

Konsep Dasar Algoritma

TEKNIK INFORMATIKA

TA 2018-2019

ALGORITMA

- Algoritma adalah urutan aksi-aksi yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu.
- Setiap aksi harus dapat dikerjakan dan mempunyai efek tertentu.
- Algoritma adalah urutan langkah berhingga untuk memecahkan masalah logika atau matematika

ALGORITMA

- Algoritma dapat dituliskan dengan banyak cara, mulai dari menggunakan bahasa alami yang digunakan sehari-hari, simbol grafik bagan alir, sampai menggunakan bahasa pemrograman seperti bahasa C atau C++.

C & C++

- Berbicara tentang C++ biasanya tidak lepas dari C, sebagai bahasa pendahulunya. Pencipta C adalah Brian W. Kernighan dan Dennis M. Ritchie pada sekitar tahun 1972, dan sekitar satu dekade setelahnya diciptakanlah C++, oleh Bjarne Stroustrup dari Laboratorium Bell, AT&T, pada tahun 1983. C++ cukup kompatibel dengan bahasa pendahulunya C. Pada mulanya C++ disebut “ a better C “. Nama C++ sendiri diberikan oleh Rick Mascitti pada tahun 1983, yang berasal dari operator increment pada bahasa C.
- Keistimewaan yang sangat berarti dari C++ ini adalah karena bahasa ini mendukung pemrograman yang berorientasi objek (OOP / Object Oriented Programming).

Langkah dalam pemrograman komputer

- Mendefinisikan masalah
- Menentukan Solusi
- Memilih Algoritma
- Menulis Program
- Menguji Program
- Menulis Dokumentasi
- Merawat Program

Ciri Penting Algoritma

- Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
- Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (Ambiguitas).
- Algoritma memiliki nol atau lebih masukan.
- Algoritma memiliki nol atau lebih keluaran.
- Algoritma harus efektif (setiap langkah harus sederhana sehingga dapat dikerjakan dalam waktu yang masuk akal).

- Dalam kehidupan sehari-hari, sebenarnya Anda juga menggunakan algoritma untuk melakukan sesuatu. Sebagai contoh, Anda ingin menulis surat
- Bagaimana langkah-langkah menulis surat ?

- **Contoh:**

Algoritma TUKAR ISI BEJANA

Diberikan 2 buah bejana A dan B, bejana A berisi larutan berwarna merah, bejana B berisi larutan berwarna biru. Tukarkan isi kedua bejana itu sedemikian sehingga bejana A berisi larutan warna biru dan bejana B berisi larutan berwarna merah.

- Deskripsi:

1. Tuangkan larutan dari bejana A ke dalam bejana B
2. Tuangkan larutan dari bejana B ke dalam bejana A

- Algoritma TUKAR ISI BEJANA di atas tidak menghasilkan pertukaran yang benar. Langkah di atas tidak logis, hasil pertukaran yang terjadi adalah pertukaran kedua larutan tersebut.

- Untuk itu pertukaran isi dua bejana, diperlukan sebuah tambahan sebagai tempat penampungan sementara, misalnya bejana C. Maka algoritma untuk menghasilkan pertukaran yang benar adalah sebagai berikut:
- Diberikan dua buah bejana A dan B, bejana A berisi larutan berwarna merah, bejana B berisi larutan berwarna biru. Tukarkan isi kedua bejana itu sedemikian hingga bejana A berisi larutan berwarna biru dan bejana B berisi larutan berwarna merah.

Deskripsi:

1. Tuangkan larutan dari bejana A ke dalam bejana C.
2. Tuangkan larutan dari bejana B ke dalam bejana A.
3. Tuangkan larutan dari bejana C ke dalam bejana B.

Proses, Instruksi, dan Aksi

- Algoritma merupakan deskripsi urutan pelaksanaan suatu proses.
- Algoritma tersusun oleh sederetan langkah instruksi yang logis.
- Tiap langkah instruksi akan mengerjakan suatu tindakan (Aksi).
- Bila Aksi dilaksanakan, maka sejumlah operasi yang bersesuaian akan dikerjakan oleh CPU

Struktur Dasar Algoritma

- Runtunan (Sequence)
- Pemilihan (Selection)
- Pengulangan (Repetition)

Struktur Dasar Algoritma Runtunan (sequence)

- Runtunan (sequence)

- Aksi-aksi dalam algoritma yang dikerjakan secara berurutan
- Tiap instruksi dikerjakan satu per satu
- Tiap instruksi dilaksanakan tepat satu kali; tidak ada instruksi yang di ulang
- Urutan instruksi yang dilaksanakan pemroses sama dengan urutan instruksi sebagaimana yang tertulis di dalam teks algoritmanya.
- Akhir dari instruksi terakhir merupakan akhir algoritma

- Contoh :

A1. Aksi 1

A2. Aksi 2

A3. Aksi 3

- mula-mula aksi 1 dilakukan
- aksi 2 dilakukan setelah aksi 1 selesai dilaksanakan
- dst.

Struktur Dasar Algoritma Pemilihan (Selection)

- **Pemilihan (Selection)**
 - memungkinkan suatu Aksi dieksekusi jika suatu **kondisi terpenuhi** atau **tidak terpenuhi**
 - **Bentuk-bentuk pemilihan**
 - `if - else`
 - `switch - case`

Struktur Dasar Algoritma Pemilihan (Selection)

```
if (ekspresi boolean)
    Aksi 1;
    Aksi 2;
    .....
end if
```

Aksi 1 dan Aksi 2 akan diproses jika dan hanya jika nilai ekspresi Boolean = TRUE

Struktur Dasar Algoritma Pemilihan (Selection)

switch - case

- Konstruksi ini digunakan sebagai **pengganti konstruksi if-else, bila konstruksi if-else bertingkat terlalu jauh**, sehingga menjadi sulit dibaca.

```
switch (ekspresi)
{
    case constant_1 : statements_1; break;
    case constant_2 : statements_2;
    ...
    case constant_N : statements_N; break;
    default : statements;
}
```



Struktur Dasar Algoritma

- **Pengulangan (*repetition*)**

- Aksi-aksi yang **dikerjakan berulang** kali

- Bentuk-bentuk Pengulangan :

1. **FOR** pencacah pengulangan dari a sampai b **DO**

Aksi

(aksi dilakukan sebanyak hitungan pencacah pengulangan,

yaitu dari a sampai b \rightarrow yakni sebanyak $b-a+1$ kali)

2. **REPEAT**

Aksi

UNTIL kondisi

(pengulangan aksi dilakukan sehingga kondisi/persyaratan berhenti terpenuhi)

3. **WHILE** kondisi **DO**

Aksi

(selama kondisi/persyaratan pengulangan masih benar, maka aksi dikerjakan)

REPRESENTASI ALGORITMA

Representasi Algoritma (Notasi Algoritmik)

- Kalimat Deklaratif
- Pseudocode
- Flowchart (diagram alir)

Kalimat Deklaratif

Judul program/ algoritma

PROGRAM Hitung Luas Lingkaran

Program untuk menghitung luas lingkaran dengan diketahui $\text{PHI}=3.14$

Algoritma:

1. Baca r
2. Hitung Luas dengan rumus = $\text{PHI} * r * r$
3. Cetak Luas

Algoritma

Kalimat Deklaratif

Judul program/ algoritma

PROGRAM Euclidean

Program untuk mencari GCD dari dua buah bilangan bulat positif m dan n ($m \geq n$). GCD dari m dan n adalah bilangan bulat positif terbesar yang habis membagi m dan n

Algoritma:

- 1. Baca m dan n**
- 2. Jika $n=0$ maka**
 m adalah jawabannya;
 stop
 Tetapi jika $n \neq 0$
 Lanjutkan ke langkah 3
- 3. Bagilah m dengan n dan misalkan r adalah sisanya**
- 4. Ganti m dengan nilai n dan nilai n dengan nilai r , lalu ulang kembali ke langkah 1**

Algoritma

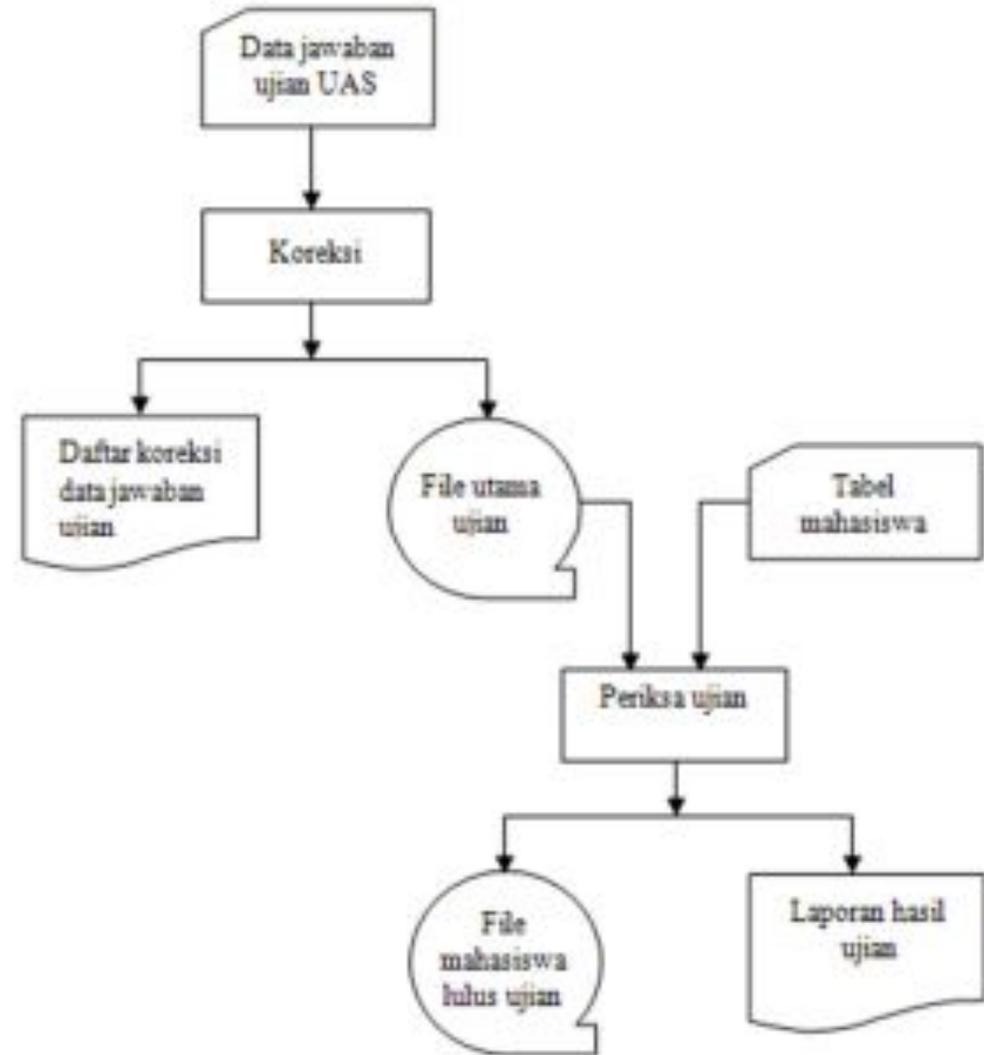
Flow Chart

- **Flow chart** suatu bagan/diagram yang menggambarkan aliran proses yang dikerjakan program dari awal sampai akhir.
- **Flow chart** adalah algoritma yang digambarkan dengan diagram.
- Fungsi dari **flow chart** adalah mendeskripsikan urutan pelaksanaan suatu proses (sama dengan fungsi algoritma).

Jenis Flowchart

- **Sistem Flowchart**

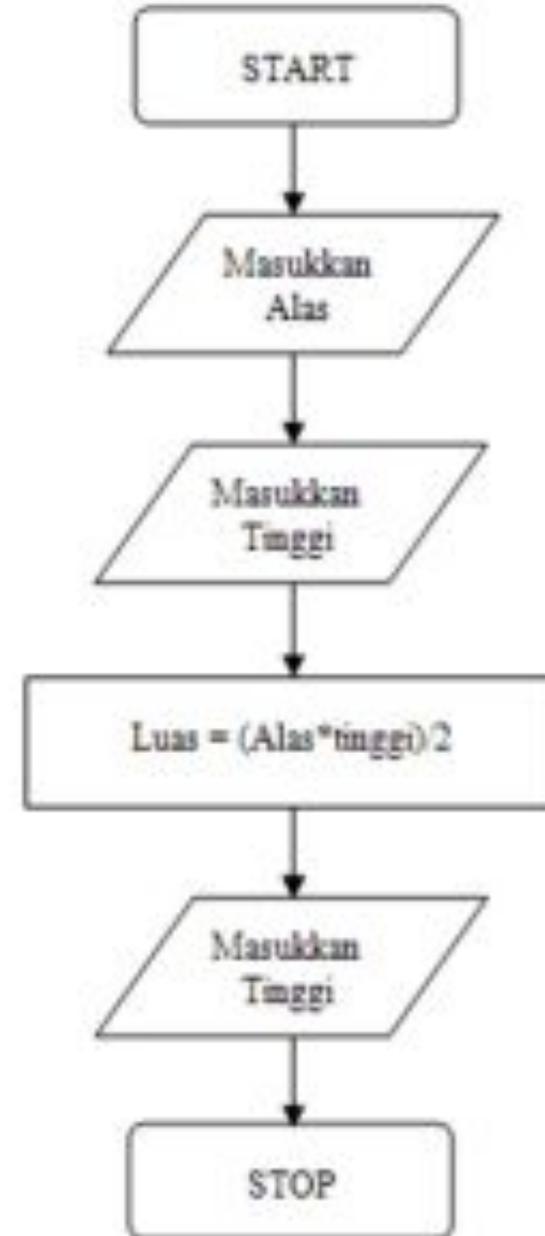
Urutan proses dalam sistem dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.



Jenis Flowchart

- **Program Flowchart**

Urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program

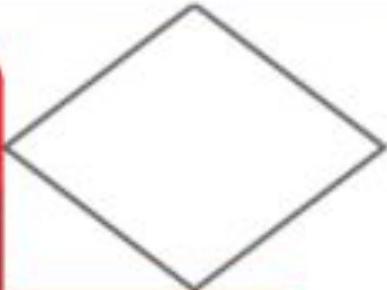


Pembuatan Flowchart

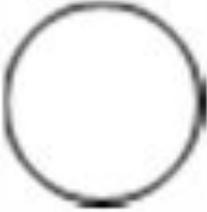
- Dalam pembuatan flowchart tidak ada kaidah yang baku.
 - Flowchart = gambaran hasil analisa suatu masalah.
 - Flowchart dapat bervariasi antara satu pemrogram dengan pemrogram lainnya.
- Secara garis besar ada 3 bagian utama:
 - **Input**
 - **Proses**
 - **Output**

- Beberapa hal yang perlu diperhatikan dalam pembuatan flowchart, yaitu :
 - Hindari pengulangan proses yang tidak perlu dan logika yang berbelit sehingga jalannya proses menjadi singkat.
 - Jalannya proses digambarkan dari atas ke bawah dan diberikan tanda panah untuk memperjelas.
 - Sebuah flowchart diawali dari satu titik **START** dan diakhiri dengan **END**.
- 

Simbol Flowchart

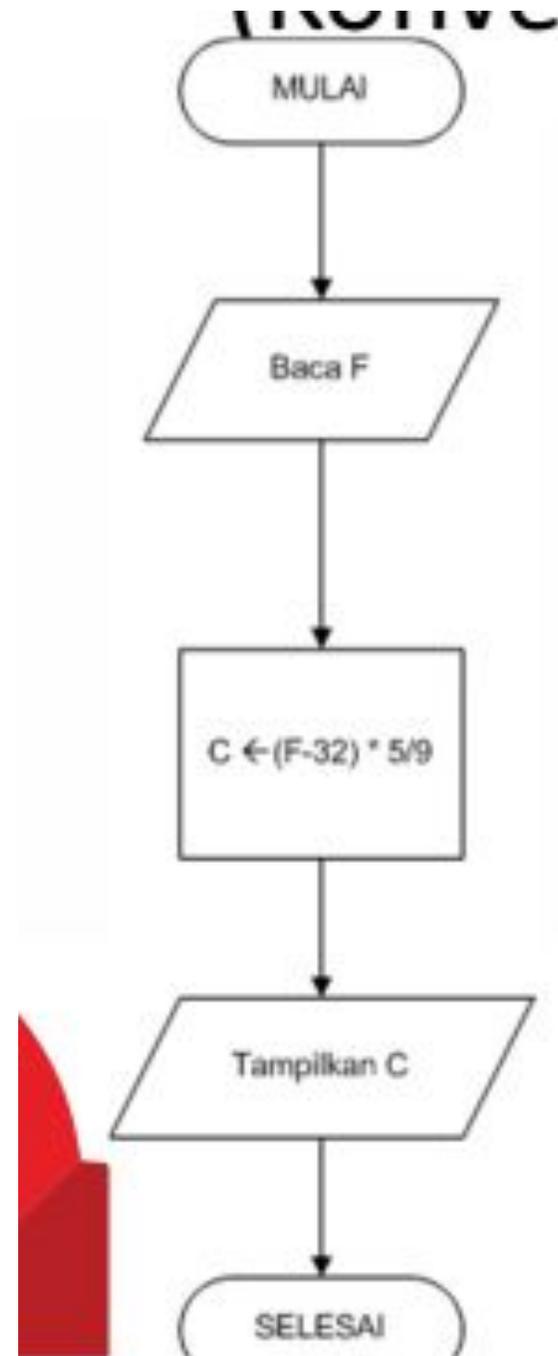
Simbol	Nama	Keterangan
	Terminator	Tanda mulai/selesai
	Input/output	Operasi pemasukan data atau penampilan data
	Proses	Menyatakan sebarang proses
	Keputusan	Pengambilan keputusan. Di dalam simbol berupa pertanyaan dengan jawaban YA atau TIDAK

Simbol Flowchart

Simbol	Nama	Keterangan
	Proses Terdefinisi	Tanda prosedur/fungsi
	Konektor	Sambungan dari satu proses ke poses yang lain dalam halaman yang sama
	Konektor	Sambungan dari satu proses ke poses yang lain dalam halaman yang berbeda

Contoh 1

(Konversi Fahrenheit - Celcius)



Contoh 2

(Menghitung Luas Lingkaran)

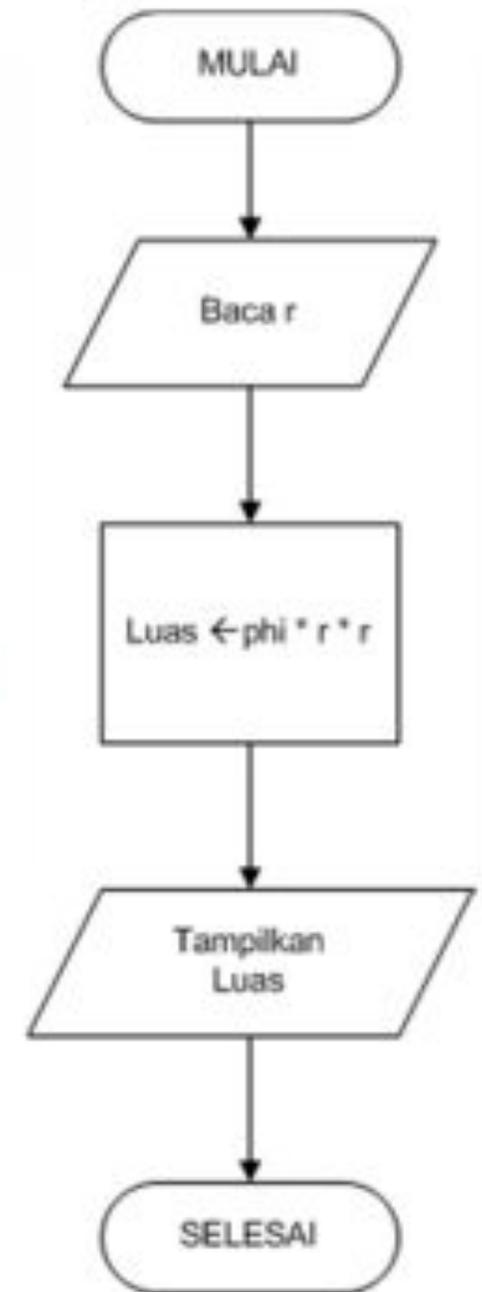
Judul program/ algoritma

PROGRAM Hitung Luas Lingkaran
Program untuk menghitung luas lingkaran dengan diketahui PHI=3.14

Algoritma:

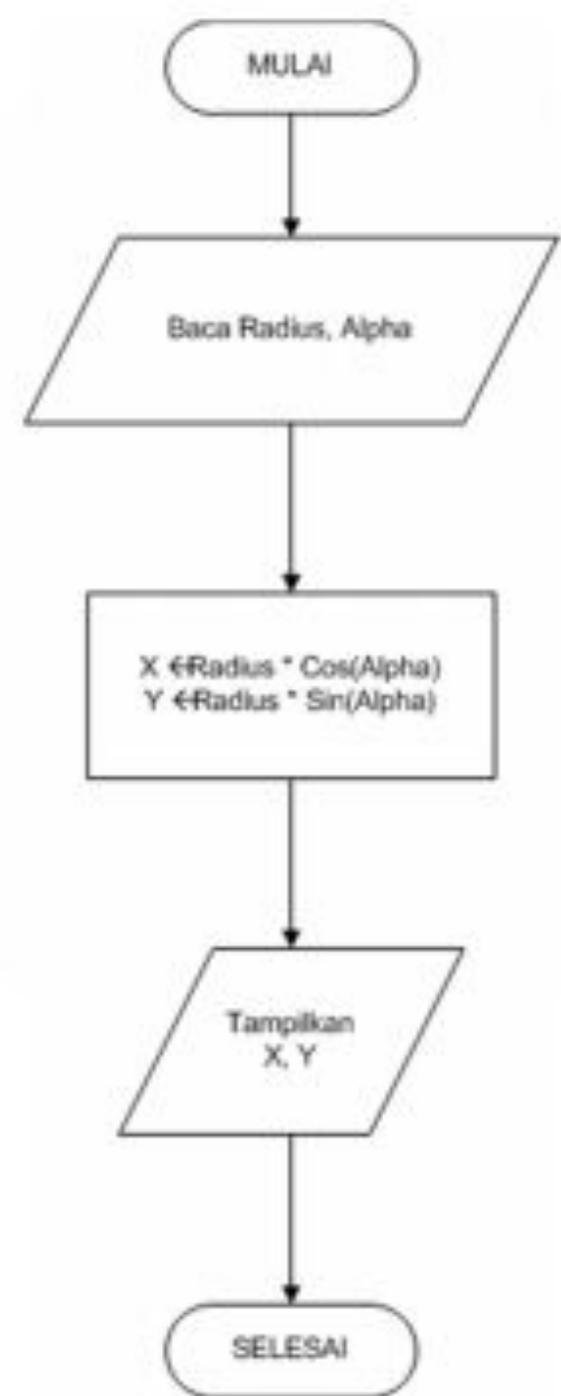
1. Baca r
2. Hitung Luas dengan rumus = $\text{PHI} * r * r$
3. Cetak Luas

Algoritma



Contoh 3

(Konversi Koordinat Polar ke Koordinat Cartesian)



Contoh 3 (Bilangan Terbesar)



Pseudocode

- Kode atau tanda yang menyerupai (*pseudo*) program atau merupakan penjelasan cara menyelesaikan suatu masalah.
- *Pseudocode* sering digunakan oleh manusia (*programmer*) untuk menuliskan algoritma sebab mudah mudah dipahami dan digunakan karena mirip dengan kode-kode program sebenarnya.

Struktur Pseudocode

- **PROGRAM Nama Program**

{Penjelasan tentang algoritma, berisi uraian singkat mengenai masalah yang akan diselesaikan}

- **DEKLARASI**

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

- **ALGORITMA:**

{semua langkah/aksi algoritma dituliskan disini}

DEKLARASI, VARIABEL dan TIPE DATA

- **Deklarasi** diperlukan bila kita akan menggunakan pengenal (identifier) dalam program. Identifier dapat berupa variable, konstanta dan fungsi.
- **Variabel** adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan.
- **Tipe Data** (Data Type) adalah jenis nilai yang dapat ditampung suatu variabel. Misalnya dapat menampung bilangan bulat, pecahan/berkoma, karakter huruf tunggal hingga simbol dan huruf yang membentuk barisan karakter.

Jenis – jenis tipe data (1)

- **Tipe Data Primitive (Sederhana)**

- Tipe data primitive adalah Tipe data dasar yang hanya dapat menyimpan satu jenis nilai pada satu buah variabel.
- Beberapa contoh tipe data primitive yang sering dijumpai adalah tipe numerik (integer, real/float), karakter (char), barisan karakter (string) dan boolean (True atau False).

Tipe Dasar	Ukuran Memori (byte)	Jangkauan Nilai	Jumlah Digit Presisi
Char	1	-128 hingga +127	-
Integer	2	-32768 hingga +32767	-
Long integer	4	-2.147.438.648 hingga 2.147.438.647	-
Float	4	3,4E-38 hingga 3,4E38	6-7
Double	8	1.7E-308 hingga 1.7E308	15-16
long double	10	3.4E-4932 hingga 1.1E4932	19

Jenis – jenis tipe data (2)

- **Tipe Data Composite**

- Composite adalah tipe data yang dapat menyimpan banyak nilai dengan jenis yang berbeda-beda.
- Contohnya: Array, Record/ struct, object, enumerasi,

Deklarasi Variabel dalam penulisan pseudocode

Bentuk umum:

Nama variabel : Tipe Data

Contoh :

x : int

nama : char

luas : float

Format Pseudocode Lengkap

Judul program/ algoritma

PROGRAM Euclidean

Program untuk mencari GCD dari dua buah bilangan bulat positif m dan n ($m \geq n$). GCD dari m dan n adalah bilangan bulat positif terbesar yang habis membagi m dan n

Deklarasi:

m, n : integer
 r : integer

{bil bulat}
{sisa hasil bagi}

Deklarasi variable

→ komentar

Algoritma

read(m, n) { $m \geq n$ }

while $n \neq 0$ do

$r \leftarrow m \text{ MOD } n$

$m \leftarrow n$

$n \leftarrow r$

end while

{kondisi selesai pengulangan $n=0$, maka $\text{gcd}(m, n) = m$ }

Algoritma

Perbandingan Antara Notasi Algoritmik Kalimat Deklaratif Dengan Pseudocode

Algoritma	Pseudocode
Nilai A dikali dengan 10	$A \leftarrow A + 10$
Cetak nilai A bila lebih besar 10	IF A > 10 THEN PRINT A
Dari dua bilangan A dan B, cari bilangan terbesar	IF A > B THEN PRINT A ELSE PRINT B

Contoh 1

(Konversi Fahrenheit - Celcius)

PROGRAM Konversi Fahrenheit ke Celcius

{algoritma ini akan menghitung nilai Celcius dari nilai Fahrenheit yang diinputkan}

DEKLARASI

f,c : float

ALGORITMA

read(f)

$c \leftarrow (F-32) * 5/9$

print(c)

Contoh 2

(Menghitung Luas Lingkaran)

PROGRAM Menghitung Luas Lingkaran

{algoritma ini akan menghitung luas lingkaran berdasarkan jari-jari yang diinputkan}

DEKLARASI

r, Luas : float

ALGORITMA

read(r)

Luas \leftarrow phi * r * r

print(Luas)

Contoh 3

(Konversi Koordinat Polar ke Koordinat Cartesian)

PROGRAM Konversi Koordinat

{algoritma ini akan mengkonversi koordinat polar menjadi koordinat cartesian}

DEKLARASI

radius, alpha, X, Y : integer

ALGORITMA

read(radius)
read(alpha)
 $X \leftarrow \text{Radius} * \text{Cos}(\text{Alpha})$
 $Y \leftarrow \text{Radius} * \text{Sin}(\text{Alpha})$
print(X)
print(Y)

Contoh 3 (Bilangan Terbesar)

PROGRAM Mendapatkan Bilangan Terbesar

{algoritma ini akan mencari dan mendapatkan bilangan terbesar dari dua bilangan yang diinputkan}

DEKLARASI

bilanganPertama, bilanganKedua : integer

ALGORITMA

```
read(bilanganPertama)
read(bilanganKedua)
if bilanganPertama > bilanganKedua
    print(bilanganPertama)
else
    print(bilanganKedua)
end if
```

UTS

- Buatlah resume tentang pengenalan algoritma
- Minimal 2 halaman A4
- Save file dengan nama file: [ALPRO] UTS dengan tipe .doc, .docx atau .pdf
- Kirim via email paling lambat hari Sabtu tanggal 27 Oktober 2018 Pukul 21.00
- Ke email hastie.audytra@gmail.com
- Subject email: [ALPRO] UTS