

Algoritma dan Pemecahan Masalah

Pertemuan III

TIPE, DATA DAN HARGA

Pada umumnya, program komputer bekerja dengan memanipulasi objek (data) didalam memory. Objek yang akan diprogram bermacam-macam jenis atau tipenya, misalnya bilangan, karakter, string, dan rekaman (*record*). Suatu tipe menyatakan pola penyajian data dalam komputer. Kita harus mendefinisikan tipe data yang dapat diprogramkan dengan komputer.

Mendefinisikan tipe data berarti :

- a. Menentukan nama tipe data itu
- b. Mendefinisikan ranah (*domain*) nilai yang dapat dipunyai oleh nama tersebut
- c. Perjanjian tentang cara menulis tetapan (*constant*) bertipe tersebut
- d. Operator yang dapat dioperasikan terhadap objek bertipe tersebut

Tipe data dapat dikelompokkan menjadi 2 macam yakni : **tipe data dasar** dan **tipe bentukan**. Tipe data dasar adalah tipe data yang langsung dapat dipakai, sedangkan tipe data bentukan dibentuk dari tipe dasar atau dari tipe bentukan lain yang sudah didefinisikan.

Yang termasuk kedalam tipe dasar adalah :

- a. Bilangan logik (*boolean*)
- b. Bilangan bulat :
- c. Bilangan riil, dan
- d. Karakter

Bilangan Logik

Bilangan logik dinamakan juga boolean (dari ilmuwan matematika Robert Boole). Bilangan logik hanya mengenal dua nilai, yakni : *true (1)* atau *false (0)*.

Tipe bilangan logik didefinisikan sebagai berikut :

Nama : *boolean*

Ranah : *true, false*

Tetapan: *true, false*

Operator : *not, and, or, xor*

Hasil operasi bilangan logik juga menghasilkan bilangan logik

Tabel kebenaran (Truth Table)

A	Not A
True	False
False	True

A	B	A and B	A or B	A xor B
True	True	True	True	False
True	False	False	True	True
False	True	False	True	True
False	False	False	False	false

Bilangan Bulat

Bilangan bulat atau *integer* adalah bilangan yang tidak mengandung pecahan desimal, misalnya : 0, 4, 20, 56, 75 dsb.

Tipe bilangan bulat didefinisikan sebagai berikut :

Nama	: <i>integer</i>
Ranah	: <i>/</i>
Tetapan	: <i>78 -14 7654 0 5 9991</i>
Operator	: <i>(a) Operator matematika</i>
	<i>+</i> (tambah)
	<i>-</i> (kurang)
	<i>*</i> (kali)
	<i>div</i> (bagi)
	<i>mod</i> (siswa hasil bagi)

Contoh :

3 + 10	hasil : 13
87 – 11	hasil : 76
5 * 10	hasil : 50
10 div 3	hasil : 3
10 mod 3	hasil : 1

Operator : *(b) Operator relasional/perbandingan*

$<$	<i>(lebih kecil)</i>
\leq	<i>(lebih kecil atau sama dengan)</i>
$>$	<i>(lebih besar)</i>
\geq	<i>(lebih besar atau sama dengan)</i>
$=$	<i>(sama dengan)</i>
\neq	<i>(tidak sama dengan)</i>

Contoh :

$3 < 8$	hasil : true
$75 > 100$	hasil : false
$9 \leq 45$	hasil : true
$17 = 17$	hasil : true
$98 \neq 25$	hasil : true

Bilangan Riil

Tipe bilangan riil didefinisikan sebagai berikut :

Nama : *Real*

Ranah : *R*

Tetapan : *0. 18. 23.1 0.333 9E-6 -3.0085*

Operator : *(a) Operator matematika*

+ (tambah)

- (kurang)

*** (kali)

/ (bagi)

Contoh :

$3.6 + 2.3$

hasil : 5.9

$8.0 - 2.8$

hasil : 5.2

$10/3$

hasil : 3.333..

$7.2 * 0.5$

hasil : 3.6

Operator : *(b) Operator relasional/perbandingan*

<	<i>(lebih kecil)</i>
≤	<i>(lebih kecil atau sama dengan)</i>
>	<i>(lebih besar)</i>
≥	<i>(lebih besar atau sama dengan)</i>
≠	<i>(sama dengan)</i>

Contoh :

0.03 < 0.3	hasil : false
8.0 ≥ 5.0	hasil : true
3.0 ≠ 3.5	hasil : true

Karakter

Tipe karakter didefinisikan sebagai berikut :

Nama	: <i>char</i>
Ranah	: <i>(‘0’, ‘1’, .. , ‘9’, ‘a’, ‘b’, .. , ‘z’, ‘A’, ‘B’, .. , ‘Z’, ‘ ‘, ‘.’, ‘#’, ‘@’, dan karakter khusus lainnya)</i>
Tetapan	: <i>‘a’ ‘Y’ ‘-’ ‘ ‘ ‘9’ ‘\$’</i>

Operator : *Operator relasional/perbandingan*

=	(sama dengan)
≠	(tidak sama dengan)
<	(lebih kecil)
≤	(lebih kecil atau sama dengan)
>	(lebih besar)
≥	(lebih besar atau sama dengan)

Contoh :

'a' = 'a'	hasil : true
'T' = 't'	hasil : false
'y' ≠ 'Y'	hasil : true
'Q' > 'Z'	hasil : false
'm' < 'z'	hasil : true

Tipe bentuk dibentuk dari beberapa elemen yang bertipe dasar atau dari tipe bentuk lain yang sudah didefinisikan sebelumnya. Tipe bentuk disebut juga dengan tipe terstruktur. Tipe bentuk diberi nama oleh pemrogram.

Ada dua jenis tipe bentuk, yakni *String* dan *Rekaman (record)*

1. String

String adalah deretan karakter dengan panjang tertentu.

Tipe string didefinisikan sebagai berikut :

Nama : *string*

Ranah : *deretan karakter yang didefinisikan pada ranah karakter*

Tetapan: 'BANDUNG' 'Ganesha' 'Jurusan Sistem Komputer' 'ABC 7865'

Semua tetapan string harus diapit oleh tanda petik tunggal

Operator : (a) *Operator penggabungan (concatenation)*
+
(b) *Operator relasional/perbandingan*
= (sama dengan)
≠ (tidak sama dengan)
< (lebih kecil)
≤ (lebih kecil atau sama dengan)
> (lebih besar)
≥ (lebih besar atau sama dengan)

Contoh :

'a' + 'b' = 'ab'

'1' + '2' = '12'

'Sistem ' + 'Komputer' = 'Sistem Komputer'

' aa' + ' bb'+ 'cc' = ' aa bbcc'

'abcd' = 'abc' hasil : false

'aku' < 'AKU' hasil : true

String yang disusun oleh gabungan numerik dan karakter string dinamakan dengan *alfanumeric*

2. Rekaman

Sebuah rekaman disusun oleh satu atau lebih *field*. Tiap *field* berisi data dari tipe dasar tertentu atau tipe bentukan lain yang sudah didefinisikan sebelumnya. Nama rekaman ditentukan oleh pemrogram. Sebuah rekaman dengan tiga field dapat digambarkan sbb :

FIELD1	FIELD2	FIELD3
--------	--------	--------

Berikut adalah contoh beberapa cara untuk mendefinisikan tipe data rekaman :

1.Didefinisikan tipe bentukan yang menyatakan data mahasiswa yang terdiri atas NIM, nama dan usia. Misalkan tipe bentukan ini diberi nama MHS. Cara mendefinisikan adalah :

Type MHS : record <NIM : integer, nama : string, usia : integer>

Jika dideklarasikan **M** adalah peubah bertipe MHS, maka cara mengacu tiap field pada rekaman M adalah :

M.NIM

M.nama

M.usia

Nama : MHS

Ranah : sesuai dengan ranah masing-masing *field*

Tetapan: <0910453084, 'Samurya Rahmadhoni', 19>

<0910453058, 'Hafis Fajri', 18>

Operator : Tidak ada operator untuk MHS tetapi kita dapat melakukan :

- operasi *integer* terhadap MHS.NIM
- operasi string pada MHS.nama
- operasi *integer* pada usia

2. Titik dalam koordinat kartesian dinyatakan sebagai (x,y), dengan x adalah nilai arah sumbu X dan y adalah nilai dalam arah sumbu Y, x dan y R. Kita dapat menyatakan titik sebagai tipe bentukan dengan x dan y sebagai fieldnya sbb :

Type Titik : record <x : real, y : real>

Jika dideklarasikan **A** adalah peubah bertipe Titik, maka cara mengacu tiap field pada rekaman A adalah :

A.x

A.y

Tipe Titik didefinisikan sebagai berikut :

Nama : Titik

Ranah : (real, real)

Tetapan : <2.7, 15.2> <-1.4, -6.5>

Operator : operator riil terhadap A.x dan A.y

3. Didefinisikan tipe bentukan yang mewakili tanggal dalam kalender Masehi. Hari dinyatakan sebagai tanggal, bulan dan tahun. Misalkan tipe tersebut diberi nama TGL.

type tanggal	: integer[1..31]	
type bulan	: integer[1..12]	
type tahun	: integer > 0	
type TGL	: record <tangg	: tanggal,
	bul	: bulan
	thn	: tahun>

Jika **DATE** adalah peubah bertipe TGL, maka cara mengacu tiap field adalah :

DATE.tang [1..31]

DATE.bul [1..12]

DATE.thn > 0

Tipe TGL didefinisikan sebagai berikut :

Nama : TGL

Ranah : Sesuai ranah masing-masing field

Tetapan : <12, 7, 1997> <31, 12, 2008> <29, 2, 2009>

Operator : bergantung tipe tiap field

4. Jadwal keberangkatan kereta api

type jam : integer[0..23]

type menit : integer[0..59]

type WAKTU : record (j : jam, m : menit)

type JADWAL_KA : record <NoKA : string,
KotaAsal : string,
JamBerangkat: WAKTU,
KotaTujuan : string,
JamTiba : WAKTU>

Jika JKA peubah bertipe JADWAL_KA, maka cara mengacu tiap-tiap field adalah :

JKA.NoKA

JKA.KotaAsal

JKA.JamBerangkat.j

JKA.JamBerangkat.m

JKA.KotaTujuan

JKA.JamTiba.j

JKA.JamTiba.m

Tipe JADWAL_KA didefinisikan sebagai berikut :

Nama : JADWAL_KA

Ranah : Sesuai ranah masing-masing field

Tetapan : <'KA01', 'Jakarta', <17,2>, 'Seamarang', <05,54>>
<'KA02', 'Bandung', <9,40>, 'Solo', <14,10>>

Operator : sesuai operator masing-masing tipe field

Nama

Setiap objek dalam mempunyai nama. Nama itu didefinisikan oleh manusia agar objek tersebut mudah diidentifikasi dan dibedakan dari objek yang lain. Didalam program, nama dipakai sebagai pengidentifikasi “sesuatu” dan pemrogram mengacu “sesuatu” itu melalui namanya. Karena itu tiap nama haruslah unik, tidak boleh ada dua buah nama yang sama.

Didalam program “sesuatu” yang diberi nama dapat berupa :

1. Peubah (*variabel*)
2. Tetapan/konstanta
3. Tipe
4. Nama fungsi
5. Nama prosedur

Sebuah nama memiliki aturan penulisan, yakni :

1. Nama harus dimulai dengan huruf alfabet
2. Karakter penyusun nama hanya boleh alfabet, angka dan “_” (underscore). Namun tidak boleh menggunakan simbol/karakter khusus lainnya seperti operator, aritmatika, operator relasional, dan karakter khusus lainnya.

3. Karakter didalam nama tidak boleh dipisah dengan spasi. Spasi dapat diganti dengan karakter “_” (underscore)
4. Panjang nama tidak dibatasi

Berikut adalah contoh penamaan :

Penamaan yang salah	Penamaan yang benar
6Titik	Titik6 atau Titik_6
Nilai Ujian	NilaiUjian atau Nilai_Ujian
PT-1	PT_1 atau PT1
hari@	Hari
A 1	A1

Berikut ini adalah Konversi Nama, Tipe dan Operator kedalam Bahasa Pascal dan C

KELOMPOK	ALGORITMA	PASCAL	C
1. Tipe Dasar	Boolean Integer Real Char String Record	boolean integer real char string record	Tidak ada *) Int Float Char Tidak ada **) Struct
2. Operator	+ - * / div mod < ≤ > ≥ = ≠ not and or xor	+ - * / div Mod < <= > >= := <> not and or xor	+ - * / / % < <= > >= = != ! && Tidak ada

KELOMPOK	ALGORITMA	PASCAL	C
3. Lain-lain	const type true false	const type true false	#define typedef tidak ada tidak ada

*) Bahasa C tidak menyediakan type boolean yang eksplisit. Namun tipe Boolean dapat didefinisikan dengan typedef sebagai berikut :

```
typedef enum (false=0, true=1) boolean;
```

Bahasa C tidak menyediakan tipe string secara eksplisit. Nama peubah yang bertipe string didefinisikan sebagai char disertai panjang string tersebut :

```
char nama_peubah[n]
```

Dengan n adalah panjang string (termasuk didalamnya karakter null yang ditambahkan secara otomatis) yang akan ditampung oleh nama_peubah.

Contoh : char kota[20]

Panjang maksimum string yang dapat ditampung oleh peubah kota dalah 19 + karakter null. Dalam bahasa Pascal, string yang tidak disertai dengan panjang dianggap panjangnya 255.

Harga

Harga adalah nilai besaran yang sudah dikenal. Harga berupa nilai yang dikandung oleh nama peubah atau nama tetapan. Harga yang dikandung oleh peubah dimanipulasi dengan cara : mengisikannya ke peubah lain yang bertipe sama, dipakai untuk perhitungan, atau dituliskan ke piranti keluaran.

Beberapa cara pengisian harga ke dalam nama peubah :

1. Penugasan (*assignment*)

adalah mengisikan sebuah harga pada nama peubah secara langsung. Aksi penugasan dilambangkan dengan tanda “ ←”. Harga yang diberikan dapat berupa tetapan, harga nama peubah lain, atau nilai sebuah ekspresi.

Notasi algoritma untuk penugasan :

Nama \leftarrow tetapan (harga tetapan diisikan kedalam nama)

Nama1 \leftarrow nama2 (harga nama2 disalin kenama1)

Nama \leftarrow ekspresi (hasil perhitungan diisikan kedalam nama)

Contoh :

Dalam Kamus sudah didefinisikan nama K bertipe integer, Jarak bertipe real, Ketemu bertipe boolean dan NamaKota bertipe string. Maka cara pengisian nama peubah tersebut dengan harga tetapan adalah :

ALGORITMA

K \leftarrow 5

Jarak \leftarrow 0.03

Ketemu \leftarrow false

NamaKota \leftarrow 'Bandung'

Dari contoh penugasan ini berarti elemen memory yang masing-masing bernama K, Jarak, Ketemu, dan NamaKota diisi berturut-turut 5, 0.03, false, 'Bandung'. Kadaan dimemory diperlihatkan pada gambar berikut ini :

	.
	.
K	5
Jarak	0.03
Ketemu	False
NamaKota	'Bandung'
	.
	.

2. Pembacaan Harga dari Piranti Masukan

Harga untuk nama dapat diisi dari piranti masukan, misalnya papan ketik. Mengisi harga dari piranti masukan dinamakan operasi *pembacaan* data. Istilah 'baca' ini timbul karena komputer seakan-akan membaca harga yang diberikan oleh pemakai. Dalam algoritma, pembacaan harga untuk nama peubah dilakukan dengan perintah ***input***.

Notasi Algoritma untuk pembacaan harga dari piranti masukan adalah :

ALGORITMA

Input(nama1, nama2, ... , namaN)

Dengan syarat nama1, nama2, .. , namaN adalah peubah dan tidak boleh tetapan.

Contoh :

Misalkan M, a1, a2,, a3 bertipe real, nama_kota dan nama_mhs bertipe string, NIP dan nilai bertipe integer, dan P adalah rekaman. Cara pembacaan harga untuk semua peubah adalah :

ALGORITMA

```
input(M)
input(a1, a2, a3)
input(nama_kota)
input(nama_mhs, NIP, nilai)
input(P.x, P.y)
```

Ekspresi

Suatu harga dipakai untuk proses transformasi menjadi keluaran yang diinginkan. Transformasi harga menjadi keluaran dilakukan melalui suatu perhitungan (komputasi). Cara perhitungan itu dinyatakan dengan suatu ekspresi. Suatu ekspresi terdiri dari operand dan operator. Operand adalah harga yang dioperasikan dengan operator tertentu. Dikenal 3 macam ekspresi, yakni *ekspresi numerik*, *ekspresi boolean* dan *ekspresi relasional*.

Menuliskan harga ke piranti keluaran

Harga yang disimpan didalam memory dapat ditampilkan ke piranti keluaran (misalnya layar peraga atau printer). Perintah penulisan harga adalah dengan *output*.

Notasi penulisan harga :

ALGORITMA

```
output(nama1, nama2, .. , namaN)  
output(tetapan)  
output(nama, tetapan, ekspresi)  
output(ekspresi)
```

Nama1, nama2, .. , namaN dapat berupa nama peubah atau nama tetapan.

Contoh :

Misalkan A, B, nama_mhs, Nobp, dan nilai telah didefinisikan tipenya di dalam kamus. Algoritma untuk mencetak harga yang disimpan oleh nama-nama tersebut adalah :

ALGORITMA

```
output(100)
output(A)
output('A')
output('Program Studi Sistem Komputer')
output('Nilai A = ',A)
output(nama_mhs, nobp, nilai)
output(A+B)
output('Nilai seluruhnya ', A+B/2 * 10)
```

Tabel Konversi Penugasan, Pembacaan, Penulisan ke dalam Bahasa Pasca dan C

KELOMPOK	ALGORITMA	PASCAL	C/C++
Penugasan	←	:=	=
Pembacaan	input	read readln	scanf; cin
Penulisan	output	write writeln	Printf;cout

Algoritma dan Pemecahan Masalah

PENYAJIAN ALGORITMA

Secara garis besar Algoritma dapat disajikan dengan dua teknik yaitu **teknik tulisan/teks** dan **teknik gambar**.

Teknik tulisan/teks biasanya menggunakan *English Structure* atau *Pseudocode*. Basis dari *English Structure* adalah bahasa Inggris, tetapi juga dapat ditulis dalam bahasa Indonesia. Sedangkan teknik gambar biasanya menggunakan *metode structure chart*, *hierarchy plus input-process-output*, *flowchart* dan *Nassi Schneiderman chart*. (pembahasan hanya pada *flowchart*)

Penulisan Teks Algoritma

Pada dasarnya teks algoritma selalu disusun oleh tiga bagian/blok, yakni : judul algoritma, kamus dan algoritma.

Algoritma dan Pemecahan Masalah

Setiap blok disertai dengan komentar untuk memperjelas maksud teks yang dituliskan. Komentar adalah kalimat yang diapit oleh tanda kurung krawal “{}” atau kurung buka dan asterisk dan asterisk kurung tutup “(* *)”

JUDUL

{penjelasan tentang algoritma, yang berisi uraian singkat mengenai apa yang dilakukan oleh algoritma}

KAMUS

{semua nama yang dipakai, meliputi tipe, nama tetapan, nama peubah, nama prosedur dan nama fungsi yang didefinisikan}

ALGORITMA

{semua langkah/aksi algoritma dituliskan disini}

Algoritma dan Pemecahan Masalah

Contoh Algoritma menghitung luas segitiga.

HIT_LUAS_SEGITIGA

{menghitung luas segitiga untuk panjang dan tinggi tertentu. Panjang alas (a) dan tinggi segitiga (t) dibaca dari piranti masukan. Luas segitiga adalah $L = 1/2 at$ }

KAMUS

a : real {panjang alas segitiga, dalam satuan cm }
t : real {tinggi srgitiga, dalam satuan cm }
L : real {luas segitiga, dalam satuan cm }

ALGORITMA

input(a,t) ←
L $a*t/2$
output(L)

Terjemahan algoritma kedalam Program Pascal dan C :

Pascal :

Program HIT_LUAS_SEGITIGA;

{menghitung luas segitiga untuk panjang segitiga dan tinggi tertentu. Panjang alas (a) dan tinggi segitiga (t) dibaca dari piranti masukan. Luas segitiga adalah $L=1/2 at$ }

KAMUS

Var

a	: real;	{panjang alas segitiga, dalam satuan cm}
t	: real;	{tinggi segitiga, dalam satuan cm}
L	: real;	{luas lingkaran, dalam satuan cm}

ALGORITMA

Begin

```
write('Ketikkan panjang alas segitiga : ');readln(a);
write('Ketikkan tinggi segitiga           : ');readln(t);
L := a*t/2;
writeln('Luas segi tiga           = ', L:4:2);
```

End.

C:

PROGRAM HIT_LUAS_SEGITIGA

/* menghitung luas segitiga untuk panjang segitiga dan tinggi tertentu. Panjang alas (a) dan tinggi segitiga (t) dibaca dari piranti masukan. Luas segitiga adalah $L = \frac{1}{2} a t$ */

```
#include <stdio.h>
```

```
main()
```

```
{
```

KAMUS

```
float a; /*panjang alas segitiga, dalam satuan cm */
```

```
float t; /*tinggi segitiga, dalam satuan cm */
```

```
float L;          /*luas lingkaran, dalam satuan cm2 */
```

ALGORITMA

```
printf ("Ketikkan panjang alas segitiga      : "); scanf("%f", &a);
```

```
printf("Ketikkan tinggi segitiga : "); scanf("%f", &t);
```

```
L = a*t/2;
```

```
printf("Luas segi tiga =  " "%f \n", L);
```

```
}
```

Beberapa hal penting yang harus diketahui dari bahasa C :

1. Bahasa C bersifat *case sensitive*. Artinya bahasa C membedakan huruf besar (kapital) dengan huruf kecil . Nama yang ditulis dengan huruf besar dianggap berbeda kalau ditulis dengan huruf kecil).

Misalnya :

N tidak sama dengan *n*

nama_orang tidak sama dengan *NAMA_ORANG*

HitKar tidak sama dengan *hitkar*

Berbeda dengan Pascal, Pascal tidak memberdakan nama yang ditulis dengan huruf besar dan huruf kecil. Jadi :

N sama dengan *n*

nama_orang sama dengan *NAMA_ORANG*

HitKar sama dengan *hitkar*

2. Dalam bahasa C, kamus yang didefinisikan diluar **sebelum kata main()** dianggap sebagai kamus **global**, artinya semua nama yang didefinisikan didalam kamus akan dikenal diseluruh bagian program, termasuk didalam fungsi dan prosedur yang digunakan. Apabila kamus didefinisikan **di dalam main()**, maka nama didalam kamus hanya dikenal oleh program utama saja dan tidak dikenal oleh fungsi atau prosedur.
3. Dalam bahasa C komentar ditulis diantara **“/*”** dan **“*/”**.
Sedangkan dalam bahasa Pascal komentar dapat ditulis diantara **“{ dan }”** atau diantara **“(dan *)”**.

Algoritma dan Pemecahan Masalah

Pseudo Code

Pseudo artinya **mirip/imitasi/tiruan**.

Code artinya **program**.

Jadi Pseudo Code adalah *algoritma yang ditulis dalam bahasa yang mirip bahasa pemrograman*.

Biasanya penulisan Pseudo Code mendekati bahasa pemrograman Basic, Pascal maupun C.

Algoritma dan Pemecahan Masalah

Contoh Pesudo Code:

```
Open(ArsipMhs1,1)
Open(ArsipMhs2,2)
Fread(ArsipMhs1, Mhs)
if (Mark(Mhs) = true) then
    Output('Arsip kosong...')
else
    While (Mark(Mhs) = false) do
        JumNil = 0
        nMK = 1
        currentNIM  $\square$  Mhs.NIM {record1 dari ArsipMhs1}
        Repeat
            JumNil  $\square$  JumNil + Mhs.Nilai
            nMK  $\square$  nMK + 1
            Fread(ArsipMhs1,Mhs)
        Until (currentNIM  $\neq$  Mhs.NIM)
        Rata  $\square$  JumNil/nMK
        Fwrite(ArsipMhs2, <currentNIM, Rata>)
        Output(currentNIM,Rata)
    Endwhile
EndIf
Close(ArsipMhs1)
Close(ArsipMhs2)
```

Algoritma dan Pemecahan Masalah

FLOWCHART

Adalah suatu metode untuk menggambarkan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dimengerti, mudah digunakan dan standar.

Tujuan utama penggunaan flowchart adalah untuk menggambarkan suatu tahap penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol standar.

Dalam penulisan flowchart dikenal dua model, yaitu *sistem flowchart* dan *program flowchart*.



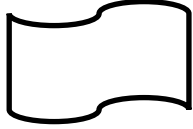


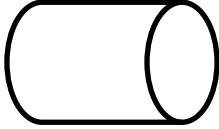


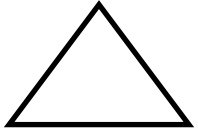
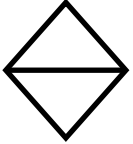

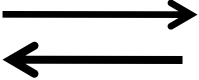
Algoritma dan Pemecahan Masalah

1.Sistem Flowchart

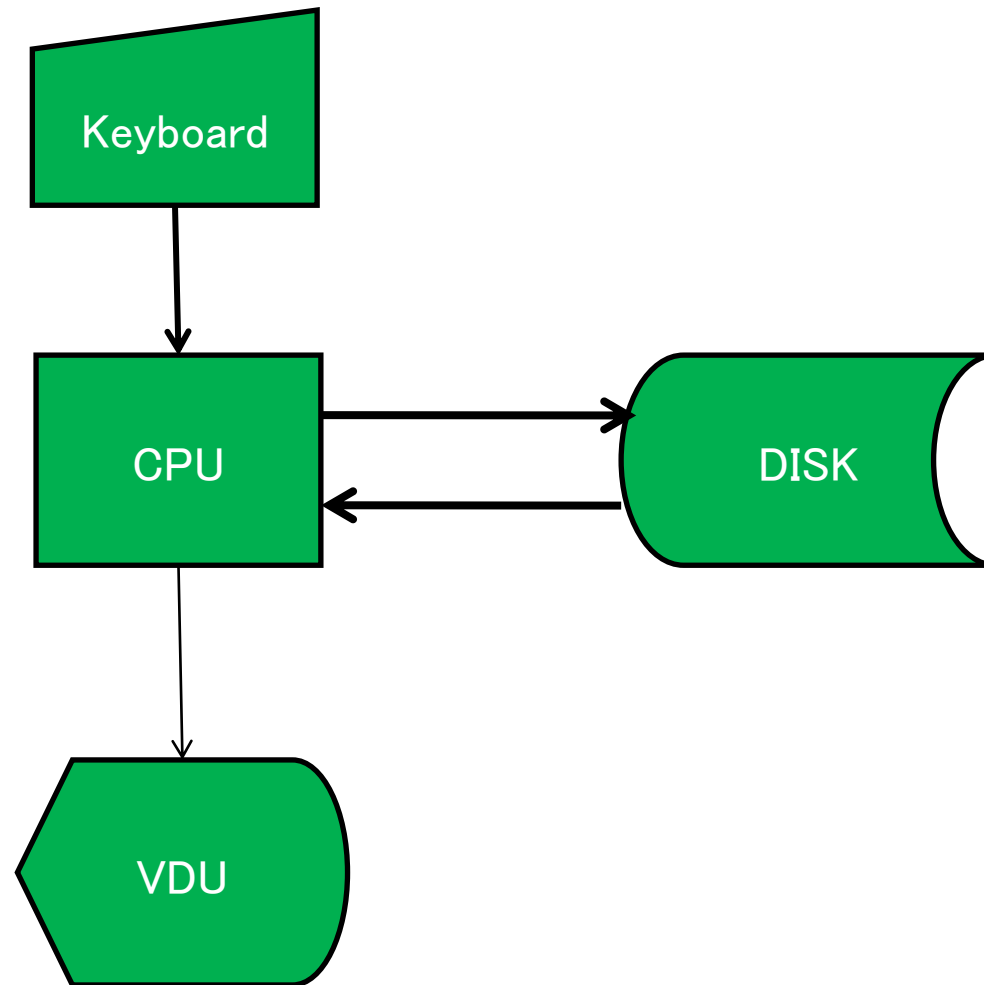
Merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut.

Berikut adalah gambar simbol-simbol standar yang banyak digunakan pada penggambaran sistem flowchart seberta contoh penggunaannya.

Algoritma dan Pemecahan Masalah




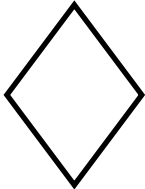


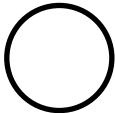

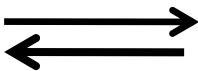
			
Pita magnetik	Kartu plong/ keyboard	Punched Paper Tape	On Line Storage / VDU
			
Input/Output	Magnetic Drum	Process	Magnetic Disc
			
Off Line Storage	Proses sortir	Proses merge	Arus/alir

Contoh penerapan sistem flowchart



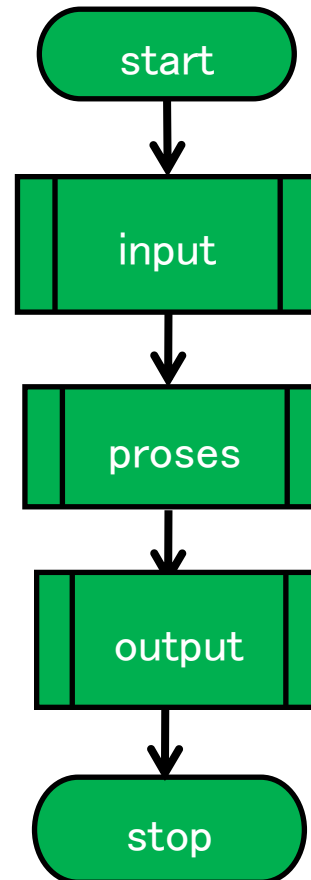
2. Program Flowchart

Merupakan diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Berikut adalah gambar dari simbol-simbol standar yang banyak digunakan pada program flowchart.

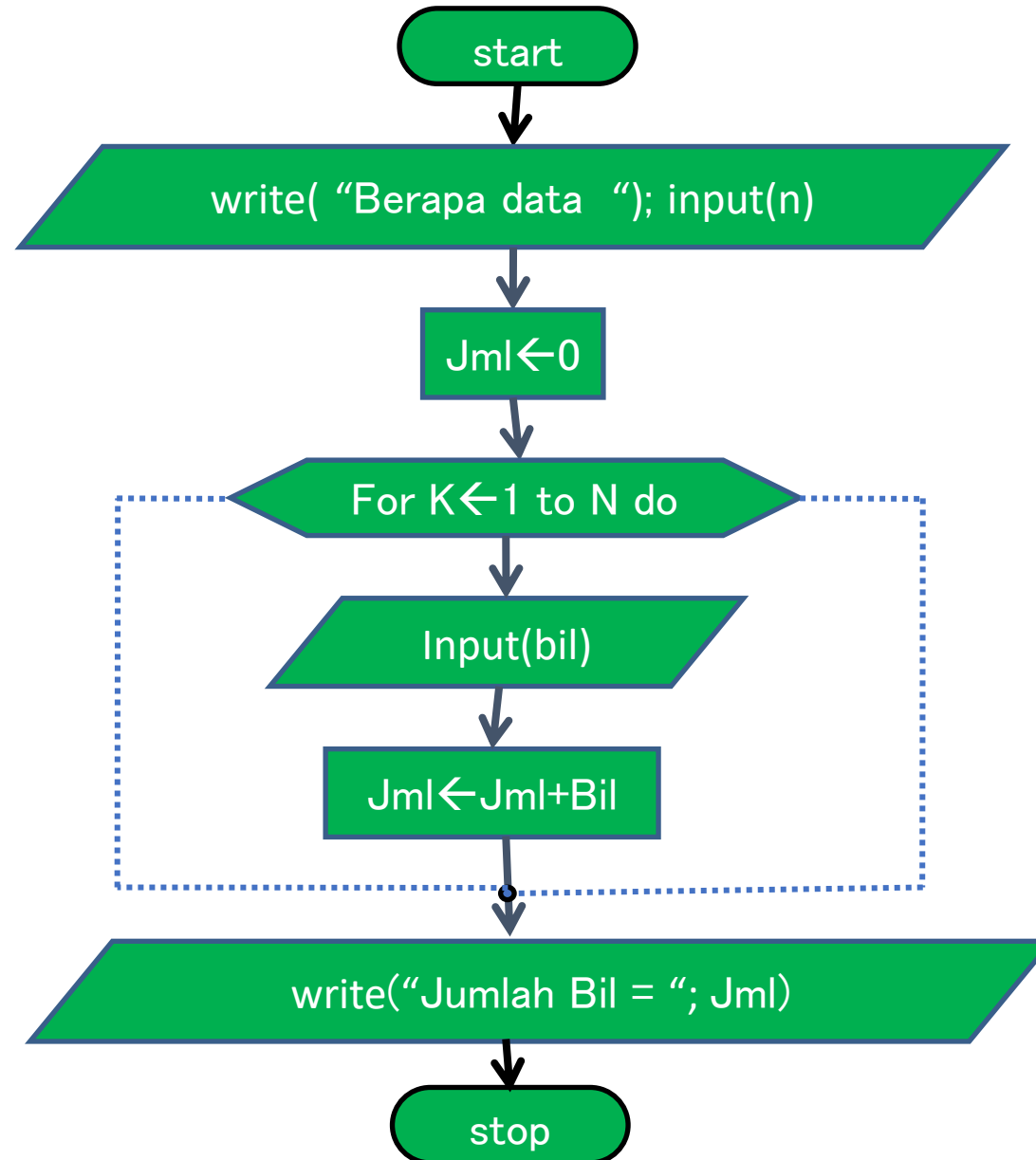
		
Proses	Input Output	Keterangan
		
Pengujian	Pemberian Nilai Awal	Awal/Akhir Program
		
Konektor pada suatu halaman	Konektor pada halaman lain	Arah/alir/arus

Pada penggambaran program flowchart terdapat dua jenis metode, yaitu *conceptual flowchart* dan *detail flowchart*. Conceptual flowchart menggambarkan tentang alur dari suatu pemecahan masalah secara global saja, sedangkan detail flowchart menggambarkan alur pemecahan masalah secara rinci.

Contoh Proses **Conceptual Flowchart**



Contoh Detail Flowchart



SELESAI