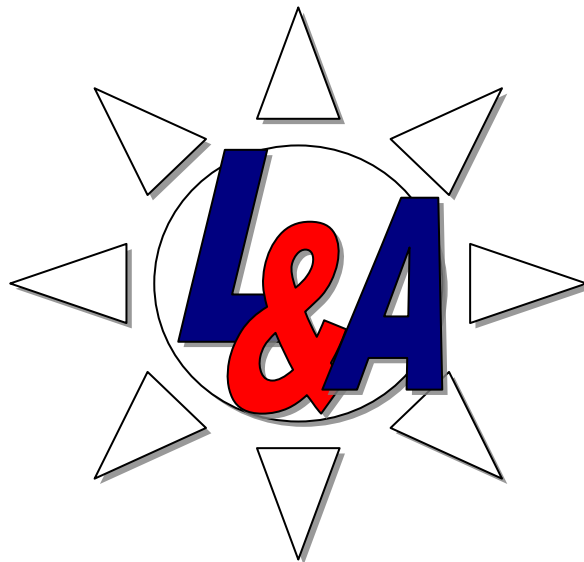


MODUL PRAKTIKUM

LOGIKA DAN ALGORITMA DENGAN PASCAL

**DIGUNAKAN UNTUK BAHAN PRAKTIKUM
MATAKULIAH PRAKTIKUM LOGIKA DAN ALGORITMA**



**Disusun Oleh:
Z A I D I R, ST.**

**PROGRAM STUDI MANAJEMEN INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS RESPATI YOGYAKARTA
2013**

DAFTAR ISI

	Hal
Halaman Judul	i
Daftar Isi	ii
Kata Pengantar	iii
Ketentuan kegiatan Praktikum	iv
Pertemuan 1 Mengetahui Aplikasi Untuk Pembuatan Flowchart	1
Pertemuan 2 Membuat Flowchart Untuk Masalah Sederhana	10
Pertemuan 3 Mengetahui Bahasa Pemrograman Pascal	18
Pertemuan 4 Struktur Program dan Statemen Input-Output	34
Pertemuan 5 Struktur Program dan Statemen Input-Output Lanjutan	39
Pertemuan 6 Pernyataan Bercabang.....	46
Pertemuan 7 Pernyataan Bercabang Lanjutan	52
Pertemuan 8 Pernyataan Berulang	61
Pertemuan 9 Pernyataan Berulang Lanjutan	66
Pertemuan 10 Variabel Berindeks (Array)	74
Pertemuan 11 Prosedur dan Fungsi	94
Pertemuan 12 Pemrograman Dengan Teknik Rekursi	104
Daftar Pustaka	114

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah S.W.T yang telah memberikan limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan modul praktikum mata kuliah Praktikum Logika dan Algoritma. Modul praktikum ini bertujuan untuk dapat dijadikan bahan kelengkapan kegiatan praktikum agar bisa berjalan dengan baik dan lancar.

Dalam penyusunan modul ini tentu saja banyak pihak yang ikut membantu dalam hal materi maupun dukungan moril. Pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada berbagai pihak yang tidak bisa disebutkan satu persatu yang telah membantu dalam penyelesaian modul ini.

Penulis sadari bahwa modul praktikum ini masih memiliki banyak kekurangan. Untuk itu saran dan kritik membangun dari pembaca sangat penulis harapkan dan nantikan demi perbaikan dan pengembangan di masa mendatang.

Yogyakarta, September 2013

Zaidir

KETENTUAN KEGIATAN PRAKTIKUM

A. Kelengkapan Praktikum

1. Kegiatan praktikum dilengkapi dengan modul praktikum
2. Perlu memiliki *flashdisk*
3. Hasil kegiatan praktikum disimpan pada folder masing-masing (bisa dibuat pada drive komputer atau flashdisk atau dibuat di server)

B. Tata Tertib Praktikum

1. Peserta praktikum (praktikan) harus memenuhi atau membawa kelengkapan praktikum setiap kali mengikuti kegiatan praktikum.
2. Untuk mengikuti praktikum harus berpakaian rapi dan sopan.
3. Pada saat kegiatan praktikum, peserta praktikum harus menjaga ketenangan, ketertiban, kebersihan dan kerapian.
4. Peserta praktikum harus datang tepat waktu, dengan waktu toleransi keterlambatan 15 menit.
5. Alat komunikasi dinyalakan dalam mode silent atau dimatikan.
6. Selama praktikum, peserta praktikum tidak diperkenankan meninggalkan ruangan praktikum/lab tanpa seizin pengampu/dosen praktikum.
7. Peserta praktikum harus bertutur kata yang baik dan sopan kepada pengampu/dosen praktikum.
8. Peserta praktikum harus dapat menunjukkan sikap kejujuran. Apabila menemukan perlengkapan praktikum peserta praktikum lain yang tertinggal di komputer atau meja yang digunakan, maka dapat memberitahukan kepada petugas lab untuk selanjutnya akan dibuat pengumuman.
9. Jadwal kegiatan praktikum yang karena sesuatu sebab tidak bisa dilaksanakan, maka akan dicarikan jadwal pengganti dihari lain berdasarkan kesepakatan pengampu/dosen praktikum dengan peserta praktikum.
10. Peserta praktikum harus memenuhi minimal kehadiran yaitu 75% dari total pertemuan.
11. Pada saat dilaksanakan **ujian responsi**, peserta praktikum diwajibkan menggunakan **jas almamater**.
12. Hal-hal yang belum diatur pada tata tertib ini akan diatur kemudian.

C. Sangsi

1. Pengampu/dosen praktikum berhak memperingatkan bahkan mengeluarkan peserta praktikum yang tidak dapat menjaga ketenangan, ketertiban, kebersihan dan kerapian.
2. Apabila peserta praktikum datang lebih dari 15 menit, maka tidak diperkenankan mengikuti praktikum.
3. Peserta praktikum yang tidak memenuhi minimal kehadiran 75% dari total pertemuan, maka tidak diperkenankan mengikuti responsi/UAS. Dengan demikian maka nilai akhiri untuk mata kuliah praktikum adalah E.

D. Penilaian

Kegiatan praktikum akan dinilai dengan beberapa komponen sebagai berikut:

No	Komponen Penilaian	Bobot Penilaian
1	Praktikum Harian	30%
2	Tugas	30%
3	Responsi/UAS	40%

Mengenal Aplikasi Untuk Pembuatan Flowchart

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat menggunakan aplikasi untuk membuat flowchart, mengetahui simbol-simbol flowchart, mengetahui langkah-langkah yang harus dilakukan untuk membuat flowchart, dapat menyusun flowchart untuk sebuah permasalahan.

Pertemuan ini akan menjelaskan secara singkat tentang pengertian flowchart, simbol-simbol flowchart, langkah-langkah dalam menyusun flowchart, dan praktek untuk berbagai persoalan.

1.1 Definisi

1. Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program.
2. Flowchart Program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan.
3. Flowchart ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi.
4. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut
5. Programmer menggunakan flowchart program untuk menggambarkan urutan instruksi dari program komputer.
6. Analis Sistem menggunakan flowchart program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi

1.2 Pedoman-Pedoman Dalam Membuat Flowchart

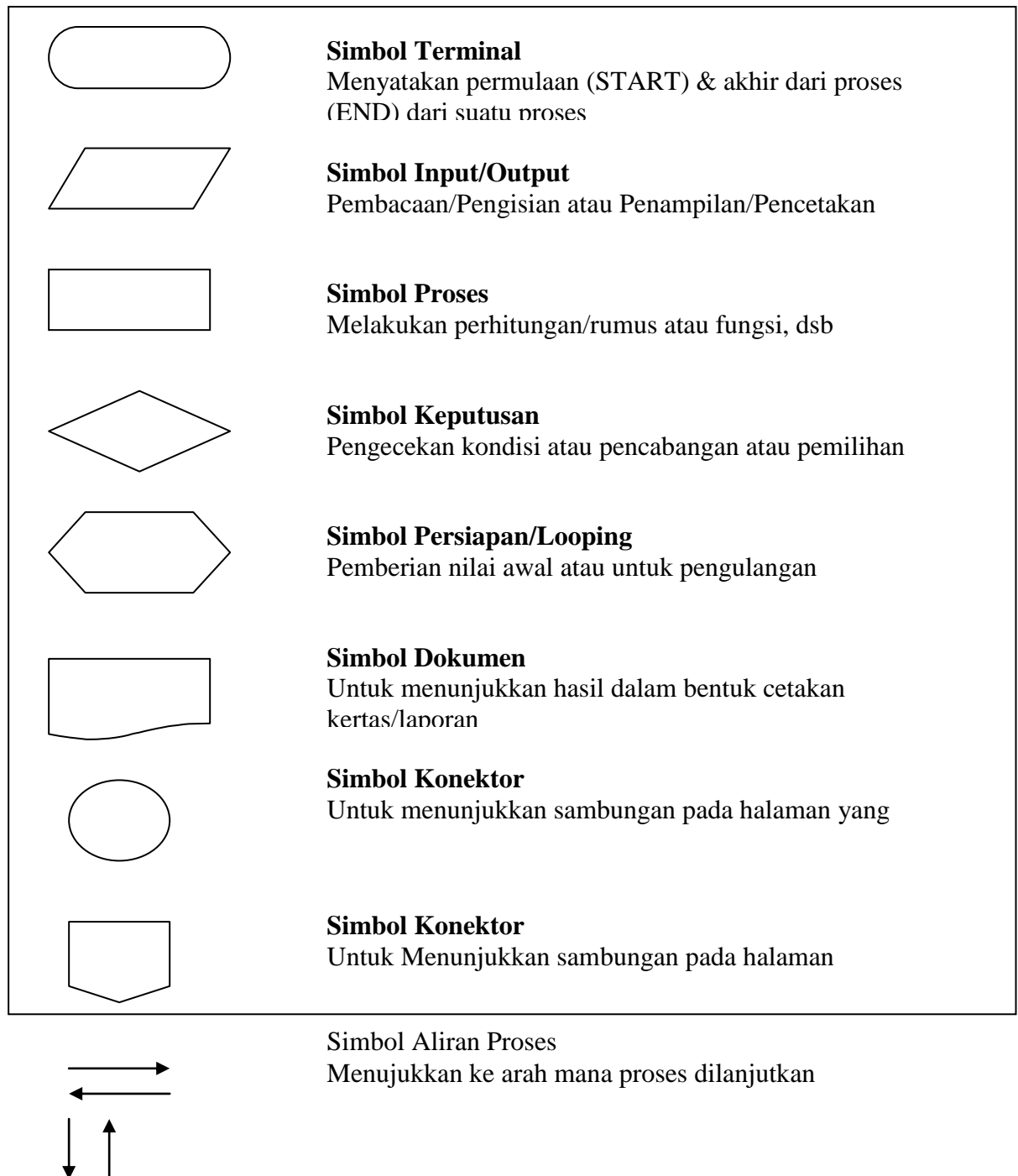
Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti :

1. Flowchart digambarkan dari halaman **atas** ke **bawah** dan dari **kiri** ke **kanan**.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan **MENGHITUNG PAJAK PENJUALAN**.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu

digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol flowchart yang standar.

1.3 Simbol-Simbol Flowchart



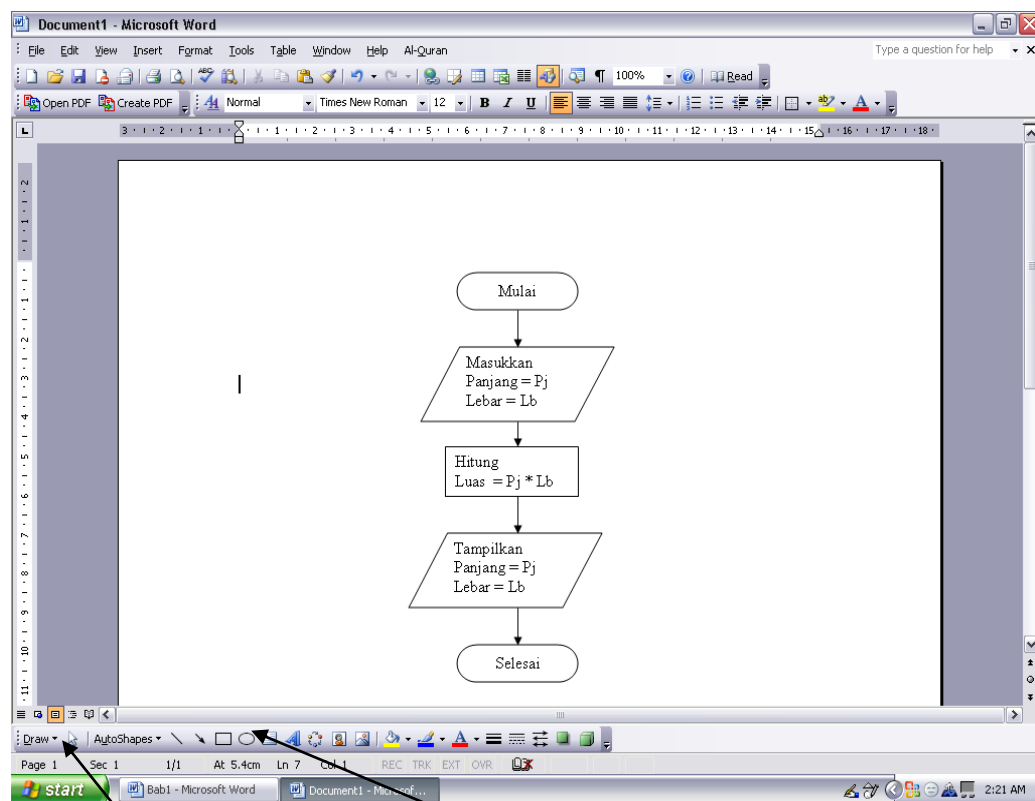
Gambar 1 : Simbol-simbol Flowchart

1.4 Aplikasi Untuk Membuat Flowchart

Aplikasi yang bisa digunakan untuk menggambar flowchart cukup banyak, diantaranya: Software Flowchart, MS Word, Visio, dll. Tentu saja masing-masing perangkat lunak (software) tersebut memiliki kelebihan dan kekurangan masing-masing. Tipe file yang dihasilkan juga berbeda-beda.

Jika menggunakan software Flowchart, memang perangkat lunak yang khusus dibuat untuk keperluan perancangan flowchart, maka pengguna cukup fokus. Jika menggunakan MS Word sifatnya merupakan fasilitas tambahan. Sedangkan Visio, merupakan perangkat lunak yang cukup kompleks, selain flowchart juga menyediakan untuk kepentingan lain seperti untuk bidang teknik dan lain-lain.

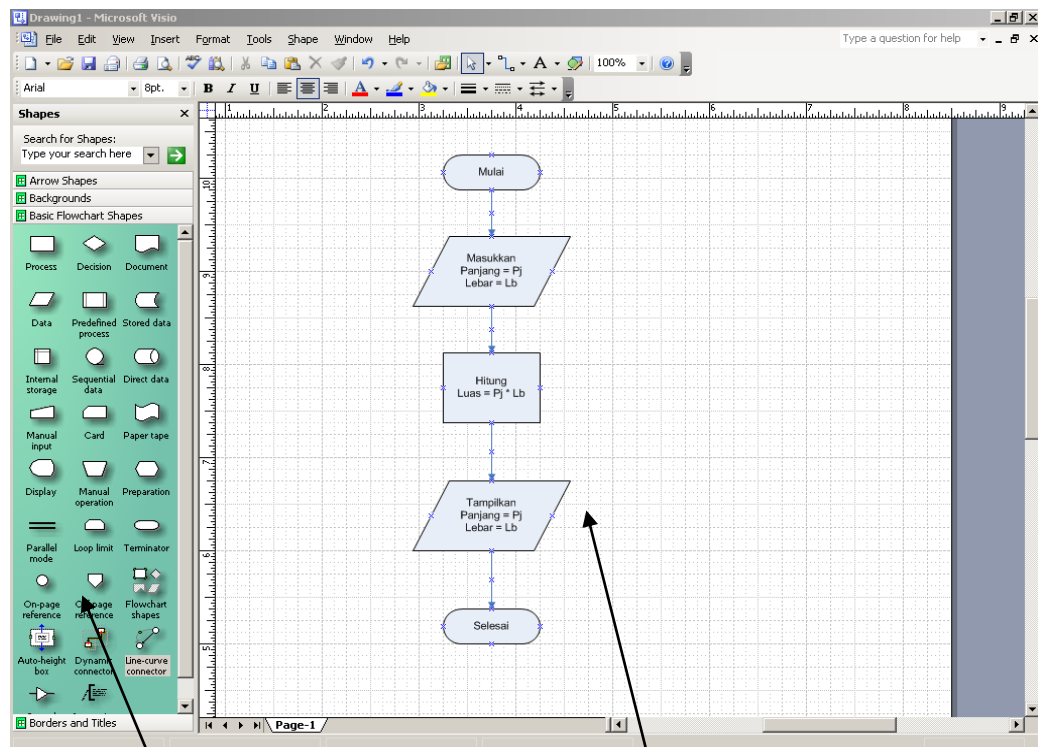
Untuk kepentingan praktikum akan digunakan MS Word dan Visio, sedangkan untuk sistem flowchart dapat dicoba sendiri dengan menginstal pada komputer masing-masing.



Untuk merubah gambar

Untuk membuat gambar

Gambar 2 : Membuat Flowchart Dengan MS Word



Simbol-simbol yang digunakan

Area untuk membuat

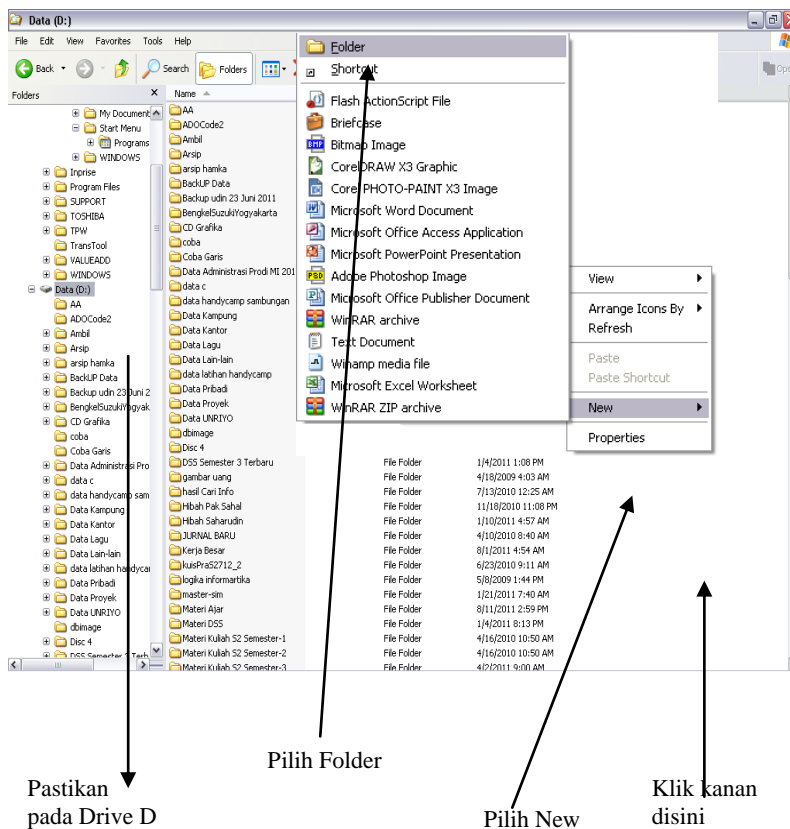
Gambar 3 : Membuat Flowchart Dengan Visio

1.5 Praktek

1.5.1 Menyiapkan Folder Kerja

Folder kerja perlu disiapkan dengan seksama agar hasil praktek yang dilakukan dapat tersimpan dengan baik dan mudah untuk ditemukan kembali. Untuk keperluan praktek matakuliah Praktikum Logika dan Algoritma, maka siapkanlah folder kerja dengan langkah sebagai berikut:

1. Tampilkanlah windows explorer dengan cara klik kanan pada tombol **Start** pada windows taskbar, kemudian pilih **Explorer**. Bisa juga dengan cara yang lain tergantung dari sistem operasi yang anda hadapi saat ini, yang penting masuk ke windows explorer.
2. Setelah window explorer ditampilkan, klik drive (drive C, Drive D, drive E, dll) tempat folder/directory akan diciptakan. Sebaiknya pada drive yang aman dan kapasitasnya masih mencukupi. Untuk contoh praktek ini kita pakai drive D.



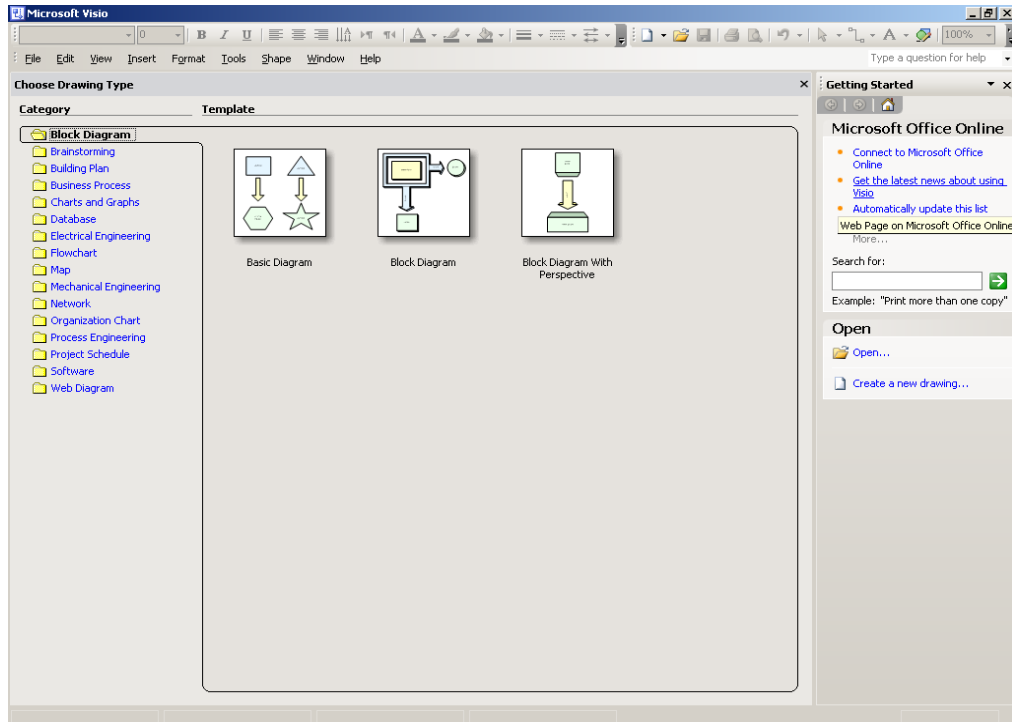
Gambar 1.5 Window Explorer

3. Klik kanan pada belahan bagian kanan window explorer pada bagian yang kosong, sehingga akan ditampilkan menu layang. Pada menu tersebut pilihlah **New**, lalu pilih **Folder**. Ketiklah nama folder yang diinginkan, misalnya **LA-16240001**.

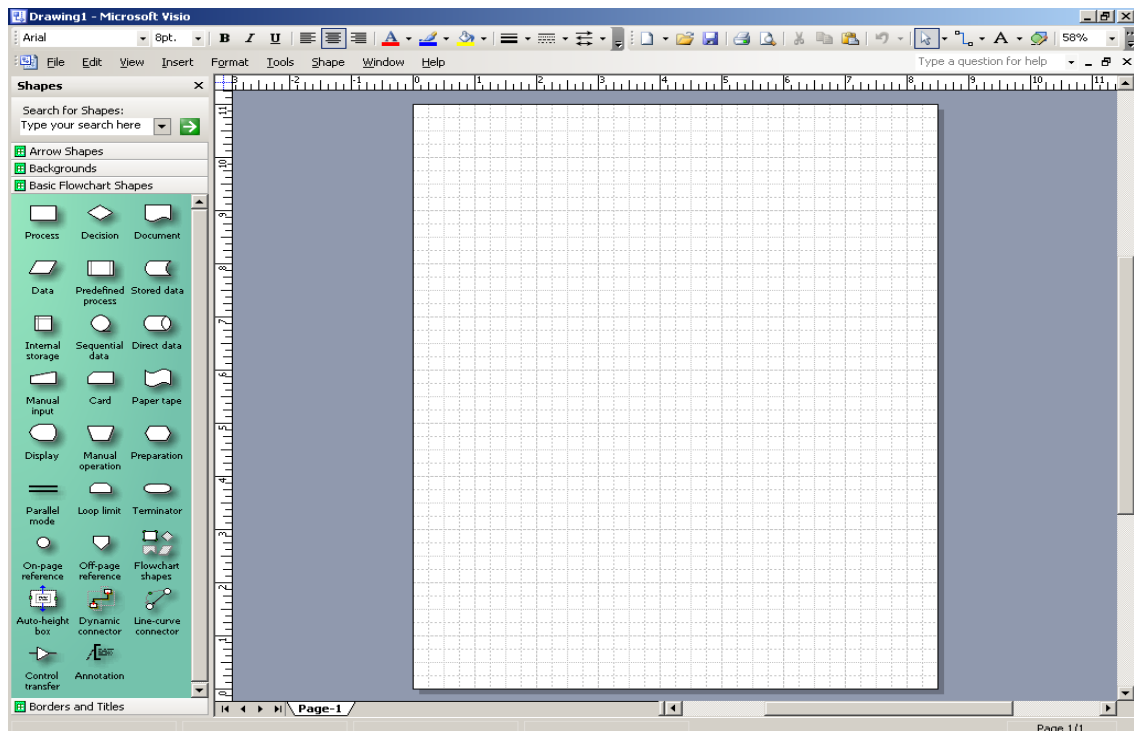
1.5.2 Membuat flowchart

Untuk melakukan praktikum pada kesempatan ini akan digunakan software **Visio**. Supaya bisa membuat flow chart, jalankanlah Visio dengan langkah sebagai berikut:

- Klik menu Start
- Pilih Microsoft Office Visio 2003 atau versi yang lain (biasanya Visio dijadikan satu dengan Microsoft Office) sampai ditampilkan seperti gambar di bawah ini



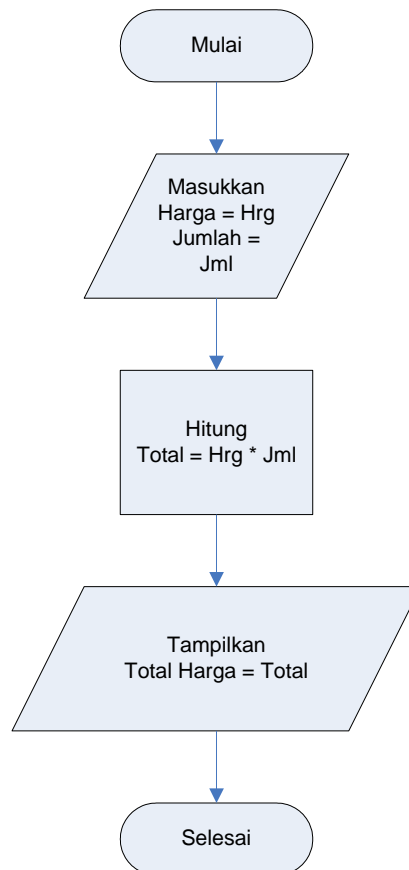
- Pilih Flowchart dan lanjutkan memilih Basic Flowchart, sampai ditampilkan tampilan seperti gambar di bawah ini



- Silahkan memulai membuat flowchart, untuk mengambil gambar yang dibutuhkan, lakukan dengan cara drag (klik gambar lalu geser ke arah tempat gambar diletakkan).

1. Membuat flowchart menghitung total harga

- Buatlah flowchart seperti gambar di bawah

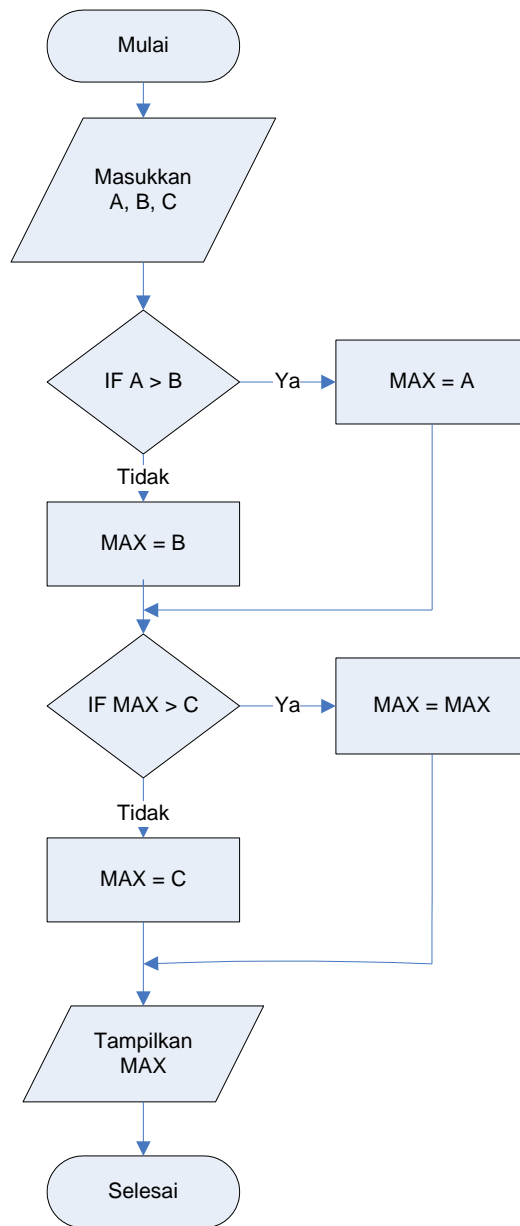


Gambar 4 : Flowchart Menghitung Total Harga

- Setelah selesai, simpanlah dengan nama **Flow1**

2. Membuat flowchart mencari nilai terbesar dari 3 nilai

- Buatlah flowchart seperti gambar di bawah

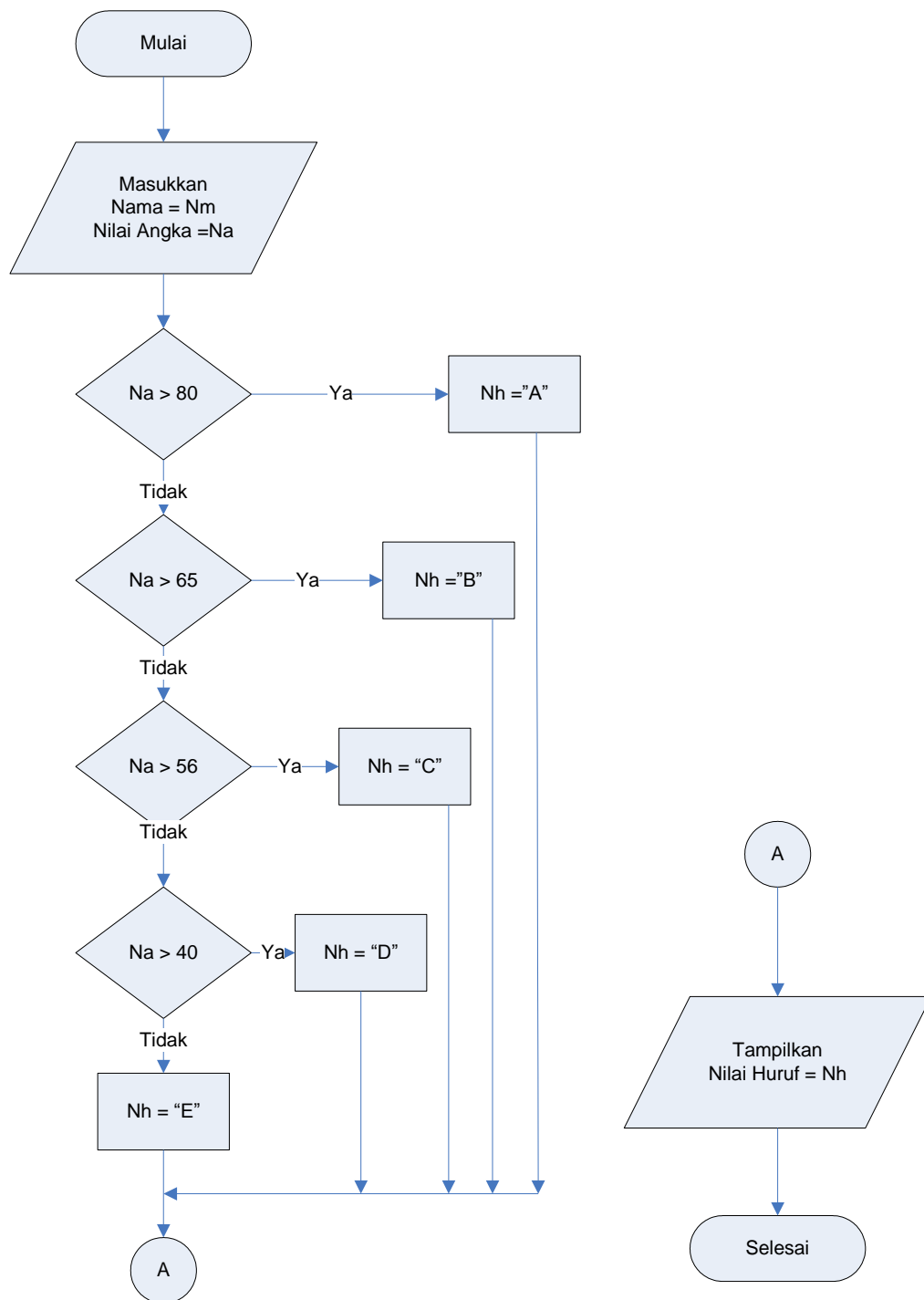


Gambar 5 : Flowchart Mencari Nilai Terbesar dari 3 Data

- Setelah selesai, simpanlah dengan nama **Flow2**

3. Membuat flowchart mengkonversi nilai angka ke nilai huruf

- Buatlah flowchart seperti gambar di bawah

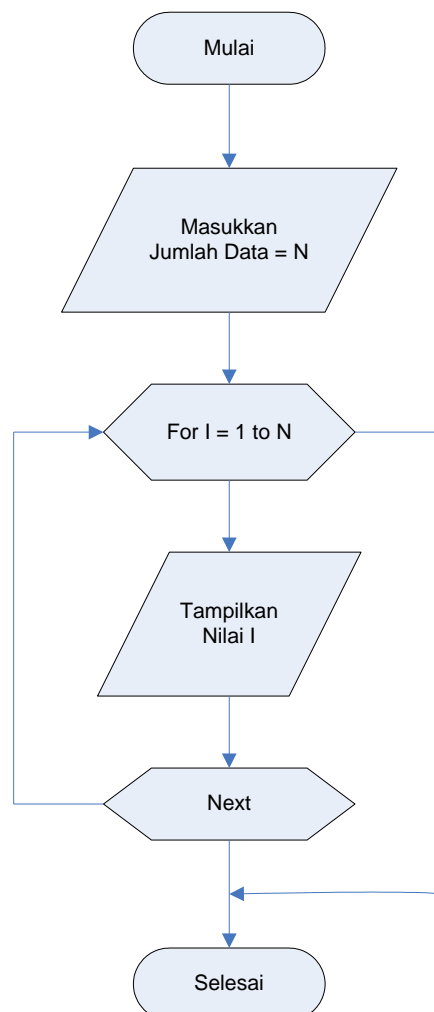


Gambar 6 : Flowchart Mengkonversi Nilai Angka ke Nilai Huruf

- Setelah selesai, simpanlah dengan nama **Flow3**

4. Membuat flowchart menampilkan data berurutan untuk sejumlah data dengan pengulangan

- Buatlah flowchart seperti gambar di bawah



Gambar 7 : Flowchart Menampilkan Data Berurutan Dengan Pengulangan

- Setelah selesai, simpanlah dengan nama **Flow4**

Membuat Flowchart Untuk Masalah-masalah Sederhana

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat menggunakan aplikasi untuk membuat flowchart, mengetahui simbol-simbol flowchart, mengetahui langkah-langkah yang harus dilakukan untuk membuat flowchart, dapat menyusun flowchart untuk sebuah permasalahan.

Pertemuan ini akan menjelaskan secara singkat tentang prinsip dasar algoritma, contoh kasus dan pemecahan dan praktek untuk berbagai masalah sederhana.

2.1 Prinsip Dasar Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu sistem komputer melibatkan 5 tahapan yaitu:

7. Analisis masalah
8. Merancang algoritma
9. Membuat program komputer
10. Menguji hasil program komputer
11. Dokumentasi

2.1.1 Analisis masalah

Langkah awal untuk mengetahui lebih jauh tentang permasalahan yang akan diselesaikan. Tujuannya adalah untuk menemukan ide-ide kemungkinan solusi terhadap permasalahan

2.1.2 Merancang algoritma

Tujuan merancang algoritma adalah:

1. Menentukan ide solusi: ide prosedur/urutan proses penyelesaian permasalahan secara global
2. Menyatakan algoritma: uraian rinci dari prosedur global yang berupa serangkaian langkah yang terstruktur
3. Memvalidasi algoritma: untuk menguji validitas prosedur yang telah dinyatakan dalam algoritma terhadap seluruh kemungkinan data yang di-input-kan
4. Menganalisis algoritma: untuk mengetahui tingkat efisiensi proses dalam algoritma (efisiensi waktu dan efisiensi penggunaan memori)

2.1.3 Kriteria penyelesaian masalah dalam bentuk algoritma

- Setiap Langkah harus bersifat pasti/tertentu
- Minimal menghasilkan sebuah output
- Terstruktur dan Sistematis
- Memiliki kriteria untuk menghentikan proses

Algoritma yang baik adalah: bersifat efisien, baik dari segi waktu maupun penggunaan memori dalam sistem komputer.

Tujuan Algoritma adalah : memberikan petunjuk tentang langkah-langkah logika penyelesaian permasalahan dalam bentuk yang mudah dipahami nalar manusia sebagai acuan yang membantu dalam mengembangkan program komputer

Pemahaman tentang algoritma akan mencegah sejak dini kemungkinan terjadinya kesalahan logika pada program komputer yang dikembangkan.

Syarat untuk mencapai tujuan tersebut adalah :

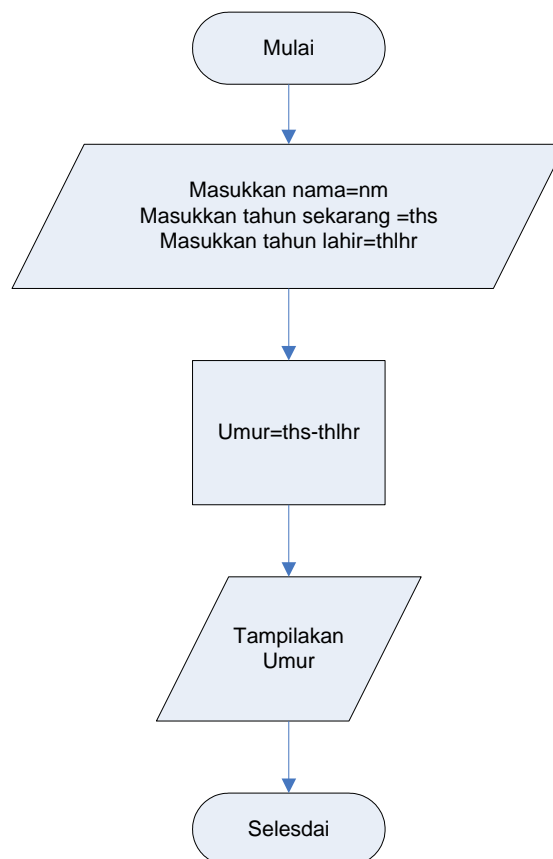
1. Logika prosedur pada algoritma harus cukup mudah dipahami nalar manusia.
2. Validitas prosedur pada algoritma dapat ditelusuri dengan mudah
3. Prosedur pada algoritma harus cukup mudah dikonversi ke program komputer
4. Prosedur pada algoritma tidak terpengaruh atau bergantung pada bahasa pemrograman

2.2 Contoh Kasus dan Pemecahan

1. Kasus menghitung umur

Pada suatu kecamatan, akan dilakukan pendataan bagi warganya agar mudah untuk mengetahui umur masing-masing. Data yang dimasukkan adalah nama, tahun lahir dan tahun sekarang. Output yang diinginkan adalah umur (dalam tahun).

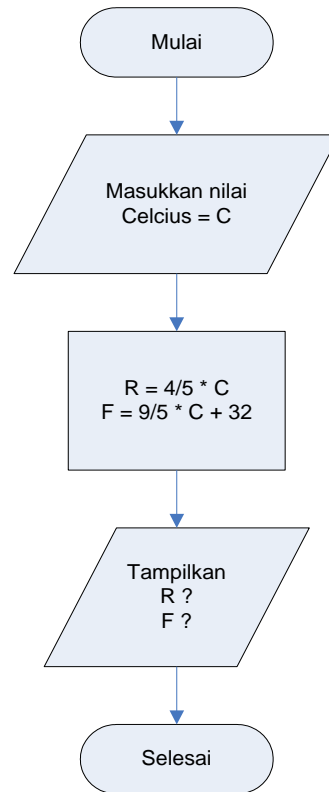
Pemecahan:



2. Kasus konversi dari Celcius ke Reumur dan Farenheit

Untuk keperluan pemahaman terhadap suhu, diperlukan proses konversi dari suhu Celcius ke bentuk suhu dalam bentuk Reumur dan Fareinheit. Data yang menjadi masukan adalah ukuran suhu dalam derajat Celcius. Output yang diinginkan adalah derajat Reumur dan Fareinheit. Rumus yang berlaku untuk konversi ke Reumur adalah $R = 4/5 * C$, sedangkan rumus untuk konversi ke Farenheit adalah $F = 9/5 * C + 32$.

Pemecahan:



3. Kasus penggajian karyawan

Sebuah perusahaan garmen, ingin membuat program aplikasi penggajian untuk karyawannya. Saat ini jumlah karyawan keseluruhan sebanyak 200 orang, dan kemungkinan akan bertambah untuk masa yang akan datang. Dalam sistem penggajian tersebut, data-data yang perlu dimasukkan melalui keyboard adalah NIK, Nama, UnitKerja, Gol, dan StKawin. Output yang diinginkan adalah Gaji Pokok, Tunjangan dan Jumlah Gaji.

Gaji Pokok, Tunjangan dan Jumlah Gaji akan muncul secara otomatis dengan ketentuan sebagai berikut :

Gaji Pokok : didapat dengan menguji Golongan, dengan aturan :

Jika Gol = "IV" maka Gaji Pokok = 800000

Jika Gol = "III" maka Gaji Pokok = 600000

Jika Gol = "II" maka Gaji Pokok = 500000

Jika Gol = "I" maka Gaji Pokok = 400000

Tunjangan : didapat dengan menguji Status Kawin, dengan aturan :

Jika StKawin = "Kawin" maka Tunjangan = 10% dari Gaji Pokok

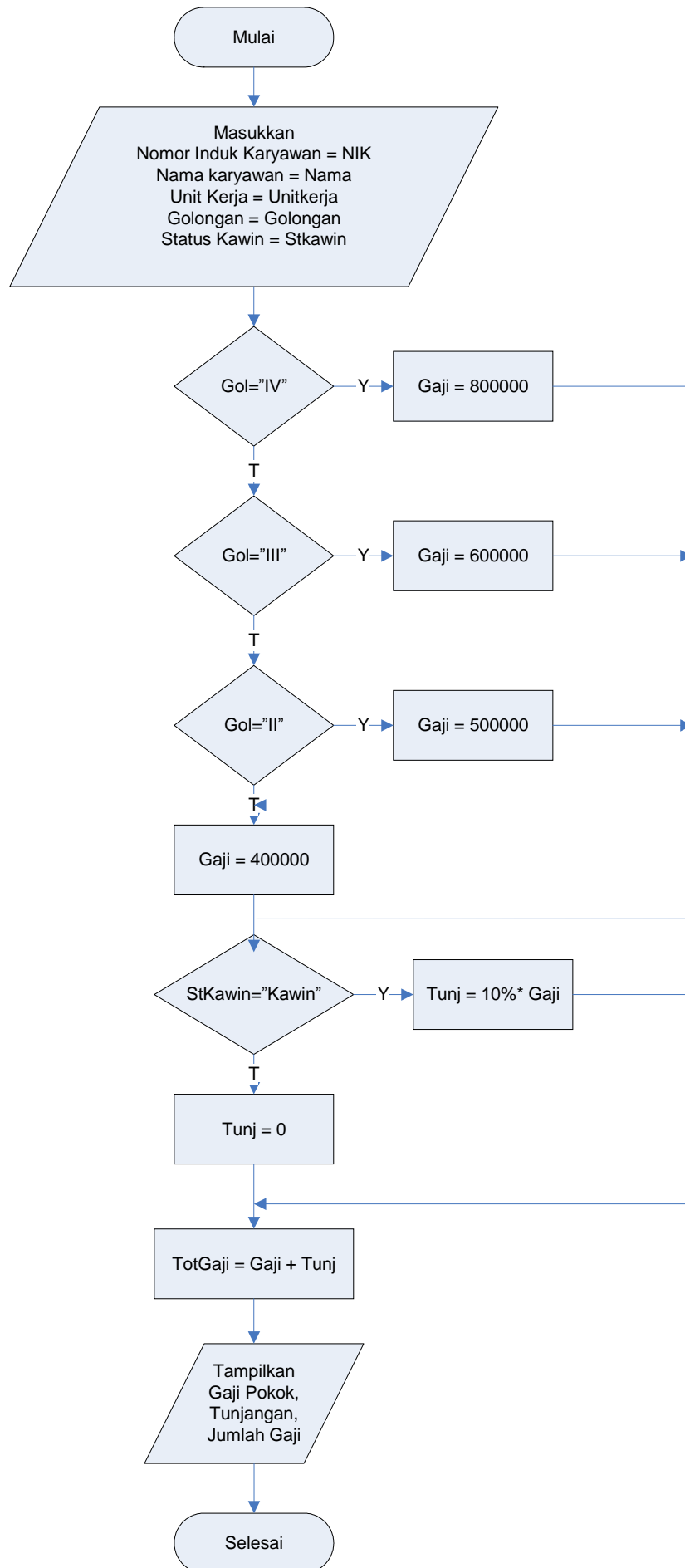
Jika StKawin = "Tidak Kawin" maka Tunjangan = 0

Jumlah Gaji : didapat dengan menjumlahkan Gaji Pokok ditambah Tunjangan

Permintaan:

Buatlah algoritma dalam bentuk flow chart

Pemecahan:



2.3 Praktek

1. Kasus Pengolahan data lembur karyawan

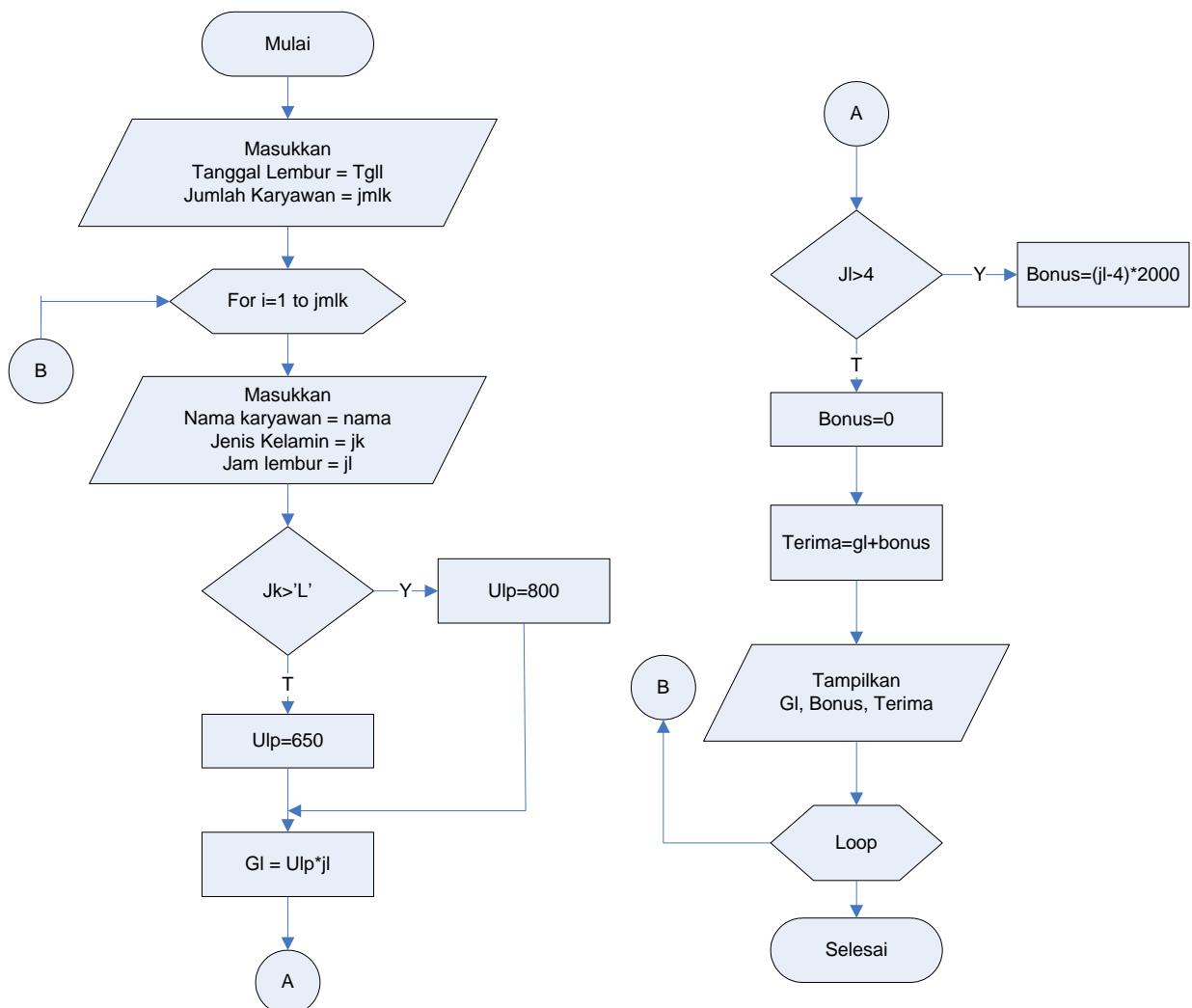
Sebuah perusahaan pakaian jadi (garmen), ingin membuat program untuk mengatasi permasalahan lembur karyawan. Proses pengolahan data yang dilakukan adalah diawali dengan mengisi tanggal lembur, dilanjutkan mengisi jumlah karyawan yang lembur. Setelah mengisi data di atas, proses selanjutnya adalah mengisi data lembur karyawan secara berulang-ulang sebanyak jumlah karyawan. Data yang perlu diisi secara berulang adalah nama karyawan, jenis kelamin (L = laki-laki, P = perempuan), dan jumlah jam lembur.

Aturan lembur yang digunakan adalah :

- Jika jenis kelamin = "L" maka upah lembur per jam = 8000, selain dari itu (jika jenis kelamin = "P") upah lembur per jam = 6500
- Gaji Lbr = jumlah jam lembur * upah lembur per jam
- Bagi yang lembur diatas 4 jam, mendapat Bonus 2000 setiap kelebihan jamnya.
- Jml Terima = Gaji Lbr + Bonus

Tampilkanlah gaji lembur, bonus, dan jumlah terima!

Permintaan : Buatlah flowchart untuk menyelesaikan permasalahan di atas. Gunakanlah variabel-variabel yang mudah dimengerti dan singkat



- Setelah selesai, simpanlah dengan nama **Flow5**

2. Kasus tarif PDAM

Perusahaan Daerah Air Minum ingin membuat program otomatisasi untuk perhitungan biaya pemakaian pelanggan. Hal pertama yang harus diisikan saat program dijalankan adalah mengisi **Jumlah Pelanggan** yang akan diisikan. Setelah itu proses akan diulangi sebanyak jumlah pelanggan tersebut untuk memasukkan data yang berupa **Nama Pelanggan, Alamat Pelanggan, Status Pelanggan** dan **Jumlah Pemakaian**.

Setelah melakukan pengisian data maka Biaya Tarif Dasar, Biaya Pemakaian, Jumlah Biaya, Potongan Biaya dan Jumlah Bayar akan dihitung dan ditampilkan secara otomatis.

Aturan Tarif Dasar:

- Jika Status Pelanggan="Umum" maka **Biaya Tarif Dasar** = 35000
- Jika Status Pelanggan ="Perusahaan" maka **Biaya Tarif Dasar** = 55000
- Jika Status Pelanggan ="Yayasan" maka **Biaya Tarif Dasar** = 15000

Aturan Biaya Pemakaian:

- Jika Jumlah Pemakaian lebih kecil dari 100 M3, maka **Biaya Pemakaian** 20% dari Biaya Tarif Dasar
- Jika Jumlah Pemakaian lebih besar atau sama dengan 100 M3, maka **Biaya Pemakaian** = 10% dari Biaya Tarif Dasar.

Aturan Jumlah Biaya

Jumlah Biaya = Biaya Tarif Dasar + Biaya Pemakaian

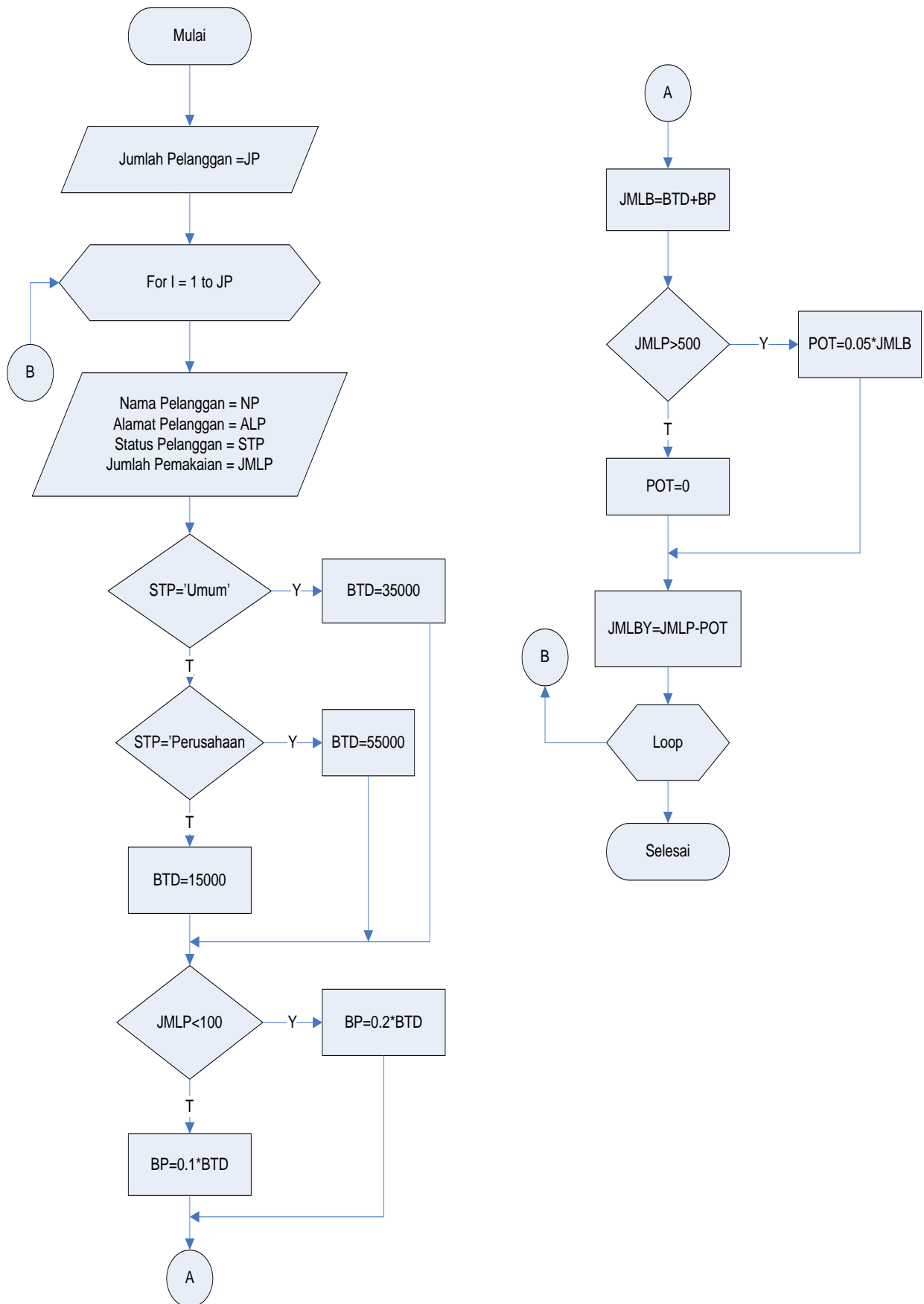
Aturan Potongan:

- Jika Jumlah Pemakaian lebih besar dari 500 M3, maka diberikan **Potongan** = 5% dari Jumlah Biaya
- Jika Jumlah Pemakaian lebih kecil atau sama, maka **Potongan** = 0

Aturan Jumlah Bayar:

Jumlah Bayar = Jumlah Biaya - Potongan

Permintaan: Buatlah flowchart



- Setelah selesai, simpanlah dengan nama **Flow6**

2.4 Soal-soal

1. Kasus pengelolaan cagar budaya

Instansi pemerintah yang mengurus bagian pengelolaan cagar budaya akan melakukan proses komputerisasi mengenai data-data pengelola cagar budaya agar lebih tertata dan beberapa proses bisa dilakukan secara otomatis. Data-data yang perlu dimasukkan adalah nomor kontrak, nama pengelola, alamat pengelola, telepon pengelola, jenis cagar budaya. Output yang diinginkan setelah mengisi jenis cagar budaya adalah besar investasi, dana abadi dan total setoran. Untuk menghasilkan output tersebut ada aturan yang harus dipenuhi yaitu:

a. Besar investasi:

Besar investasi yang harus dikeluarkan oleh pengelola adalah:

- Jika jenis cagar budaya='Museum' maka besar investasi = 300
- Jika jenis cagar budaya='Candi' maka besar investasi = 750
- Jika jenis cagar budaya='Situs' maka besar investasi = 200

b. Dana Abadi:

- Jika jenis cagar budaya='Museum' atau jenis cagar budaya='Candi' maka dana abadi = 900
- Jika jenis cagar budaya='Situs' maka besar investasi = 500

c. Total setoran:

Rumus untuk Total setoran = besar investasi + dana abadi

2. Kasus layanan penduduk

Pada saat ini terdapat permasalahan di kantor kecamatan yang terkait dengan pelayanan kepada penduduk. Untuk itu akan dilakukan proses komputerisasi agar pelayanan semakin cepat dan prima. Data yang menjadi masukan antara lain: nama penduduk, alamat, nomor telepon dan jenis layanan yang diinginkan. Output yang diinginkan setelah mengisi jenis layanan adalah biaya administrasi, biaya perangko, dan jumlah bayar. Ketentuan yang berlaku untuk menghasilkan output tersebut adalah:

a. Biaya administrasi

- Jika jenis layanan='Kelahiran' maka biaya administrasi=5000
- Jika jenis layanan='Kematian' maka biaya administrasi=0
- Jika jenis layanan='Pindah' maka biaya administrasi=10000
- Jika jenis layanan='Izin Usaha' maka biaya administrasi=50000

b. Biaya perangko

Biaya perangko hanya dikenakan untuk jenis layanan='Izin usaha' sebesar 18000, sedangkan jenis layanan yang lain tidak dikenakan atau biaya perangko=0

c. Jumlah bayar

Rumus untuk Jumlah bayar = biaya administrasi + biaya perangko

3. Kasus penjualan Lembar Kerja Siswa (LKS)

Sebuah perusahaan yang bergerak dibidang percetakan dan penjualan LKS membutuhkan pengolahan data yang dilakukan secara terkomputerisasi.

Mengenal Bahasa Pemrograman Pascal

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman, standar penyusunan program, elemen program. Selain itu juga diharapkan mahasiswa bisa menggunakan pemrograman Turbo Pascal, mengenali lingkungan Turbo Pascal, mengetahui langkah-langkah yang harus dilakukan untuk membuat aplikasi dengan Turbo Pascal.

Pertemuan ini akan menjelaskan secara singkat tentang konsep pemrograman Turbo Pascal dan lingkungannya dan konsep-konsep penting dalam membangun sebuah aplikasi dengan menggunakan Turbo Pascal.

3.1 Pengertian

Pemrograman / programming : adalah pekerjaan penulisan instruksi bagi komputer untuk menyelesaikan suatu masalah.

Program : adalah himpunan instruksi yang diperuntukkan bagi komputer, untuk menyelesaikan masalah yang diberikan.

Interpreter : menerjemahkan perintah baris demi baris dan langsung melaksanakannya. contoh : BASIC.

Compiler : menerjemahkan dulu seluruh perintah dalam bahasa mesin. Baru kemudian terjemahan tersebut dijalankan oleh komputer. Contoh : FORTRAN, C, PASCAL.

3.2 Sejarah Singkat Tentang Bahasa Pascal

Bahasa PASCAL pertama kali dikembangkan pada awal tahun 70-an oleh NIKLAUS WIRTH di Technical University, Zurich – Swiss.

Nama PASCAL diambil dari nama seorang ahli matematika bangsa Perancis, yaitu BLEISE PASCAL yang telah berjasa menemukan alat hitung mekanis pertama di dunia pada abad ke-17.

Bahasa pemrograman ini termasuk kategori "High Level Language". Instruksi-instruksi yang digunakan dalam bahasa pemrograman ini sangat sistematis dan terstruktur.

Pada Awalnya bahasa pemrograman ini diperkenalkan dengan tujuan untuk menjelaskan masalah pemrograman komputer bagi mahasiswa yang belajar pemrograman komputer.

Ternyata dalam waktu singkat, bahasa pemrograman ini menjadi salah satu bahasa yang sangat populer di kalangan universitas, sehingga menjadi julukan sebagai bahasa universitas.

Mulai dari awal perkembangannya hingga saat ini banyak sekali jenis bahasa pemrograman ini, masing-masing merupakan hasil pengembangannya, antara lain :
UCSD Pascal

Microsoft Pascal
Apple Pascal
Turbo Pascal
dsb

Diantara versi-versi yang ada, Turbo Pascal merupakan versi yang sangat populer saat ini.

3.3 Struktur Dan Komponen Dasar Program Pascal.

Struktur dari suatu program Pascal terdiri dari sebuah judul program dan suatu blok program atau badan program. Blok program dibagi lagi menjadi dua bagian, yaitu bagian deklarasi dan bagian pernyataan (statement).

3.3.1 Struktur program :

Judul Program	PROGRAM nama-program;
Blok Program	
Bagian deklarasi	
- deklarasi label	LABEL nama-label;
- deklarasi konstanta	CONST.....;
- deklarasi tipe	TYPE;
- deklarasi variabel	VAR;
- deklarasi prosedur	PROCEDURE nama-prosedur;;
- deklarasi fungsi	FUNCTION nama-fungsi;;
Bagian Pernyataan	
Begin	
(statement)	
.....;	
.....;	
end.	

Contoh : Menghitung luas bidang berbentuk empat persegi panjang dengan panjang P dan lebar L.

PROGRAM Luas;	{Judul}
VAR P,L,Luas : real;	{Deklarasi variabel}
BEGIN	
Read(P);	{Statement}
Read(L);	{Statement}
Luas := P*L;	{Statement}
Write(P,L,Luas);	{Statement}
END.	

Judul program sifatnya adalah optional, dan bila ditulis, harus terletak pada awal dari program dan diakhiri dengan titik koma.

Bagian deklarasi digunakan bila di dalam program digunakan pengenalan (identifier). Identifier dapat berupa label, konstanta, tipe, variabel, prosedur dan fungsi. Kalau suatu program menggunakan identifier, Pascal menuntut supaya identifier tersebut diperkenalkan terlebih dahulu sebelum digunakan, yaitu dideklarasikan terlebih dahulu pada bagian ini.

Beberapa aturan dalam program Pascal :

- Akhir sebuah program Pascal ditandai dengan tanda baca titik (.) setelah END yang paling akhir.
- Tanda titik koma (;) merupakan pemisah antar instruksi satu dengan lainnya.
- Beberapa statement boleh ditulis menjadi satu baris dipisahkan dengan tanda baca titik koma (;)
Contoh : `simpan := akhir; akhir := simpan + awal;`
- Baris komentar diletakkan diantara tanda (* dan *) atau diantara tanda { dan }
Contoh : `Var rerata : real; (*nilai rata-rata*)
 Nil1 : real; {nilai ujian}`

Statement (pernyataan)

Adalah instruksi atau gabungan instruksi, yang menyebabkan komputer melakukan aksi.

Type statement dalam Pascal terdiri atas :

1. Sederhana :
 - menandai sebuah item data ke sebuah variabel (assignment statement)
contoh : `X := Y * 4.135`
 - pemanggilan procedure dan goto statement
2. Terstruktur:
 - Compound Statement
contoh : `Begin
 read (x) ;
 y := x + 5;
 write (y)
 End.`
 - Repetitive Statement
contoh : `For I := 1 to 100 do
 write (count);`
 - Conditional Statement
contoh : `If x > 100 then write (s)
 else write (p) ;`

3.3.2 Komponen Dasar Program Pascal

Pola susun bahasa Pascal dibentuk dengan menggunakan komponen bahasa pemrograman yang umum, yaitu :

1. Simbol Dasar
2. Reserved Word (kata pasti)
3. Identifier (penyebut)

1. Simbol Dasar.

Simbol dasar terdiri atas :

1. Simbol huruf, yaitu huruf A sampai dengan Z atau a sampai dengan z.
(huruf besar dan kecil).
2. Simbol angka atau digit yaitu : 0,1,2,3,4,5,6,7,8,9.
3. Simbol khusus, yaitu
`+ - * / ; := , ' = < > <= >= <> : { } () []`

2. Reserved Word (kata pasti)

Reserved Word adalah suatu kata yang secara mutlak tidak boleh diartikan lain dan harus digunakan sebagaimana yang telah didefinisikan atau ditentukan kegunaannya oleh bahasa Pascal. Reserved word ini tidak boleh didefinisikan ulang oleh pemakai, sehingga tidak dapat dipergunakan sebagai pengenalan (identifikasi)

Reserved Word ini jumlahnya berbeda untuk masing-masing bahasa Pascal.

Contoh beberapa reserved word yang telah didefinisikan oleh bahasa pascal antara lain

AND	ELSE	LABEL	SET
ARRAY	END	OF	TYPE
BEGIN	FUNCTION	OR	UNTIL
CASE	FOR	PROCEDURE	VAR
CONST	GOTO	PROGRAM	WHILE
DO	IF	RECORD	WITH
DOWNTON	IN	REPEAT	dsb

3. Identifier (sebutan/pengenal)

Identifier merupakan sebuah kata yang digunakan sebagai nama atau sebutan terhadap sesuatu didalam program. Pemakai dapat mendefinisikan sendiri suatu nama sebagai identifier.

Identifier ini terdiri atas :

- a. Identifier Standar, yaitu identifier yang telah didefinisikan oleh bahasa pascal.

Contoh dari Identifier standar ini antara lain:

ABS	LN
ARCTAN	ODB
BOOLEAN	PRED
CHAR	ROUND
CHR	READ
COS	READLN
EOF	SQR
EOLN	SQRT
EXP	SUCC

Dan masih banyak lagi.

- b. Identifier Non Standar; yaitu identifier yang didefinisikan oleh pemakai bahasa pascal; misalnya;
- c. nama suatu program
- d. nama suatu konstanta
- e. nama suatu variabel
- f. nama suatu procedure
- Identifier ini bebas, tetapi dengan ketentuan-ketentuan sebagai berikut :
- g. terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Huruf besar dan huruf kecil dianggap sama.
- h. Tidak boleh mengandung blank.
- i. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah.
- j. Panjangnya bebas, tetapi hanya 63 karakter pertama yang dianggap signifikan.

Contoh :

Identifier	Keterangan
GajiKaryawan	Benar
No_Mhs	Benar
P3K	Benar
1X	Salah, karakter pertama harus huruf
A&B	Salah, tidak boleh mengandung simbol khusus
A B	Salah, tidak boleh mengandung blank

3.4 Jenis-jenis data

Jenis – jenis data yang dikenal dalam bahasa pascal antara lain yaitu:

1. Jenis data sederhana
 - a. Jenis data yang standar; yaitu :
 - Integer
 - Real
 - Karakter
 - Boolean
 - b. Jenis data yang non standar (user defined), yaitu;
 - Enumerated
 - Sub-range
2. Jenis data berstruktur, yaitu:
 - a. Array
 - b. Record
 - c. Set
 - d. File
3. Jenis Data Pointer

INTEGER

Jenis data ini terdiri atas integer positif, integer negatif dan nol. Merupakan nilai bilangan bulat.

Pada TURBO PASCAL jenis data ini di bagi atas beberapa bagian, yaitu :

Tipe	Ukuran memori (dalam byte)	Jangkauan nilai
BYTE	1	0..255
SHORTINT	1	-128..127
INTEGER	2	-32768..32767
WORD	2	0..65535
LONGINT	4	-2147483648..2147483647

Operator Integer terdiri atas : + , - , * , / , DIV dan MOD

Contoh :

Var

Jumlah : byte;

Begin

Jumlah := 200;

WriteLn('Nilai JUMLAH = ',Jumlah);

End.

Hasilnya bila dijalankan :

Nilai JUMLAH = 200

REAL

Penulisan untuk jenis data ini selalu menggunakan titik desimal. Nilai konstanta numerik real berkisar dari 1E-38 sampai dengan 1E+38 dengan mantissa yang signifikan sampai dengan 11 digit. E menunjukkan nilai 10 pangkat. Nilai konstanta numerik real menempati memori sebesar 6 byte.

Contoh :

123.45

12345. → salah, titik desimal tidak boleh dibelakang

12E5

12E+5

-12.34

.1234 → salah, titik desimal tidak boleh dimuka

Pada TURBO PASCAL, jenis data ini dibedakan atas :

Type	Ukuran memori (dalam byte)	Jangkauan nilai	Digit signifikan
SINGLE	4	$1.5 \times 10^{-45} .. 3.4 \times 10^{38}$	7-8
DOUBLE	8	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$	15-16
EXTENDED	10	$1.9 \times 10^{-4951} .. 1.1 \times 10^{4932}$	19-20
COMP	8	$-2E+63+1 .. 2E+63-1$	19-20

Operator untuk jenis data ini terdiri atas : + , - , * dan /

KARAKTER

Yang dimaksud dengan jenis data ini adalah karakter tunggal atau sebuah karakter yang ditulis diantara tanda petik tunggal, seperti misalnya 'A','a','!', '5' dsb.

Dasarnya adalah ASCII CHARACTER SET.

Misalnya : 032 pada tabel ASCII CHARACTER SET menunjukkan karakter.

	Blank.	
033	_____	!
048	_____	0
076	_____	L
	dst	

STRING

Nilai data string merupakan urutan-urutan dari karakter yang terletak di antara tanda petik tunggal. Nilai data string akan menempati memori sebesar banyaknya karakter stringnya ditambah dengan 1 byte. Bila panjang dari suatu string di dalam deklarasi variabel tidak disebutkan, maka dianggap panjangnya adalah 255 karakter.

Contoh :

Var

Kampus : string[10];

Begin

Kampus := 'Gunadarma';

Write(Kampus);

End.

BOOLEAN

Jenis data ini mempunyai nilai TRUE atau FALSE.

Operator untuk jenis data ini adalah :

1. Logical Operator, yaitu : NOT, AND dan OR
2. Relational Operator, yaitu : >, <, >=, <=, <> dan =

JENIS DATA NON-STANDARD (USER DEFINED)

1. ENUMERATED.

Jenis data ini terdiri atas barisan identifier yang terurut dimana setiap identifier tersebut dianggap sebagai suatu individual data item (elemen data yang berdiri sendiri).

Pada saat mendeklarasikan jenis data ini kita harus menuliskan semua elemen-elemennya.

Bentuk umum deklarasinya adalah :

TYPE nama = (data_item_1, data_item_2,, data_item_n);

Contoh :

TYPE hari = (sen, sel, rab, kam, jum, sab, ming);

TYPE warna = (red, blue, green, yellow, black, white);

Setelah jenis data ini dideklarasikan, maka selanjutnya kita dapat mendeklarasikan suatu variabel yang berjenis data sama dengan jenis data ini.

Misalnya :

```
TYPE nama_hari = (sen, sel, rab, kam, jum, sab, ming);  
VAR libur : nama_hari;
```

Fungsi standar yang dapat digunakan pada jenis data ini adalah :

PRED, SUCC dan ORD

Misalnya :

```
PRED (sel) = sen  
SUCC (sen) = sel  
ORD (sen) = 0  
ORD (sel) = 1
```

2. SUB-RANGE.

Jenis data ini berupa range dari suatu kumpulan data yang mempunyai urutan..

Bentuk umum deklarasinya adalah :

TYPE nama = data_item_pertama .. data_item_terakhir;

Contoh :

- a. TYPE jam_kuliah = 1 .. 10;
tanggal = 1 .. 31;
abjad = 'A' .. 'Z';

b. TYPE bulan = (jan,feb,mar,apr,mei,jun,jul,agt,sep,okt,nov,des);
 hari = (sen,sel,rab,kam,jum,sab,ming);
 ata = agt .. jan;
 pta = feb .. jun;
 hari_kerja = sen .. jum;

TANDA OPERASI

Tanda operasi (operator) di dalam bahasa Pascal dikelompokkan ke dalam 9 kategori, yaitu

1. Assignment operator.
2. Binary operator.
3. Unary operator.
4. Bitwise operator.
5. Relational operator.
6. Logical operator.
7. Address operator.
8. Set operator.
9. String operator.

Assignment operator

Assignment operator (operator pengerjaan) menggunakan simbol titik dua diikuti oleh tanda sama dengan (:=).

Contoh :

A:=B;

Binary operator

Digunakan untuk mengoperasikan dua buah operand. Operand dapat berbentuk konstanta ataupun variabel. Operator ini digunakan untuk operasi aritmatika yang berhubungan dengan nilai tipe data integer dan real.

OPerator	OPerasi	Tipe Operand	Tipe hasil
*	Perkalian	real,real integer,integer real,integer	real integer real
DIV	Pembagian bulat	integer,integer	integer
/	Pembagian real	real,real integer,integer real,integer	real real real
MOD	Sisa pembagian	integer,integer	integer
+	Pertambahan	real,real integer,integer real,integer	real integer real
-	pengurangan	real,real integer,integer real,real	real integer real

Contoh :

15*5	hasilnya	75
20/3	hasilnya	6.6666666667E+00
20 div 3	hasilnya	6
20 mod 3	hasilnya	2

Unary operator

Operator ini hanya menggunakan sebuah operand saja. Dapat berupa unary minus dan unary plus. Unary minus digunakan untuk menunjukkan nilai negatif, baik pada operand numerik real maupun integer. Unary plus adalah operator untuk memberai tanda plus.

Contoh :

```
-5      +7
-2.5    +2.5
```

Bitwise operator

Digunakan untuk operasi bit per bit pada nilai integer. Terdiri dari operator NOT, AND, OR, XOR, Shl, Shr.

Relational operator

Relational operator digunakan untuk membandingkan hubungan antara dua buah operand dan akan didapatkan hasil tipe boolean, yaitu True atau False. Terdiri dari operator : =, <, >, <=, >=, <>

Logical operator

Terdapat 4 buah logical operator yaitu : NOT, AND, OR dan XOR. Operator ini bekerja dengan nilai-nilai logika, yaitu True dan False.

Set operator

Digunakan untuk operasi himpunan.

String operator

Digunakan untuk operasi string. Hanya ada sebuah operator string saja, yaitu operator + yang digunakan untuk menggabungkan dua buah nilai string.

Contoh :

```
Nama1 := 'Arief';
Nama2 := 'Kurniawan';
Nama3 := Nama1 + Nama2;
```

3.5 Konstanta, Variabel dan Ekspresi.

Konstanta.

Konstanta adalah suatu identifier non-standar yang nilainya telah ditetapkan dalam suatu program dan dideklarasikan pada bagian deklarasi.

Bagian umum deklarasinya adalah :

```
CONST identifier = nilai;
```

Contoh :

```
CONST phi = 3.14;
      x = 2;
      y = 'S';
```


Variabel.

Variabel adalah suatu identifier non-standar yang nilainya tidak tetap atau nilainya merupakan hasil dari suatu proses.

(variabel yang dimaksudkan disini sama seperti halnya arti variabel pada aljabar).

Bentuk umum deklarasinya adalah :

1. Jika hanya 1 (satu) variabel yang dideklarasikan :
VAR identifier : jenis data;
2. Jika lebih dari 1 (satu) variabel dan masing-masing memiliki jenis data yang sama :
VAR id-1, id-2, ..., id-3 : jenis data;
3. Jika beberapa variabel yang berbeda jenis datanya :
VAR identifier-1 : jenis data;
 identifier-2 : jenis data;
 .
 .
 identifier-n : jenis data;
4. Jika variabel yang dideklarasikan berjenis data non-standar :

TYPE id = (data_item_1,data_item_2,...,data_item_n);
VAR id-v : id;

SUB-RANGE :

TYPE id = data_item_pertama .. data_item_terakhir;
VAR id-v : id;

Contoh :

1. VAR x : integer;
2. VAR p,q,r : real;
3. VAR a,b : char;
 m,n : boolean;
4. TYPE warna = (merah,hitam,biru,putih,kuning);
 VAR x1,x2 : warna;
5. TYPE tahun = 1900 .. 1993;
 VAR thn-1,thn-2,thn-3 : tahun;

Ekspresi.

Sebuah ekspresi merupakan kumpulan dari operand-operand (seperti : bilangan, konstanta, variabel dll) yang bersama-sama dengan operator membentuk suatu bentuk aljabar dan menyatakan suatu nilai.

Ada 2 (dua) jenis ekspresi dalam bahasa Pascal, yaitu :

1. Ekspresi numerik / aritmatika, yaitu suatu ekspresi yang menghasilkan nilai numerik / aritmatika.
2. Ekspresi Boolean atau ekspresi logika, yaitu suatu ekspresi yang menghasilkan nilai boolean / logika (true/false).

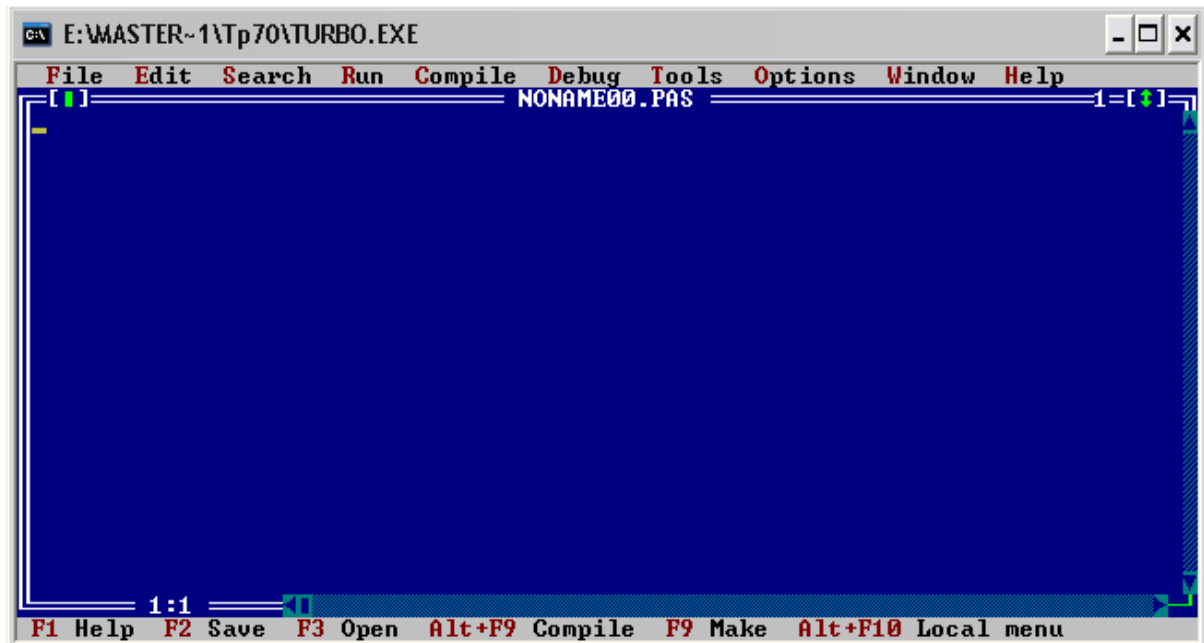
Contoh :

1. $(b * b - 4 * a * c) / (2 * a) / (2 * a) \Rightarrow$ ekspresi numerik, jika a,b dan c adalah bilangan (variabel bernilai numerik).
2. $Upah < 1000.0 \Rightarrow$ ekspresi boolean ("upah" adalah suatu variabel bernilai real).

3.6 Bekerja Dengan Turbo Pscal

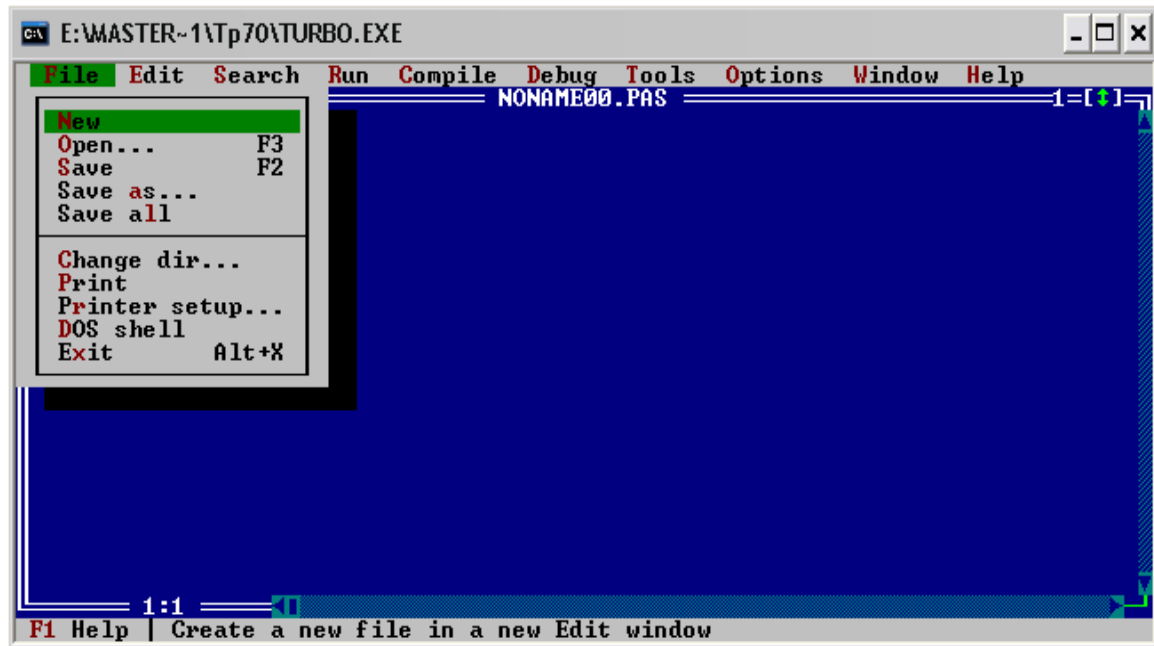
Untuk dapat menggunakan bahasa pemrograman Turbo Pascal, maka terlebih dahulu memanggil sistem dengan cara sebagai berikut:

1. Cari folder yang bernama TP atau TP70 (klik 2 kali)
2. Cari file dengan nama TURBO (klik 2 kali) dan tunggu beberapa saat sampai ditampilkan window (jendela) seperti terlihat pada gambar 8



Gambar 8 : Jendela Kerja Turbo Pascal

3. Saat sudah berada pada tampilan seperti gambar, maka dikatakan sudah siap untuk bekerja dengan Turbo Pascal. Gunakanlah menu-menu yang tersedia untuk memberikan instruksi tertentu sesuai kebutuhan.
4. Untuk menampilkan masing-masing pilihan menu dapat dilakukan dengan cara menekan tombol ALT bersama-sama dengan tombol huruf awal menu atau bisa juga dengan cara mengklikkan mouse ke menu yang bersangkutan. Contoh: untuk menampilkan menu File dapat dilakukan dengan cara menekan tombol ALT bersama-sama dengan tombol F, seperti terlihat pada gambar 9. Untuk membatalkan tampilan menu gunakan tombol Esc.
5. Jika tampilan window (jendela kerja dirasa kurang besar, dapat diperbesar dengan cara menekan tombol ALT bersama-sama dengan tombol ENTER. Begitu juga sebaliknya.



Gambar 9 : Pilihan Pada Menu File

3.6.1. Mengganti Directory

- Tampilkan menu File (ALT + F)
- Pilih menu Change Dir (sorot lalu tekan ENTER atau dengan mengklik)
- Ketikkan Directory yang diinginkan, misal :
D:\LA-14240001
- Setelah mengetikkan directory dengan benar, akhiri dengan menekan ENTER (catatan : folder LA-14240001 sudah dibuat sebelumnya. Jika belum dibuat, maka akan ditampilkan pesan kesalahan)

3.6.2. Membuat File/Program Baru

Sebenarnya saat pertama kali memanggil aplikasi Turbo Pascal, secara otomatis sudah diberikan sebuah file kosong yang siap untuk digunakan untuk menulis program. Namun bila setelah selesai membuat suatu program dan ingin membuat program yang baru, maka dapat melakukan langkah sebagai berikut:

- Tampilkan menu File (ALT + F)
- Pilih menu NEW (sorot lalu tekan ENTER atau dengan mengklik)
- Berikut akan ditampilkan lembar editor program yang siap untuk digunakan.

Untuk menutup layar editor program, dapat dilakukan dengan mengklik tanda kotak hijau kecil yang ada di sebelah kiri bagian atas.

3.6.3. Mengcompile Program

Setelah selesai menulis program, ada baiknya untuk melakukan proses kompilasi untuk memastikan apakah program sudah benar atau tidak ada kesalahan lagi. Untuk melakukan proses kompilasi lakukan dengan cara menekan tombol ALT bersama-sama dengan huruf C. Setelah itu pilih Compile atau bisa langsung menekan tombol ALT+F9.

3.6.4. Menyimpan File/Program

Untuk menyimpan program yang telah dibuat dapat dilakukan cara:

- Tampilkan menu File (ALT + F)
- Pilih menu SAVE (sorot lalu tekan ENTER atau dengan mengklik)
- Ketik nama file yang akan diberikan, kemudian dilanjutkan dengan menekan tombol ENTER atau mengklik tombol OK.

3.6.5. Menjalankan Program

Untuk menjalankan program yang sudah dibuat dapat dilakukan dengan cara menekan tombol **ALT** bersama-sama dengan tombol **R** (huruf awal menu Run). Setelah itu pilih menu **Run**, atau dengan cara mengklik.

3.6.6. Memanggil File/Program

Untuk memanggil file yang pernah disimpan, dapat dilakukan dengan cara:

- a. Tampilkan menu File (ALT + F)
- b. Pilih menu Open (sorot lalu tekan ENTER atau dengan mengklik)
- c. Saat ditampilkan kotak dialog "Open a File", tekan tombol Tab untuk memindahkan kursor ke daftar pilihan file.
- d. Sorotlah nama file yang akan dibuka, lalu saat penunjuk berada pada file tersebut, tekan tombol ENTER, atau dapat juga dengan mengklik tombol OK

2.8 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal seperti terlihat pada gambar 8. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

1. Program Pertama

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

Program Pertama;

Begin

```
Writeln('=====');  
Writeln('Hallo Dunia.....');  
Writeln;  
Writeln('Salam kenal, nama saya Budi');  
Writeln('Saya kuliah di UNRIYO.....');  
writeln('=====');  
Readln;
```

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER) atau tekan tombol fungsi F2, ketikkan nama program yaitu : Lat1, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

2. Program Kedua

Program Contoh_Konstanta;

Const

```
Discount = 0.20;  
Beli = 200000;  
Toko = 'PT. UNRIYO JAYA';
```

Begin

```
Writeln('Discount : ',Discount:4:2);  
Writeln('Pembelian      : ',Beli;  
Writeln('Nama Toko      : ',Toko);  
Readln;
```

end.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat2, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

3. Program Ketiga

```
Program Contoh_Variabel;  
Var  
  TotBayar, Beli, Discount : Real;  
  Keterangan : String[25];  
Begin  
  Beli := 150000;  
  Discount := 0.20 * Beli;  
  TotBayar := Beli - Discount;  
  Keterangan := 'Pelanggan';  
  Writeln('Pembelian           : Rp ',Beli:6:2);  
  Writeln('Discount           : Rp ',Discount:6:2);  
  Writeln('Total Pembayaran   : Rp ',TotBayar:6:2);  
  Writeln('Keterangan          : ',Keterangan);  
  Readln;  
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat3, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

4. Program Keempat

```
Program Perkalian;  
Var  
  A, B, C : Integer;  
Begin  
  Write('Masukan Nilai A   : ');  
  Readln(A);  
  Write('Masukan Naiali B   : ');  
  Readln(B);  
  C := A * B;  
  Writeln('Hasil A * B       : ',C);  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat4, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

5. Program Kelima

```
Program Gaji;  
Uses crt;  
Const  
    Pti=0.10;  
    Pta=0.05;  
Var  
    Nama,Gol,Jabatan : String;  
    Gapok, Tjis, Tjan, JmlGaji : real;  
Begin  
    clrscr;  
    Write('Nama Pegawai   : ');  
    Readln>Nama);  
    Write('Golongan       : ');  
    Readln(Gol);  
    Write('Jabatan        : ');  
    Readln(Jabatan);  
    Write('Gaji Pokok      : ');  
    Readln(Gapok);  
    Tjis := Gapok * Pti;  
    Tjan := Gapok * Pta;  
    JmlGaji := Gapok+Tjis+Tjan;  
    Writeln('Tunjangan Istri : ',Tjis:8:2);  
    Writeln('Tunjangan Anak  : ',Tjan:8:2);  
    Writeln('Jumlah Gaji    : ',JmlGaji:8:2);  
    readln;  
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat5, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Struktur Program dan Statemen Input-Output

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang struktur program dan perintah-perintah input dan output untuk membuat aplikasi dengan Turbo Pascal.

Pertemuan ini akan menjelaskan secara singkat tentang konsep pemrograman Turbo Pascal dan lingkungannya dan konsep-konsep penting dalam membangun sebuah aplikasi dengan menggunakan Turbo Pascal.

4.1 Struktur Dan Komponen Dasar Program Pascal.

Struktur dari suatu program Pascal terdiri dari sebuah judul program dan suatu blok program atau badan program. Blok program dibagi lagi menjadi dua bagian, yaitu bagian deklarasi dan bagian pernyataan (statement).

4.1.1 Struktur program :

Judul Program	PROGRAM nama-program;
Blok Program	
Bagian deklarasi	
- deklarasi label	LABEL nama-label;
- deklarasi konstanta	CONST.....;
- deklarasi tipe	TYPE;
- deklarasi variabel	VAR;
- deklarasi prosedur	PROCEDURE nama-prosedur;
;
- deklarasi fungsi	FUNCTION nama-fungsi;
;
Bagian Pernyataan	
Begin	
(statement)	
.....;	
.....;	
end.	

Contoh : Menghitung luas bidang berbentuk empat persegi panjang dengan panjang P dan lebar L.

PROGRAM Luas;	{Judul}
VAR P,L,Luas : real;	{Deklarasi variabel}
BEGIN	
Read(P);	{Statement}
Read(L);	{Statement}
Luas := P*L;	{Statement}
Write(P,L,Luas);	{Statement}
END.	

Judul program sifatnya adalah optional, dan bila ditulis, harus terletak pada awal dari program dan diakhiri dengan titik koma.

Bagian deklarasi digunakan bila di dalam program digunakan pengenalan (identifikasi). Identifikasi dapat berupa label, konstanta, tipe, variabel, prosedur dan fungsi. Kalau suatu program menggunakan identifikasi, Pascal menuntut supaya identifikasi tersebut diperkenalkan terlebih dahulu sebelum digunakan, yaitu dideklarasikan terlebih dahulu pada bagian ini.

Beberapa aturan dalam program Pascal :

- Akhir sebuah program Pascal ditandai dengan tanda baca titik (.) setelah END yang paling akhir.
- Tanda titik koma (;) merupakan pemisah antar instruksi satu dengan lainnya.
- Beberapa statement boleh ditulis menjadi satu baris dipisahkan dengan tanda baca titik koma (;)
Contoh : `simpan := akhir; akhir := simpan + awal;`
- Baris komentar diletakkan diantara tanda (* dan *) atau diantara tanda { dan }
Contoh : `Var rata : real; (*nilai rata-rata*)
Nil1 : real; {nilai ujian}`

Statement (pernyataan)

Adalah instruksi atau gabungan instruksi, yang menyebabkan komputer melakukan aksi.

Type statement dalam Pascal terdiri atas :

3. Sederhana :
 - menandai sebuah item data ke sebuah variabel (assignment statement)
contoh : `X := Y * 4.135`
 - pemanggilan procedure dan goto statement
4. Terstruktur:
 - Compound Statement
contoh : `Begin
 read (x) ;
 y := x + 5;
 write (y)
End.`
 - Repetitive Statement
contoh : `For I := 1 to 100 do
 write (count);`
 - Conditional Statement
contoh : `If x > 100 then write (s)
 else write (p) ;`

4.2 Input dan output (Standar I/O).

Dalam bahasa Pascal untuk keperluan input (membaca input) digunakan identifier standar READ atau READLN.

Identifier standar ReadLn sedikit berbeda dengan Read. ReadLn digunakan untuk memasukkan data perbaris, artinya setelah tombol Enter ditekan, maka akan ganti baris, sedangkan Read tidak ganti baris, masih dalam baris yang sama.

Sedangkan untuk keperluan output (mencetak output) digunakan identifier standar WRITE atau WRITLN.

Perbedaan Write dengan WriteLn adalah bahwa Write menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama. Sedang WriteLn digunakan untuk menampilkan tampilan perbaris, akan ganti baris untuk tampilan berikutnya.

Bentuk umum penulisannya :

1. READ (variabel input) ;

Atau

READLN (variabel input);

2. WRITE (variabel output);

Atau

WRITELN (variabel output);

Contoh :

```
Var  nama : string{15};
Begin nama := 'Dewi';
      write('nama : ');
      write(nama);
End.
```

Hasilnya :
Nama : Dewi

Contoh :

```
Var  nama : string{15};
Begin nama := 'Dewi';
      WriteLn('Nama : ');
      WriteLn(Nama);
End.
```

Hasilnya :
Nama :
Dewi

4.3 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal seperti terlihat pada gambar 8. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

6. Program Keenam

```
Program Bilpositif1;  
Uses crt;  
Var  
  Bil : Integer;  
Begin  
  Clrscr;  
  Write('Masukkan sebuah bilangan : ');  
  Readln(Bil);  
  If Bil > 0 Then  
    Writeln('Bilangan Positif');  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat6, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

7. Program Ketujuh

```
Program Bilpositif2;  
Uses crt;  
Var  
  Bil : Integer;  
Begin  
  Clrscr;  
  Write('Masukkan sebuah bilangan : ');  
  Readln(Bil);  
  If Bil > 0 Then  
    Writeln('Bilangan Positif')  
  Else  
    Writeln('Bilangan Negatif');  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat7, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

8. Program Kedelapan

```
Program Nilai1;  
Uses Crt;  
Var  
  NA : Integer;  
  Angka : Char;  
Begin  
  clrscr;  
  Write('Masukan Nilai Akhir      : ');  
  Readln(NA);  
  If NA<=45 Then  
    Angka :='E'  
  Else If NA<=55 Then  
    Angka :='D'  
  Else If NA<=70 Then  
    Angka :='C'  
  Else If NA<80 Then  
    Angka :='B'  
  Else  
    Angka :='A';  
  Writeln('Nilai Angka Anda adalah  : ',Angka);  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat8, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

9. Program Kesembilan

```
Program Nilai2;  
Uses Crt;  
Var  
  NA : Integer;  
  Angka : Char;  
Begin  
  clrscr;  
  Write('Masukan Nilai Akhir      : ');  
  Readln(NA);  
  Case NA of  
    0 .. 45 : Angka :='E';  
    46 .. 55 : Angka :='D';  
    56 .. 70 : Angka :='C';  
    71 .. 79 : Angka :='B';  
    80 .. 100 : Angka :='A';  
  End;  
  Writeln('Nilai Angka Anda adalah  : ',Angka);  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat9, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Struktur Program dan Statemen Input-Output Lanjutan

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang struktur program dan statemen input-output secara lebih lanjut (mendalam) untuk membuat aplikasi dengan Turbo Pascal.

Pertemuan ini akan menjelaskan secara singkat tentang tinjauan khusus mengenai operator logika agar dapat memecahkan masalah dengan berbagai kondisi.

5.1 Tinjauan Khusus Operator Logika

Operator logika digunakan untuk melakukan operasi-operasi yang menghasilkan nilai logik (true dan false). Adapun operand dalam operasi ini juga bertipe logik. Bahasa Pascal menyediakan empat buah operator logika yaitu seperti yang tampak pada tabel berikut:

Operator	Jenis Operasi	Tipe Operand	Tipe Hasil	Contoh
NOT	Negasi	Boolean	Boolean	Not (x=y)
AND	Conjunction	Boolean	Boolean	(x>1) and (x<=5)
OR	Disjunction	Boolean	Boolean	(x<1) or (x>5)
XOR	Exclusive Disjunction	Boolean	Boolean	A xor B

5.1.1 Operator NOT

Operator ini digunakan untuk menentukan negasi dari nilai logik lain. Apabila kita melakukan negasi pada nilai true, maka nilai yang dihasilkan adalah false. Apabila kita melakukan negasi terhadap nilai false, maka nilai yang akan dihasilkan adalah true. Berikut ini tabel yang menunjukkan hubungan operator not.

A	Not A
True	False
False	True

Contoh Program penggunaan operator NOT

```

Program OperatorNot;
Uses
  Crt;

Var
  A : boolean;

Begin
  Clrscr;
  A := false;
  A := not A;
  Writeln(A);
  Readln;

```

Adapun hasil yang akan diberikan dari program tersebut adalah sebagai berikut:

True

5.1.2 Operator AND

Untuk memudahkan penggunaan operator and, ingatlah bahwa operator and hanya akan menghasilkan nilai true apabila semua operand-nya bernilai true, selain itu operasi akan menghasilkan nilai false. Perhatikan tabel hubungan operator and berikut:

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

Contoh Program penggunaan operator AND

```
Program OperatorAnd;  
Uses  
  Crt;  
  
Begin  
  Clrscr;  
  Writeln('True AND True   = ', true and true);  
  Writeln('True AND False  = ', true and false);  
  Writeln('False AND True   = ', false and true);  
  Writeln('False AND False  = ', false and false);  
  Readln;  
End.
```

Adapun hasil yang akan diberikan dari program tersebut adalah sebagai berikut:

```
True AND True   = true  
True AND False  = false  
False AND True  = false  
False AND False = false
```

5.1.3 Operator OR

Operator or akan menghasilkan nilai true apabila satu atau semua operand-nya bernilai true. Namun untuk memudahkan, ingatlah bahwa operator or akan menghasilkan nilai false apabila semua operand-nya bernilai false, selain itu operator ini akan menghasilkan nilai true. Perhatikan tabel hubungan operator or di bawah ini:

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

Contoh Program penggunaan operator OR

```
Program OperatorOr;  
Uses  
Crt;  
  
Begin  
Clrscr;  
Writeln('True OR True    = ', true or true);  
Writeln('True OR False   = ', true or false);  
Writeln('False OR True    = ', false or true);  
Writeln('False OR False   = ', false or false);  
Readln;  
End.
```

Adapun hasil yang akan diberikan dari program tersebut adalah sebagai berikut:

```
True OR True      = true  
True OR False     = true  
False OR True     = true  
False OR False    = false
```

5.1.4 Operator XOR

Berbeda dengan operator or, exclusive or (xor) ini akan menghasilkan nilai true apabila hanya terdapat satu operand yang bernilai true. Apabila kedua operand-nya bernilai true maka operasi akan menghasilkan nilai false, begitu juga apabila keduanya bernilai false. Perhatikan

A	B	A xor B
True	True	False
True	False	True
False	True	True
False	False	False

Contoh Program penggunaan operator XOR

```
Program OperatorXor;  
Uses  
Crt;  
  
Begin  
Clrscr;  
Writeln('True XOR True    = ', true xor true);  
Writeln('True XOR False   = ', true xor false);  
Writeln('False XOR True    = ', false xor true);  
Writeln('False XOR False   = ', false xor false);  
Readln;  
End.
```

Adapun hasil yang akan diberikan dari program tersebut adalah sebagai berikut:

```
True XOR True     = false  
True XOR False    = true  
False XOR True    = true  
False XOR False   = false
```

5.2 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal seperti terlihat pada gambar 8. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

10. Program Kesepuluh

```
Program Perulangan1;  
Uses Crt;  
Var  
  I : Integer;  
Begin  
  Clrscr;  
  For I :=1 to 10 Do  
    Begin  
      Writeln(I);  
    end;  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat10, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

11. Program Kesebelas

```
Program Perulangan2;  
Uses Crt;  
Var  
  I : Integer;  
Begin  
  Clrscr;  
  For I :=10 DownTo 1 Do  
    Begin  
      Writeln(I);  
    end;  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat11, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

12. Program Keduabelas

```
Program Perulangan3;  
Uses Crt;  
Var  
  I : Integer;  
Begin  
  Clrscr;  
  I := 1;  
  While I<=10 Do  
    Begin  
      Writeln(I);  
      I :=I + 1;  
    end;  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat12, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

13. Program Ketigabelas

```
Program Perulangan4;  
Uses Crt;  
Var  
  I : Integer;  
Begin  
  Clrscr;  
  I := 1;  
  Repeat  
    Writeln(I);  
    I :=I + 1;  
  Until I>10;  
  Readln;  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat13, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

14. Program Keempatbelas

```
Program Perulangan5;  
Uses Crt;  
Var  
  N, I, Jum : Integer;  
  Rata : Real;  
  Lagi : Char;  
Begin  
  Clrscr;  
  Lagi := 'Y';  
  While Lagi = 'Y' DO  
    Begin  
      Clrscr;
```

```

Write('Banyak Data : ');
Readln(N);
Begin
Jum :=0;
I :=1;
While I <= N DO
Begin
Jum := Jum + I;
I := I + 1;
End;
Rata := Jum/N;
Writeln('Jumlah   : ',Jum);
Writeln('Rata     : ',Rata:5:2);
Writeln;
Write('Ingin menghitung lagi (Y/T) ? : ');
Readln(Lagi);
End;
End;
Readln;
End.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat14, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

15. Program Kelimabelas

```

Program Perulangan6;
Uses Crt;
Var
  A, B, C : Integer;
Begin
  Clrscr;
  Writeln('=====');
  Writeln('Nilai A   Nilai B   Nilai C   ');
  Writeln('-----');
  A := 1;
  Repeat
    B :=0;
    Repeat
      C := A * A + B * B;
      Writeln(A:7, B:9, C:9);
      B :=B + 5;
    Until B > 25;
    A := A + 1;
  Until A > 3;
  Writeln('=====');
  Readln;
End.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat15, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Pertemuan 6

Pernyataan Bercabang

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan struktur bercabang pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang pernyataan bercabang yang menggunakan beberapa perintah yaitu Pernyataan If, dan Pernyataan Case-Of. Tentu saja pemilihan pernyataan yang digunakan dalam pemecahan masalah disesuaikan dengan tingkat kompleksitas permasalahan dan kebiasaan pemrogram.

6.1 Pernyataan If

Pernyataan If ada yang sederhana, dan ada pula yang lebih kompleks.

1. Pernyataan If-Then

Bentuk Umum Perintah:

.
. .
.

IF <Ekspresi> Then Pernyataan;

.
. .
.

Atau dapat juga ditulis:

.
. .
.

**IF <Ekspresi> Then
Pernyataan;**

.
. .
.

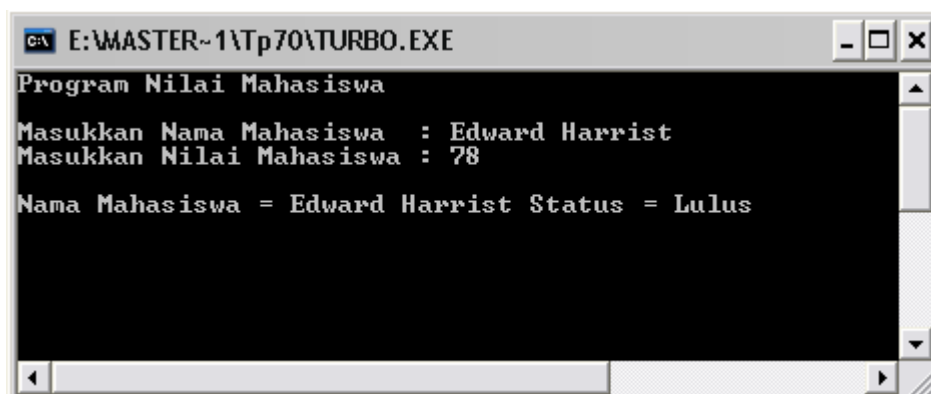
Contoh1:

```
Program lat8-1;  
uses crt;  
var nilai : integer;  
    mhs, ket : string;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    ket:='Tidak Lulus';  
    if nilai>=60 then ket:='Lulus';  
    writeln;  
    writeln('Nama Mahasiswa = ',mhs,' Status = ',ket);  
    readln;  
    end.
```

Contoh 2:

```
Program lat8-2;  
uses crt;  
var nilai : integer;  
    mhs, ket : string;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    if nilai>=60 then  
        ket:='Lulus';  
    if nilai<60 then  
        ket:='Tidak Lulus';  
    writeln;  
    writeln('Nama Mahasiswa = ',mhs,' Status = ',ket);  
    readln;  
    end.
```

Hasil bila program dijalankan adalah:



The screenshot shows a window titled "E:\MASTER~1\Tp70\TURBO.EXE". The program output is as follows:

```
Program Nilai Mahasiswa  
Masukkan Nama Mahasiswa : Edward Harrist  
Masukkan Nilai Mahasiswa : 78  
  
Nama Mahasiswa = Edward Harrist Status = Lulus
```

Contoh 3:

```
Program lat8_5;  
uses crt;  
var nilai : integer;  
    mhs : string;  
    p : char;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    if (nilai <=100) and (nilai >= 85) then p:='A';  
    if (nilai < 85) and (nilai >= 75) then p:='B';  
    if (nilai < 75) and (nilai >= 60) then p:='C';  
    if (nilai < 60) and (nilai >= 50) then p:='D';  
    if (nilai < 50) and (nilai >= 0) then p:='E';  
    writeln('Nama Mahasiswa = ',mhs,' Nilainya = ',p);  
    readln;  
end.
```

2. Pernyataan If-Then-Else

Bentuk Umum Perintah:

```
.  
. .  
. .  
IF <Ekspresi> Then Pernyataan1 Else Pernyataan2;  
. .  
. .
```

Atau dapat juga ditulis:

```
.  
. .  
. .  
IF <Ekspresi> Then  
Pernyataan1  
Else  
Pernyataan2;  
. .  
. .
```

Contoh 1: If-Then-Else, bentuk pertama

```
Program lat8_2;  
uses crt;  
var nilai : integer;  
    mhs, ket : string;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    if nilai >= 60 then ket:='Lulus' else ket:='Tidak Lulus';  
    writeln;  
    writeln('Nama Mahasiswa = ',mhs,' Status = ',ket);  
    readln;  
    end.
```

Contoh 2: If-Then-Else, bentuk kedua

```
Program lat8_3;  
uses crt;  
var nilai : integer;  
    mhs, ket : string;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    if nilai >= 60 then  
        ket:='Lulus'  
    else  
        ket:='Tidak Lulus';  
    writeln;  
    writeln('Nama Mahasiswa = ',mhs,' Status = ',ket);  
    readln;  
    end.
```

Contoh 3: If-Then-Else, bentuk lainnya

```
Program lat8_6;  
uses crt;  
var nilai : integer;  
    mhs : string;  
    p : char;  
Begin  
    clrscr;  
    writeln('Program Nilai Mahasiswa');  
    writeln;  
    write('Masukkan Nama Mahasiswa : ');  
    readln(mhs);  
    write('Masukkan Nilai Mahasiswa : ');  
    readln(nilai);  
    if (nilai <= 100) and (nilai >= 85) then p:='A' else
```

```
if (nilai < 85) and (nilai >= 75) then p:='B' else
  if (nilai < 75) and (nilai >= 60) then p:='C' else
    if (nilai < 60) and (nilai >= 50) then p:='D' else
      if (nilai < 50) and (nilai >= 0) then p:='E';
writeln('Nama Mahasiswa = ',mhs,' Nilainya = ',p);
readln;
end.
```

6.2 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

16. Program Keenambelas

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Tahun_Kabisat;  
uses crt;  
var thkabisat : boolean;  
    Th      : integer;  
Begin  
  clrscr;  
  writeln('Mengecek Tahun Kabisat');  
  Repeat  
    writeln;  
    write('Masukkan Tahun --> 0 = selesai : ');  
    readln(th);  
    if th>0 then  
      Begin  
        if th mod 100 = 0 then thkabisat := (th mod 400) = 0  
          else thkabisat := (th mod 4) = 0;  
        if thkabisat then writeln('Tahun ',th,' adalah tahun kabisat')  
          else writeln('Tahun ',th,' bukan tahun kabisat')  
      end  
    until th=0;  
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat16, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

17. Program Ketujuhbelas

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Piramida;  
uses crt;  
var i,j, tinggi, lok : byte;  
begin  
  clrscr;  
  write('Tinggi Piramida (2..10) : ');  
  repeat  
    gotoxy(26,1);clreol;  
    readln(tinggi);  
  until tinggi in [2..10];  
  for i:=1 to tinggi do  
    begin  
      lok:=40-i;  
      for j:=1 to i do  
        begin  
          gotoxy(lok,i);  
          write('*');  
          lok:=lok+2;  
        end;  
      end;  
    readln;  
  end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat17, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Pernyataan Bercabang Lanjutan

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan struktur bercabang pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang pernyataan bercabang yang menggunakan beberapa perintah yaitu Pernyataan If, dan Pernyataan Case-Of. Tentu saja pemilihan pernyataan yang digunakan dalam pemecahan masalah disesuaikan dengan tingkat kompleksitas permasalahan dan kebiasaan pemrogram.

7.1 Pernyataan Case-Of

Pernyataan Case-Of merupakan alternatif lain dari perintah untuk proses bercabang. Bentuk Umum Perintah:

.
. .
.

**Case <Ekspresi> of
<Konstanta> : Pernyataan
End;**

.
. .
.

Konstanta bisa berupa data konstanta numerik, konstanta karakter, atau konstanta dalam bentuk nilai rentangan. Sedangkan pernyataan bisa berupa pernyataan tunggal maupun pernyataan jamak.

Contoh 1: Case-of untuk konstanta numerik

```
Program lat8_10;  
uses crt;  
var nilai : integer;  
nama : string;  
Begin  
clrscr;  
writeln('Program Mencoba Pernyataan Case Of');  
writeln;  
write('Masukkan Nama Anda : ');  
readln(nama);  
write('Masukkan Nilai 1 - 10 : ');  
readln(nilai);  
case nilai of  
1 : writeln('Anda menekan angka satu ');  
2 : writeln('Anda menekan angka dua ');  
3 : writeln('Anda menekan angka tiga ');  
4 : writeln('Anda menekan angka empat ');  
5 : writeln('Anda menekan angka lima ');  
6 : writeln('Anda menekan angka enam ');  
7 : writeln('Anda menekan angka tujuh ');  
8 : writeln('Anda menekan angka delapan ');  
9 : writeln('Anda menekan angka sembilan ');
```

```

10 : writeln('Anda menekan angka sepuluh');
end;
writeln('Terima kasih atas partisipasi Anda...!');
readln;
end.

```

Contoh 2: Case-of untuk konstanta karakter

```

Program lat8_11;
uses crt;
var nilai : char;
    nama : string;
Begin
clrscr;
writeln('Program Mencoba Pernyataan Case Of');
writeln;
write('Masukkan Nama Anda : ');
readln(nama);
write('Masukkan Nilai A - E : ');
readln(nilai);
case nilai of
'A' : writeln('Anda menekan huruf A ');
'B' : writeln('Anda menekan huruf B ');
'C' : writeln('Anda menekan huruf C ');
'D' : writeln('Anda menekan huruf D ');
'E' : writeln('Anda menekan huruf E ');
end;
writeln('Terima kasih atas partisipasi Anda...!');
readln;
end.

```

Contoh 3: Case-of untuk konstanta karakter lainnya

```

Program lat8_12;
uses crt;
var nilai : char;
    nama : string;
Begin
clrscr;
writeln('Program Mencoba Pernyataan Case Of');
writeln;
write('Masukkan Nama Anda : ');
readln(nama);
write('Masukkan Nilai A - E : ');
readln(nilai);
case nilai of
'A','a' : writeln('Anda menekan huruf A ');
'B','b' : writeln('Anda menekan huruf B ');
'C','c' : writeln('Anda menekan huruf C ');
'D','d' : writeln('Anda menekan huruf D ');
'E','e' : writeln('Anda menekan huruf E ');
end;
writeln('Terima kasih atas partisipasi Anda...!');
readln;
end.

```

Contoh 4: Case-of untuk pernyataan jamak

```
Program lat8_13;  
uses crt;  
var nilai : char;  
    nama : string;  
Begin  
clrscr;  
writeln('Program Mencoba Pernyataan Case Of');  
writeln;  
write('Masukkan Nama Anda : ');  
readln(nama);  
write('Masukkan Nilai A - E : ');  
readln(nilai);  
writeln;  
case nilai of  
'A','a' :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai A ');  
    end;  
'B','b' :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai B ');  
    end;  
'C','c' :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai C ');  
    end;  
  
'D','d' :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai D ');  
    end;  
  
'E','e' :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai E ');  
    end;  
  
end;  
writeln('Terima kasih atas partisipasi Anda...!');  
readln;  
end.
```

Contoh 5: Case-of untuk konstanta dengan nilai bentangan

```
Program lat8_14;  
uses crt;  
var h : char;  
    nama : string;  
Begin  
clrscr;  
writeln('Program Mencoba Pernyataan Case Of');  
writeln;  
write('Masukkan Nama Anda      : ');  
readln(nama);  
write('Masukkan huruf atau angka ? : ');  
readln(h);  
writeln;  
case h of  
'A'..'Z' : writeln('Anda menekan huruf besar : ',h);  
'a'..'z' : writeln('Anda menekan huruf kecil : ',h);  
'0'..'9' : writeln('Anda menekan angka : ',h);  
end;  
writeln('Terima kasih atas partisipasi Anda...!');  
readln;  
end.
```

Contoh 6:

```
Program lat8_15;  
uses crt;  
var nilai : integer;  
    nama : string;  
Begin  
clrscr;  
writeln('Program Mencoba Pernyataan Case Of');  
writeln;  
write('Masukkan Nama Anda  : ');  
readln(nama);  
write('Masukkan Nilai 0 - 100 : ');  
readln(nilai);  
writeln;  
case nilai of  
85 .. 100 :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai A ');  
    end;  
75 .. 84 :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai B ');  
    end;  
60 .. 74 :  
    begin  
        writeln('Nama Anda : ',nama);  
        writeln('Mendapat nilai C ');  
    end;
```

```

50 .. 59 :
    begin
        writeln('Nama Anda : ',nama);
        writeln('Mendapat nilai D ');
    end;
0 .. 49 :
    begin
        writeln('Nama Anda : ',nama);
        writeln('Mendapat nilai E ');
    end;
end;
writeln('Terima kasih atas partisipasi Anda...!');
readln;
end.

```

Contoh 7:

```

Program lat8_16;
uses crt;
var h : char;
    nama : string;
Begin
clrscr;
writeln('Program Mencoba Pernyataan Case Of');
writeln;
write('Masukkan Nama Anda      : ');
readln(nama);
write('Masukkan huruf atau angka ? : ');
readln(h);
writeln;
case h of
'A'..'Z' : writeln('Anda menekan huruf besar : ',h);
'a'..'z' : writeln('Anda menekan huruf kecil : ',h);
'0'..'9' : writeln('Anda menekan angka : ',h);
else
writeln('Anda menekan tombol yang salah, coba lagi..!');
end;
writeln('Terima kasih atas partisipasi Anda...!');
readln;
end.

```

7.2 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

18. Program Delapanbelas

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Case_Of_Lanjut;
uses crt;
var pilih,jml,diskon          : integer;
    nmbrg                    : string;
    ulang                    : boolean;
    lagi                      : char;
    harsat, hrgbrg, hrgpajak, tothrg : real;
begin
  clrscr;
  ulang:=true;
  while ulang do
    begin
      clrscr;
      writeln('***** Menu Utama *****');
      writeln('=====');
      writeln('1. Hitung Harga Barang          ');
      writeln('2. Hitung Diskon Barang          ');
      writeln('3. Hitung Pajak Barang          ');
      writeln('4. Selesai                          ');
      writeln('=====');
      write('Pilih (1 - 4) : ');readln(pilih);
      writeln('=====');
      case pilih of
        1 : Begin
          clrscr;
          lagi:='Y';
          while upcase(lagi)='Y' do
            begin
              write('Nama Barang : ');
              readln(nmbrg);
              write('Jumlah Barang : ');
              readln(jml);
              write('Harga Satuan : ');
              readln(harsat);
              tothrg:=jml*harsat;
              writeln('Total Harga : ',tothrg:12:2);
              writeln;
              write('Mau isi lagi (Y/T) ? : ');
              readln(lagi);
            end;
          end;
        2 : Begin
          clrscr;
          lagi:='Y';
          while upcase(lagi)='Y' do
```

```

begin
  write('Nama Barang   : ');
  readln(nmbrg);
  write('Harga Satuan : ');
  readln(harsat);
  write('Diskon (%)    : ');
  readln(diskon);
  hrgbrg:=harsat-(harsat*(diskon/100));
  writeln('Harga Barang : ',hrgbrg:12:2);
  writeln;
  write('Mau isi lagi (Y/T) ? : ');
  readln(lagi);
  end;
end;
3 : Begin
  clrscr;
  lagi:='Y';
  while upcase(lagi)='Y' do
  begin
    write('Nama Barang           : ');
    readln(nmbrg);
    write('Harga Barang           : ');
    readln(hrgbrg);
    writeln('Kena Pajak 10 % ');
    hrgpajak:=10/100*hrgbrg;
    writeln('Harga Barang Setelah Pajak : ',hrgpajak:12:2);
    writeln;
    write('Mau isi lagi (Y/T) ? : ');
    readln(lagi);
    end;
  end;
4 : exit;
end;
end;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat18, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

19. Program Sembilanbelas

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program animasi_tulisan;
uses crt;
var i,j           : byte;
    pil           : char;
    kal           : string;
    pos, arah, x, y : byte;
begin
  clrscr;
  gotoxy(20,2);writeln('**** Program Animasi Huruf Berjalan ****');
  gotoxy(17,3);writeln('=====');
  gotoxy(23,5);write('Masukkan Kalimat : ');readln(kal);
  gotoxy(23,9);write('Ke Kanan (R) ');

```



```

gotoxy(23,10);write('Ke Kiri (L) ');
gotoxy(23,11);write('Ke Atas (U) ');
gotoxy(23,12);write('Ke Bawah (D) ');
gotoxy(23,13);write('Selesai (X) ');
gotoxy(17,14);write('=====
=====');
gotoxy(23,15);write('Pilihan Anda : ');
repeat
gotoxy(38,15);clreol;
pil:=upcase(readkey);
until (pil='R') or (pil='L') or (pil='U') or (pil='D') or (pil='X');
clrscr;
y:=15;
x:=(80-length(kal))div 2;
repeat
begin
if keypressed then pil:=upcase(readkey);
case pil of
'R' :
begin
inc(x);
if x=79-length(kal) then
x:=1;
end;
'L' :
begin
dec(x);
if x=2 then x:=79-length(kal)
end;
'U' :
begin
dec(y);
if y=1 then y:=25;
end;
'D' :
begin
inc(y);
if y=25 then y:=1;
end;

#0 :
begin
pil:=readkey;
case pil of
#72 :
begin
dec(y);
pil:='U';
if y=1 then y:=25;
end;
#75 :
begin
dec(x);
pil:='L';
if x=2 then x:=79-length(kal);
end;
#77 :
begin
inc(x);

```

```

        pil:='R';
        if x=79-length(kal) then x:=1;
    end;
#80 :
    begin
        inc(y);
        pil:='D';
        if y=25 then y:=1;
    end;
end;
end;
end;
'X': exit;
end;
gotoxy(x,y);write(kal);delay(100);
gotoxy(x,y);
for i:=1 to length(kal) do
write(' ');
end;
until (pil='X');
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat19, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

20. Program Duapuluh

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program Piramida;
uses crt;
var i,j, tinggi, lok : byte;
begin
    clrscr;
    write('Tinggi Piramida (2..10) : ');
    repeat
        gotoxy(26,1);clreol;
        readln(tinggi);
    until tinggi in [2..10];
    for i:=1 to tinggi do
        begin
            lok:=40-i;
            for j:=1 to i do
                begin
                    gotoxy(lok,i);
                    write('*');
                    lok:=lok+2;
                end;
            end;
        readln;
    end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat20, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Pertemuan 8

Pernyataan Berulang

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan struktur berulang pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang pernyataan berulang yang menggunakan Pernyataan For.

8.1 Pernyataan For

Untuk perintah perulangan dengan menggunakan For, ada dua bentuk yang dapat digunakan yaitu For-To-Do atau For-DownTo-Do. Bentuk perintah pertama cocok digunakan untuk kounter menaik. Sedangkan bentuk perintah kedua cocok digunakan untuk kounter menurun. Secara sederhana dapat dikatakan bahwa perintah ini paling mudah dimanaatkan untuk perulangan yang sudah pasti dan untuk kondisi perulangan jenis data numerik.

Bentuk Umum Perintah For-To-Do

•
•
•

**For <Variabel>:=<awal> To <akhir> Do
Pernyataan;**

•
•
•

Contoh1:

```
Program ForToDo_1;  
Uses crt;  
var k : integer;  
Begin  
Clrscr;  
K:=0;  
For k:=1 to 5 do  
Writeln('UNRIYO');  
Writeln('Universitas Respati');  
Writeln('Yogyakarta');  
Readln;  
end.
```

Contoh2:

```
Program ForToDo_2;  
Uses crt;  
var k : integer;  
Begin  
Clrscr;  
K:=0;  
For k:=1 to 5 do  
Begin  
Writeln('UNRIYO');  
Writeln('Universitas Respati');  
Writeln('Yogyakarta');  
end;  
Readln;  
end.
```

Contoh3:

```
Program ForToDo_3;  
Uses crt;  
var k : integer;  
Begin  
Clrscr;  
K:=0;  
For k:=1 to 5 do  
Writeln(k);  
Readln;  
end.
```

Bentuk Umum Perintah For-DownTo-Do

```
.  
.   
.   
For <Variabel>:=<awal> DownTo <akhir> Do  
  Pernyataan;  
.   
.   
.
```

Contoh Program:

```
Program ForDownToDo_1;  
Uses crt;  
var k : integer;  
Begin  
  Clrscr;  
  K:=0;  
  For k:=5 Downto 1 do  
    Writeln(k);  
    Readln;  
  end.
```

8.2 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

21. Program Keduapuluhsatu

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prog21;
Uses crt;
var k : integer;
    nama : char;
    gaji : longint;
    totgaji : longint;
Begin
Clrscr;
K:=0;
totgaji:=0;
For k:=1 to 5 Do
Begin
Writeln('Data Ke : ',k);
Write('Nama Pegawai : ');
Readln(nama);
Write('Gaji Pokok : ');
Readln(gaji);
totgaji:=totgaji+gaji;
end;
writeln('Jumlah data : ',k);
Write('Total Gaji yang harus dibayar : ');
writeln(totgaji);
Readln;
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat21, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

22. Program Keduapuluhdua

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prog22;
Uses crt;
var k : integer;
    n : integer;
    namabrg : char;
    harga : longint;
    jml : integer;
    totharga : longint;
    diskon : real;
    bayar : real;
    umasuk : real;
Begin
Clrscr;
K:=0;
Write('Jumlah Data      : ');
readln(n);
writeln;
For k:=1 to n do
  Begin
    Writeln('Data Ke      : ',k);
    Write('Nama Barang    : ');
    Readln(namabrg);
    Write('Harga Satuan    : ');
    Readln(harga);
    Write('Jumlah            : ');
    readln(jml);
    totharga:=harga*jml;
    diskon:=totharga*0.1;
    bayar:=totharga-diskon;
    umasuk:=umasuk+bayar;
    Writeln('Total Harga      : ',totharga);
    Writeln('Diskon           : ',diskon:8:2);
    Writeln('Jumlah Bayar      : ',bayar:8:2);
    writeln;
  end;
  Writeln('Jumlah Uang Masuk : ',umasuk:8:2);
  Readln;
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat22, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Pernyataan Berulang Lanjutan

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan struktur berulang pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang pernyataan berulang yang menggunakan Pernyataan While-do dan Pernyataan Repeat-until.

9.1 Pernyataan While-Do

Pernyataan While-Do atau lebih sederhananya dikenal sebagai pernyataan While, mengeksekusi pernyataan lain selama beberapa kondisi bernilai true.

Bentuk Umum Perintah:

.
. .
.

While <Ekspresi> Do
 Pernyataan;

.
. .
.

Catatan:

Pernyataan yang ada setelah While <Ekspresi> Do harus jelas, dan ekspresi yang diberikan harus terpenuhi. Jika tidak, maka program bisa menghasilkan sesuatu yang diluar perkiraan.

Contoh1:

```
Program WhileDo_1;  
Uses Crt;  
Var k : integer;  
Begin  
    Clrscr;  
    K:=0  
    While k<5 Do  
        Writeln('UNRIYO');  
End.
```


Contoh2:

```
Program WhileDo_2;  
Uses Crt;  
Var k : integer;  
Begin  
  Clrscr;  
  k:=0  
  While k<5 Do  
    Begin  
      Writeln('UNRIYO');  
      k:=k+1;  
    end;  
End.
```

Pernyataan While-Do dapat disajikan dengan beberapa kondisi yaitu kondisi dengan konstanta, kondisi boolean, kondisi tipe karakter, dan kondisi jamak.

1. Putaran While-Do dengan kondisi konstanta

Maksud dari kondisi konstanta adalah ekspresi yang diberikan melakukan pengujian untuk data yang sudah pasti.

Contoh Program:

```
Program WhileDo_Konstanta;  
Uses Crt;  
Var n : integer;  
Begin  
  Clrscr;  
  n:=0  
  While n<10 Do  
    Begin  
      Write('Data ');  
      Readln(n);  
      n:=n+1;  
    end;  
End.
```

2. Putaran While-Do dengan kondisi boolean

Maksud dari kondisi boolean adalah ekspresi yang diberikan melakukan pengujian untuk data yang jenisnya adalah boolean (true atau false).

Contoh Program:

```
Program WhileDo_Boolean;  
Uses Crt;  
Var k : integer;  
Benar : boolean;  
Begin  
  Clrscr;  
  benar:=true;  
  k:=0  
  While benar Do  
    Begin  
      Writeln('UNRIYO');  
      k:=k+1;  
      benar:=(k>10);  
      Writeln('Universitas Respat');  
    end;  
    Writeln('Yogyakarta');  
    Readln;  
End.
```

3. Putaran While-Do dengan kondisi tipe karakter
Maksud dari kondisi tipe karakter adalah ekspresi yang diberikan melakukan pengujian untuk data yang jenisnya karakter.
Contoh Program:

```
Program WhileDo_Karakter;  
Uses Crt;  
Var kar : char;  
Begin  
  Clrscr;  
  Writeln('Putaran While dengan tipe karakter');  
  Kar:='A';  
  While kar<='z' Do  
    Begin  
      Write(kar);  
      kar:=succ(kar);  
    end;  
    writeln;  
    readln;  
End.
```

Keterangan:

Fungsi succ pada program di atas adalah untuk menaikkan nilai ascii dari karakter sebelumnya

4. Putaran While-Do dengan kondisi jamak
Maksud dari kondisi jamak adalah ekspresi yang diberikan lebih dari satu.
Contoh Program:

```
Program WhileDo_Jamak;  
Uses Crt;  
Var k : integer;  
  A, B : integer;  
Begin  
  Clrscr;  
  A:=1;  
  B:=0;  
  K:=1  
  While (a<5) and (B<5) Do  
    Begin  
      Write('Jumlah putaran = ',k);  
      Write('Nilai A = ',A);  
      Write('Nilai B = ',B);  
      A:=succ(A); B:=succ(B);  
      K:=succ(k)  
    end;  
    writeln;  
    readln;  
End.
```

9.2 Pernyataan Repeat-Until

Pernyataan Repeat-Until merupakan alternatif lain dari perintah untuk proses perulangan. Dalam perintah ini, minimal satu kali proses dijalankan sampai dilakukan pengujian ekspresi.

Bentuk Umum Perintah:

```
.  
.   
.   
Repeat  
  Pernyataan;  
Until <Ekspresi>  
.   
.   
.
```

Contoh:

```
Program Repeat_Until_1;  
Uses crt;  
var k : integer;  
Begin  
  Clrscr;  
  K:=0;  
  Repeat  
    Writeln('UNRIYO');  
    k:=k+1;  
  Until K>5;  
  Writeln('Universitas Respati');  
  Writeln('Yogyakarta');  
  readln;  
end.
```

9.3 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Selain itu sebaiknya juga telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

23. Program Keduapuluhtiga

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prog23;
Uses crt;
var k : integer;
    nama : char;
    gaji : longint;
    totgaji : longint;
Begin
Clrscr;
K:=0;
totgaji:=0;
While k<5 do
Begin
k:=k+1;
Writeln('Data Ke : ',k);
Write('Nama Pegawai : ');
Readln(nama);
Write('Gaji Pokok : ');
Readln(gaji);
totgaji:=totgaji+gaji;
end;
writeln('Jumlah data : ',k);
Write('Total Gaji yang harus dibayar : ');
writeln(totgaji);
Readln;
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat23, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

24. Program Keduapuluhempat

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prog24;
Uses crt;
var k : integer;
    nama : char;
    gaji : longint;
    totgaji : longint;
Begin
Clrscr;
K:=0;
totgaji:=0;
Repeat
    k:=k+1;
    Writeln('Data Ke : ',k);
    Write('Nama Pegawai : ');
    Readln(nama);
    Write('Gaji Pokok : ');
    Readln(gaji);
    totgaji:=totgaji+gaji;
until k>4;
    writeln('Jumlah data : ',k);
    Write('Total Gaji yang harus dibayar : ');
    writeln(totgaji);
    Readln;
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat24, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

25. Program Keduapuluhlima

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prog25;
Uses crt;
var k : integer;
    n : integer;
    namabrg : char;
    harga : longint;
    jml : integer;
    totharga : longint;
    diskon : real;
    bayar : real;
    umasuk : real;
Begin
Clrscr;
K:=0;
Write('Jumlah Data : ');
readln(n);
```

```

writeln;
While k<n do
  Begin
    k:=k+1;
    Writeln('Data Ke      : ',k);
    Write('Nama Barang    : ');
    Readln(namabrg);
    Write('Harga Satuan   : ');
    Readln(harga);
    Write('Jumlah          : ');
    readln(jml);
    totharga:=harga*jml;
    diskon:=totharga*0.1;
    bayar:=totharga-diskon;
    umasuk:=umasuk+bayar;
    Writeln('Total Harga    : ',totharga);
    Writeln('Diskon         : ',diskon:8:2);
    Writeln('Jumlah Bayar     : ',bayar:8:2);
    writeln;
  end;
  Writeln('Jumlah Uang Masuk : ',umasuk:8:2);
  Readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat25, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

26. Program Keduapuluhenam

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program Prog26;
Uses crt;
var k : integer;
    n : integer;
    namabrg : char;
    harga : longint;
    jml : integer;
    totharga : longint;
    diskon : real;
    bayar : real;
    umasuk : real;
Begin
  Clrscr;
  K:=0;
  Write('Jumlah Data      : ');
  readln(n);
  writeln;
  Repeat
    k:=k+1;
    Writeln('Data Ke          : ',k);
    Write('Nama Barang      : ');
    Readln(namabrg);

```

```

Write('Harga Satuan    : ');
Readln(harga);
Write('Jumlah          : ');
readln(jml);
totharga:=harga*jml;
diskon:=totharga*0.1;
bayar:=totharga-diskon;
umasuk:=umasuk+bayar;
Writeln('Total Harga    : ',totharga);
Writeln('Diskon        : ',diskon:8:2);
Writeln('Jumlah Bayar    : ',bayar:8:2);
writeln;
until k>=n;
  Writeln('Jumlah Uang Masuk : ',umasuk:8:2);
  Readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat26, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Pertemuan 10

Variabel Berindeks (Array)

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan penggunaan array pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang array dengan berbagai tipe data dan berbagai bentuk. Beberapa tipe array yang akan dibahas yaitu array tipe integer, array tipe char, array tipe string. Sedangkan bentuk array yang dibahas adalah array dimensi satu dan array dimensi banyak. Tentu saja pemilihan tipe data dan bentuk yang digunakan dalam pemecahan masalah disesuaikan dengan tingkat kompleksitas permasalahan dan kebiasaan pemrogram.

10.1 Array dengan tipe integer

Untuk array tipe ini juga akan dibagi menjadi beberapa tipe, yaitu tipe indeks integer, tipe indeks char, tipe indeks boolean, tipe indeks subrange, dan tipe indeks terbilang.

1. Tipe indeks integer

Contoh 1:

```
Program array1_1;  
uses crt;  
var a : array[1..10] of integer;  
n,i : integer;  
begin  
  clrscr;  
  n:=0;  
  for i:=1 to 10 do  
    begin  
      a[i] :=n  
      n:=n+1  
    end;  
  writeln('  I  A[I]');  
  for i:=1 to 10 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



Contoh 2:

```
Program array1_2;  
uses crt;  
var a : array[1..10] of integer;  
n,i : integer;  
begin  
  clrscr;  
  n:=0;  
  for i:=10 downto 1 do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=1 to 10 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



I	A[I]
1	9
2	8
3	7
4	6
5	5
6	4
7	3
8	2
9	1
10	0

Contoh 3:

```
Program array1_3;  
uses crt;  
var a : array[shortint] of integer;  
n,i : integer;  
begin  
  clrscr;  
  n:=0;  
  for i:=1 to 10 do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=1 to 10 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan sama seperti hasil pada contoh 1

Contoh 4:

```
Program array1_4;  
uses crt;  
var a : array[byte] of integer;  
n,i : integer;  
begin  
  clrscr;  
  n:=0;  
  for i:=1 to 10 do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=1 to 10 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan sama seperti hasil pada contoh 1

2. Tipe indek char

Contoh 5:

```
Program array1_5;  
uses crt;  
var a : array['A'..'E'] of integer;  
n : integer;  
i : char;  
begin  
  clrscr;  
  n:=0;  
  for i:='A' to 'E' do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:='A' to 'E' do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```

I  A[I]
=====
A  0
B  1
C  2
D  3
E  4
```

Contoh 6:

```
Program array1_6;  
uses crt;  
var a : array[char] of integer;  
n : integer;  
i : char;  
begin  
  clrscr;  
  n:=0;  
  for i:='E' downto 'A' do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:='A' to 'E' do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```

E:\MASTER~1\Tp70\TURBO.EXE
=====
I  A[I]
A  4
B  3
C  2
D  1
E  0
=====
```

Contoh 7:

```
Program array1_7;  
uses crt;  
var a : array['1'..'5'] of integer;  
n : integer;  
i : char;  
begin  
  clrscr;  
  n:=0;  
  for i:='1' to '5' do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:='1' to '5' do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



The screenshot shows a Turbo Pascal window titled "E:\MASTER~1\Tp70\TURBO.EXE". The program has printed the following output:

I	A[I]
1	0
2	1
3	2
4	3
5	4

3. Tipe indek boolean

Contoh 8:

```
Program array1_8;  
uses crt;  
var a : array[false..true] of integer;  
n : integer;  
i : boolean;  
begin  
  clrscr;  
  n:=0;  
  for i:=false to true do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=false to true do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



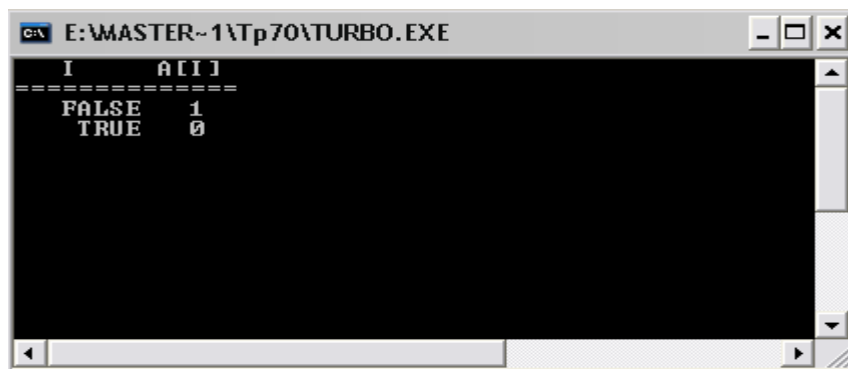
The screenshot shows a Turbo Pascal window titled "E:\MASTER~1\Tp70\TURBO.EXE". The program has printed the following output:

I	A[I]
FALSE	0
TRUE	1

Contoh 9:

```
Program array1_9;  
uses crt;  
var a : array[false..true] of integer;  
n : integer;  
i : boolean;  
begin  
  clrscr;  
  n:=0;  
  for i:=true downto false do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=false to true do  
    writeln(' ',i:6,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:




4. Tipe indeks subrange

Contoh 10:

```
Program array1_10;  
uses crt;  
var a : array[1..5] of integer;  
n,i : integer;  
begin  
  clrscr;  
  n:=0;  
  for i:=5 downto 1 do  
    begin  
      a[i] :=n;  
      n:=n+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=1 to 5 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```

=====
I      A[I]
1      4
2      3
3      2
4      1
5      0

```

Contoh 11:

```

Program array1_11;
uses crt;
var a : array['5'..'9'] of integer;
n : integer;
i : char;
begin
  clrscr;
  n:=0;
  for i:='5' to '9' do
    begin
      a[i] :=n;
      n:=n+1;
    end;
  writeln(' I A[I]');
  writeln('=====');
  for i:='9' downto '5' do
    writeln(' ',i:2,' ',a[i]);
  readln;
end.

```

Hasil saat program dijalankan seperti tampilan berikut:



```

=====
I      A[I]
9      4
8      3
7      2
6      1
5      0

```

Contoh 12:

```
Program array1_12;  
uses crt;  
const awal=1;  
    akhir=10;  
var a : array[awal..akhir] of integer;  
n,i : integer;  
begin  
    clrscr;  
    n:=0;  
    for i:=10 downto 1 do  
        begin  
            a[i] :=n;  
            n:=n+1;  
        end;  
    writeln(' I A[I]');  
    writeln('=====');  
    for i:=1 to 10 do  
        writeln(' ',i:2,' ',a[i]);  
    readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



I	A[I]
1	9
2	8
3	7
4	6
5	5
6	4
7	3
8	2
9	1
10	0

5. Tipe indek terbilang

Contoh 13:

```
Program array1_13;  
uses crt;  
type t=(macan, kucing, harimau, leo, gajah);  
var a : array[t] of integer;  
n : integer;  
i : t;  
begin  
    clrscr;  
    n:=0;  
    for i:=macan to gajah do  
        begin  
            a[i] :=n;  
            n:=n+1;  
        end;  
    writeln(' A[I]');  
    writeln('=====');  
    for i:=gajah downto macan do  
        writeln(' ',a[i]);  
    readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```
E:\MASTER~1\Tp70\TURBO.EXE
A[I]
=====
4
3
2
1
0
```

Contoh 14:

```
Program array1_14;
uses crt;
type t=(macan, kucing, harimau, leo, gajah);
var a : array[t] of integer;
n : integer;
i : t;
begin
  clrscr;
  n:=0;
  for i:=macan to leo do
    begin
      a[i] :=n;
      n:=n+1;
    end;
  writeln(' A[I]');
  writeln('=====');
  for i:=leo downto macan do
    writeln(' ',a[i]);
  readln;
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```
E:\MASTER~1\Tp70\TURBO.EXE
A[I]
=====
3
2
1
0

```

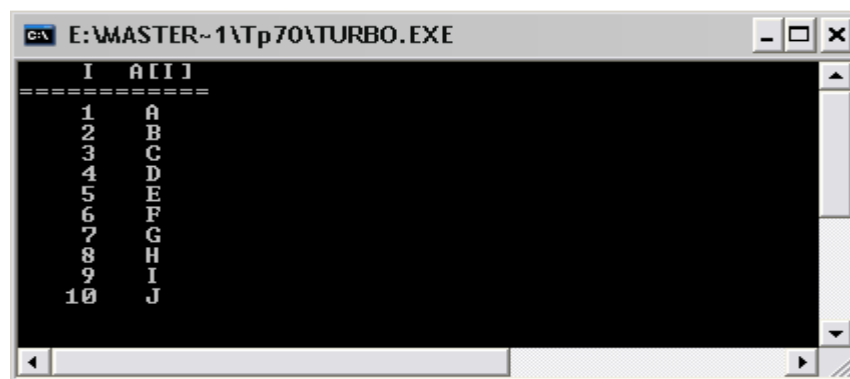

10.2 Array dengan tipe char

1. Tipe indek integer

Contoh 15:

```
Program array2_1;  
uses crt;  
var a : array[1..10] of char;  
i : integer;  
n : char;  
begin  
  clrscr;  
  for i:=1 to 10 do  
    begin  
      a[i] :=chr(64+i);  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=1 to 10 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:




```

E:\MASTER~1\Tp70\TURBO.EXE
  I  A[I]
=====
  1  A
  2  B
  3  C
  4  D
  5  E
  6  F
  7  G
  8  H
  9  I
 10  J
```

Contoh 16:

```
Program array2_2;  
uses crt;  
var a : array[byte] of char;  
i,k : integer;  
n : char;  
begin  
  clrscr;  
  k:=1;  
  for i:=100 downto 85 do  
    begin  
      a[i] :=chr(96+k);  
      k:=k+1;  
    end;  
  writeln('  I A[I]');  
  writeln('=====');  
  for i:=100 downto 85 do  
    writeln(' ',i:3,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



The screenshot shows a Turbo Pascal window titled "E:\MASTER~1\Tp70\TURBO.EXE". The program output is as follows:

I	A[I]
100	a
99	b
98	c
97	d
96	e
95	f
94	g
93	h
92	i
91	j
90	k
89	l
88	m
87	n
86	o
85	p

2. Tipe indek char

Contoh 17:

```
Program array2_3;  
uses crt;  
var a : array['A'..'E'] of char;  
n : integer;  
i : char;  
begin  
  clrscr;  
  n:=1;  
  for i:='A' to 'E' do  
    begin  
      a[i] :=chr(64+n);  
      n:=n+1;  
    end;  
  writeln(' I A[I]');  
  writeln('=====');  
  for i:='A' to 'E' do  
    writeln(' ',i,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



The screenshot shows a Turbo Pascal window titled "E:\MASTER~1\Tp70\TURBO.EXE". The program output is as follows:

I	A[I]
A	A
B	B
C	C
D	D
E	E

3. Tipe indek boolean

Contoh 18:

```
Program array2_4;  
uses crt;  
var a : array[boolean] of char;  
n : integer;  
i : boolean;  
begin  
  clrscr;  
  n:=1;  
  for i:=true downto false do  
    begin  
      a[i] :=chr(97+n);  
      n:=n+1;  
    end;  
  writeln('  I    A[I]');  
  writeln('=====');  
  for i:=false to true do  
    writeln(' ',i:6,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



4. Tipe indek subrange

Contoh 19:

```
Program array2_5;  
uses crt;  
var a : array['5'..'9'] of char;  
n : integer;  
i : char;  
begin  
  clrscr;  
  n:=1;  
  for i:='5' to '9' do  
    begin  
      a[i] :=chr(64+n);  
      n:=n+1;  
    end;  
  writeln('  I    A[I]');  
  writeln('=====');  
  for i:='9' to '5' do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



5. Tipe indek terbilang

Contoh 20:

```
Program array2_6;  
uses crt;  
type t=(macan, kucing, harimau, leo, gajah);  
var a : array[t] of char;  
n : integer;  
i : t;  
begin  
  clrscr;  
  n:=0;  
  for i:=macan to leo do  
    begin  
      a[i] :=chr(97+n);  
      n:=n+1;  
    end;  
  writeln('  A[I]');  
  writeln('=====');  
  for i:=leo downto macan do  
    writeln(' ',a[i]);  
  readln;  
end.
```

Hasil saat program dijalankan seperti tampilan berikut:



10.3 Array dengan tipe string

1. Tipe indek integer

Contoh 21:

```
Program array3_1;  
uses crt;  
var a : array[1..10] of string[20];  
i : integer;  
begin  
  clrscr;  
  writeln('Masukkan 5 nama mahasiswa');  
  for i:=1 to 5 do  
    begin  
      write('Masukkan Nama Mahasiswa : ');  
      readln(a[i]);  
    end;  
  writeln('Nama Mahasiswa yang Anda Masukkan adalah:');  
  for i:=1 to 5 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

Contoh 22:

```
Program array3_2;  
uses crt;  
var a : array[shortint] of string[20];  
i : integer;  
begin  
  clrscr;  
  writeln('Masukkan 5 nama mahasiswa');  
  for i:=-2 to 2 do  
    begin  
      write('Masukkan Nama Mahasiswa : ');  
      readln(a[i]);  
    end;  
  writeln('Nama Mahasiswa yang Anda Masukkan adalah:');  
  for i:=-2 to 2 do  
    writeln(' ',i:2,' ',a[i]);  
  readln;  
end.
```

2. Tipe indek char

Contoh 23:

```
Program array3_3;  
uses crt;  
var a : array[char] of string[20];  
i : char;  
begin  
  clrscr;  
  writeln('Masukkan 5 nama mahasiswa');  
  for i:='a' to 'e' do  
    begin  
      write('Masukkan Nama Mahasiswa : ');  
      readln(a[i]);  
    end;  
  writeln('Nama Mahasiswa yang Anda Masukkan adalah:');
```

```

for i:='a' to 'e' do
  writeln(' ',i:2,' ',a[i]);
readln;
end.

```

3. Tipe indek boolean

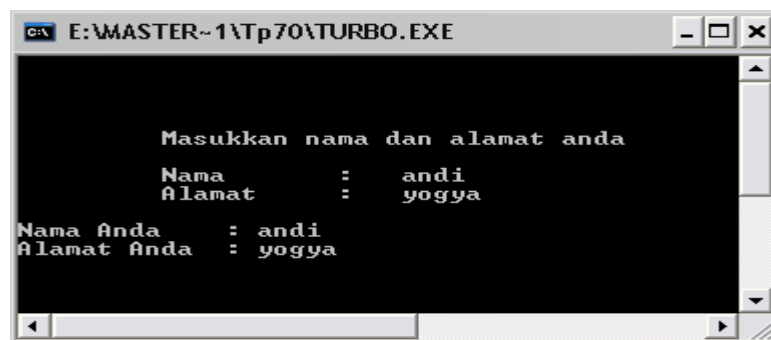
Contoh 24:

```

Program array3_4;
uses crt;
var a : array[boolean] of string[20];
i : boolean;
n : integer;
begin
  clrscr;
  gotoxy(10,5);writeln('Masukkan nama dan alamat anda');
  gotoxy(10,7);writeln('Nama      : ');
  gotoxy(10,8);writeln('Alamat   : ');
  n:=1;
  for i:=false to true do
  begin
    gotoxy(25,6+n);readln(a[i])
    n:=n+1;
  end;
  writeln;
  writeln('Nama Anda   : ',a[false]);
  writeln('Alamat Anda  : ',a[true]);
  readln;
end.

```

Hasil saat program dijalankan seperti tampilan berikut:



4. Tipe indek subrange

Contoh 25:

```

Program array3_5;
uses crt;
var a : array['D'..'H'] of string[20];
i : char;
begin
  clrscr;
  writeln('Masukkan 5 nama mahasiswa');
  for i:='H' downto 'D' do
  begin
    write('Masukkan Nama Mahasiswa : ');
    readln(a[i]);
  end;
end.

```

```

    end;
    writeln('Nama Mahasiswa yang Anda Masukkan adalah:');
    for i:='H' downto 'D' do
        writeln(' ',i:2,' ',a[i]);
    end;
    readln;
end.

```

Contoh 26:

```

Program array3_6;
uses crt;
var a : array[20..25] of string[20];
i : integer;
begin
    clrscr;
    writeln('Masukkan 5 nama mahasiswa');
    for i:=25 downto 20 do
        begin
            write('Masukkan Nama Mahasiswa : ');
            readln(a[i]);
        end;
    writeln('Nama Mahasiswa yang Anda Masukkan adalah:');
    for i:=25 downto 20 do
        writeln(' ',i:2,' ',a[i]);
    end;
    readln;
end.

```

5. Tipe indek terbilang

Contoh 27:

```

Program array3_7;
uses crt;
type t=(merah, kuning, hijau, biru);
var a : array[t] of string[20];
n : integer;
i : t;
begin
    clrscr;
    n:=1;
    writeln('Masukkan 4 kota di Indonesia');
    for i:=merah to biru do
        begin
            write('Kota : ',n,' ');readln(a[i]);
            n:=n+1;
        end;
    writeln;
    writeln('Kota yang anda masukkan,urut dari yang terakhir adalah:');
    n:=1;
    for i:=biru downto merah do
        begin
            writeln(' ',n:2,' ',a[i]);
            n:=n+1;
        end;
    readln;
end.

```

Hasil saat program dijalankan seperti tampilan berikut:



```
E:\MASTER~1\Tp70\TURBO.EXE
Masukkan 4 kota di Indonesia
Kota : 1 a
Kota : 2 b
Kota : 3 c
Kota : 4 d

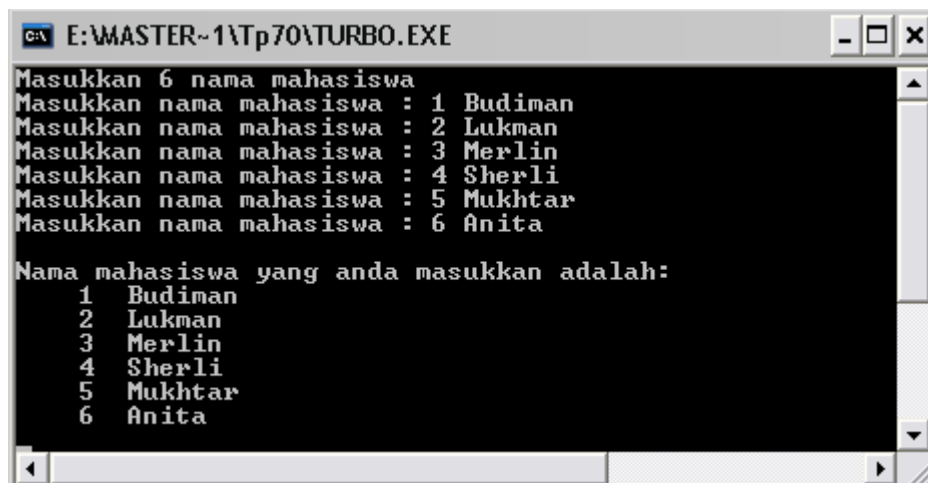
Kota yang anda masukkan, urut dari yang terakhir adalah:
1 d
2 c
3 b
4 a
```

10.4 Array dengan dimensi banyak

Contoh 28:

```
Program array4_1;
uses crt;
var a : array[1..3,1..3] of string;
i,k,n : integer;
begin
  clrscr;
  n:=1;
  writeln('Masukkan 6 nama mahasiswa');
  for i:=1 to 2 do
    for k:=1 to 3 do
      begin
        write('Masukkan nama mahasiswa : ',n,' ');
        readln(a[i,k]);
        n:=n+1;
      end;
    writeln;
  writeln('Nama mahasiswa yang anda masukkan adalah:');
  n:=1;
  for i:=1 to 2 do
    for k:=1 to 3 do
      begin
        writeln(' ',n:2,' ',a[i,k]);
        n:=n+1;
      end;
    readln;
  end.
```

Hasil saat program dijalankan seperti tampilan berikut:



```
E:\MASTER~1\Tp70\TURBO.EXE
Masukkan 6 nama mahasiswa
Masukkan nama mahasiswa : 1 Budiman
Masukkan nama mahasiswa : 2 Lukman
Masukkan nama mahasiswa : 3 Merlin
Masukkan nama mahasiswa : 4 Sherli
Masukkan nama mahasiswa : 5 Mukhtar
Masukkan nama mahasiswa : 6 Anita

Nama mahasiswa yang anda masukkan adalah:
1 Budiman
2 Lukman
3 Merlin
4 Sherli
5 Mukhtar
6 Anita
```


10.5 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

27. Program Keduapuluhtujuh

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program urutbuble;  
uses crt;  
var dt : array[1..10] of integer;  
i,j, bub : integer;  
begin  
  clrscr;  
  writeln('Pengurutan Dengan Metode Bubble Sort');  
  writeln('Masukkan Data Yang Akan Diurutkan');  
  for i:=1 to 10 do  
    begin  
      write('Data ke ',i,'= ');  
      readln(dt[i]);  
    end;  
  for i:=1 to 10-1 do  
    for j:=10 downto i+1 do  
      if dt[j]<dt[j-1] then  
        begin  
          bub:=dt[j];  
          dt[j]:=dt[j-1];  
          dt[j-1]:=bub;  
        end;  
      writeln('Hasil Pengurutan');  
    for i:=1 to 10 do  
      begin  
        writeln(dt[i]);  
      end;  
    readln;  
  end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat27, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

28. Program Keduapuluhdelapan

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program caridata;  
uses crt;  
var deret : array[1..10] of integer;  
i,j, n, data, lok : integer;  
ada : boolean;
```

```

begin
  clrscr;
  write('Masukkan banyaknya data : ');
  readln(n);
  for i:=1 to n do
    begin
      write('Bilangan ke ',i:3, ' = ');
      readln(deret[i]);
    end;
  writeln;
  write('Bilangan yang akan dicari : ');
  readln(data);
  ada:=false;
  i:=0;
  while (i<=n) and (not ada) do
    begin
      if deret[i]=data then
        begin
          ada:=true;
          writeln;
          writeln('Bilangan yang dicari ada pada deret ',i);
        end
      else
        i:=i+1;
      end;
    if (not ada) then
      writeln('Data yang dicari tidak ada..!');
      readln;
    end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat28, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

29. Program Keduapuluhsembilan

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program Perkalian_Matriks;
uses crt;
var a,f,k      : array[1..10,1..10] of integer;
    b,c,i,j,l,z : longint; m,n,d,e,g,h:byte;
begin
  clrscr;
  writeln('Program Perkalian 2 Matriks');
  writeln('Masukkan data untuk matriks pertama (M1)');
  write('banyaknya baris matriks pertama: '); readln(b);
  write('banyaknya kolom matriks pertama: '); readln(c);
  for d:= 1 to b do
    begin
      for e:= 1 to c do
        begin
          write('entri baris ke-',d,' kolom ke-',e,': '); readln(a[d,e]);
        end;
      end;
    end;

```



```
baris)');  
end;  
readln  
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat29, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Prosedur dan Fungsi

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan prosedur dan fungsi pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang prosedur secara sederhana, prosedur tanpa parameter, dan prosedur dengan parameter. Selain itu juga akan dijelaskan mengenai fungsi secara sederhana, fungsi tanpa parameter, dan fungsi dengan parameter.

11.1 Prosedur

1. Deklarasi Prosedur

Prosedur adalah suatu program terpisah dalam blok tersendiri yang berfungsi sebagai sub program (program bagian) dan diawali dengan kata cadangan Procedure.

Bentuk Umum Prosedur :

Procedure nama (daftar_parameter);
Bagian deklarasi;
Bagian pernyataan;

2. Paramater Dalam Prosedur

- a. *Parameter Bersifat Lokal* artinya bahwa nilai yang terdapat didalam suatu modul program hanya dapat digunakan pada modul atau unit program yang bersangkutan saja sehingga tidak dapat digunakan pada modul atau unit program lain.

Contoh program

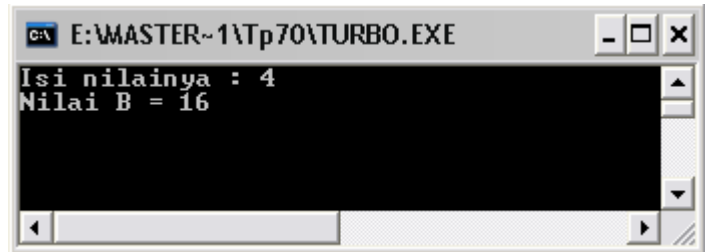
```
Procedure Kali;  
Var  
A, B : Byte;  
Begin  
Write ('Isi nilainya : '); Readln (A);  
B := A * A;  
End;  
  
Begin  
Kali;  
Writeln ('Nilai B = ', B);  
End.
```

Saat program dijalankan, akan ditampilkan komentar "Unknown identifier"
Hal ini disebabkan pada program utama ada perintah untuk menampilkan nilai B, padahal variabel B hanya didefinisikan pada

- b. *Parameter Bersifat Global* adalah kebalikan dari lokal. Agar nilainya dapat digunakan untuk beberapa atau semua modul/unit program maka nilai tersebut harus dideklarasikan di atas modul yang akan menggunakannya.

Contoh program

```
Program Contoh_Prosedur_1;  
Uses crt;  
Var  
A, B : Byte;  
  
Procedure Kali;  
Begin  
Write ('Isi nilainya : ');  
Readln (A);  
B := A * A;  
End;  
  
Begin  
Clrscr;  
Kali;  
Writeln ('Nilai B = ', B);  
End.
```



Istilah Di Dalam Parameter

- Actual parameter (parameter nyata) yaitu parameter yang dikirimkan dari modul utama ke modul prosedur
- Formal Parameter (parameter formal) yaitu parameter yang ada dan dituliskan pada judul prosedur
- Parameter Passing yaitu proses Pemanggilan data lewat parameter nyata ke parameter formal.
- By Value yaitu Pemanggilan parameter secara nilai
- By Reference yaitu Pemanggilan parameter secara acuan
- Value Parameter yaitu parameter-parameter yang digunakan dalam Pemanggilan secara nilai

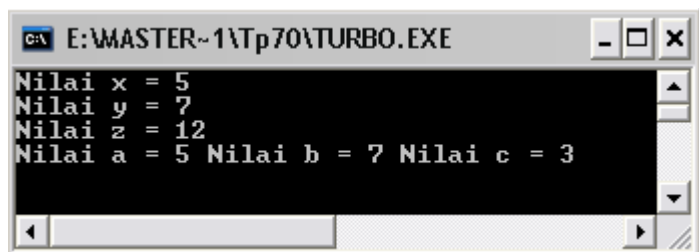
3. Pemanggilan Parameter

- Pemanggilan secara nilai (by Value)

Pemanggilan parameter secara nilai bersifat searah yaitu dari parameter nyata ke parameter formal. Bila nilai parameter formal berubah, maka nilai parameter nyata tidak berubah.

Contoh program

```
Procedure Hitung (X, Y, Z : Byte);  
Begin  
Z := X + Y;  
Writeln ('Nilai X = ', X);  
Writeln ('Nilai Y = ', Y);  
Writeln ('Nilai Z = ', Z);  
End;  
Var  
A, B, C : Byte;  
Begin  
A := 5; B := 7; C := 3;  
Hitung (A, B, C);  
Writeln ('Nilai A = ', A, 'Nilai B = ', B, 'Nilai C = ', C);  
Readln;  
End.
```



- Pemanggilan secara Acuan (by reference)

Pemanggilan parameter secara acuan bersifat dua arah (bolak – balik). Bila nilai parameter formal berubah, maka nilai parameter nyata ikut berubah.

Contoh program

```
Procedure Hitung (Var X, Y, Z : Byte);
```

```
Begin
```

```
Z := X + Y;
```

```
Writeln ('Nilai X = ', X);
```

```
Writeln ('Nilai Y = ', Y);
```

```
Writeln ('Nilai Z = ', Z);
```

```
End;
```

```
Var
```

```
A, B, C : Byte;
```

```
Begin
```

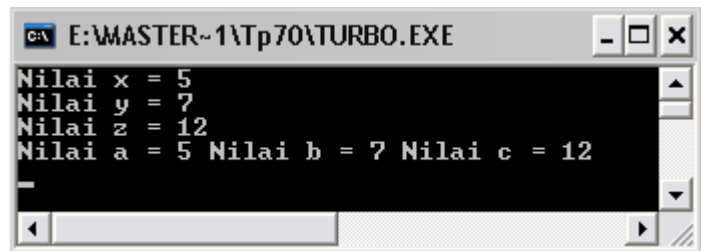
```
A := 5; B := 7; C := 3;
```

```
Hitung (A, B, C);
```

```
Writeln ('Nilai A = ', A, 'Nilai B = ', B, 'Nilai C = ', C);
```

```
Readln;
```

```
End.
```



- c. Pemanggilan parameter sebagian secara nilai dan sebagian secara acuan

Contoh program

```
Procedure Hitung (X, Y : Byte; Var Z : Byte);
```

```
Begin
```

```
Z := X + Y;
```

```
End;
```

```
Var
```

```
A, B, C : Byte;
```

```
Begin
```

```
A := 5; B := 7; C := 3;
```

```
Hitung (A, B, C);
```

```
Writeln ('Nilai X = ', X);
```

```
Writeln ('Nilai Y = ', Y);
```

```
Writeln ('Nilai Z = ', Z);
```

```
Readln;
```

```
End.
```

Saat program dijalankan, akan ditampilkan komentar "Unknown identifier"

11.2 Fungsi

1. Deklarasi Fungsi

Fungsi secara garis besar sama dengan procedure yang membedakannya adalah nama fungsi harus dideklarasikan dengan type datanya.

Bentuk Umum

Function Identifier (daftar-parameter):

Type;

Contoh Penulisan :

Function faktorial (var fak, hasil : Integer) : integer;

2. Parameter Pada Fungsi

Sifat parameter dalam fungsi sama dengan sifat parameter dalam prosedur, yaitu bersifat lokal dan global.

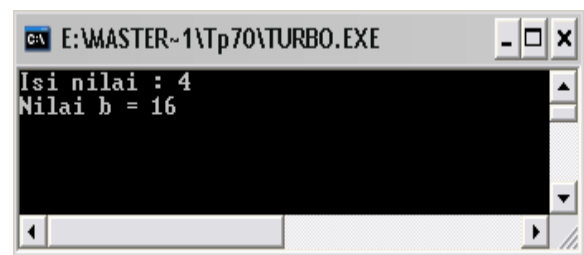
Contoh program (parameter bersifat lokal)

```
Function Kali: Byte;  
Var  
A, B : Byte;  
Begin  
Write ('Isi nilainya : ');  
Readln (A);  
B := A * A;  
End;  
Begin  
Kali;  
Writeln ('Nilai B = ', B);  
End.
```

Saat program dijalankan, akan ditampilkan komentar "Unknown identifier"
Hal ini disebabkan pada program utama ada perintah untuk menampilkan nilai B, padahal variabel B hanya didefinisikan pada

Contoh program (parameter bersifat global)

```
Var  
A, B : Byte;  
Function Kali: Byte;  
Begin  
Write ('Isi nilainya : ');  
Readln (A);  
B := A * A;  
End;  
Begin  
Kali;  
Writeln ('Nilai B = ', B);  
End.
```

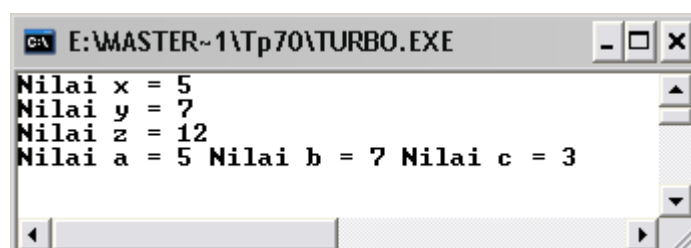


3. Pemanggilan Fungsi

Parameter dalam fungsi (idem dengan prosedur), yaitu dapat dilakukan pemanggilan secara nilai (by Value) atau secara acuan (by reference)

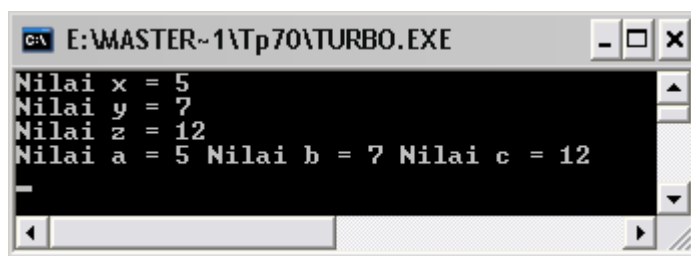
Contoh program (by Value)

```
Function Hitung (X, Y, Z : Byte): Byte;  
Begin  
Z := X + Y;  
Writeln ('Nilai X = ', X);  
Writeln ('Nilai Y = ', Y);  
Writeln ('Nilai Z = ', Z);  
End;  
Var  
A, B, C : Byte;  
Begin  
A := 5; B := 7; C := 3;  
Hitung (A, B, C);  
Writeln ('Nilai A = ', A, 'Nilai B = ', B, 'Nilai C = ', C);  
Readln;  
End.
```



Contoh program (by Reference)

```
Function Hitung (Var X, Y, Z : Byte): Byte;  
Begin  
  Z := X + Y;  
  Writeln ('Nilai X = ', X);  
  Writeln ('Nilai Y = ', Y);  
  Writeln ('Nilai Z = ', Z);  
End;  
Var  
  A, B, C : Byte;  
Begin  
  A := 5; B:= 7; C:=3;  
  Hitung (A, B, C);  
  Writeln ('Nilai A = ',A,'Nilai B = ',B,'Nilai C = ',C);  
  Readln;  
End.
```



10.2 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Sebaiknya telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

30. Program Ketigapuluh

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

Program Prosedur_Tanpa_Parameter;

```
uses crt;  
var a,b : real;
```

Procedure Perkalian;

```
var c : real; {variabel c merupakan variabel lokal yang hanya}  
begin        {berlaku di prosedur Perkalian}  
  c:=a*b;  
  writeln('Nilai c ');  
  writeln('selama menjalankan prosedur Perkalian = ',c:4:2);  
end;
```

{Program Utama}

```
var c : real; {variabel c merupakan variabel lokal yang hanya}  
begin        {berlaku di program utama}  
  clrscr;  
  a:=5.0;
```

```

b:=10.0;
c:=9.0;
writeln('a = ',a:4:2);
writeln('b = ',b:4:2);
writeln;
writeln('Nilai c');
writeln('sebelum menjalankan prosedur Perkalian = ',c:4:2);
Perkalian;
writeln;
writeln('Nilai c');
writeln('sesudah menjalankan prosedur Perkalian = ',c:4:2);
writeln('a x b = ',c:4:2);
readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat30, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

31. Program Ketigapuluhsatu

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program Fungsi_Tanpa_Parameter;
uses crt;
var a,b,c : real;

Function Perkalian:real;
begin
  perkalian:=a*b;
end;

{Program Utama}
begin
  clrscr;
  a:=5.0;
  b:=10.0;
  c:=9.0;
  writeln('a = ',a:4:2);
  writeln('b = ',b:4:2);
  c:=a*b;
  writeln('Hasil Perkalian tanpa fungsi Perkalian : ');
  writeln('a x b = ',c:4:2);
  writeln;
  writeln('Hasil Perkalian dengan fungsi Perkalian : ');
  writeln('a x b = ',perkalian:4:2);
  readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat31, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

32. Program Ketigapuluhdua

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Prosedur_Dengan_Parameter_ByReference;  
uses crt;
```

```
Procedure Perkalian(var a,b,c:real);  
begin  
  c:=a*b;  
end;
```

```
{Program Utama}  
var x,y,z :real;  
begin  
  clrscr;  
  x:=5.0;  
  y:=10.0;  
  z:=9.0;  
  writeln('x = ',x:4:2);  
  writeln('y = ',y:4:2);  
  writeln('z = ',z:4:2);  
  Perkalian(x,y,z);  
  writeln('x * z = ',z:4:2);  
  readln;  
end.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat32, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

33. Program Ketigapuluhtiga

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Fungsi_Dengan_Parameter_ByValue;  
uses crt;
```

```
Function Perkalian(a,b,c:real):real;  
begin  
  perkalian:=a*b;  
  c:=a/b;  
end;
```

```
{Program Utama}  
var x,y,z :real;  
begin  
  clrscr;  
  x:=5.0;  
  y:=10.0;  
  z:=9.0;  
  writeln('x = ',x:4:2);
```

```

writeln('y = ',y:4:2);
writeln('z = ',z:4:2);
writeln('x y z = ',Perkalian(x,y,z):4:2);
writeln('Nilai z setelah fungsi Perkalian dijalankan : ');
writeln('z = ',z:4:2);
readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat33, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

34. Program Ketigapuluhempat

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```

Program Fungsi_Dengan_Parameter_ByReference;
uses crt;

```

```

Function Perkalian(var a,b,c:real):real;
begin
  perkalian:=a*b;
  c:=a/b;
end;

```

```

{Program Utama}
var x,y,z :real;
begin
  clrscr;
  x:=5.0;
  y:=10.0;
  z:=9.0;
  writeln('x = ',x:4:2);
  writeln('y = ',y:4:2);
  writeln('z = ',z:4:2);
  writeln('x y z = ',Perkalian(x,y,z):4:2);
  writeln('Nilai z setelah fungsi Perkalian dijalankan : ');
  writeln('z = ',z:4:2);
readln;
end.

```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat34, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

35. Program Ketigapuluhlima

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

```
Program Penentuan_Nilai;  
uses crt;
```

```
type  
arr = array[1..4] of byte;  
var  
N : Arr;  
Jawab : Char;
```

```
Procedure Input;  
var i : byte;  
Begin  
  clrscr;  
  gotoxy(3,1); write('Perhitungan Nilai Akhir');  
  gotoxy(3,2); write('=====');  
  gotoxy(3,4); write('Nilai Presensi : ');  
  gotoxy(3,5); write('Nilai TM : ');  
  gotoxy(3,6); write('Nilai Mid : ');  
  gotoxy(3,7); write('Nilai Ujian Sem : ');  
  gotoxy(3,8); write('Nilai Akhir : ');  
  gotoxy(34,8); write('==> Grade : ');  
  for i := 1 to 4 do  
  begin  
    repeat  
      gotoxy(21,3+i);clrEol;  
      gotoxy(21,3+i); readln(N[i]);  
      until N[i] in [0..100];  
    end;  
  end;
```

```
Procedure Hitung;  
var  
akhir : real;  
grade : char;  
begin  
  akhir :=(0.1*N[1])+(0.1*N[2])+(0.3*N[3])+(0.5*N[4]);  
  gotoxy(21,8); write(akhir:-5:2);  
  if akhir >= 85 then grade :='A' else  
  if akhir >= 75 then grade :='B' else  
  if akhir >= 65 then grade :='C' else  
  if akhir >= 55 then grade :='D' else grade :='E';  
  gotoxy(46,8); write(grade);  
end;
```

```
begin  
  repeat  
    Input;  
    Hitung;  
    gotoxy(13,10); write('Mau coba lagi (Y/T)? : ');  
  repeat  
    gotoxy(36,10); jawab:=Ucase(readkey);  
    until (jawab='Y') or (jawab='T');  
    write(jawab);
```

until (jawab ='T');
end.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat35, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

36. Program Ketigapuluhenam

- Tampilkan menu File
- Pilih (sorot lalu ENTER) pada pilihan New
- Ketiklah program seperti di bawah ini

Program BilPrima;

uses crt;

Function factor (n:LongInt) : LongInt;

var

i, temp : LongInt;

Begin

temp :=0;

for i := 1 to n do

if n mod i=0 then inc(temp);

factor:=temp;

end;

var x, a, b, n : LongInt;

Begin

clrscr;

gotoxy(30,4); writeln('Deret Bilangan Prima');

gotoxy(30,5); writeln('=====');

gotoxy(3,7); writeln('Jumlah bilangan yang mau ditampilkan : ');

repeat

gotoxy(45,7); clrEol;

readln(a);

until (a>=2);

gotoxy(3,9); writeln('Deret bilangan prima : ');

b :=1;

x :=0;

repeat

if factor(b) = 2 then

begin

write(b,', ');

write(#8); {backspace}

inc(x);

end;

inc(b);

until (x=a);

readln;

end.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat36, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Bab 12

Pemrograman Dengan Teknik Rekursi

Tujuan Instruksional Khusus:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat mengerti tentang pemrograman dengan teknik rekursi pada Turbo Pascal, serta mampu mengimplementasikan pada permasalahan-permasalahan yang dihadapi.

Pertemuan ini akan menjelaskan tentang dasar rekursi, contoh proses rekursi, mengenal kata kunci forward, overload prosedur dan fungsi, serta program-program sederhana lainnya.

12.1 Pengertian Rekursi

Rekursi merupakan teknik pemanggilan sebuah rutin oleh dirinya sendiri. Istilah "rekursi" berasal dari bahasa Latin, "recursus", yang berarti "menjalankan ke belakang" atau "menjalankan kembali". Pada kasus pemrograman tertentu kita dapat menggunakan rekursi untuk menyederhanakan algoritma yang ada sehingga program yang kita tulis menjadi lebih efisien.

12.2 Contoh Proses Rekursi

Sebagai contoh untuk pembahasan rekursi, perhatikan terlebih dahulu contoh fungsi Faktorial sebagai berikut:

Function Faktorial(x: integer) : integer;

Var

i, hasil : integer;

Begin

Hasil := 1;

For i := x downto 1 do

begin

hasil := hasil * i;

end;

Faktorial := hasil;

End;

Fungsi di atas sebenarnya dapat kita sederhanakan dengan menggunakan rekursi, yaitu dengan kode sebagai berikut:

Function Faktorial(x: integer) : integer;

Begin

If (x=0) then

Faktorial := 1;

else

Faktorial := x * Faktorial(x-1);

End;

Pada kode tersebut terlihat bahwa fungsi Faktorial memanggil dirinya sendiri untuk melakukan proses perhitungan nilainya. Sebenarnya di dalam proses tersebut terdapat pengulangan sampai x bernilai 0. Apabila kita melewati nilai 5 sebagai parameternya, maka nilai yang akan dihasilkan adalah 120, yang berasal dari perhitungan di bawah ini:

$5 \times 4 \times 3 \times 2 \times 1 = 120$

Untuk kasus ini, proses yang akan terjadi di dalam rutin tersebut adalah sebagai berikut:

```

Faktorial(5) = 5 * Faktorial(4)
Faktorial(4) = 4 * Faktorial(3)
Faktorial(3) = 3 * Faktorial(2)
Faktorial(2) = 2 * Faktorial(1)
Faktorial(1) = 1 * Faktorial(0)
Faktorial(0) = 1
Faktorial(1) = 1 * 1
Faktorial(2) = 2 * 1
Faktorial(3) = 3 * 2
Faktorial(4) = 4 * 6
Faktorial(5) = 5 * 24
              = 120

```

12.3 Mengetahui Kata Kunci Forward

Dalam kasus-kasus pemrograman tertentu, terkadang kita menemui kejadian dimana dua buah rutin saling menggunakan satu sama lain. Apabila kita memiliki prosedur A dan B. Di dalam prosedur A kita memanggil prosedur B, begitu juga di dalam prosedur B kita akan memanggil prosedur A. Pertanyaan yang pasti muncul adalah prosedur mana yang harus didefinisikan terlebih dahulu? Prosedur A atau prosedur B? Perhatikan contoh pendefinisian dua buah prosedur berikut:

Procedure A;

Begin

B: {Salah, karena prosedur B belum dikenal}

End;

Procedure B;

Begin

A;

End;

Procedure B;

Begin

A: {Salah, karena prosedur A belum dikenal}

End;

Procedure A;

Begin

B;

End;

Hal ini terjadi karena di bahasa Pascal, kode akan dijalankan secara berurutan dari atas ke bawah. Dengan demikian apabila menggunakan prosedur B, sedangkan prosedur B belum didefinisikan di atasnya, maka hal ini akan menimbulkan masalah. Solusi untuk mengatasi kasus ini adalah dengan menggunakan kata kunci **forward** yang berfungsi untuk mendeklarasikan terlebih dahulu prosedur yang dimaksud tanpa harus menuliskan definisinya. Definisi prosedur yang baru akan diimplementasikan di bawah (harus dalam satu unit). Prosedur ini sering dinamakan dengan **prototyping** (pembuatan prototipe). Untuk itu kode di atas seharusnya ditulis seperti berikut:

Procedure B; forward;

Procedure A;

Begin

B: {Benar, karena prosedur B telah dikenalkan di atas}

End;

Procedure B;

Begin

A;

End;

12.4 Overload Prosedur dan Fungsi

Beberapa kompilator Pascal seperti Free Pascal, Pascal Pro dan kompilator Pascal gratis lainnya telah mendukung adanya overload prosedur atau fungsi, namun Turbo Pascal tidak mendukung fitur ini. Definisi overload sendiri adalah dua atau beberapa prosedur yang memiliki nama sama tetapi parameternya berbeda. Kata "berbeda" disini mengandung arti berikut:

1. Berbeda tipe datanya
2. Berbeda jumlah parameternya
3. Gabungan dari nomor 1 dan 2 (berbeda tipe data dan jumlah parameternya)

12.4.1 Overload dengan tipe data berbeda

Perhatikan contoh deklarasi fungsi di bawah ini:

Procedure Bagi(x, y: integer): integer;

Procedure Bagi(x, y: real): real;

Proses di atas disebut dengan overload fungsi, artinya fungsi di atas memiliki nama yang sama tetapi tipe parameternya berbeda. Fungsi yang pertama menggunakan tipe integer dan fungsi kedua menggunakan tipe real. Pada saat pemanggilan, kompilator secara otomatis akan memanggil fungsi yang sesuai dengan tipe parameter yang diisikan, misalnya:

Bagi(25, 5);

Maka fungsi yang akan diproses oleh kompilator adalah fungsi Bagi pertama karena kita mengisi parameter aktual di atas dengan tipe integer.

12.4.2 Overload dengan jumlah parameter berbeda

Pada bagian ini kita akan melakukan overload prosedur atau fungsi dengan menggunakan cara kedua, yaitu dengan mendefinisikan jumlah parameter yang berbeda. Perhatikan contoh deklarasi dua fungsi di bawah ini:

Function Maksimal(nilai1, nilai2: integer): integer;

Function Maksimal(nilai1, nilai2, nilai3: integer): integer;

Dari kode tersebut tampak bahwa parameter dari masing-masing fungsi bertipe sama, yaitu integer, Namun jumlah parameter dari setiap fungsi berbeda. Proses ini juga dikatakan sebagai overload fungsi. Pada saat pemanggilan, kompilator juga secara otomatis akan memilih fungsi yang sesuai dengan jumlah parameternya. Sebagai contoh kita memanggil fungsi dengan menggunakan kode berikut:

Maksimal(12, 25, 10);

Maka fungsi yang akan dipanggil adalah fungsi Maksimal kedua karena kita melewatkan tiga buah parameter.

12.4.3 Overload dengan tipe dan jumlah parameter berbeda

Proses overload prosedur atau fungsi jenis ini merupakan gabungan dari bentuk pertama dan kedua, artinya masing-masing parameter dari setiap rutin akan berbeda, baik tipe data maupun jumlahnya. Perhatikan contoh deklarasi prosedur-prosedur di bawah ini:

{prosedur yang memiliki 1 parameter bertipe string}

Procedure Tulis(s: string);

{prosedur yang memiliki 2 parameter bertipe integer}

Procedure Tulis(x, y: integer);

{prosedur yang memiliki 3 parameter bertipe real}

Procedure Tulis(a, b, c: real);

Ketiga prosedur di atas masing-masing memiliki parameter yang berbeda, baik tipe datanya maupun jumlahnya. Apabila kita memanggilnya dengan menggunakan kode berikut:

Tulis(21.05, 30.45, 3.33);

Maka yang akan diproses oleh kompiler adalah prosedur yang ketiga, yaitu prosedur yang memiliki tiga buah parameter dan bertipe real.

12.5 Praktek

Untuk kegiatan praktek pada kesempatan ini, diharapkan semua sudah berada pada kondisi jendela kerja Turbo Pascal. Selain itu sebaiknya juga telah melakukan perubahan direktori, agar program yang dibuat dapat disimpan direktori yang sudah disediakan agar mudah untuk diambil kembali.

1. Program Keduapuluhsembilan
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program RekursiFaktorial;

Uses

Crt;

Function Faktorial(x: integer): integer;

Begin

if (x=0) then

Faktorial := 1

Else

Faktorial := x + Faktorial(x-1)

End;

{Program Utama}

Var

Bil, hasil: integer;

Begin

Write('Masukkan bilangan yang akan dihitung : ');

Readln(bil);

Hasil := Faktorial(bil);

Write(bil, ' ! = ', hasil);

Readln;

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat29, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

2. Program Ketigapuluh
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program RekursiPerpangkatan;

Uses

Crt;

Function Pangkat(basis, eksp : integer) : integer;

Begin

If (eksp=0) then

Pangkat :=1

Else

Pangkat := basis * Pangkat(basis,eksp-1);

End;

{Program Utama}

Var

Hasil : integer;

Begin

Hasil := Pangkat(2,5); {Melakukan perhitungan 2^5}

Writeln('2^5 = ',hasil);

Readln;

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat30, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

Keterangan Proses atau alur proses yang terjadi dari program di atas:

```

Pangkat(2, 5) = 2 * Pangkat(2,4)
  Pangkat(2, 4) = 2 * Pangkat(2,3)
    Pangkat(2, 3) = 2 * Pangkat(2,2)
      Pangkat(2, 2) = 2 * Pangkat(2,1)
        Pangkat(2, 1) = 2 * Pangkat(2,0)
          Pangkat(2, 0) = 1
            Pangkat(2, 1) = 2 * 1
              Pangkat(2, 1) = 2 * 2
                Pangkat(2, 1) = 2 * 4
                  Pangkat(2, 1) = 2 * 8
                    Pangkat(2, 1) = 2 * 16

```

3. Program Ketigapuluhsatu
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

```
Program ContohForward1;  
Uses  
Crt;
```

```
Procedure B(x: integer) : integer;
```

```
Procedure A(x: integer);  
Begin  
If (x > 0) then  
begin  
Write('Pada saat x = ', x, ', prosedur A dijalankan');  
B(x-1); {Memanggil prosedur B}  
End;  
End;
```

```
Procedure B(x: integer);  
Begin  
If (x > 0) then  
begin  
Writeln('Pada saat x = ', x, ', prosedur A dijalankan');  
A(x-1); {Memanggil prosedur A}  
End;  
End;
```

```
{Program Utama}  
Begin  
Writeln('Memanggil prosedur A:');  
A(5); {Memanggil prosedur A dari program utama}  
Writeln;  
Writeln('Memanggil prosedur B:');  
B(5); {Memanggil prosedur B dari program utama}  
End.
```

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat31, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

4. Program Ketigapuluhdua
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program ContohForward2;

Uses

Crt;

Procedure Tulis(var N: integer) : forward;

Procedure TurunkanNilai(var indeks: integer);

Begin

Indeks := indeks - 1;

If indeks > 0 then

Begin

Tulis(indeks); {Memanggil prosedur Tulis}

End;

End;

Procedure Tulis(vr N: integer);

Writeln(N);

TurunkanNilai(N); {Memanggil prosedur TurunkanNilai}

End;

{Program utama}

Var

I: integer;

Begin

I := 5;

Writeln('Menggunakan prosedur TurunkanNilai');

TurunkanNilai(i);

Writeln;

I := 5;

Writeln('Menggunakan prosedur Tulis');

Tulis(i);

Readln;

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat32, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

5. Program Ketigapuluhtiga
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program Overload1; {Tipe data berbeda}

Uses

Crt;

Function Bagi(x, y: integer): integer;

Begin

Bagi := x div y;

End;

Function Bagi(x, y : real) : real;

Begin

Bagi := x / y;

End;

Var

Hasil1: integer;

Hasil2: real;

Begin

Hasil1 := Bagi(10, 3); {Menggunakan parameter bertipe integer}

Hasil1 := Bagi(10.0, 3.0); {Menggunakan parameter bertipe real}

Writeln(hasil1);

Writeln(hasil2:0:3);

Readln;

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat33, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

6. Program Ketigapuluhempat
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program Overload2; {Jumlah parameter berbeda}
Uses
Crt;

Function Maksimal(nilai1, nilai2: integer): integer;
Begin
 If (nilai1 > nilai2 then
 Begin
 Maksimal := nilai1;
 End
 Else
 Maksimal := nilai2;
 End;
End;

Function Maksimal(nilai1, nilai2, nilai3: integer): integer;
Begin
 If (nilai1 > nilai2) and (nilai1 > nilai3) then
 Begin
 Maksimal := nilai1;
 End
 Else
 Begin
 If (nilai2 > nilai1) and (nilai2 > nilai3) then
 Begin
 Maksimal := nilai2;
 End
 Else
 Begin
 Maksimal := nilai3;
 End;
 End;
 End;
End;

{Program Utama}
Var
Maks1, maks2: integer;
Begin
Maks1 := Maksimal(10, 20); {Menggunakan 2 parameter}
Maks2 := Maksimal(200, 72, 150); {Menggunakan 3 parameter}

{Menampilkan hasil}
Writeln('Nilai maksimal dari 10 dan 20 : ', maks1);
Writeln('Nilai maksimal dari 200, 75 dan 150 : ', maks2);
Readln;
End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat34, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.

7. Program Ketigapuluhlima
 - Tampilkan menu File
 - Pilih (sorot lalu ENTER) pada pilihan New
 - Ketiklah program seperti di bawah ini

Program Overload3; {Tipe dan jumlah parameter berbeda}

Uses

Crt;

Procedure Tulis(s: string);

Begin

Writeln(s);

End;

Procedure Tuli(x, y: integer);

Begin

Writeln(x, ' dan ', y);

End;

Procedure Tulis(a, b, c: real);

Begin

Writeln(a:0:2, ' ', b:0:2, ' ', dan ',c:0:2);

End;

{Program utama}

Begin

Tulis('Saya sedang belajar bahasa Pascal');

Tulis(34.12, 32.24, 10.78);

Readln;

End.

- Simpanlah program dengan cara: tampilkan menu File (ALT + F), lalu pilih Save (sorot lalu ENTER), ketikkan nama program yaitu : Lat35, lalu akhiri dengan ENTER.
- Jalankan program dengan cara menekan tombol ALT bersama-sama dengan huruf R (ALT + R), lanjutkan dengan memilih menu Run.