan yang akan mampu memperbaikinya dilain waktu dan kesempatan. Semoga saja Modul ini bermanfaat dan telah menjadi satu harta yang dapat ditinggalkan untuk dibaca-baca dan dijadikan rujukan atau referensi dalam penyelesaian suatu tugas dan proyek tertentu. Modul ini menjadi bahagian yang paling-paling terkecil dari lautan dunia ilmu pengetahuan yang maha tidak terbatas.

Medan, November 2014

Penulis

DAFTAR ISI

PRAKATA	i
DAFTAR ISI	ii
PENDAHULUAN	iv
TUJUAN PENULISAN MODUL	v
MANFAAT PENULISAN MODUL	vi
RUANG LINGKUP	vii
KOMPETENSI	vii
PETA KONSEP	vii
BAB I : PENGENALAN BAHASA DAN PEMOGRAMAN	1
Kegiatan Belajar 1	
1. Definisi Bahasa Pemograman	1
2. Perkembangan Pemograman	3
3. Perkembangan Pemograman Visual Basic 6.0	3
4. Spesifikasi Perangkat Keras dan Lunak	5
Rangkuman	5
Soal	6
Tindak Lanjut (<i>Feed Back</i>)	6
BAB II : MENGENAL PEMOGRAMAN VISUAL BASIC 6.0	7
Kegiatan Belajar 2	
1. Memulai Visual Basic	7
2. Antar Muka / <i>Interface</i> Visual Basic (<i>IDE</i>)	9
3. Komponen-Komponen Visual Basic	10
4. Perkakas Visual Basic	15
5. Membuka dan Menyimpan Proyek Visual Basic	19
Rangkuman	21
Latihan	21
Tindak Lanjut (<i>Feed Back</i>)	22
BAB III : MENGENAL <i>METHOD, EVENT</i> DAN <i>MESSAGE</i>	23
Kegiatan Belajar 3	
1. Pengertian <i>Property</i>	23
2. Pengertian <i>Method</i>	28
3. Pengertian <i>Event</i>	30
4. Kotak Pesan (<i>Message Box</i>)	31
Rangkuman	32
Latihan	33
Tindak Lanjut (<i>Feed Back</i>)	39

BAB IV : MENGENAL TIPE DATA VISUAL BASIC 6.0	40
Kegiatan Belajar 4	
1. Pengertian Variabel	40
2. Pengertian Konstanta	43
3. Tipe-Tipe Data	44
4. Operator-Operator	46
5. Fungsi-Fungsi Khusus Matematika	49
Rangkuman	53
Latihan dan Soal	53
Tindak Lanjut (<i>Feed Back</i>)	58
 BAB V : PERNYATAAN (<i>STATEMENT</i>) PENCABANGAN DAN PERULANGAN	59
Kegiatan Belajar 5	
1. Prosedur (<i>Procedure</i>)	59
2. Struktur Kendali (Pencabangan)	61
3. Struktur Perulangan (<i>Looping</i>)	64
Rangkuman	67
Latihan dan Soal	67
Tindak Lanjut (<i>Feed Back</i>)	75
Daftar Pustaka	76

Pendahuluan

Komputer telah dipergunakan untuk meringankan beban kerja kantor (aplikasi bisnis) semenjak sekitar tahun 1950-an, sejak saat itu sampai dengan sekarang ini, sudah lebih dari 60 tahun, usia dari perangkat tersebut. Sejak saat itu perangkat keras (*hard ware*) dan perangkat lunak (*soft ware*) mendapatkan kemajuan yang fantastis. Pesatnya perkembangan dibidang teknologi komputer telah meminimalisasi pemakaian pena dan kertas. Komputer lebih menjadikan istilah virtual semakin akrab dengan kegiatan perkantoran modern. Kegiatan tulis menulis atau catat mencatat dan membuat laporan, sekarang ini telah berbasiskan komputer. Komputer telah mengambil alih fungsi-fungsi dari alat tulis kantor yang lama, seperti mesin tik. Data dan informasi sekarang ini telah disimpan dan ditempatkan pada komputer, tidak lagi diatas kertas atau ditumpukan kertas, dilaci-laci meja dan dilemari-lemari kantor. Media tulis menulis dan penyimpanan data semakin beragam, format data, cara-cara penyimpanan data serta pengelolaan data telah mengalami banyak perubahan.

Sejarah perkembangan suatu sistem operasi dan sistem aplikasi (termasuk pemograman) tidak terlepas dari sejarah perkembangan komputer. Karena komputer merupakan suatu sistem, maka dapat dikatakan bahwa perkembangan sistem operasi dan sistem aplikasi (termasuk pemograman) berbanding lurus dengan perkembangan fisiknya (*hard ware*), tidak mungkin memisahkan *hard ware* dari *soft ware* nya. Banyaknya sistem operasi dan sistem aplikasi yang telah dibuat secara gratisan (*free ware*), coba pakai beberapa hari (*trial*), pakai secara bebas (*share ware*), pakai secara bebas dan dapat dikembangkan (*open source*) dan berbayar (*commercial*) menjadi tanda bahwa telah terjadi peningkatan yang signifikan dari *soft ware*. Kita sebagai pengguna (*user*) komputer saat ini dapat dengan mudahnya membeli atau mendownload sistem operasi dan atau sistem aplikasi. Yang perlu diingat bahwa sistem operasi dan atau sistem aplikasi (bahasa pemograman) merupakan suatu *soft ware*.

Tujuan Penulisan Modul

Sebelum kegiatan pembelajaran dilaksanakan, dosen pengampu harus menyiapkan bahan ajar yang diperlukan dalam kegiatan belajar-mengajar. Kelengkapan bahan ajar akan membantu dosen dalam proses pembelajaran, dan membantu mahasiswa dalam proses belajar. Penyusunan bahan ajar yang lengkap, yang diolah secara sistematis dapat menciptakan pembelajaran yang efektif dan efisien. Pembelajaran yang efektif dan efisien diharapkan dapat menjadi wahana untuk mencapai tujuan pembelajaran yang merupakan penjabaran dari kompetensi-kompetensi yang ingin diraih.

Pengembangan bahan ajar harus dapat menjawab dan menjadi solusi terhadap kesulitan dalam belajar. Terdapat sejumlah mata kuliah dan materi yang seringkali sulit bagi mahasiswa untuk memahaminya ataupun dosen sulit untuk menjelaskannya. Untuk mengatasi kesulitan ini maka perlu dikembangkan bahan ajar yang tepat. Banyak jenis bahan ajar cetak dan non cetak yang dapat dikembangkan, ini harus disesuaikan dengan situasi dan kondisi seperti sarana dan prasana, karakteristik mata kuliah dan mahasiswa, lingkungan sosial mahasiswa, tuntutan kurikulum dan lain sebagainya. Dengan tersedianya bahan ajar cetak yang dalam kesempatan ini berupa modul dapat mempermudah dosen dalam kegiatan belajar-mengajar dan mahasiswa dapat secara pribadi mengembangkan seluruh potensi yang dimilikinya. Adapun yang menjadi tujuan penulisan modul ini adalah sebagai berikut:

1. Mengenalkan konsep dasar bahasa pemrograman
2. Memberikan pemahaman terhadap pemrograman berbasis visual
3. Memberikan bekal kepada mahasiswa mengenai pemrograman *Visual Basic 6.0*
4. Memberikan gambaran aplikatif mengenai pemrograman *Visual Basic 6.0*
5. Memberikan gambaran konkrit proses perancangan program pada *Visual Basic 6.0*
6. Mengembangkan program matematika sederhana (pengembangan skill) memanfaatkan pemrograman *Visual Basic 6.0*.

Manfaat Penulisan Modul

Ada beberapa manfaat dari kepemilikan dan atau penyusunan penulisan modul untuk mata kuliah Bahasa dan Pemograman I bagi dosen yang mengasuh, Ketua Prodi, dan mahasiswa dapat dijelaskan sebagaimana berikut ini:

1. Bagi Mahasiswa
 - a) Sebagai bahan dan sumber belajar pada semester v di Prodi Pendidikan Matematika Fakultas Ilmu Tarbiyah dan Keguruan IAIN Sumatera Utara Medan.
 - b) Menjadi pedoman wajib untuk dapat mengikuti kegiatan perkuliahan dikelas
 - c) Menjadi buku ajar dan buku ilmiah pendukung atau tambahan pendalaman menguasai pemograman Visual Basic 6.0
2. Bagi Dosen
 - a) Menjadi modul wajib bagi dosen yang tidak memiliki bahan ajar cetak untuk mata kuliah Bahasa dan Pemograman I
 - b) Menjadi bahan rujukan untuk mengembangkan dan mengajarkan mata kuliah Bahasa dan Pemograman I
3. Bagi Ketua Prodi
 - a) Adanya modul menjadi alat ukur dan pertimbangan kebijakan atau pendukung keputusan atas mata kuliah Bahasa dan Pemograman I, untuk tetap mempertahankan atau mengganti mata kuliah atau nama mata kuliah pada kurikulum selanjutnya pada tingkat Prodi PMM FITK IAIN SU Medan.
 - b) Dengan adanya modul tersebut dapat berkontribusi pada peningkatan kualitas Dosen yang mengampu mata kuliah Bahasa dan Pemograman I dan sekaligus mutu mata kuliah itu sendiri.

Ruang Lingkup

Penulisan modul ini dikembangkan dan dibuat secara khusus untuk mata kuliah Bahasa dan Pemograman I yang belum ada pada semester v di Prodi PMM FITK IAIN SU Medan. Ruang lingkup bahasan atau materi cakupan terdiri dari:

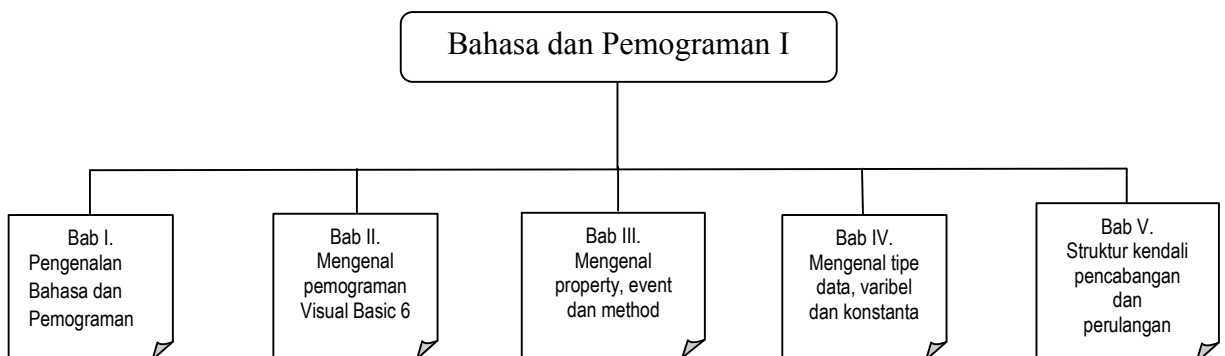
1. Mengetahui Teori/Konsep Bahasa Pemograman
2. Mengetahui *Visual Basic 6.0*
3. Mengetahui *Property, Method* dan *Event*
4. Mengetahui Tipe Data *Visual Basic 6.0*
5. Pernyataan (*Statement*) Kendali Pencabangan dan Perulangan (*Looping*)

Kompetensi

Kompetensi Inti : Mampu merancang bangun program matematika sederhana (pengembangan skill) memanfaatkan pemograman *Visual Basic 6.0*.

Kompetensi Dasar : a) Memahami konsep dasar bahasa dan pemograman
 b) Mengetahui antarmuka *Visual Basic 6.0*
 c) Mampu membedakan *property, event* dan *method*
 d) Mampu membedakan tipe data, variabel dan konstanta
 e) Mampu bekerja dengan struktur kendali dan perulangan

Peta Konsep





Kegiatan Belajar 1.

BAB I

PENGENALAN BAHASA DAN PEMOGRAMAN

1.1. Definisi Bahasa Pemograman

Bahasa pemograman adalah suatu cara atau teknik berkomunikasi atau interaksi antara manusia sebagai makhluk hidup dengan mesin sebagai benda yang dalam hal ini adalah komputer sebagai suatu perangkat peralatan. Bahasa pemograman berisi kumpulan perintah atau instruksi yang diketikkan oleh manusia menggunakan perantara suatu perangkat keras berupa papan ketik (*keyboard*). Bahasa pemograman juga akan menentukan bagaimana data akan disimpan, dikirim dan mendapatkan kembali data tersebut serta apa yang akan dilakukan selanjutnya jika terdapat kondisi yang beragam seperti misalnya dalam memberikan instruksi prioritas terhadap peralatan I/O (*input/output*), multimedia dan lain-lain.

Kumpulan instruksi tersebut harus dapat dimengerti oleh komputer, karenanya maka bahasa yang diketikkan tersebut mempunyai urutan langkah-langkah yang jelas, logis, sistematis, aturan-aturan khusus (tertentu), berstruktur tertentu dan mempunyai elemen-elemen tertentu (*syntaksis*). Rancangan awal bahasa pemograman ini lahir dari bahasa alami (*natural language*) yang biasa dipakai oleh manusia dalam berinteraksi dan berkomunikasi dengan manusia lain. Bahasa pemograman menyediakan kebutuhan seorang programmer untuk dapat secara spesifik menyatakan atau memberikan instruksi apa yang akan dilakukan oleh komputer.

Komputer hanya memahami satu bahasa, yaitu bahasa mesin, bahasa mesin terdiri dari nilai 0 dan 1 yang disebut dengan digital. Untuk menterjemahkan bahasa pemograman yang dituliskan oleh manusia dibutuhkan suatu teknik khusus yang disebut *compiler* dan *interpreter*. **Compiler** adalah unit khusus pada masing-masing bahasa pemograman yang menterjemahkan atau mengkonversikan kode-kode program atau instruksi atau *source code* yang dituliskan menjadi nilai 0 dan 1. **Interpreter** merupakan suatu teknik dalam menganalisa dan mengeksekusi baris per baris suatu set instruksi yang diberikan atau yang dituliskan oleh programmer dalam suatu bahasa pemograman tertentu. Berikut ini akan diberikan perbedaan antara *compiler* dengan *interpreter*.

<i>Compiler</i>	<i>Interpreter</i>
Mengkonversikan kode program atau instruksi (<i>source code</i>) secara keseluruhan	Mengkonversikan instruksi per instruksi
Jika ada instruksi yang salah maka kode program harus diperbaiki dan di kompilasi ulang	Ketika memberikan instruksi terdapat kesalahan, maka kesalahan dapat langsung diperbaiki.
Menghasilkan suatu objek	Tidak menghasilkan suatu objek
Menghasilkan ekstensi <i>Executable</i> (.exe)	Tidak menghasilkan ekstensi <i>Executable</i> (.exe)
Pengecekan <i>source code</i> lebih cepat	Pengecekan <i>source code</i> lebih lambat
Setelah <i>source code</i> dikompilasi dan menghasilkan ekstensi <i>Executable</i> (.exe) maka keamanan <i>source code</i> dapat terjamin.	<i>Source code</i> dapat terus menerus dipergunakan dan diperbaiki, terkadang dianggap sebagai virus dan keamanan <i>source code</i> tidak terjamin.

Tabel 1.1. Perbedaan *compiler* dan *interpreter*

Belajar memprogram tidak sama dengan belajar bahasa pemrograman. Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu yang mudah dibaca dan dipahami. Sedangkan belajar bahasa pemrograman berarti belajar memakai suatu bahasa, aturan-aturan tata bahasanya, instruksi-instruksinya, tata cara pengoperasian *compiler*-nya, dan memanfaatkan instruksi-instruksi tersebut untuk membuat program yang ditulis hanya dalam bahasa itu saja. Belajar memprogram adalah belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah kemudian menuliskannya dalam notasi yang disepakati bersama. Belajar memprogram merupakan bersifat pemahaman persoalan, analisis dan sintesis. Belajar memprogram, menitik beratkan persoalan bagaimana cara-cara atau metode-metode merancang sebuah program. Belajar bahasa pemrograman sama dengan belajar memakai suatu bahasa pemrograman, aturan sintaks, tatacara untuk memanfaatkan instruksi yang spesifik untuk setiap bahasa dan membuat pengkodean

Pada pemrograman prosedural, program dibedakan antara bagian data dengan bagian instruksi. Bagian instruksi terdiri atas runtutan (*sequence*) instruksi yang dilaksanakan satu per satu secara berurutan oleh pemroses. Alur pelaksanaan instruksi dapat berubah karena adanya pencabangan kondisional. Data yang disimpan di dalam memori dimanipulasi oleh instruksi secara beruntun atau prosedural. Paradigma pemrograman seperti ini dinamakan pemrograman prosedural. Selain paradigma pemrograman prosedural, ada lagi paradigma yang lain yaitu pemrograman berorientasi objek (*object oriented programming*).

1.2. Perkembangan Pemograman

Sejarah perkembangan *soft ware* komputer berbanding lurus atau sejalan dengan perkembangan komputer itu sendiri/*hard ware*. Sistem operasi penting karena sistem ini adalah program yang menjalankan dan atau mengelola seluruh sumber daya dari sistem komputer. Sistem operasi juga diibaratkan sebagai seorang menejer yang berada disuatu perusahaan. Sejarah perkembangan suatu sistem operasi tidak terlepas dari sejarah perkembangan komputer secara umum. Evolusi pemograman berbasis DOS dan *Windows*:

Generasi I (1940) : Bahasa mesin

Generasi II (1950) : Bahasa Rakitan

Generasi III (1960) : Bahasa Tingkat Tinggi

Generasi IV (1970) : Bahasa *Query and Database*

Generasi V(1980) : Bahasa Kecerdasan Buatan

1.3. Perkembangan Pemograman Visual Basic 6.0

Bahasa Visual Basic adalah satu diantara sekian bahasa pemrograman yang cukup banyak digunakan oleh para programmer. Visual Basic adalah bahasa pemrograman visual dengan dasar pemrograman bahasa Basic. Bahasa BASIC (*Beginner's All-purpose Symbolic Instruction Code*) dikembangkan pertama kali pada awal 1950-an. Sementara bahasa Visual Basic, yang merupakan pengembangan bahasa basic, dikembangkan pertama kali oleh Microsoft pada tahun 1991. Visual Basic ini merupakan *Development Tool* yaitu alat bantu yang dapat digunakan untuk membuat berbagai macam program komputer, khususnya yang berbasis *Windows*. Pembuatan program dalam bahasa *Visual Basic* berbeda dengan pembuatan program-program DOS umumnya atau yang sering disebut sebagai pemrograman konvensional. Namun jika dilihat dari visualnya, *Visual Basic* sedikit mirip dengan delphi. Hanya saja bahasa *Visual Basic* merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming = OOP*).

Visual Basic 6.0 ialah bahasa pemrograman *event-driven* yang berasal dari **BASIC**. *Event driven* artinya program menunggu sampai adanya respons dari pemakai berupa kejadian tertentu, misalnya tombol diklik atau menu dipilih. Ketika event terdeteksi, event yang berhubungan akan melakukan aksi sesuai dengan kode yang diberikan. Ada tiga edisi yang dikeluarkan *Microsoft*, yaitu:

a. Standard Edition

Standard Edition sangat merekomendasikan bagi pemula yang ingin mempelajari Visual Basic 6.0 dan mempunyai fasilitas sebagai berikut:

- Kemampuan aplikasi 32 bit yang berjalan di Microsoft Win 9x dan Win NT untuk pemula.
- Terdiri dari control seperti grid, tab, dan Data Bound.
- Memuat *Learn Visual Basic Now* dan *Online Help*.
- *Microsoft Developer Network* CD berisi dokumentasi.

b. Professional Edition

Professional Edition umumnya digunakan oleh para professional yang sudah cukup mendalami Visual Basic 6.0. Tidak terlalu banyak perbedaan dengan Standard Edition, hanya ada beberapa tambahan, diantaranya:

- *ActiveX Control*, termasuk *Internet Control*
- IIS (*Internet Information Server*)
- *Dynamic Hyper Text Markup Language (DHTML) Page Designer*

c. Enterprise Edition

Lebih ditekankan untuk membuat aplikasi yang bersifat *server based*, tapi program-program aplikasi standard dapat berjalan dengan baik jika menggunakan versi ini. Fasilitas tambahan antara lain:

- *Application Performance Explorer*
- IIS (*Internet Information Server*)
- *Support for Microsoft Transaction Server 2.0*
- *SQL Debugging*
- *Visual Component Manager*
- *Visual Database Tool*

Kekurangan *Visual Basic* 6.0 sampai saat ini hanya dapat berjalan diatas lingkungan sistem operasi *Windows*. Untuk kalangan sistem operasi yang lain seperti *Linux* misalnya, *Visual Basic* masih belum dapat berjalan optimal walaupun saat ini sudah mulai dirintis sebuah *framework* berbasis .NET agar dapat menjalankan aplikasi VB.NET diatas *platform Linux*.

1.4. Spesifikasi Perangkat

Ada persyaratan tertentu yang harus disiapkan sebelum memulai pemrograman dan mengenal antar muka (*user interface*) dari *visual basic 6.0*. Persyaratan ini hanya sebagai suatu pengetahuan awal bahwa untuk memulai kegiatan pemrograman dibutuhkan segala sesuatu misalnya perkakas yang nantinya akan dipakai. Spesifikasi perangkat standar sebaiknya perlu diketahui agar kegiatan pemrograman dapat berjalan dengan baik. Perangkat yang dimaksud adalah perangkat keras (*hard ware*) dan perangkat lunak (*soft ware*). Minimal spesifikasi perangkat keras yang dibutuhkan adalah sebagai berikut :

1. *Prosesor Intel Pentium IV* dan yang sederajat
2. *Motherboard onboard*
3. *Memory 512 MB*
4. *Hard Disk Drive 80 GB*
5. *Graphic card 32 bit* dan *Sound card*
6. *Keyboard* dan *Mouse*

Minimal spesifikasi perangkat lunak (*soft ware*) yang dibutuhkan adalah sebagai berikut :

1. *Sistem Operasi Windows XP*
2. *DirectX Version 9*
3. *Visual Basic 6.0 Standart Edition*

Rangkuman

Pemrograman adalah suatu cara atau teknik berkomunikasi atau interaksi antara manusia sebagai makhluk hidup dengan mesin sebagai benda yang dalam hal ini adalah komputer sebagai suatu perangkat peralatan. Bahasa pemrograman berisi kumpulan perintah atau instruksi yang diketikkan oleh manusia menggunakan perantara suatu perangkat keras berupa papan ketik (*keyboard*). Belajar memprogram tidak sama dengan belajar bahasa pemrograman. Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu seperti sintak bahasanya dan cara *compiler*-nya. *Visual Basic 6.0* ialah bahasa pemrograman *event-driven* yang berasal dari *BASIC*, *event driven* artinya program menunggu sampai adanya respons dari pemakai berupa kejadian tertentu, misalnya tombol diklik atau menu dipilih. Tombol diklik dan menu dipilih ini sama dengan konsep *Object Oriented Programming (OOP)*.

SOAL

1. Jelaskan sebanyak dua paragraph apakah yang dimaksud dengan pemograman !
2. Bedakan belajar bahasa pemograman dengan memprogram !
3. Apakah yang harus dipersiapkan sebelum memulai pemograman !

Tindak Lanjut (*Feed Back*)

Skor	Kriteria
100-80	Mampu dan dapat menjelaskan secara baik dan benar mengenai : <ol style="list-style-type: none">1. Pengertian bahasa pemograman2. Membedakan antara belajar bahasa pemograman dan memprogram3. Menyebutkan atau menuliskan hal-hal yang harus dipersiapkan sebelum memprogram
70-60	Hanya mampu menjawab satu sampai dua soal saja (tidak tuntas) dan harus mengulang bahagian tertentu saja
50 ke bawah	Tidak mampu sama sekali dan harus mengulang Bab 1



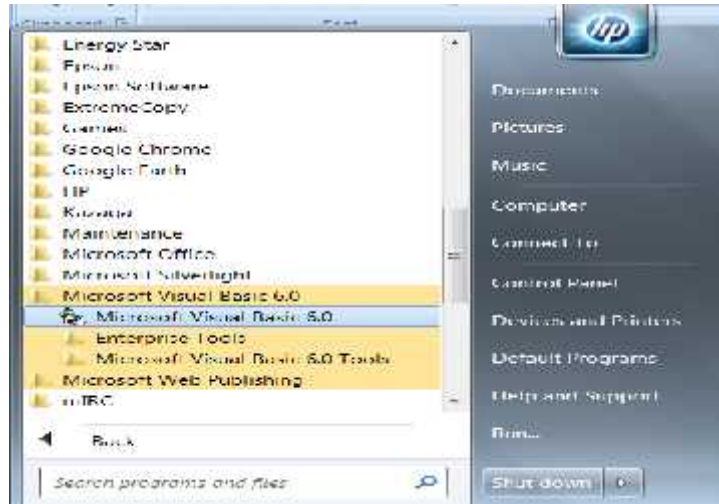
Kegiatan Belajar 2.

**BAB
II****MENGENAL
PEMOGRAMAN VISUAL BASIC 6.0****2.1. Memulai *Visual Basic 6.0***

Visual Basic pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemrograman di dalam bahasa Basic dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program. Hal ini lebih mudah lagi setelah hadirnya *Microsoft Visual Basic*, yang dibangun dari ide untuk membuat bahasa yang sederhana dan mudah dalam pembuatan scriptnya (*simple scripting language*) untuk *graphic user interface* yang dikembangkan dalam sistem operasi *Microsoft Windows*. *Visual Basic* merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman visual yang memungkinkan penggunanya untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam visual basic adalah *FORM*, dimana pengguna dapat mengatur tampilan form kemudian dijalankan dalam script yang sangat mudah. Dengan komponen ini memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows. Hal ini juga ditunjang dengan teknik pemrograman di dalam Visual Basic yang mengadopsi dua macam jenis pemrograman yaitu Pemrograman Visual dan *Object Oriented Programming* (OOP).

Visual Basic 6.0 sebenarnya perkembangan dari versi sebelumnya dengan beberapa penambahan komponen yang sedang tren saat ini, seperti kemampuan pemrograman internet dengan DHTML (*Dynamic HyperText Mark Language*), dan beberapa penambahan fitur database dan multimedia yang semakin baik. Sampai saat buku ini ditulis bisa dikatakan bahwa Visual Basic 6.0 masih merupakan pilih pertama di dalam membuat program aplikasi yang ada di pasar perangkat lunak nasional. Hal ini disebabkan oleh kemudahan dalam melakukan proses *development* dari aplikasi yang dibuat. Pemograman visual merupakan dimensi baru dalam pembuatan aplikasi karena dapat langsung menggambarkan objek-objek ke layar sebelum dieksekusi. Dalam lingkungan pengembangan visual, sekarang objek yang anda buat hasilnya langsung tampil di layar. Objek yang dibuat itu akan sama hasilnya pada saat program dijalankan, dengan demikian tidak perlu lagi melakukan perubahan kode program secara manual.

Untuk membuka dan memulai program Visual Basic 6.0 (yang dikemas dalam Microsoft *Visual Studio* 6.0), langkah-langkahnya adalah sebagai berikut: Klik **Start – All Programs (Programs) - Microsoft Visual Basic 6.0 – Microsoft Visual Basic 6.0**. Terlihat pada layar awal akan muncul tampilan seperti pada gambar 1.1 dibawah ini



Gambar 2.1. Memulai pemrograman visual basic

Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di (*Toolbox*), dan setiap komponen yang dipakai harus diatur propertinya lewat jendela (*Property*). Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project.

Di samping itu terdapat fasilitas-fasilitas pemakaian visual basic pada menu. Untuk lebih jelasnya Visual Basic menyediakan bantuan yang sangat lengkap dan detail dalam MSDN. Pada pemrograman Visual, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari objek-objek yang digunakan dalam user interface, dan baru dilakukan *penulisan kode program* untuk menangani kejadian-kejadian (*event*). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*. Langkah awal dari belajar Visual Basic adalah mengenal IDE (*Integrated Development Environment*). Dengan menggunakan IDE *programmer* dapat membuat user interface, melakukan koding, melakukan *testing* dan *debuging* serta mengkompilasi program menjadi *executable*.

2.2. Antarmuka (*Interface*) *Visual Basic*

Tampilan antarmuka (*interface*) *visual basic* untuk *Integrated Development Environment* (IDE) adalah bidang atau area atau lembar kerja tempat kita bekerja untuk membuat dan menghasilkan program aplikasi. Setelah memulai mengklik Microsoft Visual Basic 6.0 seperti yang diperlihatkan pada gambar 2.1 diatas, pada keadaan standar ketika menjalankan visual basic, akan muncul tampilan seperti terlihat pada gambar 2.2 dibawah ini

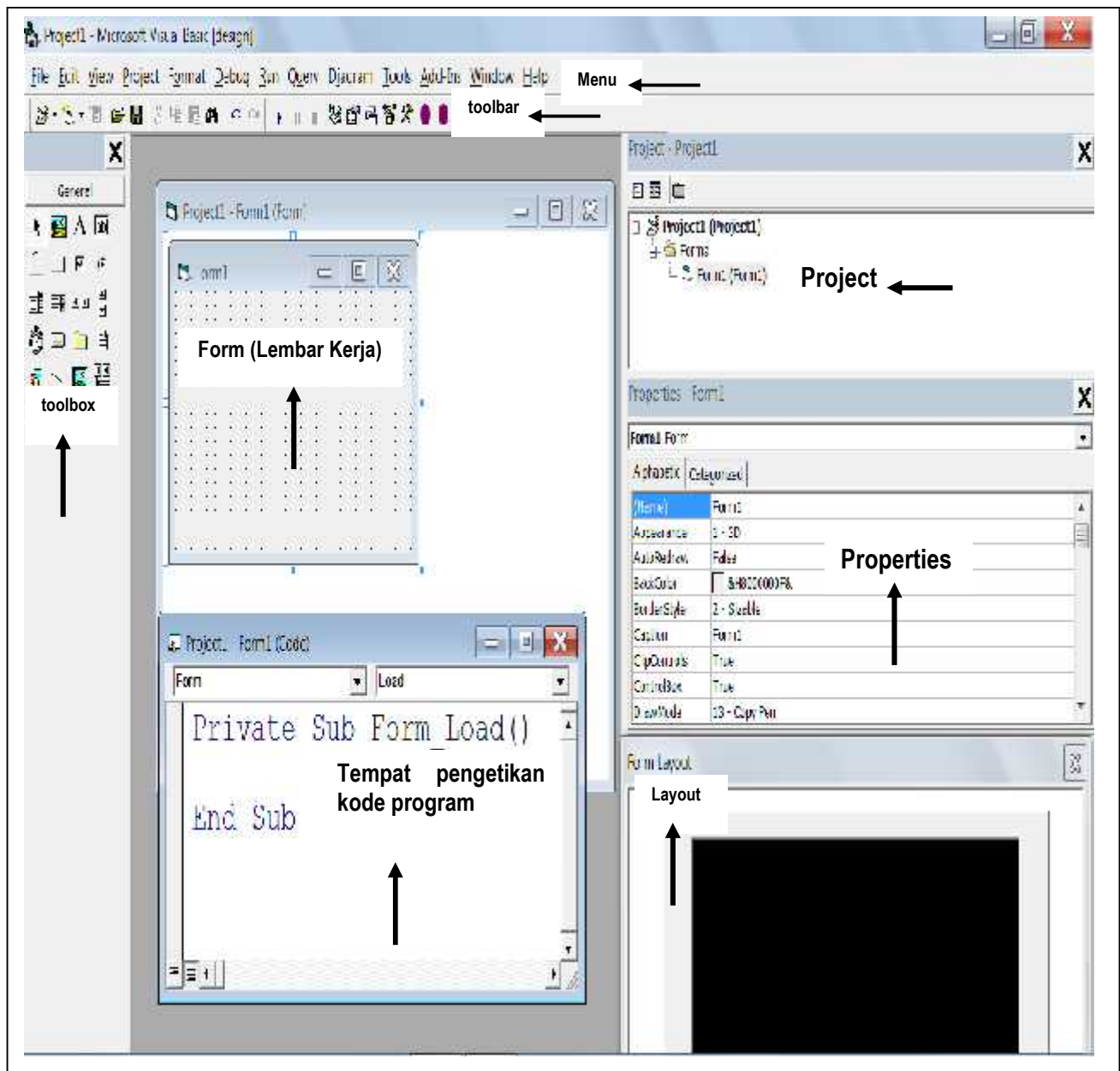


Gambar 2.2. Awal tampilan menjalankan IDE Visual Basic 6.0

Visual basic menyediakan banyak jenis project aplikasi. Beberapa pilihan yang terdapat pada kotak dialog New Project adalah sebagai berikut :

- Standard EXE : membuat aplikasi Visual Basic Standar
 - Active EXE : membuat aplikasi ActiveX
 - Active DLL : membuat library ActiveX
 - ActiveX Control : membuat kontrol ActiveX
- Versi Enterprise
- VB Application Wizard : membuat aplikasi dengan bantuan Wizard
 - VB Wizard Manager : pusat pengelolaan Wizard Visual Basic
 - IIS Application : membuat aplikasi IIS (*Internet Information Server*)
 - DHTML Application (*Dynamic Hypertext Mark-up Language*) untuk internet.

Nah, setiap kali memulai bekerja dengan visual basic 6.0 mulailah dengan pilihan **Standard EXE**, kemudian klik pada tombol **Open**. Lembar kerja dan area antarmuka visual basic ini menggunakan model MDI (*Multiple Document Interface*), yaitu satu jendela layar akan terlihat banyak bagian-bagian jendela yang lainnya. Gambar lembar kerja dan jendela visual basic terlihat seperti tampilan gambar dibawah ini



Gambar 2.3. Jendela lembar kerja visual basic

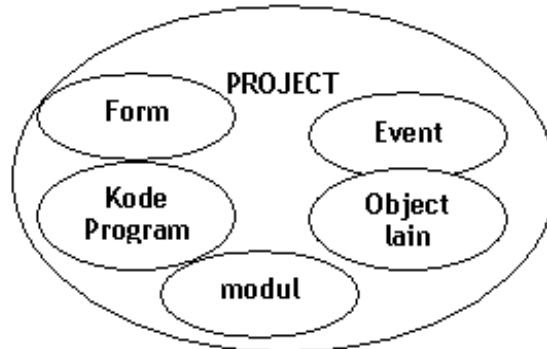
2.3. Komponen-komponen *Visual Basic*

Jendela IDE visual basic memiliki sejumlah menu, toolbar, dan window seperti yang terlihat pada gambar 2.3. Berikut ini akan dijelaskan kegunaan masing-masing komponen :

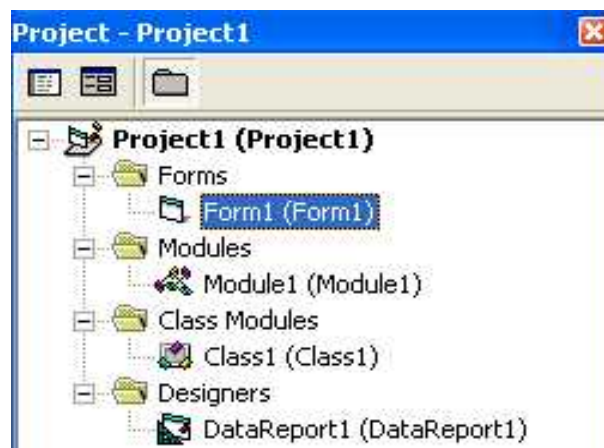
a. *Project Window*

Jika membuat program aplikasi, akan terdapat jendela project yang berisi semua file yang dibutuhkan untuk menjalankan program aplikasi Visual Basic yang dibuat. Pada jendela project terdapat tiga icon yaitu icon **View Code** untuk menampilkan jendela editor, **View Object** untuk menampilkan bentuk formulir (Form) dan **Toggle Folders** digunakan untuk menampilkan folder (tempat penyimpanan file). Pertama kali

ketika menggunakan program *visual basic* maka komponen *project* yang akan diload, selanjutnya adalah menambah form-form atau membuat modul atau mungkin membuat kode program. Secara lebih sederhana posisi *project* dalam setiap komponen yang ada adalah seperti terlihat pada gambar dibawah ini :



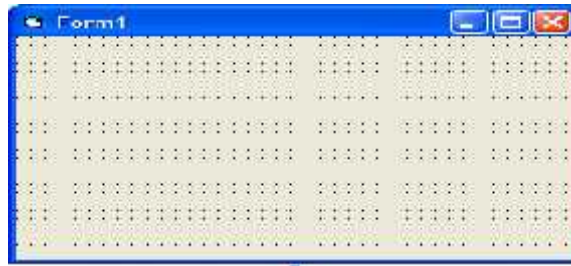
Gambar 2.4. Project pada visual basic



Gambar 2.5. Project dengan komponen lainnya

b. Form Designer

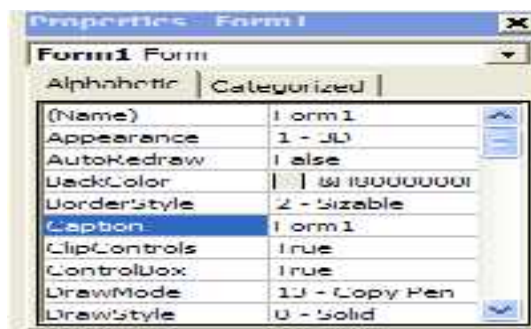
Form digunakan ketika akan meletakkan object-object apa saja yang akan digunakan dalam program, object-object yang terdapat dalam toolbox, diletakkan dan didesain dalam bagian form. Form sebenarnya adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. Secara otomatis akan tersedia form yang baru jika kita membuat program aplikasi baru, yaitu dengan nama ***Form1***. Umumnya dalam suatu form terdapat garis titik-titik yang disebut dengan ***Grid***.



Gambar 2.6. Form atau lembar kerja kosong

c. *Properties Window*

Properties digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela **Properties** atau lewat kode program.



Gambar 2.7. Property untuk Form1

d. *Code Window*

Digunakan untuk menulis kode program yang menentukan tingkah laku dari form dan objek-objek yang ada pada aplikasi bersangkutan. Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu objek dijalankan. Kode program ini akan mengontrol dan menentukan jalannya suatu objek.

Gambar 2.8. jendela mengetikkan perintah (*coding*) pada visual basic

e. Module

Module dapat disejajarkan dengan form, tetapi tidak mengandung object. Module dapat berisi kode-kode program atau procedure yang dapat digunakan dalam program aplikasi. Macam-macam modul yang terdapat dalam Visual Basic, adalah

- *Modul Form* (File berekstensi .Frm)

Berisi deskripsi secara grafis tampilan serta kontrol yang digunakan dalam suatu proyek. Termasuk di dalamnya setting properti masing-masing kontrol serta rutin untuk masing-masing obyek yang terdapat pada form tersebut.

- *Modul Class* (File berekstensi .Cls)

Berfungsi sama dengan modul form tetapi tidak terlihat dalam bentuk tampilan. Melalui modul ini dapat membuat sendiri obyek yang ingin ditampilkan. Masing-masing obyek harus memiliki properti dan metode tersendiri.

- *Modul Standar* (File berekstensi .Bas)

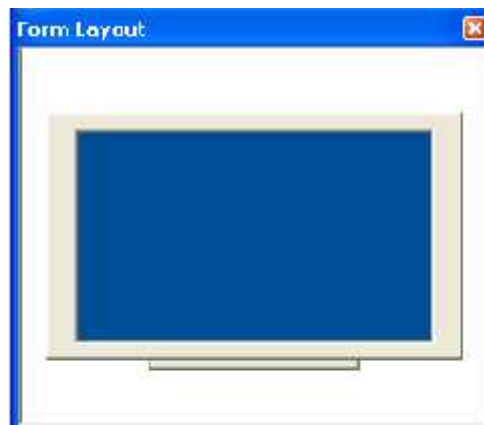
Modul ini hanya berisi rutin program, yang berupa deklarasi tipe, konstanta, variabel, prosedur dan fungsi eksternal serta prosedur public. Umumnya file ini berisi logika yang kompleks sehingga perlu dipisahkan dari modul form agar penulisannya tidak terlalu rumit.

f. Color Pallete Window

Digunakan untuk menentukan warna dari suatu objek seperti warna dari objek yang sedang aktif saat ini.

g. Form Layout Window

Menunjukkan bagaimana peletakan sebuah form akan ditampilkan pada saat dijalankan.



Gambar 2.9. Bentuk form layout window

h. Immediate Window

Digunakan untuk memasukkan ekspresi untuk melihat hasilnya dengan menggunakan perintah “print” atau “?”. Jendela ini biasanya digunakan bersama watch window pada saat sebuah program di-debug. Short-cut untuk jendela ini adalah Ctrl-G.

i. Object Browser

Digunakan untuk menyelusuri external libraries sehingga anda dapat mempelajari objek-objek dan properti, kejadian (events), dan method yang dimilikinya. *Short-cut* untuk jendela ini adalah F2.

j. Locals Window

Akan aktif hanya pada saat program di jalankan, berisi nilai dari sejumlah variabel yang bersifat lokal pada sebuah prosedur atau module.

k. Watch Window

Digunakan untuk memonitor nilai dari suatu varibel baik bersifat lokal maupun global.

l. Call Stack Window

Hanya ditampilkan jika suatu program yang dijalankan dihentikan untuk sementara (break) dan menekan Ctrl-L. Menunjukkan semua prosedur yang menunggu prosedur aktif telah selesai. Jendela ini penting untuk proses debugging untuk mengetahui jalur eksekusi program hingga sampai situasi sekarang.

m. Menu secara garis besar terdiri dari :

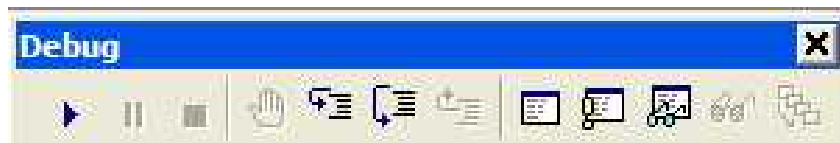
- ***Menu File*** berisi perintah antara lain, *open* untuk membuka suatu project dan *save* untuk menyimpan project atau form, *make* digunakan untuk mengkompile project menjadi ***Executable file***.
- ***Menu Edit*** berkaitan dengan perintah-perintah editing seperti *cut*, *copy*, *paste*, *find*, *replace*, *undo* dan *redo*.
- ***Menu View*** digunakan untuk menampilkan window-window pada IDE visual basic seperti yang telah dijelaskan diatas.
- ***Menu Project*** digunakan untuk menambahkan objek-objek baru seperti pada form, standart module (bas), class module, User control module, dll.
- ***Menu Format*** digunakan untuk mengatur posisi dan ukuran satu atau beberapa kontrol.
- ***Menu Run*** berisi perintah-perintah untuk menjalankan aplikasi yang dibangun.

- n. **Toolbar** berisi tool yang digunakan untuk perintah-perintah seperti membuka atau menyimpan sebuah project.



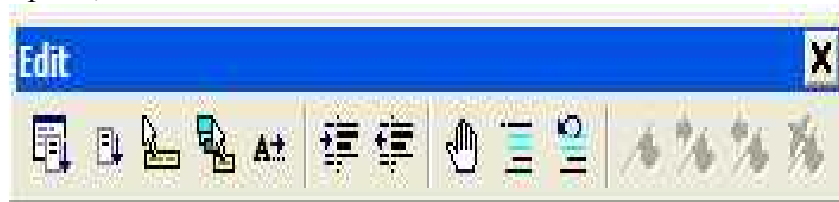
Gambar 2.10. Standar toolbar

- o. **Debug Toolbar** digunakan berisi perintah seperti yang terdapat pada menu debug, jika toolbar tersebut tidak terdapat pada IDE, anda dapat memilih menu view - toolbars untuk menampilkan toolbar tersebut.



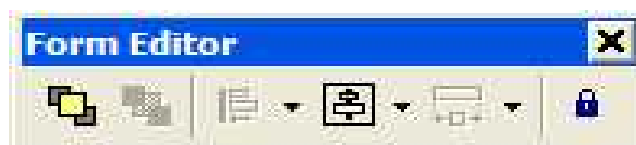
Gambar 2.11 Debug Toolbar

- p. **Edit Toolbar** berfungsi pada saat kita mengedit sebuah kode, menambahkan break point, dan bookmarks.



Gambar 2.12 Edit Toolbar

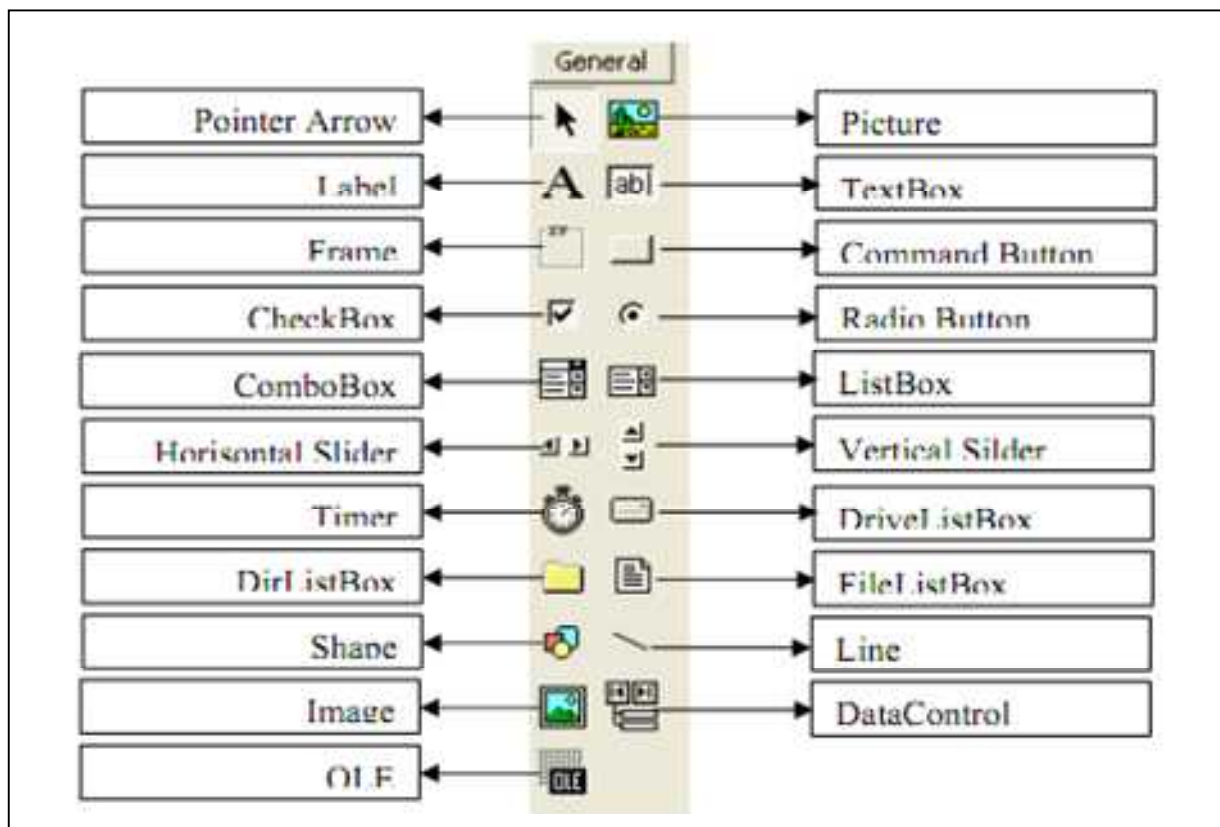
- q. **Form Editor Toolbar** digunakan untuk mengatur posisi control yang ada pada form.



Gambar 2.13 Form Editor toolbar

2.4. Perkakas Visual Basic (*Toolbox*)

Toolbox adalah kotak alat yang berisi icon-icon untuk memasukkan objek tertentu ke dalam jendela form. Objek-objek standar atau perkakas yang tampil pada jendela visual basic seperti tampilan gambar dibawah ini







Gambar 2.14. Komponen standar dalam toolbox

- **Pointer** bukan sebuah kontrol, pointer digunakan untuk memilih control yang sudah ada pada sebuah form.
- **Picture Box Control**, digunakan untuk menampilkan gambar seperti BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, and JPEG.
- **Label Control**, digunakan sebagai text static yang tidak dapat diedit, biasanya digunakan untuk memberi nama pada kontrol yang lain seperti *textbox*.
- **TextBox Control**, berisi string dari karakter yang dapat diisi oleh user.
- **Frame Control**, digunakan untuk mengelompokkan control-control yang lain.
- **Command Button Control**, dipakai hampir pada setiap form, biasanya digunakan untuk *event mouse click*.
- **Check Box Control**, digunakan untuk pilihan ya/tidak atau benar/salah.
- **Option Button Control**, digunakan dalam suatu group, dimana seorang user dapat memilih salah satu saja dalam suatu group.








- **ListBox Control**, berisi sejumlah item dimana seorang user dapat memilih salah satu atau beberapa item.
- **ComboBox Control**, merupakan gabungan antara TextBox dan ListBox Control, dimana daftar item hanya akan ditampilkan jika user men-click pada panah kebawah disebelah kanan Control tersebut.
- **HscrollBar** dan **VscrollBar Control**, digunakan untuk ScrollBar.
- **Timer Control**, merupakan kontrol yang tidak terlihat pada saat program dijalankan.
- **DriveList Box, DirList Box, dan FileList Box Control**, ketiganya digunakan untuk membuat suatu dialog untuk sistem file.
- **Shape** dan **Line**, digunakan untuk memperindah tampilan dari suatu form.
- **Image Control**, mirip dengan *Picture Box Control* tetapi digunakan sebagai *container* kontrol-kontrol yang lain.
- **Data Control**, merupakan kunci dari data binding pada visual basic. Dengan menggunakan kontrol ini, kita dapat menghubungkan sebuah database pada visual basic.
- **OLE Control**, digunakan sebagai tempat untuk program eksternal yang ada pada *windows* seperti *spread sheet* yang ada pada *Microsoft Excel*. Dengan menggunakan kontrol tersebut kita dapat menampilkan program lain pada sebuah aplikasi.

Tambahan penjelasan dan fungsi dari masing-masing kontrol yang ada pada **toolbox visual basic 6.0**.


Tabel 2.1 Penjelasan *toolbox visual basic*

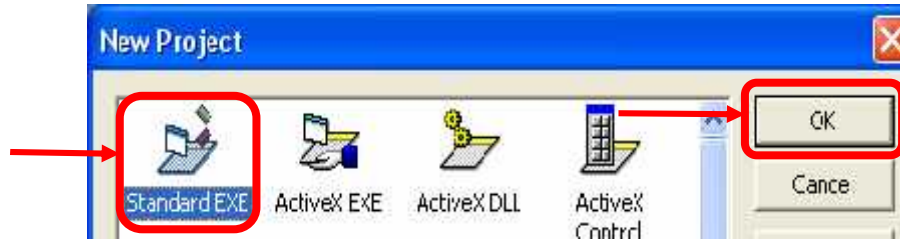
Bentuk	Nama Kontrol	Fungsi
	Pointer	Pointer sebagai penunjuk kontrol yang berfungsi untuk memindahkan atau mengubah ukuran objek.
	PictureBox	Untuk menampilkan file gambar (<i>bitmaps, icon, gif, jpeg</i> dsb)
	Label	Untuk menampilkan teks, dan menerima <i>event</i> dari objek lain
	TextBox	Untuk menerima masukan dari user. Penulisanya : txtangka, txt_bil dan lain-lain

	Frame	Untuk mengelompokkan beberapa kontrol (Group) pada suatu form. Misal : frm_Style dan lain-lain
	Command Button	Untuk membuat tombol pelaksana suatu perintah atau tindakan ketika digunakan. Misalnya : cmd_proses dan lain-lain
	CheckBox	Untuk membuat kotak check yang dapat memilih satu atau banyak pilihan yang ada. Misal : chk_makanan favorit, dll
	Option Button	Untuk memilih dan mengaktifkan hanya satu pilihan dari banyak pilihan yang ada. Misal : opt_Agama dan lain-lain
	ComboBox	Sebagai tempat mengetikkan pilihan atau memilih suatu pilihan lewat Drop Down-List. Misal : cmb_ukuran, dll
	ListBox	Untuk menampilkan daftar pilihan yang dapat digulung secara horisontal maupun vertikal. Misal : lst_jenis
	HScrollBar	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi horisontal. Misal : hsb_warna
	VScrollBar	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi vertikal. Misal : vsb_warna
	Timer	Untuk mengoperasikan waktu kejadian pada rutin program dalam interval yang ditentukan. Misalnya : Time_waktu
	DriveListBox	Untuk menampilkan daftar drive komputer yang aktif dan dapat dipilih sebuah drive.

	DirListBox	Untuk menampilkan daftar directory dan path pada drive kerja terpilih.
	FileListBox	Untuk menampilkan daftar file pada directory dan drive yang aktif. Penulisannya seperti: <code>fil_gambar</code>
	Shape	Untuk membentuk objek dua dimensi seperti <i>square</i> , <i>oval</i> , <i>ellips</i> dan lain-lain. Penulisannya seperti: <code>shape_lingkaran</code>
	Line	Untuk menggambar garis lurus dengan banyak variasi dengan ketebalan yang dapat diatur. Penulisannya seperti : <code>line_satu</code> dan lain-lain
	Image	Untuk menampilkan gambar <i>icon</i> , <i>bitmap</i> atau metafile pada form. Penulisannya seperti <code>img_foto</code>
	Data Control	Sebagai sarana akses data dalam suatu database. Fasilitas ini ada dalam konsep DAO.
	OLE	Untuk menghasilkan proses <i>Link</i> dan <i>Embed</i> objek antar aplikasi.

2.5. Membuka *Project* dan Menyimpan *Project*

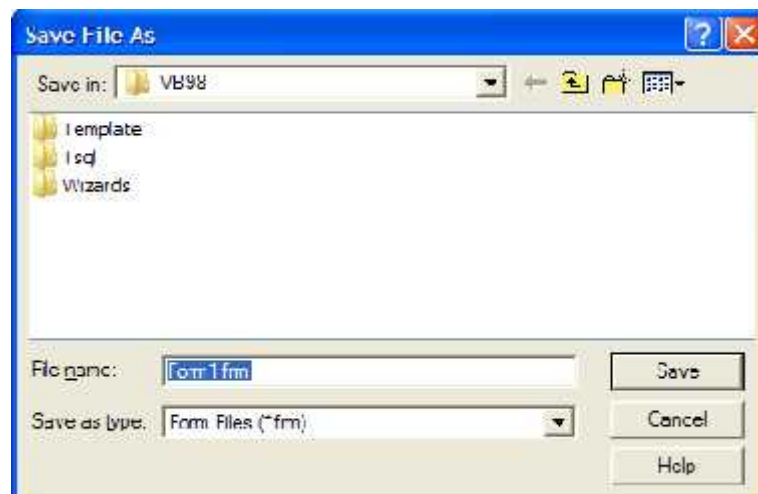
Untuk memulai pembuatan program aplikasi di dalam Visual Basic, yang dilakukan adalah membuat project baru. Project adalah sekumpulan form, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat project baru dapat dilakukan dengan memilih menu **File >> New Project** atau dengan menekan ikon **new project**  pada toolbar yang terletak di pojok kiri atas >> **Klik OK**. Setelah itu akan muncul konfirmasi untuk jenis project dari program aplikasi yang akan dibuat seperti terlihat pada gambar 2.15 berikut:



Gambar 2.15. Memulai project visual basic

Sebuah proyek pada Visual Basic terdiri atas beberapa file. Jika program semakin kompleks, maka program akan terdiri atas beberapa form. Form-form itu diletakkan pada file terpisah. Itulah yang kemudian menyebabkan saat program disimpan, proses penyimpanan akan membuat 2 file, yaitu file untuk menyimpan proyek dan sebuah file untuk menyimpan form. File proyek diberi nama *.vbp, sedangkan file form diberi extension *.frm. Untuk menyimpan *project visual basic* yang telah dikerjakan, langkah-langkahnya adalah sebagai berikut:



- ✓ Pada menu **File**, klik perintah **Save Project As** kemudian akan muncul Kotak dialog **File Project** seperti terlihat pada gambar 2.16 dibawah ini. Melalui kotak dialog tersebut dapat menyimpan program Visual Basic yang telah selesai dikerjakan.
- ✓ Pilih direktori tempat menyimpan hasil kerja project misalkan D:\VB\Latihan01 dengan mengklik pada kontrol **combo box**.
- ✓ Simpan form dengan nama **Form1**, kemudian klik tombol Save. Simpan **Project** dengan nama **Project1**, kemudian klik tombol **Save**.



Gambar 2.16. Menyimpan project

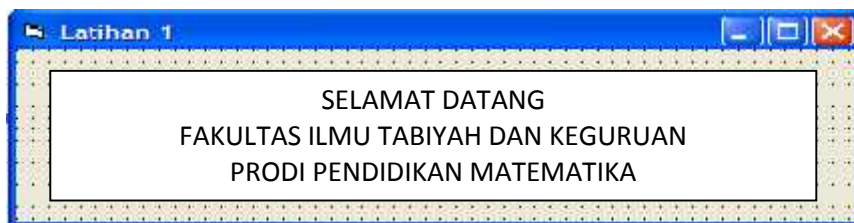
Rangkuman

Untuk membuka dan memulai program visual basic 6.0 (yang dikemas dalam *Microsoft Visual Studio atau Enterprise 6.0*), langkah-langkahnya adalah sebagai berikut: Klik *Start – All Programs (Programs) - Microsoft Visual Basic 6.0 – Microsoft Visual Basic 6.0*. Pembuatan program aplikasi menggunakan visual basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di (*toolbox*), dan setiap komponen yang dipakai harus diatur propertinya lewat jendela (*property*). Untuk menjalankan atau mengeksekusi *project* yang telah dibuat dan menghentikannya, lakukan langkah-langkah sebagai berikut:

- ✓ Klik tombol *start* () pada *toolbar* atau dengan menekan tombol **F5** atau juga dapat menggunakan *start* pada menu *run*.
- ✓ Sedangkan untuk menghentikan program yang sedang berjalan dapat menggunakan tombol *End* () pada *toolbar* atau menggunakan *End* pada menu *run*.

Latihan

1. Buat project baru, letakkan *object label* pada *Form*, atur *property Form* dan *label* seperti terlihat pada gambar di bawah ini :



- ✓ Jalankan program yang telah dibuat tadi dan lihat hasilnya
- ✓ Hentikan program kemudian simpan program
- ✓ Simpan *project* dengan nama *File Project (*.vbp)* beri nama : “ *PrjLat01.vbp* “ dan *File Form1 (*.frm)* beri nama : “ *FrmLat01.frm* “

Tindak Lanjut (*Feed Back*)

Skor	Kriteria
100-80	Mampu dan dapat melakukan kegiatan latihan : 1. Membuka <i>project standart Exe</i> 2. Meletakkan <i>object label</i> pada <i>form</i> 3. Mengatur properti <i>object label</i> dan <i>form</i>
70-60	Hanya mampu melakukan kegiatan menyimpan, membuka dan meletakkan objek label pada <i>Form</i> dan <i>Project</i> saja. Tidak mampu bekerja dengan properti <i>label</i> dan <i>form</i> , (tidak tuntas) dan harus mengulang bahagian tertentu saja
50 ke bawah	Tidak mampu sama sekali dan harus mengulang kegiatan belajar 2 secara keseluruhan



Kegiatan Belajar 3.

BAB
III

MENGENAL

METHOD, EVENT DAN MESSAGE**3.1. Pengertian *Properties***

Pemrograman visual basic 6.0 merupakan pemrograman yang berorientasi objek (*Object Oriented Programming / OOP*) yang berbasis visual. Visual adalah suatu pemrograman, dimana pembuatan program dikerjakan dengan memanfaatkan antarmuka interaksi pengguna dengan berbagai objek yang terlihat pada *user-interface*. Properties adalah karakteristik atau sifat-sifat dari sebuah obyek. Properties merupakan jendela yang mengandung semua informasi mengenai semua objek yang terdapat pada aplikasi visual basic. Untuk menentukan atau mengubah properti suatu objek atau kontrol harus melakukannya di dalam jendela properties.

Properties memiliki dua buah tab yaitu *Alphabetic* dan *Categorized*. Pada tab *alphabetic* semua properti dari objek akan diurutkan berdasarkan abjad. Sedangkan pada tab *categorized* semua properti dikelompokkan berdasarkan jenis. Setiap komponen di dalam pemrograman visual basic dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Properties yang tidak boleh dilupakan pada setiap komponen adalah “*Name*”, yang berarti nama variabel (komponen) yang akan digunakan dalam *script*. Properties “*Name*” ini hanya dapat diatur melalui jendela properties, sedangkan nilai perproperties yang lain dapat diatur melalui *script*. Contoh *properties object label* yang diatur adalah “(*Name*) diganti *lblname* “, “ *BackColor* diganti dengan memilih *tab palette red* “ dan “ *Caption* diganti dengan warna “. Perhatikan keterangan gambar untuk properties dibawah ini:



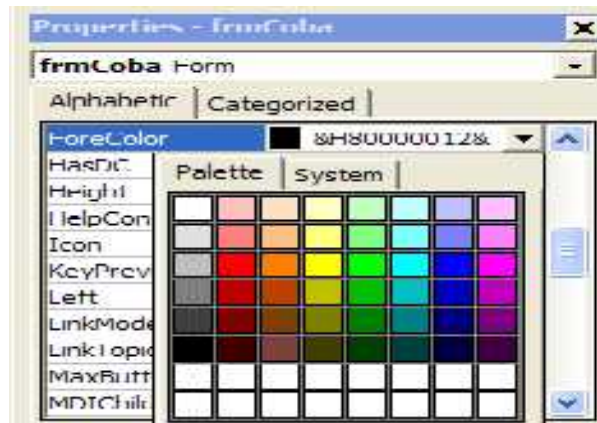
Gambar 3.1. Properties visual basic.

Form merupakan objek utama dalam pemrograman visual basic, karena form merupakan tempat dimana semua objek diletakkan. Form sebagai lembar kerja pada visual basic yang memiliki **properties**, **method** dan **event**. Pada bagian ini akan dibahas tentang properti, metode (*methods*), dan **event** yang berhubungan dengan objek-objek tersebut. Semua objek-objek yang terlihat (*visible*) memiliki properties *Left*, *Top*, *Width*, dan *Height*. Properties ini digunakan untuk menempatkan posisi dan ukuran suatu objek. Nilai dari properti ini selalu relatif terhadap objek kontainernya (tempat dimana objek tersebut menempel) dan satuan ukuran default yang digunakan adalah *twips*.

Properti-properti tersebut dapat diubah melalui jendela properti (*properties window*) dengan memasukkan suatu nilai numerik pada saat waktu merancang atau dengan memindahkan atau mengubah ukuran form secara interaktif dengan menggunakan mouse. Tidak semua kontrol dalam visual basic memiliki **properties left, top, width, dan height**, sebagai contoh objek **timer** Selain itu perubahan properti juga dapat dilakukan pada saat program dijalankan (*run time*) melalui kode seperti contoh dibawah ini :

```
' Membuat lebar frmCoba menjadi 2 kali lebar semula  
frmCoba.Width = frmCoba.Width * 2  
  
' Membuat tinggi frmCoba menjadi 3 kali tinggi semula  
frmCoba.Height = frmCoba.Height * 3  
  
' Meletakkan frmCoba di sudut kiri atas  
frmCoba.Left = 0  
frmCoba.Top = 0
```

Hampir semua objek mendukung **properties ForeColor** dan **BackColor**, properties *ForeColor* digunakan untuk mempengaruhi warna tulisan dan properti *BackColor* digunakan untuk memilih warna dasar dari objek. Dalam kasus tertentu, properti tersebut tergantung pada properti lain. Sebagai contoh mengganti warna dasar (*background*) tidak akan berpengaruh jika nilai dari properti *backstyle* bernilai *0-Transparent*.



Gambar 3.2. Properties ForeColor

Ada 2 macam warna yang dapat digunakan pada properti **BackColor** dan **ForeColor** yaitu **Standart Color (tab System)** dan **Custom Color (tab Palette)**. Kedua jenis warna tersebut dapat dipilih dengan menggunakan tab, namun sebaiknya digunakan warna standar (*standart color*), kecuali ada alasan yang kuat untuk menggunakan warna khusus (*custom color*). Alasan penggunaan warna standar adalah warna ini akan bekerja dengan baik pada semua komputer dengan sistem operasi *Windows*. Visual Basic telah menyediakan sejumlah konstanta simbolik yang menyatakan sebuah warna. Ada beberapa cara untuk mengisi nilai properties *BackColor* dan *ForeColor* melalui kode:

```
' Mengatur property ForeColor dan BackColor
' Pada lblCoba
lblCoba.ForeColor = vbHighlightText
lblCoba.BackColor = vbHighlight
```

Properties ini juga dapat diisi dengan menggunakan konstanta simbol yang lain seperti *vbBlack*, *vbBlue*, *vbCyan*, *vbGreen*, *vbMagenta*, *vbRed*, *vbWhite*, and *vbYellow* atau menggunakan nilai konstanta desimal/heksadesimal.

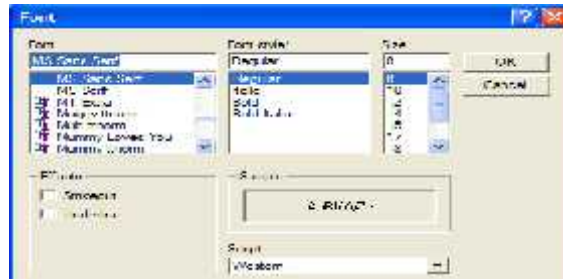
```
' Ketiga kode program di bawah ini mempunyai hasil sama
txtCoba.BackColor = vbCyan
txtCoba.BackColor = 16776960
txtCoba.BackColor = &HFFFF00
```

Atau dapat pula digunakan fungsi RGB yang membangun warna berdasarkan warna Merah (R), Hijau (G). dan Biru(B) (*Red, Green, dan Blue*) atau fungsi *QBColor* yang merupakan fungsi dari *Quick Basic*.

Tabel 3.1. Nilai konstanta warna

Konstanta	Heksadesimal	Keterangan
vb3DDKShadow	&H80000015	Darkest shadow
vb3Dface	&H8000000F	Dark shadow color for 3-D display elements
vb3Dhighlight	&H80000014	Highlight color for 3-D display elements
vb3Dlight	&H80000016	Second lightest of the 3-D colors after vb3Dhighlight
vb3Dshadow	&H80000010	Color of automatic window shadows
vbActiveBorder	&H8000000A	Active window border color
vbActiveTitleBar	&H80000002	Active window caption color
vbActiveTitleBarText	&H80000009	Text color in active caption, size box, scroll bar arrow box
vbApplicationWorkspace	&H8000000C	Background color of multiple-document interface (MDI) applications
vbButtonFace	&H8000000F	Face shading on command buttons
vbButtonShadow	&H80000010	Edge shading on command buttons
vbButtonText	&H80000012	Text color on push buttons
vbDesktop	&H80000001	Desktop color
vbGrayText	&H80000011	Grayed (disabled) text
vbHighlight	&H8000000D	Background color of items selected in a control
vbHighlightText	&H8000000E	Text color of items selected in a control
vbInactiveBorder	&H8000000B	Inactive window border color
vbInactiveCaptionText	&H80000013	Color of text in an inactive caption
vbInactiveTitleBar	&H80000003	Inactive window caption color
vbInactiveTitleBarText	&H80000013	Text color in inactive window caption, size box, scroll bar arrow box
vbInfoBackground	&H80000018	Background color of ToolTips
vbInfoText	&H80000017	Color of text in ToolTips
vbMenuBar	&H80000004	Menu background color
vbMenuText	&H80000007	Text color in menus
vbScrollBars	&H80000000	Scroll bar gray area color
vbTitleBarText	&H80000009	Text color in active caption, size box, scroll bar arrow box
vbWindowBackground	&H80000005	Window background color
vbWindowFrame	&H80000006	Window frame color
vbWindowText	&H80000008	Text color in windows

Font merupakan objek gabungan (*Compound Object*) yang masih mempunyai properti-properti yang terpisah yaitu *Font Name*, *Font Style*, *Font Size* dan *Effects*. *Properties Font* dapat diisi dengan menggunakan *Font Dialog* seperti terlihat pada gambar 3.3 dibawah ini



Gambar 3.3 Kotak dialog Font

Cara lain untuk mengubah nilai properties dapat dilakukan melalui kode program, seperti contoh di bawah ini:

```
txtnama.Font.Name = "Yahfizham"
txtnama.Font.Size = 16
txtnama.Font.Bold = True
txtnama.Font.Italic = True
```

Properties *Caption* digunakan untuk menentukan kata atau kalimat yang ditampilkan pada sebuah kontrol, seperti judul dari *Form* dan kalimat pada sebuah *Label*. Sedangkan Properti *Text* sama seperti properti *Caption* hanya saja kalimat dalam properti *Text* dapat diganti. Tidak ada kontrol yang mendukung kedua properti ini secara bersamaan. Kontrol *Label*, *CommandButton*, *CheckBox*, *OptionButton*, *Data*, dan *Frame* mendukung properti *Caption* sedangkan kontrol *TextBox*, *ListBox*, dan *ComboBox* mendukung properti *Text*. Khusus untuk properti *Caption* mendukung penggunaan karakter ampersand (&) untuk menentukan *hotkey* atau *shortcut* dari kontrol.

```
'Kedua kode program di bawah ini mempunyai hasil sama
txtCoba.Text = Text1.Text
txtCoba = Text1
```

Secara default nilai dari properti *Enabled* dan *Visible* adalah *True*, tetapi mungkin pada saat program dijalankan mungkin sebuah kontrol ingin disembunyikan (*hide*) atau ditampilkan dengan keadaan tidak bisa digunakan (*disabled*). Nilai properti *Enable* dan *Visible* dapat diubah melalui kode program seperti terlihat pada kotak berikut ini.

```
' Membuat txtCoba yang disembunyikan
txtCoba.Visible = False

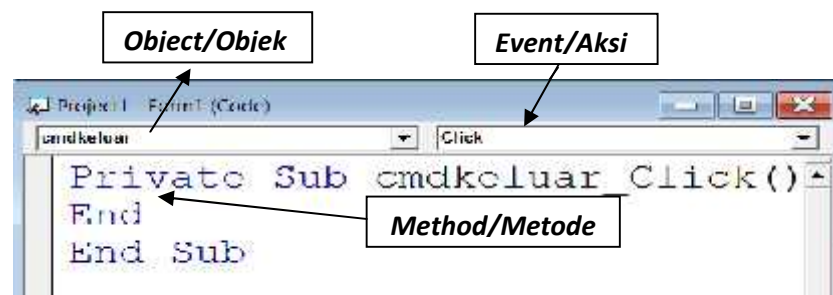
' Membuat txtCoba tidak dapat diisi digunakan
txtCoba.Enabled = False
```

3.2. Pengertian *Method*

Method adalah aksi atau perbuatan yang dapat dimiliki oleh kontrol (objek) sehingga user (*programmer*) dapat memakainya untuk memanipulasi sesuatu. *Method* tergantung dari instruksi yang diberikan oleh programmer melalui penulisan kode. Metode (*method*) merupakan serangkaian *script* yang dimiliki oleh suatu obyek yang akan dijalankan sesuai dengan respon yang diberikan oleh suatu perintah atau kejadian yang dilakukan oleh user. Contohnya objek *command button* memiliki *properties name* : *cmdkeluar*, *caption* : *keluar*, memiliki *method* : baris atas : *Private Sub cmdkeluar()*, baris tengah : *End* dan baris akhir : *End Sub*. Lihat penjelasan gambar untuk objek *command button* dan *method* dibawah ini



Gambar 3.4. Objek *command button* dengan *method*



Gambar 3.5. *Script* untuk objek *command button*

Jika suatu kontrol mempunyai properti *Left*, *Top*, *Width*, dan *Height*, maka kontrol tersebut pasti juga mendukung metode *move*. Metode ini digunakan untuk mengubah properti *Left*, *Top*, *Width*, dan *Height* dengan menggunakan sebuah operasi tunggal seperti terlihat pada kotak dibawah ini.

```
' Lebar Form menjadi dua kali lipat dan
' posisinya berada pada pojok kiri atas.
' Syntaxnya : Move(Left As Single, [Top], [Width], [Height])
frmCoba.Move 0, 0, frmCoba.Width * 2, frmCoba.Width
```

Method Refresh merupakan metode yang digunakan untuk menggambar kembali suatu kontrol. Pada keadaan normal biasanya kita tidak perlu memanggil metode ini, sebab Visual Basic secara otomatis akan menyegarkan kembali penampilan suatu kontrol ketika sudah diubah. Metode ini digunakan jika suatu kontrol ingin segera disegarkan kembali tanpa menunggu proses lain yang masih harus dilakukan. Penggunaan metode tersebut dalam kode program dapat dilihat pada kotak dibawah ini.

'Menyegarkan kembali *command button*
cmdrefresh.Refresh

Method SetFocus digunakan untuk memindahkan fokus atau kursor aktif dari masukan (*input*) yang diisikan oleh user ke suatu kontrol tertentu. Metode ini digunakan untuk memodifikasi nilai *TabOrder* yang sudah ditetapkan sebelumnya.

3.3. Pengertian *Event*

Event atau aksi adalah kejadian atau segala sesuatu yang dapat dialami oleh sebuah objek. Masing-masing objek memiliki aksi yang berbeda atau dapat juga mempunyai aksi yang sama. Suatu objek, selain memiliki *properties* dan *method*, visual basic 6.0 juga menyediakan sejumlah aksi/kejadian (*events*). Berikut ini merupakan aksi-aksi yang lazim dipakai atau disediakan:

- Aksi *Click* dan *DoubleClick*

Aksi *Click* dan *DoubleClick* terjadi pada saat user melakukan *click* atau *double click*

- Aksi *Change*

Aksi *Change* dibangkitkan pada saat isi dari suatu kontrol mengalami perubahan.

- Aksi *GotFocus* dan *LostFocus*

Aksi *GotFocus* akan dibangkitkan pada saat sebuah kontrol menerima fokus input, sedangkan *LostFocus* merupakan kebalikan dari kejadian *GotFocus*.

- Aksi *KeyDown*, *KeyUp*, dan *KeyPress*

Aksi *KeyDown*, *KeyUp*, dan *KeyPress* adalah aksi ke bawah, dilepas, dan ditekan.

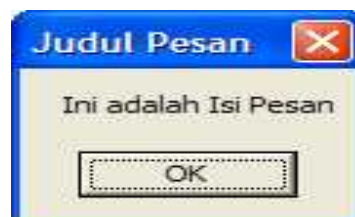
- Aksi *MouseDown*, *MouseUp*, dan *MouseMove*

Aksi ini terjadi pada saat mouse di-*click*, dilepas, dan digerakkan.

- *Activate*
Aksi ini terjadi ketika sebuah form menjadi window yang aktif.
- *Deactive*
Aksi ini terjadi ketika anda berpindah dari satu form ke form lain.
- *GotFocus*
Aksi ini terjadi ketika sebuah objek menjadi satu-satunya focus. Sebuah form dapat membuat objek focus ketika form tersebut menjadi window aktif.
- *Lostfocus*
Aksi ini terjadi ketika sebuah objek kehilangan fokus karena ada fokus.
- *Load*
Aksi ini terjadi ketika sebuah form dibuka atau dipanggil.
- *Unload*
Aksi ini terjadi ketika sebuah form ditutup.
- *Resize*
Aksi ini terjadi ketika sebuah form diubah ukurannya.
- *Timer*
Aksi ini terjadi ketika aksi lain dijalankan sebelum aksi timer ini.

3.4. Kotak Pesan (*Message Box*)

Seorang programmer yang cerdas selalu memberikan kotak pesan (*message box*) kepada user sebagai suatu konfirmasi ketika user berinteraksi dengan aplikasi yang dirancangnya. Adakalanya sebuah program ingin menampilkan pesan kesalahan atau bertanya pada user, sebab kontrol – kontrol yang ada pada *form* kurang jelas. *Message Box* berisi pesan yang akan ditampilkan tepat diatas sebuah form dan akan hilang ketika mendapat respon dari user ketika mengklik tombol yang ada pada *message box*. Contoh kotak pesan seperti gambar yang terlihat dibawah ini:



Gambar 3.6. Kotak pesan

3.4.1. Statement MsgBox

Pernyataan pesan yang ditampilkan melalui *Msgbox* menampilkan tombol OK, ketika user selesai membaca pesan tersebut, maka user akan menekan tombol Ok untuk menutup pesan tersebut. Berikut ini format statement MsgBox :

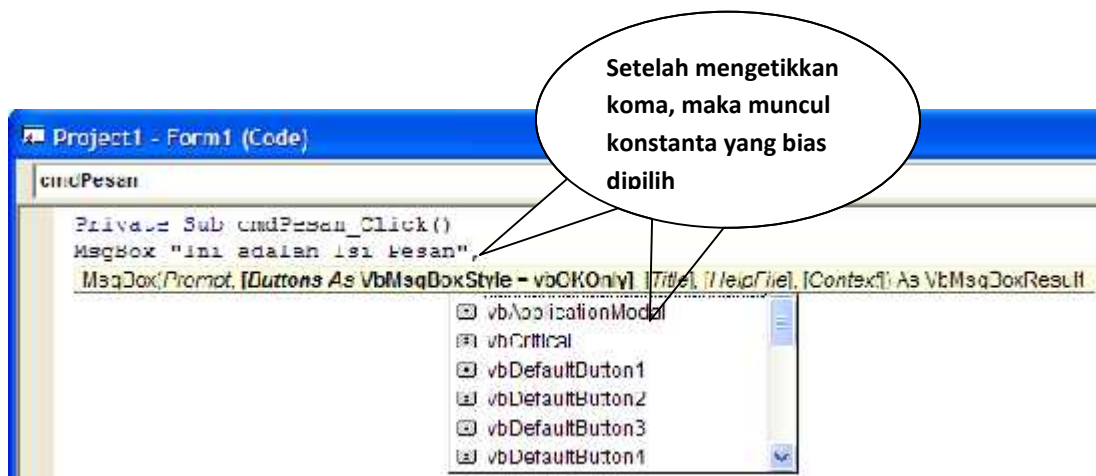
MsgBox Prompt, Style Value, Title

Prompt merupakan kalimat atau variabel yang berisi string yang akan ditampilkan. *Style* menentukan tipe dari command button yang akan terlihat pada message box dan nilainya dapat dipilih salah satu seperti pada tabel 3.2. Sedangkan *Title* menunjukkan judul dari message box.

Tabel 3.2. Nilai konstanta *message box*

STYLE VALUE	KONSTANTA	TOMBOL YG DITAMPILKAN
0	vbOkOnly	Ok button
1	vbOkCancel	Ok dan Cancel buttons
2	vbAbortRetryIgnore	Abort, Retry dan Ignore buttons.
3	vbYesNoCancel	Yes, No dan Cancel buttons
4	vbYesNo	Yes dan No buttons
5	vbRetryCancel	Retry dan Cancel buttons

Kita dapat menggunakan nama konstanta atau *style value* untuk menggantikan nilai integer pada argumen kedua. Penggunaan nama konstanta akan lebih mudah dibaca dibandingkan menggunakan *style value*. Visual Basic akan menampilkan daftar dari nama konstanta tersebut begitu kita mengetikkan tanda koma setelah argumen pertama.



Gambar 3.7. Pernyataan pesan *MsgBox*

3.4.2. Fungsi MsgBox

Untuk menampung tombol mana yang ditekan oleh user, maka digunakan fungsi *MsgBox*. Format pada fungsi `MsgBox()` sedikit berbeda dengan statement `MsgBox`. Fungsi ini menyediakan type yang lebih luas dibanding Statement `MsgBox`.

PesanAnda=MsgBox(Prompt, Style Value, Title

PesanAnda adalah variabel yang akan menampung nilai dari fungsi `MsgBox()`. Nilainya menentukan type dari tombol yang diklik oleh user. Variabel tersebut haruslah dideklarasikan sebagai tipe data integer pada *general declaration section*.

Tabel 3.3 Nilai konstanta saat tombol *MessageBox* diklik

NILAI	KONSTANTA	TOMBOL YG DIKLIK
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button
6	vbYes	Yes button
7	vbNo	No button

`InputBox` akan menampilkan message dimana user dapat menginputkan suatu nilai atau message pada form tersebut. `MyMessage` adalah tipe data variant yang dideklarasikan sebagai string. Argumen yang tersedia dapat dijelaskan sebagai berikut:

- Prompt : Pesan yang ditampilkan.
- Title : Judul dari Input Box.
- Default-text: Default text yang ditampilkan pada field dimana user dapat menggunakannya atau menggantinya.

MyMessage=InputBox(Prompt, Title, default_text, x-position, y-position)

Rangkuman

Properties adalah karakteristik atau sifat-sifat dari sebuah obyek. Properties merupakan jendela yang mengandung semua informasi mengenai semua objek yang terdapat pada aplikasi visual basic. Untuk menentukan atau mengubah properti suatu objek atau kontrol harus melakukannya di dalam jendela properties. Properties memiliki dua buah tab yaitu **Alphabetic** dan **Categorized**. Pada tab **alphabetic** semua properti dari objek akan diurutkan berdasarkan abjad. Sedangkan pada tab **categorized** semua properti dikelompokkan berdasarkan jenis. **Method** adalah aksi atau perbuatan yang dapat dimiliki oleh kontrol (objek) sehingga user (*programmer*) dapat memakainya untuk memanipulasi sesuatu. **Method** tergantung dari instruksi yang diberikan oleh programmer melalui penulisan kode. Metode (*method*) merupakan serangkaian **script** yang dimiliki oleh suatu obyek yang akan dijalankan sesuai dengan respon yang diberikan oleh suatu perintah atau kejadian yang dilakukan oleh user. **Event** atau aksi adalah kejadian atau segala sesuatu yang dapat dialami oleh sebuah objek. **Message box** memberikan kotak pesan (*message box*) kepada user sebagai suatu konfirmasi ketika user berinteraksi dengan aplikasi.

Latihan

1. Kerjakanlah latihan ke-2 berikut :

a) Buat Project baru dan Form seperti antarmuka (tampilan) gambar dibawah ini



Gambar 3.8. Contoh latihan ke-2 program sederhana

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	frmLat02
		BackColor	&H00C0FFFF&
		Caption	Latihan Ke-2
		Height	7020
		Width	8268
		StartPosition	2 – Center Screen
		WindowState	0 –Normal
2	Label1	(Name)	Lbljudul
		Caption	PROGRAM SEDERHANA
		Font	Ms Sans Serif – Bold – 19
		ForeColor	&H00FF0000&
3	Label2	(Name)	Lblket
		Caption	ARAHKAN KURSOR PADA KOTAK DIBAWAH INI DAN ISIKAN SUATU KALIMAT BEBAS
		Font	Ms Sans Serif – Bold – 14
		ForeColor	&H00000000&
4	CommandButton1	(Name)	Cmdproses
		BackColor	&H8000000F&
		Caption	PROSES
		Font	Ms Sans Serif – Bold – 19
5	CommandButton2	(Name)	Cmdhapus
		BackColor	&H8000000F&
		Caption	HAPUS
		Font	Ms Sans Serif – Bold – 19
6	CommandButton3	(Name)	Cmdkeluar
		BackColor	&H8000000F&
		Caption	<<< K E L U A R >>>
		Font	Ms Sans Serif – Bold – 19
7	TextBox1	(Name)	Txthasil
		BackColor	&H80000005&
		Font	Ms Sans Serif – Bold – 19
		Text	Tulisan “ Text1” dihapus (jadi kosong)

c) Ketikkan instruksi (*source code*) dan sesuaikan dengan objek-objeknya.

```
Private Sub CmdHapus_Click()
    TxtHasil.Text = ""
End Sub
Private Sub CmdKeluar_Click()
    End
End Sub
Private Sub CMDPROSES_Click()
    TxtHasil.Text = " SELAMAT BELAJAR VISUAL BASIC 6.0"
End Sub
```

d) Jalankan program yang telah dibuat tadi dan lihat hasilnya

e) Hentikan program kemudian simpan program

f) Simpan *project* dengan nama *File Project (*.vbp)* beri nama : “ *PrjLat02.vbp* “ dan *File Form1 (*.frm)* beri nama : “ *FrmLat02.frm* “

2. Kerjakanlah latihan ke-3 berikut :

a) Buat Project baru dan Form antarmuka (tampilan) gambar dibawah ini



Gambar 3.9. Contoh latihan ke-3 program teks sederhana

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	frmLat03
		BackColor	&H00C0FFFF&
		Caption	Latihan Ke-3
		Height	6924
		Width	8556
		StartPosition	2 - Center Screen
		WindowState	0 -Normal
2	Line1	(Name)	Line1
		BorderWidth	6
3	Label1	(Name)	Label1
		Caption	Ketikkan Kata pada kursor yang aktif dibawah ini
		Font	Ms Sans Serif - Bold - 14
		ForeColor	&H00000000&
4	TextBox1	(Name)	Txtinput
		Alignment	2 - Center
		BackColor	&H80000005&
		Font	Ms Sans Serif - Bold - 19
		ForeColor	&H80000008&
		Text	Tulisan " Text1" dihapus (jadi kosong)
5	Label2	(Name)	Label2
		Caption	Setelah Mengetik Kata, lanjutkan dengan Memilih Warna dan Style Huruf Lalu Klik Tombol Proses
		Font	Ms Sans Serif - Bold - 14
		ForeColor	&H000000FF&
6	Label3	(Name)	lbloutput
		Alignment	2 - Center
		Appearance	0 - Flat
		BackColor	&H80000005&
		BorderStyle	1 - Opaque
		Font	Ms Sans Serif - Bold - 14
		ForeColor	&H000000FF&

7	Label4	(Name)	Label4
		Caption	PROGRAM TEKS SEDERHANA
		Font	Ms Sans Serif – Bold - 19
		ForeColor	&H00FF0000&
8	CommandButton1	(Name)	cmdproses
		Caption	Proses
		Font	Ms Sans Serif – Bold - 19
		BackColor	&H80000005&
		Height	732
		Width	4692
9	CommandButton2	(Name)	cmdkeluar
		Caption	Keluar
		Font	Ms Sans Serif – Bold - 19
		BackColor	&H80000005&
		Height	732
		Width	4692
10	Frame1	(Name)	Frame1
		Caption	Pilihan Warna
		Font	Ms Sans Serif – Bold - 19
11	OptionButton1	(Name)	optmerah
		Caption	Merah
		Font	Ms Sans Serif – Bold - 19
12	OptionButton2	(Name)	optbiru
		Caption	Merah
		Font	Ms Sans Serif – Bold - 19
13	OptionButton3	(Name)	opthijau
		Caption	Merah
		Font	Ms Sans Serif – Bold - 19
14	Frame2	(Name)	Frame2
		Caption	Style Huruf
		Font	Ms Sans Serif – Bold - 19
15	CheckBox1	(Name)	ckbtebal
		Caption	Tebal
		Font	Ms Sans Serif – Bold - 19
16	CheckBox2	(Name)	ckbmiring
		Caption	Miring
		Font	Ms Sans Serif – Bold - 19
17	CheckBox3	(Name)	Ckbunder
		Caption	Garis Bawah
		Font	Ms Sans Serif – Bold - 19

c) Ketikkan instruksi (*source code*) dan sesuaikan dengan objek-objeknya.

```
Private Sub Form_Load()
    Cmdproses.Enabled = False
End Sub
```

```
Private Sub CkbMiring_Click()
    LblOutput.FontItalic = CkbMiring.Value
    Cmdproses.Enabled = True
End Sub
```

```
Private Sub CkbTebal_Click()
    LblOutput.FontBold = CkbTebal.Value
    Cmdproses.Enabled = True
End Sub

Private Sub ckbunder_Click()
    LblOutput.FontUnderline = ckbunder.Value
    Cmdproses.Enabled = True
End Sub

Private Sub CmdKeluar_Click()
    End
End Sub

Private Sub Cmdproses_Click()
    LblOutput.Caption = TxtInput.Text
End Sub

Private Sub OptBiru_Click()
    LblOutput.ForeColor = vbBlue
    Cmdproses.Enabled = True
End Sub

Private Sub opthijau_Click()
    LblOutput.ForeColor = vbGreen
    Cmdproses.Enabled = True
End Sub

Private Sub OptMerah_Click()
    LblOutput.ForeColor = vbRed
    Cmdproses.Enabled = True
End Sub

Private Sub TxtInput_Change()
    TxtInput.SetFocus
End Sub
```

- d) Jalankan program yang telah dibuat tadi dan lihat hasilnya
- e) Hentikan program kemudian simpan program
- f) Simpan project dengan nama File Project (*.vbp) beri nama : “ PrjLat03.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat03.frm “

3. Kerjakanlah latihan ke-4 berikut :

- a) Buat Project baru dan Form dengan tiga command button dan 1 label.



Gambar 3.10. Latihan menggunakan *MessageBox*

b) Ketikkan *source code* pada objek *commandbutton1* dibawah ini

```
Private Sub Command1_Click()  
    Dim testMsg As Integer  
    testMsg = MsgBox(" Silahkan Anda Klik Tombol OK", 1, " Test ")  
    If testMsg = 1 Then  
        Label1.Caption = " Anda Menekan Tombol OK"  
    Else  
        Label1.Caption = " Anda Menekan Tombol Cancel"  
    End If  
End Sub
```

c) Ketikkan *source code* pada objek *commandbutton2* dibawah ini

```
Private Sub Command2_Click()  
    Dim testMsg As Integer  
    testMsg = MsgBox(" Silahkan Pilih Tombol Dibawah", vbOKCancel, " Pilih")  
    If testMsg = vbOK Then  
        Label1.Caption = " Anda Menekan Tombol OK"  
    Else  
        Label1.Caption = " Anda Menekan Tombol Cancel"  
    End If  
End Sub
```

d) Ketikkan *source code* pada objek *commandbutton3* dibawah ini

```
Private Sub Command3_Click()  
    Dim testMsg As Integer  
    testMsg = MsgBox(" Silahkan Anda Klik", vbYesNoCancel _  
        + vbExclamation, " Konfirmasi")  
    If testMsg = 6 Then  
        Label1.Caption = " Selamat, Anda Berhasil"  
    ElseIf testMsg = 7 Then  
        Label1.Caption = " Apakah Anda Benar-Benar Yakin ???"  
    Else  
        Label1.Caption = " Maaf Anda Gagal"  
    End If  
End Sub
```

e) Jalankan program yang telah dibuat tadi dan lihat hasilnya

f) Hentikan program kemudian simpan program

g) Simpan project dengan nama File Project (*.vbp) beri nama : “ PrjLat04.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat04.frm “

Tindak Lanjut (*Feed Back*)

Skor	Kriteria
100-80	Mampu dan dapat melakukan kegiatan latihan ke-2, ke-3 dan ke-4 : 1. Bekerja dengan <i>properties, method</i> dan <i>event</i> 2. Bekerja dengan objek, <i>properties, method</i> dan <i>event</i> 3. Bekerja dengan objek, <i>properties, method, event</i> dan <i>message box</i>
70-60	Hanya mampu melakukan kegiatan latihan ke-2 dan ke-3 atau ke-2 dan ke-4 atau ke-3 dan ke-4
50 ke bawah	Tidak mampu sama sekali dan harus mengulang kegiatan belajar 3 secara keseluruhan



Kegiatan Belajar 4.

BAB
IVMENGENAL
TIPE DATA VISUAL BASIC 6.0

4.1. Pengertian Variabel

Variabel adalah deklarasi sesuatu yang memiliki variasi (berubah-ubah) dan tidak tetap dari suatu nilai. Variabel digunakan untuk menyimpan nilai sesuatu yang dimiliki program aplikasi yang dibentuk oleh programmer sendiri. Nilai yang ditampung atau disimpan oleh suatu variabel dapat berubah-ubah sesuai keinginan. Variabel merupakan tempat untuk menyimpan nilai sementara dari suatu perhitungan. Untuk mendeklarasikan sebuah variabel digunakan pernyataan :

Dim Variabel [*As* Type]

Melalui *As* programmer dapat mendefinisikan sendiri tipe data atau tipe objek dari variabel yang dideklarasikan. Tipe data ini misalnya integer, string atau variant. Variabel untuk sebuah prosedur hanya boleh digunakan pada prosedur di tempatnya dideklarasikan, dan biasanya digunakan dalam perhitungan sementara (*temporary*). Aturan di dalam pembuatan atau penamaan dari variabel :

- Harus diawali dengan huruf tidak boleh angka.
- Harus unik, tidak boleh ada variabel dengan nama sama pada satu ruang lingkup yang sama.
- Tidak boleh menggunakan spasi, dapat memakai simbol garis hubung bawah (*underscore*)
- Tidak boleh menggunakan karakter-karakter khusus seperti : +, -, *, /, <>, dll.
- Tidak boleh menggunakan kata-kata kunci atau *key assignment* atau *reserved word* yang memang dimiliki oleh Visual Basic 6 misalnya : ***dim, as, string, integer***, dll).
- Tidak boleh lebih dari 255 karakter, hanya 40 karakter pertama yang dianggap sebagai nama variabel, sisanya diabaikan.

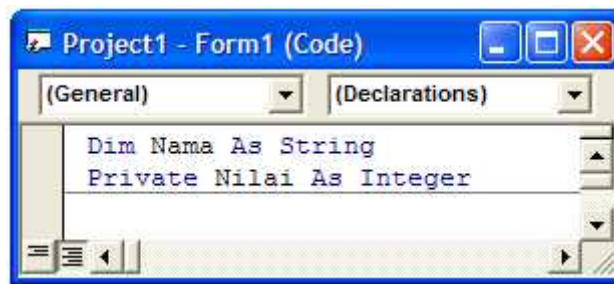
Satu variabel hanya dapat menyimpan satu nilai data sesuai dengan type datanya. Cara mengisi nilai data ke dalam sebuah variabel :

<nama_variabel> = <nilai_data>

Contoh : programmer = "Yahfizham"

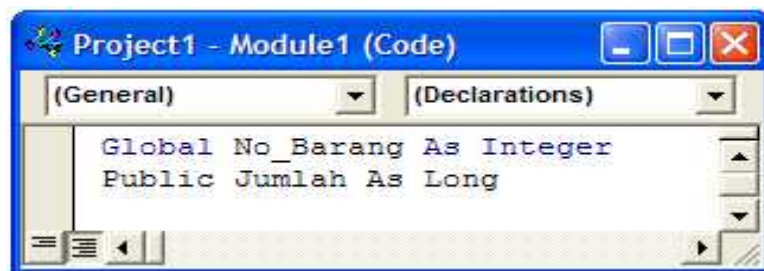
Ada tiga level ruang lingkup variabel, yaitu : *level lokal*, *level form/module*, dan *level global*.

- Variabel *level lokal* adalah variabel yang hanya dikenali dan dapat digunakan dalam prosedur tempat variabel tersebut dideklarasikan. Prosedur di bagian lain tidak dapat mengakses variabel tersebut. Variabel ini dideklarasikan menggunakan perintah **Dim**, **Private**, dan **Static**.
- Variabel dengan *level form/module* dideklarasikan dibagian **general declarations** dalam jendela kode program form/modul. Nilai variabel ini dapat dikenali dan dipakai pada semua prosedur yang ada dalam form/modul tersebut. Deklarasi variabel ini juga menggunakan pernyataan **Dim**. Perbedaan dengan variabel *lokal* adalah tempat deklarasi variabel tersebut. Contoh deklarasi variabel level form/module:



Gambar 4.1. Deklarasi variabel level form/module

- Variabel *level global* (aplikasi) dideklarasikan di bagian **general declarations** dalam jendela kode program form/modul menggunakan pernyataan **Public** atau **Global**. Variabel ini dapat dipanggil dan dipakai oleh semua form/modul dalam program aplikasi yang kita buat, dan juga dapat dipanggil dan dipakai oleh seluruh prosedur yang ada dalam form/module tersebut. Contoh deklarasi variabel level global adalah sebagai berikut:



Gambar 4.2. Deklarasi variabel level global (aplikasi)

Untuk type data tertentu nilai_data harus diapit tanda pembatas. *Type data string* dibatasi tanda petik-ganda : “nilai data”. *Type data date* dibatasi tanda pagar : #nilai data*. Type data lainnya tidak perlu tanda pembatas. Sebuah variabel mempunyai ruang-lingkup (*scope*) dan waktu-hidup (*life-time*). Dari penjelasan diatas dapat diartikan bahwa **Variabel global** adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan. Dari penjelasan diatas dapat diartikan bahwa **Variabel lokal** adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan. Aturan penulisan deklarasi variabel dengan cara eksplisit sebagai berikut :

Tabel 4.1. Deklarasi variabel secara eksplisit

Kata Kunci	Penggunaan
Static	Berlaku pada level prosedur
Dim	Berlaku pada level prosedur dan module
Private	Berlaku pada level module/form
Public	Berlaku pada level module dan aplikasi
Global	Berlaku pada level module dan aplikasi

Contoh deklarasi variabel secara *eksplisit* adalah :

```
Dim Nama As String
Private Nilai As Integer
Static Jumlah As Integer
Public Password As Variant
```

Deklarasi implisit dilakukan tanpa menggunakan kata kunci. Pada deklarasi implisit, sebuah variabel langsung digunakan disertai sebuah karakter khusus yang menandakan tipe data variabel tersebut.

Tabel 4.2. Karakter untuk deklarasi *variabel implisit*

Tipe Data	Karakter
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	\$

4.2. Pengertian Konstanta

Dalam pemrograman komputer, *konstanta* adalah pengenal (*identifier*) terkait nilai yang tidak dapat diubah (meskipun dalam beberapa kasus ini dapat dilakukan, misalnya menggunakan *self-modifying code*). Banyak pemrograman membuat perbedaan sintaksis eksplisit antara simbol konstanta dan variabel. Meskipun nilai konstanta yang ditentukan hanya sekali, sebuah konstanta dapat direferensikan berkali-kali dalam sebuah program. Konstanta sejenis dengan variabel yang nilainya tetap dan tidak dapat diubah selama program berjalan. Ada dua jenis konstanta yaitu konstanta intrinsik dan konstanta yang dibuat programmer. Konstanta intrinsik adalah konstanta yang sudah tersedia secara otomatis dalam Visual Basic 6.0. Contoh konstanta intrinsik adalah *vbYes*, *vbModal*, dan lain-lain. Programmer juga dapat membuat sebuah konstanta sendiri dengan cara mendeklarasikannya terlebih dahulu.

Konstanta merupakan variabel yang nilai di dalamnya selalu tetap. Konstanta diperlukan jika dibutuhkan sebuah nilai tetap yang harus muncul di banyak bagian dari rutin. Kemungkinan lain penggunaan konstanta adalah untuk mengingat angka yang sulit, atau nilai pasti dan tetap dari suatu rumus (formula) perhitungan matematika. Konsep ruang lingkup konstanta hampir sama dengan konsep ruang lingkup variabel. Dengan adanya konstanta, dapat meningkatkan keterbacaan kode program, menghemat memori, dan memudahkan perubahan global. Ketiga level tersebut hanya dibedakan berdasarkan kata kunci yang digunakan atau letak deklarasi konstanta. Aturan penulisan deklarasi konstanta adalah sbb:

[<KataKunci> Const <NamaKonstanta> [As <TipeData>] = <nilai>

Kata Kunci : Pilihan kata kunci yang dapat digunakan pada deklarasi konstanta adalah Private dan Public.

Nama Konstanta : Nama konstanta yang dideklarasikan

Tipe Data : Tipe data dari konstanta tersebut

Bagian yang diberi tanda kurung siku "[" dan "]" pada aturan penulisan di atas berarti boleh dituliskan, dan boleh juga tidak dituliskan. Contoh deklarasi konstanta :

```
Const Phi = 3.14
Public Const Jumlah = 500
Private Const Total = 15
Const Nama As String = "YAHFIZHAM"
```

4.3. Tipe-Tipe Data

Istilah *data* adalah suatu istilah majemuk yang berasal dari kata *datum* yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol, gambar, kata, kalimat, instruksi, angka, huruf, simbol, ide, konsep, kondisi, obyek, situasi, suara, suhu, kecepatan, detak jantung hewan dan manusia, getaran bumi dan lain-lain. Data kemudian diolah sehingga dapat dipahami secara jelas dan tepat sehingga dapat dimengerti. Pada pemrograman, yang dimaksud data adalah semua masukan (*input*) dari user ketika berinteraksi dengan aplikasi. Data yang diberikan user ini kemudian di olah menjadi informasi yang dimengerti oleh manusia.

Setiap variabel atau konstanta dalam visual basic 6.0 memiliki tipe data saat kita mendeklarasikannya sebagai suatu identifier. Tipe data suatu variabel digunakan untuk mengatur jenis data yang dibutuhkan untuk menyimpan nilai-nilai tertentu didalam memori komputer. Saat anda mendeklarasikan suatu variabel, kita juga dapat memberikan suatu tipe data pada variabel tersebut pada kondisi default, jika anda tidak menentukan tipe data untuk suatu variabel, maka variabel tersebut akan memiliki tipe data *variant*. Pada pemrograman visual basic 6.0, banyak jenis dari tipe data yang dapat digunakan. Jenis dan tipe data ini akan membedakan antara angka, huruf, simbol-simbol, karakter-karakter dll. Jenis dan tipe data juga berpengaruh terhadap ketelitian dan ketepatan perhitungan dalam aplikasi yang dibuat.

Pemberian tipe data pada suatu variabel dan konstanta penting untuk mendapatkan hasil perhitungan yang benar. Misalnya variabel A ditentukan memiliki *tipe data string* (tipe data ini menampung huruf, angka dan semua karakter) dan variabel B memiliki *tipe data integer* (tipe data khusus angka/numerik untuk bilangan bulat positif). Ketika kita ingin untuk melakukan perhitungan terhadap tipe data *string* dengan tipe data *integer*, maka akan mendapatkan kesalahan dikarenakan kedua jenis tipe data ini berbeda dan tidak dapat melakukan proses perhitungan. Yang benar adalah tipe data A harus *byte* dan tipe data B harus *byte, integer* dan semua kelompok berjenis tipe data numerik.

Mengenal jenis dan tipe data untuk pertimbangan dalam penggunaan alokasi memori. Semakin sedikit alokasi memori yang dipakai dalam suatu aplikasi maka akan berdampak terhadap kecepatan waktu pemrosesan. Terdapat perbedaan dalam pengalokasian memori untuk satu tipe data dengan tipe data yang lain. Tentu saja yang paling baik adalah penggunaan memorinya paling kecil. Misalnya jika kita memproses data nilai yang mempunyai rentang (*range*) 0 sampai 100 tanpa desimal maka yang paling tepat adalah menggunakan tipe data Byte (perlu memori 1 byte) bila dibandingkan menggunakan tipe data Integer (2 byte) atau Long (4 byte).

Setiap variabel dan atau konstanta harus memiliki kejelasan tentang jenis tipe data apa yang dapat disimpannya. Suatu variabel dan atau juga konstanta tidak dapat menyimpan semua jenis data, melainkan hanya dapat menyimpan satu jenis tipe data. Misalnya, untuk menyatakan jumlah buku di perpustakaan digunakan jenis tipe data angka (*numerik*). Untuk menyatakan kebenaran suatu pernyataan digunakan jenis tipe data *boolean*. Untuk menyatakan nama kendaraan digunakan tipe string. Untuk lebih jelas lagi lihat contoh berikut:

Contoh : Dim bilangan1 as Integer

Bilangan1 = variabel untuk menampung nilai

Integer = tipe jenis data angka (numerik) bulat positif

Tabel 4.3. Tipe-tipe data visual basic

Tipe	Ukuran bytes	Range
Integer	2 byte	-32768 s.d 32767
Long	4 byte	-2147483648 s.d 2147483647
Single	4 byte	Negatif : -3.402823E38 s.d -1.401298E-45 Positif : 1.401298E-45 s.d 3.402823E38
Double	8 bytes	Negatif: -1.79769313486232E308 s.d -4.94065645841247E-324 Positif : 4.94065645841247E-324 s.d 1.79769313486232E308
Currency	8 bytes	-922337203685477.5808 s.d 922337203685477.5807
String	1 byte per karakter	0 s.d 2 milyar karakter (95/97/NT) dan 0 s.d sekitar 65535 karakter (versi 3.1)
Byte	1 byte	0 s.d 255
Boolean	2 byte	True (benar) atau False (salah)
Date	8 byte	1 Januari 100 s.d 31 Desember 9999
Object	4 byte	Referensi Objek
Variant	16 byte – 1 byte per karakter	Null, Error, dan tipe seluruh tipe data lain, misalnya boolean, numerik, string, objek, array.

4.4. Operator-Operator

Operator di dalam pemrograman dapat diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol operator dapat berupa karakter ataupun kata khusus. Visual basic menyediakan operator-operator seperti aritmatika, komparasi dan logika, salah satu hal yang harus dipahami oleh programmer adalah tata urutan operasi dari masing-masing operator tersebut sehingga mampu membuat ekspresi yang akan menghasilkan kecepatan pemrosesan dan nilai yang benar.

Dalam pemrograman operator Matematika dapat diartikan sebagai simbol yang digunakan untuk melakukan operasi terhadap nilai data angka (*numerik*). Operator dalam pemrograman dapat berupa karakter ataupun berupa kata khusus yang melambangkan satu operasi tertentu, misalnya operator untuk operasi penjumlahan (+), pengurangan (-), pembagian (/), perkalian (*) dan sebagainya.

4.4.1. Operator Aritmatika

Operator Matematika yaitu operator yang digunakan untuk operasi matematis terhadap suatu nilai data angka (*numerik*). Ada beberapa operator aritmatika yang dapat digunakan dalam pemrograman antara lain :

Tabel 4.4. Operator aritmatika

Operasi	Simbol	Contoh
Pangkat	^	$4^2 = 16$
Perkalian	*	$4 * 2 = 8$
Pembagian	/	$5 / 2 = 2,5$
Pembagian Integer	\	$5 \setminus 2 = 2$
Sisa Bagi	Mod	$4 \text{ Mod } 2 = 0$
Penjumlahan	+	$4 + 2 = 6$
Pengurangan	-	$4 - 2 = 2$
Penggabungan String	&	$4 \& 2 = 42$

Pemangkatan atau eksponen banyak digunakan dalam operasi komputer, eksponen mempunyai operator *carre t(^)* dan sintaksnya adalah sebagai berikut :

Eksponen = bilangan1 ^ bilangan2

Contoh :

Eksponen = 5^2 hasilnya = 25

Eksponen = 3^2 hasilnya = 9

Eksponen = 2^{-2} hasilnya = 0,25

Perkalian digunakan untuk mengalikan dua buah bilangan bulat positif dan atau negatif atau lebih, simbolnya (*) dan sintaksnya dalam pemrograman adalah : Perkalian = bilangan1 * bilangan2

Contoh :

$$\text{Perkalian} = 2 * 2 \quad \text{hasilnya} = 4$$

$$\text{Perkalian} = 5 * 5 \quad \text{hasilnya} = 25$$

Pembagian digunakan untuk melakukan operasi pembagian terhadap bilangan, simbolnya (/) dan sintaksnya : Pembagian = bilangan1 / bilangan2

Contoh :

$$\text{Pembagian} = 4 / 2 \quad \text{hasilnya} = 2$$

$$\text{Pembagian} = 5 / 2 \quad \text{hasilnya} = 2,5$$

Pembagian Integer pada dasarnya sama dengan pembagian biasa, tetapi hasil yang diperoleh adalah bilangan integer atau bilangan bulat. Simbolnya (\) dan sintksnya : PembagianInt = bilangan1 \ bilangan2

Contoh :

$$\text{PembagianInt} = 4 \setminus 2 \quad \text{hasilnya} = 2$$

$$\text{PembagianInt} = 5 \setminus 2 \quad \text{hasilnya} = 2$$

Sisa bagi (Mod) digunakan untuk mendapatkan sisa dari pembagian dua buah bilangan, sintaksnya : Sisa = bilangan1 Mod bilangan2

Contoh :

$$\text{Sisa} = 4 \text{ Mod } 2 \quad \text{hasilnya} = 0$$

$$\text{Sisa} = 5 \text{ Mod } 2 \quad \text{hasilnya} = 1$$

Penjumlahan (+) digunakan untuk melakukan operasi penjumlahan pada bilangan, sintaksnya : Penjumlahan = bilangan1 + bilangan2

Contoh :

$$\text{Penjumlahan} = 4 + 2 \quad \text{hasilnya} = 6$$

$$\text{Penjumlahan} = 5 + 5 \quad \text{hasilnya} = 10$$

Pengurangan (-) digunakan untuk melakukan operasi pengurangan terhadap bilangan, sintaksnya : Pengurangan = bilangan1 – bilangan2

Contoh :

Pengurangan = 4 – 2 hasilnya = 2

Pengurangan = 5 – 2 hasilnya = 3

Penggabungan string (*concat*), simbolnya (&) digunakan untuk menggabungkan dua buah string. sintaksnya : String = string1 & string2

Contoh :

String = harry & potter hasilnya = harrypotter

String = 5 & 6 hasilnya = 56

Untuk menuliskan ekspresi dari rumus: $x = \sin(t)$ dengan t diketahui misalkan $t=0.5$ dapat dilakukan dengan: $t=0.5 \ x=\sin(t)$

Untuk memasukkan rumus $y = x^2 + 3x + 2$, dimana $x=2$ dapat dituliskan dengan: $x = 2$ dan $y = x^2 + 3*x + 2$

Untuk menuliskan rumus ABC dalam menyelesaikan persamaan kuadrat dengan nilai a, b dan c diketahui misalkan 1, 3 dan 2 adalah sebagai berikut:

$$x_{12} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

a=1: b=3: c=2

$$x1 = \frac{-b + (b^2 - 4*a*c)^{0.5}}{(2*a)}$$

$$x2 = \frac{-b - (b^2 - 4*a*c)^{0.5}}{(2*a)}$$

4.4.2. Operator Logika

Operator Logika adalah operator yang digunakan untuk membandingkan suatu perbandingan tertentu, simbol-simbol yang digunakan antara lain :

Tabel 4.5. Operator logika

Operasi	Simbol	Contoh
Dan	And	(4>1) And (1>4) = False
Atau	Or	(4>1) Or (1>4) = True
Tidak/Bukan	Not	Not (1>4) = True

Keterangan :

Dan = Operasi bernilai True jika kedua syarat benar

Or = Operasi bernilai True jika salah satu syarat benar

Not = Kebalikan dari, Misalkan jika A = True maka Not A = False

4.4.3. Operator Perbandingan

Operator Perbandingan digunakan untuk membandingkan nilai-nilai data dari dua variabel, antara lain :

Tabel 4.6. Operator pembandingan

Operasi	Simbol	Contoh	hasilnya
Samadengan	=	4=2	hasilnya False
Tidak Samadengan	<>	4<>2	hasilnya True
Lebih Kecil	<	4<2	hasilnya False
Lebih Besar	>	4>2	hasilnya True
Lebih Kecil Samadengan	<=	4<=2	hasilnya False
Lebih Besar Samadengan	>=	4>=2	hasilnya True

4.5. Fungsi-Fungsi Matematika

Dengan menggunakan fungsi-fungsi bawaan yang disediakan, waktu pembuatan sebuah program dapat disingkat. Fungsi-fungsi numerik atau rumus-rumus matematika merupakan fungsi-fungsi yang berkaitan dengan tipe data numerik. Fungsi-fungsi ini dapat dikelompokkan sebagai berikut :

a. Fungsi-fungsi konversi ke Integer

Ada tiga macam fungsi yang melakukan konversi ke tipe data integer yaitu seperti terlihat pada tabel 4.7.

Tabel 4.7. Fungsi konversi ke integer

Fungsi	Keterangan
CInt()	Pembulatan nilai desimal 0.5 dan lebih ke integer yang lebih dekat
Fix()	Pemotongan suatu nilai desimal menjadi integer
Int()	Pembulatan ke bawah suatu nilai desimal

Untuk bilangan positif, fungsi Fix() dan Int() memberikan perlakuan yang sama. Sebagai contoh, kedua pernyataan berikut ini mengembalikan nilai yang sama:

ans1 = Int(14.5)	'mengembalikan 14
ans2 = Fix(14.5)	'mengembalikan 14

Namun fungsi Fix() dan Int() memberikan perlakuan berbeda pada bilangan negatif. Kedua pernyataan berikut memberikan pengembalian yang berbeda :

ans1 = Int(-14.5)	'mengembalikan -15
ans2 = Fix(-14.5)	'mengembalikan -14

Sedangkan fungsi CInt() mengembalikan pembulatan angka yang terdekat. Berikut ini contoh penggunaan CInt() :

ans1 = CInt(14.1)	'mengembalikan 14
ans2 = CInt(14.5)	'mengembalikan 14
ans3 = CInt(14.6)	'mengembalikan 15
ans2 = CInt(-14.5)	'mengembalikan -14
ans3 = CInt(-14.6)	'mengembalikan -15

Tabel 4.8. Fungsi-fungsi konversi ke tipe data

Fungsi	Keterangan
CCur()	Mengkonversi argumen menjadi tipe data Currency
CDbl()	Mengkonversi argumen menjadi tipe data double (presisi ganda)
CLng()	Mengkonversi argumen menjadi tipe data Long Integer
CSng()	Mengkonversi argumen menjadi tipe data Single
CStr()	Mengkonversi argumen menjadi tipe data String
CVar()	Mengkonversi argumen menjadi tipe data Variant

Sebagai contoh misalkan sebuah data yang merupakan hasil bagi (1 / 7).

lblNilai = CSng(1/7) 'ditampilkan 0.1428571 lblNilai = CDbl(1/7) 'ditampilkan 0.142857142857143
--

Contoh yang lain, ketika Anda akan menghitung 2/0.5 :

'Contoh 1: MsgBox 2 / CInt(0.5) '<-- error Division by zero 'Contoh 2: MsgBox 2 / Val(0.5) '<-- error Division by zero 'Contoh 3: MsgBox 2 / CDbl(0.5) '<-- benar menghasilkan 4

Mengapa contoh 1 dan 2 menghasilkan error **Divison by zero**? Karena bilangan pecahan di kedua contoh tersebut dikonversi ke bilangan bulat, sehingga nilai 0.5 (nol koma lima) dianggap sebagai angka nol saja. Semua angka yang dibagi dengan nol maka akan menghasilkan error.

b. Fungsi–Fungsi Matematika

Fungsi-fungsi yang berkaitan dengan matematika seperti terlihat pada tabel 4.9, dibawah ini

Tabel 4.9. Fungsi-fungsi matematika

Fungsi	Kegunaan
<i>Abs(n)</i>	Menghasilkan nilai absolut dari n
<i>Atn(n)</i>	Menghasilkan arkus tangen, dalam radian, dari n
<i>Cos(n)</i>	Menghasilkan kosinus sebuah sudut n. Sudut n dinyatakan dalam radian
<i>Exp(n)</i>	Menghasilkan konstanta e pangkat n
<i>Rnd(n)</i>	Menghasilkan bilangan acak antara 0 dan 1
<i>Sgn(n)</i>	Menghasilkan -1 jika n lebih kecil dari nol, 0 jika n sama dengan nol, dan +1 jika n lebih besar dari nol
<i>Sin(n)</i>	Menghasilkan sinus dari sudut n. Sudut n dinyatakan dalam radian
<i>Sqr(n)</i>	Menghasilkan akar kuadrat dari n
<i>Str(n)</i>	Mengubah nilai numerik menjadi teks
<i>Tan(n)</i>	Menghasilkan tangen dari sudut n. Sudut n dinyatakan dalam radian
<i>Val(n)</i>	Mengubah nilai teks menjadi bilangan

c. Fungsi–Fungsi String

Selain fungsi-fungsi numerik, visual basic juga mempunyai sejumlah fungsi-fungsi String diantaranya seperti pada tabel 4.10, dibawah ini.

Tabel 4.10. Fungsi-fungsi string

Fungsi	Keterangan
<i>LCase()</i>	Mengembalikan argumen string sebagai tipe data string huruf kecil
<i>UCase()</i>	Mengembalikan argumen string sebagai tipe data string huruf besar
<i>Val()</i>	Mengembalikan nilai number dari argumen string
<i>Len()</i>	Mengembalikan nilai number yang menunjukkan panjang string

Substring merupakan bagian dari suatu string. Untuk memperoleh substring dari suatu string dapat digunakan 3 fungsi berikut :

➤ *Left(StringVal, length)*

Fungsi ini mengembalikan nilai string dari sebelah kiri sebanyak *length* karakter .

➤ *Right(StringVal, length)*

Fungsi ini mengembalikan nilai string dari sebelah kanan sebanyak *length* karakter .

➤ *Mid(StringVal, startVal, length)*

Fungsi ini mengembalikan nilai string dari *startVal* sebanyak *length* karakter, contoh :

```
Hurufkecil = LCase("yahfizham")
hurufbesar = UCase("YAHFIZHAM")
Title = "Fakultas Ilmu Tarbiyah dan Keguruan"
lTitle = Left(Title, 3)
rTitle = Right(Title, 5)
mTitle = Mid(Title, 3, 8)
Panjang = Len(Title)
```

d. Fungsi–Fungsi Tanggal dan Waktu

Jika dalam suatu aplikasi diperlukan pengambilan tanggal atau waktu dari sistem, visual basic menyediakan fungsi–fungsi *Now()*, *Date()*, dan *Time()*.

Tabel 4.11. Fungsi-fungsi tanggal dan waktu

Fungsi	Keterangan
Now()	Mengembalikan tanggal dan waktu sistem
Date()	Mengembalikan tanggal dari sistem
Time()	Mengembalikan waktu dari sistem
DateDiff	Mendapatkan selisih dari dua buah tanggal

Untuk mencari selisih dari dua buah tanggal tidaklah sulit, karena dalam visual basic telah disediakan fasilitas untuk melakukan hal tersebut dengan menggunakan fungsi "*DateDiff*", yaitu sebuah fungsi yang digunakan untuk mendapatkan selisih dari dua buah tanggal. Dari fungsi *DateDiff* kita bisa mendapatkan selisih hari, bulan dan tahun dari dua buah tanggal. Cara penggunaan dari fungsi ini adalah sebagai berikut:

```
Dim hari, bulan, tahun
'Untuk mencari selisih hari
hari=DateTime.DateDiff("d",CDate(Text1.Text),CDate(Text2.Text))
'Untuk mencari selisih bulan
bulan = DateTime.DateDiff("m",CDate(Text1.Text),CDate(Text2.Text))
'Untuk mencari selisih tahun
tahun = DateTime.DateDiff("yyyy",CDate(Text1.Text),CDate(Text2.Text))
```

Selain fungsi-fungsi diatas terdapat sebuah fungsi yaitu *format()* yang berkaitan dengan fungsi-fungsi Tanggal dan waktu.

```
Format(Ekpresi[,format[,firstdayofweek[, firstweekofyear]]])
```

Rangkuman

Variabel adalah deklarasi sesuatu yang memiliki variasi (berubah-ubah) dan tidak tetap dari suatu nilai. Variabel digunakan untuk menyimpan nilai sesuatu yang dimiliki program aplikasi yang dibentuk oleh programmer sendiri. Nilai yang ditampilkan atau disimpan oleh suatu variabel dapat berubah-ubah sesuai keinginan. Variabel merupakan tempat untuk menyimpan nilai sementara dari suatu perhitungan. Dalam pemrograman komputer, *konstanta* adalah pengenal (*identifier*) terkait nilai yang tidak dapat diubah (meskipun dalam beberapa kasus ini dapat dielakkan, misalnya menggunakan *self-modifying code*). Banyak pemrograman membuat perbedaan sintaksis eksplisit antara simbol konstanta dan variabel. Meskipun nilai konstanta yang ditentukan hanya sekali, sebuah konstanta dapat direferensikan berkali-kali dalam sebuah program. **Operator** di dalam pemrograman dapat diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol operator dapat berupa karakter ataupun kata khusus. Visual basic menyediakan operator-operator seperti aritmatika, komparasi dan logika, salah satu hal yang harus dipahami oleh programmer adalah tata urutan operasi dari masing-masing operator tersebut sehingga mampu membuat ekspresi yang akan menghasilkan kecepatan pemrosesan dan nilai yang benar.

**Latihan
dan Soal**

1. Kerjakanlah latihan ke-5 berikut :

- a) Buka **project** baru dan **form**, sesuaikan antarmukanya seperti gambar dibawah ini



Gambar 4.3. Latihan membuat teks berjalan

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	frmTeksJalan
		BackColor	&H00C0FFFF&
		Caption	"[Asik Kami Sedang Belajar VB 6.0]"
		Height	3360
		Width	5210
		StartPosition	2 – Center Screen
		WindowState	0 –Normal
2	Timer1	(Name)	Timer1
		Interval	400

c) *Source code* nya :

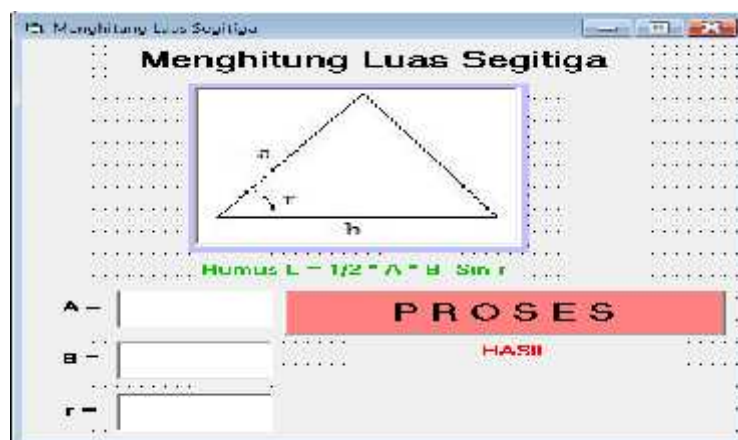
```
Private Sub Form_Load()
    frmTeksJalan.Caption = "[ Asik Kami Sedang Belajar VB 6.0 ]"
End Sub
```

```
Private Sub Timer1_Timer()
    a = Left(frmTeksJalan.Caption, 1)
    b = Len(frmTeksJalan.Caption)
    c = Right(frmTeksJalan.Caption, b - 1)
    frmTeksJalan.Caption = c + a
End Sub
```

d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat05.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat05.frm “

2. Kerjakanlah latihan ke-6 berikut :

a) Buka *project* baru dan *form*, sesuaikan antarmukanya seperti gambar dibawah ini



Gambar 4.4. Menghitung luas segitiga

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	FrmLuasSegitiga
		BackColor	&H800000F&
		Caption	Menghitung Luas Segitiga
		Height	7116
		Width	7452
		StartPosition	2 - Center Screen
		WindowState	0 -Normal
2	Label1	(Name)	Label1
		Alignment	2 - Center
		Caption	MENGHITUNG LUAS SEGITIGA
		Font	Ms Sans Serif - Bold - 19
		ForeColor	&H80000012&
3	Image1	(Name)	Image1
		Picture	Cari gambar segitiga atau buat sendiri dari objek Shape dan Line
		Height	2736
		Width	3444
4	Label2	(Name)	Label2
		Alignment	2 - Center
		Caption	Rumus $L = 1/2 * A * B * \sin r$
		Font	Ms Sans Serif - Bold - 11
		ForeColor	&H0000C000&
5	Label3	(Name)	Label3
		Alignment	2 - Center
		Caption	A =
		Font	Ms Sans Serif - Bold - 11
		ForeColor	&H80000012&
6	Label4	(Name)	Label4
		Alignment	2 - Center
		Caption	B =
		Font	Ms Sans Serif - Bold - 11
		ForeColor	&H80000012&
7	Label5	(Name)	Label5
		Alignment	2 - Center
		Caption	C =
		Font	Ms Sans Serif - Bold - 11
		ForeColor	&H80000012&
8	Label6	(Name)	Label6
		Alignment	2 - Center
		Caption	H A S I L
		Font	Ms Sans Serif - Bold - 11
		ForeColor	&H000000FF&
9	Label7	(Name)	Lbhasil
		Alignment	2 - Center
		Caption	Tulisan " Label7" dihapus (jadi kosong)
		Font	Ms Sans Serif - Bold - 19
		ForeColor	&H80000012&

10	TextBox1	(Name)	txtA
		Alignment	2 - Center
		Font	Ms Sans Serif – Bold – 14
		Text	Tulisan “ Text1” dihapus (jadi kosong)
11	TextBox2	(Name)	txtB
		Alignment	2 - Center
		Font	Ms Sans Serif – Bold – 14
		Text	Tulisan “ Text1” dihapus (jadi kosong)
12	TextBox3	(Name)	txtR
		Alignment	2 - Center
		Font	Ms Sans Serif – Bold – 14
		Text	Tulisan “ Text1” dihapus (jadi kosong)
13	CommandButton1	(Name)	cmdproses
		BackColor	&H008080FF&
		Caption	P R O S E S
		Font	Ms Sans Serif – Bold – 19
		Height	732
		Style	1 Graphical
		Width	4452

c) Private Sub cmdproses_Click()

A = Val(txtA.Text)

B = Val(txtB.Text)

R = Val(txtR.Text)

hasil = $0.5 * A * B * \sin(R * (3.14 / 180))$

lblhasil.Caption = Round(hasil, 2)

End Sub

d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat06.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat06.frm “

3. Kerjakanlah latihan ke-7 berikut :

a) Buka *project* baru dan *form*, sesuaikan antarmukanya seperti gambar dibawah ini



Gambar 4.5. Contoh program operator

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	Frmoperator
		BackColor	&H800000F&
		Caption	PROGRAM ARITMATIKA DAN OPERATOR SEDERHANA
		Height	8124
		Width	7584
		WindowState	0 –Normal
2	Label1	(Name)	Label1
		Alignment	2 – Center
		Caption	Bilangan 1
		Font	Ms Sans Serif – Bold – 19
3	Label2	ForeColor	&H800000F&
		(Name)	Label2
		Alignment	2 – Center
		Caption	Bilangan 2
4	Label3	Font	Ms Sans Serif – Bold – 19
		ForeColor	&H800000F&
		(Name)	Label3
		Alignment	2 – Center
5	Label4	Caption	H A S I L
		Font	Ms Sans Serif – Bold – 19
		ForeColor	&H800000F&
		(Name)	lblHasil
		Alignment	2 – Center
6	TextBox1	Caption	Tulisan “ Label4” dihapus (jadi kosong)
		BackColor	&H00E0E0E0&
		Font	Ms Sans Serif – Bold – 19
		ForeColor	&H80000012&
7	TextBox2	(Name)	Txtvar1
		Alignment	2 – Center
		Font	Ms Sans Serif – Bold – 19
		Text	Tulisan “ Text1” dihapus (jadi kosong)
8	Frame1	(Name)	Txtvar2
		Alignment	2 – Center
		Font	Ms Sans Serif – Bold – 19
		Text	Tulisan “ Text1” dihapus (jadi kosong)
9	Frame2	(Name)	Frame1
		BackColor	&H00C0FFFF&
		Caption	Operator Aritmatika
10	OptionButton1	(Name)	Frame2
		BackColor	&H00C0FFC0&
		Caption	Operator Perbandingan
		(Name)	Option1
11	OptionButton2	BackColor	&H000000FF&
		Caption	+
		Font	Ms Sans Serif – Bold – 19
		ForeColor	&H80000012&
12	CommandButton1	(Name)	Option2
		BackColor	&H000000FF&
		Caption	>
		Font	Ms Sans Serif – Bold – 19
12	CommandButton1	ForeColor	&H80000012&
		(Name)	Cmdkeluar
		BackColor	&H8000000F&
		Caption	K E L U A R
		Font	Ms Sans Serif – Bold – 19
		Height	852
		Width	6252

```

c) Source Code nya
Dim var, var2 As Double
Dim hasil As Single
Private Sub CmdKeluar_Click()
    End
    End Sub
Private Sub Option1_Click()
    var1 = Val(txtVar1)
    var2 = Val(txtVar2)
    hasil = var1 + var2
    lblHasil.Caption = hasil
End Sub
Private Sub Option2_Click()
    var1 = Val(txtVar1)
    var2 = Val(txtVar2)
    lblHasil.Caption = var1 > var2
End Sub

```

- d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat07.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat07.frm “

Soal

1. Buatlah satu **project** dan **form** untuk menghitung nilai persamaan $y=x^2+3x+2$ dengan nilai X sebagai masukan (input) !
2. Buatlah satu **project** dan **form** untuk menghitung luas persegi panjang !
3. Buatlah satu **project** dan **form** untuk menghitung layang-layang !
4. Buatlah satu **project** dan **form** untuk menghitung luas lingkaran !
5. Pada Latihan 3 tambahkan lagi operator aritmatika dan perbandingan : kali (*), bagi (/), kurang (-), bagi bilangan pecahan (\), modulo (mod), pangkat (^), concatenate (&), lebih besar (>), lebih besar sama dengan (>=), lebih kecil (<), lebih kecil sama dengan (<=), sama dengan (=) dan tidak sama dengan (<>)

Tindak Lanjut (Feed Back)

Skor	Kriteria
100-80	Mampu dan dapat menyelesaikan soal-soal yang diberikan: 1. Bekerja dengan variabel dan konstanta 2. Bekerja dengan operator-operator 3. Bekerja dengan fungsi-fungsi matematika
70-60	Hanya mampu menjawab 4 soal
50 ke bawah	Tidak mampu sama sekali dan harus mengulang kegiatan belajar 4 secara keseluruhan



Kegiatan Belajar 5.

BAB
VPERNYATAAN (*STATEMENT*)
PENCABANGAN DAN PERULANGAN1. Prosedur (*Procedure*)

Langkah dalam pembuatan suatu aplikasi adalah mendapatkan suatu masukan (*input*) dari pengguna aplikasi (*user*) dan kemudian akan menampilkan atau memampangkan hasil dari proses pengolahan data tadi sehingga menjadi informasi yang lebih dapat dipahami, bermakna dan berguna. Terdapat banyak cara untuk mendapatkan masukan (*input*), baik itu dari *keyboard* maupun dari *hard ware* lain misalnya saja mouse.

Pada kenyataannya sering kali program yang harus ditulis cukup panjang, sehingga kesalahan yang mungkin dibuat oleh seorang *programmer* semakin besar. Untuk mengatasi masalah tersebut dapat dilakukan dengan memecah program tersebut menjadi bagian-bagian kecil (rutin) tetapi tetap logis. Rutin-rutin kecil tersebut akan membuat penelusuran dan perawatan program menjadi lebih mudah dan terstruktur. Rutin-rutin kecil tersebut sering disebut dengan nama prosedur. Pemograman visual basic memuat 2 macam prosedur yaitu :

- a. Prosedur umum (*general purpose procedure*) merupakan prosedur yang ditemukan di dalam daftar *Drop Down* pada jendela kode.
- b. Prosedur Kejadian (*Event procedure*), prosedur yang berisi kode yang dijalankan ketika suatu kejadian dari kontrol di bangkitkan.

Sedangkan prosedur-prosedur yang ditambahkan dalam sebuah program tersebut disebut *sub program*. Terdapat 2 jenis sub program pada pemograman visual basic yaitu prosedur sub rutin (*sub routine procedures*) dan prosedur fungsi (*function procedures*). Subrutin merupakan prosedur umum (*general porpuse procedure*) yang ditulis dan ditambahkan dalam program. Format penulisan sebuah subrutin adalah sebagai berikut:

Sub nama_subRutin [(argumen)]

.

.

.

End Sub

Setiap kali prosedur dipanggil, maka pernyataan diantara Sub dan End Sub akan dijalankan. Argumen pada prosedur adalah nilai yang akan dilewatkan saat pemanggilan prosedur, misalnya :

```
Sub TotalPembelian()
  ' Inisialisasi variabel
  Dim Total As Currency
  Dim Disc As Single

  'mendefinisikan nilai variabel
  Total = txtTotal.Text
  Disc = 0.2

  lblTotal.Caption = Total - Total * Disc
End Sub
```

Sedangkan untuk memanggil sebuah subrutin digunakan format sebagai berikut :

```
[Call] SubName [(Argumen list)]
```

Untuk memanggil subrutin dari contoh diatas dapat digunakan kode sebagai berikut:

```
Call TotalBelanja() atau TotalBelanja
```

Untuk memanggil suatu sub rutin dapat digunakan perintah *Call* diikuti dengan nama sub rutin diikuti dan tanda kurung atau langsung memanggil nama sub rutusnya saja. Sub rutin dan Fungsi sebenarnya mirip yaitu prosedur umum (*general purpose procedure*) yang ditulis dan ditambahkan dalam program. Bedanya Fungsi mengembalikan nilai sedangkan Subrutin tidak. Format penulisan sebuah Fungsi adalah sebagai berikut.

```
Function nama_Fungsi [(argumen)] as [ReturnValType]
.
.
.
End Function
```

Contoh :

```
Function Total_Pembelian(Total As Currency)
  ' Inisialisasi variabel
  Dim Disc As Single
  ' Mendefinisikan nilai variabel
  Disc = 0.2
  Total_Pembelian = Total - Total * Disc
End Function
```

Pada contoh di atas merupakan fungsi TotalBelanja yang dimodifikasi. Total tidak lagi diambil dari nilai textbox (txtTotal), melainkan dikirim melalui suatu argumen. Argumen yang dikirimkan melalui suatu subrutin atau fungsi dapat digunakan langsung seperti sebuah variabel. Untuk memanggil fungsi dari contoh diatas dapat digunakan kode sebagai berikut:

'Memasukkan nilai Total sebesar 200

TotalBelanja (200)

Penulisan sebuah modul dapat dilakukan pada sebuah file berekstensi **.Bas**. File tersebut dapat dibuat melalui menu *Project-Add Module*. Secara default visual basic memberikan nama *Module1*. Pada *module*, kita dapat mendeklarasikan variabel dengan awalan *DIM*, *PRIVATE* maupun *PUBLIC*, dimana awalan *DIM* dan *PRIVATE* membentuk variabel modul level (hanya berlaku didalam pemakaian variabel bersangkutan), sedangkan awalan *PUBLIC* akan membentuk variabel global yang akan berfungsi bagi keseluruhan program, misalnya :

Dim A As Integer

Private B As Integer

Public C As Integer

Pemanggilan dapat dilakukan dengan menyebut nama module terlebih dahulu diikuti dengan tanda titik kemudian nama subrutin atau langsung nama subrutin. Sub rutin dapat ditulis pada *module*. Sub rutin tersebut dapat dipanggil kapan saja dari aplikasi yang sama dengan format sebagai berikut: [nama_module.] nama_subrutin

Contoh : *Module1.TotalBelanja* atau *TotalBelanja*

2. Struktur Kendali (Pencabangan)

Struktur kendali (pencabangan) adalah suatu proses pengujian untuk mengambil suatu keputusan apakah suatu baris atau blok instruksi akan diproses atau tidak. Pengujian kondisi ini dilakukan untuk memilih salah satu dari beberapa alternatif yang tersedia. Struktur kendali (pencabangan) pada program dibutuhkan agar dapat memberikan suatu kondisi tertentu atau kelompok kondisi tertentu.

Kondisi tertentu dapat saja berupa pemilihan pengendalian alur suatu program, sehingga program tersebut dapat berjalan sesuai dengan maksud dan tujuan yang ingin dicapai. Struktur kendali berperan dalam proses eksekusi pembuatan keputusan dari banyak kondisi. Struktur kendali (pencabangan) merupakan alur program suatu pembuatan keputusan. Pemilihan keputusan memuat suatu kondisi-kondisi yang apabila dikerjakan akan memberikan hasil A dan apabila tidak dikerjakan akan memberikan hasil B. Seorang programmer dapat membuat struktur kendali jalannya program dengan cara memberikan pencabangan dan atau perulangan.

Struktur kendali program dibuat untuk dapat membuat suatu keputusan yaitu mengulang sebagian rutin karena tidak terpenuhinya suatu kondisi dan melanjutkan suatu pernyataan jika kondisi terpenuhi dan memilih sebuah pilihan dari beberapa alternatif jika suatu kondisi terpenuhi. Pada pemrograman visual basic 6.0, struktur kendali (pencabangan) hanya ada tiga, yakni :

- a) *If.....Then.....*
- b) *If.....Then.....Elseif*
- c) *Select Case*

Gunakan struktur kendali keputusan (pencabangan) ***If...Then....*** untuk menjalankan jika satu kondisi maka satu pernyataan (pernyataan-pernyataan). Bentuk struktur kendali pencabangannya adalah

If kondisi Then pernyataan
End If
 atau

Keterangan:
 Jika <kondisi> bernilai Benar/True/
 Terpenuhi maka <pernyataan> akan
 dikerjakan

If kondisi Then pernyataan-pernyataan
End If

Gunakan struktur kendali keputusan (pencabangan) ***If...Then....Else*** untuk menjalankan satu kondisi dan dua pernyataan (pernyataan 1 dan pernyataan 2). Bentuk struktur kendali pencabangannya adalah

If kondisi Then pernyataan 1
Else
Pernyataan 2
End If

Keterangan :
 Jika <kondisi> bernilai Benar/True/
 Terpenuhi maka <pernyataan 1> akan
 dikerjakan, tetapi jika <kondisi> bernilai
 Salah/False/Tidak Terpenuhi maka
 <pernyataan 2> yang akan dikerjakan.

Gunakan struktur kendali keputusan (pencabangan) *If...Then....Elseif* untuk menjalankan kondisi 1 dan kondisi 2 serta 3 pernyataan (pernyataan 1 dan pernyataan 2 dan pernyataan 3). Bentuk struktur kendali pencabangannya adalah

If kondisi 1 Then

pernyataan 1

Elseif kondisi 2 Then

Pernyataan 2

Else

Pernyataan 3

End If

Keterangan :

Jika <kondisi 1> bernilai True maka <pernyataan 1> akan dikerjakan, lanjut jika <kondisi 2> bernilai True maka <kondisi 2> akan dikerjakan, tetapi jika <kondisi 1> dan <kondisi 2> bernilai False maka <pernyataan 3> yang akan dieksekusi/dikerjakan.

Gunakan struktur kendali keputusan (pencabangan) *If...Then....Elseif...Elseif...* untuk menjalankan jika lebih dari tiga kondisi (kondisi 1 dan kondisi 2 dan kondisi 3 dll) maka lebih dari tiga pernyataan (pernyataan 1 dan pernyataan 2 dan pernyataan 3 dll). Bentuk struktur kendali pencabangannya adalah

If kondisi 1 Then

pernyataan 1

Elseif kondisi 2 Then

Pernyataan 2

Elseif kondisi 3 Then

Pernyataan 3

Else

Pernyataan 4

End If

Keterangan :

Jika <kondisi 1> bernilai True maka <pernyataan 1> akan dikerjakan, lanjut jika <kondisi 2> bernilai True maka <kondisi 2> akan dikerjakan, lanjut lagi jika <kondisi 3> bernilai True maka <kondisi 3> akan dikerjakan, tetapi jika <kondisi 1> dan <kondisi 2> dan <kondisi 3> bernilai False maka <pernyataan 4> yang akan dieksekusi/dikerjakan.

Cara lain untuk menangani pengambilan keputusan dalam suatu program adalah dengan menggunakan pernyataan (*statement*) *select...case*. Pernyataan *select...case* mirip dengan pernyataan *If...Then...Elseif*, pada dasarnya prinsip kerja dan kegunaan keduanya sama, namun pemakaian *select...case* akan mempermudah menulis dan mengelola *source code* karena akan lebih sedikit dan lebih sederhana. Ketika pernyataan *If...Then...Elseif* memiliki banyak pernyataan, pernyataan menjadi sulit dibaca dan diikuti.

Struktur *select...case* akan menguji pernyataan tunggal yang dievaluasi sekali pada bagian atas dari struktur. Hasil dari pengujian kemudian dibandingkan dengan beberapa nilai dan jika salah satu ada yang cocok, blok pernyataan yang berhubungan akan dijalankan. Bentuk struktur kendali pencabangannya adalah

```
Select Case Nama Variabel
    Case kondisi 1
        Pernyataan 1
    Case kondisi 2
        Pernyataan 2
    Case kondisi ke-N
        .....
    Case Else
        Pernyataan ke-N
End Select
```

3. Pernyataan Perulangan (*Looping*)

Struktur perulangan (*looping*) digunakan untuk menjalankan satu atau banyak baris perintah atau pernyataan secara berulang-ulang. Pemograman *visual basic* mendukung proses perulangan suatu pernyataan. Perulangan harus mempunyai atau dapat menentukan nilai awal dan nilai akhir. Dalam merancang perulangan setidaknya harus mengetahui 3 komponen, yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti.

Fungsi perulangan adalah untuk dapat menyimpan suatu data secara berulang-ulang, menampilkan data secara berulang dan membaca data secara berulang-ulang. Dari sisi programmer, tujuan perulangan adalah efisiensi pemakaian alamat memori komputer karena tidak mengulang-ulang baris perintah (*source code*). Dari sisi user, akan memudahkan proses masukan (*input*) data sehingga pekerjaan berjalan secara lebih cepat dan lebih baik. Pada pemograman visual basic 6.0, struktur perulangan (*looping*) adalah sebagai berikut :

a) *Do While...Loop*

Do While...Loop memungkinkan kita mengeksekusi serangkaian pernyataan berulang-ulang selama kondisi pengujian tetap benar. Visual basic akan mengevaluasi ekspresi dan jika bernilai benar, pernyataan akan dijalankan.

Ketika sampai pada bagian akhir pernyataan, ekspresi akan diuji lagi dan jika bernilai benar maka pernyataan akan diulang. Namun jika ekspresi bernilai salah maka program akan dilanjutkan ke pernyataan setelah loop.

Ada empat variasi ***Do While...Loop***, dua yang pertama memiliki model yang sama. Pengulangan akan dilakukan ketika kondisi bernilai benar atau sampai kondisinya bernilai benar. Kedua variasi menggunakan kata kunci ***While*** dan ***Until*** untuk menunjukkan berapa lama kondisi menjadi benar. Struktur perulangan (*looping*) nya adalah sebagai berikut:

Do While Kondisi

Pernyataan

Loop

Untuk menjalankan pernyataan ***Until*** dengan kondisi benar struktur perulangan (*looping*) nya adalah sebagai berikut:

Do Until Kondisi

Pernyataan

Loop

Ketika visual basic mengkompilasi pengulangan di atas, yang pertama dilakukan adalah menguji kondisi, jika kondisi bernilai salah pernyataan pengulangan ***Do While*** akan dilewati tetapi pengulangan ***Do Until*** akan dijalankan. Ketika sampai pernyataan ***Loop***, visual basic akan menguji ekspresi lagi dan mengulang pernyataan dari pengulangan ***Do While*** jika ekspresi bernilai salah dan pengulangan ***Do until*** jika ekspresi bernilai benar. Jika ***While*** akan berulang selama pengujian ekspresinya benar, sedangkan ***Until*** akan berulang selama kondisi ekspresinya salah.

Variasi lain dari pengulangan ***Do Loop*** adalah menjalankan pernyataan dulu dan menguji kondisi setelah pernyataan dijalankan. Struktur perulangan (*looping*) nya adalah sebagai berikut:

Do

Pernyataan

Loop While Kondisi

Atau

Do

Pernyataan

LoopUntil Kondisi

b) While...End While

Pengulangan While...End While akan menjalankan suatu blok pernyataan selama kondisi bernilai benar. Struktur perulangan (*looping*) nya adalah sebagai berikut:

While Kondisi

Pernyataan

End While

Jika kondisi bernilai benar, semua blok pernyataan akan dijalankan dan ketika mencapai **End While**, kontrol akan dikembalikan ke pernyataan **While** dan menguji kembali kondisi. Jika masih benar prosesnya akan diulang. Jika kondisinya bernilai salah, program akan dilanjutkan pada pernyataan setelah **End While**.

c) For... Next

Pada pengulangan *For...Next*, kita harus mengetahui berapa kali pernyataan dalam pengulangan akan dijalankan. Pengulangan *For...Next* menggunakan suatu variabel (disebut *counter*) yang nilainya akan bertambah atau berkurang selama proses tiap pengulangan dari perulangan. Struktur perulangan (*looping*) nya adalah sebagai berikut:

For Counter = awal **To** Akhir [**Step** penambahan/pengurangan]

Pernyataan

Next [counter]

Argumen *counter*, diawal dan diakhir serta penambahan/pengurangan semuanya adalah *numerik*. Pengulangan akan terus dijalankan sampai mencapai nilai *counter*.

Rangkuman

Seorang programmer dapat membuat struktur kendali jalannya suatu program (aplikasi) dengan cara memberikan pencabangan dan atau perulangan. Struktur kendali (pencabangan) adalah suatu proses pengujian untuk mengambil suatu keputusan apakah suatu baris atau blok instruksi akan diproses atau tidak. Struktur kendali (pencabangan) merupakan alur program suatu pembuatan keputusan. Pemilihan keputusan memuat suatu kondisi-kondisi yang apabila dikerjakan akan memberikan hasil A dan apabila tidak dikerjakan akan memberikan hasil B. Struktur perulangan (*looping*) digunakan untuk menjalankan satu atau banyak baris perintah atau pernyataan secara berulang-ulang. Dalam merancang perulangan setidaknya ada beberapa persyaratan yang harus dipenuhi yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti.

Latihan
dan Soal

1. Kerjakanlah latihan ke-8 berikut :

a) Buka *project* baru dan *form*, sesuaikan antarmukanya seperti gambar dibawah ini

The image shows a Java Swing window titled "CONTOH PROGRAM STRUKTUR KENDALI PENCABANGAN". The window contains a form with the following elements:

- A title bar with the text "PROGRAM STRUKTUR KENDALI".
- A text field labeled "ISIKAN TOTAL NILAI ANGKA" followed by a yellow button labeled "PROSES".
- A green section containing the text "KONVERSI KE NILAI HURUF" followed by an orange text field.
- The text "KETERANGAN" followed by another orange text field.
- A yellow button labeled "HAPUS" and a yellow button labeled "BOBOT".
- A yellow button labeled "KELUAR" at the bottom.

Gambar 5.1. Antarmuka program struktur kendali *if...than...else*

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	FrmPencabangan
		BackColor	&H00FFFFC0&
		Caption	CONTOH PROGRAM STRUKTUR KENDALI PENCABANGAN
		Height	7656
		Width	8244
		StartPosition	2 – Center Screen
		WindowState	0 –Normal
2	Shape1	(Nama)	Shape1
		BorderWidth	6
		Shape	4 – Rounded Rectangle
3	Label1	(Name)	Label1
		Alignment	2 – Center
		Caption	PROGRAM STRUKTUR KENDALI
		Font	Arial Narrow – Bold – 22
		ForeColor	&H80000012&
4	Label2	(Name)	Label2
		Alignment	2 – Center
		Caption	ISIKAN TOTAL NILAI ANGKA
		Font	MS Sans Serif – Reguler – 14
		ForeColor	&H80000012&
5	TextBox1	(Name)	TxtNilai
		Alignment	2 – Center
		BackColor	&H80000005&
		Font	Arial – Reguler – 22
		Text	Tulisan “ Text1 ” dihapus (jadi kosong)
6	CommandButton1	(Name)	CmdProses
		BackColor	&H0080FFFF&
		Caption	&PROSES
		Font	MS Sans Serif – Reguler – 19
		Style	1 – Graphical
7	Shape2	(Nama)	Shape2
		BorderWidth	1
		BorderColor	&H00FF0000&
		Shape	0 – Rectangle
8	Label3	(Name)	Label3
		Alignment	2 – Center
		Caption	KONVERSI KE NILAI HURUF
		Font	Candara – Bold – 19
		ForeColor	&H80000012&
9	Label4	(Name)	IblNilaiKonversi
		Alignment	2 – Center
		BackColor	&H00C0E0FF&
		Caption	Tulisan “ Label4 ” dihapus (jadi kosong)
		Font	Arial – Reguler – 20
10	Label5	(Name)	Label5
		Alignment	2 – Center
		Caption	KETERANGAN
		Font	Candara – Bold – 19
		ForeColor	&H80000012&
11	Label6	(Name)	IblKet
		Alignment	2 – Center
		BackColor	&H00C0E0FF&
		Caption	Tulisan “ Label6 ” dihapus (jadi kosong)
		Font	Arial – Reguler – 20
		ForeColor	&H80000012&

12	CommandButton2	(Name)	CmdHapus
		BackColor	&H0080FFFF&
		Caption	&H A P U S
		Font	MS Sans Serif – Reguler – 19
		Style	1 – Graphical
13	Label7	(Name)	Label7
		Alignment	2 – Center
		Caption	BOBOT
		Font	Candara – Bold – 19
		ForeColor	&H80000012&
14	Label8	(Name)	lblBobot
		Alignment	2 – Center
		BackColor	&H00C0E0FF&
		Caption	Tulisan “ Label8 ” dihapus (jadi kosong)
		Font	Arial – Reguler – 20
		ForeColor	&H80000012&
15	CommandButton3	(Name)	CmdKeluar
		BackColor	&H0080FFFF&
		Caption	&K E L U A R
		Font	MS Sans Serif – Reguler – 19
		Style	1 – Graphical

c) *Source Code* nya :

```
Private Sub CmdHapus_Click()
    txtNilai.Text = ""
    lblNilaiKonversi.Caption = ""
    lblKet.Caption = ""
    lblBobot.Caption = ""
    txtNilai.SetFocus
End Sub
```

```
Private Sub CmdKeluar_Click()
    pesan = MsgBox("Anda Yakin untuk Keluar ...", vbQuestion + vbOKCancel, "Peringatan")
    If pesan = vbOK Then
        End
    Else
        Form1.Show
    End If
End Sub
```

```
Private Sub CmdProses_Click()
    NILAI = Val(txtNilai)
    If NILAI > 100 Or NILAI < 0 Then
        KONVERSI = MsgBox("Input Tidak Boleh >100 Yaa...", vbOKCancel + vbQuestion, "PERINGATAN")
        If KONVERSI = vbOK Then txtNilai.Text = ""
    Else
        lblNilaiKonversi.Caption = NILAI
        If NILAI > 70 Then
            lblNilaiKonversi = "A"
        End If
    End If
End Sub
```

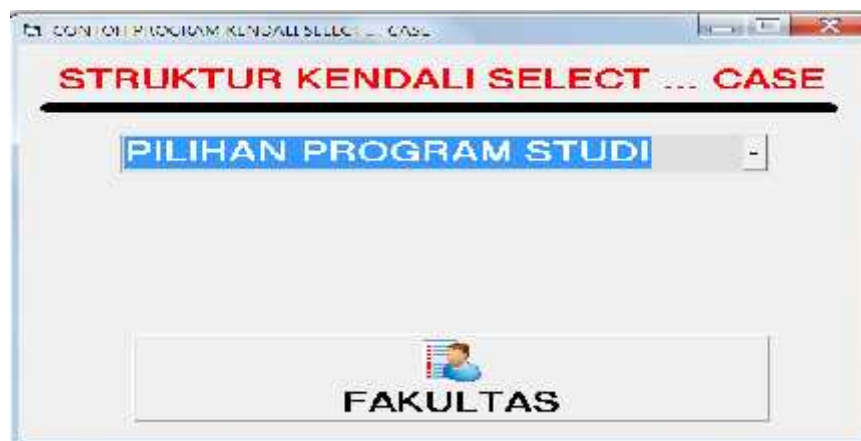


```
lblKet.Caption = "Sangat Baik"  
lblBobot.Caption = 4  
ElseIf NILAI >= 60 Then  
    lblNilaiKonversi = "B"  
    lblKet.Caption = "Baik"  
    lblBobot.Caption = 3  
ElseIf NILAI >= 50 Then  
    lblNilaiKonversi = "C"  
    lblKet.Caption = "Cukup"  
    lblBobot.Caption = 2  
ElseIf NILAI >= 40 Then  
    lblNilaiKonversi = "D"  
    lblKet.Caption = "Kurang"  
    lblBobot.Caption = 1  
Else  
    lblNilaiKonversi = "E"  
    lblKet.Caption = "Gagal"  
    lblBobot.Caption = 0  
End If  
End If  
CmdHapus.SetFocus  
End Sub
```

d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat08.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat08.frm “

2. Kerjakanlah latihan ke-9 berikut :

a) Buatlah **New Project** dan **Form**, sesuaikan antarmuka seperti gambar dibawah ini



Gambar 5.2. Antarmuka program struktur kendali *select....case*

b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	FrmKendaliCase
		BackColor	&H00FFFFC0&
		Caption	CONTOH PROGRAM KENDALI SELECT CASE
		Height	6012
		Width	8376
		StartPosition	2 – Center Screen
		WindowState	0 –Normal
2	Label1	(Name)	Label1
		Alignment	2 – Center
		Caption	STRUKTUR KENDALI SELECT ... CASE
		Font	Arial Narrow – Bold – 22
3	Line1	ForeColor	&H000000FF&
		(Name)	Line1
		BorderWidth	10
4	ComboBox1	(Name)	cmbKendaliCase
		Style	2 – Dropdown List
		BackColor	&H00E0E0E0&
		List	(List) “ Ketikkan mulai dari PAI lalu tekan tombol Enter di Keyboard ulangi terus sampai PMI “ PAI PMM PBA PBI PGRA MPI PGMI BKI KPI PMI
5	CommandButton1	(Name)	cmdfakultas
		Caption	Fakultas
		Font	Arial Narrow – Bold – 22
		Picture	(icon) “ Cari gambar dengan ekstensi .ICO “
		Style	1 – Graphical

c) *Source Code* nya :

```
Private Sub cmdfakultas_Click()
    Select Case cmbkendalicase.Text
        Case Is = "PAI"
            Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
            Case Is = "PMM"
            Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
            Case Is = "PBA"
            Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
            Case Is = "PBI"
            Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
            Case Is = "PGRA"
            Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
            Case Is = "MPI"
    
```

```

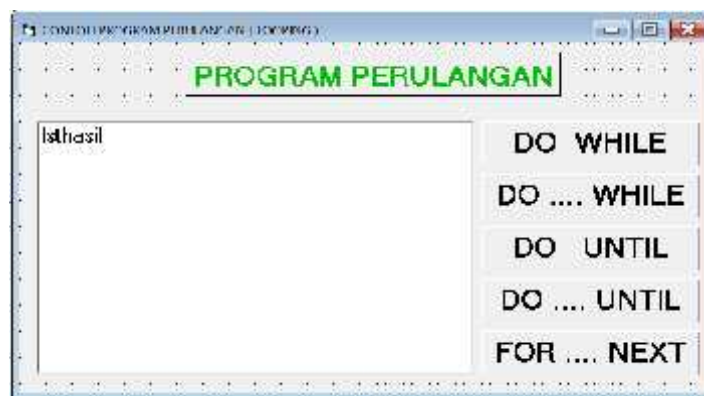
Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
  Case Is = "PGMI"
Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
  Case Is = "BKI"
Msg = MsgBox(" PILIHAN ANDA ILMU TARBIYAH DAN KEGURUAN", 0, " FAKULTAS ")
  Case Is = "KPI"
Msg = MsgBox(" PILIHAN ANDA DAKWAH ", 0, " FAKULTAS ")
  Case Is = "PMI"
Msg = MsgBox(" PILIHAN ANDA DAKWAH ", 0, " FAKULTAS ")
  Case Else
MsgBox (" TIDAK ADA PILIHAN DI DAFTAR KAMI")
  End Select
End Sub

```

- d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat09.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat09.frm “

3. Kerjakanlah latihan ke-10 berikut :

- a) Buatlah **New Project** dan **Form**, sesuaikan antarmuka seperti gambar dibawah ini



Gambar 5.3. Contoh program perulangan

- b) Atur (kontrol) seperti pada tabel dibawah ini

NO	Objek=Object=ToolBox	Properties	Ubah Propertiesnya (Kontrol)
1	Form1	(Name)	FrmPerulangan
		BackColor	&H800000F&
		Caption	CONTOH PROGRAM PERULANGAN (LOOPING)
		Height	5376
		Width	9708
		StartPosition	2 – Center Screen
		WindowState	0 –Normal
2	Label1	(Name)	Label1
		Alignment	2 – Center
		Caption	PROGRAM PERULANGAN
		Font	Arial Narrow – Bold – 22
		ForeColor	&H0000C000&

3	Shape1	(Name)	Shape1
		BorderStyle	1 - Solid
		BorderWidth	1
4	ListBox1	(Name)	Isthasil
		Font	Arial Narrow – Bold – 16
		ForeColor	&H80000008&
		Style	0 – Standart
5	CommandButton1	(Name)	cmdwhile
		Caption	DO WHILE
		Font	MS Sans Serif – Bold – 19
		Style	1 – Graphical
6	CommandButton2	(Name)	cmddowhile
		Caption	DO WHILE
		Font	MS Sans Serif – Bold – 19
		Style	1 – Graphical
7	CommandButton3	(Name)	cmddowhile
		Caption	DO UNTIL
		Font	MS Sans Serif – Bold – 19
		Style	1 – Graphical
8	CommandButton4	(Name)	cmddountil
		Caption	DO UNTIL
		Font	MS Sans Serif – Bold – 19
		Style	1 – Graphical
9	CommandButton5	(Name)	cmdfor
		Caption	FOR NEXT
		Font	MS Sans Serif – Bold – 19
		Style	1 – Graphical

c) *Source code nya*

```
Private Sub cmdUntil_Click()
```

```
Isthasil.Clear
```

```
i = 0
```

```
Do
```

```
Isthasil.AddItem " Perulangan Ke- " & i
```

```
i = i + 1
```

```
Loop Until i > 10
```

```
End Sub
```

```
Private Sub cmddountil_Click()
```

```
Isthasil.Clear
```

```
i = 0
```

```
Do Until i > 10
```

```
Isthasil.AddItem " Perulangan Yang Ke- " & i
```

```
i = i + 1
```

```
Loop
```

```
End Sub
```

```
Private Sub cmddowhile_Click()
```

```
Isthasil.Clear
```

```
i = 0
```

```
Do While i <= 10
```

```
Isthasil.AddItem " Perulangan Sebanyak Yang Ke- " & i
```

```
i = i + 1
```

```
Loop
```

```
End Sub
```

```
Private Sub cmdWhile_Click()  
Isthasil.Clear  
i = 0  
Do  
Isthasil.AddItem " Perulangan Yang Ke- " & i  
i = i + 1  
Loop While i <= 10  
End Sub
```

```
Private Sub cmdfor_Click()  
Isthasil.Clear  
For i = 0 To 10  
Isthasil.AddItem " Perulangan Sebanyak ke- " & i  
Next i  
End Sub
```

- d) Simpan *project* dengan nama **File Project (*.vbp)** beri nama : “ PrjLat10.vbp “ dan File Form1 (*.frm) beri nama : “ FrmLat10.frm “

SOAL

1. Buatlah program seperti gambar yang terlihat dibawah, untuk menghitung total nilai akhir mahasiswa, dengan ketentuan sebagai berikut:
 - a) Masukan (*input*) : NIM, Nama, Mata Kuliah, Prodi, Nilai kehadiran, nilai kuis, nilai tugas, nilai MID dan nilai UAS.
 - b) Proses : Nilai kehadiran di kali 10% + nilai kuis di kali 5% + nilai tugas-tugas di kali 15% + nilai MID di kali 30% + nilai UAS di kali 40%.
 - c) Keluaran (*Output*) :
 - Jika Total nilai akhir mahasiswa ≥ 85 s/d 100 maka konversi nilai huruf = “ A “ dan Keterangan = “ Sangat Baik “
 - Jika Total nilai akhir mahasiswa ≥ 75 s/d 84 maka konversi nilai huruf = “ B + “ dan Keterangan = “ Lebih dari Baik “
 - Jika Total nilai akhir mahasiswa ≥ 70 s/d 74 maka konversi nilai huruf = “ B “ dan Keterangan = “ Baik “
 - Jika Total nilai akhir mahasiswa ≥ 65 s/d 69 maka konversi nilai huruf = “ C + “ dan Keterangan = “ Lebih dari cukup “
 - Jika Total nilai akhir mahasiswa ≥ 60 s/d 64 maka konversi nilai huruf = “ C “ dan Keterangan = “ Cukup “
 - Jika Total nilai akhir mahasiswa ≥ 50 s/d 59 maka konversi nilai huruf = “ D “ dan Keterangan = “ Kurang “
 - Jika Total nilai akhir mahasiswa ≤ 49 maka konversi nilai huruf = “ E “ dan Keterangan = “ Gagal “

The screenshot shows a Java Swing window titled "PROGRAM HITUNG TOTAL NILAI MAHASISWA". The window has a title bar with standard OS controls. The main area is divided into several sections. At the top, there's a header with the program name. Below that, there are input fields for "NILAI KEHADIRAN", "NIM", and "NAMA". The next row has "NILAI KUIS", "PRODI", and "KONVERSI KE NILAI HURUF". The third row has "NILAI TUGAS" and "KETERANGAN". The fourth row has "NILAI MID", "MATA KULIAH", and "BOBOT". The fifth row has "NILAI UAS" and a yellow button labeled "HAPUS". At the bottom, there are two yellow buttons: "HITUNG" and "KELUAR".

Gambar 5.4. Program hitung total nilai mahasiswa

2. Buatlah program perulangan untuk deret fibonacci dan n faktorial untuk bilangan bulat positif

Tindak Lanjut (*Feed Back*)

Skor	Kriteria
100-80	Mampu dan dapat menyelesaikan latihan-latihan dan soal-soal: 1. Bekerja dengan struktur pencabangan <i>if... then ... else</i> 2. Bekerja dengan struktur pencabangan <i>select ... case</i> 3. Bekerja dengan struktur perulangan (<i>looping</i>)
70-60	Hanya mampu menjawab 1 soal
50 ke bawah	Tidak mampu sama sekali dan harus mengulang kegiatan belajar 5 secara keseluruhan

Daftar Pustaka

- Alam, M. Agus. J. *“Belajar Sendiri Microsoft Visual Basic 6.0”*, Elex Media Komputindo. Jakarta. 1999.
- Arief, Ramadhan. *“Microsoft Visual Basic 6.0”*, Elex Media Komputindo. Jakarta. 2004.
- Halvorson, Michael, *“ Step By Step MS Visual Basic 6.0”*, Microsoft Press, 1999.
- Hadi, Rahadian, *“Student Guide Series:Pengenalan Visual Basic”*, Elex Media Komputindo. Jakarta. 2006.
- Kurniadi, Adi. *“Pemrograman Visual Basic 6.0”*. Elex Media Komputindo. Jakarta. 2000
- Kusumo, Suryo, Ario, Drs, *“Microsoft Visual Basic 6.0”*, Elex Media Komputindo, PT, Gramedia, Jakarta, 2000.
- Madcom. *“Mahir Dalam 7 Hari Pemrograman Visual basic 6.0”*. Andi Off Set. Yogyakarta. 2005.
- Pamungkas, Ir, *“Microsoft Visual Basic 6.0”*, Elex Media Komputindo, PT, Gramedia, Jakarta, 2000.
- Subari & Yuswanto, *”Panduan Lengkap Pemrograman Visual Basic 6.0”*, Jakarta: Cerdas Pustaka Publisher, 2008.
- Supardi, Yuniar, Ir. *“Microsoft Visual Basic 6.0”*. Untuk Segala Tingkatan. 2006. Elex Media Komputindo. Jakarta
- Team Penyusun Modul, *“Modul VB Fundamental”*, Akademik Manajemen Informatika dan Komputer, Bina Sarana Informatika. Jakarta. 2010
- Tim Penyusun, *“Modul Praktek Laboratorium Komputer Visual Basic 6.0”*, Jakarta. AMIK Bina Sarana Informatika. 2006.
- Tim Penyusun, *“Modul Praktikum Visual Basic”*. Yogyakarta. FMIPA Universitas Gadjah Mada. 2002.
- <http://oke.or.id>. Diakses tanggal 12/09/2014
- <http://planet-source-code.com>. Diakses tanggal 13/09/2014
- <http://ilmukomputer.org/2013/05/27/visual-basic-6-0-sesi-1>. Diakses tanggal 12/10/2014
- <http://dwi.its-sby.edu/PIKTI/Visual%20Basic/last/Bab1.doc>
- <http://ikc.vip.net.id/berseri/krisna-vb6/index.php>
- http://ns1.cic.ac.id/~marsani.asfi/tulisan/Modul_1.htm
- <http://leo.apeaje.info/vb/tutor.html>
- http://ns1.cic.ac.id/~marsani.asfi/tulisan/Modul_1.htm
- http://www.ajibsusanto.site88.net/aplikasi_bisnis/MODUL_VB_6.pdf
- <http://www.masinosinaga.com/index.php?name=News&file=article&sid=234>
- <http://absanka.wordpress.com/2008/07/05/pdf-program-visual-basic/>
- <http://www.kuliahit.com/kuliahit/article/16/Fungsi-Date-and-Time-pada-VB>
- <http://ikc.cbn.net.id/berseri/krisna-vb6/krisna-vb6-06.zip>
- <http://leo.apeaje.info/vb/lesson8.html>
- <http://lecturer.eepis-its.edu/~tessy/tutorial/bab5.pdf>