



IPB University
— Bogor Indonesia —

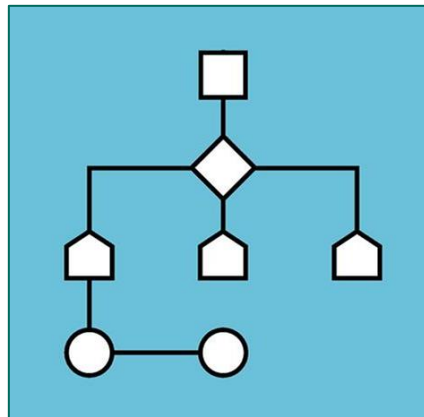
KOM120B #1

Pengantar ke Algoritme

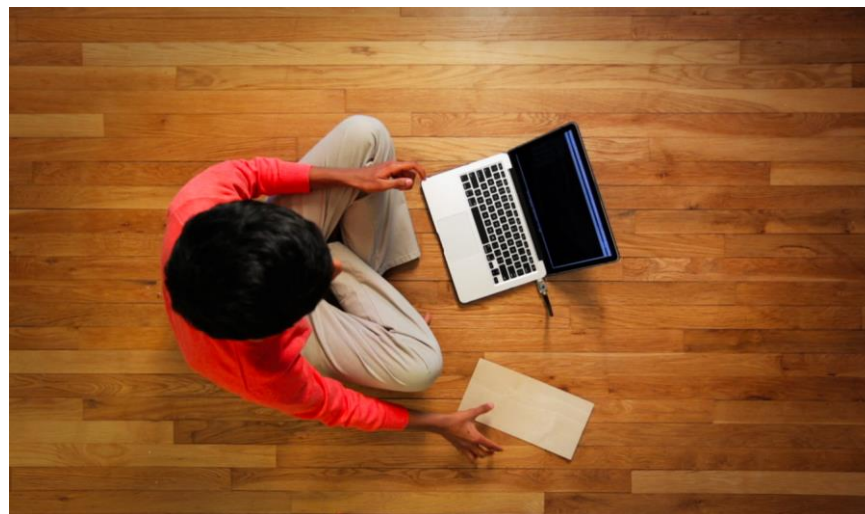
Tim Pengajar

KOM120B – Algoritme dan Dasar Pemrograman

Departemen Ilmu Komputer - FMIPA



```
7 string sInput;  
8 int iLength, iN;  
9 double dblTemp;  
10 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     iLength = sInput.length();  
29     if (iLength < 4) {  
30         again = true;  
31         continue;  
32     } else if (sInput[iLength - 3] != '.') {  
33         again = true;  
34         continue;  
35     } while (++iN < iLength) {  
36         if (isdigit(sInput[iN])) {  
37             continue;  
38         } else if (iN == (iLength - 3)) {  
39             continue;  
40         }
```



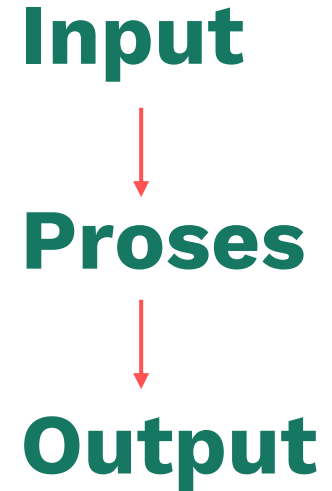
Ingat ini ?

Bagaimana Cara Memasak Mie Instant?



Kasus Memasak Mie Instant

- Tiap orang memiliki cara memasak yang berbeda (proses).
- Yang sama adalah material (input) berupa sebungkus mie instant yang masih mentah dan hasil akhirnya (output) berupa mie rebus, bukan nasi goreng, gado-gado, tiwul, atau makanan beda lainnya.
- Jika semua tahapan ditulis secara terstruktur, itulah yang disebut sebagai **algoritme**.



Ingat Kasus Ini?

Bagaimana cara mengurutkan
bilangan ini dari terkecil ke
terbesar?

10, 99, 34, 45, 13 , 2, 9, 22

Tuliskan perintah Anda !

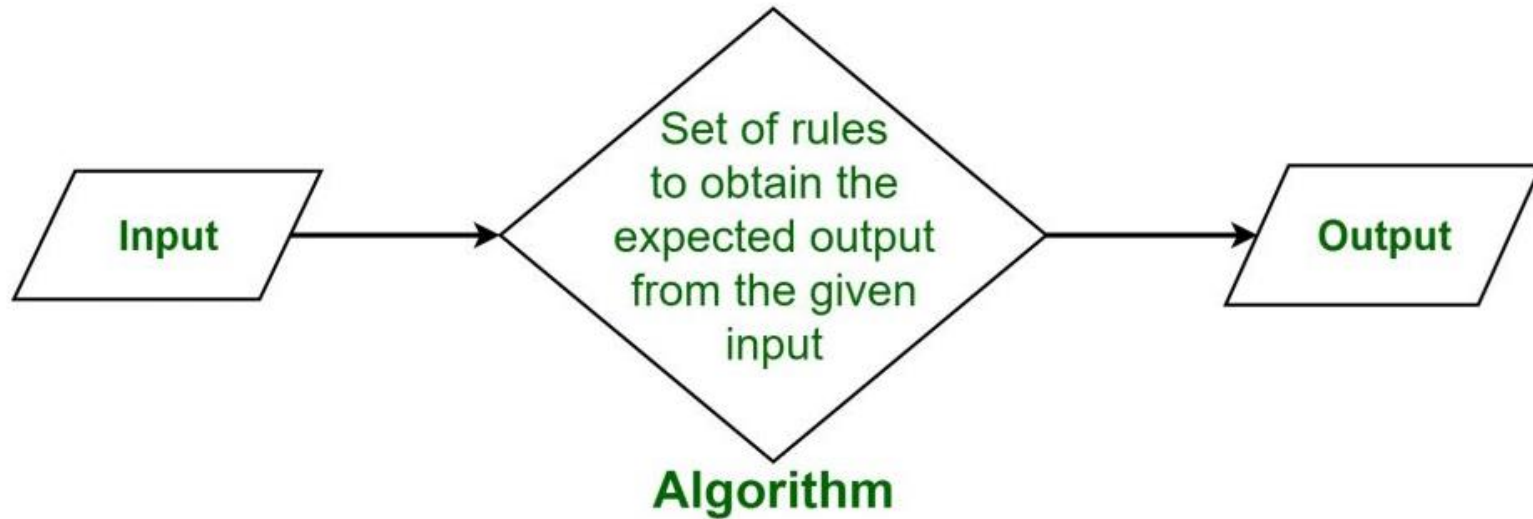


Apa inputnya?

Apa outputnya?

Bagaimana prosesnya?

Algoritme



Algoritme

Komponen algoritme

- Ada nol atau lebih input (masukan)
- Ada proses, merupakan langkah-langkah yang harus dilakukan untuk dapat mencapai tujuan akhir.
- Ada output atau tujuan akhir (bersifat unik)

Kriteria algoritme

- **Finitness** (terbatas), algoritme harus berhenti dan selesai pada output
- **Definiteness** (pasti), mempunyai arah dan tujuan yang jelas
- **Efisien**, menggunakan sumberdaya optimal dan waktu yang paling singkat

Contoh Kasus

*Diketahui dimensi suatu kotak, yaitu panjang, lebar, dan tinggi.
Tentukan volume kotak.*

- Ide: $\text{volume kotak} = \text{panjang} \times \text{lebar} \times \text{tinggi}$.
- Dekomposisi:
 - Input : panjang, lebar, tinggi
 - Proses : kalikan panjang, lebar, dan tinggi
 - Output : hasil kali (volume)
- Algoritme:
 - Dapatkan input (panjang, lebar, tinggi)
 - Kalikan panjang, lebar, dan tinggi kemudian simpan hasilnya
 - Tampilkan hasil perkalian tersebut (volume)

Cara Penulisan Algoritme

Narative

- Menggunakan bahasa sendiri (Indonesia, English, etc)
- Tidak ada ketentuan baku tentang penulisan kalimat. Bisa berbentuk narasi lengkap, atau singkat dan bahkan berbentuk mirip kode program yang disebut sebagai **pseudocode**.

Flowchart

- Dalam bentuk diagram alir (simbol-simbol)
- Memiliki kesepakatan penulisan → konvensi

Pseudocode

- **Pseudocode** adalah cara informal untuk menuliskan algoritme atau rancangan program komputer
- Bertujuan untuk mendapatkan ide secara cepat dan juga memudahkan membaca algoritme tanpa aturan *grammar* yang ketat.
- Ada banyak cara menuliskan pseudocode, tidak ada aturan yang baku. Namun untuk mengurangi *ambiguity*, maka dibuat **konvensi**.

Contoh Pseudocode

Pseudocode untuk melakukan penjumlahan:

Input: `bilangan[1...n]`

`jumlah := 0`

`for indeks = 1 to n`

`jumlah = jumlah + bilangan[indeks]`

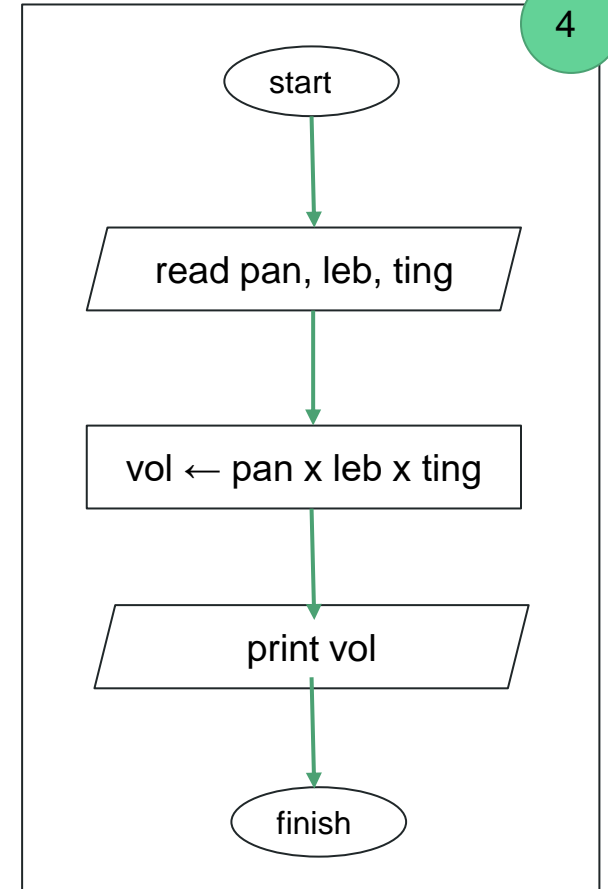
Output: `jumlah`

Bandingkan 4 Model Algoritme

1
Dapatkan input (panjang, lebar, tinggi)
Kalikan panjang, lebar, dan tinggi kemudian
simpan hasilnya
Tampilkan hasil perkalian tersebut (volume)

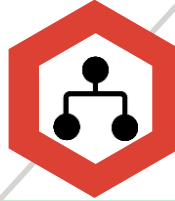
2
baca panjang, lebar, dan tinggi
 $\text{volume} = \text{panjang} \times \text{lebar} \times \text{tinggi}$
tuliskan volume

3
`read(panjang, lebar, tinggi)`
 $\text{volume} = \text{panjang} \times \text{lebar} \times \text{tinggi}$
`print(volume)`



Struktur Algoritme

Conditional
Pemilihan langkah didasarkan oleh suatu kondisi: **benar** (*true*) atau salah (*false*)



Loop (Pengulangan)

Suatu langkah dilakukan beberapa kali berdasarkan suatu kondisi.



Sequential

Setiap langkah diproses secara berurutan (dari atas ke bawah)



STRUKTUR KENDALI

- Struktur kendali adalah suatu bentuk atau struktur yang memiliki peranan khusus untuk mengatur atau mengendalikan urutan pengerjaan satu atau lebih instruksi.
- Terdiri atas 3 kelompok besar, yaitu:
 - struktur kondisi (*conditional*) → IF
 - struktur perulangan (*looping*) → WHILE, FOR
 - struktur lainnya → EXIT, CONTINUE

STRUKTUR KONDISI

1. Jika hari hujan, lakukan:
 - a. bawa payung
 - b. gunakan mobil
2. Selainnya, gunakan sepeda.

```
read(nilai)
if (nilai<60)
{
    print("tidak lulus")
    print("wajib mengulang")
}
else print("lulus")
```

STRUKTUR PERULANGAN

```
c=0
while (c<3)
{
    c=c+1
    print("hello")
}
```

```
for c=0 to 3
{
    print("hello")
}
```

EXIT dan CONTINUE

```
read(a,b)
if (a<b) swap(a,b)
while (true)
{
    r = a % b
    if (r==0) exit
    a=b
    b=r
}
print(b)
```

```
a = 10
while (a < 15)
{
    a = a + 1
    if (a<=13) continue
    print(a)
}
```


Latihan #1

Perhatikan algoritme berikut

- **Input** : Banyaknya coklat, misalnya C.
- **Output** : Nomor bebek pertama yang tidak mendapat coklat, misalnya b.
- **Perhitungan:**
 1. Hitung akar dari $2C$, ambil bilangan bulatnya, simpan sebagai n
 2. Hitung $n(n+1)$, simpan sebagai a
 3. Jika $(a \leq 2C)$, maka $b=(n+1)$. Selainnya, $b=n$.

Latihan #1

```
read(C);  
n=(int) (sqrt(2.0*C));  
a=n*(n+1);  
if (a<=2*C) b=n+1;  
else b=n;  
print(b);
```

```
#include <stdio.h>  
int main()  
{  
    int C, n, a;  
    scanf("%d", &C);  
    n=(int) (sqrt(2.0*C));  
    a=n*(n+1);  
    if (a<=2*C) b=n+1;  
    else b=n;  
    printf("%d\n", b);  
    return 0;  
}
```

Latihan #2

Tuliskan algoritme berikut

- **Input** : Tiga bilangan bulat.
- **Output** : Bilangan terbesar dari ketiganya.
- **Contoh Input** : 5 10 2
- **Contoh Output** : 10

Latihan #2

```
read(a,b,c);  
  
if (a>b && a>c)  
print(a);  
else if (b>c)  
print(b);  
else print(c);
```

```
#include <stdio.h>  
int main()  
{  
    int a,b,c;  
    scanf("%d %d %d",&a,&b,&c);  
    if (a>b && a>c) printf("%d\n", a);  
    else if (b>c) printf("%d\n", b);  
    else printf("%d\n", c);  
    return 0;  
}
```

Latihan

Kerjakan setiap soal berikut sebagai latihan mandiri dan bahan diskusi di kuliah selanjutnya.

Buatlah algoritme untuk:

1. mendapatkan nilai jawaban dari suatu persamaan kuadrat

$$ax^2 + bx + c = 0.$$

2. mendapatkan bilangan terbesar dari tiga buah bilangan
3. mendapatkan digit terakhir dari suatu bilangan bulat positif
4. menghitung jumlah dari $1 + 2 + \dots + n$