

Pertemuan 4

Diagram Alur / Flowchart

Flowchart






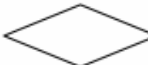


- ✓ Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu.
- ✓ Flowchart diawali dengan penerimaan input, pemrosesan input, dan diakhiri dengan penampilan output.
- ✓ bagan yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah.
- ✓ suatu diagram yang menggambarkan susunan logika suatu program

Simbol yang digunakan :

- ⊕ menunjukkan awal dan akhir dari program
- ⊕ memberikan nilai awal pada suatu variabel atau counter
- ⊕ menunjukkan pengolahan aritmatika dan pemindahan
- ⊕ menunjukkan proses input atau output
- ⊕ untuk mewakili operasi perbandingan logika
- ⊕ proses yang ditulis sebagai sub program, yaitu prosedur
- ⊕ penghubung pada halaman yang sama
- ⊕ penghubung pada halaman yang berbeda



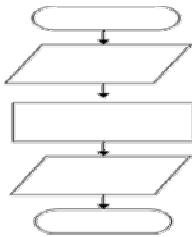
Simbol Flowchart dan fungsinya :

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	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

Flowchart terdiri dari 3 struktur :

1. Struktur Sequence /sederhana

Diagram yang alurnya mengalir secara berurutan dari atas ke bawah atau dengan kata lain tidak adanya percabangan atau pengulangan. Flowchart dengan struktur yang berurutan → alirannya dari atas ke bawah secara berurutan.

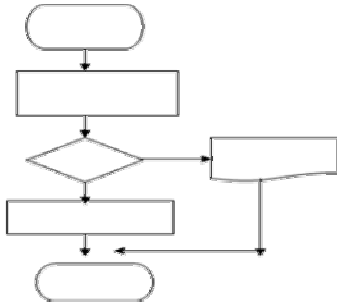


Contoh : flowchart dari algoritma mencari luas persegi panjang, Luas Lingkaran.

2. Struktur Branching

Diagram yg alurnya terjadi/terdapat alih kontrol berupa percabangan. Flowchart dengan struktur percabangan → digunakan untuk menyeleksi kondisi dan menentukan pilihan proses selanjutnya.

contoh : flowchart dari algoritma menentukan apakah bilangan yang dimasukkan ganjil atau genap.

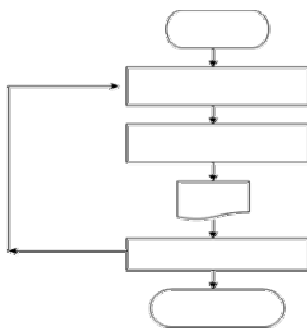


3. Struktur looping

Flowchart dengan Struktur perulangan → digunakan untuk mengulangi langkah-langkah sebelumnya sampai suatu kondisi terpenuhi.

Contoh:

flowchart dari algoritma untuk menampilkan bilangan ganjil dibawah nilai 10. → sehingga proses mencetak bilangan tersebut akan dilakukan sampai kondisi terpenuhi yaitu 10.



Catatan: Flowchart yang dibuat bisa juga merupakan gabungan dari ketiga struktur diatas.

VARIABEL

Variabel, sebagai tempat untuk menyimpan suatu nilai yang sejenis. Terdiri dari nama dari variable itu sendiri dan nilai yang disimpan.

variabel / Peubah → suatu nilai yg dapat berubah harganya.

Contoh pemberian nilai ke variabel :

$A = 5$ → variabel A diberi nilai 5.

$A = B$ → variabel A diberi nilai sama dengan nilai variabel B.
variabel B sudah memiliki nilai sebelumnya

$A = A + 1$ → variabel A dirubah isinya dengan variabel A yang dijumlahkan dengan 1.
(proses increment)

Jenis variabel terbagi atas :

1. Variabel numerik → berisi angka numerik /bilangan
2. Variabel String → berisi karakter.

STRUKTUR BRANCHING /Percabangan

1. Bersyarat

Diagram yg alurnya ada/banyak terjadi alih kontrol berupa percabangan & terjadi apabila kita dihadapkan pada suatu Kondisi dengan dua pilihan BENAR/ SALAH

Struktur :

- If then
- If then else
- If then elseif
- Case of.

2. Tidak Bersyarat

Struktur : GOTO

Studi kasus

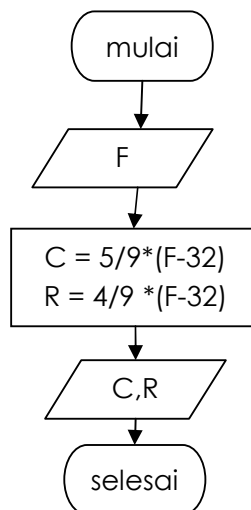
Buat diagram alur utk masalah menghitung temperatur dlm derajat Fahrenheit yg diubah kedlm derajat Celcius & Reamur.

Dengan rumus :

$$C = \frac{5}{9} (F-32) \qquad R = \frac{4}{9} (F-32)$$

Derajat Celsius (°C) adalah suatu satuan ukur suhu yang mendapatkan namanya dari ahli astronomi Anders Celsius (1701–1744), yang pertama kali mengusulkannya pada tahun 1742. Skala suhu celsius didesain supaya titik beku air berada pada 0 derajat dan titik didih pada 100 derajat di tekanan atmosferik standar.

Fahreheit adalah salah satu skala temperatur selain Celsius dan kelvin. Nama Fahrenheit diambil dari ilmuwan Jerman yang bernama Gabriel Fahrenheit (1686-1736). Dalam skala ini, titik beku air adalah 32 derajat Fahrenheit (ditulis 32°F) dan titik didih air adalah 212 derajat Fahrenheit. Negatif 40 derajat Fahreheit sama dengan negatif 40 derajat Celsius.



Soal Latihan :

1. Algoritma konversi jam ke menit. Dengan masukannya jam dan menit.

2. Algoritma untuk menghitung jumlah yang harus dibayar oleh pembeli dari sejumlah barang yang dibeli, setelah mendapatkan diskon 10% dengan syarat jumlah total pembelian > Rp.1.500.000,-

Jawaban Tugas 1 dan 2 (Pertemuan 1)

1. Algoritma untuk menampilkan bilangan Ganjil dari 1 sampai dengan 10

model 1:

1. Mulai
2. Tetapkan nilai Bilangan = 1 dan Batas_Bilangan = 10
3. Jika sisa pembagian (Bilangan/2) tidak sama dengan 0 (bilangan **mod** 2 \neq 0) maka Cetak "Bilangan", dan kelangkah 5.
4. Jika (Bilangan = Batas_Bilangan) maka ke-langkah 6
5. Nilai Bilangan ditambah 1 (Bilangan=Bilangan+1) dan kembali kelangkah 3
6. Selesai.

2. Menghitung jumlah deret dari $1+2+3+ \dots + N$.

1. Mulai
2. Masukan Nilai N
3. Tetapkan Bilangan = 1, Deret = 0
4. Hitung Deret = Deret + Bilangan
5. Jika Bilangan = N maka cetak Deret dan stop
6. Jika tidak, Bilangan ditambah 1 (Bilangan = Bilangan + 1) dan kembali kelangkah 4.

A = 10	A= 20
B = 20	B= 10
C = 100	C= 100
Perintah yang dijalankan :	Perintah yang dijalankan :

Tersedia potongan Program berikut ini :

```
If (A>B) then
begin
  Perintah 1
  If ((A< B) or (C>B)) then Perintah 2
  Else perintah 3
end
Else
  If C > A then
  begin
  perintah 4
  if (A<B and A<C) then perintah 5
  else perintah 6
  end;
perintah 8
```

Buatlah bentuk Flowchart dari potongan Program diatas.

Pertemuan 5

LOOPING (Perulangan)

Suatu algoritma memiliki struktur

- ⊕ sequence/berurutan
- ⊕ branching/percabangan
- ⊕ looping/berurutan.

Struktur looping digunakan untuk mengulangi langkah-langkah sebelumnya yang telah dikerjakan, kondisi perulangan dilakukan sampai suatu kondisi berhenti terpenuhi.

proses Perulangan digunakan contohnya untuk membuat algoritma :

- ☑ "menampilkan bilangan berurutan dari 1 sampai dengan 10"
- ☑ "menampilkan deret : 3, 7, 11, 15 N, sampai sejumlah N"
- ☑ "menampilkan bilangan dari 10 sampai dengan 1 "

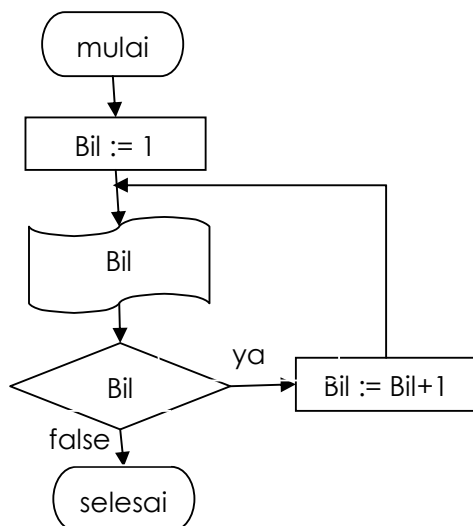
Dari algoritma diatas dipilih algoritma pertama.

Proses perulangan akan dilakukan yaitu :

- ⊕ mencetak bilangan dari 1 -10
- ⊕ melakukan proses increament (penambahan bilangan dengan 1)

proses perulangan ini akan dilakukan sampai kondisi terakhir yaitu mencetak bilangan 10.

Buat Flowchart algoritma diatas ?



Tips:

Dalam membuat algoritma (contoh: menggunakan flowchart. Sebelum membuat flowchart terlebih dahulu kita identifikasi kira-kira ada berapa variabel/peubah yang digunakan dalam proses pembuatan algoritma.

Bila sebuah rumus : luas = panjang x lebar

Maka bila dibuat algoritmanya maka

Bentuk umum proses perulangan:

1. while [kondisi] do [.....] end_while
2. repeat until [kondisi]
3. for [.....] end_for
 - for [nilai awal] to [nilai akhir] do
 - for [nilai akhir] to [nilai awal] do
 - Nested for (perulangan for bersarang)

Perulangan	Statement dan contoh program
while do: Evaluasi kondisi dilakukan di bagian awal	<pre>while (kondisi) do (statement) Contoh: i := 1; while (i<10) do begin i := i + 2; writeln(i); end; hasil : 3 5 7 9 12</pre>
Repeat loop: Evaluasi kondisi dilakukan di bagian akhir.	<pre>repeat (statement); ... (statement); until (kondisi); Contoh: i := 1; repeat i := i + 2; write(i); until i >=10; Hasil = 3 5 7 9 12</pre>
for..loop: Perulangan dengan increment nilai	<pre>for counter := lower to upper do (statement) Atau for counter := upper downto lower do (statement) Contoh: for i := 1 to 10 do write(i); hasil : 1 2 3 4 5 6 7 8 9 10</pre>
Nested Loop for	<pre>for i := 1 to 10 do begin write(i); for j := 10 downto i do write(j);</pre>

		<pre>writeln; end;</pre>
--	--	--------------------------

Dari soal berikut Buatlah Flowchart nya :

1. Menentukan suatu bil. bulat positif merupakan bil.genap/ganjil.
2. Mencetak 10 suku pertama dr barisan geometri dgn suku pertama 3 dan rasio = 6
3. Mencetak suku barisan aritmatika dgn suku pertama 3 dan beda 4, s/d suku yg harganya tdk melebihi nilai 100.
4. Mencetak suku deret aritmatika dgn hasilnya adalah 3, 7, 11 sampai 12 suku
5. Mencetak deret bergoyang 1, -2, 4, -8 sampai dengan 10 suku.

Barisan dan Deret Geometri

Barisan geometri adalah suatu barisan yang suku selanjutnya diperoleh dengan mengalikan suatu bilangan tertentu kepada suku sebelumnya.

Bilangan tetap tersebut dinamakan pembanding/rasio.

Bentuk umum barisan geometri : a, ar, ar^2, \dots, ar^n

a = suku awal, r = rasio,

Deret geometri adalah jumlah semua suku-suku barisan geometri. (disebut juga deret ukur).

Bentuk umum : $a + ar + ar^2 + \dots + ar^n$

Pertemuan 6

Struktur Rekursif

Rekursif adalah suatu proses yang bisa memanggil dirinya sendiri .

Perulangan Rekursif dan Perulangan Iteratif.

Perulangan rekursif merupakan salah satu metode didalam pemrograman yang mana dalam sebuah fungsi terdapat intruksi yang memanggil fungsi itu sendiri, atau lebih sering disebut memanggil dirinya sendiri.

Perulangan iteratif merupakan perulangan yang melakukan proses perulangan terhadap sekelompok intruksi. Perulangan dilakukan dalam batasan syarat tertentu. Ketika syarat tersebut tidak terpenuhi lagi maka perulangan aka terhenti.

Persamaan :

1. Iteratif dan rekursif merupakan metode atau teknik didalam perulangan(looping)
2. Sama-sama memiliki bagian yang berfungsi sebagai batas dalam sebuah perulangan

Perbedaan :

- ☑ Iteratif dalam melakukan perulangan membutuhkan suatu instruksi program seperti **for**, **repeat until** dan **while do**, sedangkan rekursi tidak memakai instruksi program seperti itu. Cukup dengan fungsi tersebut.

```

Procedure Rekursif; ← nama fungsi: rekursif.
Begin
Write('AMIK BSI ');
    Rekursif; ← fungsi bernama rekursif ini dikatakan
                Sebagai fungsi rekursi karena dia memanggil
                Dirinya sendiri
End;
Begin
    Rekursif;
End.

```

Pada contoh diatas: sebuah fungsi/prosedur adalah sebuah proses. Contoh fungsi **rekursif** diatas adalah sebuah proses, dan didalam fungsi **rekursif** tersebut terdapat perintah/proses yang mengerjakan/memanggil proses **rekursif** (memanggil dirinya sendiri), sehingga fungsi/prosedur **rekursif** ini dinamakan fungsi rekursi.

Pada contoh diatas proses rekursi ini tidak memiliki batas berhenti. Untuk mengetahui contoh fungsi rekursif ini silahkan melihat pada slide perkuliahan.

Catatan: Fungsi/prosedur dalam sebuah bahasa pemrograman disebut juga subrutin/sub program.
Subprogram ini berisi perintah –perintah khusus yang sering digunakan

untuk proses pemrograman. Sehingga untuk menghindari penulisan kode program yang sering digunakan maka dibuatlah fungsi (subprogram). Cara untuk menggunakan subprogram yaitu dengan memanggil nama_fungsi tersebut melalui blok program utama. Contoh pemanggilan terdapat pada program faktorial.

FUNGSI FAKTORIAL

$0! = 1$

$N! = N \times (N-1)!$ Untuk $N > 0$

Nama fungsi:

```
function faktorial(N:integer):integer;
var i:integer;
begin
  if N = 0 then
  begin
    faktorial :=1;
    exit;
  end
  else
    faktorial := n * faktorial(n-1);
  end;

begin
  write('Masukan Bilangan = '); readln(Bil);
  write('jumlah faktorial = ',faktorial(Bil));
  readln;
end.
```

Fungsi FAKTORIAL
yang memanggil

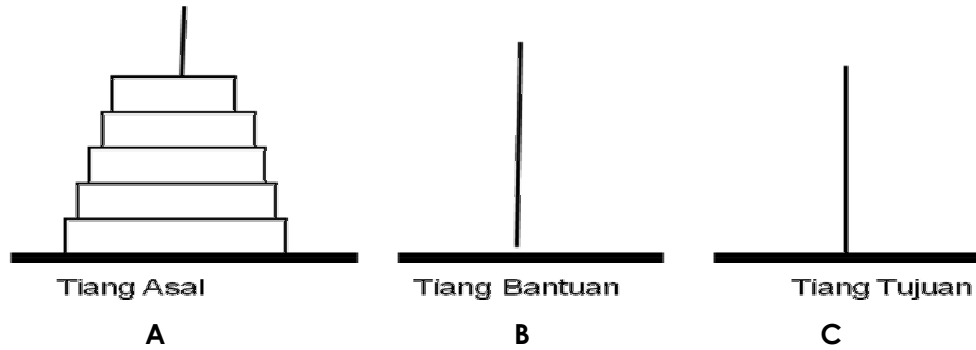
1. Cara
menggunakan fungsi
yaitu dengan
memanggil nama
fungsi tersebut dari
program blok utama

2. Karena fungsi factorial ini
merupakan fungsi rekursif
maka ketika dijalankan
akan terjadi proses
perulangan

HASIL:
Masukan Bilangan = 5
jumlah faktorial = 120

MENARA HANOI

Tujuan permainan ini adalah memindahkan n buah piringan dari tonggak asal A melalui tonggak bantu B menuju tonggak tujuan C. dengan aturan piring yang lebih kecil tidak boleh berada dipiringan yang lebih besar.



Bayangkan keadaan berikut:

Ada 3 tiang (a, b, c) tempat piringan dengan ukuran yang bervariasi dapat ditumpuk. Pada mulanya semua piringan ada di "a". Tugasnya adalah Memindah semua piringan ke "c" dengan aturan sbb:

- pada satu saat hanya boleh memindah 1 piringan
- setiap perpindahan berupa pengambilan piringan teratas dari satu tiang dan memasukkannya ketiang lain, diatas piringan lain yang mungkin sudah ada pada tiang tersebut.
- Tidak boleh meletakkan piringan diatas piringan lain yang lebih kecil. (piringan yang lebih besar tidak boleh berada di atas piringan yang lebih kecil).
- pada setiap akhir pemindahan semua piringan harus berada di tiang

Untuk Penyelesaian masalah kita daftarkan terlebih dahulu langkah-langkah penyelesaiannya:

- ketika kondisi piringan $\rightarrow N=1$
Piringan 1 dipindahkan dari A ke C.
- $N=2$
 - \rightarrow Pindahkan piringan 1 dari A ke B
 - \rightarrow Pindahkan piringan 2 dari A ke C
 - \rightarrow Pindahkan piringan 1 dari B ke C

Dari langkah – langkah penyelesaian diatas, maka dapat disimpulkan :

- Untuk memindahkan piringan dari tonggak asal (A) ke tonggak tujuan (C) maka piringan ke N harus berada di tonggak tujuan (C).
- Sedangkan piringan ke 1 sampai dengan (N-1) harus berada ditonggak bantu(B).
- Setelah piringan ke 1 s/d N-1 berada di B, Kemudian pindahkan piringan ke 1 sampai dengan N-1 dari tonggak bantu (B) ke tonggak tujuan (C).

Untuk menyelesaikan masalah tersebut dapat digunakan algoritma Rekursif :

```
uses crt;
procedure Hanoi(n:integer;asal,bantu,tujuan:char);
begin
  if n=0 then exit;
  hanoi(n-1,asal,tujuan,bantu);
  writeln('. Pindahkan piringan ke ',n,' dari ',asal,' ke ',tujuan);
```

```
    hanoi(n-1,bantu,asal,tujuan);
end;

var N : integer;

begin
    clrscr;
    write('Jumlah Piringan = ');
    readln(N);
    hanoi(N,'A','B','C');
    readln;
end.
```

Hasil :

```
Jumlah Piringan = 4
. Pindahkan piringan ke 1 dari A ke B
. Pindahkan piringan ke 2 dari A ke C
. Pindahkan piringan ke 1 dari B ke C
. Pindahkan piringan ke 3 dari A ke B
. Pindahkan piringan ke 1 dari C ke A
. Pindahkan piringan ke 2 dari C ke B
. Pindahkan piringan ke 1 dari A ke B
. Pindahkan piringan ke 4 dari A ke C
. Pindahkan piringan ke 1 dari B ke C
. Pindahkan piringan ke 2 dari B ke A
. Pindahkan piringan ke 1 dari C ke A
. Pindahkan piringan ke 3 dari B ke C
. Pindahkan piringan ke 1 dari A ke B
. Pindahkan piringan ke 2 dari A ke C
. Pindahkan piringan ke 1 dari B ke C
```

Hasil 2 :

```
Jumlah Piringan = 3
. Pindahkan piringan ke 1 dari A ke C
. Pindahkan piringan ke 2 dari A ke B
. Pindahkan piringan ke 1 dari C ke B
. Pindahkan piringan ke 3 dari A ke C
. Pindahkan piringan ke 1 dari B ke A
. Pindahkan piringan ke 2 dari B ke C
. Pindahkan piringan ke 1 dari A ke C
```