

Representasi Algoritma



I Gusti Agung Gede Arya Kadyanan, S.Kom.,M.Kom

Syarat algoritma

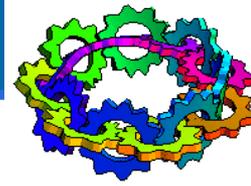
Horowitz, Eliis and Sahni, Sartaj; **FUNDAMENTAL OF DATA STRUCTURES**; Computer Science Press, Inc.; Rocville, Maryland; 1983

Secara bebas definisi diatas dapat diterjemahkan sebagai berikut :

Algoritma adalah sekumpulan instruksi, yang apabila dijalankan, akan menyelesaikan suatu tugas tertentu. Sebagai tambahan, setiap algoritma harus memenuhi kriteria sebagai berikut

- 1). Tidak harus ada data masukan yang dimasukkan dari luar.
- 2). Paling tidak ada satu buah keluaran
- 3) Setiap instruksi jelas maksudnya dan tidak meragukan
- 4). Algoritma baik secara keseluruhan maupun sub algoritma bila ditelusuri harus ada titik berhentinya.
- 5). Setiap instruksi selain jelas juga harus dapat dilaksanakan, dan juga efektif dalam arti harus menghasilkan sesuatu. Sebagai contoh $A = A + 0$ atau $A = A * 1$, adalah termasuk instruksi yang tidak efektif.

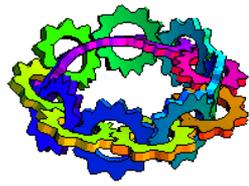
ALGORITMA



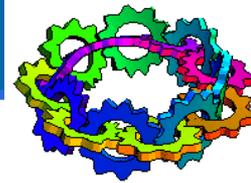
1. Pseudocode

2. Flowchart

Pseudocode



1. Pseudocode digunakan untuk menuliskan algoritma menggunakan serangkaian pernyataan yang ditulis dalam bahasa sehari-hari.



Pseudocode

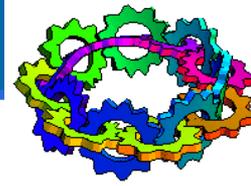
1. Pseudocode digunakan untuk menuliskan algoritma menggunakan serangkaian pernyataan yang ditulis dalam bahasa sehari-hari.
2. Setiap pernyataan dalam pseudocode ditulis dalam sebuah nomor baris yang menyatakan nomor urut pernyataan. Jika sebuah nomor memiliki sub-kegiatan, maka digunakan aturan penomoran secara *indent*.

1. Nomor utama

1.1 Sub-nomor

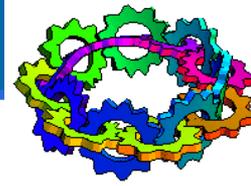
- a. Rincian pertama
- b. Rincian kedua
- c. ...

Pseudocode



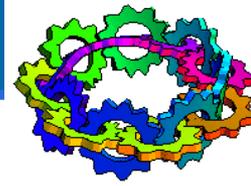
1. Pseudocode digunakan untuk menuliskan algoritma menggunakan serangkaian pernyataan yang ditulis dalam bahasa sehari-hari.
2. Setiap pernyataan dalam pseudocode ditulis dalam sebuah nomor baris yang menyatakan nomor urut pernyataan. Jika sebuah nomor memiliki sub-kegiatan, maka digunakan aturan penomoran secara *indent*.
3. Pembacaan pseudocode dilakukan secara urut baris-demi-baris.

Pseudocode



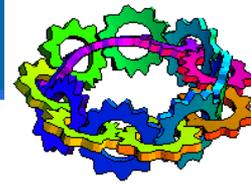
1. Pseudocode digunakan untuk menuliskan algoritma menggunakan serangkaian pernyataan yang ditulis dalam bahasa sehari-hari.
2. Setiap pernyataan dalam pseudocode ditulis dalam sebuah nomor baris yang menyatakan nomor urut pernyataan. Jika sebuah nomor memiliki sub-kegiatan, maka digunakan aturan penomoran secara *indent*.
3. Pembacaan pseudocode dilakukan secara urut baris-demi-baris.
4. Pada kondisi tertentu, ada kalanya pembacaan pseudocode tidak dilakukan urut dari baris satu ke baris selanjutnya, akan tetapi melompat ke baris yang lain.

Pseudocode



1. Pseudocode digunakan untuk menuliskan algoritma menggunakan serangkaian pernyataan yang ditulis dalam bahasa sehari-hari.
2. Setiap pernyataan dalam pseudocode ditulis dalam sebuah nomor baris yang menyatakan nomor urut pernyataan. Jika sebuah nomor memiliki sub-kegiatan, maka digunakan aturan penomoran secara *indent*.
3. Pembacaan pseudocode dilakukan secara urut baris-demi-baris.
4. Pada kondisi tertentu, ada kalanya pembacaan pseudocode tidak dilakukan urut dari baris satu ke baris selanjutnya, akan tetapi melompat ke baris yang lain.
5. Terdapat beberapa *keyword* atau “kata kunci” yang digunakan dalam menulis pseudocode.

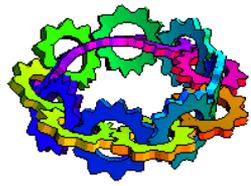
Keyword pada Pseudocode



1. BEGIN

Menyatakan awal pseudocode.

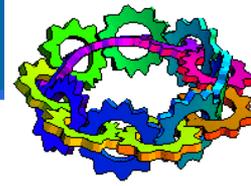
Keyword ini harus ada dan diletakkan di nomor 1 (satu). Dari sini seluruh instruksi bermula.



2. END

Menyatakan akhir pseudocode.

Keyword ini harus ada dan diletakkan di nomor terakhir. Di sini seluruh instruksi berakhir.

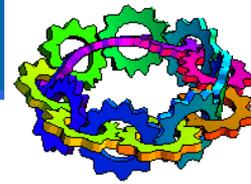


3. IF - THEN

Menyatakan alternatif pilihan dengan sebuah kemungkinan proses terusan.

```
IF <kondisi> THEN <pernyataan>
```

Jika <kondisi> bernilai BENAR maka <pernyataan> akan dikerjakan. Sebaliknya, baris ini akan dilompati.

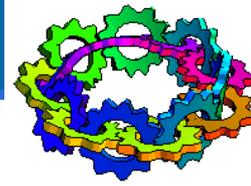


4. IF - THEN - ELSE

Menyatakan alternatif pilihan dengan 2 (dua) kemungkinan alur terusan.

```
IF <kondisi> THEN <pernyataan_1>  
    ELSE <pernyataan_2>
```

Jika <kondisi> bernilai BENAR maka <pernyataan_1> akan dikerjakan. Sebaliknya, <pernyataan_2> yang akan dikerjakan.

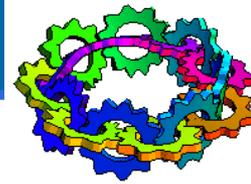


5. WHILE

Menyatakan adanya proses berulang bersyarat.

```
WHILE <kondisi>  
    <pernyataan_1>  
    ...  
    <pernyataan_n>
```

Selama <kondisi> bernilai BENAR maka <pernyataan_1> sampai dengan <pernyataan_n> yang akan dikerjakan.



6. FOR

Menyatakan adanya proses berulang tanpa syarat.

```
FOR var = awal TO akhir
```

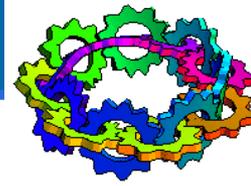
```
    <pernyataan_1>
```

```
    ...
```

```
    <pernyataan_n>
```

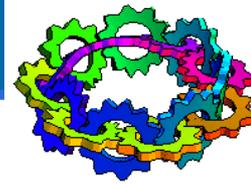
<pernyataan_1> sampai dengan <pernyataan_n> akan dikerjakan seiring dengan pergerakan nilai variabel “var” dari nilai “awal” sampai nilai “akhir”.

Flowchart

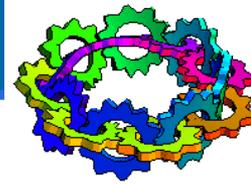


1. Flowchart digunakan untuk menuliskan algoritma menggunakan serangkaian simbol khusus.

Flowchart



1. Flowchart digunakan untuk menuliskan algoritma menggunakan serangkaian simbol khusus.
2. Setiap pernyataan dalam flowchart ditulis dalam sebuah simbol. Beberapa *keyword* juga ditulis di dalam simbol, mendahului pernyataan dimaksud.



1. Flowchart digunakan untuk menuliskan algoritma menggunakan serangkaian simbol khusus.
2. Setiap pernyataan dalam flowchart ditulis dalam sebuah simbol. Beberapa *keyword* juga ditulis di dalam simbol, mendahului pernyataan dimaksud.
3. Terdapat simbol GARIS-BERARAH yang menyatakan arah proses berikutnya dari suatu simbol. Pembacaan flowchart dilakukan dari satu simbol ke simbol lain mengikuti arah yang ditunjuk oleh GARIS-BERARAH.

Notasi Flowchart

Simbol dalam Flowchart ini menggunakan standar ANSI (American National Standard Institute)



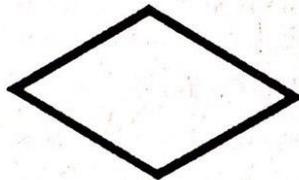
Terminal untuk menyatakan START dan END



PROSES, untuk menyatakan Assignment Statement



I/O , Input/Output operation, untuk menyatakan proses Baca (READ) dan proses tulis (WRITE)



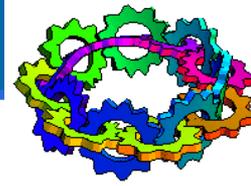
Decision, untuk menyatakan pengambilan keputusan sesuai dengan suatu kondisi



Garis untuk menyatakan urutan pelaksanaan, atau alur proses



Scanned with CamScanner



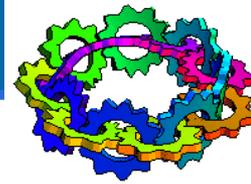
1. TERMINATOR

Menyatakan AWAL atau AKHIR flowchart.

- Keyword BEGIN untuk menyatakan awal flowchart.

BEGIN

Simbol pada Flowchart



1. ARROW

Menyatakan arah proses yang bergerak dari satu simbol menuju simbol berikutnya. Pembacaan flowchart harus mengikuti arah panah yang keluar dari sebuah simbol.

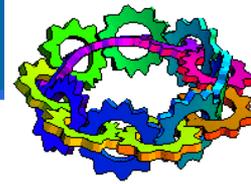
Penggambaran garis tanpa disertai arah panah merupakan kesalahan.



Arah proses horisontal



Arah proses vertikal



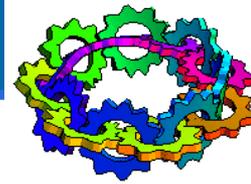
2. TERMINATOR

Menyatakan AWAL atau AKHIR flowchart.

- Keyword BEGIN untuk menyatakan awal flowchart.
- Keyword END untuk menyatakan akhir flowchart.



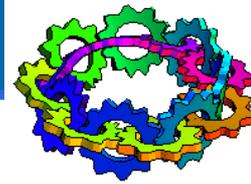
END



3. PROCESS

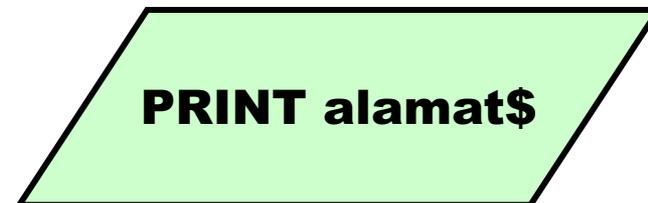
- Menyatakan proses yang terjadi secara internal di dalam komputer. Proses ini dilakukan TANPA campur tangan manusia.
- Seringkali berupa ekspresi matematika.

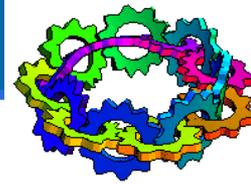
X ← 10
Y ← A³
Z ← P / Q



4. INPUT/OUTPUT

- Digunakan untuk mencetak informasi ke layar ATAU meminta data dari user melalui keyboard.
- Keyword PRINT digunakan untuk mencetak informasi ke layar, diikuti dengan data yang akan ditampilkan.

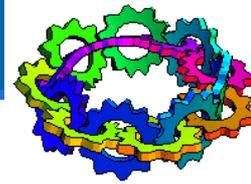




4. INPUT/OUTPUT

- Digunakan untuk mencetak informasi ke layar ATAU meminta data dari user melalui keyboard.
- Keyword PRINT digunakan untuk mencetak informasi ke layar, diikuti dengan data yang akan ditampilkan.
- Jika data yang akan ditampilkan lebih dari satu, pisahkan data-data tersebut dengan koma.

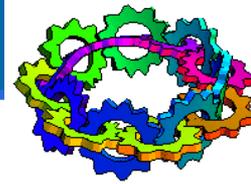
**PRINT alamat\$, kodePos,
gajiPokok, nilai**



4. INPUT/OUTPUT

- Digunakan untuk mencetak informasi ke layar ATAU meminta data dari user melalui keyboard.
- Keyword INPUT digunakan untuk meminta data dari user melalui keyboard, diikuti dengan nama variabel yang akan menampung data tersebut.

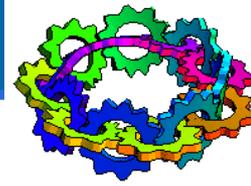




4. INPUT/OUTPUT

- Digunakan untuk mencetak informasi ke layar ATAU meminta data dari user melalui keyboard.
- Keyword INPUT digunakan untuk meminta data dari user melalui keyboard, diikuti dengan nama variabel yang akan menampung data tersebut.
- Jika data yang diminta lebih dari satu, pisahkan variabel-variabel penampungnya dengan koma.

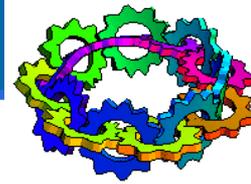
**INPUT nama\$, nilaiUjian,
indeksKerja**



5. DECISION

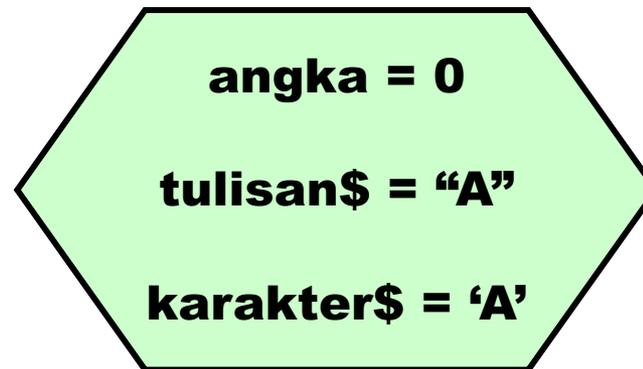
- Digunakan untuk menyatakan pemilihan alternatif.
- Hasil dari keputusan adalah BENAR atau SALAH; tidak bisa keduanya terjadi secara bersamaan.
- Terdapat 2 (dua) panah yang keluar dari simbol ini.

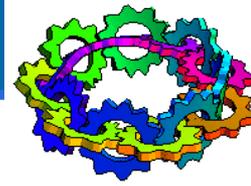




6. PREPARATION

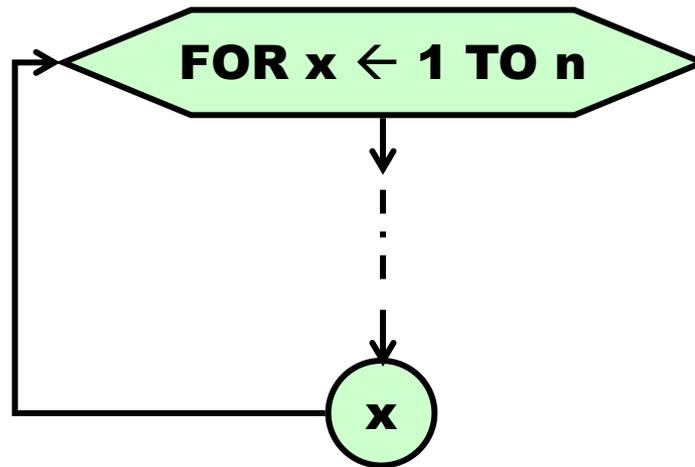
- Digunakan untuk menginisialisasi variabel.
- Digunakan juga sebagai sarana untuk menentukan tipe variabel.

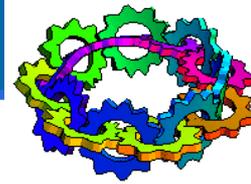




7. LOOPING

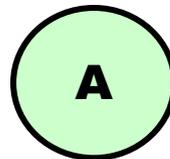
- Digunakan untuk mem-*block* instruksi berulang.
- Proses *update* counter dilakukan secara otomatis satu-demi-satu.

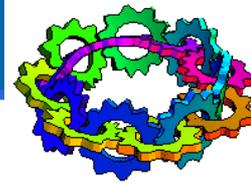




8. CONNECTOR

- Digunakan sebagai titik sambungan jika penggambaran flowchart terputus. Huruf yang tertera di dalam simbol ini menyatakan ID sambungan.
- ON-PAGE connector digunakan jika flowchart sambungan ditulis pada halaman yang sama dengan flowchart sebelumnya.

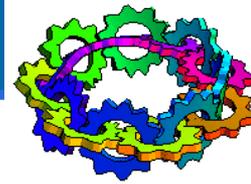




8. CONNECTOR

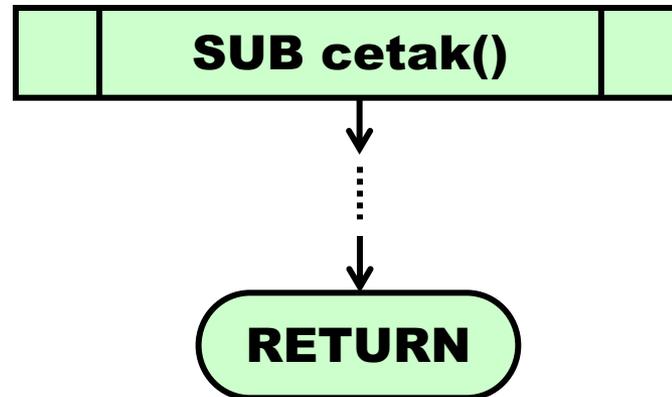
- Digunakan sebagai titik sambungan jika penggambaran flowchart terputus. Huruf yang tertera di dalam simbol ini menyatakan ID sambungan.
- ON-PAGE connector digunakan jika flowchart sambungan ditulis pada halaman yang sama dengan flowchart sebelumnya.
- OFF-PAGE connector digunakan jika flowchart sambungan ditulis pada halaman yang berbeda dengan flowchart sebelumnya.

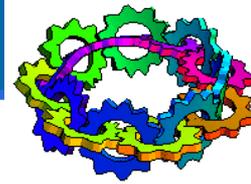




9. SUB-PROCESS → procedure

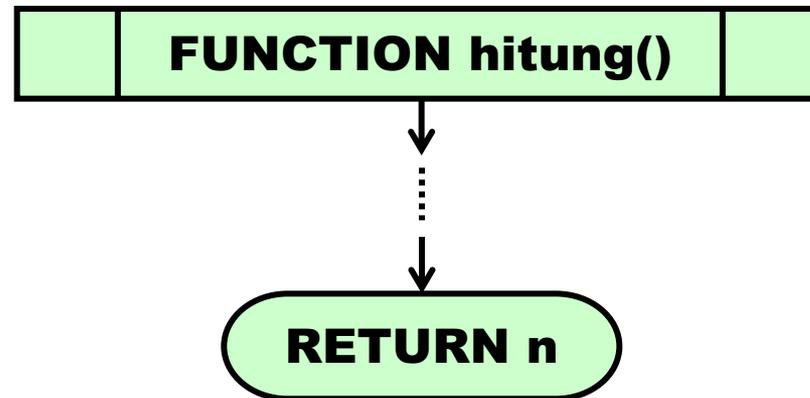
- Sub-proses berjenis procedure diawali dengan keyword SUB.
- Akhir sub-proses diakhiri dengan keyword RETURN.

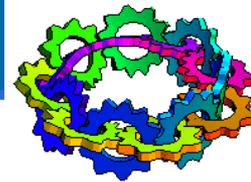




10. SUB-PROCESS → function

- Sub-proses berjenis function diawali dengan keyword **FUNCTION**.
- Akhir sub-proses diakhiri dengan keyword **RETURN** diikuti dengan sebuah nilai balik.

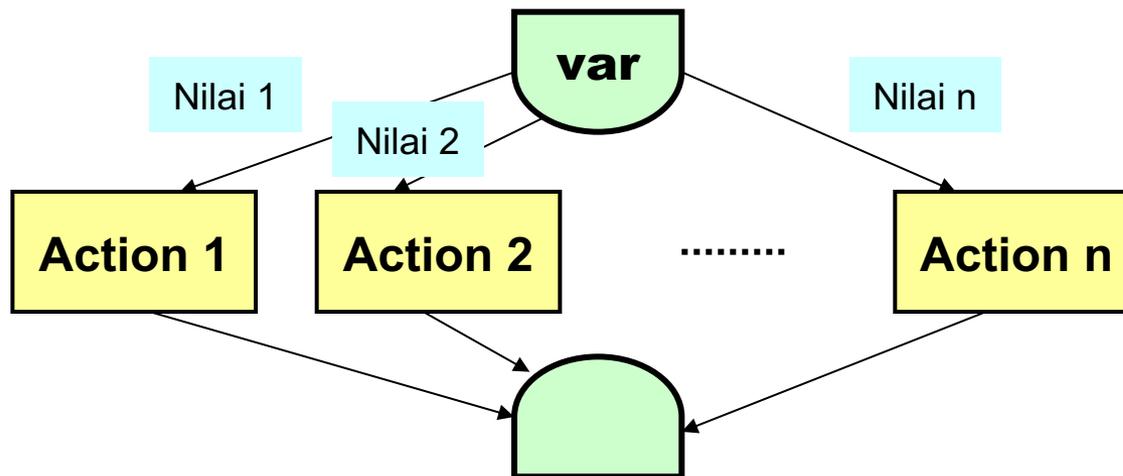


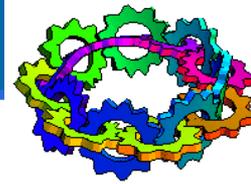


Simbol pada Flowchart

11. SELECTION

- Digunakan untuk mempermudah pembacaan alur percabangan yang terlalu banyak.
- Hanya untuk memeriksa variabel dengan nilai tertentu, bukan pada suatu sub-range.





12. LAIN-LAIN

Untuk simbol yang lain silahkan dibaca sendiri.

MK ini hanya menampilkan beberapa simbol yang sering dipakai.

Kaitan Pseudocode dengan Flowchart

Untuk menunjukkan kaitan antara Pseudo Code dan Algoritma, berikut ini diperlihatkan beberapa contoh penulisan Algoritma untuk Pseudo Code yang diberikan

Pseudo Code

- Nilai A ditambah dengan 5
- Cetak nilai A, bila nilai tersebut lebih besar dari 5
- Dari dua buah nilai A dan B cetak salah satu yang terbesar
- Kurangi dengan 2 nilai A terus menerus sampai nilainya lebih kecil atau sama dengan nol

Algoritma

- $A = A + 5$
- IF (A > 5) THEN WRITE (A)
- IF (A > B) THEN WRITE (A)
ELSE WRITE B)
- While (A > 0) do
A = A - 2
End Do



Scanned with CamScanner

Contoh program mengurutkan isi array

a. Bahasa BASIC

```
FOR I=1 TO N-1
  J = 1
  FOR K=J+1 TO N
    IF A(K)< A(J) THEN
      J = K
    END IF
  NEXT K
  T=A(I) : A(I)=A(J) : A(J)=T
NEXT I
```

b. Bahasa PASCAL

```
FOR I := 1 TO N-1
  Begin
    J := 1;
    FOR K := J+1 TO N
      Begin
        IF A[ K ] < A[ J ]
          J := K;
        End;
      T:=A[ I ] ; A[ I ] :=A[ J ] ; A[ J ] :=T;
    End;
```

Bahasa C atau C++

```
for(I=1; I <= N-1; I++)
{
  J = 1;
  for(K = J+1; K <= N; K++)
  {
    if(A[ K ] < A[ J ] )
      J = K;
  }
  T=A[ I ] ; A[ I ] =A[ J ] ; A[ J ] =T;
}
```

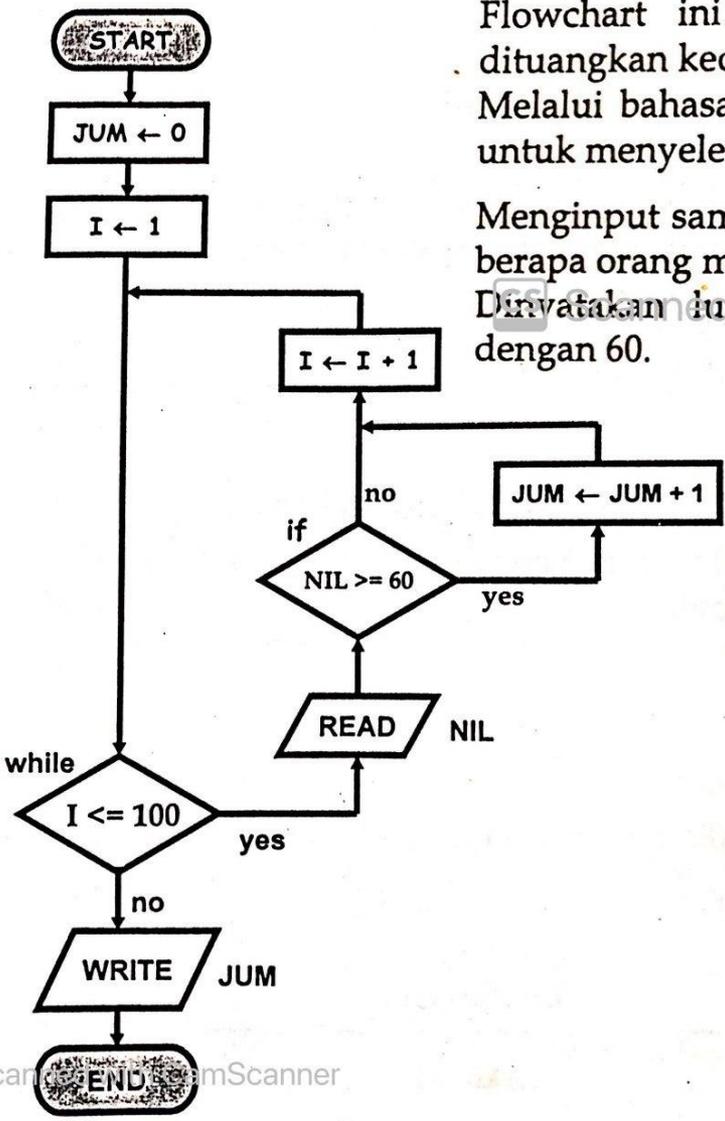
d. Bahasa Java

```
for(I=1; I <= N-1; I++)
{
  J = 1;
  for(K = J+1; K <= N; K++)
  {
    if(A[ K ] < A[ J ] )
      J = K;
  }
  T=A[ I ] ; A[ I ] =A[ J ] ; A[ J ] =T;
}
```



Scanned with CamScanner

Notasi Flowchart cetak nilai mhs



Flowchart ini menggambarkan alur pikiran yang akan dituangkan kedalam suatu bahasa pemrograman. Melalui bahasa pemrograman tersebut komputer diperintahkan untuk menyelesaikan suatu pekerjaan yaitu :
Menginput sampel 100 nilai ujian mahasiswa dan mencetak ada berapa orang mahasiswa yang lulus dari 100 sampel tersebut. Dinyatakan lulus apabila nilai ujian lebih besar atau sama dengan 60.

Notasi Flowchart cetak nilai mhs

Flowchart diatas, bila dituangkan ke dalam bentuk algoritma, dapat dibuat sebagai berikut :

```
JUM ← 0
I ← 1
WHILE I ≤ 100
  DO
    READ(NIL)
    IF NIL ≥ 60
      THEN JUM ← JUM + 1
    ENDIF
    I ← I + 1
  ENDDO
WRITE(JUM)
```

Bila dituangkan kedalam Bahasa C menjadi :

```
#include<stdio.h>
void main()
{ int JUM, I, NIL;
  JUM = 0;
  I = 1;
  while(I ≤ 100)
  { scanf("%i", &NIL);
    if (NIL ≥ 60)
      JUM = JUM + 1;
    I = I + 1;
  }
  printf("\n %i", JUM);
}
```

2). Mencari suatu nilai dalam array dengan cara Binary Search.

Bila ditulis dalam bentuk Pseudo Code :

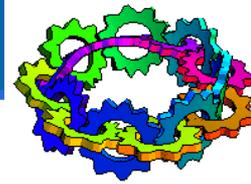
PSEUDO-CODE

```
Procedure BINSRCH(A,n,x,j)  
  initialize lower and upper  
  while there are more elements to check do  
    let A(mid) be the middle element  
    case  
      :  $x > A(\text{mid})$  : set lower to  $\text{mid} + 1$   
      :  $x < A(\text{mid})$  : set upper to  $\text{mid} - 1$   
      : else : found  
    end  
  end  
  not found  
end BINSRCH
```

Bila ditulis dalam bentuk algoritma:

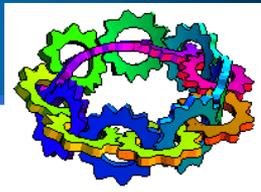
ALGORITMA

```
Procedure BINSCRH(A,n,x,j)  
  lower ← 1; upper ← n  
  while lower ≤ upper do  
    mid ← [ (lower + upper) / 2 ]  
    case  
      :  $x > A(\text{mid})$  : lower ← mid + 1  
      :  $x < A(\text{mid})$  : upper ← 1  
      : else : j ← mid; return  
    end  
  end  
  j ← 0  
end
```



1. **Perhitungan bangun geometri: LUAS dan VOLUME.**
2. **Menukar isi dua variabel.**
3. **Menghitung umur seseorang ATAU tahun kelahiran seseorang.**
4. **Menghitung nilai suhu dalam berbagai satuan: celcius, reamur, fahrenheit, kelvin.**
5. **Menghitung nilai akhir mahasiswa jika diketahui nilai tugas, quis, uts dan uas serta bobot masing-masing nilai tersebut.**
6. **Menghitung discount belanja pada sebuah supermarket.**

End of slide



Terima kasih