

# ALGORITMA DAN PEMROGRAMAN

[Komang Aryasa | [Pertemuan 02]

# Outline



Konsep Dasar Algoritma

Definisi Algoritma

Ciri Algoritma

Contoh Algoritma

Struktur Algoritma dan Pseudo-Code



# Konsep Algoritma

- Algoritma merupakan fondasi yang harus dikuasai oleh setiap mahasiswa yang ingin menyelesaikan suatu masalah secara berstruktur, efektif, dan efisien, teristimewa lagi bagi mahasiswa yang ingin menyusun program komputer untuk menyelesaikan suatu persoalan.



# ***Definisi*** Algoritma :

1. Teknik penyusunan langkah-langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas tetapi tersusun secara logis dan sistematis.
2. Suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya.
3. Susunan langkah yang pasti, bila diikuti maka akan mentransformasi data input menjadi informasi output.

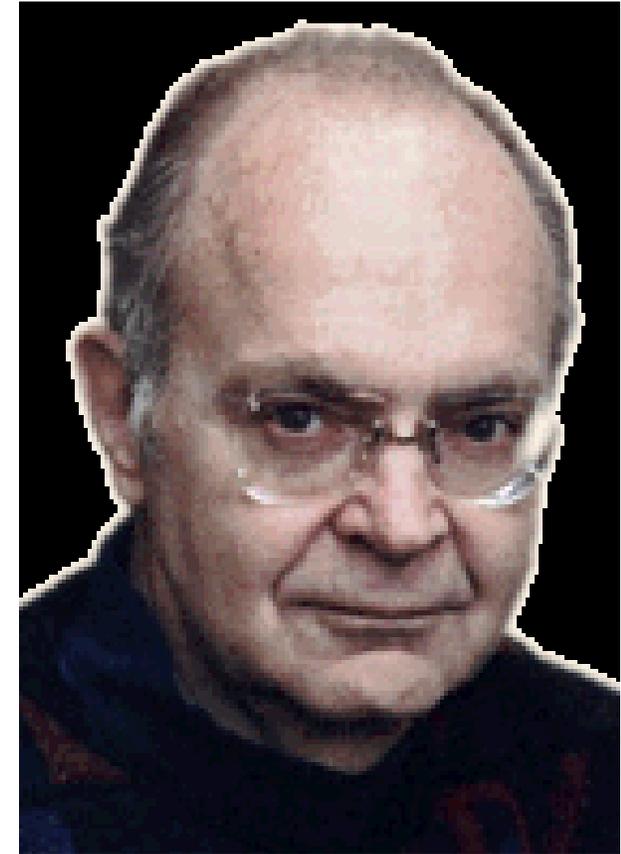
# Catatan Sejarah

- **Abu Ja'far Muhammad Ibnu Musa Al-Kwarizmi** : Matematisi dan Astronomer Persia, penulis buku "Aljabar wal muqabala" beberapa abad yang lalu (diperkirakan wafat pada tahun 850), dianggap sebagai pencetus pertama dari Algoritma, karena didalam buku tersebut Abu Ja'far menjelaskan langkah-langkah dalam menyelesaikan berbagai persoalan aritmetika (aljabar), Kemungkinan besar kata "Algoritma" diambil dari kata "Al-Kwarizmi" yang kemudian berubah menjadi "Algorism", selanjutnya menjadi "Algorithm".



# Ciri Algoritma

- **Donald E. Knuth** : seorang penulis beberapa buku algoritma abad XX, menyatakan bahwa ada beberapa ciri algoritma, yaitu:
  1. Algoritma mempunyai awal dan akhir, suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas atau dengan kata lain suatu algoritma memiliki langkah yang terbatas.
  2. Setiap langkah harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda, tidak membingungkan (not ambiguous).
  3. Memiliki masukan (input) atau kondisi awal.
  4. Memiliki keluaran (output) atau kondisi akhir.
  5. Algoritma harus efektif, bila diikuti benar-benar akan menyelesaikan persoalan.



# Algoritma bisa ditemukan dalam kehidupan sehari-hari

Proses	Algoritma	Contoh langkah
1. Membuat Kue	Resep Kue	Campurkan 200 gram gula, 400 gram terigu dan 5 butir telur, kemudian diaduk hingga menjadi adonan.
2. Menggunakan pesawat Televisi	Buku Petunjuk	Pastikan kabel power sudah disambung ke sumber listrik, kemudian tekan tombol "power" hingga lampu indikator merah menyala.
3. Praktikum Kimia	Petunjuk Praktikum	Campurkan 15 ml Asam Nitrat ke dalam 15 ml Natrium hidroksida.

# Algoritma makan siang di kedai cepat saji:

1. Cari lokasi terdekat dari kedai siap saji tersebut.
2. Bergerak menuju lokasi tersebut.
3. Setelah sampai, melihat dimana tempat memesan makanan.
4. Antri ditempat antrian pemesanan.
5. Melihat daftar menu, dan memilih makanan.
6. Melakukan pesanan sesuai selera.
7. Membayar harga pesanan.
8. Menunggu pesanan.
9. Menunggu pesanan.
10. Menerima pesanan.
11. Mencari meja kosong.
12. Meletakkan makanan diatas meja, dan duduk
13. Menikmati makanan.
14. Setelah selesai, meninggalkan kedai tersebut.

# Sifat Algoritma

- **input** : suatu algoritma memiliki input atau kondisi awal sebelum algoritma dilaksanakan dan bisa berupa nilai-nilai peubah yang diambil dari himpunan khusus.
- **output** : suatu algoritma akan menghasilkan output setelah dilaksanakan, atau algoritma akan mengubah kondisi awal menjadi kondisi akhir, dimana nilai output diperoleh dari nilai input yang telah diproses melalui algoritma.
- **definiteness** : langkah-langkah yang dituliskan dalam algoritma ter-definisi dengan jelas sehingga mudah dilaksanakan oleh pengguna algoritma



# Sifat Algoritma

- **finiteness** : suatu algoritma harus memberi kondisi akhir atau output setelah melakukan sejumlah langkah yang terbatas jumlahnya untuk setiap kondisi awal atau input yang diberikan
- **effectiveness** : setiap langkah dalam algoritma bisa dilaksanakan dalam suatu selang waktu tertentu sehingga pada akhirnya memberi solusi sesuai dengan yang diharapkan.
- **generality** : langkah-langkah algoritma berlaku untuk setiap himpunan input yang sesuai dengan persoalan yang akan diberikan, tidak hanya untuk himpunan tertentu.



# Contoh

Susun algoritma untuk mencari angka terbesar (maksimum) dari suatu kumpulan bilangan bulat yang terbatas banyaknya.

# Algoritma

1. Ambil satu angka pertama dan anggaplah angka pertama dalam kumpulan ini sebagai yang terbesar (maksimum).
2. Bandingkan angka maksimum ini dengan angka berikutnya dalam kumpulan, bila angka berikut tersebut lebih besar maka jadikanlah ia maksimum.
3. Ulangi langkah 2 ini sehingga tidak ada lagi angka yang tersisa dalam himpunan
4. Hentikan perbandingan setelah semua angka selesai dibandingkan, maka angka terbesar dalam himpunan tersebut adalah angka maksimum terakhir.



# Lebih rinci:

1. andaikan  $N =$  banyaknya angka dalam himpunan
2.  $index = 1$
3.  $maksimum = angka(index)$
4. selama  $index < N$  , lakukan:
  - i. bila  $angka(index) > maksimum$  maka  
 $maksimum = angka(index)$
  - ii.  $index = index + 1$
  - iii. ulangi lagi untuk index berikutnya
5. angka terbesar dalam himpunan adalah maksimum terakhir



# Pseudo-code

Bentuk algoritma yang menyerupai instruksi program komputer disebut pseudo-code



# Contoh Pseudo-code

1.  $N = \text{input}(\text{'Banyaknya angka : '})$
2.  $\text{index} = 1$
3.  $\text{Max} = \text{angka}(\text{index})$
4. While  $\text{index} < N$  do:
  1. If  $\text{angka}(\text{index}) > \text{max}$  then  $\text{max} = \text{angka}(\text{index})$
  2.  $\text{index} = \text{index} + 1$
5. Print ( $\text{"terbesar = "}, \text{max}$ )

# Struktur Algoritma / pseudo-code

- **Bagian Kepala (Header)** : memuat nama algoritma serta informasi atau keterangan tentang algoritma yang ditulis.
- **Bagian Deklarasi / Definisi Variabel** : memuat defenisi tentang nama variable, nama tetapan, nama prosedur, nama fungsi, tipe data yang akan digunakan dalam algoritma.
- **Bagian Deskripsi / Rincian Langkah** : memuat langkah-langkah penyelesaian masalah, termasuk beberapa perintah seperti baca data, tampilkan, ulangi, yang mengubah data input menjadi output, dsb.



# Contoh

- Susun algoritma / pseudo-code yang menghitung luas sebuah lingkaran, dimana jejari lingkaran dimasukkan oleh user



# Contoh

## **Algoritma** Luas\_lingkaran

{ menghitung luas sebuah lingkaran apabila jari-jari lingkaran tersebut diberikan }

## **Deklarasi**

{ Definisi nama tetapan }

**const** phi = 3.14;

{ definisi nama peubah / variable }

**real** jari\_jari, luas;

## **Deskripsi**

**read**(jari\_jari);

luas = phi \* jari\_jari \* jari\_jari;

**write**(luas);



# Contoh

- Susun algoritma untuk menghitung rata-rata sejumlah angka yang dimasukkan lewat keyboard oleh user.



## Algoritma Nilai\_Rata

{menghitung nilai rata sejumlah bilangan yang dimasukkan lewat keyboard }

### Definisi Variabel

**integer** x, N, k, jumlah;

**real** nilai\_rata;



## Rincian Langkah

{ masukkan jumlah data }

**read(N);**

$k \leftarrow 1;$

jumlah  $\leftarrow 0;$

**while** ( $k \leq N$ ) **do**

    { baca data }

**read(x);**

    jumlah  $\leftarrow$  jumlah + x;

$k \leftarrow k + 1;$

**endwhile**

    { hitung nilai rata }

    nilai\_rata  $\leftarrow$  jumlah / N;

**write**(nilai\_rata);



# Struktur Bahasa C++

Bagian Header:

1. Komentar tentang program:

```
//Program ini untuk ....
```

2. Pre-processor Directive:

```
#include <....>
```

```
using namespace std;
```



# Struktur C++

- Bagian Deklarasi:

1. Deklarasi fungsi : `int nama-fungsi();`

2. Deklarasi variabel : `int x, y, z;`  
`float a, b, c;`

- Bagian Deskripsi:

```
void main() {  
    ... instruksi ....  
    return;  
}
```

# Contoh

## Algoritma Luas\_lingkaran

{ menghitung luas sebuah lingkaran apabila jari-jari lingkaran tersebut diberikan }

## Deklarasi

{ Definisi nama tetapan }

**const** phi = 3.14;

{ definisi nama peubah / variable }

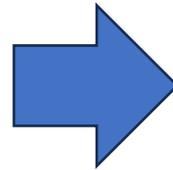
**real** jari\_jari, luas;

## Deskripsi

**read**(jari\_jari);

luas = phi \* jari\_jari \* jari\_jari;

**write**(luas);



```
//Program lingkaran  
#include <iostream>  
Using namespace std;
```

```
// definisi nama tetapan  
#define phi 3.14  
// definisi nama peubah / variabel  
float jari_jari, luas;
```

```
void main() {  
    cin >> jari_jari;  
    luas = phi * jari_jari * jari_jari;  
    cout >> luas;  
    return;  
}
```



# Next Vidio Berikutnya