

Notasi dan Struktur Dasar Algoritma

Notasi Algoritma

1. Notasi Menyatakan dalam Langkah-langkah
2. Notasi Bagan Alir (flowchart)
3. Notasi Pseudocode




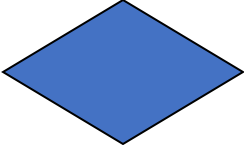
1. Menyatakan dalam Langkah-langkah

- Menggunakan Bahasa tingkat tinggi
- Berisi Langkah-Langkah penyelesaian persoalan
- Contoh : Persoalan Luas segitiga
- Algoritma
 1. Nilai alas dan tinggi diketahui
 2. Kalikan nilai alas dan tinggi
 3. Bagi 2 hasil perkalian Langkah ke 2
 4. Tampilkan hasil Luas segitiga






2. Bagan Alir (Flowchart)

- bagan yang menggambarkan urutan instruksi untuk proses dengan komputer dan hubungan antara suatu proses dengan proses lainnya dengan menggunakan simbol
- tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk

Flowchart

	Starter dan Terminator
	Input/Output
	Proses
	Condisional / Decision

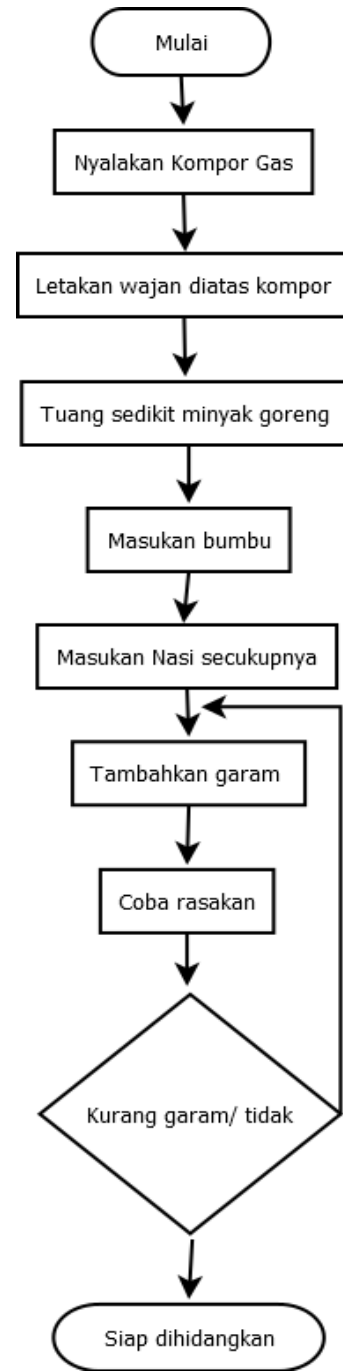
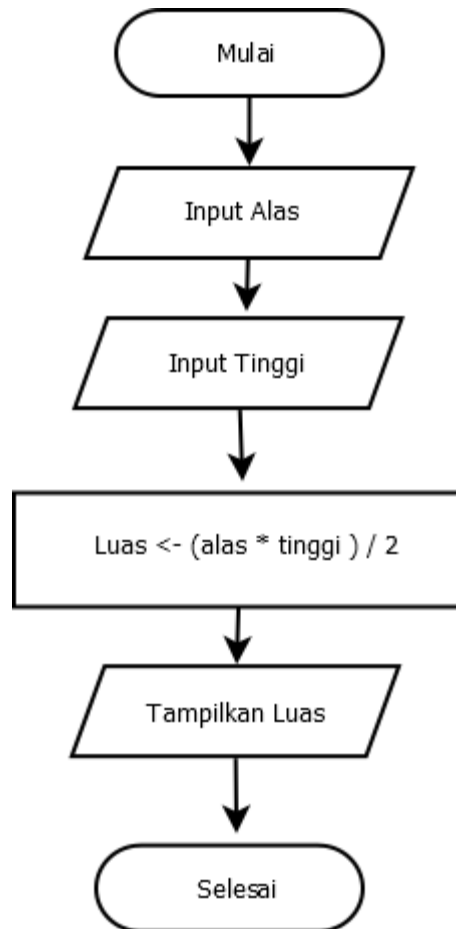
Flowchart (cont')

	Preparation
	Arrow
	Connector (on page)
	Connector (off page)
	Display

Flowchart (cont')

- Keuntungan :
lebih mudah dilihat dan dipahami
- Kelemahan :
jika digunakan untuk menggambarkan proses atau algoritma untuk skala kasus yang besar, maka akan dibutuhkan banyak kertas

Contoh :



Pseudocode

- Diarahkan untuk menyerupai aturan penulisan bahasa pemrograman tertentu

Pseudocode

Judul
Kamus
Algoritma

Judul

- Tidak diawali dengan angka atau karakter lain selain alphabet
- Tidak terdapat karakter spasi atau karakter selain alphabet kecuali karakter underscore '_' (sebagai pengganti karakter spasi)
- Mana yang benar ?
 1. Algoritma berhitung;
 2. Algoritma konversi suhu;
 3. Algoritma hitungan_pajak;
 4. Algoritma 2bilangan;
 5. Algoritma *kecil;

Kamus

- digunakan sebagai tempat untuk mencantumkan variabel, konstanta, dan *record*
- Hal-hal yang dideklarasikan pada bagian ini digunakan sebagai 'reservasi' alokasi *memory* untuk penyimpanan data dan akan digunakan selama program bekerja

Algoritma

- digunakan untuk meletakkan semua algoritma atau kode-kode program algoritma
- Diawali dengan 'BEGIN' dan diakhiri dengan 'END'

Pseudocode

- Pseudocode adalah kode yang mirip dengan kode pemrograman yang sebenarnya.
- Pseudocode berasal dari kata Pseudo yang mempunyai arti mirip atau imitasi atau menyerupai, dan code berarti program.
- Pseudocode ditulis berdasarkan pada bahasa pemrograman BASIC, PASCAL, atau C sehingga lebih tepat digunakan untuk menggambarkan algoritma program yang akan dikomunikasikan kepada programmer yang menggunakan program tsb.
- Diarahkan untuk menyerupai aturan penulisan bahasa pemrograman tertentu

Pseudocode

Judul
Kamus
Algoritma

Judul

- Tidak diawali dengan angka atau karakter lain selain alphabet
- Tidak terdapat karakter spasi atau karakter selain alphabet kecuali karakter underscore '_' (sebagai pengganti karakter spasi)
- Mana yang benar ?
 1. Algoritma berhitung;
 2. Algoritma konversi suhu;
 3. Algoritma hitungan_pajak;
 4. Algoritma 2bilangan;
 5. Algoritma *kecil;

Kamus

- digunakan sebagai tempat untuk mencantumkan variabel, konstanta, dan *record*
- Hal-hal yang dideklarasikan pada bagian ini digunakan sebagai 'reservasi' alokasi *memory* untuk penyimpanan data dan akan digunakan selama program bekerja

Algoritma

- digunakan untuk meletakkan semua algoritma atau kode-kode program
- Diawali dengan 'BEGIN' dan diakhiri dengan 'END'

Struktur Dasar Algoritma

- Algoritma adalah rangkaian langkah-langkah yang logis untuk menyelesaikan masalah.
- Ada tiga struktur dasar untuk membangun algoritma. Ketiga struktur tersebut adalah :
 - Runtunan (Sequence)
 - Pemilihan (Selection)
 - Pengulangan (Repetition)

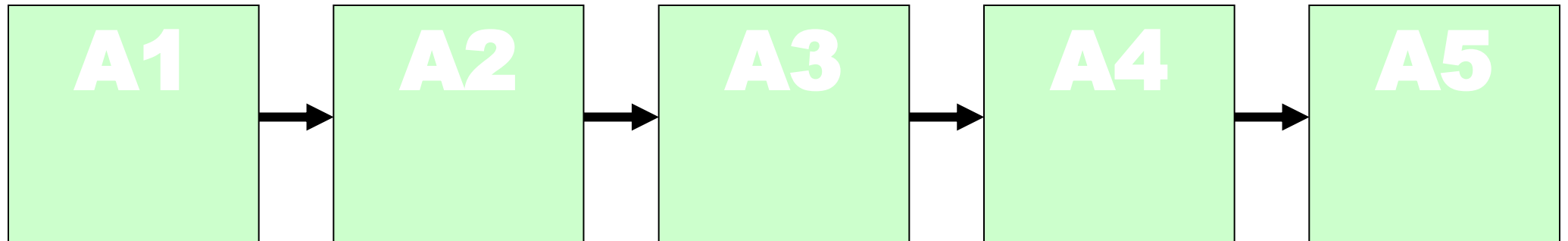
Runtunan / Sequence

Algoritma adalah merupakan runtunan proses, yang berarti :

1. Tiap proses dikerjakan satu per satu.
2. Tiap proses dilaksanakan tepat satu kali, tidak ada proses yang diulang.
3. Urutan proses yang dilaksanakan dalam pemrosesan sama dengan urutan proses sebagaimana yang tertulis di dalam algoritma.
4. Akhir dari proses terakhir merupakan akhir algoritma.

Runtunan / Sequence

Bila runtunan proses dalam algoritma dilambangkan dengan A1, A2, A3, A4 dan A5, maka urutan pelaksanaan proses-proses tersebut adalah seperti gambar di bawah ini :



Runtunan / Sequence

Contoh :

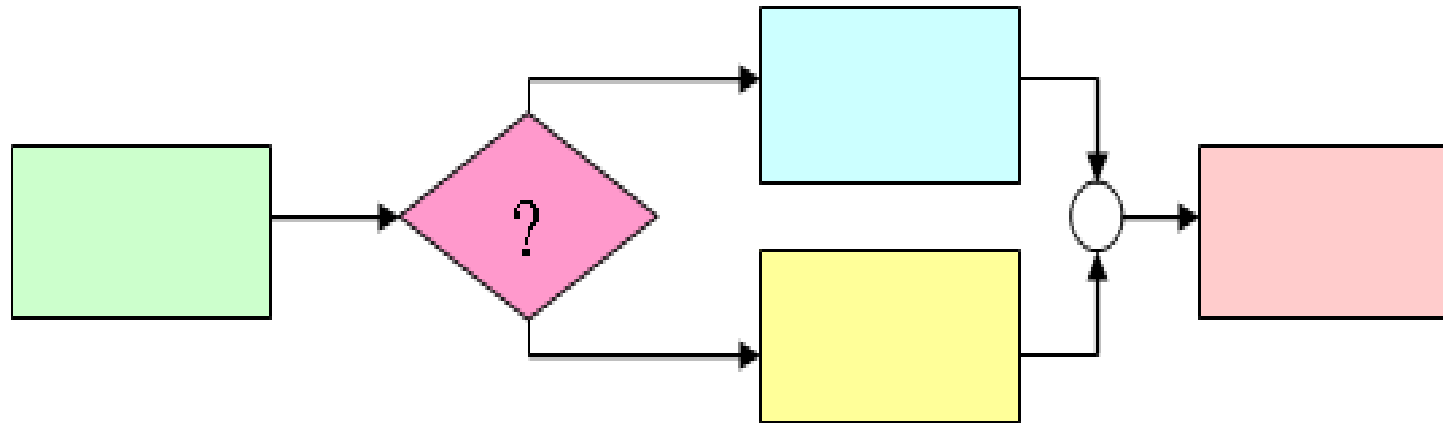
Algoritma membuat Kue Bolu.

{ Catatan : semua bahan sudah tersedia dan sudah sesuai dengan takarannya }

- *Masukan Putih Telur*
- *Kocok Sampai mengembang*
- *Masukan gula pasir*
- *Kocok Sampai merata*
- *Masukan tepung*
- *Kocok Sampai merata*

Pemilihan / Selection

- Dalam beberapa kasus tertentu kita dihadapkan kepada persoalan yang mempunyai pilihan atau beberapa jawaban yang berbeda dan salah satunya harus dipilih
- Untuk hal yang demikian, maka pelaksanaan proses adalah seperti di bawah ini :



Pemilihan / Selection

Notasi analisis satu masalah

Algoritma :

If Kondisi **Then**

Aksi

Endif

Jika *kondisi* bernilai benar Maka Aksi sesudah *then* akan dilaksanakan. Jika kondisi bernilai salah maka aksi tersebut tidak akan dikerjakan (dilewat).

Contoh:

- ***Jika*** Amir menjadi juara kelas
Maka akan dibelikan Sepeda Mini

Pemilihan / Selection

Notasi analisis beberapa masalah

Algoritma :

IF kondisi **THEN**

Aksi1

ELSE

Aksi2

ENDIF

Jika kondisi bernilai benar Maka Aksi1 sesudah then akan dilaksanakan, jika kondisi bernilai salah maka Aksi2 yang akan dikerjakan.

Pemilihan / Selection

Contoh:

- ***Jika*** Amir menjadi juara kelas
 Maka akan dibelikan Sepeda Mini
 Selain itu mendapatkan cat crayon

Pengulangan / Looping

- Selain dapat melakukan proses pemilihan aksi, komputer juga dapat melakukan aksi pengulangan.
- Aksi pengulangan ini dilakukan sebanyak proses yang diinginkan atau sampai kondisi tertentu dipenuhi.
- Secara umum pengulangan terdiri atas dua bagian :
 - 1. *Kondisi***, yaitu ekspresi boolean yang mengakibatkan pengulangan berjalan dan sampai suatu saat berhenti.
 - 2. *Aksi*** yang diulang selama kondisi memenuhi atau belum memenuhi kondisi berhenti.

Pengulangan / Looping

- Di dalam program terdapat beberapa notasi pengulangan yang berbeda, demikian juga dengan notasi dalam algoritma.
- Beberapa notasi dapat digunakan untuk satu masalah yang sama, namun ada notasi pengulangan tertentu yang hanya cocok dipakai untuk permasalahan tertentu.
- Terdapat 3 macam jenis perintah pengulangan

Pengulangan / Looping

Notasi FOR - DO

For Kondisi ← nilai *awal* To nilai *akhir* **do**
aksi

Endfor

- Aksi di dalam badan pengulangan akan dilakukan dari **nilai awal** sampai dengan ***nilai akhir*** dipenuhi.
- Apabila satu aksi pertama telah dilakukan maka secara otomatis variable yang tadinya mempunyai nilai awal akan berubah ke nilai berikutnya berdasarkan pencacah tertentu.
- Misal nilai awal 1 menjadi 2 -> pencacah +1
- Atau nilai awal 10 menjadi 9 -> pencacah -1
- Atau nilai awal 1 menjadi 3 -> pencacah +2

Pengulangan / Looping

Notasi Repeat – Until

Notasi Algoritma :

Repeat

Aksi

Until kondisi

- Aksi di dalam badan pengulangan akan dilakukan selama kondisi bernilai benar.
- Apabila bernilai salah, maka pengulangan akan berhenti.
- Namun demikian diperlukan suatu perintah untuk mengubah kondisi sehingga memungkinkan untuk kondisi tidak dipenuhi.
- Apabila tidak ada perintah untuk mengubah kondisi yang telah ada, maka proses pengulangan tidak akan pernah berhenti.

Pengulangan / Looping

Notasi While – Do

Notasi Algoritma :

While kondisi **do**

Aksi

Endwhile

- Aksi di dalam badan pengulangan akan dilakukan selama kondisi bernilai benar.
- Apabila bernilai salah, maka pengulangan akan berhenti.
- Namun demikian diperlukan suatu perintah untuk mengubah kondisi sehingga memungkinkan untuk kondisi tidak dipenuhi.

Pengulangan / Looping

- Apabila tidak ada perintah untuk mengubah kondisi yang telah ada, maka proses pengulangan tidak akan pernah berhenti.
- Jika pada ***Repeat-Until*** minimal akan dilakukan aksi sebanyak 1 kali.
- Sedangkan ***While-Do*** adalah ada kemungkinan aksi tidak dijalankan karena kondisi dari awal bernilai salah.

Pengulangan / Looping

Contoh :

Algoritma Pengulangan_While_Do

Kamus Lokal :

J,N : Integer

Algoritma :

Input(N)

$J \leftarrow 1$

WHILE $J \leq N$ **DO**

 Output (Universitas)

$J \leftarrow J+1$

ENDWHILE

Setelah melakukan proses Input N maka algoritma ini akan mencetak tulisan 'Universitas' selama nilai J lebih kecil atau sama dengan nilai N.

Pemilihan (*Selection*)

```
If kondisi then  
    aksi  
endif
```

```
If kondisi then  
    aksi1  
else  
    aksi2  
endif
```

Contoh (Pemilihan) :

If A>B then Max \leftarrow A endif

If B>A then Max \leftarrow B endif

If A>B then Max \leftarrow A
Else Max \leftarrow B
endif

Pengulangan (*Repetition*)

```
for var  $\leftarrow$  awal to akhir do  
    aksi  
Endfor
```

```
repeat  
    aksi  
until kondisi_stop
```

```
while kondisi_ulang do  
    aksi  
endwhile
```

Contoh (Pengulangan) :

```
For i  $\leftarrow$  1 to 5 do  
    output("Algoritma dan Pemrograman")  
EndFor
```

```
i  $\leftarrow$  1  
Repeat  
    output("Algoritma dan Pemrograman")  
    i  $\leftarrow$  i+1  
Until (i>5)
```

```
i  $\leftarrow$  1  
While (i<=5) do  
    output("Algoritma dan Pemrograman")  
    i  $\leftarrow$  i+1  
endwhile
```

Input dan Output

Algoritma Masukkan_data

Kamus data

x : integer

BEGIN

input(x) /*x adalah variabel penampung nilai*/

END.

Algoritma Masukkan_nilai

Kamus data

x : integer

BEGIN

$x \leftarrow 5$ /*panah ke kiri arah masuknya nilai*/

END.

Input dan Output

Algoritma Tampilan

Kamus data

x : integer

BEGIN

output(x) /*x adalah variabel yang berisi nilai*/

END.

Reference

- Rinaldi Munir. “Algoritma dan Pemrograman”
- Inggriani Liem.”Diktat Kuliah Algoritma dan Pemrograman”. ITB
- “Diktat Kuliah Algoritma dan Pemrograman”. Politeknik Telkom