










:: MODUL 1 ::
SEQUENTIAL & PERCABANGAN/ BRANCHING
(RAPTOR)

1. SEQUENTIAL

Flowchart atau Diagram Alir merupakan salah satu metode untuk mempresentasikan Algoritma dengan menggunakan symbol-simbol bangun datar beserta anak panah sebagai penunjuk arah proses, berikut symbol-simbol yang akan digunakan ketika membuat flowchart di Microsoft Visio atau sejenisnya :

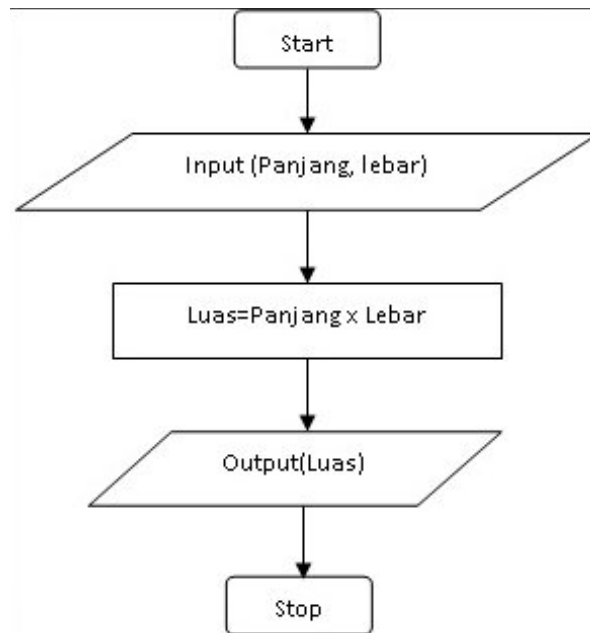
SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inialisasi/pemberian harga awal
	PROCESS	Proses perhitungan/proses pengolahan data
	INPUT/ OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Gambar 1
Simbol Flowchart

Contoh 1a :

Buatlah sebuah flowchart berdasarkan algoritma menghitung luas persegi panjang :

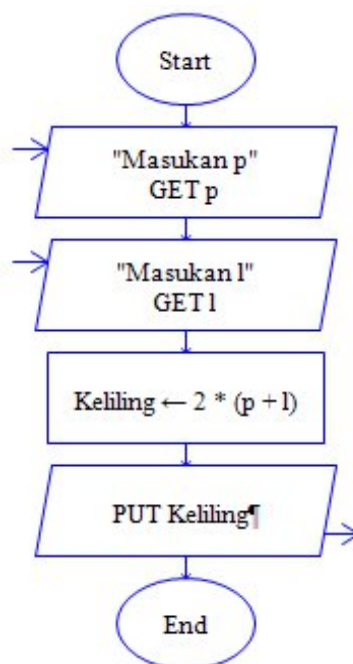
Jawab :



Contoh 1b :

Buatlah sebuah Flowchart dari Raptor untuk mencari Nilai Keliling Persegi Panjang ?

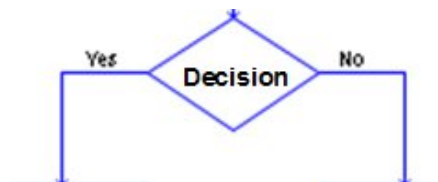
Jawab :



Berdasarkan dari contoh 1a dan 1b terlihat perbedaan pada jenis *Software* yang digunakan pada contoh 1a memakai M. Visio atau M.Word yang menghasilkan seperti gambar contoh 1a, tetapi pada contoh 1b pembuatan flowchart dikerjakan dengan menggunakan *Software Raptor* sehingga hasilnya bisa dilihat pada gambar 1b, dan pada Raptor flowchart tersebut bisa dijalankan atau di *compile*.

2.2 BRANCHING

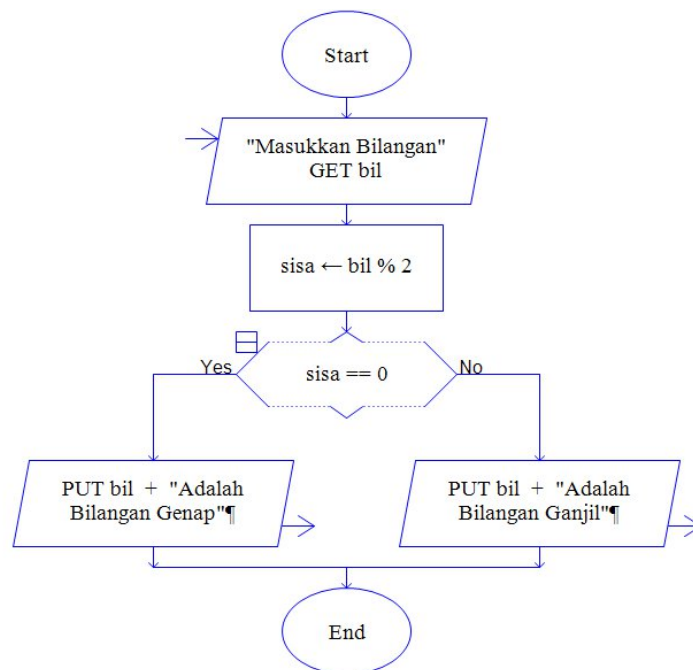
Branching atau percabangan merupakan struktur pengambilan keputusan yang digunakan untuk memilih salah satu alternatif jawaban yang tepat dari pilihan yang ada. Perintah percabangan / branching berguna untuk memilih salah satu dari 2 atau lebih alternatif jawaban yang tersedia.



Gambar 3
Struktur Branching pada Raptor

Kontrol seleksi seperti pada gambar 3 diatas itu akan memilih sebuah keputusan terhadap kondisi sebuah data dan melanjutkan eksekusi program ke salah satu dari dua alternatif jalur menuju statement berikutnya.

Contoh 1c :



Gambar 4

Flowchart Raptor Decision (Branching)

SOAL 1.1 :

1. Flowchart sebuah bilangan dan kemudian cetaklah nilai satuan, puluhan, dan ratusan dari bilangan tersebut.
2. Buatlah sebuah flowchart untuk Menghitung Luas dan Volume Balok
3. Buatlah sebuah flowchart untuk Menghitung Luas & Keliling dari
 - a. Segitiga
 - b. Lingkaran
4. Buatlah sebuah flowchart untuk menghitung konversi suhu celcius – fahrenheit dan fahrenheit - celcius
5. Buatlah sebuah flowchart untuk mencari nilai dari
 - 1) Mean
 - 2) Median
 - 3) Modus

SOAL 1.2 :

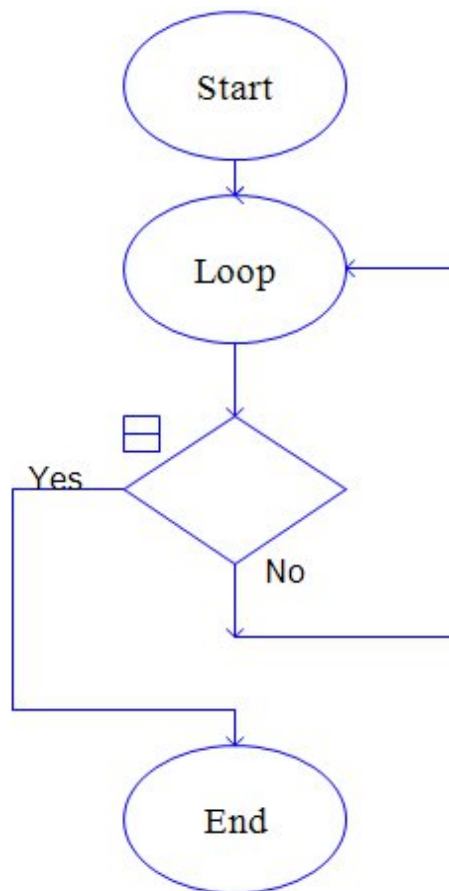
1. Buatlah Konversi dari :
 - A. Detik ke Menit (dan sebaliknya)
 - B. Menit ke Jam (dan sebaliknya)
 - C. Detik ke Jam (dan sebaliknya)
2. Buatlah dalam Satu Menu dari Soal 1.1 yang ada diatas !!!

Ketika memilih No 1 maka akan masuk ke Soal 1.1 No 1, Jika memilih No.2 maka akan masuk ke Soal 1.1 No 2, dan seterusnya, jika masukkan pilihan tidak ada maka muncul pesan “MAAF INPUTAN ANDA SALAH”.
3. Buatlah algoritma dan flowchart menggunakan raptor dari program prediksi jabatan berdasarkan nilai tes berikut :
 - Nilai 17 – 20 : Manager
 - Nilai 14 - 16 : Asisten Manager
 - Nilai 11 – 13 : Senior Eksekutif
 - Nilai 8 – 10 : Eksekutif
 - Nilai 6 – 8 : Staff
 - Nilai 1 – 5 : Operator

:: MODUL 2 ::

PERULANGAN/ LOOPING & NESTED LOOPING (RAPTOR)

Kontrol program perulangan memungkinkan program untuk mengulangi 1 atau lebih baris perintah sampai kondisi berhenti bernilai benar. Simbol perulangan pada RAPTOR sbb:



Gambar 5
Struktur Looping

Simbol keputusan (bentuk: wajib) menentukan perulangan di lanjutkan atau tidak. Jika keputusan bernilai “SALAH” maka perulangan berlanjut, tetapi jika bernilai “BENAR” maka perulangan berhenti.

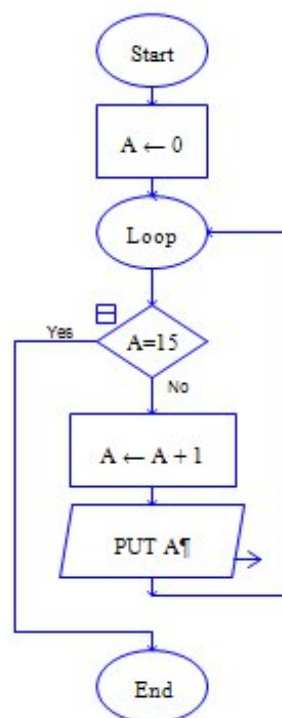
Contoh dari Looping

Menampilkan huruf "A" (tanpa tanda kutip) sebanyak 15 kali menggunakan Looping

Algoritma :

1. Beri nilai awal sama dengan 0 (Nol)
2. Dicek bahwa $0 \leq 15$, Jika True maka langsung ke END, Jika False maka lanjut ke $0 + 1$, sehingga nilainya berubah menjadi 1, setelah itu ditampilkan huruf "A" (tanpa tanda kutip)
3. Kembali ke atas untuk dicek apakah nilai $1 \leq 15$ Jika True maka langsung ke END, Jika False maka lanjut ke $1 + 1$, sehingga nilainya berubah menjadi 2, setelah itu ditampilkan huruf "A" (tanpa tanda kutip)
4. Ulangi langkah no 2 dan 3 sampai kondisi $16 \leq 15$ itu terpenuhi maka langsung ke END.

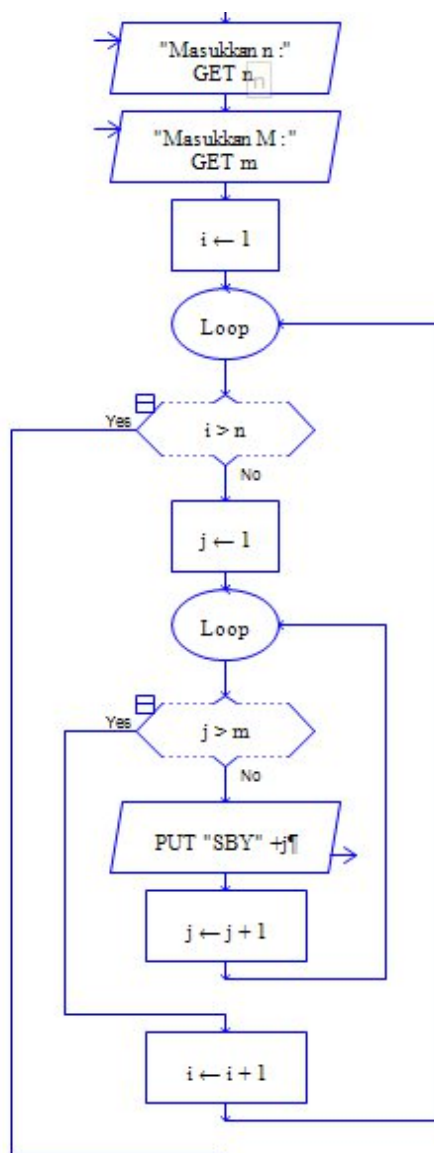
Flowchart dalam raptor :



Gambar 6
Flowchart Looping

Dalam Looping ini ada proses pengulangan / Looping ini mempunyai peran yang besar untuk memecahkan beberapa proses yang cukup panjang, sehingga dalam sebuah perulangan dapat muncul di dalam perulangan yang lain. Kondisi seperti ini disebut sebagai perulangan bersarang (nested looping). Jumlah “sarang” dapat lebih dari satu, menjadikan sebuah looping di dalam looping di dalam looping di dalam looping, dan seterusnya.

Perulangan bersarang di gunakan untuk melakukan iterasi pada data dengan dimensi 2 atau lebih. Contoh: data table dan matriks (dimensi baris dan kolom), data koordinat Cartesian 3 dimensi (sumbu x,y, dan z), data array berdimensi tiga, empat,lima, dan seterusnya. Selain itu, perulangan Bersarang juga dapat di gunakan dalam sebuah program yang telah mengandung sebuah looping didalamnya untuk mengulangi kembali proses perhitungan, sebelum program keluar/selesai. Variasi perulangan bersarang pada RAPTOR dapat di gambarkan sebagai berikut :



Pada gambar 7 flowchart disamping dijelaskan bahwa membuat sebuah tampilan atau oupt yang di inginkan adalah menampilkan “SBY” (tanpa tanda kutip) sebanyak 15x, dimana pada pembuatan Flowchart di raptor tersebut memakai kondisi I dan J sebagai Decision yang akan memberikan Kondisi supaya tidak terjadi Looping yang tidak kunjung berhenti. Pada variable N dan M sebagai nilai inputan Sembarang serta nilai batasan untuk proses Nested Looping yang akan diproses nantinya. Dalam sebuah permasalahan yang serupa akan dimungkinkan seorang individu akan memiliki Algoritma yang berbeda untuk membuat sebuah Solusi yang akan dipakai untuk bisa menampilkan output yang dipakai.

Soal 2.1 :

1. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang akan menghasilkan karakter (*) membentuk gambar berikut:

Inputan : 4

Output : * * * *
 * * *
 * *
 *
 * *
 * * *
 * * * *

2. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat mementuk formasi seperti buku

Input : 4

Output : 4

3 3

2 2 2

1 1 1 1

3. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat membentuk formasi seperti berikut :

Input : 4

Output : 1

1 2

1 2 3

1 2 3 4

4. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat membentuk formasi seperti berikut :

Input : 4

Output: *

* *
* * *
* * * *
* * *
* *
*

5. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat menerima dan menjumlahkan data matriks 2x2 berdasarkan inputan user.
6. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat membentuk formasi seperti berikut :
Input : 4
Output : $1 + 2 + 3 + 4 = 10$
7. Modifikasi soal no 3 dari modul 4 dengan memberikan fitur untuk mengulangi proses perhitungan sekali lagi atau tidak. Jika user memilih Ya maka proses perhitungan akan diawali kembali, Jika Tidak maka Program Flowchart langsung END.

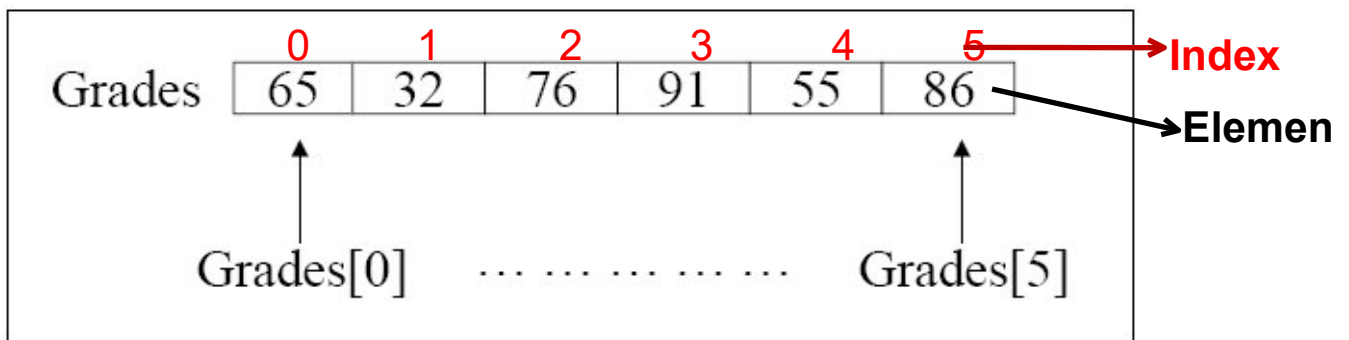
Note : *Input yang tertera pada soal diatas merupakan Inputan User sehingga Nilainya bisa berubah-ubah*

:: MODUL 3 ::

ARRAY/ ARRAY LANJUT & PROSEDUR/ FUNGSI (RAPTOR)

Array adalah struktur data yang menyimpan sekumpulan elemen yang sama. Setiap elemen di akses langsung melalui indeksnya. Dengan kata lain array memungkinkan penyimpanan data dalam 1 variabel yang diatur berdasarkan indeks. Contoh sederhana penggunaan array misalnya adalah untuk menyimpan sederetan angka (mis. 5 angka). Jika kita menggunakan konsep satu variabel satu data, maka kita memerlukan 5 variabel yang berbeda untuk menyimpan data tersebut. Tetapi jika di gunakan array, maka cukup menggunakan 1 nama variabel untuk menyimpan data.

Contoh penulisan inisialisasi pada Array :



Gambar 8

Struktur Array

Pada gambar 8 diatas dijelaskan bahwa sekumpulan array ini memiliki 6 Element dan memiliki nama Array “Grades”, dimana Nilai dari 6 Element itu adalah

“ Grades[0] = 65 ; Grades[1] = 32 ; Grades[2] = 76 ; “

“ Grades[3] = 91 ; Grades[4] = 55 ; Grades[5] = 86 ; “

Contoh 3.1 :

Buat algoritma dan flowchart untuk mengisi sebuah array yang dapat menyimpan maksimal 3 data dan kemudian tampilkan ke layar isi array tersebut ... (Flowchart bisa dilihat di Flowchart 3.1)

Algoritma :

1. Mulai
2. Inisialisasi maksimal data yang bias di simpan ke array maks=3
3. Inisialisasi variable indeks $i=1$
4. Jika $i > \text{maks}$ bernilai SALAH, maka
 - 1.1 masukkan data array indeks # i , simpan ke array $A(i)$ kemudian tampilkan ke layar data array $A(i)$
 - 1.2 Tambah nilai variable indeks i , $i=i+1$
 - 1.3 Ulangi langkah 3
5. Jika $i > \text{maks}$ bernilai BENAR, maka SELESAI

Contoh 3.2

Buat algoritma dan flowchart untuk mengisi sebuah array 2 dimensi yang dapat menyimpan 9 data dengan ukuran 3 baris dan 3 kolom. Kemudian tampilkan ke layar isi array tersebut ... (Flowchart bisa dilihat di Flowchart 3.2)

Algoritma:

1. Mulai
2. Inisialisasi maksimal data yang bias di simpan ke array maks=3
3. Inisialisasi variable indeks baris=1
4. Jika baris $> \text{maks}$ bernilai SALAH, maka
 - 4.1 inisialisasi variabel indeks kolom=1
 - 4.2 jika kolom $> \text{maks}$ bernilai SALAH, maka
 - 4.2.1 masukan data array indeks #baris #kolom, simpan ke array $A(\text{baris}, \text{kolom})$
 - 4.2.2 kemudian tampilkan ke layar data array $A(\text{baris}, \text{kolom})$
 - 4.2.3 tambah nilai variabel indeks baris, $\text{baris}=\text{baris}+1$
 - 4.2.4 ulangi langkah 3.2
 - 4.3 jika kolom $> \text{maks}$ bernilai BENAR, maka lanjutkan langkah 4
5. Tambah nilai variabel indeks baris, $\text{baris}=\text{baris}+1$
6. Jika baris $> \text{maks}$ bernilai BENAR, maka SELESAI

Pembahasan selanjutnya mengenai Modul **PROSEDUR**, dimana Prosedur merupakan blok dari kode yang di rancang untuk melakukan tugas khusus. Adapun tujuan dari pembuatan suatu prosedur adalah :

1. Program menjadi terstruktur, program yang besar dapat di pecah menjadi beberapa program yang kecil, yang tiap satu program tersebut mempunyai tugas tertentu.
2. Mengurangi duplikasi kode.
3. Prosedur dapat di panggil dari program atau prosedur yang lain.
4. Menghemat ukuran program, manfaat ini akan terasa kalau ada beberapa deretan intruksi yang sama di gunakan pada beberapa tempat didalam program.
5. Memudahkan dalam mengembangkan program, halini merupakan kunci dalam pembuatan program terstruktur.

Pada umumnya prosedur memerlukan masukan yang dinamakan argument atau parameter. Hasil akhir prosedur akan berupa sebuah nilai (nilai balik fungsi), yang akan di kirim kembali ke bagian flowchart yang memanggil.

Untuk membuat prosedur klik kanan pada tab **main** kemudia pilih **add procedure**. Maka akan tampil form untuk member nama prosedur dan membuat parameter dari prosedur seperti yang tampak pada Gambar 3.3. Setiap parameter prosedur dapat berfungsi sebagai :

1. Parameter input: parameter yang nilainya sudah diberikan pada sat pembagian prosedur.
2. Parameter input/output : parameter yang berfungsi ganda, sebagai parameter input dan parameter output.

main

Create Procedure

Names must begin with letter, and contain only letters, numbers and underscores.

Examples:
Draw_Boxes
Find_Smallest

Procedure Name

Parameter 1 (or blank) ☒ Input ☐ Output

Parameter 2 (or blank) ☒ Input ☐ Output

Gambar 9
Add Procedure

Buat algoritma dan flowchart terbagi menjadi 2 bagian yaitu algoritma program utama (main) dan prosedur hitung luas. Prosedur hitung luas memiliki 3 parameter, yaitu 2 parameter input dan 1 parameter output. Algoritma utama memanggil prosedur hitung luas dengan memberikan nilai dari parameter input prosedur.

Contoh 3.3 :

Algoritma program utama (main)

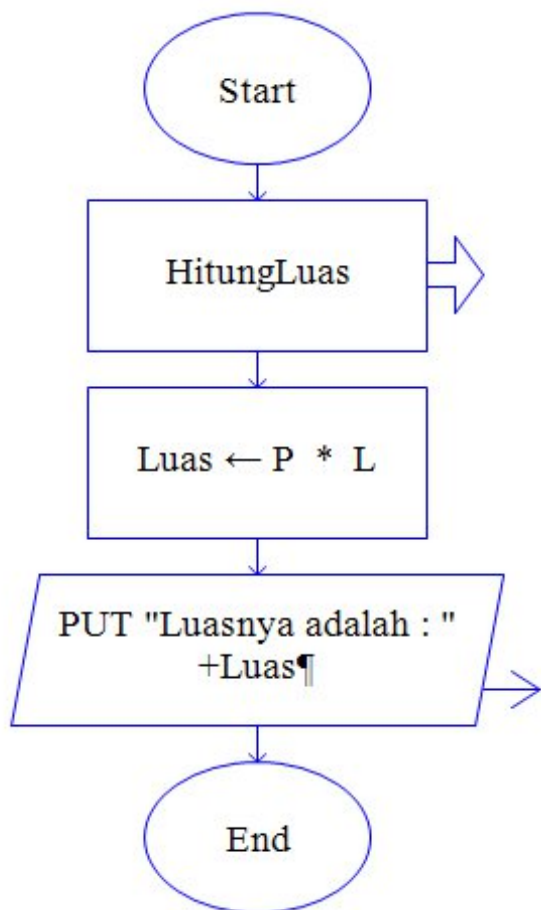
1. *Mulai*
2. *Input panjang, lebar persegi panjang*
3. *Panggil prosedur hitung luas (panjang,lebar,luas)*
4. *Cetak hasil luas*
5. *Selesai*

Algoritma prosedur hitung luas :

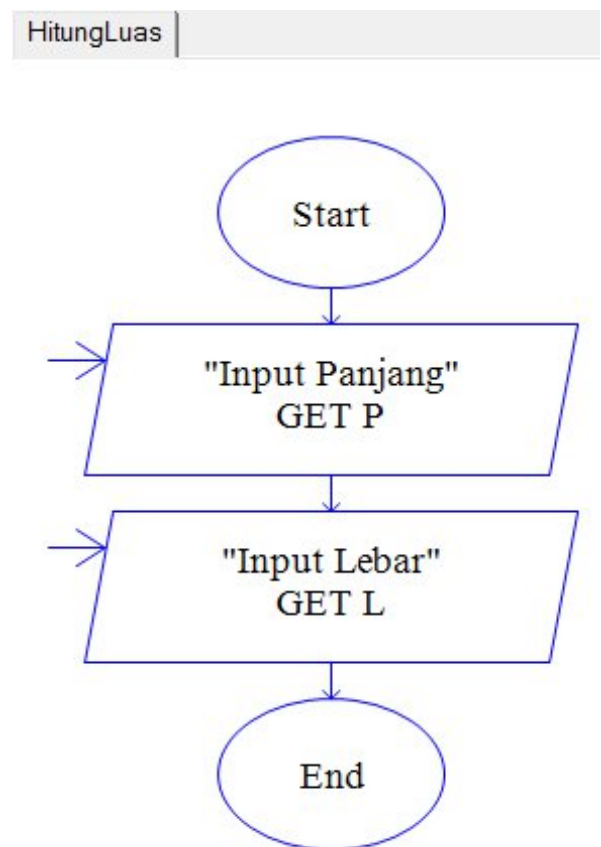
1. Mulai. Menerima nilai parameter Hitung Luas (P,L,Luas)
2. Hitung luas persegi panjang
$$\text{Luas} = P * L$$
3. Selesai

Flowchart :

Flowchart program utama lihat gambar 10(a) dan prosedur HitungLuas pada gambar 11(b)



Gambar 10(a)



Gambar 10(b)

SOAL 3 :

1. Buatlah algoritma dan flowchart menggunakan raptor untuk mencari nilai tertinggi, nilai terendah, dan rata rata pada suatu array ?
2. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat mencatat data suhu dan kelembaban dari sebuah rumah kaca selama satu minggu (7 hari).
3. Tambahkan perhitungan rata rata suhu dan kelembaban dari flowchart yang anda telah buat pada nomor 2 di atas.
4. Buatlah algoritma dan flowchart menggunakan raptor untuk menampilkan bilangan Fibonacci pada deret ke-n! Bilangan Fibonacci adalah bilangan seperti : 1 1 2 3 5 8 13 ... dst jadi jika inputan inputan $n=7$ maka hasil adalah 13.
5. Buatlah algoritma dan flowchart menggunakan raptor untuk menghitung estimasi total biaya penjualan pada sebuah apotik.
6. Buatlah dua buah array (A dan B) dengan jumlah anggota array (n) ditentukan oleh user, kedua array tersebut harus memiliki jumlah anggota yang sama. kemudian lakukan operasi-operasi berikut, dan simpan hasilnya dalam sebuah array yang laen.
 - a. $A+B$
 - b. $A*B$
 - c. A/B
 - d. Cari nilai yang terbesar dan terkecil dari A dan B
 - e. Balik urutan nilai A, dan jumlahkan dengan array B
7. Buatlah algoritma dan flowchart menggunakan raptor untuk mengimputkan data mahasiswa nama, npm, dan jurusan dengan menggunakan prosedur.
8. Buatlah Sebuah Menu dari Soal 3 No 1 – 7 (Atau Soal yang dipilih pada saat Praktikum) Menggunakan Prosedur dari masing-masing fungsi.

:: MODUL 4 ::

PERCABANGAN / BRANCHING

(BAHASA C)

4.1 Sequential

Secara umum langkah-langkah dalam membuat program baru adalah sebagai berikut:



Pengarah praprosesor	<pre>#include<stdio.h> main() { //perintah C dimulai dari sini }</pre>
-------------------------	---

Saat anda memasuki layar C (Dev C++) anda bisa menekan tombol file dan tekan new untuk memulai pembuatan program sesuai petunjuk. Saat program telah usai tekan compile untuk melihat letak kesalahan pada program yang telah anda kerjakan, dalam compile terdapat petunjuk di mana letak baris kesalahan yang anda perbuat. Setelah program non error anda bisa melihat tampilannya dengan menekan run.

Dalam program sequential ini anda dapat membuat program dengan memasukkan rumus dari program tersebut sehingga saat menampilkan hasil kita hanya memasukkan sebuah angka dan hasil pun akan langsung tampil sesuai rumus dan bangun yang ditentukan.

Contoh program sequential ini :

```
{
    int .....;
    printf("masukkan panjang : "); scanf("%d", &panjang);
    printf("masukkan lebar : "); scanf("%d", &lebar);
    luas = panjang*lebar;
    printf("jadi luas bangun tersebut : %d", luas);
}
```

Bila program di atas dijalankan akan terdapat hasil :

<code>masukkan panjang</code>	<code>:</code>	<code>7</code>
<code>masukkan lebar</code>	<code>:</code>	<code>3</code>
<code>jadi luas bangun tersebut</code>	<code>:</code>	<code>21</code>

Dari contoh program di atas, bisa dilihat bahwa untuk memasukkan variable panjang dan lebar ke dalam program menggunakan fungsi pustaka scanf dan semuanya itu dimasukkan melalui keyboard. Sedangkan fungsi pustaka printf digunakan untuk menampilkan hasil yang terformat dengan menggunakan format specification. Penggunaan %d hanya digunakan untuk bilangan integer sedangkan dan setiap kata yang berada dalam printf harus diawali tanda “ dan diakhiri pula tanda “.

Jadi inti dalam sequential programming adalah bahwa sebuah program itu dikerjakan dengan jalan berurutan langkah-langkahnya. Setahap demi setahap. Secara umum, diawali dengan input kemudian dilanjutkan dengan pengolahan input. Setelah itu hasilnya akan dioutputkan kembali kepada user.

SOAL LATIHAN

1. Buatlah program dengan memasukkan nilai tes mata kuliah kemudian menampilkan hasil nilai rata-rata. Misal diasumsikan ada 10 mata kuliah yang diambil oleh mahasiswa.
2. Buatlah program untuk menghitung jarak peluru yang ditembakkan dari suatu lokasi dengan sudut penembakan (sudut) dan kecepatan (vo) dimasukkan dari keyboard. Jarak yang diperoleh ditampilkan ke layar monitor.
3. Buatlah program untuk menghitung luas dan keliling lingkaran kemudian ditampilkan ke layar monitor.
4. Tulislah program sekuensial untuk menghitung sisi miring, sinus, cosinus, dan tangent dari suatu segitiga dengan masukan dari pengguna berupa panjang alas dan tingginya.

5. Buatlah program untuk menghitung Luas dan Keliling Segi Tiga, Bujur Sangkar dan Persegi Panjang.
6. Buatlah program untuk menghitung Volume dan Luas Permukaan Kubus, Limas dan Prisma.
7. Buatlah program untuk menghitung $Y = 5X^4 + 3X^3 - X^2 + 7$.
8. Buatlah program untuk menghitung $D = 2A^4 - 6B^3 + 8C^2 - 10$.
9. Buatlah program untuk menghitung $Z = \sin 2X + 3 \cdot \cos 3X - 5 \cdot \tan 4X$.
10. Buatlah program untuk menghitung $S = S_0 + V_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2$

4.2 Branching / Percabangan

Struktur pengambilan keputusan (decision) yang digunakan untuk memilih salah satu alternatif jawaban yang tepat dari pilihan yang ada. Dimana dalam bahasa C menyediakan perintah pengambilan keputusan yaitu perintah IF dan perintah SWITCH.

Perintah **IF** berguna untuk memilih salah satu dari 2 atau lebih alternatif jawaban yang tersedia. Jika perintah IF tersebut terdiri dari 2 atau lebih pernyataan, maka pernyataan tersebut harus berada di antara begin dan end.

Bentuk IF dibagi tiga:

- Perintah **IF**.

<pre> IF (kondisi) { pernyataan 1 pernyataan 2 pernyataan_n } </pre>
--

Bentuk di atas mempunyai arti jika kondisi bernilai benar maka pernyataan di bawahnya akan dikerjakan. Jadi perintah IF merupakan perintah yang menentukan pengambilan keputusan bahwa pernyataan itu akan dikerjakan atau tidak.

▪ Perintah **IF.....ELSE**

IF (kondisi)

{

 pernyataan 1

 pernyataan 2

 pernyataan_m

}

ELSE

{

 pernyataan 4

 pernyataan 5

 pernyataan_n

}

Bentuk diatas mempunyai arti jika kondisi benar maka pernyataan 1 s/d m akan dikerjakan dan jika kondisi salah maka pernyataan 4 s/d n yang akan dikerjakan.

Perintah IF dalam IF yang mana sering disebut **nested-if**

Bentuk umum :

IF (kondisi 1)

 IF (kondisi 2)

 pernyataan 1;

 ELSE

 pernyataan 2;

ELSE

 pernyataan 3;

Pernyataan IF dalam IF juga mempunyai bentuk yang majemuk yang sering disebut IF bertingkat.

Bentuk umumnya :

```
IF (kondisi 1)
    pernyataan 1;
ELSE IF (kondisi 2)
    pernyataan 2;
ELSE IF (kondisi 3)
    pernyataan 3;
    .....
    .....
    .....
ELSE
    pernyataan_n;
```

Bentuk diatas mempunyai arti jika kondisi 1 benar maka pernyataan 1 yang akan dikerjakan dan apabila kondisi 1 tidak benar maka kondisi 2 yang akan dilihat. Jika kondisi 2 benar maka pernyataan 2 yang akan dikerjakan dan apabila kondisi 2 tidak benar maka kondisi 3 yang akan dilihat sampai pernyataan terakhir sesuai bentuk pernyataan diatas.

Perintah **SWITCH** merupakan pernyataan yang dirancang untuk menangani pengambilan keputusan yang melibatkan sejumlah pilihan alternatif yang diantaranya untuk menggantikan pernyataan IF bertingkat.

Bentuk umumnya adalah sebagai berikut

```
SWITCH (ekspresi )
{
    case konstanta 1:
        pernyataan 1;
        break;
    case konstanta 2:
        pernyataan 2;
        break;
    case konstanta 3:
        pernyataan 3;
```

```

break;
.....
.....
.....
case konstanta_n:
    pernyataan_n;
break;
default;
    pernyataan_x;
}

```

Bentuk diatas dapat dijelaskan bahwa setiap konstanta 1 s/d konstanta_n dapat berupa konstanta integer atau konstanta karakter. Arti pernyataan tersebut adalah dimulai dari konstanta 1 apabila nilainya cocok dengan ekspresi maka pernyataan 1 akan dijalankan. Kalau tidak cocok maka akan beralih ke pernyataan 2 dan seterusnya sampai hasil yang diinginkan.

Perintah break menyebabkan eksekusi yang diarahkan ke akhir switch kalau nilai konstanta 1 tidak sama dengan nilai ekspresi dan kemudian baru diteruskan pengujian dengan konstanta 2 dan seterusnya. Jika sampai pada pengujian akhir tidak ada yang cocok maka default akan dijalankan.

Jika pernyataan break tidak diikutsertakan pada setiap case maka walaupun konstantanya cocok dengan ekspresi, maka setelah pernyataan pada case tersebut dikerjakan akan dilanjutkan ke case berikutnya sampai ditemui pernyataan break pada akhir switch.

Contoh program if

```

/* Program : cabang1.c */
#include "stdio.h"
main( )
{
    int nilai;
    printf("masukkan sebuah nilai :");
    scanf("%d",&nilai);
    if (nilai > 70)
        printf("lulus\n");
}

```

```

        else
            printf("tidak lulus\n");
    }

```

Contoh program switch

```

/* Program : case.c */
#include "stdio.h"
main( )
{
    int jen_bunga;
    printf("masukkan jenis bunga [1-3]:");
    scanf("%d",&jen_bunga);
    switch (jen_bunga)
    {
        case 1:
            printf("Anggrek");
            break;
        case 2:
            printf("Mawar");
            break;
        case 3:
            printf("Melati");
            break;
        default:
            printf("Jenis tidak ada");
    }
}

```

SOAL LATIHAN

1. Buatlah sebuah program untuk menyeleksi kode bulan. Bila user menginputkan angka yang merepresentasikan bulan yang dimaksud (yakni 1 s/d 12), maka output akan menampilkan tulisan huruf dari bulan tersebut (januari s/d desember). Dan bila user menginputkan angka selain yang dimaksud, maka output akan menampilkan pernyataan bahwa user telah salah menginputkan data.
2. Buatlah program dengan menggunakan branching dengan inputan : umur anda dan output : anda
 - 1 – 16 tampilkan “anak-anak”
 - 17 – 25 tampilkan “remaja”
 - 26 – 45 tampilkan “dewasa”
 - ≥ 45 tampilkan “tua banget”
3. Tulislah program untuk menganalisa nilai angka mahasiswa yang diinputkan dan outputnya berupa nilai huruf (A, B, C, D, E).
4. Tulislah program permainan menebak sebuah angka integer yang berkisar antara 1 hingga 99 yang telah ditetapkan, berikan batasan kesempatan menebak, dan tunjukkan pada pengguna dalam berapa kesempatan ia berhasil menebak.
5. Buatlah program yang mempunyai pilihan menu untuk menghitung beberapa macam rumus yang berbeda dengan menggunakan statemen **case-break** dan **if-else**.
6. Buatlah sebuah konversi (Celcius ke Rearmur, Farenheit, dan Kelvin serta kebalikkannya), dengan menggunakan menu dari tiap-tiap konversi tersebut.

:: MODUL 5 ::

PERULANGAN/ LOOPING & NESTED LOOP (BAHASA C)

Perintah pengulangan di gunakan untuk melakukan suatu proses secara berulang –ulang jika suatu kondisi dipenuhi atau sebaliknya. Mengulang suatu proses merupakan tindakan yang banyak dijumpai dalam suatu pemrograman. Pada semua bahasa pemrograman pengulangan proses ditangani dengan suatu mekanisme yang disebut LOOP. Misal , menampilkan tulisan sebanyak mungkin pada layar dapat diimplementasikan dengan kode program yang pendek. Pada bahasa C terdapat tiga buah struktur loop, yaitu berupa FOR, DO – WHILE, & WHILE.

1. Perulangan FOR

Pernyataan ini digunakan bila anda sudah tau berapa kali anda akan

mengulang satu atau beberapa pernyataan.

Contoh :

```
for (nilai = 20 ; nilai <= 100 ; nilai + = 10)
    printf ( " % d \ n " , nilai );
```

2. Perulangan DO – WHILE

Perintah ini menyatakan perulangan proses selama kondisi tertentu.

Contoh :

```
counter = 1;
do
{
    printf ( " C + + \ n " ) ;
    counter + + ;
} while (counter < 15) ;
```

3. Perulangan WHILE.

Perintah WHILE sama dengan perintah DO – WHILE, hanya pengujian kondisinya terletak pada awal loop.

Contoh :

```
i = 30;
while ( i > 0 )
{
    printf ( " % d \ n " , i ) ;
    i - - ;
}
```

SOAL LATIHAN

1. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang akan menghasilkan karakter (*) membentuk gambar berikut:

Input : 4

Output : * * * *
* * *
* *
*
* *
* * *
* * * *

2. Buatlah algoritma dan flowchart menggunakan raptor untuk sebuah program yang dapat membentuk formasi seperti buku

Input : 4

Output : a) 4	b) 4	c) 4 4 4 4
4 3	3 3	3 3 3
4 3 2	2 2 2	2 2
4 3 2 1	1 1 1 1	1

3. Buatlah program yang mempunyai fungsi berparameter untuk menghitung N! (N faktorial).

$N! = N \times (N - 1) \times (N - 2) \times \dots \times 1$.

Contoh: $4! = 4 \times 3 \times 2 \times 1 = 24$

N adalah inputan dari user. Nilai maksimal N adalah 7. Hasil perhitungan ditampilkan ke layar. Buat menu pilihan untuk menawarkan kepada user apakah ingin mengulangi lagi prosesnya ataukah tidak.

4. Buatlah program yang memiliki fungsi untuk menampilkan N (= inputan user) bilangan pertama deret Fibonacci. Deret Fibonacci adalah: 0, 1, 2, 3, 5, 8, 13, 21, 34, ...

Fibbo ke N = Fibbo ke (N – 1) + Fibbo (N – 2)

Default: Fibbo ke 1 = 0, Fibbo ke 2 = 1

Contoh: Input : N = 5

Output : 0 1 2 3 5

5. Buatlah program FibboRial yaitu yang menggabungkan konsep Fibonacci dan Faktorial dengan cara memfaktorialkan deret angka fibonacci.

Contoh: Input : N = 5

Output : 1 1 2 6 120

:: MODUL 6 ::

ARRAY/ LARIK

(BAHASA C)

Array adalah struktur data yang menyimpan sekumpulan elemen yang sama. Setiap elemen diakses langsung melalui indeksnya. Indeks larik haruslah tipe data yang menyatakan keurutan misalnya interger atau karakter.

Mendefinisikan Array

Mendefinisikan array berarti :

1. Mendefinisikan banyaknya elemen array.
2. Mendefinisikan tipe elemen array.

contoh algoritma pendeklarasian array :

DEKLARASI

Const Nmaks = 15

Harga_barang : array [1..Nmaks] of interger

Karena N dinyatakan sebagai konstanta yang bernilai 15

Pemrosesan Array

Algoritma SKEMA UMUM pemrosesan array.

{memproses setiap elemen array secara berurutan mulai dari indeks terkecil sampai indeks terbesar}

DEKLARASI

Const Nmaks = 100 {banyaknya eleent array}

Type array int : array [1 Nmaks]of interger

A: array int

K: interger { indeks array}

DESKRIPSI:

Inisialisasi nilai

```
k ← 1 {mulai dari element pertama}
while k ≤ Nmaks do
    pemrosesan terhadap A[k]
    k ← k + 1 {tinjau elemen berikutnya }
endwhile
{k > Nmaks}
tereliminasi
```

Mencari nilai maksimum array

Nai maksimum pada array adalah element yang mempunyai nilai terbesar diantara elemen lainnya. Kadangkala algoritma mencari nilai maksimum array dinyatakan sebagai fungsi:

function maksimum 1(input A : arrayint , input N : interger) → interger
{ *mengembalikan elemen terbesar di dalam array A[1..N]*}

DEKLARASI

K : integer {pencatat indeks array }

DESKRIPSI

Maks ← - 9999 {nilai maksimum pertama}

```
For k ← 1 to N do
    If A[k] > maks then
        maks ← A[k]
    endif
endfor
return maks
```

POINTER

Pointer (variabel penunjuk) adalah suatu variabel yang berisi alamat memori dari suatu variabel lain. Alamat ini merupakan lokasi dari obyek lain (biasanya variabel lain) di dalam memori. Contoh, jika sebuah variabel berisi alamat dari variabel lain, variabel pertama dikatakan menunjuk ke variabel kedua. Operator Pointer ada dua, yaitu :

- a. Operator &
Operator & bersifat unary (hanya memerlukan satu operand saja).
Operator & menghasilkan alamat dari operandnya.
- b. Operator *
Operator * bersifat unary (hanya memerlukan satu operand saja).
Operator * menghasilkan nilai yang berada pada sebuah alamat.

DEKLARASI POINTER

Seperti halnya variabel yang lain, variabel pointer juga harus dideklarasikan terlebih dahulu sebelum digunakan. Bentuk Umum :

Tipe_data *nama_pointer;

Tipe data pointer mendefinisikan tipe dari obyek yang ditunjuk oleh pointer. Secara teknis, tipe apapun dari pointer dapat menunjukkan lokasi (dimanapun) dalam memori. Bahkan operasi pointer dapat dilaksanakan relatif terhadap tipe dasar apapun yang ditunjuk. Contoh, ketika kita mendeklarasikan pointer dengan tipe `int*`, kompiler akan menganggap alamat yang ditunjuk menyimpan nilai integer - walaupun sebenarnya bukan (sebuah pointer `int*` selalu menganggap bahwa ia menunjuk ke sebuah obyek bertipe integer, tidak peduli isi sebenarnya). Karenanya, sebelum mendeklarasikan sebuah pointer, pastikan tipenya sesuai dengan tipe obyek yang akan ditunjuk.

Contoh :

```
int *px;  
char *sh;
```

Contoh Program

```
#include "stdio.h"  
#include "conio.h"  
#include "stdlib.h"  
void main()  
{   int x, y; /* x dan y bertipe int */  
  
    int *px; /* px pointer yang menunjuk objek */  
    system ("CLS");  
    x = 87; px = &x; /* px berisi alamat dari x */  
    y = *px; /* y berisi nilai yang ditunjuk px */  
    printf("Alamat x = %p\n", &x);  
    printf("Isi px      = %p\n", px);  
    printf("Isi x       = %i\n", x);  
    printf("Nilai yang ditunjuk oleh px = %i\n", *px);  
    printf("Nilai y      = %i\n", y);      getch();
```

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{    static int tgl_lahir[] = { 13,9,1982 };    }

int *ptgl;

ptgl = tgl_lahir;    /* ptgl berisi alamat array */
printf("Diakses dengan pointer");           printf("Tanggal    =
%i\n", *ptgl);
printf("Bulan          = %i\n", *(ptgl + 1));
printf("Tahun          = %i\n", *(ptgl + 2));
printf("\nDiakses dengan array biasa\n");
printf("Tanggal = %i\n", tgl_lahir[0]);
printf("Bulan   = %i\n", tgl_lahir[1]);
printf("Tahun   = %i\n", tgl_lahir[2]);           getch();
```

SOAL LATIHAN

1. Tulislah program untuk menginput dan output data pemilih dalam pemilu presiden. Data yang diinput berupa nama, no.KTP, dan calon yang dipilih. Data yang disimpan dalam bentuk array.
2. Buatlah program dengan menerima inputan "Teknik Informatika" dari user kemudian output berupa tampilan bentuk segitiga.

Contoh:

```
Teknik Informatika
Teknik_
Tekni
Tekn
Tek
Te
T
```

3. Buat sebuah program fungsi untuk melakukan pencarian terhadap suatu nilai yang berupa array (didefinisikan terlebih dahulu) dimana jika ingin mencari tinggal menginput dari keyboard no kode yang terdaftar sehingga layar akan mengeluarkan kode produk, stock barang dan harga suatu barang.

Contoh output:

```
** Program pencarian inventory **
```

```
Enter kode barang: 45
```

```
Kode produk      : 45
```

```
stock barang     : 2
```

```
Harga            : Rp 6500
```

```
Press any key to continue
```

4. Buatlah program statistik (input berupa array yg sudah terinisialisasi bisa pada saat design time maupun otomatis) yang mempunyai fungsi untuk:
 - a. Menghitung nilai rata-rata.
 - b. Menghitung modus (nilai yang paling sering muncul).
5. Buatlah program untuk menggabungkan 2 buah array berdimensi satu.
6. Buatlah program dengan tipe data string (array of char) untuk melakukan:
 - a. Insert karakter pada indeks tertentu
 - b. Insert array pada indeks tertentu
 - c. Pencarian substring
 - d. Menggabungkan dua buah string
7. Buatlah program untuk menghitung tinggi pantulan bola yang jatuh, dengan asumsi tinggi pantulan bola adalah 0.8 kali tinggi sebelumnya. Tinggi awal bola jatuh diisikan melalui keyboard. Contoh tampilan program terlihat di bawah ini. Maksimal perhitungan adalah sampai pantulan ke 20 atau apabila tinggi pantulan sudah di bawah 1.

:: MODUL 7 ::

OPERASI FILE & PROSEDUR/ FUNGSI (BAHASA C)

OPERASI FILE

Dalam pemrograman, operasi-operasi yang dapat dilakukan pada file adalah input/output file dan manipulasi file. Operasi I/O (input/output) melibatkan pembacaan file (input) dari file atau perekaman (output) ke file. Manipulasi file melibatkan operasi terhadap suatu file tanpa melakukan proses input atau output, misalnya mengecek keberadaan file di disk, mengganti nama file, menghapus file, dan sebagainya.

Dalam bahasa C / C++ menyediakan class-class berikut untuk melakukan output dan input karakter-karakter dari/ke file-file.

- a. `ofstream`: class untuk menulis karakter ke file-file.
- b. `ifstream`: class untuk membaca karakter dari file-file.
- c. `fstream`: class untuk menulis dan membaca karakter dari/ke file-file.

Kelas-kelas tersebut berada di library `fstream`

Selanjutnya akan dibahas mengenai operasi-operasi pada file dan implementasinya dalam bahasa C.

11.1. Membuka file

1. Membuka file untuk di baca

Untuk membuka file untuk dibaca, pertama kali harus membuat objek dari kelas `ifstream`, dapat menggunakan sintaks:

```
ifstream file_objek;
```

Setelah objek_input diciptakan, maka file dibuka dengan cara

```
file_objek.open("nama_file");
```

Adapun dua pernyataan diatas dapat disederhanakan sebagai berikut:

```
ifstream file_objek("nama_file");
```

2. Membuka file untuk ditulisi

Untuk membuka file untuk ditulisi, pertama kali harus membuat objek dari kelas `ofstream`, dapat menggunakan sintaks:

```
ofstream file_objek;
```

Setelah objek_output diciptakan, maka file dibuka dengan cara

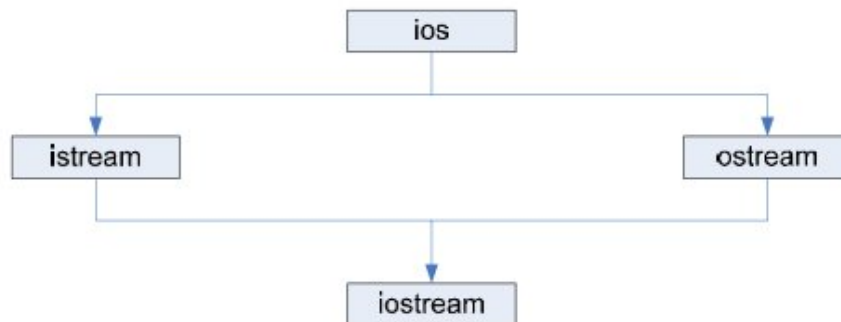
```
file_objek.open("nama_file");
```

Adapun dua pernyataan diatas dapat disederhanakan sebagai berikut:

```
ofstream file_objek("nama_file");
```

3. Membuka File dengan Modus Tertentu

Ada hirarki dari struktur IO / Operasi File ini



Penjelasan :

- ios adalah virtual base class untuk class istream dan ostream. Berisi fasilitas dasar untuk melakukan proses input/output. Dideklarasikan untuk pointer ke buffer untuk tempat penyimpanan data sementara.
- Istream (input stream) mendefinisikan fasilitas untuk melakukan input suatu informasi. Di dalamnya didefinisikan fungsi get (), getline (), extractor operator >>
- Ostream (output stream) mendefinisikan fasilitas untuk melakukan setting terhadap output.
- Iostream : berisi semua fasilitas dari ios, istream, ostream ditambah beberapa fungsi untuk menyempurnakan kerja dari fungsi yang dideklarasikan pada base class.

Pada suatu waktu mungkin anda memerlukan cara supaya file yang anda buka dapat dipergunakan untuk membaca dan menulis sekaligus, atau isi file yang sudah ada tidak dihapus jika anda ingin menambah isi file yang baru. Untuk keperluan itu, anda harus memformat modus pembukaan file. Adapun modus pembukaan file yang disediakan oleh C/C++ adalah sebagai berikut

Modus	Keterangan
ios::app	Membuka file dengan modus keluaran dan memungkinkan operasi penambahan data pada file yang telah ada. Jika file belum ada, maka membuat file baru.
ios::ate	Membuka file dengan modus masukan dan keluaran. Secara otomatis menempatkan pointer file ke posisi akhir file
ios::in	Membuka file dengan modus masukan. Penggunaannya sama dengan ifstream.
ios::out	Membuka file dengan modus keluaran. Penggunaannya sama dengan ofstream.
ios::nocreate	Membuka file yang sudah ada. Jika file yang akan dibuka belum ada, maka C++ tidak akan membuat file baru.
ios::noreplace	Membuka file baru. Jika file sudah ada maka operasi pembukaan menjadi gagal. Jika
	file belum ada, maka akan dibuat file baru. Hal ini bertentangan dengan ios::nocreate
ios::trunc	Menghapus file yang sudah ada dan menciptakan file baru (replace)
ios::binary	Membuka file dengan operasi baca-tulis secara binary.

Adapun contoh penggunaan dari modus – modus pembukaan file diatas adalah sebagai berikut :

fstream file_objek (“nama_file”,ios::in | ios::out);

Contoh Program :

```
#include <stdio.h>
#include <string.h>
#include <iostream>
using namespace std;
main()
{
    char namafileinput[] = "latihan.txt";
    char teks[80];
    FILE *fin;
    fin = fopen(namafileinput, "rt");
    printf("Isi File dari latihan.txt adalah : (Tekan Enter)\n");
    gets(namafileinput);
    while (1) {
        if (fgets(teks, 80, fin) == NULL) {
            break;
        }
        puts(teks);
    }
    fclose(fin);
}
```

SOAL LATIHAN

1. Buatlah sebuah file simpan.txt menggunakan fstream dalam bahasa C, serta dapat menulis dan membaca pada isi dari file simpan.txt
2. Buatlah sebuah program mini calculator, yang setiap percobaan/ujicoba perhitungan bisa disimpan dalam file calculator.txt/doc, serta tampilkan isi dari ujicoba mini calculator tadi.
3. Buatlah program yang mempunyai fungsi untuk:
 - a. Memasukkan data sebuah matrik ordo 3x3.
 - b. Menjumlahkan 2 buah matrik ordo 3x3.
 - c. Mengalikan 2 buah matrik ordo 3x3.
 - d. Mentranspose sebuah matrik ordo 3x3.

Simpan dan panggilah program 3a-3d tersebut kedalam txt / doc

PROSEDUR/ FUNGSI

Fungsi merupakan blok dari kode yang dirancang untuk melakukan tugas khusus. Adapun tujuan dari pembuatan suatu fungsi ini adalah :

1. Program menjadi terstruktur, program yang besar dapat dipecah menjadi beberapa program yang kecil, yang tiap satu program tersebut mempunyai tugas tertentu.
2. Mengurangi duplikasi kode.
3. Fungsi dapat dipanggil dari program atau fungsi yang lain.
4. Menghemat ukuran program , manfaat ini akan terasa kalau ada beberapa deretan intruksi yang sama digunakan pada beberapa tempat didalam program.
5. Memudahkan dalam mengembangkan program, hal ini merupakan kunci dalam pembuatan program terstruktur.

Pada umumnya fungsi memerlukan masukan yang dinamakan argumen atau parameter . Hasil akhir fungsi akan berupa sebuah nilai (nilai balik fungsi).

Contoh penggunaan fungsi:

```
#include<stdio.h>
void luas_segi3(int a, int t, int L);
void main()
{
    int alas, tinggi, luas;
    printf(" ..... \n\n ");
    printf("masukan alas : ");
    scanf("%i", &alas)
    printf("masukan tinggi : ");
    scanf("%i", &tinggi);
    luas_segi3(alas, tinggi, luas); //function call
    printf("maka luasnya : %i\n", luas);
}
void luas_segi3( int alas, int tinggi, int &luas)
{
    luas = (alas * tinggi)/2;  \\rumus
}
```

SOAL LATIHAN

1. Kerjakan soal-soal di modul 1 dengan menggunakan konsep fungsi yang:
 - a. Tanpa passing parameter.
 - b. Dengan passing parameter.
2. Kerjakan soal-soal di modul 2 dengan menggunakan konsep fungsi yang:
 - a. Tanpa passing parameter.
 - b. Dengan passing parameter.
3. Buatlah program untuk menghitung luas lingkaran dengan input panjang jari-jari lingkaran. Sedang output adalah luas lingkaran. Nilai input jari-jari lingkaran dilewatkan ke dalam sebuah fungsi untuk dihitung luasnya. Hasil perhitungan luas tersebut dikirimkan kembali ke main() untuk dicetak hasilnya.
4. Buat sebuah program untuk melakukan beberapa perhitungan dengan menggunakan ketiga rumus (menu) seperti yang tampak pada contoh tampilan dan output di bawah ini (dalam kotak). User (pengguna) pertama kali ditampilkan pilihan menu, kemudian input A, input N dan selanjutnya input pilihan rumus/ menu (1/2/3). Setelah proses dikerjakan, nilai output B ditampilkan. Bilangan A dan N bertipe real atau float.

Ketentuan : Gunakan fungsi tersendiri/ berbeda untuk setiap pilihan rumus (1/2/3) dan proses kalkulasinya (nilai B), sementara untuk nilai A dan N serta pencetakan/ output B ada dalam main() program, serta gunakan tempat penyimpanan data dengan menggunakan fungsi Operasi File / Fstream.