

BA	10	KKME2012	39
----	----	----------	----

BUKU I BAHAN AJAR

Komputer dan Dasar Pemrograman

**Penyusunan Bahan Ajar Dalam Kurikulum Berbasis
Kompetensi (Kurikulum 2007) ini dibiayai dari DIPA
Politeknik Negeri Bandung
Departemen Pendidikan Nasional
Tahun Anggaran 2010**

**Disusun Oleh :
Tria Mariz Arief
NIP : 132305208**



**PROGRAM STUDI TEKNIK AERONAUTIKA
JURUSAN TEKNIK MESIN
POLITEKNIK NEGERI BANDUNG
2010**



HALAMAN PENGESAHAN

1. Identitas Bahan Ajar

- a. Judul Bahan Ajar : Komputer dan Dasar Pemrograman
- b. Mata Kuliah / Semester : Komputer / III
- c. SKS (T-P) /Jam (T-P) : 2(2-2)
- d. Jurusan : Teknik Mesin
- e. Program Studi : Teknik Mesin
- e. Nomor Kode Mata Kuliah : KKME2012

2. Penulis

- a. Nama : Tria Mariz Arief
- b. NIP : 132305308
- c. Pangkat / Golongan : III/a
- d. Jabatan Fungsional : Asisten ahli
- e. Program Studi : Teknik Aeronautika
- f. Jurusan : Teknik Mesin

Bandung, 30 Juni 2010

Mengetahui,
Ketua KBK

Penulis,

Ir. Soni Wahyudi, MT
NIP. 131870096

Tria Mariz Arief
NIP. 132305308

Menyetujui,
Ketua Jurusan / Program Studi

Dr. Carolus Bintoro, Dipl. Ing., MT.
NIP. 19620602 199102 1 001



DAFTAR ISI

HALAMAN PENGESAHAN	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	ix
DESKRIPSI MATA KULIAH	x
CARA PENGGUNAAN	xi
BAB 1	1
PENDAHULUAN	1
1.1 Dasar tentang Komputer.....	1
1.2 Bagian-Bagian Komputer dan Fungsinya	3
1.3 Penggunaan Komputer	5
BAB 2	8
PROGRAM APLIKASI PENGOLAH KATA.....	8
2.1 Dasar Aplikasi Pengolah Kata (<i>Word Processing</i>)	8
2.2 Perintah-Perintah Dasar dalam Microsoft Word 2007	9
2.3 Penggunaan Dokumen <i>Template</i>	11
Contoh Penggunaan Template	12
2.4 Pengaturan Layout (Tata Letak Kertas Kerja)	14
2.4.1 Pengaturan Halaman (<i>Page Layout</i>)	14
2.4.2 Pengaturan <i>Header</i> dan <i>Footer</i> (<i>Section</i>).....	15
2.5 Pengaturan Format Tulisan dengan <i>STYLE</i>	16
2.6 Pembuatan Daftar Isi dan Daftar Gambar Secara Otomatis	17
2.6.1 Multilevel Numbering.....	18
2.6.2 Caption.....	19
2.6.3 Membuat Daftar Isi	20
2.6.4 Membuat Daftar Gambar	21
BAB 3	22
PROGRAM APLIKASI SPREADSHEET	22
3.1 Dasar Aplikasi <i>Spreadsheet</i> dengan Microsoft EXCEL 2007	23
3.2 Memasukkan Data dan Operasi Dasar pada Cell.....	24
3.2.1 Konstanta	25
3.2.2 Formula	26

3.2.3	Cara penulisan referensi cell	28
3.3	Fungsi-Fungsi Umum dalam Microsoft Excel 2007	31
3.4	Fungsi LOOKUP	33
3.5	Pembuatan <i>Calculation Sheet</i>	34
	Contoh: Perhitungan dan Pengecekan tebal pipa (ASME B31.8)	35
3.6	Pembuatan Grafik	37
3.6.1	Jenis-Jenis Grafik	37
3.6.2	Cara Membuat Grafik Sederhana	39
BAB 4	41
PROGRAM APLIKASI PRESENTASI	41
4.1	Konsep Dasar Tentang Presentasi	41
4.2	Tipografi, Layout dan Desain	42
4.2.1	Mengubah Font secara Global dan Lokal	47
4.2.2	Mengorganisasi Informasi untuk Kejelasan	48
4.3	Pengaturan	48
BAB 5	49
DASAR PEMROGRAMAN (PROBLEM SOLVING)	49
5.1	Siklus Pengembangan Program	49
5.1.1	Gambaran Umum	49
5.1.2	Perencanaan Program	50
5.2	Flowchart sebagai Algoritma Pemrograman	51
BAB 6	54
DASAR PEMROGRAMAN DENGAN VISUAL BASIC	54
6.1	Mengenali Antarmuka Visual Basic 6.0	54
6.2	Memahami Istilah Object, Property, Method dan Event	56
6.3	Objek-Objek di Visual Basic 6.0	57
6.4	Menggunakan Event dan Property	57
6.4.1	Membuat <i>User Interface</i> (UI)	57
	Contoh Program: Antarmuka dan Objek	58
6.4.2	Mengatur Property Object	58
6.4.3	Menulis Kode Program : Property Test	59
6.4.4	Modifikasi Kode Program	61
6.5	Menggunakan Event dan Method.	62
6.5.1	Membuat <i>User Interface</i> (UI)	62
6.5.2	Menulis Kode Program : Method Test	64
6.6	Pendefinisian Data dan Variabel	66
	Contoh Program : Variabel Test	68
BAB 7	73
PROGRAM PILIHAN (DECISION)	73

7.1	Operator Relasional dan Logikal.....	73
	Contoh Program : Operator Test	74
7.2	Pilihan dengan Blok - IF	79
	Contoh Program : Struktur IF... THEN	80
	TANTANGAN :	83
7.3	Pilihan dengan Blok SELECT-CASE	83
	Contoh Program : Struktur SELECT...CASE	83
BAB 8		88
PROGRAM PENGULANGAN (REPETITION)		88
8.1	Pengulangan dengan FOR .. NEXT	88
8.2	Pengulangan dengan DO .. LOOP	88
	Contoh Program : Struktur Looping.....	89
BAB 9		92
PEMROGRAMAN DENGAN ARRAY		92
9.1	Menggunakan ARRAY	92
	Contoh Program: Array Test	93
9.2	Control ARRAY	97
	Contoh Program: Kalkulator	98
BAB 10		102
PENANGANAN ERROR		102
10.1	Jenis-Jenis Error	102
10.2	Menangani Runtime Error.....	103
	Contoh Program : Error Handle	104
BAB 11		109
PENGUNAAN PROCEDURE		109
11.1	Apa itu Procedure ?	109
11.2	Jenis-jenis Procedure	109
	Contoh Program : Procedure Test	110
11.3	Pernyataan Optional	113
11.4	Pernyataan ByRef dan ByVal	113
11.5	Procedure Built-in	114
BAB 12		115
PEMROGRAMAN GRAFIK		115
12.1	Dasar-Dasar Grafik	115
12.1.1	Mendefinisikan Sistem Koordinat	116
12.1.2	Metoda Grafik untuk menggambar garis, titik dan lingkaran.....	117
	Contoh Program Grafik Persamaan Akar Kuadrat.....	118
12.1.3	Memosisikan TEXT	119

12.2 Diagram GARIS (<i>LINE Charts</i>).....	120
Contoh Program Diagram Garis.....	120
REFERENSI	124
GBPP (Garis-garis Besar Program Pengajaran)	125
SAP (Satuan Acara Perkuliahan)	126



DAFTAR GAMBAR

Gambar 1.1	Diagram sistem komputer	2
Gambar 1.2	Sistem enjiniring yang tak lepas dari penggunaan sistem komputer.	6
Gambar 1.3	CATIA V5 yang merupakan Sistem CAD/ CAE/CAM terintegrasi.	7
Gambar 2.1	Antarmuka aplikasi pengolah kata berbasis <i>opensource</i>	8
Gambar 2.2	Antarmuka program aplikasi pengolah kata komersial.	9
Gambar 2.3	Antarmuka Aplikasi Microsoft Word 2007	9
Gambar 2.4	Menu-menu dalam Office Button	10
Gambar 2.5	Status Bar	10
Gambar 2.6	Kotak dialog membuat dokumen baru dengan pilihan template yang ada. 11	
Gambar 2.7	Styles Quick Gallery untuk mengakses dengan cepat format style dari paragraf dokumen.	12
Gambar 2.8	Format dokumen dengan menggunakan <i>style formatting</i>	13
Gambar 2.9	Tampilan layout dokumen yang akan disiapkan.	14
Gambar 2.10	Kotak dialog untuk melakukan pengaturan margin dan jenis kertas yang digunakan.	14
Gambar 2.11	Kotak dialog untuk melakukan pengaturan section, header dan footer. .	15
Gambar 2.12	Ilustrasi pengaturan Header dan footer untuk section yang berbeda maupun dalam section yang sama tetapi beda antara halaman ganjil dan genap.	16
Gambar 2.13	Cara melakukan modifikasi pada style yang sudah ada dari Styles Quick Gallery	16
Gambar 2.14	Kotak dialog untuk modifikasi style.	17
Gambar 2.15	Cara melakukan modifikasi style atau membuat style baru dari Manage Styles.	17
Gambar 2.16	Contoh multilevel list dengan penomoran berjenjang	18
Gambar 2.17	Contoh multilevel list dengan penomoran gabungan huruf dan angka ...	18
Gambar 2.18	Menu di Multilevel List yang bisa digunakan.	19
Gambar 2.19	Contoh caption dan cara mengakses perintah-nya.	19
Gambar 2.20	Kotak dialog caption.	20
Gambar 2.21	Urutan perintah untuk membuat daftar isi secara otomatis	20
Gambar 2.22	Cara mengatur pembuatan daftar isi sesuai format yang diinginkan dan style mana saja yang termasuk dalam daftar isi.	21

Gambar 2.23	Urutan perintah untuk membuat daftar gambar/tabel secara otomatis ...	21
Gambar 3.1	Contoh sederhana pemanfaatan aplikasi spreadsheet untuk membuat grafik.	22
Gambar 3.2	Contoh pengoperasian bilangan dalam spreadsheet dengan formula tertentu. SUM = penjumlahan group. Operator + (tambah), * (kali) dan lainnya digunakan dalam operasi matematik.	23
Gambar 3.3	Contoh antarmuka aplikasi spreadsheet	23
Gambar 3.4	Tampilan antarmuka aplikasi Microsoft Excel 2007.	24
Gambar 3.5	Cara penulisan formula harus diawali dengan tanda (=) untuk memberdakan dengan teks.	26
Gambar 3.6	Contoh penggunaan operator logika dengan fungsi IF. Dalam contoh ini nilai yang dihasilkan adalah 25, karena nilai logikanya adalah TRUE. .	28
Gambar 3.7	Contoh penggunaan formula teks	28
Gambar 3.8	Contoh penggunaan operator jangkauan dalam formula.	29
Gambar 3.9	Contoh penggunaan operator gabungan. Sebagai contoh A1=1 dan B1:B3 = 2, 5 dan 8 menghasilkan 16 jika disumasikan.	29
Gambar 3.10	Contoh penggunaan operator irisan. Angka 5, 6, 8 dan 9 adalah irisan .	30
Gambar 3.11	Cara perujukan cell referensi dengan cara relatif, absolut dan gabungan.	30
Gambar 3.13	Kelompok Function Library yang disediakan MS Excel dalam menu tab Formulas	32
Gambar 3.12	Nama cell yang dirujuk seolah sebagai rujukan absolut. Dalam contoh disini adalah cell B1 yang diubah namanya menjadi "phi"	31
Gambar 5.1	Diagram Alir Pemecahan Masalah (<i>Problem Solving</i>).	50
Gambar 5.2	Contoh Flowchart untuk permasalahan stempel pos.	53
Gambar 5.3	Contoh flowchart untuk (a) Program pilihan dan (b) Program pengulangan.	53
Gambar 6.1	Tampilan awal Program Visual Basic 6.0	54
Gambar 6.2	Tampilan Antarmuka Program Visual Basic 6.0.	55
Gambar 6.3	Ilustrasi pengertian Object, Property, Method dan Event.	56
Gambar 6.4	Kelompok Toolbar ToolBox yang berisi komponen-komponen Control dasar.	57
Gambar 6.5	Contoh Form dengan beberapa objek di dalamnya.	58
Gambar 6.6	Bagian-bagian di dalam jendela property	58
Gambar 6.7	Hasil pengubahan property pada objek, bandingkan dengan Gambar 6.5.	59
Gambar 6.8	Tampilan jendela Code	59
Gambar 6.9	Hasil Run program	60
Gambar 6.10	Fasilitas pengenalan terhadap property maupun method untuk objek pada saat pengetikan di Layar Code secara otomatis.	61

Gambar ¶6.11	Form baru dengan penambahan objek frame, optionButton dan CheckBox.....	61
Gambar ¶11.1	Object browser untuk mencari built in procedure (prosedur yang disediakan oleh Visual Basic)	114
Gambar ¶12.1	Empat jenis diagram yang bisa dibuat dengan menggunakan Visual Basic.....	115
Gambar ¶12.2	Posisi pendefinisian sistem koordinat di VB.	117
Gambar ¶12.3	Hasil gambar berupa titik, garis dan lingkaran	118
Gambar ¶12.4	Jenis-jenis garis yang diatur melalui metoda .DrawStyle.....	118
Gambar ¶12.5	Hasil running program grafik persamaan akar kuadrat $y = x$	119
Gambar ¶12.6	Hasil pencetakan teks pada grafik (a) kasus pertama (b) kasus kedua .	119
Gambar ¶12.7	Diagram garis dari contoh program.	120
Gambar ¶12.8	Hasil running program diagram garis.	123



DAFTAR TABEL

Tabel 5.1	Tabulasi urutan proses penyelesaian masalah dalam contoh soal cerita matematika sederhana.	50
Tabel 5.2	Daftar lambang gambar yang digunakan dalam flowchart.	52
Tabel 6.1	Tabulasi perubahan nilai dari property objek	59
Tabel 7.1	Daftar simbol operator aritmatika dan contohnya	73
Tabel 7.2	Daftar simbol operator relasional (perbandingan) beserta contoh	74
Tabel 7.3	Daftar simbol operator logika dan contohnya	74
Tabel 12.1	Two-year Colleges Enrollment (in Thousands)	120



DESKRIPSI MATA KULIAH

Identitas Mata Kuliah

- | | | |
|-----------------------|---|--------------|
| 1. Judul Mata Kuliah | : | Komputer |
| 2. Nomor Kode / SKS | : | KKME2012 / 3 |
| 3. Semester / Tingkat | : | III |
| 4. Prasyarat | : | - |
| 5. Jumlah Jam/Minggu | : | 6 |

Ringkasan Topik / Silabus

Kompetensi Yang Ditunjang

Tujuan Pembelajaran Umum

Tujuan Pembelajaran Khusus



CARA PENGGUNAAN

Pedoman Mahasiswa

Pedoman Pengajar

Penggunaan Ilustrasi dalam Bahan Ajar



BAB 1 PENDAHULUAN

Tujuan Pembelajaran Umum

1. Mengetahui dasar tentang komputer

Tujuan Pembelajaran Khusus

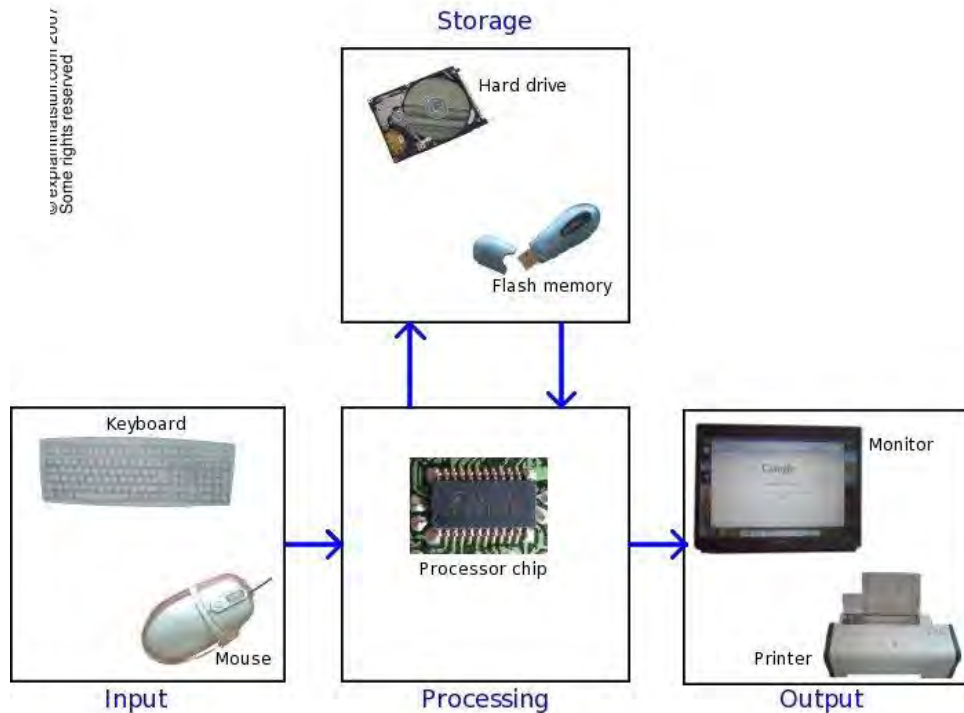
1. Mahasiswa dapat memahami konsep dasar komputer dari segi arsitektur dan kegunaannya.
2. Mahasiswa dapat mengetahui perkembangan komputer baik dari segi *hardware* maupun *software*.

1.1 Dasar tentang Komputer

Kehidupan manusia saat ini nampaknya sudah tidak dapat lagi dilepaskan dari teknologi, khususnya Komputer. Bahkan, banyak peralatan berbasis komputer saat ini yang umum digunakan dan dimiliki oleh masyarakat, misalnya *personal data assistant (PDA)*, *global position system (GPS)*, *mobile computer (desktop, laptop)*, *mobile phone (HP)*, *translator*, dan sebagainya. Komputer (*computer*) diambil dari *computare* (bhs latin) yang berarti menghitung (*to compute* atau *to reckon*). Kata komputer semula dipergunakan untuk menggambarkan orang yang pekerjaannya melakukan perhitungan aritmatika, dengan atau tanpa alat bantu, tetapi arti kata ini kemudian dipindahkan kepada mesin itu sendiri. Beberapa pakar dan peneliti memiliki pengertian tersendiri dengan komputer diantaranya sebagai berikut :

1. **Robert H. Blissmer** dalam buku *Computer Annual*, Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas seperti menerima **input**, memproses input tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil pengolahan, serta menyediakan **output** dalam bentuk informasi.
2. **Donald H. Sanders** dalam buku *Computer Today*, Komputer adalah sistem elektronik utk memanipulasi data yg cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data **input**, memprosesnya dan menghasilkan **output** dibawah pengawasan suatu langkah-langkah, instruksi2 program yg tersimpan di memori (*stored program*).
3. **Hamacher dkk**, dalam buku *Computer Organization*, Komputer adalah mesin penghitung elektronik yg cepat dapat menerima informasi **input** digital, memprosesnya sesuai dengan suatu program yg tersimpan di memorinya dan menghasilkan **output** informasi.

Jadi dapat disimpulkan bahwa komputer adalah sekelompok alat elektronik yang terdiri atas perintah input, alat yang mengolah input, dan peralatan output yang memberikan informasi serta bekerja secara otomatis. Jadi cara kerja komputer dapat digambarkan seperti pada Gambar 1.1.



Gambar 1.1 Diagram sistem komputer

Input Device, adalah perangkat-perangkat keras komputer yang berfungsi untuk memasukkan data ke dalam memori komputer, seperti *keyboard*, *mouse*, *joystick* dan lain-lain.

Prosesor, adalah perangkat utama komputer yang mengelola seluruh aktifitas komputer itu sendiri. Prosesor terdiri dari dua bagian utama, yaitu:

- **Control Unit (CU)**, merupakan komponen utama prosesor yang mengontrol semua perangkat yang terpasang pada komputer, mulai dari *input device* sampai *output device*.
- **Arithmetic Logic Unit (ALU)**, merupakan bagian dari prosesor yang khusus mengolah data aritmatika (menambah, mengurangi dll) serta data logika (perbandingan).

Memori adalah media penyimpanan data pada komputer. Memori ini terbagi atas dua macam, yaitu:

- **Read Only Memory (ROM)**, yaitu memori yang hanya bisa dibaca saja, tidak dapat dirubah dan dihapus dan sudah diisi oleh pabrik pembuat komputer. Isi ROM diperlukan pada saat komputer dihidupkan. Perintah yang ada pada ROM sebagian akan dipindahkan ke RAM. Perintah yang ada di ROM antara lain adalah perintah untuk membaca sistem operasi dari disk, perintah untuk mengecek semua peralatan yang ada di unit sistem

dan perintah untuk menampilkan pesan di layar. Isi ROM tidak akan hilang meskipun tidak ada aliran listrik. Tapi pada saat sekarang ini ROM telah mengalami perkembangan dan banyak macamnya, al;

1. **PROM (Programable ROM)**, yaitu ROM yang bisa kita program kembali dengan catatan hanya boleh satu kali perubahan setelah itu tidak dapat lagi diprogram.
 2. **RPROM (Re-Programable ROM)**, merupakan perkembangan dari versi PROM dimana kita dapat melakukan perubahan berulang kali sesuai dengan yang diinginkan.
 3. **EPROM (Erasable Program ROM)**, merupakan ROM yang dapat kita hapus dan program kembali, tapi cara penghapusannya dengan menggunakan sinar ultraviolet.
 4. **EEPROM (Electrically Erasable Program ROM)**, perkembangan mutakhir dari ROM dimana kita dapat mengubah dan menghapus program ROM dengan menggunakan teknik elektrik. EEPROM ini merupakan jenis yang paling banyak digunakan saat ini.
- **Random Access Memori (RAM)**, dari namanya kita dapat artikan bahwa RAM adalah memori yang dapat diakses secara random. RAM berfungsi untuk menyimpan program yang kita olah untuk sementara waktu (power on) jika komputer kita matikan, maka seluruh data yang tersimpan dalam RAM akan hilang. Tujuan dari RAM ini adalah mempercepat pemroses data pada komputer.

Agar data yang kita buat tidak dapat hilang pada saat komputer dimatikan, maka diperlukan media penyimpanan eksternal, seperti Disket, Harddisk, PCMCIA card dan lain-lain.

Output Device, adalah perangkat komputer yang berguna untuk menghasilkan keluaran, apakah itu ke kertas (*hardcopy*), ke layar monitor (*softcopy*) atau keluaran berupa suara. Contohnya printer, speaker, plotter, monitor dan banyak yang lainnya.

Dari penjelasan diatas dapat kita simpulkan bahwa **prinsip kerja komputer** tersebut diawali memasukan data dari **perangkat input**, lalu data tersebut diolah sedemikian rupa oleh **CPU** sesuai yang kita inginkan dan data yang telah diolah tadi disimpan dalam **memori** komputer atau disk. Data yang disimpan dapat kita lihat hasilnya melalui **perangkat keluaran**.

1.2 Bagian-Bagian Komputer dan Fungsinya

Komputer terdiri dari tiga komponen utama yang tidak dapat dipisahkan, yaitu;

Hardware (perangkat keras), merupakan peralatan fisik dari komputer yang dapat kita lihat dan rasakan. Hardware ini terdiri dari;

- **Input/Output Device (I/O Device)**

Terdiri dari perangkat masukan dan keluaran, seperti keyboard dan printer. (lihat sub menu periferal)

- **Storage Device** (perangkat penyimpanan)

Merupakan media untuk menyimpan data seperti disket, harddisk, CD-I, dll.

- **Monitor /Screen**

Monitor merupakan sarana untuk menampilkan apa yang kita ketikkan pada papan keyboard setelah diolah oleh prosesor. Monitor disebut juga dengan *Visual Display Unit* (VDU).

- **Casing Unit**

Casing unit adalah tempat dari semua peralatan komputer, baik itu motherboard, card, peripheral lain dan Central Processing Unit (CPU). Casing unit ini disebut juga dengan System Unit.

- **Central Processing Unit (CPU)**

Central Processing Unit adalah salah satu bagian komputer yang paling penting, karena jenis prosesor menentukan pula jenis komputer. Baik tidaknya suatu komputer, jenis komputer, harga komputer, ditentukan terutama oleh jenis prosesor. Semakin canggih prosesor komputer, maka kemampuannya akan semakin baik dan biasanya harganya akan semakin mahal.

Software (perangkat lunak), merupakan program-program komputer yang berguna untuk menjalankan suatu pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis dengan bahasa khusus yang dimengerti oleh komputer. Software terdiri dari beberapa jenis, yaitu ;

- **Sistem Operasi**, seperti DOS, Unix, Novell, OS/2, Windows, dll.

Adalah software yang berfungsi untuk mengaktifkan seluruh perangkat yang terpasang pada komputer sehingga masing-masingnya dapat saling berkomunikasi. Tanpa ada sistem operasi maka komputer tak dapat difungsikan sama sekali.

- **Program Utility**, seperti Norton Utility, Scandisk, PC Tools, dll.

Program utility berfungsi untuk membantu atau mengisi kekurangan/kelemahan dari sistem operasi, misalnya PC Tools dapat melakukan perintah format sebagaimana DOS, tapi PC Tools mampu memberikan keterangan dan animasi yang bagus dalam proses pemformatan. File yang telah dihapus oleh DOS tidak dapat dikembalikan lagi tapi dengan program bantu hal ini dapat dilakukan.

- **Program Aplikasi**, seperti GL, MYOB, Payroll, dll.

Merupakan program yang khusus melakukan suatu pekerjaan tertentu, seperti program gaji pada suatu perusahaan. Maka program ini hanya digunakan oleh bagian keuangan saja tidak dapat digunakan oleh departemen yang lain. Biasanya program aplikasi ini dibuat oleh seorang programmer komputer sesuai dengan permintaan/kebutuhan seseorang/lembaga/perusahaan guna keperluan interennya.

- **Program Paket**, seperti MS-Word, MS-Excel, Lotus 125, dll

Adalah program yang disusun sedemikian rupa sehingga dapat digunakan oleh banyak orang dengan berbagai kepentingan. Seperti MS-Word, dapat digunakan oleh departemen keuangan untuk membuat nota, atau bagian administrasi untuk membuat surat penawaran dan lain sebagainya.

- **Bahasa Pemrograman**, Pascal, Fortran, Clipper, dBase, dll.

Merupakan software yang khusus digunakan untuk membuat program komputer, apakah itu sistem operasi, program paket dll. Bahasa pemrograman ini biasanya dibagi atas 3 tingkatan, yaitu ;

1. **Low Level Language**, bahasa pemrograman generasi pertama, bahasa pemrograman jenis ini sangat sulit dimengerti karena instruksinya menggunakan bahasa mesin. Biasanya yang mengerti hanyalah pembuatnya saja.
2. **Midle Level Language**, merupakan bahasa pemrograman tingkat menengah dimana penggunaan instruksi sudah mendekati bahasa sehari-hari, walaupun begitu masih sulit untuk di mengerti karena banyak menggunakan singkatan-singakatan seperti STO artinya simpan (singkatan dari STORE) dan MOV artinya pindah (singkatan dari MOVE).Yang tergolong kedalam bahasa ini adalah Assembler, ForTran (Formula Translator).
3. **High Level Language**, merupakan bahasa tingkat tinggi yang mempunyai ciri mudah dimengerti, karena menggunakan bahasa sehari-hari, seperti BASIC, COBOL, dBase dll.

Brainware (User), adalah personil-personil yang terlibat langsung dalam pemakaian komputer, seperti Sistem analis, programmer, operator, user, dll. Pada organisasi yang cukup besar, masalah komputerisasi biasanya ditangani oleh bagian khusus yang dikenal dengan bagian **EDP** (Electronic Data Processing), atau sering disebut dengan EDP Departemen, yang dikepalai oleh seorang Manager EDP.

1.3 Penggunaan Komputer

Akhir dasawarsa 80-an, Jepang berhasil merebut posisi sebagai negara industri yang paling terkemuka di dunia, dimana pada satu dasawarsa sebelumnya posisi tersebut masih dimiliki oleh negara Amerika Serikat. Salah satu produk utama jepang pada saat itu adalah kendaraan bermotor seperti : *Honda, Toyota, Nissan, Daihatsu, dsb.*

Dari sekian banyak pengkajian yang telah dilakukan ditemukan beberapa fakta yaitu :

Jepang disamping ketekunan kerjanya, telah memanfaatkan jasa komputer dan robot dalam industri mereka

Satu hal yang membedakan robot dengan mesin lainnya dalam otomasi industri adalah **kemampuannya untuk diprogram.**



Gambar 1.2 Sistem enjinering yang tak lepas dari penggunaan sistem komputer.

Saat ini, dalam sistem enjinering (seperti pada Gambar 1.2); pesawat terbang, otomotif, elektronik, konstruksi, ruang angkasa, manufaktur dan lain-lain tidak dapat lepas dari penggunaan komputer. Komputer telah mengambil peran penting dalam industri. Beberapa istilah pokok aplikasi komputer di industri :

- **CAD (Computer Aided Design)**

Berfungsi sebagai meja gambar elektronik untuk para perancang dan juru gambar, hingga saat ini telah berkembang hingga mengerjakan semua proses desain.

Aplikasinya industri : penerbangan, automotive, electronic, arsitektur dll.

Software : AutoCAD, ArchiCAD, CATIA, ProEningeer dll.

- **CAE (Computer Aided Engineering)**

Berfungsi sebagai alat bantu analisis dan simulasi engineering. Stress analysis di struktur, CFD (Computational Fluid Dynamic) adalah beberapa contoh aplikasi CAE.

Aplikasinya industri : penerbangan, automotive, electronic, arsitektur dll.

Software : ANSYS, SIMULIA ABAQUS, NASTRAN dll.

- **CAM (Computer Aided Manufacturing)**

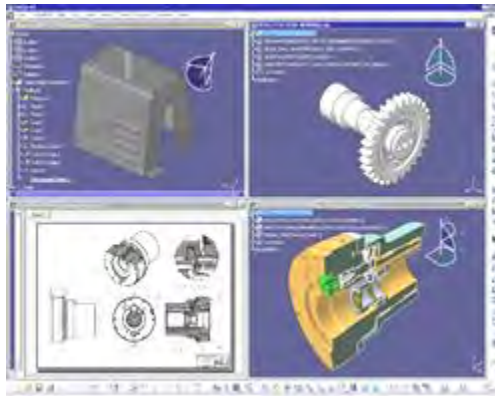
Berfungsi membantu proses manufaktur dari pemrograman NC sampai dengan *virtual manufacturing process*.

Aplikasinya industri : NC Programming, Tooling, robot dll

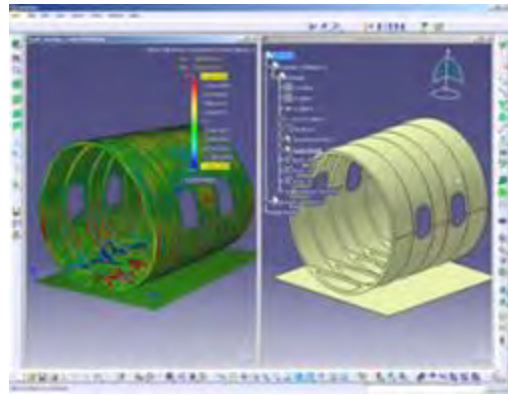
Software : MasterCAM, Delcam, ProCAM, Cimatron dll.

Saat ini system CAD/CAE/CAM terintegrasi merupakan sistem yang dikembangkan untuk mengakomodir kegiatan desain dari hulu hingga ke hilir atau disebut sebagai *End-to-End Design System*. Beberapa pengembang telah

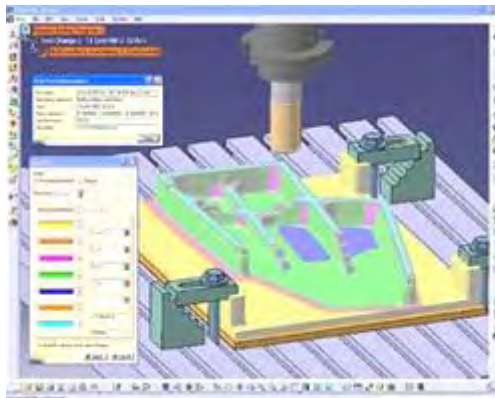
berkontribusi dalam sistem ini, Dassaul Systemes dengan CATIA (Gambar 1.3), PTC dengan ProEngineer dan EDS dengan Unigraphics.



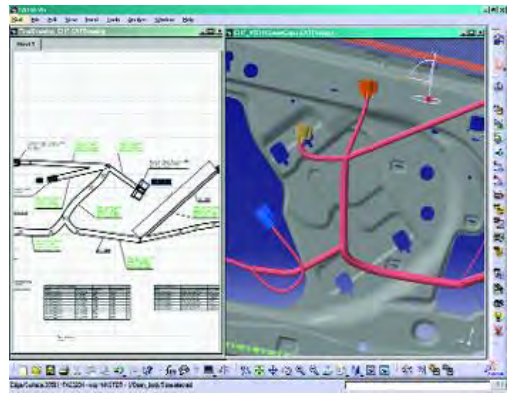
CAD Modeling



CAE (Finite Element Analysis)



Machining



System routing

Gambar 1.3 CATIA V5 yang merupakan Sistem CAD/CAE/CAM terintegrasi.

Alat bantu manajemen dan penyusunan strategi dalam industri yang terpenting adalah :

- **CIM (Computer Integrated Manufacturing)**

Berfungsi memadukan dan mengkoordinasikan perancangan, manufaktur dan manajemen berbasis komputer

- **CAP (Computer Aided Planning) & CAPP (Computer Aided Process Planning)**

Berperan dalam mengatur aliran pekerjaan secara efisien, termasuk menghasilkan aliran produksi yang optimal.



BAB 2 PROGRAM APLIKASI PENGOLAH KATA

Tujuan Pembelajaran Umum

1. Dapat memahami bagaimana menggunakan program aplikasi pengolah kata untuk keperluan teknik dan penulisan laporan.

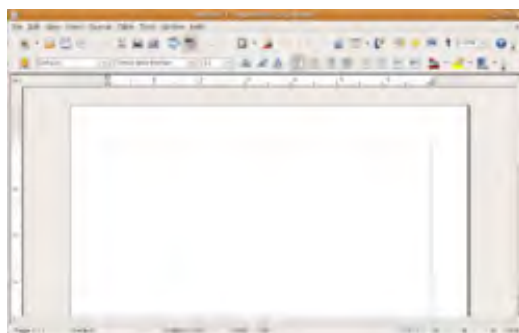
Tujuan Pembelajaran Khusus

1. Mahasiswa dapat menggunakan aplikasi pengolah kata lebih lanjut dalam hal ini membuat layout dokumen dengan pengaturan lanjut menggunakan section formatting.
2. Mahasiswa dapat menggunakan dan membuat *template* dan bekerja dengan aturan penulisan yang konsisten dengan menggunakan *style formatting* dan otomasi.

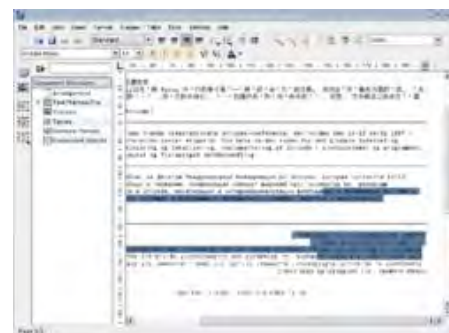
2.1 Dasar Aplikasi Pengolah Kata (*Word Processing*)

Pengolah kata yang berasal dari kata *word processing* adalah merupakan program aplikasi komputer yang menghasilkan makalah tertulis dengan format tertentu dengan proses meliputi pengaturan layout, format tulisan, mengedit tulisan dan mencetak dokumen. Dengan kata lain melakukan proses penyiapan dokumen (*document preparation*).

Aplikasi pengolah kata banyak tersedia baik yang bersifat *opensource* maupun yang komersial. Saat ini aplikasi komersial yang cukup banyak digunakan adalah **Microsoft Word**. Microsoft mengestimasi pengguna Microsoft Office melebihi 500 juta pengguna, termasuk di dalamnya Microsoft Word. Program aplikasi komersial lain adalah **Word Perfect** yang pernah mendominasi pasar pada pertengahan 1980-an hingga akhir 1990-an. Sedangkan aplikasi *opensource* tersedia seperti **OpenOffice.Org Writer**, **AbiWord**, **KWord** dan **Lyx**. Contoh tampilan antarmuka program aplikasi pengolah kata pada Gambar 2.1 dan Gambar 2.2.

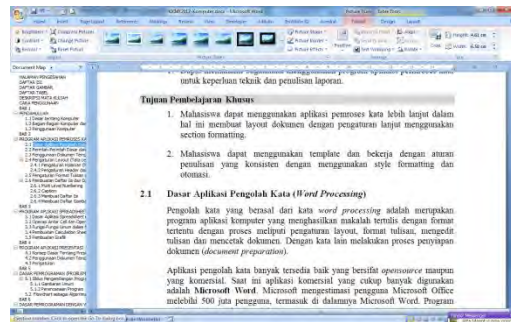


OpenOffice.Org Writer

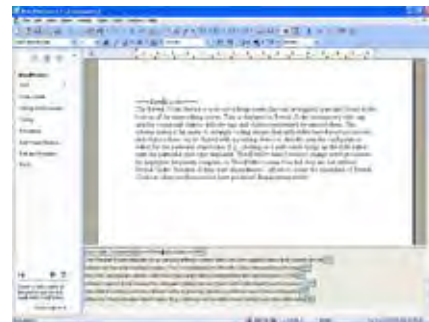


KWord

Gambar 2.1 Antarmuka aplikasi pengolah kata berbasis *opensource*.



Microsoft Word 2007



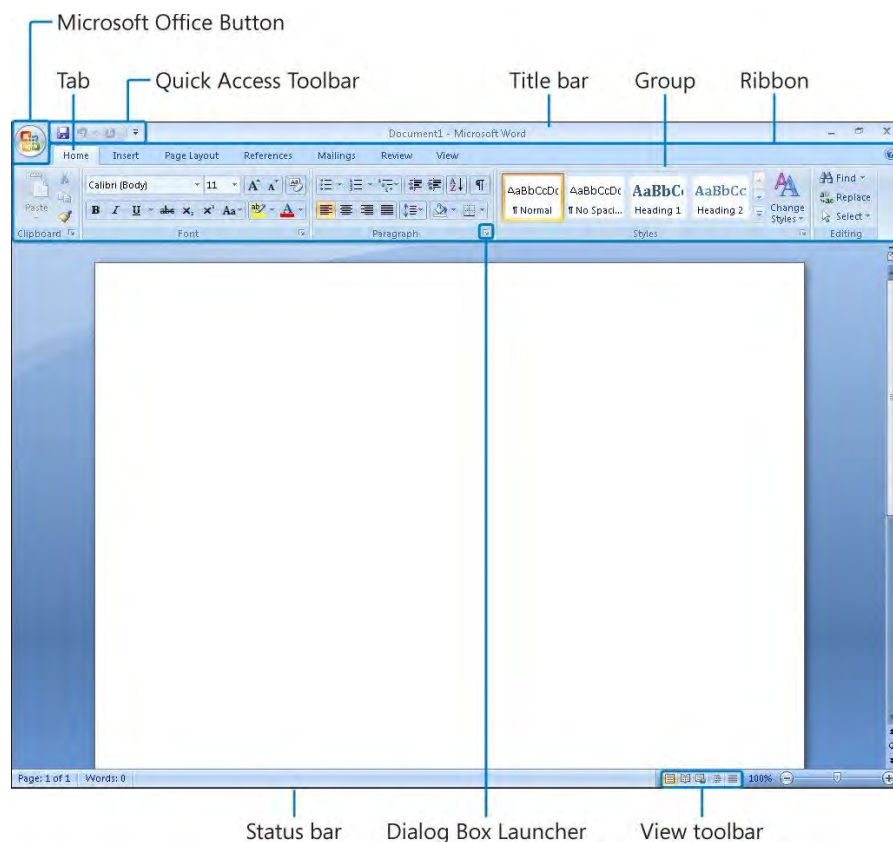
WordPerfect X3 dengan *Reveal code*

Gambar 2.2 Antarmuka program aplikasi pengolah kata komersial.

Dalam buku ajar ini akan digunakan aplikasi *Microsoft Word 2007* sebagai aplikasi pengolah kata.

2.2 Perintah-Perintah Dasar dalam Microsoft Word 2007

Microsoft Word 2007 dilihat dari tampilan antarmukanya sangat berbeda dengan versi-versi sebelumnya. Tampilan menu sudah menggunakan teknologi Microsoft Windows yang baru yaitu **Ribbon Bar**. Dengan keberadaan Ribbon Bar sangat mempermudah pengguna untuk mencari dan mengakses perintah-perintah.



Gambar 2.3 Antarmuka Aplikasi Microsoft Word 2007

Title Bar berisi nama file yang sedang dikerjakan, serta tombol menampilkan, atau menyembunyikan jendela program, dan menutup program, yaitu tombol **Minimize**, **Maximize/Restore**, dan **Close**.

Microsoft Office Button berisi perintah-perintah yang berkaitan dengan dokumen seperti membuat dokumen baru (**New**), membuka dokumen (**Open**), menyimpan dokumen (**Save**), mencetak dokumen (**Print**), dan sebagainya (Gambar 2.4).



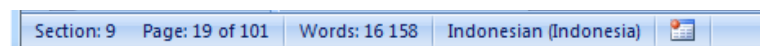
Gambar 2.4 Menu-menu dalam Office Button

Quick Access Toolbar berisi tombol-tombol yang berfungsi sebagai alternatif penggunaan perintah yang sering digunakan. Sebagai contoh, tool **Open** merupakan *shortcut* dari **Office Button>Open**, atau tool **Save** merupakan *shortcut* dari **Office Button>Save**.

Tab (Menu Bar) berisi serangkaian perintah yang didalamnya terdapat sub-sub perintah sesuai kategorinya. Sebagai contoh, pada menu **Home** terdapat submenu **Clipboard**, **Font**, **Paragraph**, **Styles**, dan **Editing** yang didalamnya berisi perintah-perintah sesuai kategorinya.

Ribbon berisi perintah-perintah khusus yang merupakan submenu dari **Menu Bar**.

Status Bar adalah baris horizontal yang menampilkan informasi jendela dokumen yang sedang ditampilkan, antara lain (Gambar 2.5):



Gambar 2.5 Status Bar

View Toolbar berisi pengaturan jenis tampilan dokumen, antara lain:

- **Print Layout:** memberikan tampilan sesuai hasil yang akan diterima padasaat dicetak (default).
- **Full Screen Reading:** memberikan tampilan halaman penuh.

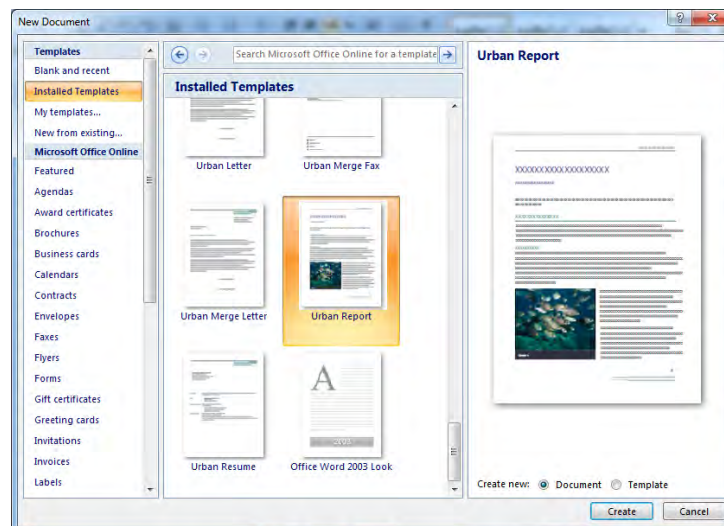
- **Web Layout:** memberikan tampilan sesuai hasil yang ditampilkan didalam jendela browser.
- **Outline:** memberikan tampilan sesuai heading di dalam dokumen dan tingkatan di dalam struktur dokumen.
- **Draft:** memberikan tampilan yang berkelanjutan (menggabungkan seluruh halaman seperti dalam satu gulungan). Tiap halaman dipisahkan oleh garis titik-titik.

Secara detail bagaimana memulai membuka aplikasi, menyimpan dokumen, mencetak dokumen dan lain-lain perintah dasar penggunaan Microsoft Office tidak akan dibahas secara rinci di dalam buku ajar ini. Yang akan dibahas hanyalah beberapa sub bahasan penting saja yang berkaitan dengan teknik dan penulisan laporan.

2.3 Penggunaan Dokumen *Template*

File dokumen yang disimpan oleh Microsoft Word 2007 mempunyai format file <Nama File>.docx untuk standard dokumen, dan <Nama File>.dotx untuk dokumen *template*. Dokumen *template* digunakan untuk mempermudah pengguna dalam mempersiapkan dokumen sesuai standard format dokumen baku yang harus diikuti. Contoh standard dokumen baku misalnya penulisan Tugas Akhir, Thesis, Desertasi, Laporan Teknik di perusahaan dan lain-lain.

Dokumen *template* secara default sudah disediakan oleh program MS Word 2007 yang dapat diakses melalui **Office Button** → **New**. Maka akan tampil kotak dialog membuat dokumen baru dengan berbagai *template* yang telah tersedia (Gambar 2.6). Microsoft menyediakan *template-template* standard secara percuma atau kita juga bisa membuat sesuai kebutuhan.



Gambar 2.6 Kotak dialog membuat dokumen baru dengan pilihan *template* yang ada.

Setelah masuk ke dalam dokumen baru dengan format dari *template* yang digunakan, maka pada dokumen tersebut akan terdapat keterangan format-format yang digunakan. Pada umumnya terdapat beberapa komponen yang bisa diisi langsung, dan ada bagian format yang dituliskan. Umumnya style yang

diberikan sebagai contoh adalah Heading 1, Heading 2, Heading 3 dan Caption. Tetapi biasanya format yang lain bisa dilihat pada **Styles Quick Gallery** seperti ditampilkan pada Gambar 2.7.



Gambar 2.7 Styles Quick Gallery untuk mengakses dengan cepat format style dari paragraf dokumen.

Catatan:

- ✓ Dianjurkan untuk menggunakan format dengan **STYLE** dalam pembuatan dokumen dengan MS Word 2007. Penggunaan format dengan Style membuat dokumen menjadi lebih konsisten.
- ✓ Format manual hanya dilakukan jika formatnya memang khusus dan tidak banyak digunakan dalam dokumen.

Contoh Penggunaan Template

Buka file **Template.dotx** yang diberikan.

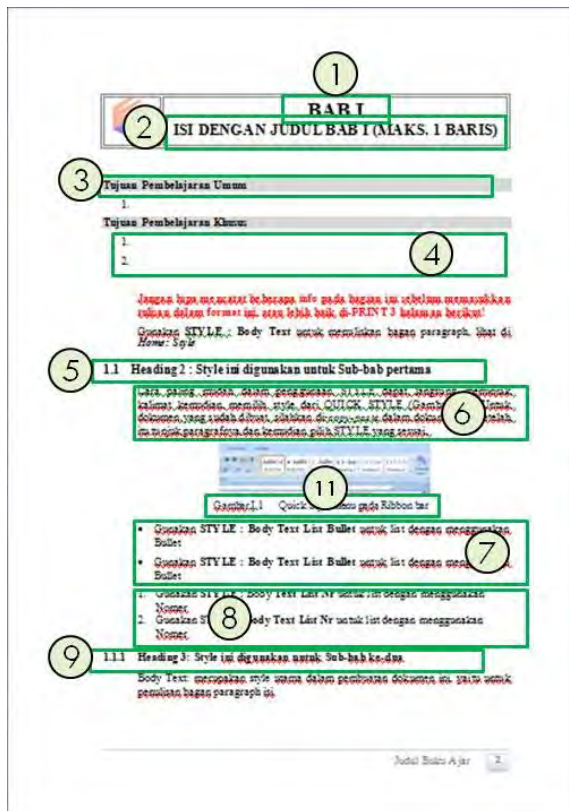
Pada saat membuka file template dengan klik double, maka secara otomatis MS Word akan membuat dokumen baru dengan format yang sesuai dengan template-nya.

Perhatikan Gambar 2.8. Di sebelah kiri adalah gambar layout dokumen yang telah dibuat dengan contoh template. Bagian paragraf dengan format style diberikan tanda kotak dengan penomoran. Bagian kanan adalah nama style yang digunakan sesuai dengan nomer-nya.

Lakukan beberapa perintah berikut untuk memahami bagaimana format dalam contoh template ini bekerja:

1. Buat Bab baru dari dokumen dengan meletakkan kursor di halaman ketiga dari dokumen, kemudian dari Styles Quick Gallery pilih Style **Heading 1**.
2. Setelah di ENTER, buatlah sebuah paragraf dan kemudian tekan ENTER.
3. Buat Sub-bab pertama dengan memilih style **Heading 2**.
4. Buat Sub-bab berikutnya dengan memilih style **Heading 3**, lakukan hal ini 2 kali atau lebih.

5. Pilih style Heading 2 untuk membuat sub-bab berikutnya.
6. Ulangi langkah 1-5



1. Heading 1
2. Title
3. Tujuan
4. Tujuan-Isi
5. Heading 2
6. Body Text
7. Body Text List Bullet
8. Body Text List Nr.
9. Heading 3
10. Heading 4 ... Dst ← Higher Level
11. Caption
12. TOC1, TOC2 dst ← Table of Content
13. Table of Figure

Gambar 2.8 Format dokumen dengan menggunakan *style formatting*.

Perhatikan!

- ✓ Apa yang bisa anda simpulkan tentang perubahan angka pada **Heading 2** dan **Heading 3**?

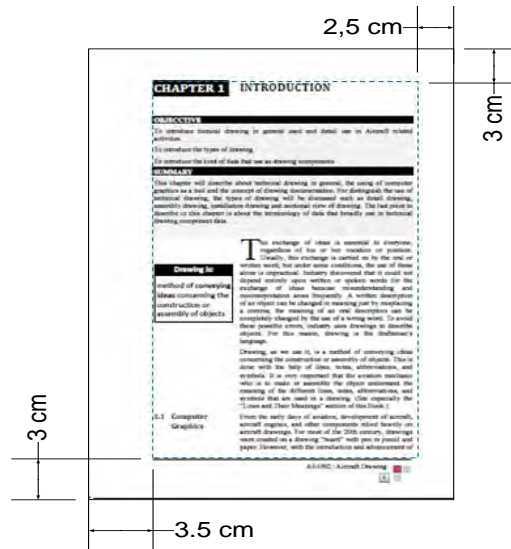
Setelah mengetahui bagaimana cara menggunakan dokumen template yang sudah ada, berikut akan dibahas langkah apa saja yang harus dilakukan untuk menyiapkan sebuah dokumen dengan format baku mulai dari awal.

Untuk menyiapkan dokumen berformat style beberapa langkah berikut harus dilakukan:

1. Pengaturan Layout (Tata letak kertas kerja)
2. Pengaturan format tulisan dengan STYLE
3. Membuat label untuk caption
4. Membuat Daftar isi, daftar gambar dan gambar tabel secara otomatis.
5. Menambahkan beberapa komponen untuk menambahkan informasi dokumen yang tersedia melalui developmen tools.

2.4 Pengaturan Layout (Tata Letak Kertas Kerja)

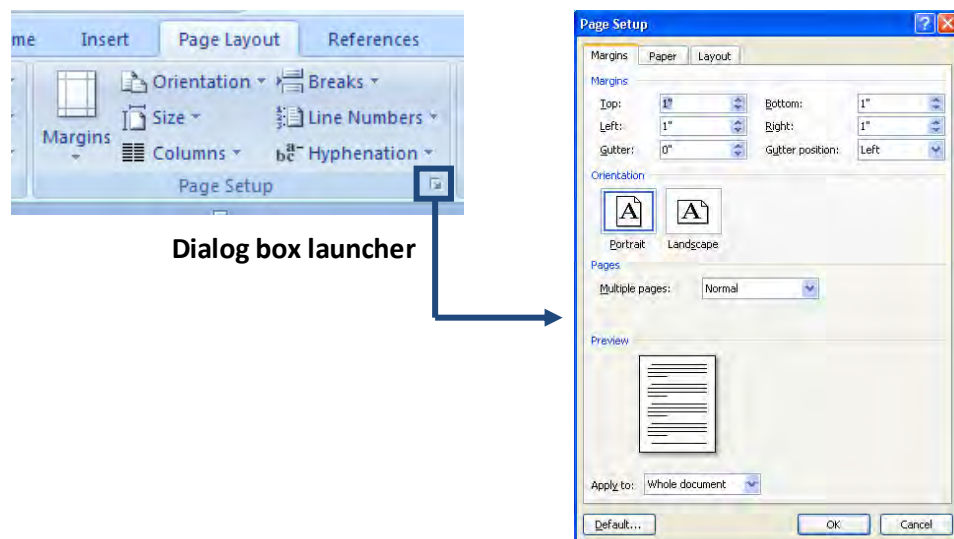
Langkah pertama yang penting adalah mengatur tata letak kertas kerja (layout tulisan). Ilustrasi pada Gambar 2.9 memberikan gambaran bagaimana dokumen harus di atur. Bagian ini mengatur margin dan section untuk header dan footer.



Gambar 2.9 Tampilan layout dokumen yang akan disiapkan.

2.4.1 Pengaturan Halaman (*Page Layout*)

Pengaturan halaman dapat dilakukan dari menu tab **Page Layout** dan kemudian pilih **Page Setup** melalui tombol **dialog box launcher**.



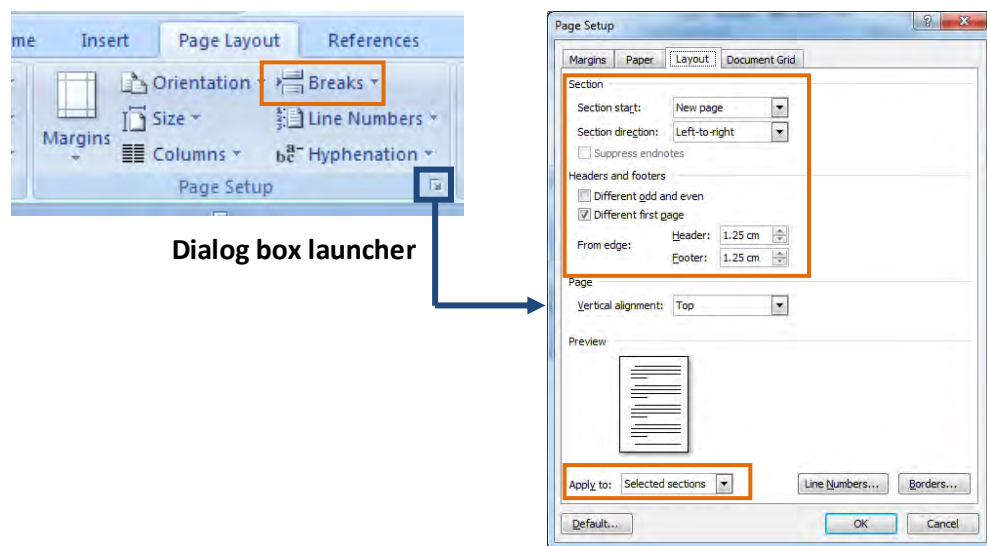
Gambar 2.10 Kotak dialog untuk melakukan pengaturan margin dan jenis kertas yang digunakan.

2.4.2 Pengaturan *Header* dan *Footer* (Section)

Pengaturan layout selanjutnya adalah berkenaan dengan **section**. Section merupakan bagian dari pengaturan layout dokumen berkaitan dengan perubahan-perubahan yang terjadi pada layout karena:

1. Berubahnya orientasi halaman dari portrait menjadi lanscape.
2. Berubahnya aturan header maupun footer antara halaman ganjil dan genap. (Gambar 2.12 section 2)
3. Berubahnya cara pemberian halaman yang berbeda, contoh dari i, ii, iii, ... menjadi 1, 2, 3, ... (Gambar 2.12: dari section 1 ke section 2)

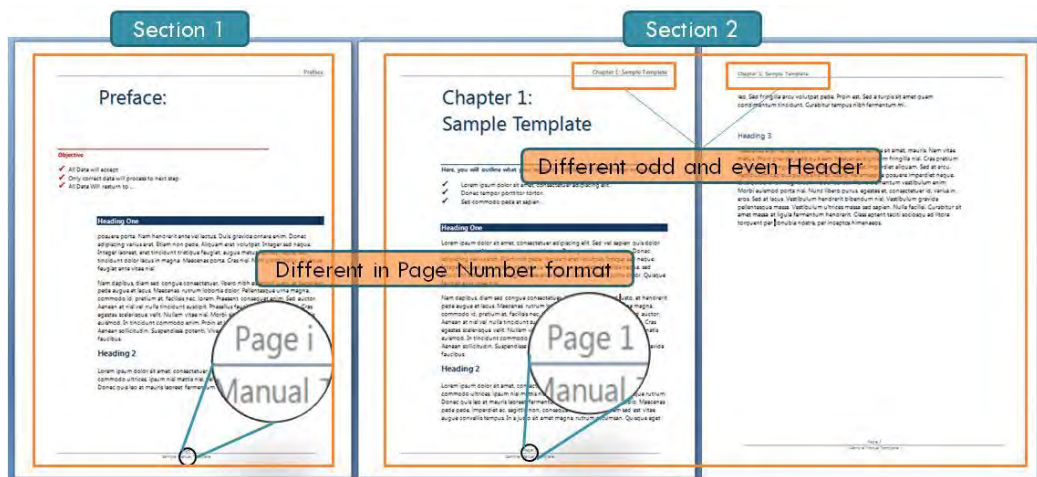
Cara menambahkan section melalui menu tab **Page Layout** dan kemudian pilih **Breaks** → **next page**. Setelah muncul kotak dialog Page Setup pilih tab **Layout**.



Gambar 2.11 Kotak dialog untuk melakukan pengaturan section, header dan footer.

Pengaturan **section** harus direncanakan berdasarkan kebutuhan layout dan pengaturan tampilan **header** dan **footer**. Sebagai contoh pada Gambar 2.12 direncanakan dokumen mempunyai sistem penomoran halaman yang berbeda sehingga *layout* dokumen harus terdiri dari 2 (dua) *section*. Kemudian di *section* yang kedua diinginkan mempunyai aturan *header* yang berbeda antara halaman ganjil dan genap karena direncanakan akan dicetak bolak-balik sehingga aturan header pada section yang ke dua harus dipilih **different odd and even** dan pengaturan halaman harus dipilih **mirror margin**.

Untuk mengatur sistem penomoran halaman yang berbeda harus diperhatikan aturan pada **page number**. Sedangkan untuk mengatur header dan *footer* harus diperhatikan pengaturan navigasi apakah *section* yang berbeda mempunyai *header* atau *footer* yang sama. Jika direncanakan berbeda, maka harus dihilangkan pilihan **Link to previous** dihilangkan dari pengaturan navigasi.



Gambar 2.12 Ilustrasi pengaturan Header dan footer untuk section yang berbeda maupun dalam section yang sama tetapi beda antara halaman ganjil dan genap.

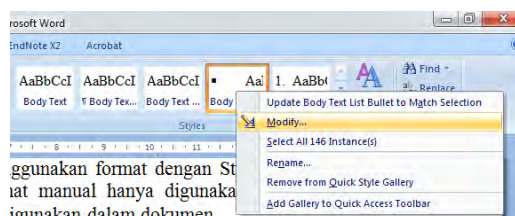
2.5 Pengaturan Format Tulisan dengan STYLE

Seperti telah diinformasikan pada catatan sebelumnya bahwa dalam penulisan dokumen laporan sebaiknya menggunakan format dengan Style agar penulisan bisa konsisten. Pengaturan format manual hanya digunakan jika dibutuhkan format khusus dan tidak banyak digunakan dalam dokumen.

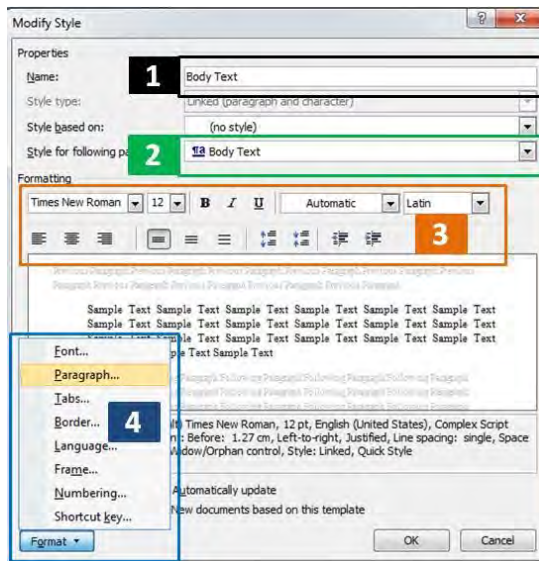
Secara default Microsoft Word telah menyediakan style-styles standard yang bisa langsung digunakan seperti: Heading 1, Heading 2, dst., Body Text, Caption, TOC 1, TOC 2, dst., dan lain-lain. Untuk mempermudah pengaturan style dianjurkan untuk menggunakan style yang sudah ada kemudian di edit sesuai aturan yang dikehendaki.

Cara mengakses untuk mengedit style yang sudah ada bisa dengan salah satu cara berikut:

1. Klik kanan nama style dari **Quick Gallery** kemudian pilih **Modify**. (Gambar 2.13)
2. Klik **Styles Dialog Box Launcher** kemudian pilih menu di bawah daftar style icon **Manage Styles**. (Gambar 2.15)



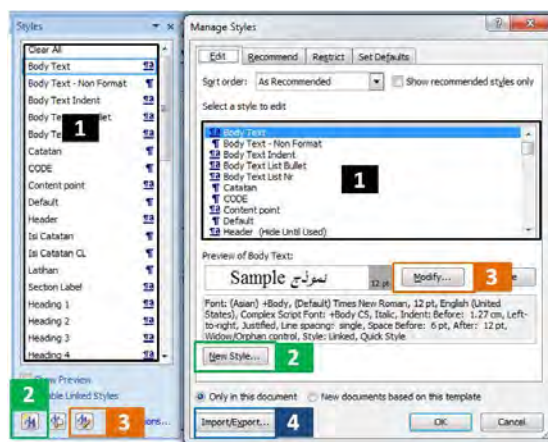
Gambar 2.13 Cara melakukan modifikasi pada style yang sudah ada dari **Styles Quick Gallery**.



- 1** Nama style yang akan dimodifikasi
- 2** Style yang akan diaktifkan saat ditekan ENTER
- 3** Format umum style
- 4** Format lebih rinci yang bisa dilakukan terhadap style

Gambar 2.14 Kotak dialog untuk modifikasi style.

Semua fasilitas format untuk paragraf tersedia di dalam style formatting. Perhatikan Gambar 2.14. Di keterangan bagian nomor 4 terlihat bahwa ada 7 jenis format yang bisa lakukan yaitu; Font, Paragraf, Tabs, Border, Language, Frame dan Numbering. Dari semua format paragraf yang merupakan pengaturan paling penting karena berhubungan dengan pengaturan posisi di layout.



- 1** Daftar style yang tersedia
- 2** Membuat style Baru
- 3** Memodifikasi style yang ada
- 4** Import/export style antar dokumen

Gambar 2.15 Cara melakukan modifikasi style atau membuat style baru dari Manage Styles.

2.6 Pembuatan Daftar Isi dan Daftar Gambar Secara Otomatis

Jika kita sudah membuat tulisan dengan aturan yang konsisten, maka kemudahan yang akan didapatkan adalah pembuatan **daftar isi**, **daftar gambar** dan **daftar tabel** secara otomatis. MS Word 2007 telah menyediakan fasilitas pembuatannya secara otomatis dengan syarat pendefinisian yang terkait harus sudah dilakukan di awal. Untuk membuat daftar isi yang paling mudah adalah mengatur **Multilevel numbering**. Pada dasarnya *multilevel numbering* memberikan informasi tentang urutan penomoran yang berjenjang dan memanfaatkan style yang sudah ada dan begitu pula dengan daftar isi yang

memanfaatkan style. Sedangkan daftar gambar dan daftar tabel berkaitan dengan pemberian **caption** pada gambar dan tabel.

2.6.1 Multilevel Numbering

Multi level numbering adalah sistem penomoran berjenjang yang secara otomatis berurut sesuai dengan levelnya. Di Microsoft Word menggunakan istilah **Multilevel List**. Sebagai ilustrasi perhatikan Gambar 2.16 dan Gambar 2.17 yang menunjukkan cara penulisan penomoran yang berbeda yaitu nomor angka berjenjang dan gabungan huruf dan angka.

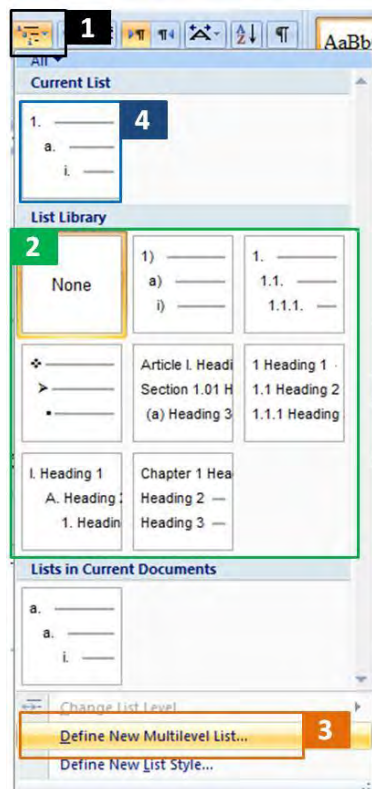
CHAPTER 1 (Level-1)	CHAPTER 2 (Level-1)
1.1 Heading-2 (Level-2) 1.1.1 Heading-3 (Level-3) 1.1.2 Heding-3 (Level-3) 1.1.2.1 Heading-4 (Level-4) 1.1.2.2 Heading-4 (Level-4) 1.1.3 Heading-3 (Level3) 1.2 Heading-2 (Level-2) 1.2.1 Heading-3 (Level-3) 1.2.2 Heding-3 (Level-3) 1.2.2.1 Heading-4 (Level-4) 1.2.2.2 Heading-4 (Level-4) 1.2.3 Heading-3 (Level3)	2.1 Heading-2 (Level-2) 2.1.1 Heading-3 (Level-3) 2.1.2 Heding-3 (Level-3) 2.1.2.1 Heading-4 (Level-4) 2.1.2.2 Heading-4 (Level-4) 2.1.3 Heading-3 (Level3) 2.2 Heading-2 (Level-2) 2.2.1 Heading-3 (Level-3) 2.2.2 Heding-3 (Level-3) 2.2.2.1 Heading-4 (Level-4) 2.2.2.2 Heading-4 (Level-4) 2.2.3 Heading-3 (Level3)

Gambar 2.16 Contoh multilevel list dengan penomoran berjenjang.

I. (Level-1)	II. (Level-1)
A. Heading-2 (Level-2) 1. Heading-3 (Level-3) 2. Heding-3 (Level-3) a) Heading-4 (Level-4) b) Heading-4 (Level-4) 3. Heading-3 (Level3) B. Heading-2 (Level-2) 1. Heading-3 (Level-3) 2. Heding-3 (Level-3) a) Heading-4 (Level-4) b) Heading-4 (Level-4) 3. Heading-3 (Level3)	A. Heading-2 (Level-2) 1. Heading-3 (Level-3) 2. Heding-3 (Level-3) a) Heading-4 (Level-4) b) Heading-4 (Level-4) 3. Heading-3 (Level3) B. Heading-2 (Level-2) 1. Heading-3 (Level-3) 2. Heding-3 (Level-3) a) Heading-4 (Level-4) b) Heading-4 (Level-4) 3. Heading-3 (Level3)

Gambar 2.17 Contoh multilevel list dengan penomoran gabungan huruf dan angka.

Cara melakukan pengaturan **Multilevel List** dapat melalui kelompok Ribbon Paragraph di Tab Home, maka akan muncul kotak dialog seperti pada Gambar 2.18. Untuk mempermudah pilih salah satu format multilevel list yang tersedia. Kemudian dimodifikasi sesuai dengan yang diinginkan.



- 1 Icon Multilevel Lists
- 2 Format Multilevel List yang tersedia
- 3 Mendefinikan ulang multilevel list yang akan digunakan
- 4 Multilevel List yang digunakan saat ini

Gambar 2.18 Menu di Multilevel List yang bisa digunakan.

2.6.2 Caption

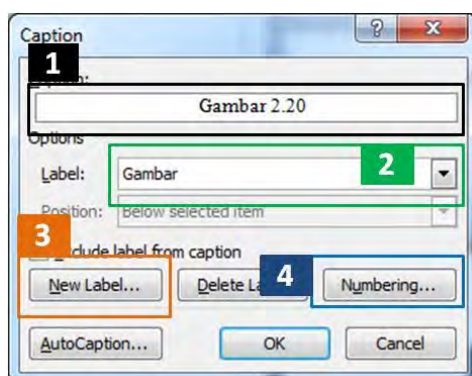
Caption merupakan fitur yang sangat bermanfaat bukan hanya untuk memberikan hubungan antar teks yang ada di dalamnya dengan daftar gambar yang akan dibuat, tetapi juga menginformasikan kepada pembaca tentang gambar, tabel, bagan apa yang sedang ditampilkan. Membuat caption sebaiknya **dengan kalimat yang bisa menjelaskan lebih** bukan hanya sekedar memberikan nama gambar.



- 1 Menu Tab References
- 2 Icon Insert Caption
- 3 Kotak dialog caption
- 4 Caption "Gambar"

Gambar 2.19 Contoh caption dan cara mengakses perintahnya.

Kita bisa membuat caption tidak hanya dengan label yang sudah disediakan, figure, table dan lain-lain, tetapi juga bisa menambahkan label yang lain sesuai dengan kebutuhan kita, seperti misalnya gambar atau tabel. Bagaimana cara membuat dan ilustrasi tentang caption dapat dilihat pada Gambar 2.19. Setelah gambar di masukkan ke dalam dokumen, kemudian di bawah gambar tersebut di letakkan cursor dan pilih perintah insert caption. Pilih label yang sesuai, jika tidak ada, maka bisa membuat label baru dengan memilih **New Label...**, tipe penonoran bisa diatur melalui tombol **Numbering...**

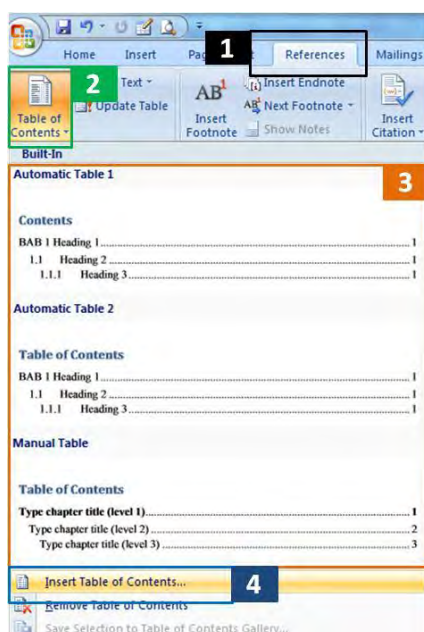


- 1** Contoh tampilan caption yang akan dibuat
- 2** Drop down daftar label untuk caption yang tersedia
- 3** Tombol untuk membuat label jika belum ada di daftar label
- 4** Cara penomoran caption.

Gambar 2.20 Kotak dialog caption.

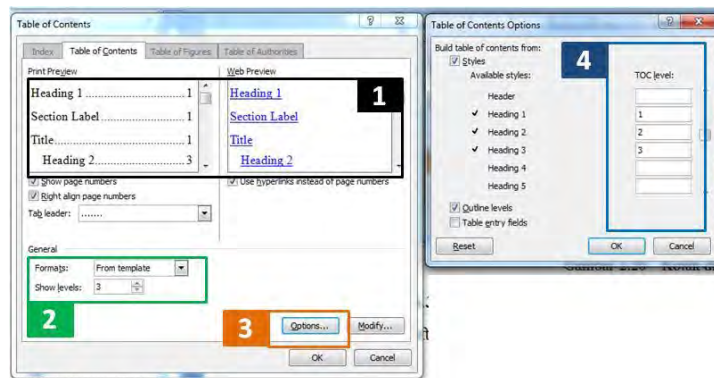
2.6.3 Membuat Daftar Isi

Daftar isi (*table of contents*) berfungsi untuk menampilkan garis besar (*outline*) dokumen yang merupakan topik utama dan dimana halamannya. MS Word membuat daftar isi berdasarkan style di dokumen yang bisa kita tentukan. Secara *default*, Heading 1 menjadi daftar di level-1, Heading 2 di level-2 dan seterusnya. Urutan pembuatannya bisa dilihat di Gambar 2.21 dan Gambar 2.22.



- 1** Menu Tab References
- 2** Icon perintah membuat Daftar isi
- 3** Format daftar isi yang ada
- 4** Perintah membuat daftar isi dengan format sesuai keinginan pengguna.

Gambar 2.21 Urutan perintah untuk membuat daftar isi secara otomatis

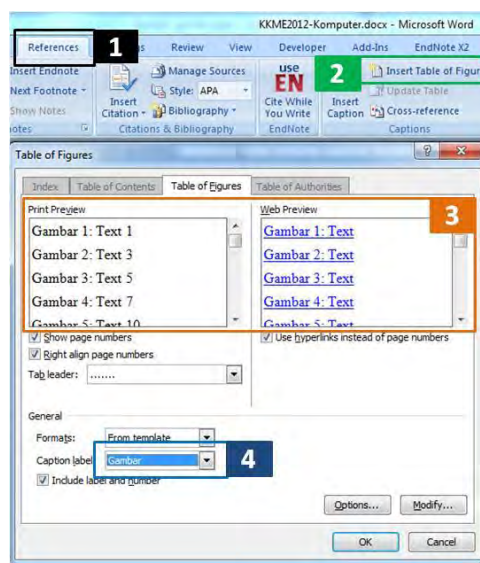


- 1 Contoh tampilan daftar isi yang akan dibuat.
- 2 Format tampilan yang diinginkan, from template berarti dari style template yang ada, TOC1, TOC2, ...
- 3 Tombol untuk menentukan style mana saja yang akan dimasukkan dalam daftar isi dan di level berapa.
- 4 Level dari style yang dipilih untuk dimasukkan dalam daftar isi.

Gambar 2.22 Cara mengatur pembuatan daftar isi sesuai format yang diinginkan dan style mana saja yang termasuk dalam daftar isi.

2.6.4 Membuat Daftar Gambar

Jika kita membuat dokumen yang mengandung banyak ilustrasi (gambar, foto, bagan, diagram dan lain-lain), maka akan memudahkan pembaca untuk menelusuri gambar jika terdapat daftar gambar. **Daftar gambar** (*table of figures*) seperti daftar isi tetapi memuat tentang keterangan gambar atau table saja. Dengan kata lain, style yang diidentifikasi hanya satu yaitu **caption**.



- 1 Menu Tab References
- 2 Icon untuk membuat daftar gambar
- 3 Contoh tampilan daftar gambar yang akan dibuat
- 4 Caption label yang akan dibuat daftarnya, misal daftar gambar → label gambar.

Gambar 2.23 Urutan perintah untuk membuat daftar gambar/tabel secara otomatis



BAB 3

PROGRAM APLIKASI SPREADSHEET

Tujuan Pembelajaran Umum

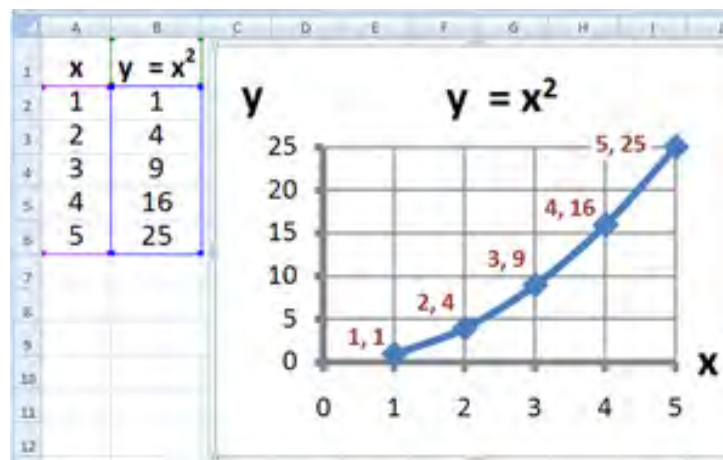
1. Dapat menggunakan program aplikasi spreadsheet untuk keperluan enjiniring yang efektif.

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat menggunakan fungsi-fungsi standard dalam program spreadsheet untuk digunakan sebagai Lembar perhitungan teknik.
2. Mahasiswa dapat merepresentasikan data dalam aplikasi spreadsheet dengan grafik-grafik yang sesuai.

Menyusun dan menampilkan kumpulan data akan lebih mudah dipahami dalam bentuk tabel, grafik, bagan dan lain-lain dalam keperluan analisis kegiatan di pekerjaan sehari-hari. Sebagai contoh; lembaga pemasaran suatu produk ingin mengetahui gambaran penjualan produk di semua daerah pemasarannya untuk mengambil keputusan target penjualan. Di bidang engineering, perhitungan *engineering* dapat ditampilkan dalam bentuk lembar perhitungan (*calculation sheet*)

Ketika data dalam jumlah kecil, mungkin peralatan hitung seperti kalkulator dapat digunakan. Tetapi jika data sudah mulai banyak, maka keterbatasan alat hitung sederhana menjadi terbatas. Dalam kasus ini, maka perhitungan data berpola tabel sangatlah bermanfaat. Aplikasi dengan metoda ini disebut dengan *spreadsheet*.



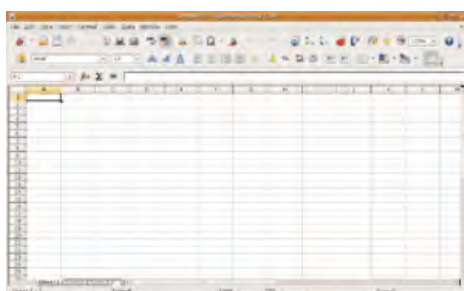
Gambar 3.1 Contoh sederhana pemanfaatan aplikasi spreadsheet untuk membuat grafik.

Cara kerja **spreadsheet** adalah dengan memanfaatkan pengoperasian **cell** yang mempunyai nomor baris (angka) dan nama kolom (huruf). Ilustrasi proses pengoperasiannya dapat dilihat pada Gambar 3.2.

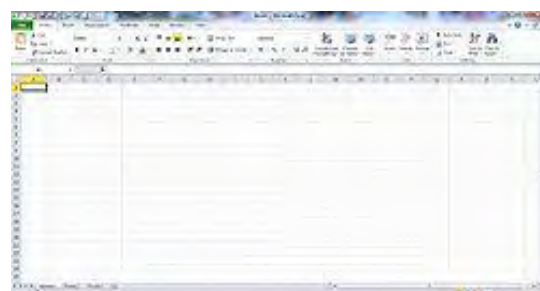
Nama kolom		A	B	C	D	E	F
1			Jan	Feb	Mar	Apr	TOTAL
2	Kripik pisang		54	58	64	78	254
3	Kripik kentang		78	80	75	83	316
4	Krupuk kulit		34	29	23	15	101
5	Krupuk Susu		15	35	43	47	140
6	Krupuk Kemplang		23	25	21	17	85
7	TOTAL		204	227	226	240	897
8				=SUM(C2:C6)			
9							
10			A =	4			=SUM(B4:E4)
11			B =	3			
12							
13			C =	7	=C10+C11		
14			D =	12	=C10*C11		

Gambar 3.2 Contoh pengoperasian bilangan dalam spreadsheet dengan formula tertentu. SUM = penjumlahan group. Operator + (tambah), * (kali) dan lainnya digunakan dalam operasi matematik.

Aplikasi spreadsheet tersedia baik yang bersifat opensource maupun yang bersifat komersial. Aplikasi bersifat *opensource* yang populer adalah **OpenOffice.Org Calc**, **Kspread** dan lainnya. Sedangkan yang bersifat komersial diantaranya adalah **Lotus Improv** (awalnya Lotus 1-2-3) sekarang dikembangkan IBM, **Quattro Pro** sekarang dikembangkan oleh Corel dan **Microsoft Excel**. Selain aplikasi spesifik yang telah disebutkan sebelumnya, di dalam aplikasi pengolah kata WordPerfect mempunyai tabel yang juga berfungsi sebagai spreadsheet, kecuali tidak bisa membuat grafik.



OpenOffice.Org Calc



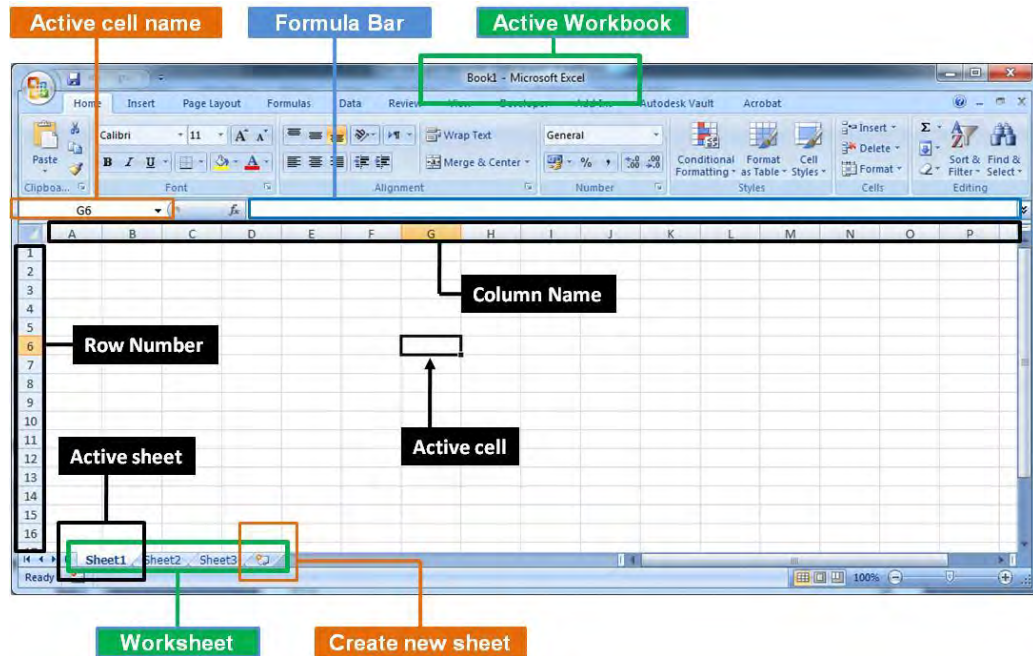
Microsoft Excel 2007

Gambar 3.3 Contoh antarmuka aplikasi spreadsheet

3.1 Dasar Aplikasi *Spreadsheet* dengan Microsoft EXCEL 2007

Microsoft Excel 2007 mengalami banyak perubahan dibandingkan dengan versi sebelumnya. Perubahan tidak hanya dari tampilan antarmukanya, tetapi

kapasitas dan kemampuannya juga mengalami peningkatan. Istilah menu-menu teknologi baru Microsoft dengan ribbon bar bisa merujuk lagi pada saat penjelasan tentang MS Word 2007. Bagian-bagian penting pada aplikasi ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan antarmuka aplikasi Microsoft Excel 2007.

Active workbook berisi informasi dokumen yang sedang dibuka.

Active cell name merupakan nama dari cell yang sedang dipilih.

Formula bar adalah tempat dimana kita dapat memberikan isi dari sebuah cell, atau kebalikannya; memberikan informasi isi (formula) yang berada pada cell yang aktif.

Worksheet adalah merupakan tempat dari cell dengan isi yang spesifik sesuai dengan nama yang diberikan. MS Excel menyediakan 3 (tiga) sheet secara default.

Perintah-perintah dasar menggunakan MS Excel yang mendasar tidak akan diungkap secara rinci dalam buku ajar ini. Beberapa perintah dasar hanya diberikan sekilas untuk mengulang saja. Pembahasan akan diberikan untuk hal prinsip dan beberapa yang terkait dengan keteknikan.

3.2 Memasukkan Data dan Operasi Dasar pada Cell

Sebelum membahas tentang operasi antar cell akan lebih baik kalau kita menyamakan persepsi kita tentang data dan tipe data yang bisa bekerja di excel. Memasukkan data di dalam worksheet sedikit berbeda dengan menuliskan text di dalam pengolah kata. Di excel kita tidak memasukkan semua data secara kontinyu tetapi harus mengisinya secara terpisah di dalam cell. Misalkan kita

punya 3 (tiga) data; Dadang, 200, 444; kemudian kita harus mengisikan data: dadang di sebuah cell dan pindah ke cell lain untuk mengisikan 200, begitu pula untuk data 444 setelah pindah ke cell yang lain.

Data yang dapat dimasukkan ke dalam cell secara umum dibagi menjadi 2(dua) jenis yaitu:

1. Konstanta
2. Formula (rumus)

3.2.1 Konstanta

Konstanta adalah sebuah nilai yang tidak akan berubah dengan sendirinya. Sebagai contoh misalnya kita memasukkan data di *cell* A1 dengan “Dadang”, nilai tersebut hingga kita dengan sengaja mengubahnya, misal mengubahnya dengan “diding”. Demikian juga dengan data berupa angka, misal 200 dan lain-lain. Dengan kata lain kita bisa menyebut data sebagai konstanta karena data tersebut tidak bisa berubah secara otomatis.

Selanjutnya konstanta dapat dikategorikan menjadi:

1. Angka (numbers)
2. Teks (text)
3. Tanggal (date)
4. Waktu (time)

Angka

Kombinasi yang disusun oleh karakter 0 – 9 yang dipelakukan sebagai angka atau konstanta numerik. Sebagai contoh, 200 adalah angka. Karakter lain dapat menyertai angka dengan maksud tertentu yaitu:

- (+) menunjukkan nilai positif, biasanya tidak dituliskan ($+275 = 275$)
- (-) menunjukkan nilai negatif (-275)
- (.) (titik) atau (,) (koma) menunjukkan pemisah desimal (2.75 atau 2,75)
- (%) menunjukkan persentase ($/100$) ($95\% = 95/100 = 0.95$)
- (\$, Rp, £ dan lain-lain) untuk menunjukkan mata uang sesuai format

Teks

Semua data yang bukan angka, tanggal dan waktu diperlakukan sebagai angka di MS Excel. Sebagai contoh Dadang, Dudung, Diding, January, Bulan dan lain-lain adalah contoh dari teks. Perlu diketahui bahwa dalam sebuah cell (mulai MS Excel 2007) bisa menampilkan hingga 32678 karakter termasuk spasi.

Tanggal

MS Excel mengijinkan penggunaan bentuk tanggal sebagai sebuah data yang diperlakukan sebagai identitas tersendiri. Dengan demikian perhitungan

matematik berbasis tanggal menjadi bisa dilakukan. Berikut adalah format tanggal yang betul:

22/06/1975, 16/MAY/1975, 10-3-1988 dan lain-lain.

Waktu

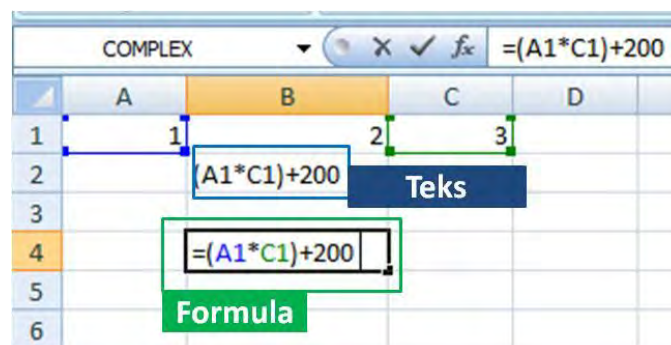
Waktu adalah bentuk angka yang dikenal sebagai format tersendiri. Untuk memasukkan data waktu dapat mengikuti aturan berikut:

1. Nilai waktu bisa dengan format 12 jam maupun 24 jam
2. Jika waktu berformat 12 jam, maka harus diikuti dengan AM atau PM. Sebagai contoh 7 AM, 8 PM dan lain-lain.
3. Disamping AM atau PM, A dan P juga bisa digunakan untuk menggantikannya. Sebagai contoh 7 A = 7 AM, 8 P = 8 PM.
4. Jika waktu meliputi jam, menit dan detik, maka pemisah yang digunakan adalah titik dua (:). Sebagai contoh 10:45 AM, 1:10:05 PM dan lain-lain.
5. Secara default MS Excel akan memberlakukan format 24 jam atau sebaliknya sesuai format cell yang digunakan.
6. Waktu dapat digunakan secara bersamaan dengan tanggal sebagai nilai data yang benar. Sebagai contoh 10/5/2006 11:55, 3/5/1957 10:20 PM dan lain-lain.

3.2.2 Formula

Formula pada dasarnya adalah fungsi pengguna (*user defined fuction*) yang dapat dibentuk sesuai kebutuhan dengan menggunakan konstanta, operator matematika dan fungsi standard yang disediakan oleh MS Excel. Sebagai contoh (A1*C1)+200 atau SUM(A1:A10) dan lain-lain.

Untuk membedakan penulisan formula dan teks, semua formula harus dimulai dengan lambang (=). Sebagai contoh formula pada paragraf sebelumnya dituliskan dalam cell sebagai =(A1*C1)+200 atau =SUM(A1:A10).



Gambar 3.5 Cara penulisan formula harus diawali dengan tanda (=) untuk membedakan dengan teks.

Berdasarkan fungsi, data yang digunakan dan hasilnya, formula secara umum dapat dikategorikan sebagai berikut:

1. Formula numerik
2. Formula logika
3. Formula teks

Berikut ini penjelasan singkat dari masing-masing kategori di atas:

Formula numerik

Formula yang menggunakan operator aritmatika disebut sebagai formula numerik. Sebagai contoh, $=(A+B)-C1$ adalah formula numerik karena menggunakan operator aritmatika (+) dan (-). Daftar lambang-lambang yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar lambang untuk operator aritmatika (matematika)

Lambang	Operasi yang dilakukan
+	Penjumlahan
-	Pengurangan (negatif jika diletakkan sebelum angka; -27)
/	Pembagian
*	Perkalian
%	Persentase (diletakkan setelah angka; 25%)
^	Pemangkatan (20^2)

Berikut adalah contoh penulisan formula dalam MS Excel:

$=(A1+B1+C1+D1)$, $=(A1*B1)/C1$, $=(C1+C5)*25\%$, $(5^3)+B1$

Formula Logika

Formula yang menggunakan operator logika yang menghasilkan nilai TRUE atau FALSE. Sebagai contoh $=(A1>B1)$ yang merupakan formula logika karena menggunakan tanda (>) yang merupakan operator logika. Daftar operator logika dapat dilihat pada Tabel 3.2.

Tabel 3.2 Daftar lambang operator logika

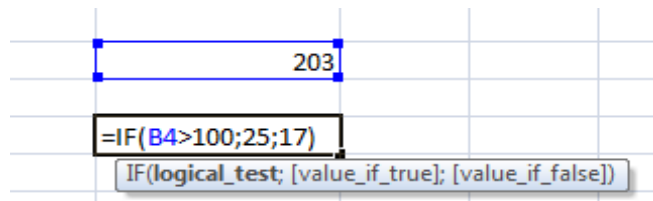
Lambang	Arti	Lambang	Arti
=	Sama dengan	>=	Lebih besar atau sama dengan
>	Lebih besar	<=	Lebih kecil atau sama dengan
<	Lebih kecil	<>	Tidak sama dengan

Untuk gambaran lebih jelas tentang penggunaan operator ini dan hasil yang didapatkan dapat dilihat pada Tabel 3.3. Misal $A1=1000$ dan $B1=500$.

Tabel 3.3 Contoh penggunaan operator logika

Cell	Formula	Hasil
C1	=(A1=B1)	FALSE
C2	=(A1>B1)	TRUE
C3	=(A1<>B1)	TRUE
C4	=(A1>=B1)	TRUE
C5	=(A1<B1)	FALSE
C6	=(A1<=B1)	FALSE

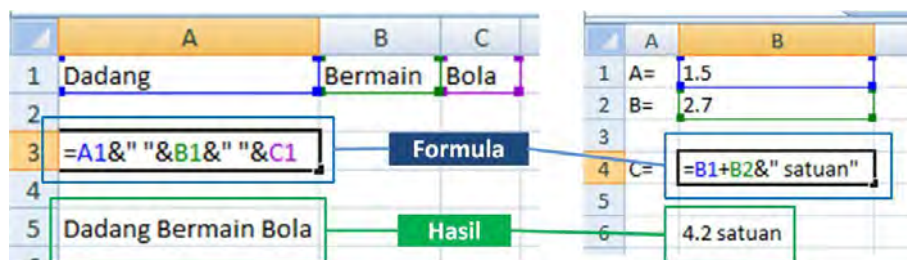
Penggunaan lebih lanjut dari operator logika digunakan bersama dengan fungsi IF. Gambaran contohnya dapat dilihat pada Gambar 3.6.



Gambar 3.6 Contoh penggunaan operator logika dengan fungsi IF. Dalam contoh ini nilai yang dihasilkan adalah 25, karena nilai logikanya adalah TRUE.

Formula Teks

Formula ini digunakan untuk mengolah data teks. Operator yang digunakan adalah tanda (&) yang berfungsi menggabungkan teks dari beberapa cell maupun teks dan angka.



Gambar 3.7 Contoh penggunaan formula teks

3.2.3 Cara penulisan referensi cell

Referensi yang dimaksud di adalah bagaimana menunjukkan cell yang digunakan dalam penulisan formula. Contoh =(A1+B1) atau =SUM(A1:A5). A1 dan B1 adalah referensi tunggal sedangkan A1:A5 adalah referensi dalam kelompok. Bayangkan jika tidak ada referensi kelompok, maka penulisan A1:A1000 harus dituliskan sebagai A1,A2,A3,...

Operator jangkauan (*Range operator*)

Dari namanya dapat diketahui bahwa referensi ini menunjukkan sebuah kelompok cell. Lambang yang digunakan adalah (:). Untuk menuliskan referensi ini diperlukan 2 (dua) operand, dituliskan sebagai Opr1:Opr2, dimana operan adalah batas-batas dari cell. Contoh penggunaannya dapat dilihat pada Gambar 3.8.

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4			
5	12	6	45
	=SUM(A1:A5)	=SUM(A1:C1)	=SUM(A1:C3)

Gambar 3.8 Contoh penggunaan operator jangkauan dalam formula.

Operator Gabungan (*Union operator*)

Operator gabungan adalah menggabungkan semua referensi yang dirujuk. Operator ini menggunakan lambang titik koma (;) untuk memisahkan setiap referensi, baik tunggal maupun kelompok. Contoh perhatikan Gambar 3.9.

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4			
5	16		18
	=SUM(A1;B1:B3)		=SUM(A3;B1:C1)

Gambar 3.9 Contoh penggunaan operator gabungan. Sebagai contoh A1=1 dan B1:B3 = 2, 5 dan 8 menghasilkan 16 jika disumasikan.

Operator Irisan (*Intersection operator*)

Merupakan referensi irisan antara umumnya 2 (dua) kelompok data yang diberikan. Sebagai contoh pada Gambar 3.10 ditunjukkan bahwa nilai (5, 6, 8 dan 9) adalah irisan dari kelompok nilai (4, 5, 6, 7, 8 dan 9) dan (2, 3, 5, 6, 8 dan 9).

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4			
5			28

=SUM(A2:C3 B1:C3)

Gambar 3.10 Contoh penggunaan operator irisan. Angka 5, 6, 8 dan 9 adalah irisan

Penulisan referensi dalam formula seperti ditunjukkan pada contoh-contoh sebelumnya adalah merupakan salah satu jenis penulisan referensi yang umum ditemukan untuk menuliskan formula dalam 1 (satu) cell. Jika penulisan formula hendak digandakan, maka perlu diperhatikan cara perujukan (*cell referencing*). Metoda penulisannya dibagi menjadi 3 (tiga) jenis yaitu:

- Perujukan relatif (Relative referencing)
- Perujukan absolut (Absolute referencing)
- Perujukan gabungan (Mixed referencing)

	A	B	C
1	Kecepatan benda jatuh		
2	g = 9.81	m/s	
3		posisi	kecepatan
4	0	=SQRT(2*\$B\$2*B4)	
5	1	=SQRT(2*\$B\$2*B5)	
6	2	=SQRT(2*\$B\$2*B6)	
7	3	=SQRT(2*\$B\$2*B7)	
8	4	=SQRT(2*\$B\$2*B8)	
9	5	=SQRT(2*\$B\$2*B9)	

Relatif

	A	B	C
1	I	II	III
2	0	=2*\$A2*A2/2	=2*\$A2*B2/2
3	1	=2*\$A3*A3/2	=2*\$A3*B3/2
4	2	=2*\$A4*A4/2	=2*\$A4*B4/2
5	3	=2*\$A5*A5/2	=2*\$A5*B5/2
6	4	=2*\$A6*A6/2	=2*\$A6*B6/2
7	5	=2*\$A7*A7/2	=2*\$A7*B7/2

Gabungan

➔ Arah peng-COPY-an

Gambar 3.11 Cara perujukan cell referensi dengan cara relatif, absolut dan gabungan.

Perujukan Relatif

Posisi perujukan cell akan berpindah ke cell rujukan lain yang relatif terhadap posisi cell yang berformula pada saat dilakukan penggandaan formula. Seperti contoh di Gambar 3.11 B4 yang terus berubah secara relatif hingga B9.

Perujukan Absolut

Posisi perujukan cell tidak akan berpindah ke cell lain pada saat dilakukan penggandaan formula. Perujukan ditandai dengan lambang (\$). Seperti contoh di

Gambar 3.11 \$B\$2 tidak mengalami perubahan walaupun digandakan dari posisi C4 sampai dengan C9. Bandingkan dengan B4.

Perujukan Gabungan

Perujukan juga bisa dilakukan dengan cara menggabungkan antara metoda absolut dan relatif. Dalam Gambar 3.11 sebelah kanan ditunjukkan rujukan \$A2 yang menyatakan secara absolut kolom A (\$A) dan relatif terhadap baris (2). Jika di-copy ke arah horisontal, maka \$A akan tetap karena absolut untuk kolomm sedangkan jika di-copy ke arah bawah, maka 2 akan berubah secara relatif terhadap barisnya (2 hingga 5). Sehingga menjadi \$A2, \$A3 dan seterusnya.

Catatan:

Perujukan absolut dapat juga dilakukan tanpa melakukan dengan metoda rujukan, tetapi dapat memberikan nama pada cell yang dirujuk secara absolute. Ubah nama cell dengan nama yang diinginkan.

Nama cell	
	phi
1	phi
2	radius
3	0
4	5
5	10
6	15
7	20
8	25

	A	B
1	phi	=PI()
2	radius	Luas
3	0	=phi*A3
4	5	=phi*A4^2
5	10	=phi*A5^2
6	15	=phi*A6^2
7	20	=phi*A7^2
8	25	=phi*A8^2

Rujukan absolut ke "phi"

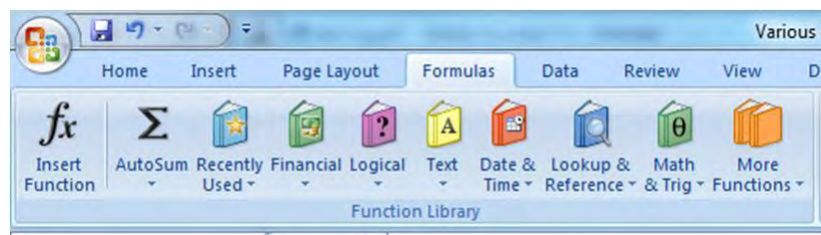
✓ Nama cell yang dirujuk seolah sebagai rujukan absolut. Dalam contoh disini adalah cell B1 yang diubah namanya menjadi "phi"

3.3 Fungsi-Fungsi Umum dalam Microsoft Excel 2007

MS Excel menyediakan fungsi standard yang bisa langsung digunakan. Perhatikan Gambar 3.12 yang menunjukkan kelompok fungsi-fungsi yang disediakan. Formula dituliskan dengan nilai yang diberikan langsung atau merujuk pada cell tertentu dan menghasilkan nilai tertentu pula. Format penulis formula adalah:

Nama Fungsi (argumen)

Nama fungsi adalah fungsi yang dimaksud sedangkan argumen berisi nilai yang biasanya batasi oleh titik koma (;). Sebagai contoh =SUM(A1:A5;B1:B5) atau =LEFT("Dadang",2) menghasilkan Da.



Gambar 3.12 Kelompok **Function Library** yang disediakan MS Excel dalam menu tab **Formulas**.

Untuk memahami beberapa fungsi dari prosedur, contoh dan hasil yang didapatkan dapat dilihat Tabel 3.4. Perhatikan cara penulisannya dan apa yang dihasilkan.

Tabel 3.4 Daftar sebagian fungsi yang umum digunakan yang disediakan oleh MS Excel

No	Fungsi	Prosedur	Contoh	Hasil
1	ABS(number)	Menghasilkan nilai absolut	=ABS(200)	200
2	INT(number)	Pembulatan ke arah bilangan bulat terdekat	=INT(200.37)	200
4	SQRT(number)	Menghasilkan nilai akar kuadrat	=SQRT(64)	8
5	SUM(number1; number2....)	Menambahkan semua bilangan	=SUM(140;30)	170
6	MOD(number1; number2...)	Menghasilkan sisa dari pembagian.	=MOD(127,5)	2
7	PRODUCT(number 1; number2...)	Perkalian semua bilangan.	=PRODUCT(5;6)	30
8	ROUND(number1; number2...)	Pembulatan nilai dengan desimal tertentu.	=ROUND(122.66;1) =ROUND(122.6679; 2)	122.7 122.66
9	AVERAGE(number 1; number2...)	Menghasilkan nilai rata-rata.	=AVERAGE(15;17)	16
10	COUNT(value1; value2)	Menghitung jumlah bilangan	=COUNT(1;8;4;9;1)	5
11	MAX(number1; number2)	Menghasilkan nilai tertinggi	=MAX(100;135)	135
12	MIN(number1; number2)	Menghasilkan nilai terendah	=MIN(100;135)	100
13	LEN(text)	Menghasilkan panjang karakter	=LEN("INDIA")	5
14	LEFT(text; value)	Mengambil sejumlah karakter dari kiri	=LEFT("INDIA",2)	IN
15	RIGHT(text; value)	Mengambil sejumlah karakter dari kanan	=RIGHT("INDIA",2)	IA

No	Fungsi	Prosedur	Contoh	Hasil
16	LOWER(text)	Mengubah semua karakter menjadi huruf kecil	=LOWER("INDIA")	india
17	UPPER(text)	Mengubah semua karakter menjadi huruf besar	=UPPER("India")	INDIA
18	TRIM(text)	Menghilangkan spasi di awal atau akhir teks	=TRIM(" Hi Dear")	Hi Dear
19	REPT(text, number of times)	Pengulangan teks	=REPT("S",5)	SSSSS

3.4 Fungsi LOOKUP

Fungsi LOOKUP dibahas secara khusus karena sangat penting untuk dipahami. Dengan fungsi ini kita dapat melakukan sorting sebuah nilai dari sebuah data tabulasi selayaknya pengambilan data dari sebuah *database*. Ada beberapa fungsi berdasarkan fungsi lookup yaitu LOOKUP, VLOOKUP, HLOOKUP, INDEX dan MATCH. Untuk pemahaman secara umum dalam bahasan ini hanya dibahas tentang VLOOKUP sedangkan HLOOKUP dapat dianalogikan dengan VLOOKUP dengan Baris sebagai data.

Fungsi VLOOKUP dituliskan sebagai berikut:

=VLOOKUP(lookup_value;table_array;col_index_num;[range_lookup])

Argumen dari fungsi tersebut terdiri dari 3 (argumen) dan 1 (satu) argumen pilihan (optional). Perhatikan Gambar 3.13.

Lookup_value: Nilai yang menjadi acuan (jenis material = API 5L - A)

Table_array: kelompok data yang akan disortir, termasuk di dalamnya kolom jenis material.

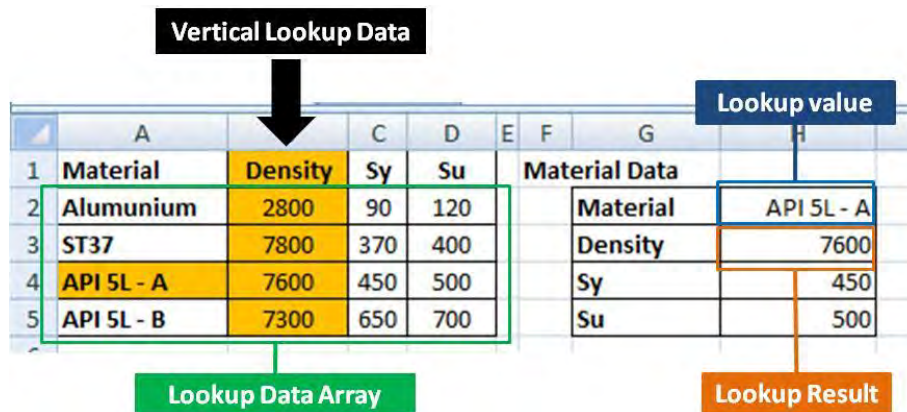
Col_index_num: nomor urutan kolom yang datanya akan disortir (dalam hal ini kolom density)

[range_lookup]: merupakan data optional yang terdiri dari 2 (dua) nilai yaitu TRUE dan FALSE. Argumen tambahan ini digunakan jika nilai acuan lookup tidak sama persis dengan nilai yang ada dan bagaimana cara pengambilan nilai di datanya, apakah pembulatan ke nilai terdekat (FALSE) atau perbandingan proporsional antar data yang ada (TRUE).

Untuk mendapatkan nilai seperti contoh pada Gambar 3.13, maka pada cell H3 dituliskan formula sebagai berikut:

=VLOOKUP(H2;A2:D5;2;FALSE)

Untuk mendapatkan nilai lain spt Sy maka hanya cukup mengubah **col_index_num** menjadi 3, dan untuk Su diubah menjadi 4 dan seterusnya.



Gambar 3.13 Ilustrasi cara kerja fungsi vlookup pada data properti material.

Catatan:

Untuk mendapatkan nilai yang sesuai pada pemilihan material di cell H2 pada Gambar 3.13, maka kelompok data material harus dibuat nama khusus misal material. Kemudian dengan menggunakan data validation berupa list pada cell H2 dan merujuk pada nama “material”.

Nama kelompok data

	A
1	Material
2	Alumunium
3	ST37
4	API 5L - A
5	API 5L - B

Daerah Kelompok data

- 1** Membuat nama pada kelompok data
- 2** Membuat list data pada cell pilihan
- 3** Jenis data validation → List
- 4** Source diarahkan pada nama kelompok data yang sudah dibuat (=material)

3.5 Pembuatan *Calculation Sheet*

Lembar perhitungan (*calculation sheet*) merupakan aplikasi dari MS Excel yang bisa dimanfaatkan dalam data teknik (*engineering data*). Dari penjelasan di atas tentang dasar penggunaan MS-EXCEL diharapkan bisa digunakan untuk membuat *calculation sheet*. Sebagai dasar pembuatannya, kita harus paham betul

apa saja yang termasuk dalam data masukan dan apa keluarannya dan kemudian proses apa saja yang harus dilakukan (Lihat bahasan pada BAB 5).

Contoh: Perhitungan dan Pengecekan tebal pipa (ASME B31.8)

Untuk menghitung tebal pipa berdasarkan standard yang ada diperlukan beberapa informasi sebagai berikut:

Output:

- Tebal Pipa (t_{req})
- Pengecekan tebal nominal (t_n) terhadap tebal yang dibutuhkan (t_{req})

Input:

- Tekanan Desain (P)
- Faktor Desain (F)
- Temperature Derating Factor (T)
- Corrosion Allowance (CA)
- Joint Factor (E)
- Pipe Outside Diameter (D)
- Material Properties (S)
- T nominal (t_n)

Proses perhitungan:

$$t_{req} = \frac{PD}{2SFET} + CA$$

Dari analisis di atas maka diperlukan beberapa informasi berikut:

Data masukan numerik :

- Tekanan Desain (P)
- Faktor Desain (F)
- Temperature Derating Factor (T)
- Corrosion Allowance (CA)
- Joint Factor (E)

Data masukan tabulasi:

- Pipe Outside Diameter (D), berdasarkan NPS berbeda untuk NPS 14 ke bawah.
- Material Properties (S), Berbeda setiap material

Dengan informasi di atas dapat dibuat sebuah lembar perhitungan dengan contoh bentuk seperti dapat dilihat pada Gambar 3.14. sheet lain yang dibutuhkan adalah data untuk nominal outside diameter (NPS) dan data properti material yang masing-masing berisi data kolom berupa OD dan SMYS.

Yang perlu dilakukan untuk menyiapkan perhitungan:

Penamaan data tabulasi:

- Pilih Formulas → Name Manager → New ...
- Nama: NPS dan Material

Fungsi lookup untuk OD (G22) dan SMYS (G25):

- =VLOOKUP(G21;NPS!B6:C27;2)
- =VLOOKUP(G24;Material!B3:C6;2)

Fungsi IF untuk mengecek apakah tebal nominal (inspeksi) OK atau TIDAK

- =IF(G29>G27;" OK";" NOT OK")

PT. POLBAN JAYA ABADI
THICKNESS CALCULATION

TITLE : SLUG CATCHER PAGE : 1 of 1
TAG NO. : - DATE : -
CLIENT : Pa Iping Company FILE : -
JOB NO. : - PROG. : XLS
REV. : -

DATA
DESIGN PRESSURE,
DESIGN FACTOR,
TEMP. DERATING FACTOR,
CORR. ALLOWANCE,
JOINT FACTOR,

DATA masukan numerik
P = 2500 PSIG
F = 0.5
T = 1
CA = 0.125 in
E = 1.00

PIPE SIZE : NPS 42
Pipe Outside Diameter 5

MATERIAL API 5L X65
SMYS 25000

REQUIRED PIPE THK :
 $t = P \cdot D / (2 \cdot S \cdot F \cdot E \cdot T) + CA = 0.6250$

NOM. PIPE THK (IN) : tn = 4.5
CONCLUSION OK

Worksheet yg diperlukan
Thickness Clients NPS Material

Gambar 3.14 Contoh bentuk calculation sheet untuk perhitungan pipe wall thickness berdasarkan ASME B31.8

3.6 Pembuatan Grafik

Menampilkan data dalam bentuk grafik merupakan salah satu cara yang sangat populer dan bermanfaat. Gambar dapat menjelaskan banyak hal dibandingkan dengan angka-angka. Beberapa jenis grafik dapat dibuat dengan menggunakan MS Excel (Gambar 3.15).



Gambar 3.15 Beberapa contoh grafik yang bisa dibuat di aplikasi MS Excel.

3.6.1 Jenis-Jenis Grafik

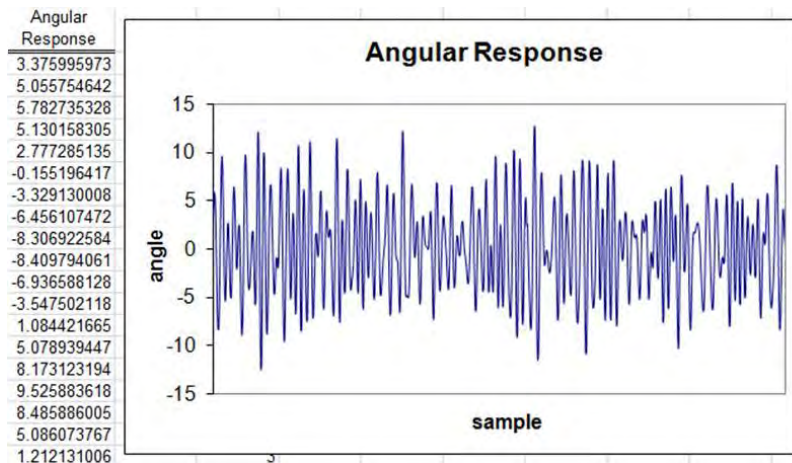
Untuk membuat masing-masing jenis grafik tentunya kita harus paham apa makna dari masing-masing grafik tersebut. Berikut adalah penjelasan singkat tentang beberapa dari grafik-grafik tersebut.

Bar chart

Bar chart dibentuk dari persegi panjang dengan panjang sesuai dengan besaran data. Umumnya bar chart digunakan untuk menampilkan data-data yang independen dan dapat dibandingkan. Contohnya data penjualan produk di setiap lokasi penjualan, data pengguna narkoba di suatu negara dibagi berdasarkan gender dan lain-lain.

Line chart

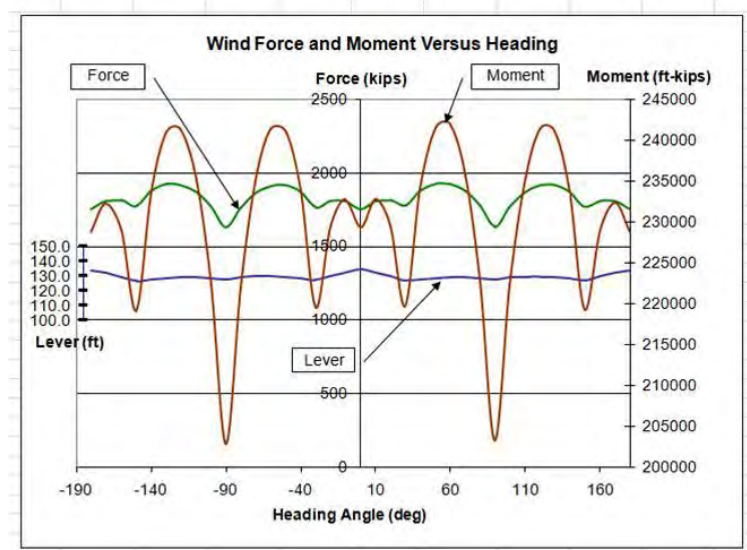
Menampilkan titik data yang dihubungkan dengan garis. Line chart digunakan untuk data tunggal (ordinat) dengan absis berupa satuan data. Grafik ini sesuai untuk menampilkan banyak data dalam sampling dalam satuan tertentu. Contoh dapat dilihat pada Gambar 3.16.



Gambar 3.16 Contoh line chart pada data pengukuran respon angular.

X-Y (scatter)

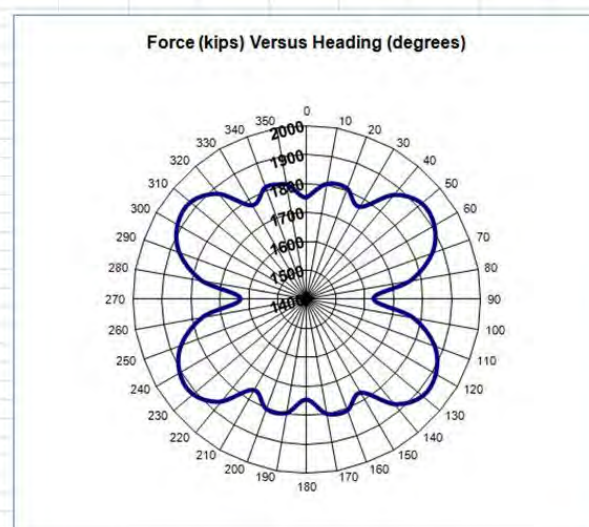
Jenis grafik ini banyak digunakan untuk menampilkan data teknik dengan data X dan Y yang berbeda. Pada umumnya data-data percobaan/pengujian mempunyai korelasi data antara nilai X dan Y. Contoh pada Gambar 3.17.



Gambar 3.17 Contoh grafik X-Y scatter yang menampilkan data gayadan momen angin terhadap arahnya.

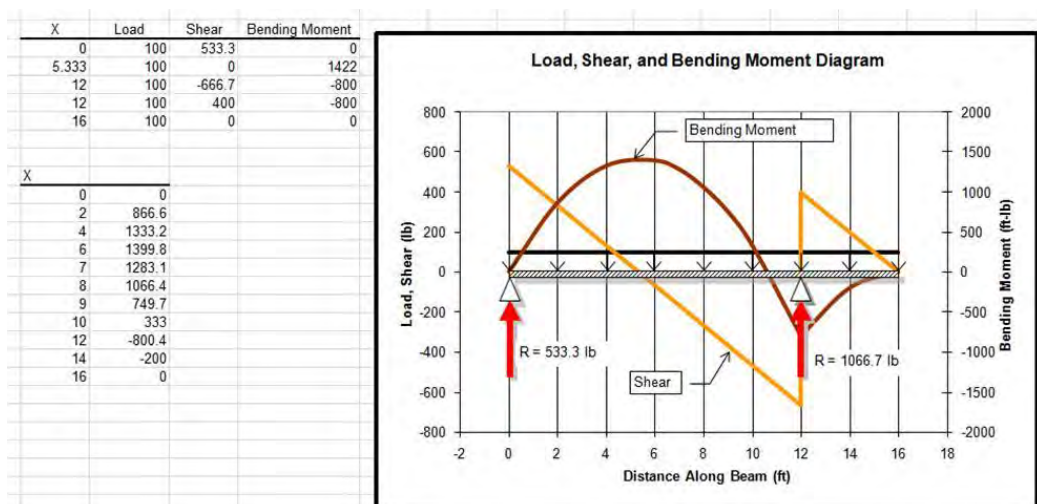
Polar chart

Polar chart menunjukkan grafik garis seperti pada X-Y scatter tetapi dengan absis merupakan posisi angular. Contoh



Gambar 3.18 Contoh polart chart yang menampilkan data gay akibat angin dan arahnya (angular).

Di dalam penggunaan teknik, MS Excel juga memungkinkan untuk menggambarkan fenomena teknik dengan grafik. Sebagai contoh dalam permasalahan Mekanika Teknik terdapat perhitungan gaya-gaya dalam dengan menghitung gaya geser dan momen bending di setiap titik dalam struktur. Contoh dapat dilihat pada Gambar 3.19.



Gambar 3.19 Contoh grafik gaya-gaya dalam sebuah struktur yang dibuat dengan aplikasi MS Excel

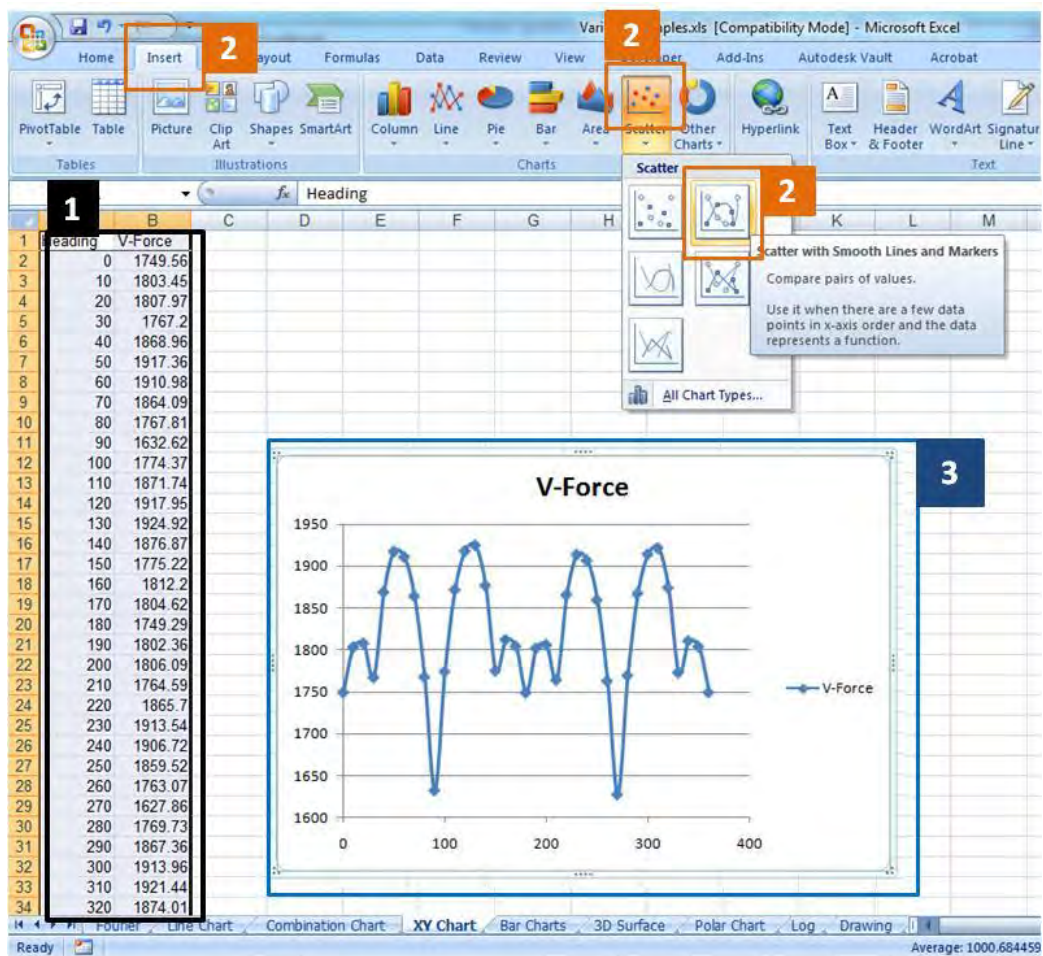
3.6.2 Cara Membuat Grafik Sederhana

Untuk membuat grafik di aplikasi MS Excel 2007 sangatlah mudah. Berikut langkah yang bisa dilakukan untuk membuat grafik sederhana. Dalam contoh ini dibuat grafik X-Y scatter. Perhatikan Gambar 3.20.

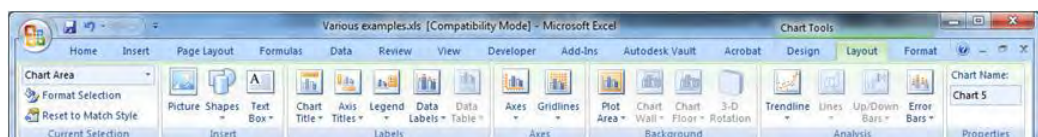
1. Persiapkan data yang akan dibuat grafik, data bisa diimport atau dimasukkan. Pilih data tersebut beserta heading-nya. Perlu dicatat

bahwa data pada kolom pertama adalah data X dan data kolom berikutnya adalah data Y (berlaku untuk banyak data Y)

2. Pilih menu tab **Insert** pilih **Scatter** dan pilihan scatter yang sesuai.
3. Kemudian klik chart yang sudah terbuat dan kemudian pilih menu tab **layout** () pada **chart tools** untuk memberikan informasi lain yang diperlukan seperti:
 - a. Chart Title
 - b. Axis Title
 - c. Gridline, dan lain-lain.



Gambar 3.20 Cara membuat grafik lebih mudah di MS Excel 2007.



Gambar 3.21 Group toolbar Layout pada kelompok chart tools.



BAB 4 PROGRAM APLIKASI PRESENTASI

Tujuan Pembelajaran Umum

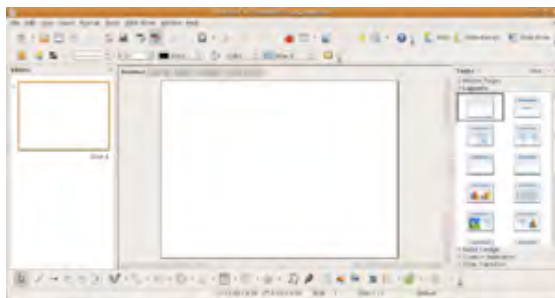
1. Dapat membuat presentasi untuk menampilkan hasil kerja atau pokok bahasan tertentu dengan baik.

Tujuan Pembelajaran Khusus

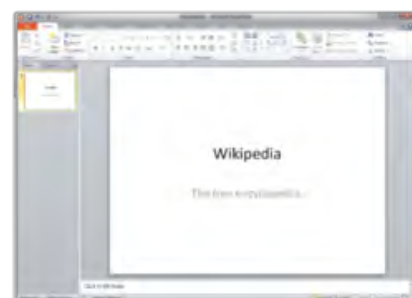
1. Mahasiswa dapat membuat presentasi dengan format susunan presentasi yang terarah.
2. Mahasiswa dapat menggabungkan informasi dalam presentasi dalam bentuk statis maupun dinamis dengan animasi.

4.1 Konsep Dasar Tentang Presentasi

Presentasi seperti diketahui adalah proses praktis untuk menampilkan dan menjelaskan isi sebuah topik kepada *audience*. Tentunya untuk menampilkan sesuatu diperlukan sesuatu yang sederhana tetapi menarik. Beberapa perangkat lunak dikembangkan untuk keperluan ini, baik yang bersifat opensource maupun yang komersial. Aplikasi presentasi opensource tersedia cukup banyak diantaranya adalah **OpenOffice.Org Impress**, **Kpresenter**, **Beamer** dan lain-lain. Sedangkan yang komersial diantaranya adalah **Apple Keynote**, **Corel Presentations**, **IBM Lotus Freelance Graphics**, **Microsoft PowerPoint** dan lain-lain.



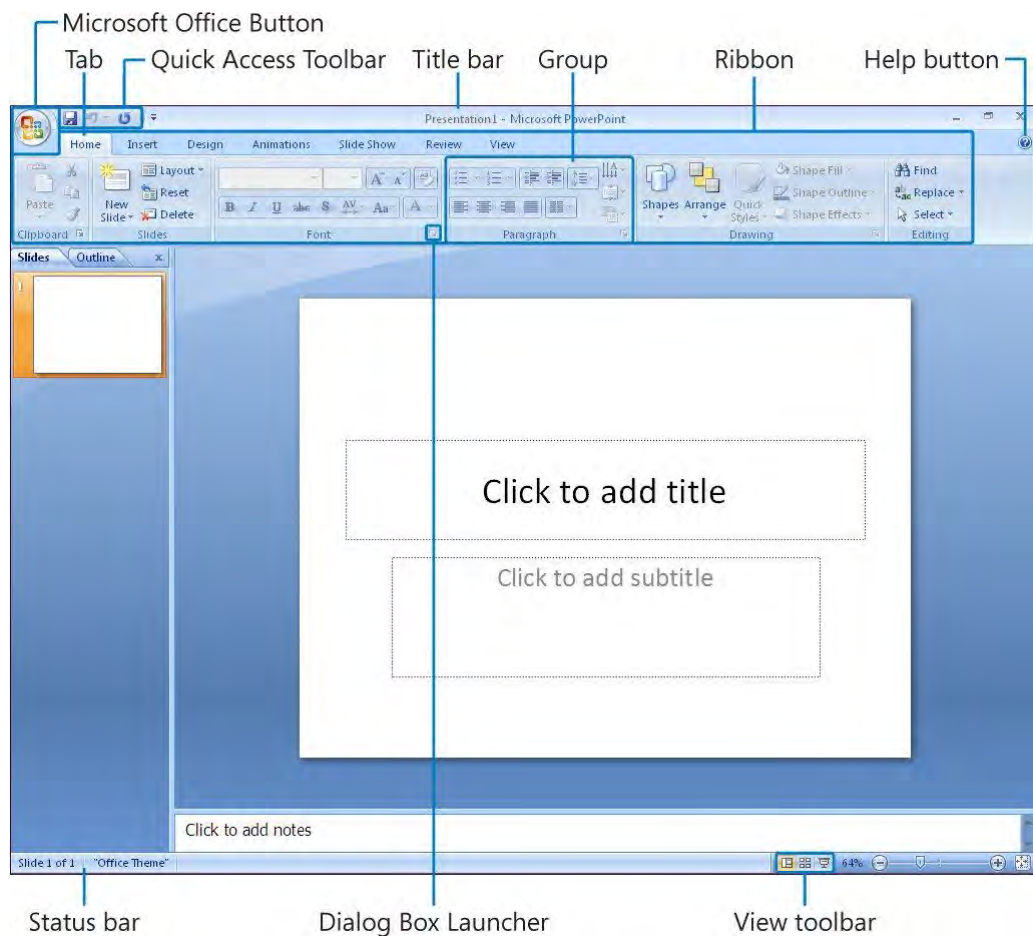
OpenOffice.Org Impress



Microsoft Powerpoint 2007

Gambar 4.1 Contoh antarmuka aplikasi presentasi

Dalam buku ajar ini akan dijelaskan tentang penggunaan aplikasi presentasi Microsoft PowerPoint 2007. Pembahasan detail tentang cara pembuatan objek di PowerPoint tidak akan dibahas secara mendalam. Pembahasan akan lebih kepada bagaimana membuat presentasi yang efektif dan menarik untuk ditampilkan.



Gambar 4.2 Antarmuka aplikasi Microsoft PowerPoint 2007.

Istilah-istilah dalam antarmuka aplikasi MS PowerPoint tidak ada yang spesifik jika dibandingkan dengan MS Word. Untuk mengingat kembali silahkan dilihat kembali pembahasan antarmuka MS Word 2007 di sub bab 2.2.

4.2 Tipografi, Layout dan Desain

Hal terpenting dalam menampilkan sebuah presentasi adalah memilih bentuk tulisan, layout dan desain tampilan. Beberapa hal berikut perlu diperhatikan dalam pemilihan atau pembuatan layout presentasi. Beberapa hal berikut perlu diperhatikan.

Themes

Themes dapat memuat pemilihan warna (color pallette), pemilihan font, efek-efek dan lain-lain. Untuk memulai pembuatan presentasi lebih cepat dapat dimulai melalui pemilihan theme yang sesuai.

KEJELASAN adalah kata kunci

Mengorganisasi informasi sehingga membuat pesan yang disampaikan menjadi **JELAS** adalah kunci. Sehebat apapun karismatik yang anda punyai, anda bisa meninggalkan kebingungan kepada audiens jika presentasi yang anda sampaikan tidak terorganisasi dengan baik. Prinsip untuk mengorganisasi kejelasan sebuah presentasi dapat dilakukan dengan prinsip-prinsip kejelasan visual.

Terlihat Keren tapi tak terbaca

Kita temui di dalam pemilihan font banyak sekali jenis yang keren-keren. Atau kita melihat presentasi dengan latar belakang yang sangat menarik, tetapi justru menghalangi kejelasan tulisan yang ditampilkan. Perlu diperhatikan **Keterbacaan** adalah kunci. Kalau audiens tidak bisa membaca informasi di presentasi, maka mereka tidak akan memperhatikan apa yang telah dipresentasikan.

Berikut adalah hal-hal yang harus diperhatikan dalam playout presentasi. Menggunakan dengan benar akan menghasilkan hasil yang diinginkan.

Ilustrasi

Yang harus dilakukan:



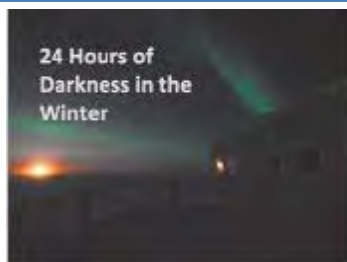
Jaga tetap sederhana

Slide yang sederhana akan membantu anda untuk menuntun audiens memahami hasil yang anda inginkan. Pikirkan bahwa slide adalah tanda-tanda di sepanjang jalan tol, audiens anda berkendara dengan kecepatan 100 km/jam. Apakah anda akan meletakkan beberapa gambar yang membutuhkan perhatian lebih?



Tetap konsisten

Konsistensi akan membuat audiens anda tetap fokus ke isi slide. Warna, ukuran dan gaya font harus tetap konsisten di semua presentasi anda.



Buat presentasi anda untuk audiens anda.

Jika anda merencanakan dan membuat presentasi bersifat mandiri tanpa presenter, maka teks dan gambar yang anda sajikan harus bisa menjadi *selfguide tour*. Presentasi bukan disiapkan untuk menjadi handout. Jika menghendaki layout, gunakan program untuk membuat layout.



Jaga Bullet tetap ringkas dan jelas

Banyak orang saat presentasi keluar dari topik dan membingungkan. Jika hal ini terjadi, sebaiknya audiens dapat sekilas melihat bullet presentasi anda dan mengerti ke mana arah presentasi anda.

Ilustrasi

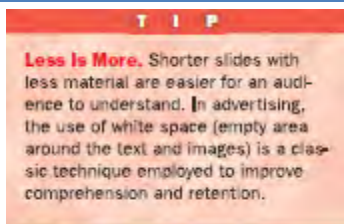


Yang harus dilakukan:

Gunakan gambar yang menarik dengan Kualitas yang tinggi

Gambar yang jelek, tidak fokus, tak berwarna dan resolusi rendah sangatlah membosankan.

Ilustrasi



Yang tidak boleh dilakukan:

Menggunakan terlalu banyak element dalam slide

Untai ide-ide yang kompleks menjadi bentuk yang paling sederhana. Jika sangat dibutuhkan sebuah bahasan yang sangat besar, sebaiknya dibuat dalam beberapa slide. Buat setiap slide jelas dan mudah untuk diikuti



Membuat presentasi berbasis bullet

Presentasi dengan semua halaman ber-bullet akan sangat membosankan. Buat lebih sedikit bullet, selingi dengan gambar besar yang menarik, atau gugah audiens dengan lembar kosong dengan pernyataan ringkas dengan tulisan yang sangat besar.



Meletakkan gambar secara acak

Sangat banyak cara untuk menyusun presentasi. Menyusun gambar secara rapi akan membuat presentasi lebih fokus.



Menggunakan slide yang dipresentasikan sebagai handout

Jika anda menyiapkan presentasi juga sebagai handout, maka anda akan berusaha memberikan informasi sebanyak mungkin. Dalam presentasi mungkin saja anda hanya meletakkan sebuah video yang bercerita banyak, dalam hal ini anda tidak perlu terikat oleh handout.

Ilustrasi



Yang tidak boleh dilakukan:

Berlebih dengan WordArt

Tetap jaga agar presentasi tetap jelas dan mudah dibaca. Banyak cara untuk membuat presentasi terlihat keren, tidak hanya dari WordArt.

Pilih dan install font yang anda inginkan.

Dalam pembuatan presentasi kita mempunyai kesempatan untuk memilih font secara acak. Kita bisa ambil salah satu, mencobanya, dan mungkin mencoba yang lainnya. Jika anda mengetahui dengan persis karakteristik dari font, maka mudah buat anda untuk memilih. Memilih font yang “sesuai” merupakan kombinasi dari beberapa faktor. Anda butuh mencari tahu tentang keterbacaan dibandingkan dengan artistik dan gaya.

Serif vs San Serif

Font mempunyai 2 perbedaan utama, yaitu: Serif dan San Serif. Font Serif (seperti **Times**, **Garamond** dan **New York**) secara umum lebih mudah dibaca. Font tersebut mempunyai alur tipis dan tebal. Sebagai alternatif adalah jenis San Serif (seperti **Gill Sans**, **Helvetica** dan **Arial**). Font tersebut mempunyai gaya yang lebih jelas. Berikut perbandingan antara 2 jenis font tersebut.

Serif

- Lebih mudah dibaca saat dicetak
- Lebih banyak pilihan font yang tersedia
- Dimodelkan setelah teks tulisan tangan dan jenis cetak pertama
- Difavoritkan oleh orang tadisional

San Serif

- Kemungkinan menimbulkan masalah ketebalan karakter
- Sudah dioptimasi untuk digunakan di layar
- Dapat dikompakkan penggunaan banyak teks dalam tempat yang lebih kecil
- Keterbacaan lebih baik untuk ukuran kecil

Beberapa font modern terlihat sangat baik di layar diantaranya adalah: **Gill Sans**, **Georgia**, **Myriad**, **Impact**, **Helvetica Neue**, **Verdana**, **Trebuchet** dan **Futura**. Banyak sekali tersedia font secara online, baik yang bersifat Cuma-Cuma maupun yang komersial. Berikut beberapa alamat wensite yang menawarkan Font:

- www.chank.com (cuma-cuma dan komersial)
- www.fonthead.com (cuma-cuma dan komersial)
- www.myfonts.com (komersial)
- www.adobe.com/type/index.html (komersial)
- www.girlswhowearglasses.com (cuma-cuma)

Menggunakan PowerPoint Themes

Dengan menggunakan PowerPoint Themes dapat mempermudah dan mempercepat pembuatan presentasi yang luar biasa. Anda bisa memodifikasi themes yang sudah ada untuk membedakan dengan yang lain.

Apa yang membuat theme itu bagus?

Dua hal yang harus diperhatikan saat memilih themes adalah **kesederhanaan** dan **keserbagunaan**. Semakin sederhana theme yang digunakan, akan semakin mengurangi kebingungan penyampaian materi dan pesan. Pemilihan warna juga harus yang ada harmonisasi satu dengan yang lainnya. Theme yang disediakan oleh PowerPoint pada dasarnya sudah cukup baik tampilannya, tapi terkadang tidak cocok pengaturan warna, font dan lainnya. Tapi jangan kuatir, karena kita bisa dengan mudah mengubahnya.

Dari mana kita bisa mulai?

Beberapa pertanyaan bisa menjadi pertimbangan dari mana kita bisa mulai. Buat beberapa list pertanyaan yang berkaitan dengan presentasi yang akan anda buat. Sesuaikan dengan branding yang hendak anda sampaikan, dan juga dengan audiens yang hendak anda hadapi.

- Apakah anda mempunyai warna yang harus sesuai dengan instansi/perusahaan anda?
- Apakah anda mempunyai logo dari team/instansi/perusahaan atau maskot sekolah yang harus dipenuhi?
- Kepada siapa ditujukan presentasi anda? Kepada para insinyur, olahragawan, artis dan lain-lain.
- Apakah audiens familiar dengan website anda?
- Dan seterusnya.



Gambar 4.3 Short cut untuk mencari theses secara online.

(1) klik bagian ini untuk mengeluarkan semua theme. (2) untuk mencari theme secara online.

Themes lain anda bisa cari secara online.

Jika anda tidak menemukan themes yang sesuai dengan beberapa pertanyaan di atas, anda bisa mencari pilihan lain yang di sediakan secara online. Microsoft menyediakan banyak themes secara online, baik yang percuma maupun yang komersial. Gambar 4.3 menunjukkan langkah untuk mendapatkan themes secara online.

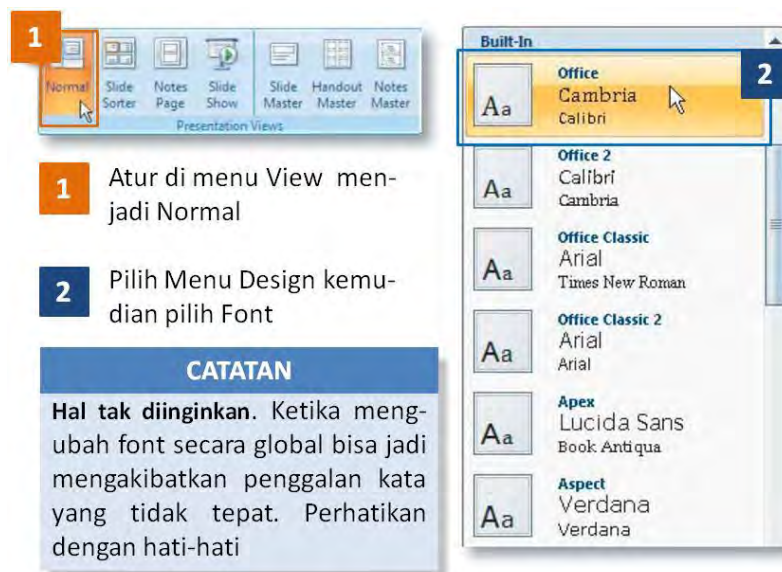
4.2.1 Mengubah Font secara Global dan Lokal

Pengubahan font di dalam presentasi dapat dilakukan secara global (menyeluruh) atau secara lokal (setempat/ hanya yang dipilih).

Pengubahan Global

Pengubahan global dapat dilakukan dengan 2 (cara):

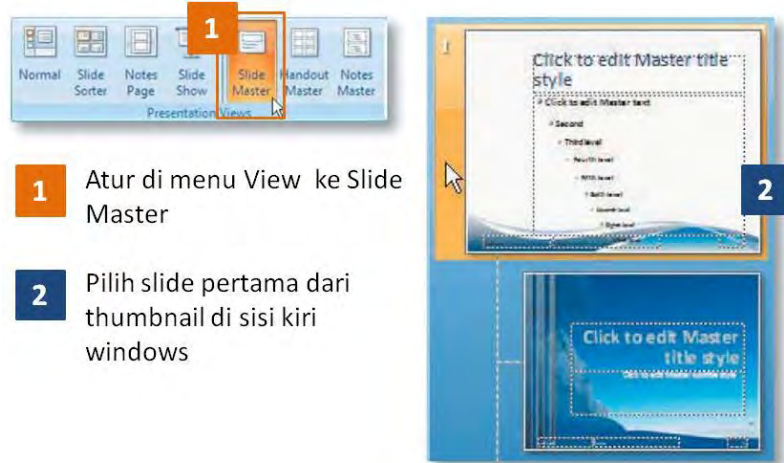
1. Dengan cara cepat melalui menu design → Font dengan sebelumnya mengubah pandangan melalui menu View → Normal. Langkahnya seperti pada Gambar 4.4.



Gambar 4.4 Pengubahan font secara global dengan cara cepat.

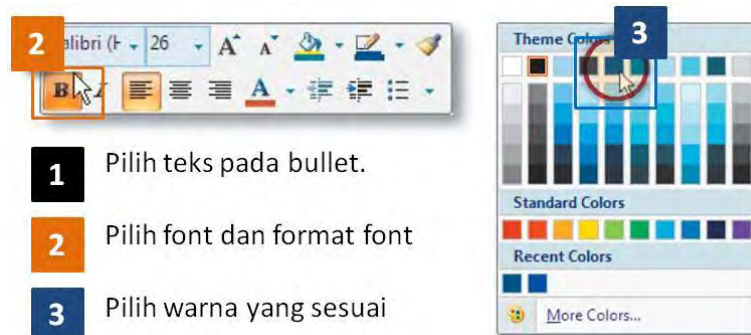
2. Dengan cara memodifikasi setting font dan layoutnya di Slide master. Untuk melakukan ini, akses melalui menu tab View → Slide Master.

Pengaturan font, format dan warnanya bisa dilakukan pada setiap level bullet yang digunakan. Langkah melakukannya dapat dilihat pada Gambar 4.5 dan Gambar 4.6.



- 1 Atur di menu View ke Slide Master
- 2 Pilih slide pertama dari thumbnail di sisi kiri windows

Gambar 4.5 Memodifikasi format tulisan (font) melalui Slide Master.



- 1 Pilih teks pada bullet.
- 2 Pilih font dan format font
- 3 Pilih warna yang sesuai

Gambar 4.6 Pengaturan format font dan warna melalui silde master.

4.2.2 Mengorganisasi Informasi agar JELAS

Tujuan utama dari setiap presentasi adalah untuk menyediakan informasi kepada audiens dengan sesuatu yang mudah untuk mengerti. Tujuan yang lain untuk memotivasi audiens untuk melakukan beberapa aksi ringan. Untuk mencapai tujuan tersebut, anda harus selalu kembali ke belakang dan bertanya pada diri anda sendiri, Apakah saya sudah mempresentasikan informasi yang paling jelas dan paling sederhana?

4.3 Pengaturan



BAB 5

DASAR PEMROGRAMAN (PROBLEM SOLVING)

Tujuan Pembelajaran Umum

1. Memahami dasar-dasar pemrograman komputer dengan konsep pemecahan masalah secara umum.

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat melakukan identifikasi masalah dan dapat membuat langkah-langkah penyelesaiannya.
2. Mahasiswa dapat membuat algoritma dalam bentuk flowchart terhadap permasalahan.

5.1 Siklus Pengembangan Program

Berbicara tentang pengembangan program komputer maka tidak bisa terlepas dengan **Hardware** yang berfungsi sebagai mesin di dalam sistem komputer (seperti monitor, *keyboard* dan CPU) dan **Software** yang merupakan kumpulan dari perintah-perintah yang disebut program (atau proyek) untuk mengatur hardware tersebut. Program ditulis untuk menyelesaikan suatu masalah atau mengerjakan tugas-tugas dengan komputer. *Programmer* (pembuat program) menerjemahkan solusi ke dalam bahasa yang dapat dimengerti oleh komputer. Perlu diingat bahwa komputer hanya menjalankan perintah-perintah yang ditulis, oleh karena itu harus berhati-hati pada saat membuat instruksi dalam program.

5.1.1 Gambaran Umum

Untuk memahami tentang penulisan instruksi-instruksi (alur program), maka tahapan berikut adalah dasarnya.

Pertama; menentukan **output** apa yang diinginkan.

Kedua; identifikasi **input** (data) yang diperlukan untuk menghasilkan output

Terakhir; menentukan **proses** apa yang harus dilakukan terhadap input untuk menghasilkan output yang diinginkan. Proses merupakan langkah untuk menentukan formula atau cara apa yang dilakukan untuk menghasilkan output.

Sebagai gambaran sederhana, pendekatan penyelesaian masalah ini dapat dianalogikan dengan penyelesaian soal cerita pada kelas matematika di sekolah dasar. Contoh yang sederhana seperti soal berikut:

Berapa kecepatan yang dibutuhkan sebuah mobil untuk menempuh jarak 50 km dalam waktu 2 jam?

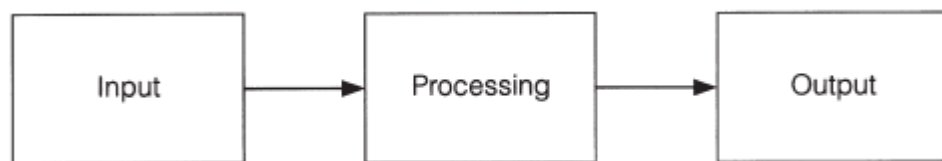
Dari soal di atas diinginkan:

Output: kecepatan (v)

Input: jarak (s) dan waktu (t) tempuh (yang diketahui)

Proses: dari pengetahuan dasar diketahui bahwa formula untuk menghasilkan kecepatan adalah jarak dibagi dengan waktu atau dapat dituliskan $v = \frac{s}{t}$

Penyelesaian masalah di atas berdasarkan urutannya dapat digambarkan seperti ditunjukkan pada Gambar 5.1. Dari **input** yang dibutuhkan (diketahui) kemudian dengan **proses** yang sudah didefinisikan akan menghasilkan **output** yang diinginkan.



Gambar 5.1 Diagram Alir Pemecahan Masalah (*Problem Solving*).

Berdasarkan diagram proses penyelesaian masalah di atas, maka masalah soal cerita di atas dapat diselesaikan sebagaimana ditunjukkan pada Tabel 5.1 dalam bentuk tabulasi seperti urutan pada Gambar 5.1.

Tabel 5.1 Tabulasi urutan proses penyelesaian masalah dalam contoh soal cerita matematika sederhana.

INPUT	PROSES	OUTPUT
Diketahui: $S = 50 \text{ km}$ $T = 2 \text{ jam}$	Formulasi penyelesaian: $v = \frac{s}{t} = \frac{50 \text{ km}}{2 \text{ jam}}$	Hasil: $v = 25 \text{ km/jam}$

5.1.2 Perencanaan Program

Contoh permasalahan pada sub bab sebelumnya merupakan contoh yang sangat sederhana hanya untuk membangun pola pikir bagaimana seharusnya menyelesaikan permasalahan. Dalam kasus pembuatan program dengan menggunakan komputer permasalahan yang dihadapi bisa jauh lebih rumit dari contoh di atas, sehingga diperlukan perencanaan penyusunan program. Tahap berurutan yang digunakan dalam perencanaan disebut sebagai *program development cycle* (siklus pengembangan program), yaitu:

1. **Analyze:** Mendefinisikan masalah

Definisikan dengan jelas output dan input yang diberikan dan hubungan antara input dan input yang diinginkan.

2. **Design:** Merencanakan langkah-langkah solusi terhadap masalah

Cari urutan logik dari langkah-langkah penyelesaian masalah yang disebut dengan **Algoritma**. Istilah Algoritma diambil dari seorang ilmuwan Islam yang bernama Al-Khawarizmy. Setiap detail dari langkah-langkah penyelesaian masalah harus ituliskan dalam algoritma. Metoda yang populer untuk algoritma diantaranya adalah **Flowchart**, **Pseudocode** dan **top-down chart**.

3. **Choose the interface:** Memilih objek (*text box, command button dll*)

Dalam konteks pemrograman grafik seperti Visual Basic 6.0 atau Delphi atau program sejenis lainnya, perlu disiapkan objek yang digunakan untuk menyediakan input (*text box*), menampilkan output (*text box, label* atau yang lain) dan tombol (*command button*) untuk akses user menjalankan program.

4. **Code:** Menerjemahkan algoritma ke dalam bahasa pemrograman

Coding adalah istilah teknik yang menggambarkan proses penulisan dalam bahasa program. Penulisan bahasa dalam bentuk kode-kode sesuai bahasa pemrograman yang digunakan, contoh misalnya Visual Basic 6.0. Kode-kode yang ditulis merupakan penerjemahan dari tahap kedua, dari algoritma.

5. **Test and Debug:** Menjalankan, mencari kesalahan dan koreksi

Testing adalah proses pencarian kesalahan pada program saat dijalankan sedangkan **debugging** adalah proses koreksi terhadap kesalahan yang ditemukan. Jika urutan langkah pada algoritma sudah benar, maka pada umumnya kesalahan terjadi karena salah penulisan bahasa program.

6. **Complete the documentation**

Dokumentasi dilakukan untuk mengumpulkan semua informasi akhir dari semua proses agar program yang telah dibuat mempunyai dokumen yang dapat digunakan untuk pengembangan lebih lanjut.

5.2 Flowchart sebagai Algoritma Pemrograman





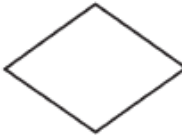




Algoritma seperti yang telah dijelaskan pada siklus pengembangan program pada sub bab sebelumnya merupakan langkah awal yang terpenting karena penyusunan bahasa pemrograman akan didasarkan pada algoritma ini. Salah satu jenis algoritma yang paling mudah dipahami dan paling umum digunakan dalam kebanyakan penyelesaian permasalahan adalah **Flowchart**. Flowchart ditampilkan dalam bentuk grafik (lambang bentuk) yang mempunyai makna tertentu. Kelemahan utama dari flowchart adalah jika program yang dibuat sangat besar dan rumit. Daftar bentuk lambang-lambang dalam flowchart dapat dilihat pada Tabel 5.2.

Algoritma secara umum dapat dibagi menjadi 3 jenis yaitu:

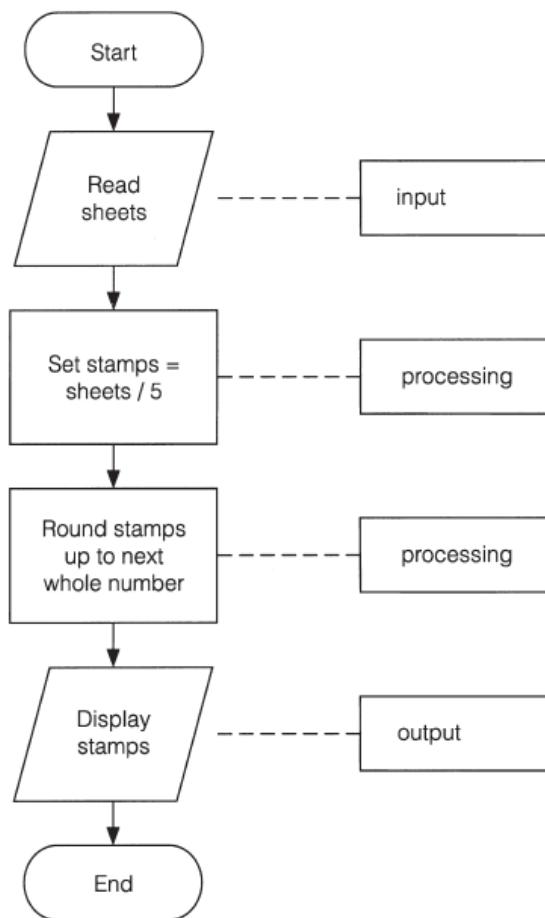
1. Program berurutan (*sequential*)

2. Program Pilihan (*Decision*)
3. Program Pengulangan (*Repetition*)

Tabel 5.2 Daftar lambang gambar yang digunakan dalam flowchart.

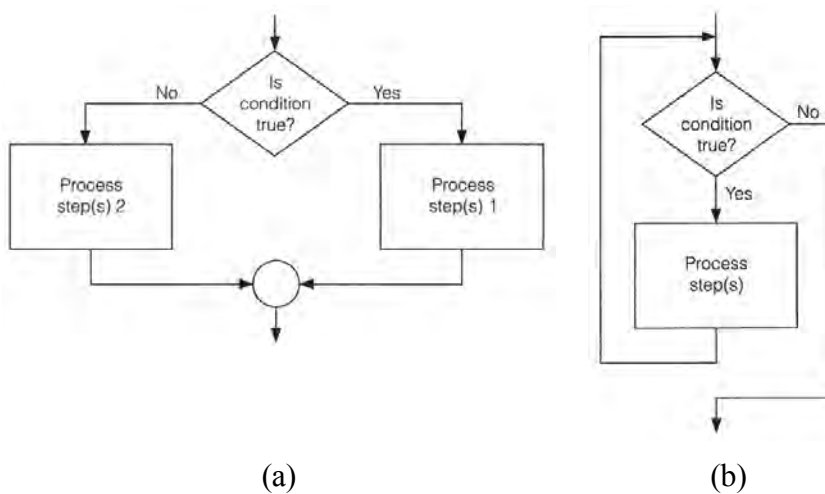
Lambang	Nama	Arti
	Garis Alur	Digunakan untuk menghubungkan antar lambang dan menunjukkan alur logika
	Terminal	Digunakan sebagai awal/akhir (<i>start/end</i>)
	Input/output	Digunakan sebagai input/output, atau dengan kata lain <i>reading</i> dan <i>printing</i> .
	Proses	Proses yang menunjukkan manipulasi data atau formulasi aritmatika.
	Pilihan	Digunakan sebagai pernyataan logika pilihan atau operasi perbandingan. Pernyataan berupa pertanyaan logika dengan jawaban ya/tidak (<i>yes/no</i>), sehingga lambang ini akan mempunyai 1 masukan dan 2 keluaran jika jawaban “ya” dan jika jawaban “tidak”.
	Konektor	Untuk menghubungkan 2 garis alur yang berbeda.
	Konektor beda halaman	Digunakan untuk menunjukkan bahwa flowchart berlanjut ke halaman berikutnya.
	Proses yang sudah terdefinisi	Untuk menunjukkan kelompok pernyataan yang mewakili satu tugas.
	Anotasi	Untuk menyediakan informasi tambahan.

Contoh penggunaan flowchart dapat dilihat pada Gambar 5.2. Flowchart dimulai dengan start dan diakhiri dengan end. Pada umumnya awal setelah start dimulai dengan input dan sebelum end diakhiri dengan output.



Gambar 5.2 Contoh Flowchart untuk permasalahan stempel pos.

Contoh flowchart di atas adalah untuk program yang berurutan, untuk program pilihan dan pengulangan dapat dilihat pada Gambar 5.3.



Gambar 5.3 Contoh flowchart untuk (a) Program pilihan dan (b) Program pengulangan.



BAB 6

DASAR PEMROGRAMAN DENGAN VISUAL BASIC

Tujuan Pembelajaran Umum

1. Memahami cara kerja bahasa pemrograman secara umum dan secara khusus adalah Visual Basic 6.0

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat membuat antarmuka program sederhana untuk memahami cara kerja pemrograman visual dengan Visual Basic 6.0
2. Mahasiswa dapat membedakan fungsi setiap objek dalam pemrograman visual dan memahami dengan jelas pengertian object, property, method dan Event dan pemrograman visual.

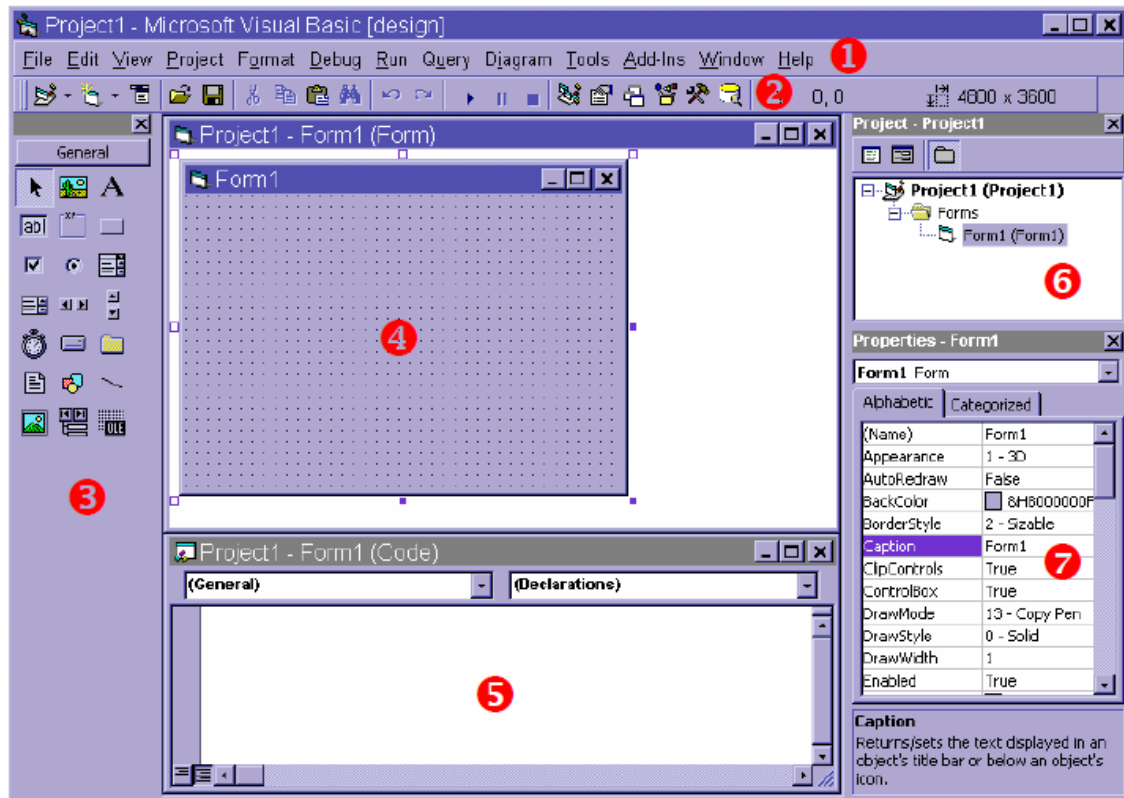
Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic yang dikembangkan oleh Microsoft sejak tahun 1991 merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP).

6.1 Mengetahui Antarmuka Visual Basic 6.0

Untuk memulai Program Visual Basic 6.0 melalui tombol **Start > Programs > Microsoft Visual Studio 6.0 > Microsoft Visual Basic 6.0**. Tampilan awal akan dimulai dengan pilihan jenis program yang akan dibuat seperti ditunjukkan pada Gambar 6.1.




Gambar 6.1 Tampilan awal Program Visual Basic 6.0



Gambar 6.2 Tampilan Antarmuka Program Visual Basic 6.0.


Keterangan :

1. **Menubar**
2. **Toolbar**
3. **Toolbox**

Bila Toolbox tidak muncul klik tombol **Toolbox** () pada bagian Toolbar atau klik menu


View > Toolbox.

4. **Jendela Form**

Bila Jendela Form tidak muncul klik tombol **View Object** () pada bagian Project Explorer atau

klik menu **View > Object.**

5. **Jendela Code**


Bila Jendela Code tidak muncul klik tombol **View Code** () di pada bagian Project Explorer atau

klik menu **View > Code.**

6. **Project Explorer**

Bila tidak muncul klik tombol **Project Explorer** () pada bagian Toolbar atau klik menu **View > Project Explorer.**

7. Jendela Properties

Bila tidak muncul klik tombol Properties Window () pada bagian Toolbar

atau klik menu View > Properties Window

6.2 Memahami Istilah Object, Property, Method dan Event

Dalam pemrograman berbasis obyek (OOP), anda perlu memahami istilah object, property, method dan event sebagai berikut :

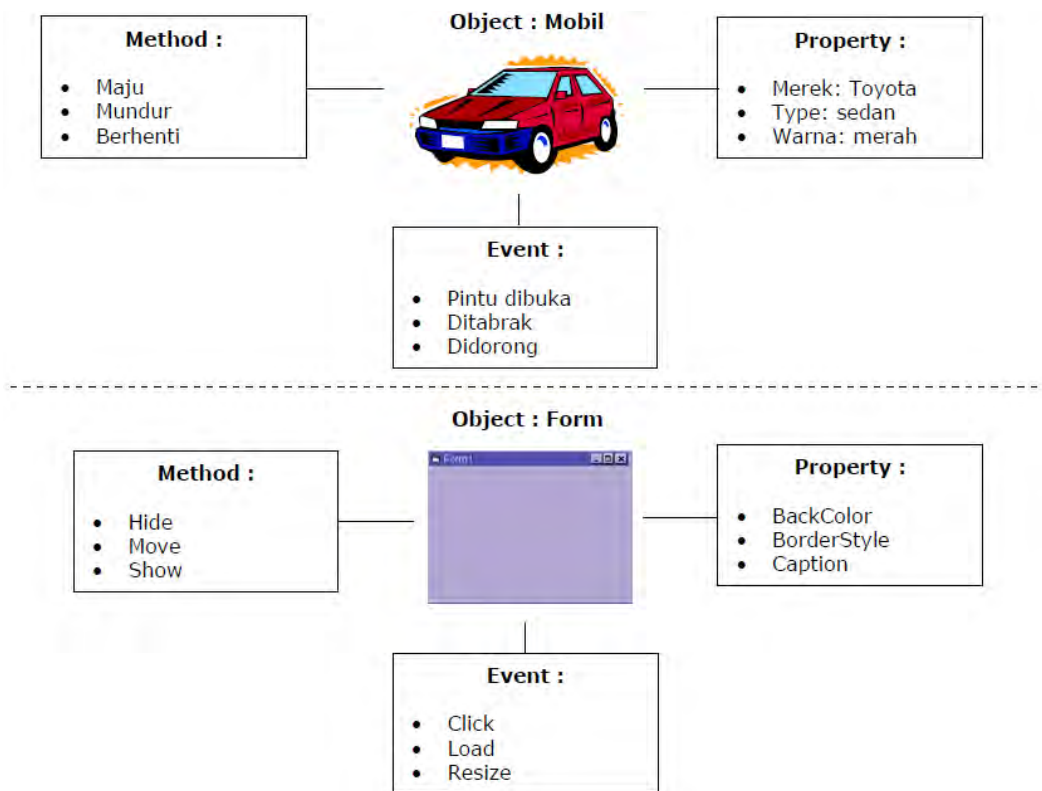
Object : komponen di dalam sebuah program

Property : karakteristik yang dimiliki object

Method : aksi yang dapat dilakukan oleh object

Event : kejadian yang dapat dialami oleh object

Sebagai ilustrasi dapat menganggap sebuah mobil sebagai obyek yang memiliki *property*, *method* dan *event*. Perhatikan Gambar 6.3 berikut.




Gambar 6.3 Ilustrasi pengertian Object, Property, Method dan Event.


Implementasinya dalam sebuah aplikasi misalnya anda membuat form, maka form tersebut memiliki property, method, dan event. Sebagaimana pemrograman visual lain seperti Delphi dan Java, VB juga bersifat event driven programming.


Artinya anda dapat menyisipkan kode program pada event yang dimiliki suatu obyek.


6.3 Objek-Objek di Visual Basic 6.0

Icon **Toolbox** merupakan objek control yang dapat digunakan di dalam form. Objek Control yang sering digunakan akan dibahas pada sub bab berikut yaitu:

Text boxes:  text box utamanya digunakan untuk mendapatkan informasi, atau **input**, dari pengguna.

Labels:  label biasanya diletakkan di sisi kiri text box yang digunakan untuk memberikan informasi kepada pengguna kegunaan text box tersebut atau memberikan keterangan tambahan. Terkadang label juga dapat digunakan untuk menampilkan output.

Command buttons:  user meng-klik command button untuk memerintahkan aksi tertentu.

Picture boxes:  picture box dapat digunakan untuk menampilkan output berupa text maupun grafik.

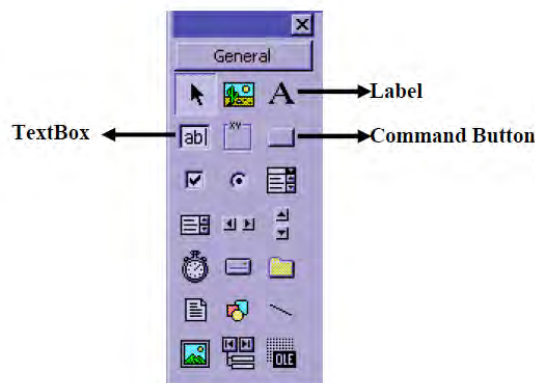
Sub-sub bab berikut akan menjelaskan objek-objek di atas dengan memberikan ilustrasi dalam program sederhana.

6.4 Menggunakan Event dan Property

Untuk mempermudah pemahaman atas pengertian objek, property, Method dan event, bahasan berikut akan menuntun secara sederhana bagaimana cara kerjanya dalam program di Visual Basic 6.0.

6.4.1 Membuat *User Interface* (UI)

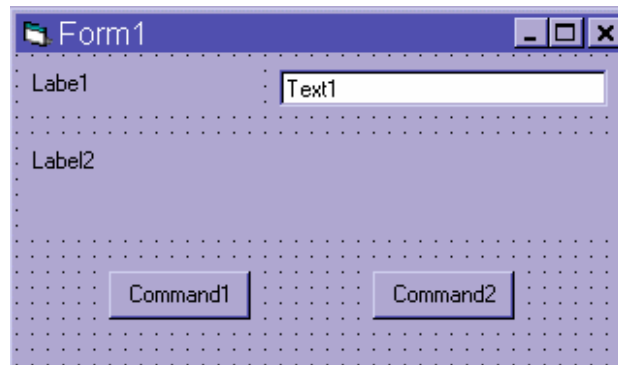
Kenali jenis-jenis komponen (*control*) yang ada di dalam bagian ToolBox. Jenis-jenis komponen tersebut dapat dilihat pada .



Gambar 6.4 Kelompok Toolbar ToolBox yang berisi komponen-komponen Control dasar.

Contoh Program: Antarmuka dan Objek

Pada Jendela Form buatlah UI seperti .



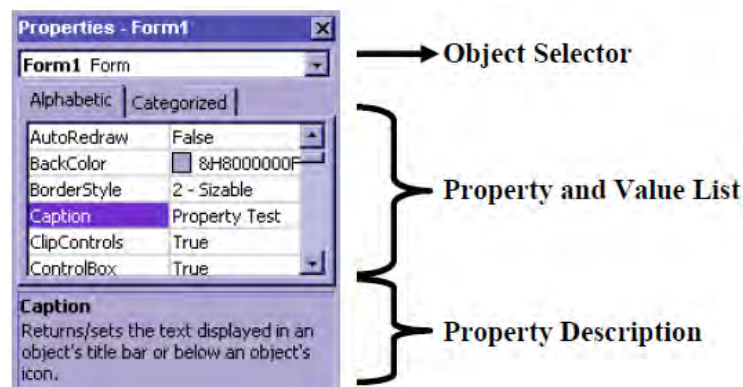
Gambar 6.5 Contoh Form dengan beberapa objek di dalamnya.

Cara membuatnya :

Klik komponen Label di dalam ToolBox kemudian “gambar” object Label1 di dalam Form dengan cara drag-drop. Lakukan langkah yang sama untuk object Label2. Object Text1 di-“gambar” menggunakan komponen TextBox, sedangkan object Command1 dan Command2 menggunakan komponen Command Button.

6.4.2 Mengatur Property Object

Setelah membuat antarmuka program yang diwakili oleh objek form dan beberapa objek lain, maka perlu dilakukan perubahan property dari objek. Kenali bagian-bagian di dalam Jendela Properties seperti ditunjukkan pada .



Gambar 6.6 Bagian-bagian di dalam jendela property

Melalui Jendela Properties atur property setiap object sebagaimana ditunjukkan dalam Tabel 6.1. Perhatikan setiap perubahan yang terjadi akibat perubahan property dari objek. Perubahan objek pada Gambar 6.7 dapat dibandingkan dengan kondisi awal seperti pada Gambar 6.5.

Tabel 6.1 Tabulasi pengubahan nilai dari property objek

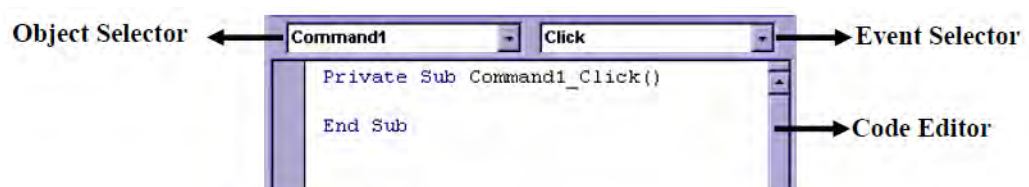
Object	Properties	Value
Form1	Caption StartPosition	Property Test 2 – CenterScreen
Label1	Caption	Tuliskan nama Anda :
Text1	-	-
Label2	Alignment BackColor Font	2 – Center Palette: <putih> Size: 24
Command1	Caption	OK
Command2	Caption	Selesai



Gambar 6.7 Hasil pengubahan property pada objek, bandingkan dengan Gambar 6.5.

6.4.3 Menulis Kode Program : Property Test

Buka Jendela Code dan kenali bagian-bagian di dalamnya dengan cara men-double-click Command Button “OK” (Command1) atau dengan menekan tombol F7.



Gambar 6.8 Tampilan jendela Code

Program yang berbasis Windows bersifat *event-driven*, artinya program bekerja berdasarkan event yang terjadi pada object di dalam program tersebut. Misalnya, jika seorang user meng-klik sebuah tombol maka program akan memberikan “reaksi” terhadap event klik tersebut. Program akan memberikan “reaksi” sesuai dengan kode-kode program yang dibuat untuk suatu event pada object tertentu.


Pilih object Command1 pada bagian Object Selector. Secara otomatis pada bagian Code Editor akan muncul blok kode program berikut :

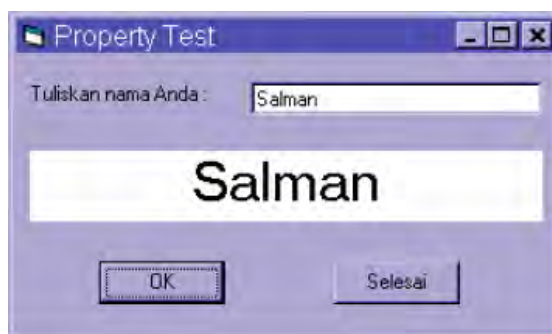
```
Private Sub Command1_Click()  
  
End Sub
```

Event Click merupakan event *default* (standar) dari object Command1, Anda bisa melihat event lainnya” pada bagian Event Selector. Pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub Command1_Click()  
    Label2.Caption = Text1.Text  
End Sub  
  
Private Sub Command2_Click()  
    End  
End Sub
```

Simpan Project1 (nama file : **Latihan.vbp**) dan Form1 (nama file : **Lat1.frm**). VB 6 akan menyimpan program yang Anda buat ke dalam beberapa file yang saling terkait, antara lain : file project (*.vbp), file form (*.frm) dan file module (*.bas).

Coba jalankan Project1 dengan meng-klik tombol Start () pada bagian Toolbar atau tekan tombol F5 pada keyboard. Ketikkan “Salman” di dalam TextBox kemudian klik tombol OK. Maka akan tampil hasil seperti pada Gambar 6.9. Bila ingin keluar klik tombol Selesai.



Gambar 6.9 Hasil Run program

Penjelasan Kode Program

Berikut diberikan penjelasan bagaimana cara kerja program sederhana di atas.

```
Private Sub Command1_Click()  
    Label2.Caption = Text1.Text  
End Sub
```

Ketika seorang *user* meng-klik tombol OK (object Command1) maka isi tulisan pada object Label2 akan sama dengan isi tulisan pada object Text1. Property Caption pada obyek Label2 digunakan untuk menampilkan text tertentu di obyek tersebut.

Selanjutnya untuk keluar dari program dituliskan kode berikut :

```
Private Sub Command2_Click()  
    End  
End Sub
```

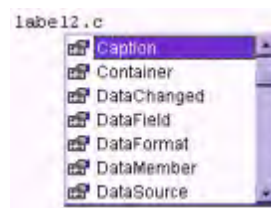
Ketika seorang *user* meng-klik tombol Selesai (object Command2) maka program akan berhenti. Perintah End digunakan untuk menutup program.

Menampilkan tulisan merupakan salah satu property yang dimiliki oleh komponen Label. Property isi tulisan pada komponen Label adalah Caption, sedangkan pada komponen Textbox adalah Text.

Cara penulisan (*syntax*) kode program untuk mengatur property sebuah object adalah sebagai berikut :

nama_object.property = value

Perhatikan, setelah Anda mengetikkan tanda titik (dot) dibelakang nama_object maka VB 6 akan menampilkan daftar property dan method yang dimiliki oleh nama_object :

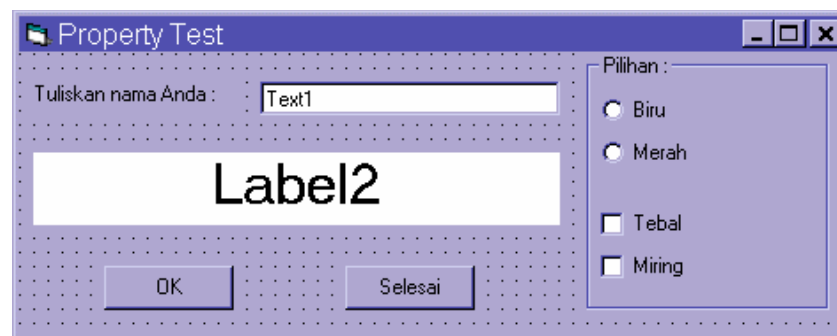


Gambar 6.10 Fasilitas pengenalan terhadap property maupun method untuk objek pada saat pengetikan di Layar Code secara otomatis.

Fasilitas ini disebut Intellisense, dan sangat membantu Anda agar tidak salah mengetikkan nama_object maupun property-nya.

6.4.4 Modifikasi Kode Program

Ubah tampilan UI-nya menjadi seperti ini pada Gambar 6.11. (Tambahkan komponen Frame, OptionButton dan CheckBox).



Gambar 6.11 Form baru dengan penambahan objek frame, optionButton dan CheckBox.

Atur property object-object tambahannya sebagai berikut:

Object	Properties	Value
Frame1	Caption	Pilihan :
Option1	Caption	Biru
Option2	Caption	Merah
Check1	Caption	Tebal
Check2	Caption	Miring

Obyek Frame1 harus dibuat terlebih dahulu di form, dan selanjutnya obyek Option dan Check dibuat di dalam Frame1 tersebut. Tujuan pembuatan berbagai pilihan tersebut adalah memperkenalkan pada Anda fasilitas Option dan CheckBox yang digunakan untuk membuat pilihan dalam aplikasi.

Tambahkan kode programnya sebagai berikut :

```
Private Sub Option1_Click()  
    Label2.ForeColor = vbBlue  
End Sub  
  
Private Sub Option2_Click()  
    Label2.ForeColor = vbRed  
End Sub  
  
Private Sub Check1_Click()  
    Label2.FontBold = Check1.Value  
End Sub  
  
Private Sub Check2_Click()  
    Label2.FontItalic = Check2.Value  
End Sub
```

Simpan kembali Form1, kemudian coba jalankan Project1 dan lihat hasilnya :



6.5 Menggunakan Event dan Method.

Berikut perhatikan perbedaan antara property dan method.

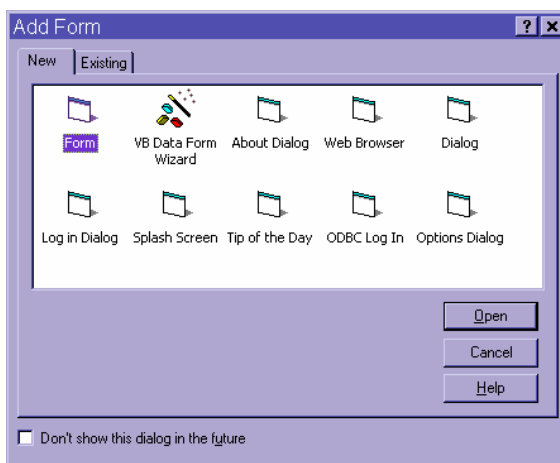
6.5.1 Membuat *User Interface* (UI)

Buka kembali project **Latihan.vbp** :



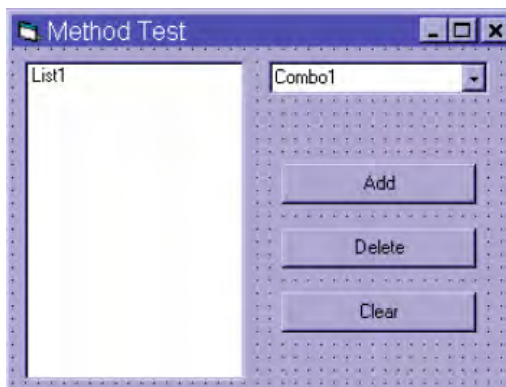
- Klik tab Existing.
- Pilih folder masing-masing pada bagian Look in.
- Pilih file Latihan.vbp.
- Klik tombol Open.

Tambahkan Form baru ke dalam Project :



- Klik menu Project > Add Form.
- Klik tab New dan pilih Form.
- Klik tombol Open.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen ListBox, ComboBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form2	Caption StartPosition	Method Test 2 – CenterScreen
List1	-	-
Combo1	-	-
Command1	Caption	Add
Command2	Caption	Delete
Command3	Caption	Clear

6.5.2 Menulis Kode Program : Method Test

Pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub Form_Load()
    Combo1.AddItem "Umar"
    Combo1.AddItem "Salman"
    Combo1.AddItem "Halimah"
    Combo1.AddItem "Shafira"
End Sub

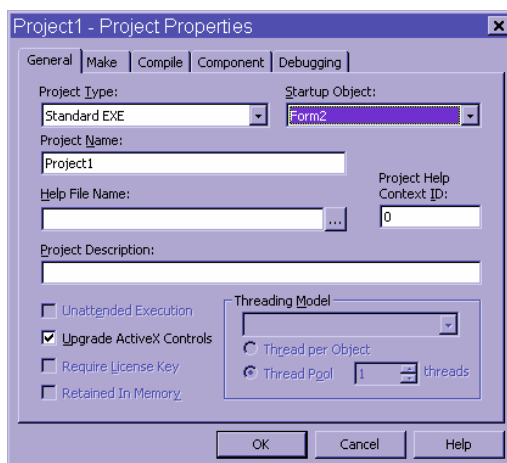
Private Sub Command1_Click()
    List1.AddItem Combo1.Text
End Sub

Private Sub Command2_Click()
    List1.RemoveItem List1.ListIndex
End Sub

Private Sub Command3_Click()
    List1.Clear
End Sub
```

Simpan Form2 (nama file : **Lat2.frm**).

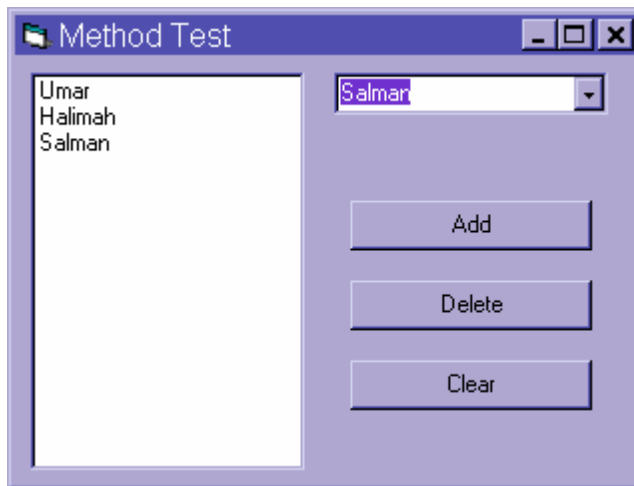
Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form2:



Keterangan: Startup Object menentukan object mana yang muncul setiap kali sebuah project dijalankan.

Coba jalankan Project1:

- Untuk menambah isi ListBox : pilih “Umar” di dalam ComboBox kemudian klik tombol Add. Ulangi untuk nama-nama yang lain secara acak.
- Untuk menghapus salah satu isi ListBox : pilih “Shafira” di dalam ListBox kemudian klik tombol Delete.
- Untuk menghapus semua isi ListBox : klik tombol Clear.



Penjelasan kode program:

```
Private Sub Form_Load()  
    Combo1.AddItem "Umar"  
    Combo1.AddItem "Salman"  
    Combo1.AddItem "Halimah"  
    Combo1.AddItem "Shafira"  
End Sub
```

Pada saat Form2 di-load ke dalam memori komputer, object Combo1 diisi dengan beberapa nama.

```
Private Sub Command1_Click()  
    List1.AddItem Combo1.Text  
End Sub
```

Ketika seorang *user* meng-klik tombol Add (object Command1) maka isi object List1 ditambah sesuai dengan isi object Combo1 yang sudah dipilih.

```
Private Sub Command2_Click()  
    List1.RemoveItem List1.ListIndex  
End Sub
```

Ketika seorang *user* meng-klik tombol Delete (object Command2) maka isi object List1 yang dipilih akan dihapus. Catatan : bila belum ada isi yang dipilih akan terjadi *error*.

```
Private Sub Command3_Click()  
    List1.Clear  
End Sub
```

Ketika seorang user meng-klik tombol Clear (object Command3) maka semua isi object List1 akan dihapus.

Cara penulisan (*syntax*) kode program untuk menggunakan method dari sebuah object adalah sebagai berikut :

nama_object.method [argument]

Perhatikan, setelah Anda mengetikkan tanda titik (dot) dibelakang nama_object maka VB 6 akan menampilkan daftar property dan method yang dimiliki oleh nama_object :

Komponen *ComboBox* dan *ListBox* memiliki beberapa method untuk mengatur isi (*item*) di dalamnya, yaitu :

- **AddItem** – method ini akan menambah isi sesuai dengan argument yang diberikan berupa tulisan. Setiap kali ada isi yang ditambahkan akan diberikan nomer index secara otomatis dimulai dari 0 (nol).

Contoh :

Combo1.AddItem “Umar” → “Umar” adalah argument sebagai isi yang akan ditambahkan ke dalam object Combo1.

List1.AddItem Combo1.Text → property Text dari object Combo1 adalah argument sebagai isi yang akan ditambahkan ke dalam object List1.

- **RemoveItem** – method ini akan menghapus isi sesuai dengan argument yang diberikan berupa nomer index dari isi yang dipilih. Setiap kali ada isi yang dihapus maka nomer index akan disusun ulang secara otomatis. Bila isinya tidak ada (kosong) nomer index-nya adalah -1.

Contoh :

List1.RemoveItem List1.ListIndex → property ListIndex dari object List1 adalah argument sebagai nomer index isi yang akan dihapus.

- **Clear** – method ini akan menghapus semua isi (tidak membutuhkan argument).

Contoh :

List1.Clear → semua isi object List1 akan dihapus.

6.6 Pendefinisian Data dan Variabel

Ketika seorang *user* (pengguna) menggunakan sebuah program komputer, seringkali komputer memintanya untuk memberikan informasi. Informasi ini kemudian disimpan atau diolah oleh komputer. Informasi inilah yang disebut dengan **DATA**.

Visual Basic 6 mengenal beberapa type data, antara lain :

- **String** adalah type data untuk teks (huruf, angka dan tanda baca).
- **Integer** adalah type data untuk angka bulat.
- **Single** adalah type data untuk angka pecahan.
- **Currency** adalah type data untuk angka mata uang.
- **Date** adalah type data untuk tanggal dan jam.
- **Boolean** adalah type data yang bernilai TRUE atau FALSE.

Data yang disimpan di dalam memory komputer membutuhkan sebuah wadah. Wadah inilah yang disebut dengan **VARIABEL**. Setiap variabel untuk menyimpan data dengan type tertentu membutuhkan alokasi jumlah memory (*byte*) yang berbeda.

Variabel dibuat melalui penulisan deklarasi variabel di dalam kode program :

Dim <nama_variabel> As <type_data>

Contoh : Dim nama_user As String

Aturan di dalam penamaan variabel :

1. Harus diawali dengan huruf.
2. Tidak boleh menggunakan spasi. Spasi bisa diganti dengan karakter underscore (_).
3. Tidak boleh menggunakan karakter-karakter khusus (seperti : +, -, *, /, <, >, dll).
4. Tidak boleh menggunakan kata-kata kunci yang sudah dikenal oleh Visual Basic 6 (seperti : dim, as, string, integer, dll).

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

<nama_variabel> = <nilai_data>

Contoh : nama_user = "krisna"

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 (*lifetime*) :

- **Variabel global** adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan.
- **Variabel lokal** adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan.

Variabel yang nilai datanya bersifat tetap dan tidak bisa diubah disebut **KONSTANTA**. Penulisan deklarasi konstanta di dalam kode program :

```
Const <nama_kontanta> As <type_data> = <nilai_data>
```

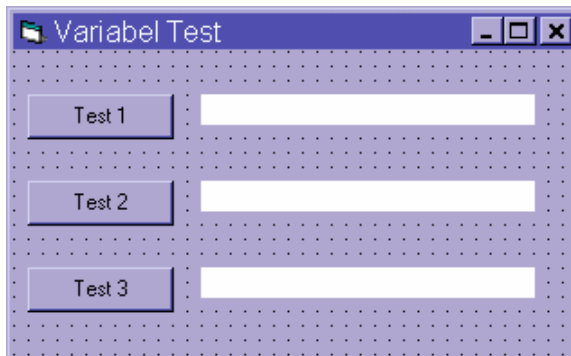
Contoh : Const tgl_gajian As Date = #25/09/2003#

Contoh Program : Variabel Test

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen CommandButton dan Label)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form3	Caption	Variable Test
	StartPosition	2 – CenterScreen
Command1-3	Caption	Test 1
		Test 2
		Test 3
Label1-3	BackColor	Palette : <putih>
	Caption	<kosong>

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Dim test2 As Integer

Private Sub Command1_Click()
    Dim test1 As String
    test1 = "nusantara"
    Label1.Caption = test1
    Label2.Caption = test2
    Label3.Caption = test3
End Sub

Private Sub Command2_Click()
    test2 = 10
    Label1.Caption = test1
    Label2.Caption = test2
```

```

Label3.Caption = test3
End Sub

Private Sub Command3_Click()
    Const test3 As Single = 90.55
    Label1.Caption = test1
    Label2.Caption = test2
    Label3.Caption = test3
End Sub

```

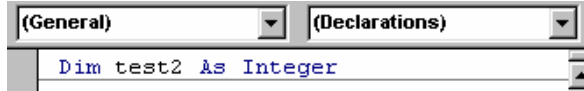
Simpan Form3 (nama file : **Lat3.frm**).

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form3.

Coba jalankan Project1. Perhatikan sifat variabel global, variabel lokal dan konstanta !

Klik Tombol	Nilai Var. Test1	Nilai Var. Test2	Nilai Var. Test3
Test1	nusantara	Kali Pertama : 0 Berikutnya : 10	<kosong>
Test2	<kosong>	10	<kosong>
Test3	<kosong>	10	90,55

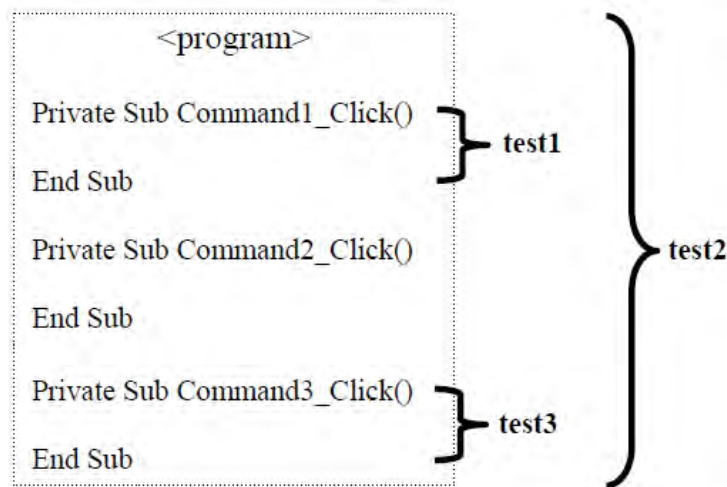
Penjelasan Kode Program

Kode Program	Deskripsi
<pre>Dim test2 As Integer</pre>	<p>Variabel test2 dideklarasikan untuk data bertipe integer sebagai variabel global sehingga bisa digunakan oleh semua blok kode program.</p> <p>Catatan : setiap variabel global harus selalu dideklarasikan pada bagian atas kode program pastikan Object Selector berisi (General) dan Event Selector berisi (Declarations).</p> 
<pre>Private Sub Command1_Click() Dim test1 As String</pre>	<p>Bila <i>user</i> meng-klik tombol Test1 :</p> <p>Variabel test1 dideklarasikan untuk data bertipe string sebagai variabel lokal khusus untuk blok kode Command1_Click.</p> <p>Catatan : setiap variabel lokal dideklarasikan pada bagian dalam blok kode program tertentu. Blok kode program disebut procedure (akan dibahas dalam bab khusus).</p>

Kode Program	Deskripsi
<pre> test1 = "nusantara" Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub </pre>	<p>Variabel test1 diisi dengan data nusantara (string). Ingat tanda “...” sebagai pembatas.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none"> Label1 akan menampilkan teks nusantara, karena variabel test1 berisi nusantara. Label2 akan menampilkan angka 0, karena variabel test2 belum ada isinya. Variabel test2 boleh digunakan karena bersifat global. Label3 tidak menampilkan apa-apa (kosong), karena variabel test3 hanya bersifat lokal untuk blok kode Command3_Click.
<pre> Private Sub Command2_Click() test2 = 10 Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub </pre>	<p>Bila <i>user</i> meng-klik tombol Test2 :</p> <p>Variabel test2 diisi dengan data 10 (integer). Variabel test2 boleh diisi/diganti isinya karena bersifat global.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none"> Label1 tidak menampilkan apa-apa (kosong), karena variabel test1 hanya bersifat lokal untuk blok kode Command1_Click. Label2 akan menampilkan angka 10, karena variabel test2 isinya angka 10. Label3 tidak menampilkan apa-apa (kosong), karena variabel test3 hanya bersifat lokal untuk blok kode Command3_Click.
<pre> Private Sub Command3_Click() Const test3 As Single = 90.55 Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub </pre>	<p>Bila <i>user</i> meng-klik tombol Test3 :</p> <p>Variabel test3 dideklarasikan sekaligus diisi dengan data 90,55 (single) – sebagai konstanta.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none"> Label1 tidak menampilkan apa-apa (kosong), karena variabel test1 hanya bersifat lokal untuk blok kode Command1_Click.

Kode Program	Deskripsi
	<ul style="list-style-type: none"> Label2 akan menampilkan angka 10, karena variabel test2 isinya masih angka 10. Variabel test2 boleh digunakan karena bersifat global. Label3 akan menampilkan angka 90,55, karena variabel test3 isinya angka 90,55.

Scope dan LifeTime masing-masing variabel :



Untuk membuktikan bahwa nilai data sebuah konstanta bersifat tetap, modifikasi kode program sbb :

```

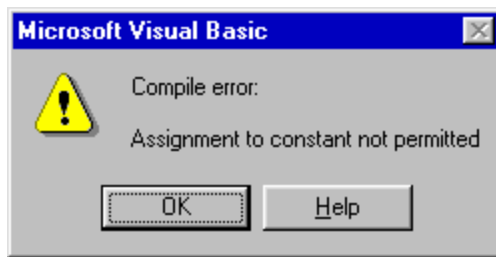
Private Sub Command3_Click()
    Const test3 As Single = 90.55

    test3 = 50.22

    Label1.Caption = test1
    Label2.Caption = test2
    Label3.Caption = test3
End Sub
  
```

Perhatikan : penulisan angka pecahan (single) pada kode program tetap menggunakan tanda titik (.) meskipun regional setting komputernya sudah Indonesia. Bila tidak akan muncul pesan error.

Coba jalankan Project1, kemudian klik tombol Test 3. Akan muncul pesan error:



Klik OK, kemudian stop programnya dengan meng-klik tombol End pada Toolbar.

Agar program tetap berjalan tanpa harus menghapus kode program yang salah, maka tambahkan tanda petik-tunggal (') di depan kode program yang salah tersebut (contoh : 'test3 = 50.22). Teks setelah tanda petiktunggal oleh Visual Basic 6 diartikan sebagai komentar dan tidak akan di-eksekusi ketika program berjalan.

Visual Basic 6 membolehkan sebuah variabel tidak dideklarasikan terlebih dahulu (dengan perintah Dim atau Const), tetapi hal ini akan membuat error pada kode program agak sulit dilacak. Jadi, sebaiknya setiap variabel selalu dideklarasikan (baik yang lokal maupun global). Agar dapat diketahui variabel mana yang belum dideklarasikan, pada bagian atas kode program tambahkan perintah berikut :

Option Explicit

Coba jalankan Project1, kemudian klik tombol Test 1. Akan muncul pesan error:



Klik OK, kemudian stop programnya dengan meng-klik tombol End pada Toolbar.



BAB 7 PROGRAM PILIHAN (DECISION)

Tujuan Pembelajaran Umum

1. Memahami program dengan algoritma pilihan atau keputusan (decision)

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat memahami penggunaan operator aritmatika, relasional dan logika dalam program
2. Mahasiswa dapat membuat program dengan algoritma pilihan baik sederhana maupun rumit dengan IF .. THEN maupun SELECT .. CASE

7.1 Operator Relasional dan Logikal

OPERATOR di dalam bahasa pemrograman bisa diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol operator bisa berupa karakter ataupun kata khusus.

Visual Basic 6 mengenal tiga jenis operator, yaitu :

1. **Operator Aritmatika** digunakan untuk operasi matematis terhadap nilai data. Simbol-simbol yang digunakan:

Tabel 7.1 Daftar simbol operator aritmatika dan contohnya

Simbol	Operasi Matematis	Contoh
\wedge	pemangkatan	$5 \wedge 2$ hasilnya 25
*	perkalian	$5 * 2$ hasilnya 10
/	pembagian (hasil pecahan)	$5 / 2$ hasilnya 2,5
\	pembagian (hasil bulat)	$5 \setminus 2$ hasilnya 2
Mod	sisa pembagian	$5 \text{ Mod } 2$ hasilnya 1
+	penjumlahan	$5 + 2$ hasilnya 7
-	pengurangan	$5 - 2$ hasilnya 3
&	penggabungan string	$5 \& 2$ hasilnya 52

2. **Operator Perbandingan** digunakan untuk operasi yang membandingkan nilai data. Simbol-simbol yang digunakan :

Tabel 7.2 Daftar simbol operator relasional (perbandingan) beserta contoh

Simbol	Operasi Perbandingan	Contoh
<	lebih kecil	5 < 2 hasilnya FALSE
>	lebih besar	5 > 2 hasilnya TRUE
<=	lebih kecil atau sama dengan	5 <= 2 hasilnya FALSE
>=	lebih besar atau sama dengan	5 >= 2 hasilnya TRUE
=	sama dengan	5 = 2 hasilnya FALSE
<>	tidak sama dengan	5 <> 2 hasilnya TRUE

3. **Operator Logika** digunakan untuk operasi yang membandingkan suatu perbandingan. Simbol yang digunakan :

Tabel 7.3 Daftar simbol operator logika dan contohnya

Simbol	Operasi Logika	Contoh
Or	atau	(5 < 2) Or (5 > 2) hasilnya TRUE
And	dan	(5 < 2) And (5 > 2) hasilnya FALSE
Not	Tidak	Not (5 < 2) hasilnya TRUE

Contoh Program : Operator Test

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form4	Caption StartUpPosition	Operator Test 2 – CenterScreen
Label1-2	Caption	Var 1 : Var 2 :
Text1-2	Text	<kosong>
Frame1-3	Caption	Op. Aritmatika Op. Perbandingan Op. Logika
Option1-14	Caption	+, -, *, /, && >, <, =, <>, >=, <= Not, Or, And
Label3	Alignment BackColor ForeColor Caption	2 – Center Palette : <putih> Palette : <merah> <kosong>

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```

Dim var1 As Single, var2 As Single

Dim hasil As Single

Private Sub Option1_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 + var2
    Label3.Caption = hasil
End Sub

```

```

Private Sub Option2_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 - var2
    Label3.Caption = hasil
End Sub

Private Sub Option3_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 * var2
    Label3.Caption = hasil
End Sub

Private Sub Option4_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 / var2
    Label3.Caption = hasil
End Sub

Private Sub Option5_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 & var2
    Label3.Caption = hasil
End Sub

Private Sub Option6_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = (var1 > var2)
    'Label3.Caption = hasil
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option7_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = (var1 < var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option8_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = (var1 = var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option9_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = (var1 <> var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option10_Click()
    var1 = Text1.Text

```

```

        var2 = Text2.Text
        hasil = (var1 >= var2)
        Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option11_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = (var1 <= var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option12_Click()
    var1 = IIf(Text1.Text = "True", -1, 0)
    hasil = Not (var1)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option13_Click()
    var1 = IIf(Text1.Text = "True", -1, 0)
    var2 = IIf(Text2.Text = "True", -1, 0)
    hasil = (var1 Or var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

Private Sub Option14_Click()
    var1 = IIf(Text1.Text = "True", -1, 0)
    var2 = IIf(Text2.Text = "True", -1, 0)
    hasil = (var1 And var2)
    Label3.Caption = Format(hasil, "True/False")
End Sub

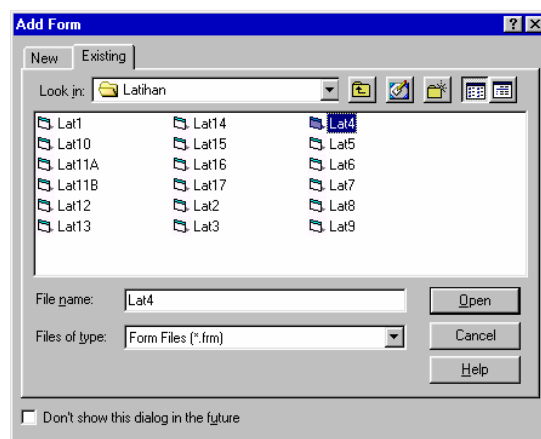
```

Simpan Form4 (nama file : **Lat4.frm**).

Catatan :

Untuk menambahkan file Form ke dalam sebuah Project, lakukan langkah-langkah berikut ini :

1. Klik menu Project > Add Form. Kemudian klik tab Existing.



2. Tentukan lokasi file pada bagian Look in, lalu pilih filenya.
3. Klik tombol Open.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form4.

Coba jalankan Project1. Perhatikan hasil setiap operasi !

Penjelasan Kode Program

Kode Program	Deskripsi
<pre>Dim var1 As Single, var2 As Single Dim hasil As Single</pre>	<p>Deklarasi variabel global : var1, var2 dan hasil untuk menyimpan data bertipe single.</p>
<pre>Private Sub Option1_Click() var1 = Text1.Text var2 = Text2.Text hasil = var1 + var2 Label3.Caption = hasil End Sub ...</pre>	<p>Ketika Option1 di-klik oleh user: Simpan angka yang diketik pada Text1 → var1. Simpan angka yang diketik pada Text2 → var2. Jumlahkan var1 dengan var2 → hasil. Tampilkan hasil pada Label3.</p> <p>Catatan:</p> <ul style="list-style-type: none"> Prosesnya sama untuk Option2 s/d Option 5. Operator & sering digunakan untuk menggabungkan data string : a & b → ab. Akan terjadi error bila Text1 atau Text2 kosong atau bukan diketik angka. Penanganan error akan dibahas pada bab lain.
<pre>Private Sub Option6_Click() var1 = Text1.Text var2 = Text2.Text hasil = (var1 > var2) 'Label3.Caption = hasil Label3.Caption = Format(hasil, "True/False") End Sub ...</pre>	<p>Ketika Option6 di-klik oleh user : Simpan angka yang diketik pada Text1 → var1. Simpan angka yang diketik pada Text2 → var2. Bandingkan apakah var1 lebih besar dari var2. Tampilkan hasil perbandingan pada Label3.</p> <p>Catatan:</p> <ul style="list-style-type: none"> Prosesnya sama untuk Option7 s/d Option11. Hasil dari suatu perbandingan adalah True (benar) atau False (salah). Komputer menampilkan True dengan angka -1 dan False dengan angka 0. Untuk mengubah angka -1 menjadi kata "True" dan angka 0 menjadi kata "False" digunakan perintah Format.

Kode Program	Deskripsi
<pre>Private Sub Option12_Click() var1 = IIf(Text1.Text = "True", -1, 0) hasil = Not (var1) Label3.Caption = Format(hasil, "True/False") End Sub</pre>	<p>Ketika Option12 di-klik oleh user:</p> <p>Bila Text1 diketik kata True ubah menjadi angka -1 selain itu ubah menjadi angka 0 → var1</p> <p>Bandingkan var1 menggunakan operator Not. Tampilkan hasilnya pada Label3.</p> <p>Catatan :</p> <p>Prosesnya sama untuk Option13 dan Option14.</p> <ul style="list-style-type: none"> ▪ Untuk mengubah kata “True” menjadi angka -1 dan selain itu menjadi angka 0 digunakan perintah IIF. Perintah IIF akan dibahas pada bab lain. ▪ Komputer hanya bisa memproses perbandingan dengan menggunakan angka -1 dan 0. ▪ Data string bersifat case sensitive. Jadi, kata “True” harus ditulis benar (huruf T kapital), karena “True” berbeda dengan “true” atau “tRUE” dll.

Coba lengkapi hasil operasi logika berikut ini :

Var1	Var2	Or	And
True	True		
True	False		
False	True		
False	False		

Bisa Anda bedakan sifat operator Or dengan And ?

7.2 Pilihan dengan Blok - IF

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu :

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi. Akan dibahas pada bab ini.
2. **Struktur kontrol pengulangan** - digunakan untuk melakukan pengulangan kode program. Akan dibahas pada bab selanjutnya.

Ada dua bentuk struktur kontrol keputusan, yaitu :

1. Struktur **IF...THEN**. Akan dibahas pada bab ini.
2. Struktur **SELECT...CASE**. Akan dibahas pada bab selanjutnya.

Bentuk penulisan (*syntax*) struktur IF...THEN :

1. **IF <kondisi> THEN <kode program>**

Bila <kondisi> bernilai True maka <kode program> akan dikerjakan.

2. **IF <kondisi> THEN**

<blok kode program 1>

ELSE

<blok kode program 2>

END IF

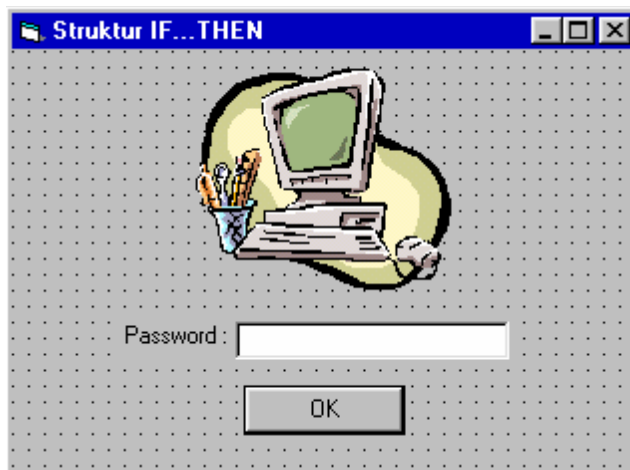
Bila <kondisi> bernilai True maka <blok kode program 1> akan dikerjakan, tetapi bila <kondisi> bernilai False maka <blok kode program 2> yang akan dikerjakan.

Contoh Program : Struktur IF...THEN

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen Image, Label, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form5	Caption StartPosition	Struktur IF...THEN 2 – CenterScreen
Image1	Stretch Picture Visible	True Komputer.wmf False
Label1	Caption	Password :
Text1	PasswordChar Text	* <kosong>
Command1	Caption Default	OK True

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub Command1_Click()  
If Text1.Text="nusantara" Then Image1.Visible=True  
End Sub
```

Simpan Form5 (nama file : **Lat5.frm**).

Catatan

Anda bisa men-download file Lat5.frm dan Komputer.wmf.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form5.

Coba jalankan Project1 :

- Ketikkan sembarang teks pada TextBox lalu klik tombol OK atau tekan Enter → tidak terjadi apaapa.
- Ketikkan “nusantara” pada TextBox lalu klik tombol OK atau tekan Enter → gambar komputer akan muncul.

Penjelasan kode program :

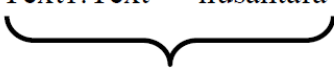
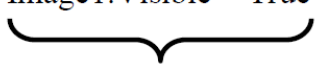
If Text1.Text = "nusantara" Then Image1.Visible = True

kondisi

 Image1.Visible = True

**kode program yang dikerjakan
bila kondisi TRUE**

Modifikasi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()  
    If Text1.Text = "nusantara" Then  
        Image1.Visible = True  
        Text1.Enabled = False  
        Command1.Enabled = False  
    Else  
        MsgBox "Passwordnya Salah !"  
        Text1.Text = ""  
        Text1.SetFocus  
    End If  
End Sub
```

Coba jalankan Project1:

- Ketikkan sembarang teks pada TextBox lalu klik tombol OK atau tekan Enter → muncul kotak pesan “Passwordnya Salah !”. Klik tombol OK pada kotak pesan tersebut.
- Ketikkan “nusantara” pada TextBox lalu klik tombol OK atau tekan Enter → gambar komputer akan muncul. TextBox dan tombol OK menjadi *disable* (tidak bisa digunakan).

Penjelasan kode program :



Catatan tambahan :

- Teks “nusantara” harus diketik huruf kecil semua. Ingat : data string bersifat *case sensitive* !
- Agar password-nya tidak bersifat *case sensitive*, modifikasi pernyataan kondisinya menjadi :

If LCASE(Text1.Text) = "nusantara" Then

Fungsi **LCASE** adalah untuk mengkonversi semua string yang diinput ke Text1.Text menjadi huruf kecil, walaupun user menginputnya dengan huruf kapital.

TANTANGAN :

Coba Anda modifikasi struktur kontrol dan kode programnya sehingga seorang *user* hanya mempunyai 3 kali kesempatan untuk mengetikkan password-nya. Setelah 3 kali kesempatan dan password-nya masih salah maka program akan otomatis berhenti.

7.3 Pilihan dengan Blok SELECT-CASE

Bentuk penulisan (*syntax*) struktur SELECT...CASE :

```
SELECT CASE <pilihan>
CASE <pilihan 1>
    <blok kode program 1>
CASE <pilihan 2>
    <blok kode program 2>
CASE <pilihan n>
    <blok kode program n>
[CASE ELSE
    <blok kode program x>]
END SELECT
```

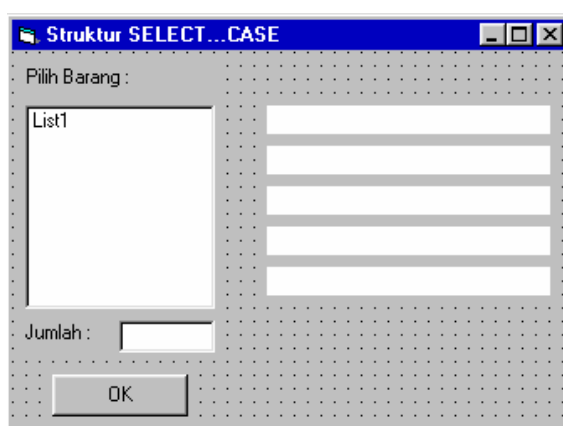
Bila <pilihan> sesuai dengan <pilihan 1> maka <blok kode program 1> akan dikerjakan, dst. Tetapi bila <pilihan> tidak ada yang sesuai dengan <pilihan 1> s/d <pilihan n> maka <blok kode program x> yang akan dikerjakan.

Contoh Program : Struktur SELECT...CASE

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen Label, ListBox, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form6	Caption	Struktur SELECT...CASE

Object	Properties	Value
	StartPosition	2 – CenterScreen
Label1	Caption	Pilih Barang :
List1	-	-
Label2	Caption	Jumlah :
Text1	Text	<kosong>
Command1	Caption	OK
Label3-7	Name BackColor Caption	lblBarang, lblHarga, lblJumlah, lblDiskon, lblTotal Palette : <putih> <kosong>

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```

Private Sub Form_Load()
    List1.AddItem "Disket"
    List1.AddItem "Buku"
    List1.AddItem "Kertas"
    List1.AddItem "Pulpen"
End Sub

Private Sub Command1_Click()
    Dim harga As Currency, total As Currency
    Dim jumlah As Integer
    Dim diskon As Single
    Dim satuan As String

    If List1.Text = "" Then
        MsgBox "Anda belum memilih barang !!"
        List1.ListIndex = 0
        Exit Sub
    End If

    If Text1.Text = "" Then
        MsgBox "Anda belum mengisi jumlah barang !!"
        Text1.SetFocus
        Exit Sub
    End If

    Select Case List1.Text
        Case "Disket"
            harga = 35000
            satuan = "Box"
        Case "Buku"
            harga = 20000
            satuan = "Lusin"
        Case "Kertas"
            harga = 25000
            satuan = "Rim"
        Case "Pulpen"
            harga = 10000
    End Select

```

```

        satuan = "Pak"
    End Select

    lblBarang.Caption = "Barang : " & List1.Text
    lblHarga.Caption = "Harga : " & Format(harga,
"Currency") & "/" & satuan
    lblJumlah.Caption = "Jumlah : " & Text1.Text & " " &
satuan
    jumlah = Text1.Text

    Select Case jumlah
        Case Is < 10
            diskon = 0
        Case 10 To 20
            diskon = 0.15
        Case Else
            diskon = 0.2
    End Select

    total = jumlah * (harga * (1 - diskon))
    lblDiskon.Caption = "Diskon : " & Format(diskon, "0 %")
    lblTotal.Caption = "Total Bayar : " & Format(total,
"Currency")

End Sub

```

Simpan Form6 (nama file : **Lat6.frm**).

Catatan :

Anda bisa men-download file Lat6.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form6.

Coba jalankan Project1 :

- List1 akan terisi nama-nama barang.
- Klik tombol OK → akan muncul kotak pesan "Anda belum memilih barang !!".
- Klik OK, nama barang pertama akan terpilih secara otomatis. Anda boleh memilih nama barang yang lainnya.
- Klik tombol OK → akan muncul kotak pesan "Anda belum mengisi jumlah barang !!".
- Klik OK, text1 akan menjadi focus. Isi jumlah barangnya, misalnya : 10.
- Klik tombol OK → akan tampil : nama barang, harga barang (per-satuannya), jumlah barang (dengan satuannya), diskon dan total bayar.
- Coba ganti pilihan nama barang (pada List1) dan jumlah barang (pada Text1) lalu klik lagi tombol OK.

Penjelasan kode program :

Kode Program	Deskripsi
--------------	-----------

Kode Program	Deskripsi
Select Case List1.Text	cek barang yang dipilih :
Case "Disket"	bila barang = Disket
harga = 35000	
satuan = "Box"	
Case "Buku"	bila barang = Buku
harga = 20000	
satuan = "Lusin"	
Case "Kertas"	bila barang = Kertas
harga = 25000	
satuan = "Rim"	
Case "Pulpen"	bila barang = Pulpen
harga = 10000	
satuan = "Pak"	
End Select	
tampilkan hasilnya	
lblBarang.Caption = "Barang : " & List1.Text	
lblHarga.Caption = "Harga : " & Format(harga, "Currency") & "/" & satuan	
lblJumlah.Caption = "Jumlah : " & Text1.Text & " " & satuan	
jumlah = Text1.Text	
Select Case jumlah	cek jumlah barang:
Case Is < 10	bila jumlah < 10 → diskon = 0%
diskon = 0	
Case 10 To 20	bila jumlah 10-20 → diskon = 15%
diskon = 0.15	
Case Else	bila jumlah > 20 → diskon = 20%
diskon = 0.2	
End Select	
Hitung total bayar dan tampilkan hasilnya	
total = jumlah * (harga * (1 - diskon))	
lblDiskon.Caption = "Diskon : " & Format(diskon, "0 %")	
lblTotal.Caption = "Total Bayar : " & Format(total, "Currency")	

Catatan :

- Bila jumlah barang diisi dengan selain angka akan muncul pesan error.
- Untuk mengecek isi Text1 angka atau bukan, tambahkan kode program berikut :

```

If Not IsNumeric(Text1.Text) Then
    MsgBox "Isi jumlah barang harus angka !!"
    Text1.SetFocus
    Exit Sub
End If

```




BAB 8

PROGRAM PENGULANGAN (REPETITION)

Tujuan Pembelajaran Umum

1. Memahami program dengan algoritma pengulangan (repetition)

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat membuat program dengan algoritma pengulangan dengan menggunakan pengulangan tertentu (FOR .. NEXT)
2. Mahasiswa dapat membuat program dengan algoritma pengulangan dengan menggunakan pengulangan dengan syarat (DO .. LOOP)

Ada dua bentuk struktur kontrol pengulangan (*looping*), yaitu :

1. Struktur **FOR...NEXT**.
2. Struktur **DO...LOOP**.

8.1 Pengulangan dengan FOR .. NEXT

Bentuk penulisan (*syntax*) struktur For...Next :

```
FOR <pencacah> = <awal> TO <akhir> [STEP <langkah>]  
  <blok kode program>  
NEXT <pencacah>
```

- <pencacah> adalah variabel (tipe: integer) yang digunakan untuk menyimpan angka pengulangan.
- <awal> adalah nilai awal dari <pencacah>.
- <akhir> adalah nilai akhir dari <pencacah>.
- <langkah> adalah perubahan nilai <pencacah> setiap pengulangan. Sifatnya optional (boleh ditulis ataupun tidak). Bila tidak ditulis maka nilai <langkah> adalah 1.

8.2 Pengulangan dengan DO .. LOOP

1. **DO WHILE <kondisi>**
 <blok kode program>

LOOP

<blok kode program> akan diulang **selama** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

2. **DO UNTIL <kondisi>**
 <blok kode program>

LOOP

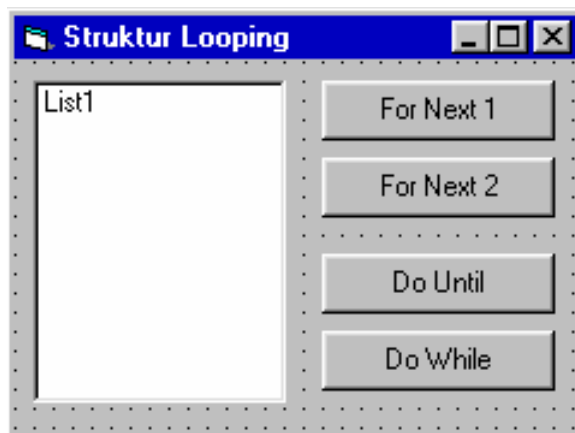
<blok kode program> akan diulang **sampai** <kondisi> bernilai TRUE.
Pengulangan berhenti bila <kondisi> sudah bernilai TRUE.

Contoh Program : Struktur Looping

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen ListBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form7	Caption StartPosition	Struktur Looping 2 – CenterScreen
List1	-	-
Command1-4	Caption	For Next 1 For Next 2 Do Until Do While

```
Dim i As Integer

Private Sub Command1_Click()
    List1.Clear
    For i = 1 To 100
        List1.AddItem "Angka " & i
    Next i
End Sub

Private Sub Command2_Click()
    List1.Clear
```

```

        For i = 100 To 1 Step -2
            List1.AddItem "Angka " & i
        Next i
    End Sub

    Private Sub Command3_Click()
        List1.Clear
        i = Asc("A")
        Do Until i > Asc("Z")
            List1.AddItem "Huruf " & Chr(i)
            i = i + 1
        Loop
    End Sub

    Private Sub Command4_Click()
        List1.Clear
        i = Asc("Z")
        Do While i >= Asc("A")
            List1.AddItem "Huruf " & Chr(i)
            i = i - 1
        Loop
    End Sub

```

Simpan Form7 (nama file : **Lat7.frm**).

Catatan

Anda bisa men-download file Lat7.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form7.

Coba jalankan Project1 :

- Klik Command1 → List1 akan berisi : Angka 1 s/d Angka 100.
- Klik Command2 → List1 akan berisi : Angka genap antara 1 s/d 100 dengan urutan *descending*.
- Klik Command3 → List1 akan berisi : Huruf A s/d Huruf Z.
- Klik Command4 → List1 akan berisi : Huruf A s/d Huruf Z dengan urutan *descending*.

Penjelasan kode program :

Kode Program	Deskripsi
For i = 1 To 100	pencacah (i) → 1 s/d 100
List1.AddItem "Angka " & i	kode program yang diulang
Next i	
For i = 100 To 1 Step -2	pencacah (i) → 1 s/d 100 dengan
List1.AddItem "Angka " & i	langkah -2 setiap pengulangan
Next i	kode program yang diulang
i = Asc("A")	nilai i awal → 65
Do Until i > Asc("Z")	pengulangan sampai nilai i > 91

Kode Program	Deskripsi
<pre> List1.AddItem "Huruf " & Chr(i) i = i + 1 Loop i = Asc("Z") Do While i >= Asc("A") List1.AddItem "Huruf " & Chr(i) i = i - 1 Loop </pre>	<p>kode program yang diulang</p> <p>nilai i ditambah 1</p> <p>nilai i awal → 91</p> <p>pengulangan selama nilai i >= 65</p> <p>kode program yang diulang</p> <p>nilai i dikurangi 1</p>

Catatan :

Bila ingin keluar dari struktur pengulangan sebelum pengulangannya selesai gunakan perintah **EXIT FOR** atau **EXIT DO**. Contoh :

```

For i = 1 To 100
    List1.AddItem "Angka " & i
    If i = 50 Then Exit For
Next i

Do Until i > Asc("Z")
    List1.AddItem "Huruf " & Chr(i)
    If Chr(i) = "M" Then Exit Do
    i = i + 1
Loop

```



BAB 9 PEMROGRAMAN DENGAN ARRAY

Tujuan Pembelajaran Umum

1. Memahami bentuk array dan penggunaannya dalam program

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat membuat variabel array dan menggunakan dalam program
2. Mahasiswa dapat membuat control array dan menggunakan dalam program

ARRAY merupakan sekumpulan nilai data yang “dikelompokkan” dalam sebuah variabel. Array digunakan bila ada beberapa nilai data yang tipe datanya sama dan akan mendapat perlakuan yang sama pula. Misalnya, ada 10 nilai data dengan tipe string dan akan diolah dengan cara yang sama, maka akan lebih mudah jika menggunakan sebuah array dibandingkan bila menggunakan 10 variabel yang berbeda. Setiap nilai data di dalam sebuah array disebut **elemen array** dan masing-masing dibedakan dengan nomer indeksny.

9.1 Menggunakan ARRAY

Sebuah array dideklarasikan dengan cara yang sama dengan variabel, yaitu menggunakan perintah Dim, kemudian diikuti dengan tanda kurung dan jumlah elemen arraynya dikurangi 1 :

```
Dim nama_array(jumlah_elemen - 1) As tipe_data
```

Contoh :

```
Dim NamaSiswa(99) As String → array NamaSiswa akan mempunyai  
elemen sebanyak 100 dengan nomer  
indeks mulai dari 0 s/d 99.
```

Selanjutnya untuk mengisi nilai data ke dalam array :

```
nama_array(no_indeks) = nilai_data
```

Contoh :

```
NamaSiswa(0) = "umar"  
NamaSiswa(1) = "fatimah"  
...
```

Untuk “mengosongkan” nilai data, bisa menggunakan struktur kontrol For...Next :

```
For i = 0 To 99  
    NamaSiswa(i) = ""
```

Next i

Lebih mudah bukan ?

Catatan :

- Agar nomer indeks array dimulai dari 1 (bukan 0), bisa menggunakan perintah **Option Base 1** sebelum perintah Dim-nya, contoh :

```
Option Base 1
Dim NamaSiswa(100) As String
```

- Atau bisa juga menggunakan kata **To** di dalam penentuan jumlah elemen array, contoh :

```
Dim NamaSiswa(1 To 100) As String
```

- Untuk mengetahui berapa jumlah elemen di dalam sebuah array, bisa menggunakan perintah **Ubound(nama_array)**, contoh :

```
Dim NamaSiswa(1 To 100) As String
Dim JmlElemen As Integer
JmlElemen = Ubound(NamaSiswa) → hasilnya = 100
```

- Bila jumlah elemen array yang dibutuhkan tidak diketahui atau ingin bisa diubah-ubah, maka bagian jumlah_elemen pada perintah Dim tidak perlu diisi, contoh :

```
Dim NamaSiswa() As String
```

Teknik ini biasa disebut sebagai array dinamis, biasanya untuk mengisi data yang berkembang terus, dan belum bisa ditentukan di awal jumlah elemennya.

- Untuk mengubah jumlah elemen array digunakan perintah **Redim** :

```
Redim [Preserve] nama_array(jumlah_elemen)
```

Contoh :

```
Redim NamaSiswa(1 To 150) → jumlah elemen array NamaSiswa
                             menjadi 150 dengan nomer
                             indeks dari 1 s/d 150.
```

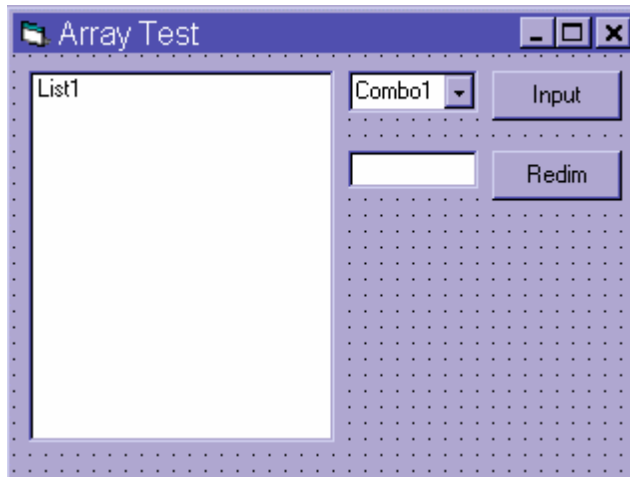
Penggunaan kata Preserve dapat dilihat pada contoh program.

Contoh Program: Array Test

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen ListBox, ComboBox, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut:

Object	Properties	Value
Form8	Caption StartPosition	Array Test 2 – CenterScreen
List1	-	-
Combo1	-	-
Text1	Text	<kosong>
Command1	Caption	Input
Command2	Caption	Redim

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```

Dim NamaSiswa(4) As String

Private Sub Form_Load()
    Dim i As Integer
    For i = 0 To 4
        Combo1.AddItem i
    Next i
    Combo1.ListIndex = 0
End Sub

Private Sub Command1_Click()
    Dim no As Integer, i As Integer
    no = CInt(Combo1.Text)
    NamaSiswa(no) = InputBox("Tuliskan nama siswa no : " &
no, "Input Nama Siswa")
    If NamaSiswa(no) <> "" Then
        List1.Clear
        For i = 0 To 4
            List1.AddItem "NamaSiswa(" & i & ")=" &
NamaSiswa(i)
        Next i
    End If
End Sub

```

```
End If
End Sub
```

Simpan Form8 (nama file : **Lat8.frm**).

Catatan :

Anda bisa men-download file Lat8.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form8.

Coba jalankan Project1 :

- Pilih nomer indeks elemen array NamaSiswa melalui Combo1 (antara 0 s/d 4)
- Klik Command1 → akan muncul kotak input untuk mengisi nama siswa
- Ketikkan nilai data untuk elemen array dengan nomer indeks yang dipilih, misalnya Salman
- Klik tombol OK
- Isi setiap elemen array NamaSiswa akan muncul di dalam List1
- Ulangi langkahnya untuk mengisi elemen array nomer indeks lainnya

Penjelasan kode program :

Kode Program	Deskripsi
<pre>Dim NamaSiswa(4) As String</pre>	Deklarasi array NamaSiswa dengan jumlah elemen 5
<pre>Private Sub Form_Load() Dim i As Integer For i = 0 To 4 Combo1.AddItem i Next i Combo1.ListIndex = 0 End Sub</pre>	Pada saat form di-load ke dalam memori : Isi Combo1 dengan angka 0 s/d 4 (sesuai dengan no indeks untuk setiap elemen di dalam array NamaSiswa)
<pre>Private Sub Command1_Click() Dim no As Integer, i As Integer no = CInt(Combo1.Text) NamaSiswa(no)= InputBox("Tuliskan nama siswa no : " & no, "Input Nama Siswa") If NamaSiswa(no) <> "" Then</pre>	Ketika user meng-klik Command1: Tampilkan kotak input untuk mengisi nilai data elemen array dengan nomer indeks yang dipilih. Jika kotak input tidak kosong

Kode Program	Deskripsi
<pre>List1.Clear For i = 0 To 4 List1.AddItem "NamaSiswa(" & i & ")=" & NamaSiswa(i) Next i End If End Sub</pre>	<p>maka: Hapus isi List1.</p> <p>Tampilkan nilai data setiap elemen array NamaSiswa.</p> <p>Catatan : Jika tombol Cancel pada kotak input di-klik sama artinya variabel ataupun array yang menyimpan hasil input bernilai kosong ("").</p>

Modifikasi kode programnya menjadi sebagai berikut :

```
Dim NamaSiswa() As String

Private Sub Form_Load()
    Dim i As Integer
    ReDim NamaSiswa(1 To 5)
    For i = 1 To 5
        Comb1.AddItem i
    Next i
    Comb1.ListIndex = 0
End Sub

Private Sub Command1_Click()
    Dim no As Integer, i As Integer
    no = CInt(Comb1.Text)
    NamaSiswa(no) = InputBox("Tuliskan nama siswa no : " &
no, "Input Nama Siswa")
    If NamaSiswa(no) <> "" Then
        List1.Clear
        For i = 1 To UBound(NamaSiswa)
            List1.AddItem "NamaSiswa(" & i & ")=" &
NamaSiswa(i)
        Next i
    End If
End Sub

Private Sub Command2_Click()
    Dim num As Integer, i As Integer
    If Not IsNumeric(Text1.Text) Then Exit Sub
    num = CInt(Text1.Text)
    ReDim NamaSiswa(1 To num)
    Comb1.Clear
    List1.Clear
    For i = 1 To UBound(NamaSiswa)
        Comb1.AddItem i
        List1.AddItem "NamaSiswa(" & i & ")=" &
NamaSiswa(i)
    Next i
    Comb1.ListIndex = 0
End Sub
```

Coba jalankan Project1 :

- Pilih nomer indeks elemen array NamaSiswa melalui Combo1 (antara 1 s/d 5)
- Klik Command1 → akan muncul kotak input untuk mengisi nama siswa
- Ketikkan nilai data untuk elemen array dengan nomer indeks yang dipilih, misalnya Salman
- Klik tombol OK
- Isi setiap elemen array NamaSiswa akan muncul di dalam List1
- Ketikkan jumlah elemen array yang diinginkan pada Text2, lalu klik Command2
- Jumlah elemen array NamaSiswa akan berubah

Catatan :

- Setelah perintah Redim maka jumlah array akan diubah, tetapi mengakibatkan isi setiap nilai data di dalam array tersebut dikosongkan.
- Agar perintah Redim tidak mengosongkan isi setiap nilai data di dalam array-nya, gunakan kata Preserve setelah perintah Redim-nya, contoh :

```
ReDim Preserve NamaSiswa(1 To num)
```

9.2 Control ARRAY

Kontrol array merupakan sekumpulan kontrol yang “dikelompokkan” dengan nama yang sama di dalam sebuah Form. Kontrol array digunakan bila ada beberapa kontrol yang sama dan akan mendapat perlakuan yang sama pula. Misalnya, ada 5 buah TextBox di dalam sebuah Form dan akan diprogram dengan cara yang sama, maka akan lebih mudah jika membuat sebuah TextBox sebagai kontrol array dibandingkan bila membuat 5 buah TextBox yang berbeda. Setiap object di dalam kontrol array masing-masing dibedakan dengan nomer indeksnya.

Untuk membuat kontrol array (misalnya TextBox) sebanyak 5 buah di dalam sebuah form, lakukan langkah berikut ini:

1. Buatlah sebuah TextBox di dalam sebuah Form.
2. Aturlah property TextBox tersebut sebagai berikut :

Name : txtData

Index : 0 (nol)

Perhatikan : kontrol TextBox-nya akan menjadi object **txtData(0)** (lihat bagian Object Selector pada Jendela Properties).

3. Klik object txtData(0) pada Form, kemudian klik tombol Copy pada bagian Toolbar.
4. Untuk membuat TextBox kedua, klik tombol Paste pada bagian Toolbar. TextBox kedua akan muncul di pojok kiri Form sebagai object txtData(1), aturlah posisinya di dalam Form.

5. Lakukan langkah ke-4 di atas sebanyak 4 kali (sesuai dengan jumlah TextBox yang dibutuhkan).
6. Di dalam Form akan ada 5 buah TextBox dengan nama yang sama (yaitu txtData) dan masing-masing menjadi object txtData yang dibedakan nomer indexnya (mulai dari 0 s/d 4).

Selanjutnya object-object yang dibuat dengan kontrol array bisa diprogram dengan lebih mudah. Misalnya untuk “mengosongkan” object txtData, bisa menggunakan struktur kontrol For...Next :

```
For i = 0 To 4
    txtData(i).Text = ""
Next i
```

Cara ini lebih mudah bila dibandingkan cara “konvensional” berikut :

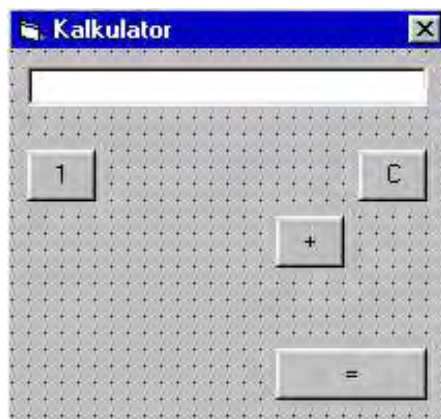
```
Text1.Text = ""
Text2.Text = ""
...
Text4.Text = ""
```

Contoh Program: Kalkulator

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut:

Object	Properties	Value
Form9	BorderStyle Caption StartPosition	1 – Fixed Single Kalkulator 2 – Center Screen
Text1	Alignment Text	1 – Right Justify <kosong>

Object	Properties	Value
Command1	Name Caption Index	cmdAngka 1 0
Command2	Name Caption	cmdHapus C
Command3	Name Caption Index	cmdOperator + 0
Command4	Name Caption	cmdHitung =

Buat kontrol array dari cmdAngka (sebanyak 10 buah) dan dari cmdOperator (sebanyak 3 buah). Kemudian atur property caption masing-masing sebagai berikut :

Object	Caption
cmdAngka(1)	2
cmdAngka(2)	3
...	...
cmdAngka(10)	,
cmdOperator(1)	-
cmdOperator(2)	*
cmdOperator(3)	/



Buka Jendela Code dan pada bagian Code Editor, lalu ketikkan kode programnya sebagai berikut:

```
Dim angka(1 To 2) As Single
Dim operator As String
```

```

Private Sub cmdAngka_Click(Index As Integer)
    Text1.Text = Text1.Text & cmdAngka(Index).Caption
End Sub

Private Sub cmdOperator_Click(Index As Integer)
    If Text1.Text = "" Then Exit Sub
    angka(1) = CSng(Text1.Text)
    operator = cmdOperator(Index).Caption
    Text1.Text = ""
End Sub

Private Sub cmdHitung_Click()
    Dim hasil As Single
    If Text1.Text = "" Then Exit Sub
    angka(2) = CSng(Text1.Text)
    Select Case operator
        Case "+"
            hasil = angka(1) + angka(2)
        Case "-"
            hasil = angka(1) - angka(2)
        Case "*"
            hasil = angka(1) * angka(2)
        Case "/"
            hasil = angka(1) / angka(2)
    End Select
    Text1.Text = hasil
End Sub

Private Sub cmdHapus_Click()
    Text1.Text = ""
End Sub

```

Simpan Form9 (nama file : **Lat9.frm**).

Catatan:

Anda bisa men-download file Lat9.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form9.

Coba jalankan Project1 :

- Klik salah satu tombol angka, misalnya 5
- Klik salah satu tombol operator, misalnya *
- Klik lagi salah satu tombol angka, misalnya 6
- Klik tombol =
- Akan muncul pada TextBox angka 30 (hasil penghitungan 5 * 6)
- Klik tombol C untuk menghapus isi TextBox

Penjelasan kode program :

Kode Program	Deskripsi
Private Sub cmdHitung_Click()	Bila user meng-klik cmdHitung :

Kode Program	Deskripsi
Dim hasil As Single	Deklarasi variabel lokal
If Text1.Text = "" Then Exit Sub	Cek apakah Text1 kosong, bila True maka proses tidak perlu dilanjutkan
angka(2) = CSng(Text1.Text)	Konversi isi Text1 menjadi Single, lalu simpan ke dalam variabel angka(2) \leftarrow variabel array
Select Case operator	Lakukan penghitungan sesuai dengan isi variabel operator dan hasil penghitungan disimpan dalam variabel hasil :
Case "+" hasil = angka(1) + angka(2)	Operasi penjumlahan
Case "-" hasil = angka(1) - angka(2)	Operasi pengurangan
Case "*" hasil = angka(1) * angka(2)	Operasi perkalian
Case "/" hasil = angka(1) / angka(2)	Operasi pembagian
End Select	
Text1.Text = hasil	
End Sub	
Private Sub cmdHapus_Click() Text1.Text = ""	Tampilkan hasil penghitungan pada Text1.
End Sub	Bila user meng-klik cmdHapus : Kosongkan Text1

Catatan:

Program kalkulator ini masih sangat sederhana, sehingga kemungkinan ada *error* ataupun “kejanggalan” lainnya.



BAB 10 PENANGANAN ERROR

Tujuan Pembelajaran Umum

1. Memahami bagaimana penanganan error yang terjadi pada suatu program

Tujuan Pembelajaran Khusus

1. Dapat mengetahui jenis-jenis error pada suatu program dan melakukan penelusuran error-nya.
2. Dapat membuat baris program untuk menangani error.

10.1 Jenis-Jenis Error

Dalam proses pembuatan program, bisa saja terjadi error yang menyebabkan program tidak berjalan sebagaimana mestinya. Dilihat dari penyebabnya ada 3 jenis error yang bisa terjadi, yaitu:

1. **Syntax error** – adalah error yang disebabkan oleh kesalahan menulis kode program. Misalnya : salah menuliskan nama object, property atau methodnya. Error jenis ini relatif mudah ditangani, IDE VB 6 akan memberi tanda kode program mana yang menimbulkan syntax error.

```
If FileName <> "" Then  
    Image1.Picture = LoadPicture(FileName  
End If  
Exit Sub
```

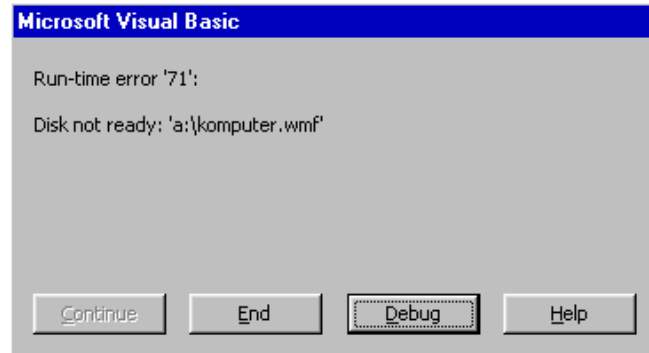


Awal:

```
Image1.Pictur = Nothing  
FileName = InputBox("Tuliskan nama file gambar :")
```



2. **Runtime error** – adalah error yang disebabkan oleh sistem komputer ketika melakukan sesuatu. Misalnya : menyimpan file ke disket tetapi disketnya tidak ada. Sistem akan “memberitahu” kepada program informasi error yang terjadi. Informasi error yang penting diantaranya adalah nomer error dan deskripsi error. VB 6 “menyimpan” informasi error tersebut pada object Err. Melalui object Err inilah kita bisa menangani runtime error.



```

If FileName <> "" Then
    Image1.Picture = LoadPicture(FileName)
End If
Exit Sub

```

←

3. **Logical error** – adalah error yang disebabkan oleh kesalahan logika pemrograman (dari si programmer). Misalnya : salah meletakkan urutan kode program. Error jenis ini relatif sulit diketahui dan bisa saja baru diketahui setelah program di-*compile* menjadi *executable file* (*.exe). Kejadian seperti ini sering disebut sebagai *bug*.

Pada bab ini akan dibahas khusus bagaimana cara menangani runtime error.

10.2 Menangani Runtime Error

Untuk menangani runtime error, di dalam kode program dibuatkan bagian khusus untuk menangani error yang terjadi. Bagian khusus ini diberi tanda berupa <label>. Agar setiap error yang terjadi dapat ditangani oleh bagian khusus ini, pada awal kode program ditulis perintah berikut ini:

```
On Error GoTo <label>
```

Contoh :

```

On Error GoTo AdaError
...
...
...
AdaError:
...
...
...

```

} kode-kode program yang mungkin menimbulkan runtime error

} kode-kode program untuk menangani runtime error

Selanjutnya untuk mengetahui informasi error apa yang terjadi, gunakan object Err. Object Err merupakan object *built-in* yang disediakan VB 6 untuk

menangani error. Seperti halnya object yang dibuat dengan kontrol/komponen (misalnya TextBox), maka object Err juga mempunyai property dan method.

Property object Err yang penting diantaranya adalah property Number dan Description. Untuk mengetahui nomer error yang terjadi, gunakan property Number. Sedangkan untuk mengetahui deskripsi error yang terjadi, gunakan property Description.

Dengan menggunakan property Number, kita bisa menangani setiap error dengan cara yang berbeda. Misalnya : untuk setiap error yang terjadi akan dimunculkan kotak pesan dengan bahasa Indonesia. Cara ini dapat Anda lihat pada contoh program.

Setelah error ditangani, kode program dapat dikembalikan ke bagian tertentu dengan perintah Resume. Ada 3 bentuk perintah Resume, yaitu:

- **Resume** – kembali ke kode program yang menimbulkan error. Biasanya digunakan untuk *retry* (mencoba lagi).
- **Resume <label>** – kembali ke bagian tertentu pada kode program yang diberi tanda <label>.
- **Resume Next** – kembali ke kode program berikutnya (kode program setelah kode program yang menimbulkan error). Biasanya digunakan untuk *abort* atau *cancel* (mengabaikan atau membatalkan).

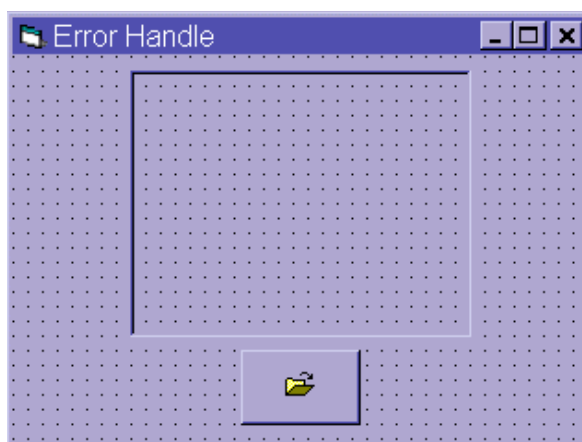
Contoh penggunaan perintah Resume dapat Anda lihat pada contoh program.

Contoh Program : Error Handle

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen Image dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut:

Object	Properties	Value
Form10	Caption StartUpPosition	Error Handle 2 – Center Screen
Image1	BorderStyle Stretch	1 – Fixed single True
Command1	Caption Style Picture ToolTipText	<kosong> 1 – Graphical OpenFile.ico Open Picture File

Buka Jendela Code dan pada bagian Code Editor, lalu ketikkan kode programnya sebagai berikut:

```
Private Sub Command1_Click()  
    Dim FileName As String  
  
    Image1.Picture = Nothing  
    FileName = InputBox("Ketikkan path dan nama file  
gambar:", "Open Picture File", FileName)  
    If FileName <> "" Then  
        Image1.Picture = LoadPicture(FileName)  
    End If  
End Sub
```

Simpan Form10 (nama file : **Lat10.frm**).

Catatan :

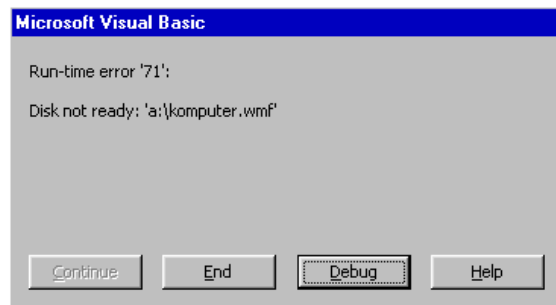
Anda bisa men-download file Lat10.frm dan OpenFile.ico.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form10.

Untuk mencoba program ini sebelumnya siapkan sebuah disket dan copy-kan 1 file gambar (misalnya file komputer.wmf pada bab 6).

Coba jalankan Project-nya:

- Klik tombol Open Picture File, akan muncul kotak input.
- Ketikkan lokasi (path) dan nama file gambar, misalnya : a:\komputer.wmf.
- Masukkan disketnya, lalu klik tombol OK.
- Gambar akan muncul ☐ tidak terjadi error.
- Keluarkan disketnya, lalu coba ulangi langkah sebelumnya.
- Akan muncul kotak pesan error :



Klik tombol Debug untuk melihat kode program mana yang menimbulkan runtime error.

- Klik tombol End pada bagian Toolbar untuk menghentikan program.

Agar program bisa menangani error, modifikasi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()
    Dim FileName As String

    On Error GoTo AdaError

    Image1.Picture = Nothing
    FileName = InputBox("Ketikkan path dan nama file gambar :", "Open Picture File", FileName)

    If FileName <> "" Then
        Image1.Picture = LoadPicture(FileName)
    End If
    Exit Sub

AdaError:
    MsgBox "Error No : " & Err.Number & vbCrLf & _
        Err.Description, vbCritical + vbOKOnly, Me.Caption
End Sub
```

Coba jalankan Project-nya

Bila terjadi error seperti sebelumnya akan muncul kotak pesan error:



Klik tombol OK.

Program akan tetap berjalan.

User bisa mengulangi perintah untuk membuka file gambar (disket dimasukkan lebih dulu agar tidak terjadi error lagi).

Agar pesan error-nya menggunakan bahasa Indonesia, modifikasi kode programnya menjadi sebagai berikut :

```

Private Sub Command1_Click()
    Dim FileName As String
    Dim ErrMsg As String

    On Error GoTo AdaError

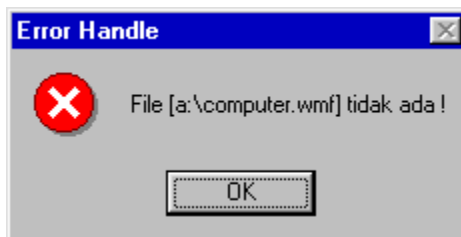
    Image1.Picture = Nothing
    FileName = InputBox("Ketikkan path dan nama file gambar
:", "Open Picture File", FileName)
    If FileName <> "" Then
        Image1.Picture = LoadPicture(FileName)
    End If
    Exit Sub

AdaError:
    Select Case Err.Number
        Case 53
            ErrMsg = "File [" & FileName & "] tidak ada !"
        Case 71
            ErrMsg = "Disket belum dimasukkan !"
        Case Else
            ErrMsg = Err.Description
        End Select
    MsgBox ErrMsg, vbCritical + vbOKOnly, Me.Caption
End Sub

```

Coba jalankan Project-nya

Bila terjadi error akan muncul kotak pesan error:



Klik tombol OK.

Program akan tetap berjalan.

Lakukan *try-and-error* untuk melihat setiap nomer dan deskripsi error yang mungkin terjadi.

Agar setelah error ditangani kode program dapat dikembalikan ke bagian tertentu, modifikasi lagi kode programnya menjadi sebagai berikut :

```

Private Sub Command1_Click()
    Dim FileName As String
    Dim ErrMsg As String
    Dim Ask As VbMsgBoxResult

    On Error GoTo AdaError

Awal:
    Image1.Picture = Nothing
    FileName = InputBox("Ketikkan path dan nama file gambar
:", "Open Picture File", FileName)
    If FileName <> "" Then
        Image1.Picture = LoadPicture(FileName)
    End If
    Exit Sub

AdaError:
    Select Case Err.Number
        Case 53
            ErrMsg = "File [" & FileName & "] tidak ada !"
        Case 71
            ErrMsg = "Disket belum dimasukkan !"
        Case Else
            ErrMsg = Err.Description
        End Select
    Ask = MsgBox ErrMsg, vbCritical + vbOKOnly, Me.Caption
    If Ask = vbOK Then
        GoTo Awal
    End If
End Sub

```

```

        End If
    Exit Sub
AdaError:
    Select Case Err.Number
        Case 53
            ErrMsg = "File [" & FileName & "] tidak ada !"
        Case 71
            ErrMsg = "Disket belum dimasukkan !"
        Case Else
            ErrMsg = Err.Description
        End Select

    Ask = MsgBox(ErrMsg, vbCritical + vbRetryCancel, Me.Caption)

    Select Case Ask
        Case vbRetry
            If Err.Number = 53 Then Resume Awal Else Resume
        Case vbCancel
            Resume Next
        End Select
    End Sub

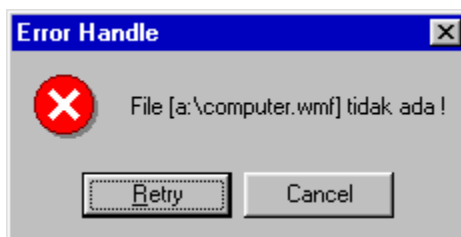
```

Coba jalankan Project-nya

Bila terjadi error akan muncul kotak pesan error:



Klik tombol Retry untuk mencoba lagi (disketnya dimasukkan lebih dulu), atau klik tombol Cancel untuk membatalkan.



Klik tombol Retry untuk mencoba lagi (lalu ganti nama filenya), atau klik tombol Cancel untuk membatalkan.



BAB 11 PENGUNAAN PROCEDURE

Tujuan Pembelajaran Umum

1. Memahami fungsi sub-program dalam pembuatan program baik dalam bentuk *procedure* maupun *function*.

Tujuan Pembelajaran Khusus

1. Mahasiswa dapat membuat dan menggunakan sub-program dalam bentuk *procedure*
2. Mahasiswa dapat membuat dan menggunakan sub-program dalam bentuk *function*

11.1 Apa itu Procedure ?

Procedure adalah blok kode program yang berisi perintah-perintah untuk mengerjakan tugas tertentu. Bila di dalam kode program yang kita buat ada perintah-perintah untuk melakukan tugas yang sama di beberapa tempat, maka akan lebih baik perintah-perintah tersebut dibuat dalam sebuah procedure. Kemudian, procedure itu bisa di-‘panggil’ bila diperlukan.

Penggunaan procedure sangat menghemat penulisan kode program, karena kode-kode program yang sama di beberapa tempat cukup dibuat pada satu bagian saja. Selain itu, procedure akan memudahkan perbaikan kode program bila terjadi perubahan atau kesalahan, karena perbaikan cukup dilakukan pada satu bagian saja.

11.2 Jenis-jenis Procedure

Pada Visual Basic 6.0 ada 4 jenis procedure, yaitu :

- **Procedure Sub** – procedure yang tidak mengembalikan nilai setelah ‘tugas’-nya selesai.
- **Procedure Function** – procedure yang mengembalikan nilai setelah ‘tugas’-nya selesai.
- **Procedure Event** – procedure untuk suatu event pada sebuah object. Digunakan di dalam class module.
- **Procedure Property** – procedure untuk mengubah (*let*) atau mengambil (*get*) nilai property pada sebuah object. Digunakan di dalam class module.

Procedure event dan property akan dibahas pada bab selanjutnya.

Bentuk penulisan (*syntax*) procedure sub :

```

[Public | Private] Sub <nama_sub> ([<argumen>])
...
<isi procedure>
...
End Sub

```

Sedangkan bentuk penulisan (*syntax*) procedure function :

```

[Public | Private] Function <nama_function> ([<argumen>])
As <tipe_data>
...
<isi procedure>
...
End Function

```

Pernyataan [Public | Private] menentukan ruang lingkup (*scope*) procedure. Sebuah procedure dengan scope public bisa digunakan dalam lingkup project. Sedangkan procedure dengan scope private hanya bisa digunakan dalam lingkup form saja.

<nama_sub> atau <nama_function> dibuat sebagai pengenalan procedure saat di-‘panggil’. Aturan penamaan sebuah procedure sama dengan aturan penamaan sebuah variabel. Nama sebuah procedure dibuat unik, tidak boleh ada yang sama.

<argumen> merupakan serangkaian nilai dan tipe data yang dipakai oleh procedure untuk mengerjakan ‘tugas’-nya. Sebuah procedure bisa saja tidak memakai argumen sama sekali.

Pernyataan As <tipe_data> pada procedure function menentukan tipe data nilai yang akan dikembalikan (*return value*) setelah ‘tugas’-nya selesai.

Untuk menggunakan sebuah procedure, maka procedure tersebut harus di-‘panggil’ pada bagian tertentu dari kode program. Procedure sub di-‘panggil’ dengan pernyataan :

```
Call <nama_sub>([<argumen>])
```

Sedangkan procedure function bisa di-‘panggil’ langsung dengan menyisipkannya di dalam kode program yang memanggilnya.

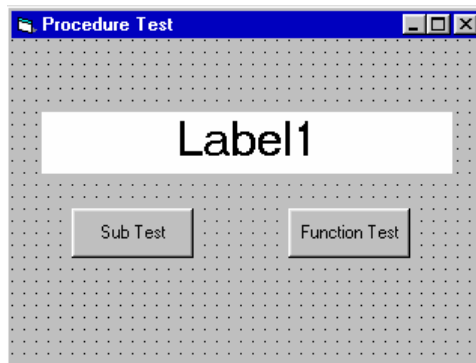
Procedure boleh ditulis dimana saja dalam kode program, tetapi biasanya ditulis dibagian atas atau bawah agar mudah ditangani.

Contoh Program : Procedure Test

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini: (Gunakan komponen Label dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form1	Caption StartUpPosition	Procedure Test 2 – Center Screen
Label1	Alignment BackColor Font ToolTipText	2 – Center Palette : <putih> Size : 24 Dobel-Klik di Sini
Command1	Caption ToolTipText	Sub Test Klik di sini!
Command2	Caption	Function Test

Buka Jendela Code, lalu pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```

Private Sub TulisTeks(teks As String, warna As
ColorConstants)
    With Label1
        .Caption = teks
        .ForeColor = warna
    End With
End Sub

Private Function JumlahAngka() As String
    Dim angka1 As String, angka2 As String
    Dim hasil As Single
    angka1 = InputBox("Tulis angka 1 :", "Jumlah Angka")
    angka2 = InputBox("Tulis angka 2 :", "Jumlah Angka")
    If angka1 <> "" And angka2 <> "" Then
        hasil = CSng(angka1) + CSng(angka2)
        JumlahAngka = Cstr(hasil)
    End If
End Function

Private Sub Label1_DblClick()
    Call TulisTeks("Hai", vbBlue)
End Sub

```

```

Private Sub Command1_Click()
    Call TulisTeks("Hallo", vbRed)
End Sub

Private Sub Command2_Click()
    Label1.Caption = "Jumlah = " & JumlahAngka()
End Sub

```

Simpan Form11 (nama file : **Lat11.frm**).

Catatan :

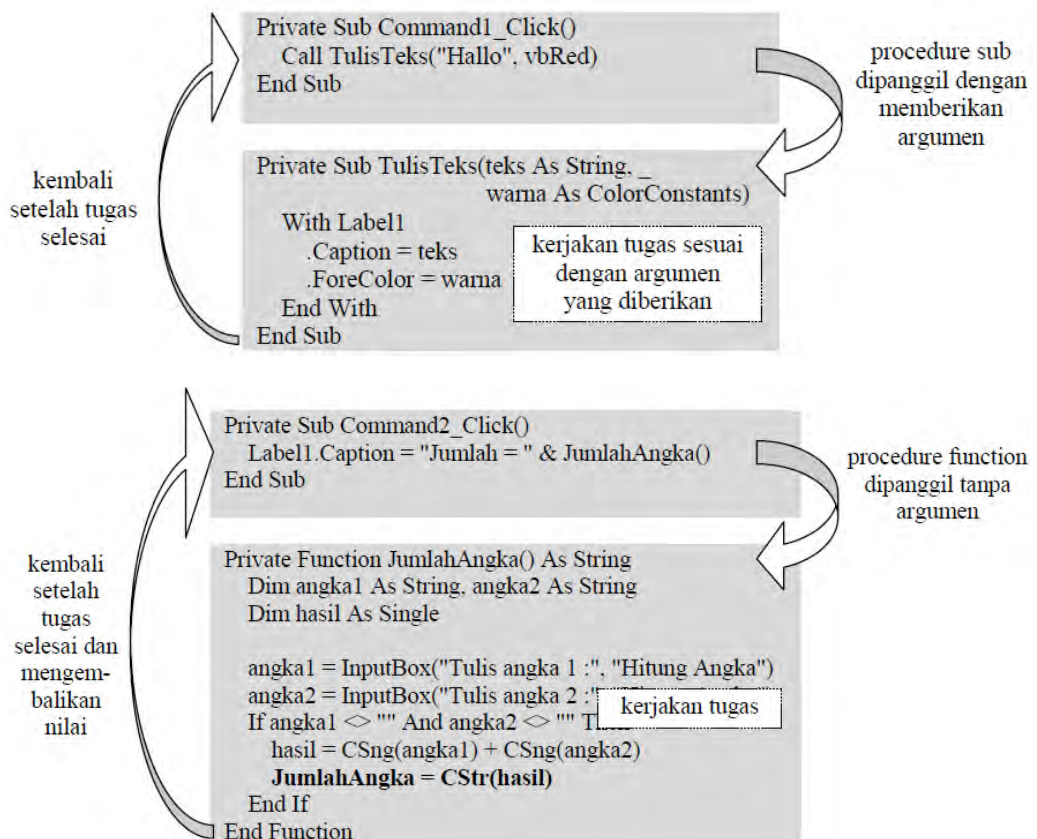
Anda bisa men-download file Lat1-11.zip.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form11.

Coba jalankan Project-nya :

- Dobel-klik Label1, akan muncul tulisan “Hai” dengan warna biru.
- Klik Command1, akan muncul tulisan “Hallo” dengan warna merah.
- Klik Command2, akan muncul kotak input.
- Ketikkan angka 5, lalu klik tombol OK.
- Ketikkan angka 10, lalu klik tombol OK.
- Akan muncul tulisan “Jumlah = 15”.

Proses ketika procedure di-‘panggil’ :



Dapat anda lihat bahwa prosedur `TulisTeks` dipanggil, dan dimasukkan 2 variabel ke dalamnya yaitu `teks` dan `warna`. Selanjutnya kedua argumen tersebut diolah oleh prosedur `TulisTeks`. Akhirnya ditampilkan tulisan dan warnanya yang merupakan hasil dari properti `Caption` dan `ForeColor` dari obyek label.

11.3 Pernyataan Optional

Nilai argumen di dalam sebuah procedure dapat dibuat menjadi optional (boleh ada atau tidak ada). Bila nilai argumen yang optional tidak ada, maka yang digunakan adalah nilai default-nya. Untuk membuat nilai argumen menjadi optional, tambahkan pernyataan **Optional** di depan argumen-nya dan berikan nilai default-nya, contoh:

```
Private Sub TulisTeks(teks As String, Optional warna As  
ColorConstants = vbMagenta)  
    With Label1  
        .Caption = teks  
        .ForeColor = warna  
    End With  
End Sub
```

Bila nilai argumen `warna` tidak ada, maka warna teks pada `Label1` adalah magenta, contoh :

```
Call TulisTeks("Hallo")
```

11.4 Pernyataan ByRef dan ByVal

Nilai argumen yang dikirimkan ke dalam procedure bisa saja berasal dari sebuah variabel. Namun demikian, perlu diperhatikan bahwa nilai argumen yang dikirimkan ke dalam procedure bisa berdasarkan referensi (`ByRef`) atau berdasarkan nilai (`ByVal`). Default-nya nilai argumen dikirimkan berdasarkan referensi. Apabila variabel dikirimkan berdasarkan referensi, maka semua perubahan yang dilakukan terhadap argumen akan mempengaruhi variabel tersebut. Hal ini terkadang menimbulkan *logical error*. Sebagai ilustrasi perhatikan contoh berikut ini:

```
Dim Harga As Currency, Total As Currency  
  
Private Sub HitungDiskon(HargaAwal As Currency, Diskon As  
Single)  
    HargaAwal = HargaAwal * (1 - Diskon)  
    Total = CCur(HargaAwal)  
End Sub  
  
Private Sub Command3_Click()  
    Harga = 10000  
    Total = 0  
    Call HitungDiskon(Harga, 0.1)  
    MsgBox "Harga " & Harga & " setelah diskon 10% menjadi  
" & Total  
End Sub
```

Hasilnya → Harga 9000 setelah diskon 10% menjadi 9000

Variabel Harga dikirimkan berdasarkan referensi (ByRef), sehingga perubahan pada argumen HargaAwal akan mempengaruhi variabel Harga.

Modifikasi procedure sub HitungDiskon menjadi sebagai berikut :

```
Private Sub HitungDiskon(ByVal HargaAwal As Currency,  
Diskon As Single)  
    HargaAwal = HargaAwal * (1 - Diskon)  
    Total = CCur(HargaAwal)  
End Sub
```

Hasilnya → Harga 10000 setelah diskon 10% menjadi 9000

Sekarang variabel Harga dikirimkan berdasarkan nilai (ByVal), sehingga perubahan pada argumen HargaAwal tidak akan mempengaruhi variabel Harga.

11.5 Procedure Built-in

VB6 menyediakan cukup banyak procedure yang siap-pakai (*built-in*) untuk berbagai ‘tugas’. Coba klik menu View – Object Browser atau tekan tombol F2 pada keyboard :



Gambar 11.1 Object browser untuk mencari built in procedure (prosedur yang disediakan oleh Visual Basic)

Untuk mencari Built in procedures, dapat melakukan beberapa hal berikut sesuai yang ditunjukkan pada :

- Cari module Strings pada bagian Classes.
- Lihat daftar pada bagian Member of ‘Strings’ yang merupakan kumpulan procedure dengan ‘tugas’ pengolahan data string.
- Klik nama procedure untuk mengetahui apa ‘tugas’-nya.



BAB 12 PEMROGRAMAN GRAFIK

Tujuan Pembelajaran Umum

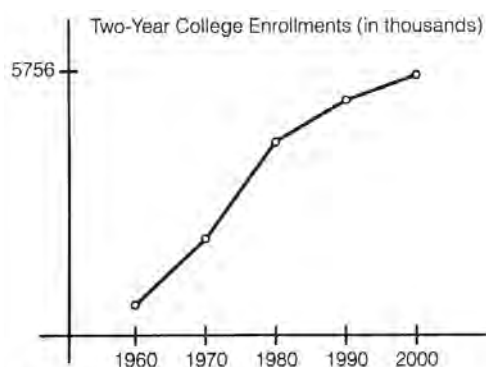
1. Mengetahui cara pembuatan program grafik

Tujuan Pembelajaran Khusus

1. Dapat membuat program grafik sederhana
2. Dapat membuat program grafik lebih rinci dengan memanfaatkan metoda-metoda yang ada.

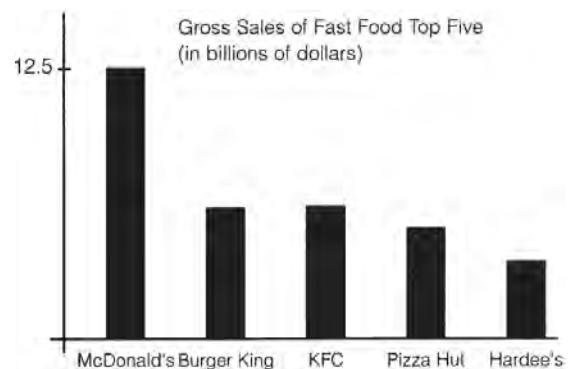
12.1 Dasar-Dasar Grafik

Visual Basic mempunyai kapasitas pembuatan diagram yang cukup baik. Gambar 12.1 menunjukkan empat jenis diagram yang dapat ditampilkan di picture box dan dicetak dengan printer.

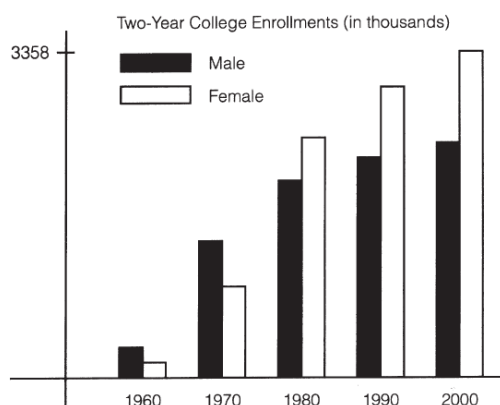


Source: Statistical Abstract of the United States

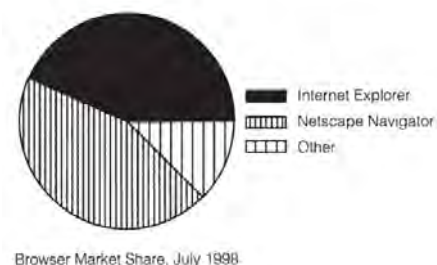
Line Chart



Bar Chart



Clustered Bar Chart



Pie Chart

Gambar 12.1 Empat jenis diagram yang bisa dibuat dengan menggunakan Visual Basic.

Dasar untuk membuat diagram seperti pada Gambar 12.1 adalah dengan melakukan 3 (tiga) langkah berikut:

1. Mendefinisikan sistem koordinat.
2. Menggunakan metoda grafik untuk menggambar bentuk-bentuk yang sesuai seperti garis, segi empat dan lingkaran.
3. Memberikan teks pada tempat yang sesuai di diagram.

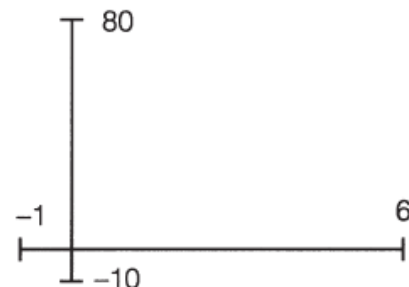
12.1.1 Mendefinisikan Sistem Koordinat

Bayangkan seperti akan membuat gambar grafik di atas kertas. Anggap kita mempunyai selembar kertas, sebuah pensil dan penggaris dan akan membuat garis dari koordinat (2, 40) ke (5, 60). Kita harus mengikuti tiga langkah berikut untuk membuatnya.

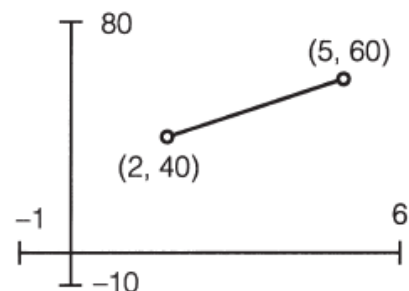
1. Membuat garis sumbu x dan y dengan menggunakan penggaris. Garis hanya fokus di quadran pertama, karena titik-titik garis tersebut berada di quadran tersebut.



2. Memilih skala yang cocok untuk kedua sumbu tersebut. Sebagai contoh, kita memberikan untuk sumbu x bilangan antara -1 sampai dengan 6 dan sumbu y bilangan antara -10 sampai dengan 80.



3. Gambarkan titik-titik dari garis tersebut dalam ruang koordinat dengan posisi sesuai skala dan hubungkan titik-titik tersebut untuk membuat garis lurus.

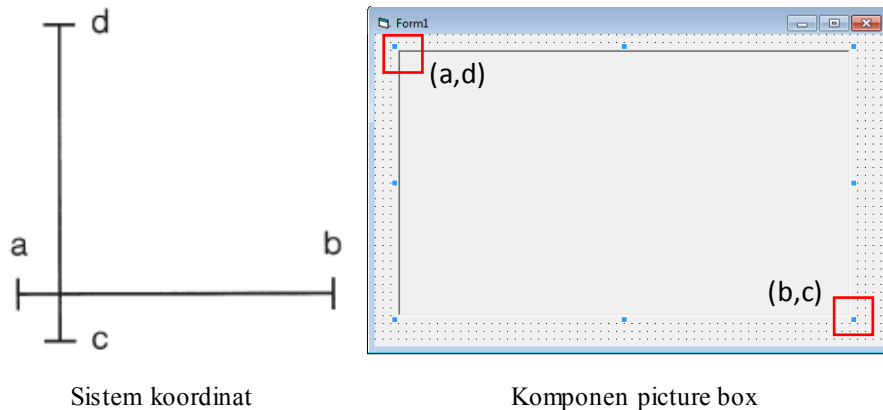


Membuat gambar di layar komputer sama halnya dengan membuat gambar di kertas dengan pensil dan penggaris seperti yang telah dijelaskan sebelumnya. Bedanya, langkah (2) dilakukan terlebih dahulu setelah itu baru langkah (1) dan kemudian langkah (3). Di Visual Basic metoda scale digunakan untuk mendefinisikan jangkauan daerah gambar yang diwakili oleh picture box.

Pernyataan di kode programnya adalah sebagai berikut:

```
picBox.Scale (a, d)-(b, c)
```

perintah tersebut menunjukkan bilangan untuk sumbu x dari a hingga b dan sumbu y dari c hingga d. Lihat Gambar 12.2. Bagian (a,d) menunjukkan koordinat ujung kiri-atas dari picture box dan bagian (b,c) menunjukkan koordinat kanan-bawah dari picture box.



Gambar 12.2 Posisi pendefinisian sistem koordinat di VB.

12.1.2 Metoda Grafik untuk menggambar garis, titik dan lingkaran

Setelah sistem koordinat ditentukan dengan perintah `picBox.Scale`, gambar dapat dibuat di dalam picture box dengan perintah `.Pset`, `.Line` dan `.Circle`. Pernyataan dalam kode programnya adalah sebagai berikut:

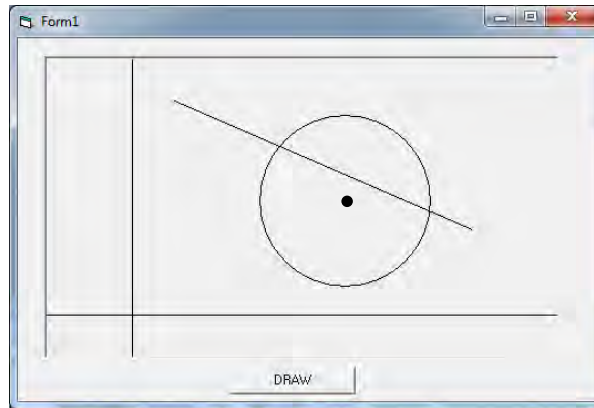
```
picBox.Line (x1, y1)-(x2, y2)
```

```
picBox.PSet (x, y)
```

```
picBox.Circle (x, y), r
```

Contoh baris program berikut akan menghasilkan bentuk seperti pada Gambar 12.3.

```
Private Sub cmdDraw_Click()  
    picOutput.Cls  
    picOutput.Scale (-2,18)-(10,-3) 'koordinat  
    picOutput.Line (-2,0)-(10,0)    'sumbu-x  
    picOutput.Line (0,-3)-(0,18)    'sumbu-y  
    picOutput.PSet (5,8)             'gambar titik  
    picOutput.Line (1,15)-(8,6)     'gambar garis  
    picOutput.Circle (1,15),5        'gambar Lingkaran  
End Sub
```

Gambar 12.3 Hasil gambar berupa titik, garis dan lingkaran

Jenis garis dapat diatur sesuai dengan gaya yang diinginkan. Untuk mengaturnya dengan mengatur property dari pictureBox yaitu `.DrawStyle`. Jenis-jenis garis sesuai dengan nilai property `DrawStyle` seperti ditunjukkan pada Gambar 12.4. Pernyataannya dapat dituliskan sebagai berikut:

```
pictureBox.DrawStyle = s
pictureBox.Line (a, b)-(c, d)
```

Draw Style #	Line Pattern
0	—————
1	- - - - -
2	- . - . -
3	- . - . -
4	- . - . -

Gambar 12.4 Jenis-jenis garis yang diatur melalui metoda `.DrawStyle`

Contoh Program Grafik Persamaan Akar Kuadrat

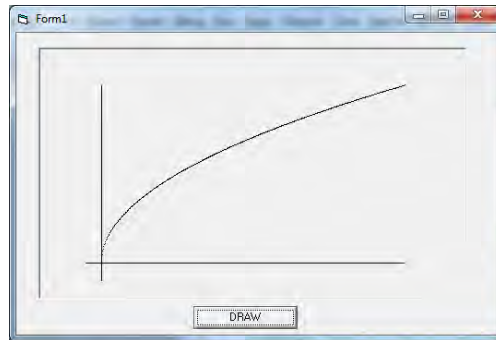
Membuat grafik persamaan akar kuadrat $y = \sqrt{x}$ dengan nilai x dari 0 – 100.

Buat antarmuka program sederhana seperti terlihat pada Gambar 12.3 dan memberikan kode program pada command button, `cmdDraw` sebagai berikut:

```
Private Sub cmdDraw_Click()
    Dim x As Single

    'Graph the Square Root Function
    picOutput.Cls
    picOutput.Scale (-20, 12)-(120, -2) 'koordinat
    picOutput.Line (-5, 0)-(100, 0) 'sumbu-x
    picOutput.Line (0, -1)-(0, 10) 'sumbu-y
    For x = 0 To 100 Step 0.2 '500 titik x
        picOutput.PSet (x, Sqr(x)) 'Plot titik
    Next x
End Sub
```

Jalankan program tersebut, maka akan menghasilkan seperti pada Gambar 12.5 berikut.



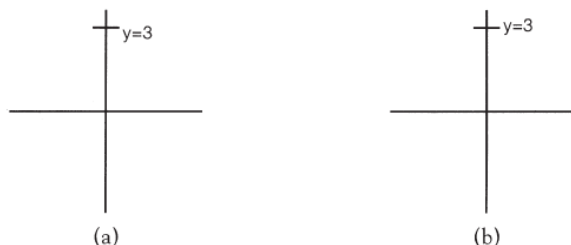
Gambar 12.5 Hasil running program grafik persamaan akar kuadrat $y = \sqrt{x}$.

12.1.3 Memosisikan TEXT

Dalam membuat grafik tanpa ada keterangan berupa teks membuat grafik tidak terlihat baik, sehingga diperlukan label untuk membuat teks tersebut. Masalahnya adalah pemosisian teks tersebut haruslah sesuai. Di dalam picture box terdapat property yang mengatur posisi yaitu `.CurrentX` dan `.CurrentY` dan berikutnya adalah mengatur tinggi dan lebar huruf dengan `.TextHeight` dan `.TextWidth`. Property `CurrentX` dan `CurrentY` mencatat posisi horozontal dan vertikal untuk teks yang berikutnya akan di cetak.

Contoh kode berikut akan menghasilkan grafik seperti pada Gambar 12.6(a).

```
Private Sub cmdDraw_Click()  
    picOutput.Cls  
    picOutput.Scale (-4, 4)-(4, -4)  
    picOutput.Line (-4, 0)-(4, 0) 'Sumbu-x  
    picOutput.Line (0, -4)-(0, 4) 'Sumbu-y  
    picOutput.Line (-.3, 3)-(.3, 3) 'tanda sumbu  
    picOutput.CurrentX = .3 'sisi kanan tanda sumbu  
    picOutput.CurrentY = 3 'sama dengan tanda sumbu  
    picOutput.Print "y=3" 'Label untuk tanda sumbu  
End Sub
```



Gambar 12.6 Hasil pencetakan teks pada grafik (a) kasus pertama (b) kasus kedua

Jika Pada baris dengan metoda `.CurrentY` diubah dengan kode seperti dibawah ini, maka akan menghasilkan label tanda sumbu seperti pada Gambar 12.6(b).

```
picOutput.CurrentY = 3 - picOutput.TextHeight("y=3") / 2
```

12.2 Diagram GARIS (*LINE Charts*)

Diagram garis menampilkan perubahan suatu nilai tertentu yang berhubungan dengan perubahan nilai lainnya (Sumbu-x; biasanya perubahan waktu). Langkah-langkah berikut digunakan untuk membuat sebuah diagram garis.

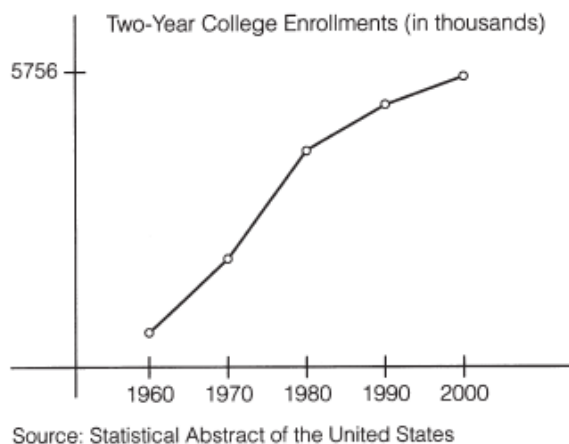
1. Lihat sekilas data yang akan ditampilkan. Diagram garis umumnya menampilkan 3 hingga 20 data yang merupakan perubahan tetap dalam satuan waktu: tahun, bulan atau hari. Posisi sumbu-x bisa berisi keterangan seperti “Jan Feb Mar Apr ...” atau “1996 1997 1998 ...”. label ini dapat diposisikan di koordinat 1, 2, 3, ... di sumbu-x.
2. Sistem koordinat diatur berdasarkan pada jumlah data dan besarnya jumlah setiap data. Skala yang ditetapkan untuk sumbu-x diatur antara -1 hingga banyak data +1. Sedangkan skala sumbu-y diatur hingga besarnya lebih dari nilai maksimum dari data.
3. Gambar segmen garis. Akan terlihat lebih bagus jika digambarkan lingkaran kecil di setiap ujung garis dari data.
4. Gambar label untuk data di sumbu-x dan sumbu-y.
5. Memberikan judul diagram dan sumber data.

Contoh Program Diagram Garis

Gambar 12.7 menunjukkan sebuah diagram garis dari data pendaftaran sebuah sekolah 2-tahunan yang diambil dari Statistical Abstract of the United States. Data ditampilkan dalam Tabel 12.1.

Tabel 12.1 Two-year Colleges Enrollment (in Thousands)

Year	1960	1970	1980	1990	2000
Male	283	1375	2047	2233	2398
Female	170	945	2479	3007	3358
Total	453	2320	4526	5240	5756



Gambar 12.7 Diagram garis dari contoh program.

SOLUSI:

Program dibuat dengan antarmuka sederhana terdiri dari picture box dan command button. Data disimpan dalam sebuah file dengan nama “ENROLL.TXT” dengan isi file adalah teks dengan format .csv dengan data “1960”, 453.

Program dibuat dengan beberapa sub-program dalam bentuk procedure dan function.

Kode program untuk permasalahan di atas adalah sebagai berikut:

```
`In (Declarations) section of (General)
Dim numYears As Integer, maxEnroll As Single
```

‘Program UTAMA

```
Private Sub cmdDraw_Click()
    `Line Chart of Total Two-Year College Enrollments
    numYears = 5
    ReDim label(1 To numYears) As String
    ReDim total(1 To numYears) As Single

    Call ReadData(label(), total())
    Call DrawAxes
    Call DrawData(total())
    Call ShowTitle
    Call ShowLabels(label())
End Sub
```

‘Sub Program (Procedure dan Function)

```
Private Sub DrawAxes()
    `Draw axes
    picEnroll.Scale (-1, 1.2 * maxEnroll) - (numYears + 1,
-.2 * maxEnroll)
    picEnroll.Line (-1, 0)-(numYears + 1, 0)
    picEnroll.Line (0, -.1 * maxEnroll)-(0, 1.1 *
maxEnroll)
End Sub

Private Sub DrawData(total() As Single)
    i As Integer
    `Draw lines connecting data and circle data points
    For i = 1 To numYears
        If i < numYears Then
            picEnroll.Line (i, total(i))-(i + 1, total(i +
1))
        End If
        picEnroll.Circle (i, total(i)), .01 * numYears
    Next i
End Sub

Private Sub Locate(x As Single, y As Single)
    picEnroll.CurrentX = x
```

```

        picEnroll.CurrentY = y
    End Sub

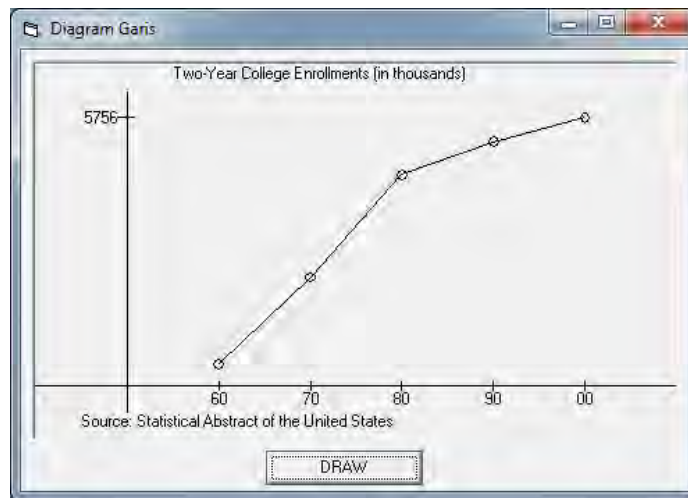
    Private Sub ReadData(label() As String, total() As Single)
        Dim i As Integer
        'Assume the data have been placed in the file "ENROLL.TXT"
        '(First line of the file is "1960",453)
        'Read data into arrays, find highest enrollment
        maxEnroll = 0
        Open "ENROLL.TXT" For Input As #1
        For i = 1 To numYears
            Input #1, label(i), total(i)
            If total(i) > maxEnroll Then
                maxEnroll = total(i)
            End If
        Next i
        Close #1
    End Sub

    Private Sub ShowLabels(label() As String)
        Dim i As Integer, lbl As String, lblWid As Single
        Dim lblHght As Single, tickFactor As Single
        'Draw tick marks and label them
        For i = 1 To numYears
            lbl = Right(label(i), 2)
            lblWid = picEnroll.TextWidth(lbl)
            tickFactor = .02 * maxEnroll
            picEnroll.Line (i, -tickFactor)-(i, tickFactor)
            Call Locate(i - lblWid / 2, -tickFactor)
            picEnroll.Print lbl
        Next i
        lbl = Str(maxEnroll)
        lblWid = picEnroll.TextWidth(lbl)
        lblHght = picEnroll.TextHeight(lbl)
        tickFactor = .02 * numYears
        picEnroll.Line (-tickFactor, maxEnroll)-(tickFactor,
maxEnroll)
        Call Locate(-tickFactor - lblWid, maxEnroll - lblHght /
2)
        picEnroll.Print lbl
    End Sub

    Private Sub ShowTitle()
        'Display source and title
        Call Locate(-.5, -.1 * maxEnroll)
        picEnroll.Print "Source: Statistical Abstract of the
United States"
        Call Locate(.5, 1.2 * maxEnroll)
        picEnroll.Print "Two-Year College Enrollments (in
thousands)"
    End Sub

```

Jika program diatas dijalankan maka akan menghasilkan grafik seperti ditunjukkan pada Gambar 12.8. Perhatikan ERROR yang terjadi!!!



Gambar 12.8 Hasil running program diagram garis.



REFERENSI

1. BROOKSHEAR, J. Glenn. *Computer Science: Suatu Pengantar*. Erlangga, 2003.
2. Rajiv Khanna, *Computer Application for Engineer*, New Age International Publisher.
3. Sherry Willard Kinkoph, *Teach yourself Visually Microsoft Office 2007*, Wiley Publishing, Inc., 2007.
4. Steve Johnson, *Microsoft Office Word 2007 on Demand*, Que Publishing, 2007.
5. Joyce Cox, Joan Preppernau, *Microsoft® Office Word 2007 Step by Step*, Microsoft Press, 2007.
6. Jill T. Freeze, *Teach yourself Computer Basics in 24 Hours*, 3rd ed., Sams Publishing, 2003.
7. Curtis D. Frye, *Step by Step Microsoft Excel 2007*, Microsoft Press, 2007.
8. Joyce Cox, Joan Preppernau, *Microsoft® Office Powerpoint 2007 Step by Step*, Microsoft Press, 2007.
9. Richarr Harrington, Scott Rekdal, *How to WOW with PowerPoint*, Peachpit Press, 2007.
10. David I. Schneider, *Computer Programming Concepts and Visual Basic*, Pearson Custom Publishing, 1999.
11. Krisna D. Octovhiana, *Cepat Mahir Visual Basic 6.0*, IlmuKomputer.com, 2003.