

Analisis Kasus - Python

Bahan Kuliah SD2104 Pemrograman Lanjut

Sevi Nurafni

Fakultas Sains dan Teknologi

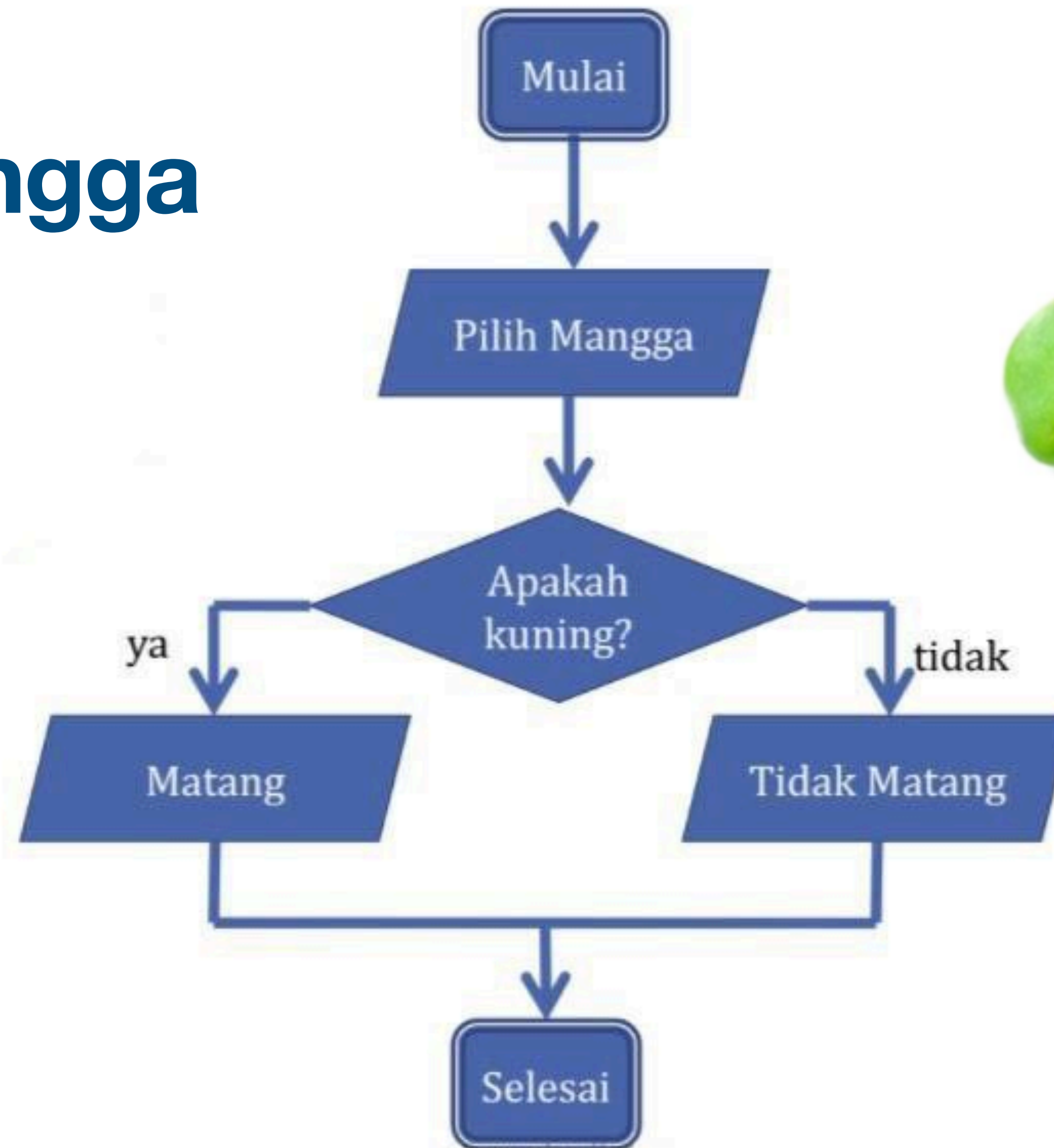
Universitas Koperasi Indonesia 2024

Contoh-1: Memilih Mangga

- Analisis kasus dapat digunakan dalam kehidupan sehari-hari,
Contoh: memilih mangga
- Mangga yang sudah matang dan siap dimakan adalah mangga yang berwarna kuning
- Jika tidak berwarna kuning maka tidak matang



Flowchart Memilih Mangga



Pseudocode - Memilih Mangga

PillihMangga

If (ApakahKuning? = true) **then**

Matang

Else {ApakahKuning? = false}

Tidak Matang

Analisis Kasus



- Memungkinkan kita membuat teks yang sama, namun menghasilkan eksekusi berbeda
- Sering disebut **percabangan / kondisional**
 - Dari satu langkah ada pilihan (bercabang) ke beberapa langkah
- Terdiri atas:
 - Kondisi: ekspresi yang menghasilkan true dan false
 - Aksi: statement yang dilaksanakan jika kondisi yang berpasangan dengan aksi dipenuhi

Analisis Kasus



- Analisis kasus harus memenuhi 2 kriteria:
 - **COMPLETE**: semua kasus terdefinisi secara lengkap
 - **DISJOINT**: tidak ada kasus yang tumpang tindih/overlapped
- Contoh: Diberikan sebuah bilangan bulat, misalnya A, nyatakan apakah bilangan tersebut adalah bilangan positif, negatif, atau nol
- Ada 3 kasus yang *complete* dan *disjoint*:
 - $A > 0$
 - $A < 0$
 - $A = 0$
 - Tidak ada kasus lain yang bisa didefinisikan dan ketiga kasus tersebut tidak tumpang tindih

Sintaks Umum

Pseudocode

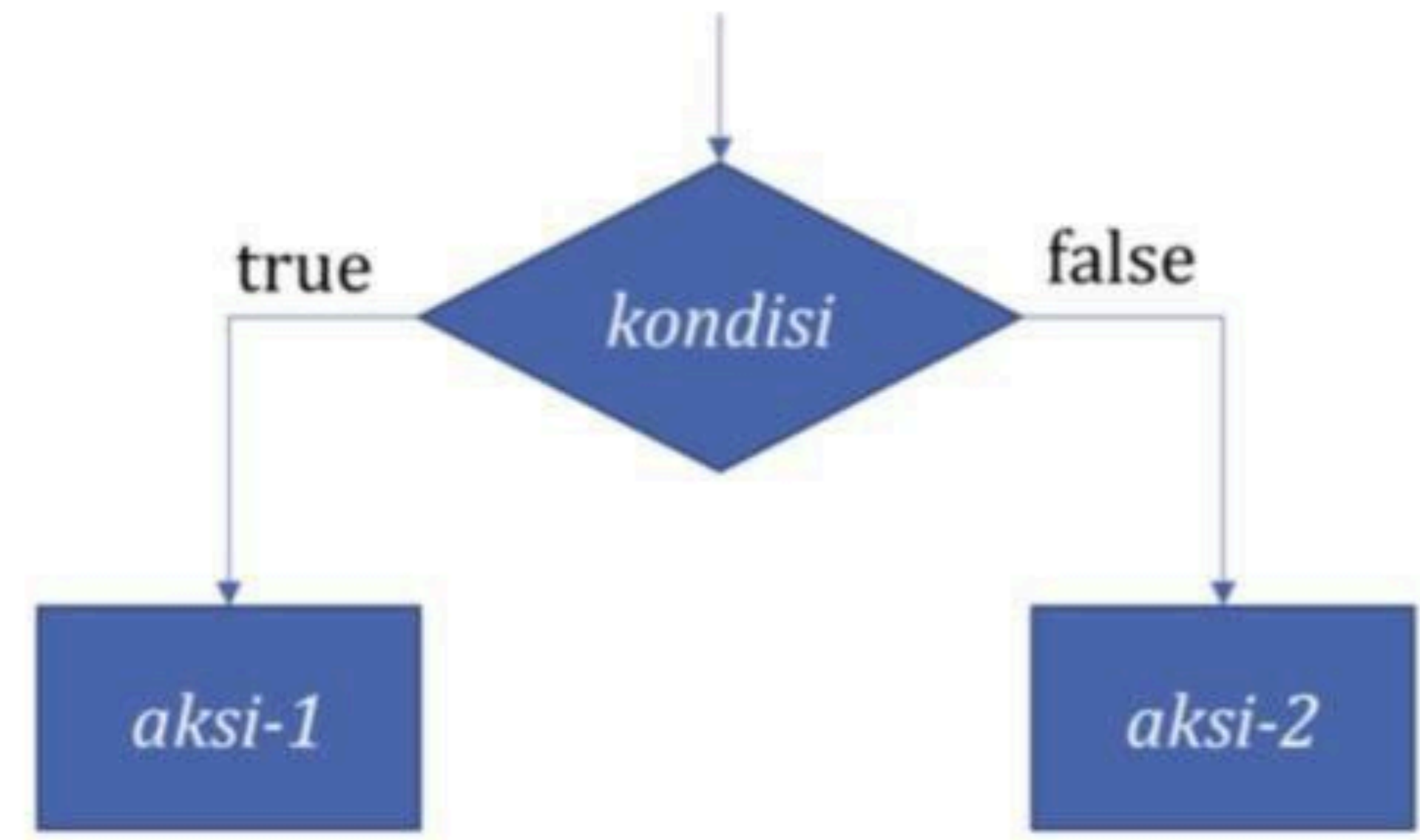
```
if ( kondisi ) then  
    aksi-1  
else { kondisi=false }  
    aksi-2
```

Python

```
if ( kondisi ):  
    aksi-1  
else: # kondisi = false  
    aksi-2
```

Jika aksi-1 atau aksi-2 terdiri dari lebih dari 1 instruksi, perhatikan bahwa indentasi harus rapi

flowchart



Jenis Analisis Kasus

Satu Kasus

```
if ( kondisi ):
    aksi-1

# jika kondisi=false
# tidak didefinisikan aksi
```

Dua Kasus [Komplementer]

```
if ( kondisi ):
    aksi-1
else: # kondisi=false
    aksi-2
```

Banyak Kasus

```
if (kondisi-1):
    aksi-1
elif (kondisi-2):
    aksi-2
elif (...):
    # kondisi-3 ... dst
    ...
else: # kondisi-n
    aksi-n
```


Contoh: Positif, Negatif, atau Nol?

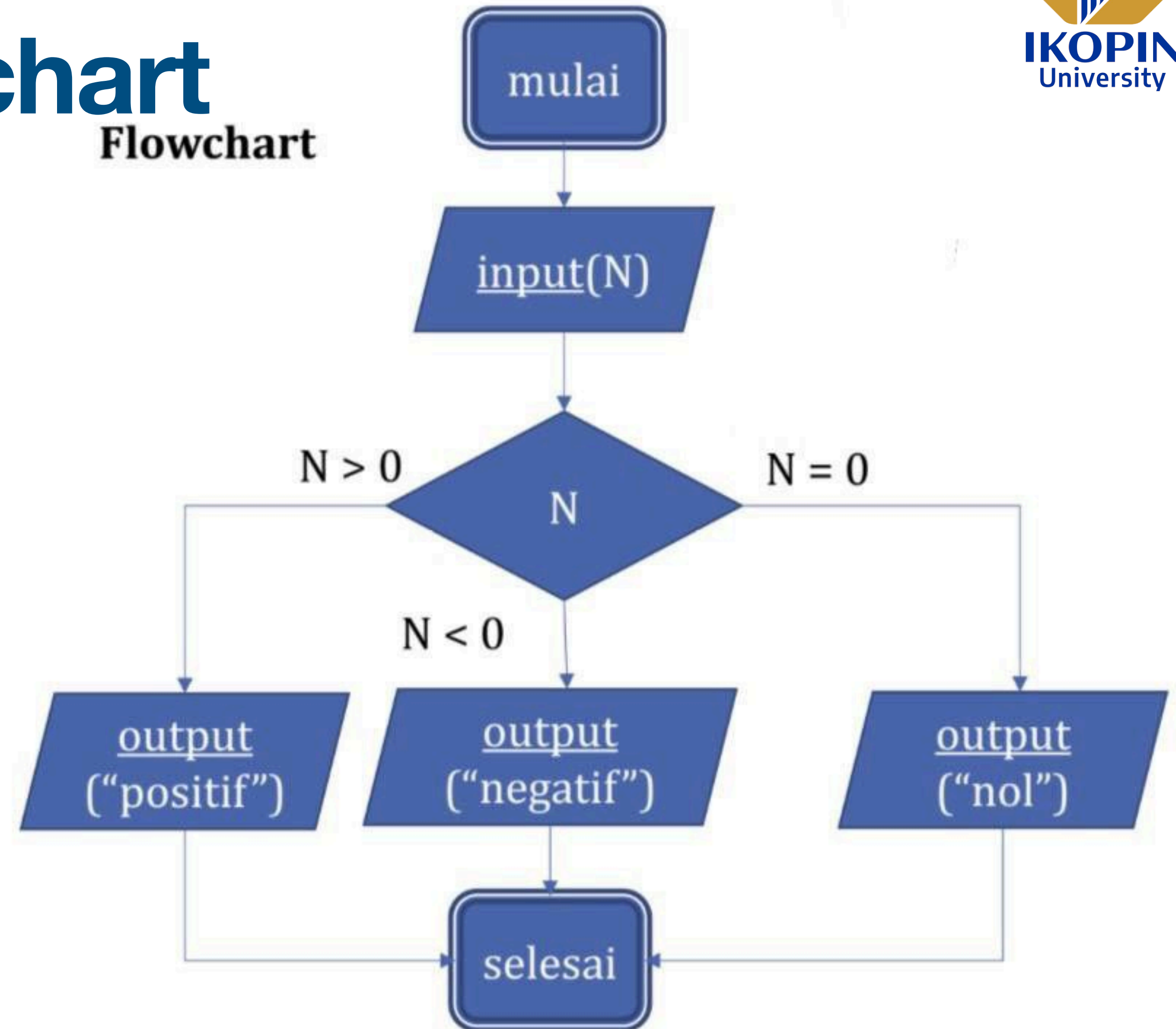
- Buatlah program yang menerima masukan sebuah integer, misalnya N, dan menentukan apakah N adalah bilangan bulat positif, negatif, atau nol
- Kasus:
 - Jika $N > 0$; cetak “positif”
 - Jika $N < 0$, cetak “negatif”
 - Jika $N = 0$; cetak “nol”

Contoh: Pseudocode + Flowchart

Pseudocode

```
input(N)  
if (N > 0) then  
    output("genap")  
else if (N < 0) then  
    output("negatif")  
else { N = 0 }  
    output("nol")
```

Flowchart



Contoh: Python

```
# Program Bilangan
# Input N. Tentukan apakah N positif, negatif, atau nol.

# KAMUS
# N : float

# ALGORITMA
N = int(input())

if (N > 0):
    print("positif")
elif (N < 0):
    print("negatif")
else: # N = 0
    print("nol")
```

Latihan-1: Maksimum 2 bilangan



- Buatlah sebuah program yang membaca masukan 2 buah bilangan bulat, misalnya A dan B, dan tuliskan di antara kedua bilangan tersebut mana yang paling besar
- Kasus:
 - Jika $A > B$, maka bilangan terbesar = A
 - Jika $A < B$, maka bilangan terbesar = B
 - Jika $A = B$, maka bilangan terbesar adalah A atau B (berarti output akan sama seperti salah satu dari 2 kasus di atas)

Latihan-2: Total Hambatan Seri



- Buatlah program yang menerima 3 buah hambatan (R1, R2, R3) dan menghasilkan hambatan total (RT) jika dirangkai seri.

$$\mathbf{RT = R1 + R2 + R3}$$

- R1, R2, dan R3 tidak boleh bernilai negatif. Jika satu saja hambatan bernilai negatif, maka total hambatan tidak bisa dihitung dan tuliskan ke layar pesan kesalahan “Hambatan total tidak bisa dihitung”.

**SELAMAT
BELAJAR**