

# Analisis Kasus - Python

Tim Penyusun Materi Pengenalan Komputasi  
Institut Teknologi Bandung © 2019





# Tujuan

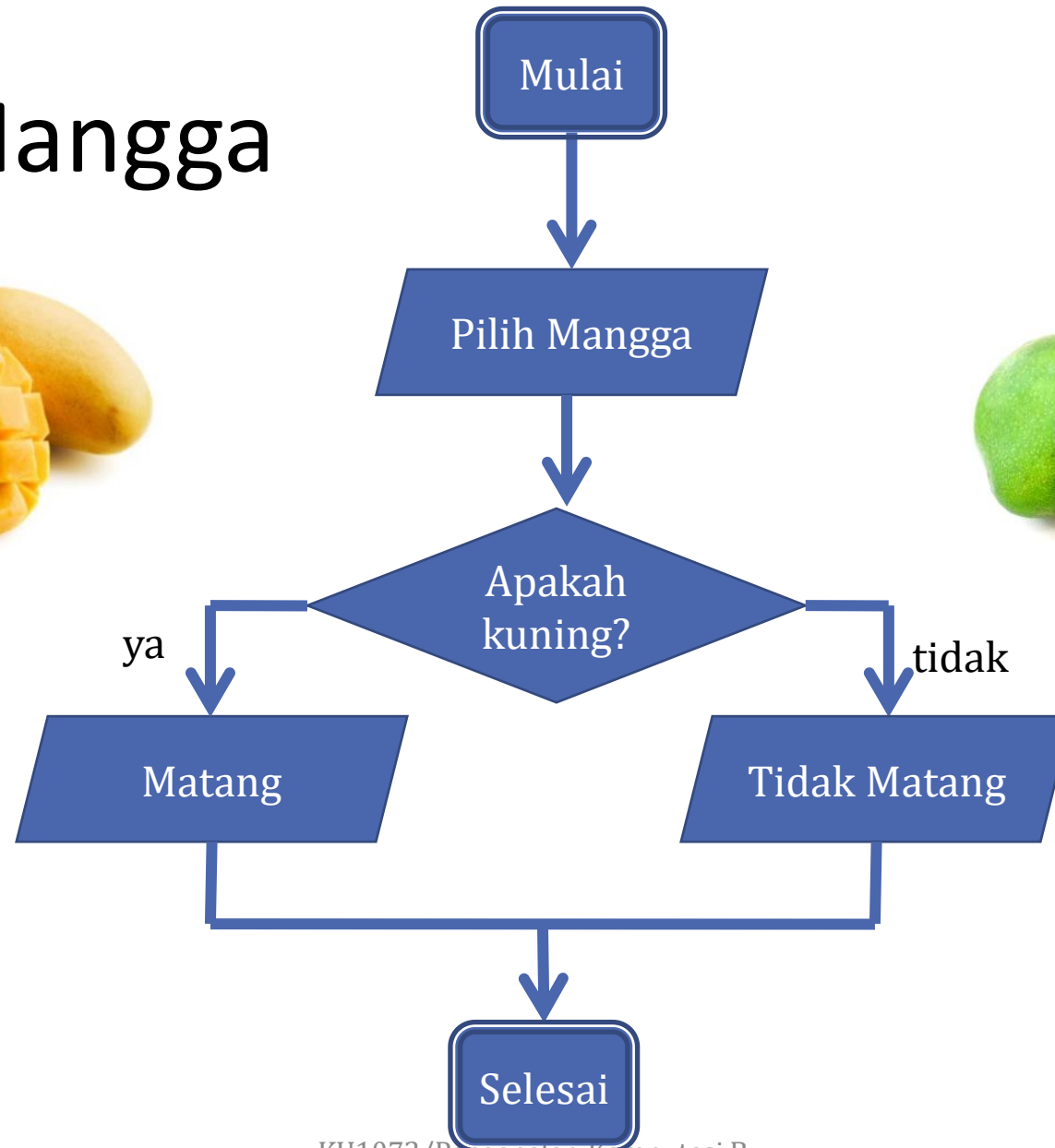
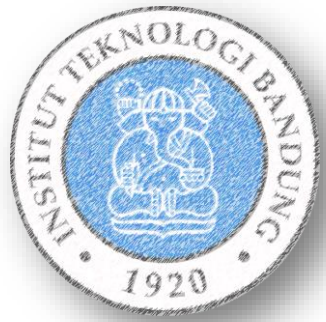
- Mahasiswa dapat menjelaskan pengertian dan jenis-jenis analisis kasus
- Mahasiswa dapat menggunakan notasi analisis kasus dengan benar
- Mahasiswa dapat memanfaatkan jenis-jenis analisis kasus dalam menyelesaikan persoalan sederhana yang diberikan

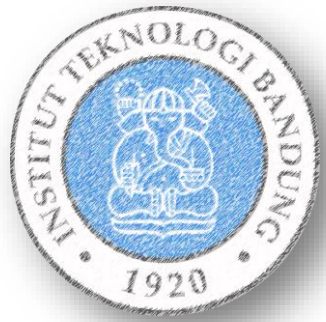
# Contoh-1: Memilih Mangga

- 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



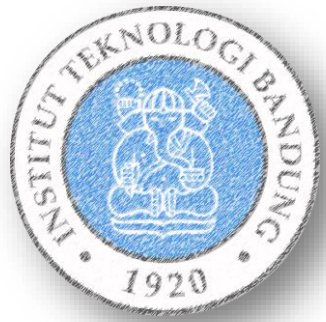
# Flowchart Memilih Mangga





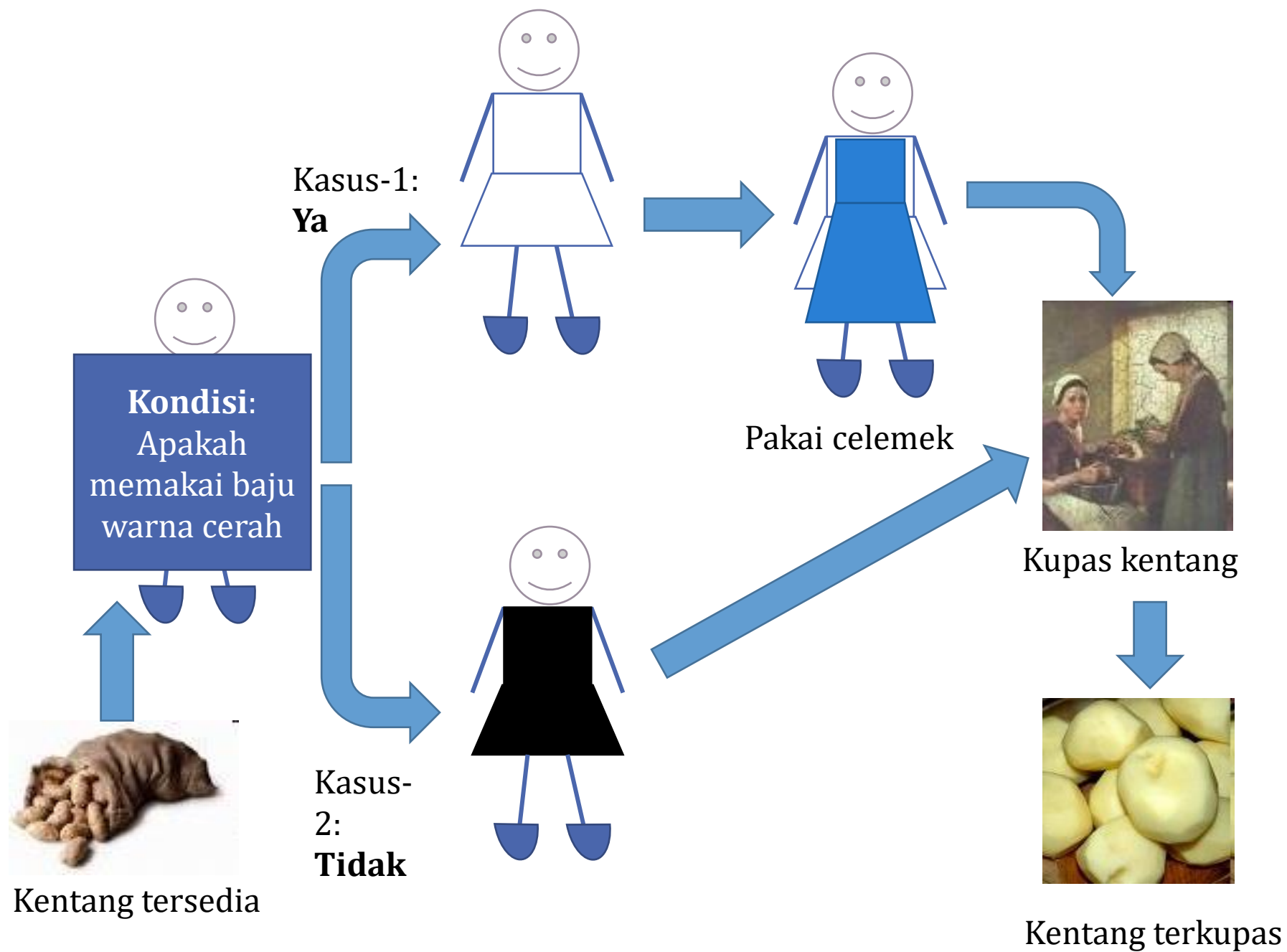
# Memilih Mangga - Pseudocode

```
PilihMangga  
if (ApakahKuning? = true) then  
    Matang  
else { ApakahKuning? = false}  
    Tidak Matang
```

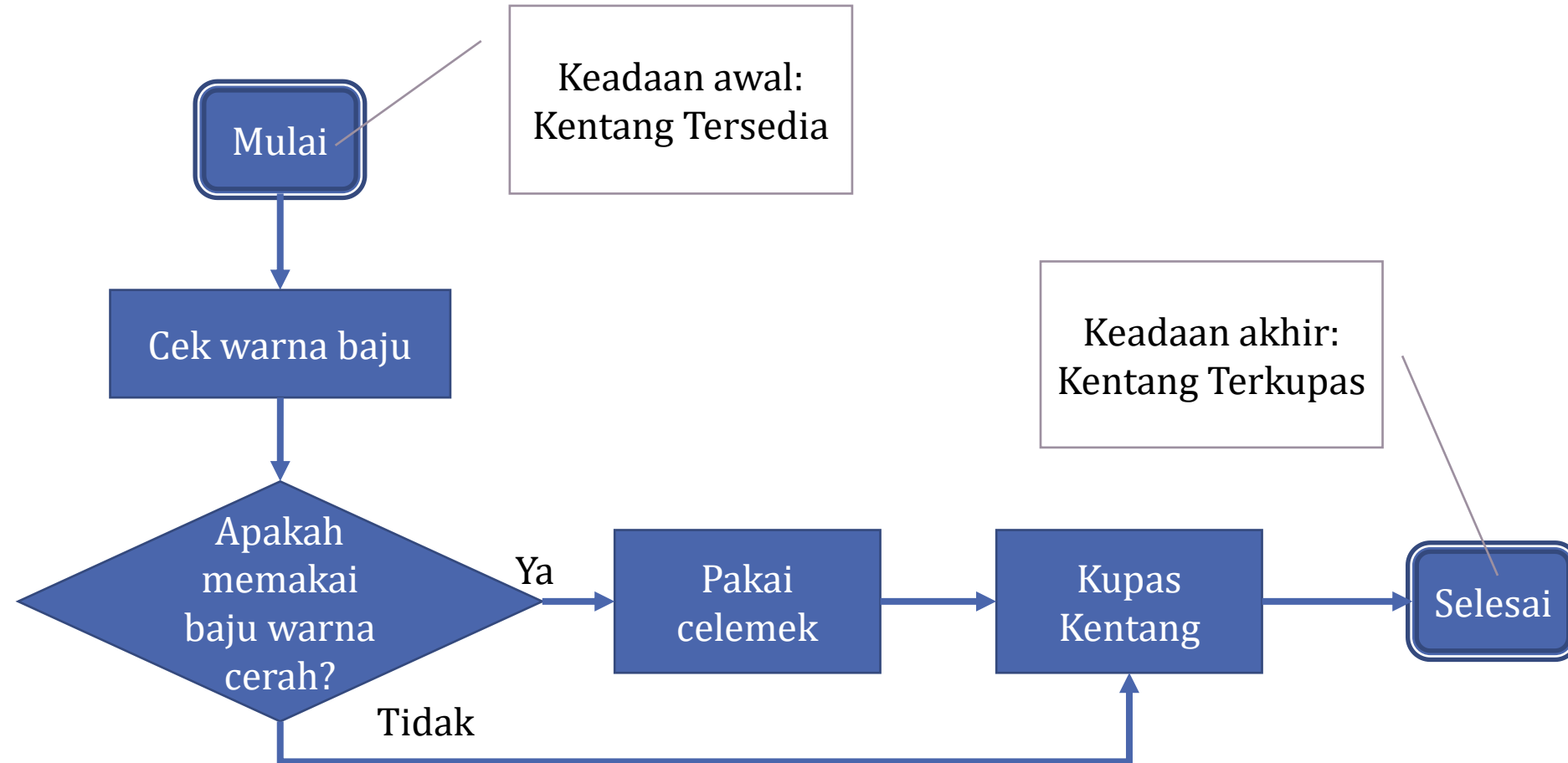


## Contoh-2: Menyiapkan kentang untuk makan malam

- Berdasarkan pengamatan, ada hari-hari di mana ibu memakai celemek ketika mengupas kentang, tapi ada hari-hari lain yang tidak
  - Setelah diamati, ternyata jika ibu sedang memakai baju berwarna cerah, maka ibu memakai celemek → takut bajunya terlihat kotor 😊
  - Jika tidak (memakai baju berwarna gelap), maka ibu tidak memakai celemek

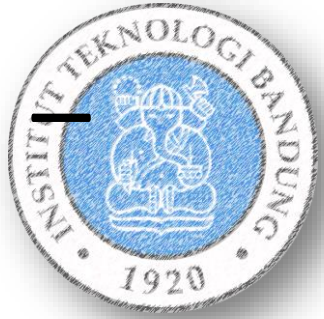


# Flowchart: Menyiapkan kentang untuk makan malam





# Menyiapkan kentang untuk makan malam – Pseudocode



**CekWarnaBaju**

if (ApakahBajuWarnaCerah? = ya) then

**PakaiCelemek**

{ else : ApakahBajuWarnaCerah? = tidak, tidak melakukan apa-apa }

**KupasKentang**



# Analisis Kasus (1)

- 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 (2)

- 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 ddefiniskan dan ketiga kasus tersebut tidak tumpang tindih



# Sintaks Umum

## Python

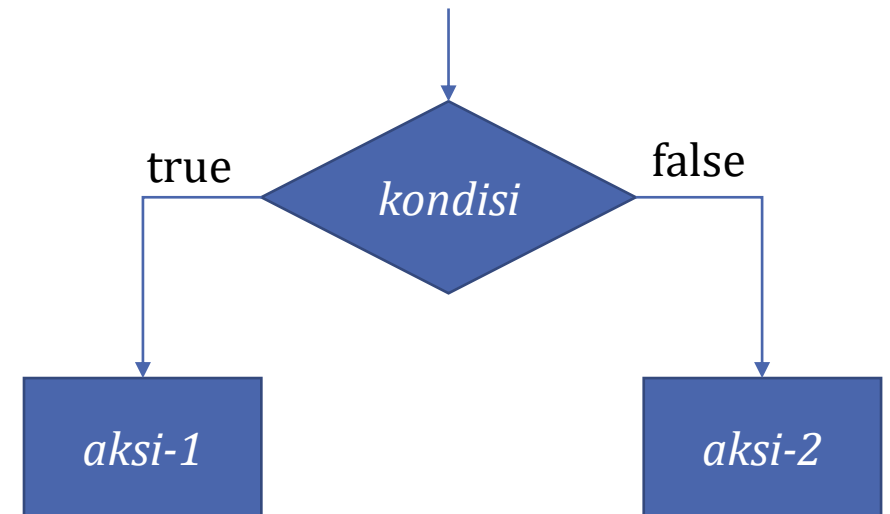
```
if ( kondisi ):  
    aksi-1  
else: # kondisi = false  
    aksi-2
```

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

## Pseudocode

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

## flowchart



# Jenis Analisis Kasus (dalam Python)

## Satu Kasus

```
if ( kondisi ):
    aksi-1

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

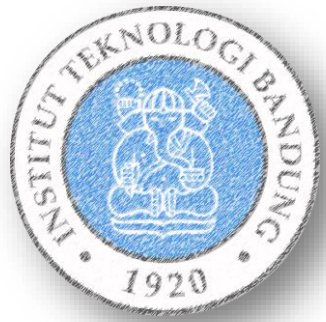
## Dua Kasus [Komplementer]

```
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
```

Pseudocode dan flowchart silakan disesuaikan atau lihat contoh-contoh berikut



# Contoh-3: Apakah bilangan positif [Contoh Satu Kasus]

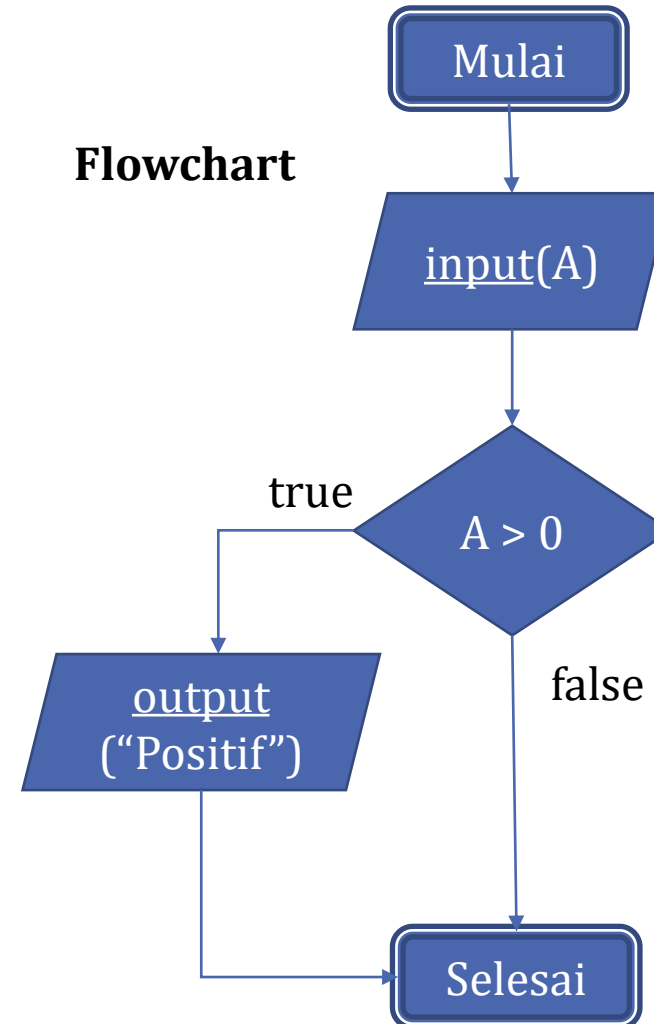
- Diberikan sebuah bilangan bulat, misalnya A, nyatakan apakah bilangan tersebut adalah bilangan positif atau bukan
- **Kondisi: Apakah  $A > 0$ ?**
- **Kasus:**
  - Jika **ya**, maka: tuliskan “Positif”
  - Jika **tidak**, tidak dilakukan apa pun

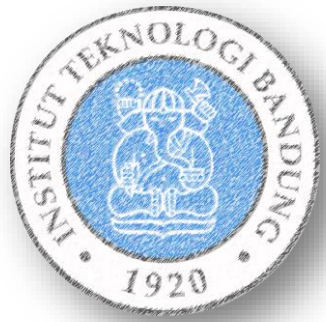
# Contoh-3: Pseudocode + Flowchart

## Pseudocode

```
input(A)  
if (A > 0) then  
    output("Positif")  
{ else: tidak dilakukan apa pun }
```

## Flowchart





# Contoh-3: Python

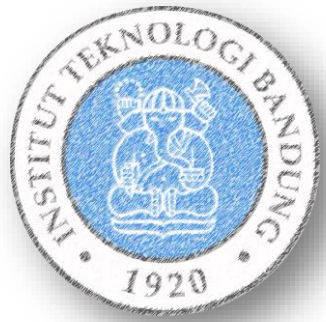
```
# Program CetakPositif
# Input A; jika A >= 0, cetak "positif"

# KAMUS }
# A : int

# ALGORITMA
A = int(input())

if (A >= 0):
    print("positif")
# else: tidak dilakukan apa pun
```





# Contoh-4: Genap atau Ganjil?

## [Contoh Dua Kasus Komplementer]

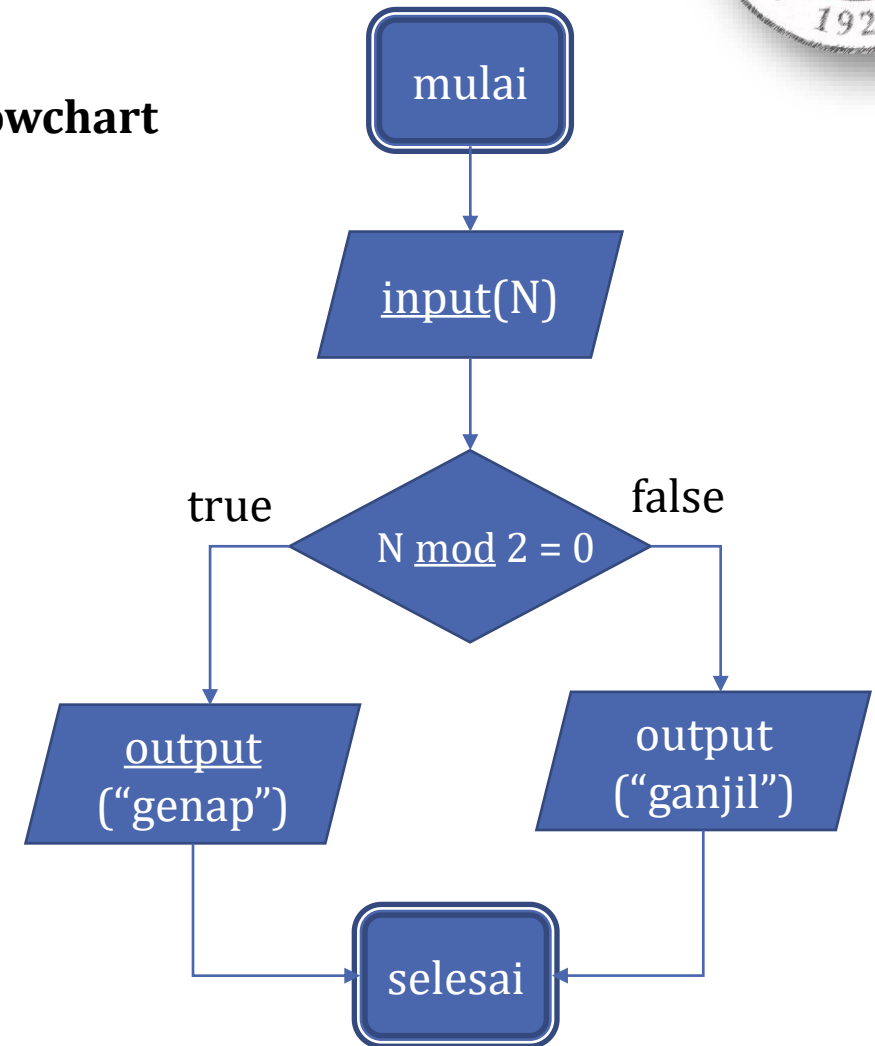
- Buatlah program yang menerima masukan sebuah integer positif (asumsikan masukan pasti benar), misalnya  $N$ , kemudian tentukan apakah bilangan tersebut genap atau ganjil.
- $N$  adalah bilangan genap jika  $N \bmod 2 = 0$ ; jika  $N \bmod 2 = 1$ , maka  $N$  adalah bilangan ganjil
  - Tidak ada kasus lain.
- Kasus:
  - Jika  $N \bmod 2 = 0$  maka cetak “genap”
  - Jika tidak ( $N \bmod 2 = 1$ ), maka cetak “ganjil”

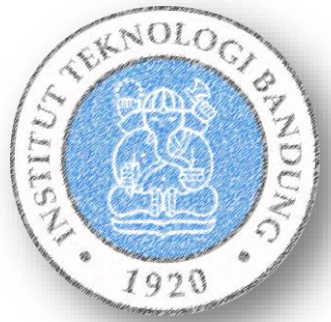
# Contoh-4: Pseudocode + Flowchart

## Pseudocode

```
input(N)  
if (N mod 2 = 0) then  
    output("genap")  
else { N mod 2 = 1 }  
    output("ganjil")
```

## Flowchart





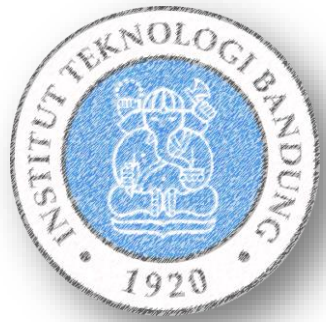
# Contoh-4: Python

```
# Program GenapGanjil
# Input N>0. Jika N genap, cetak "genap"
# Jika tidak, cetak "N ganjil"

# KAMUS
# N : int

# ALGORITMA
N = int(input()) # Asumsi N > 0

if (N % 2 == 0):
    print("genap")
else: # N % 2 == 1
    print("ganjil")
```



# Contoh-5: Positif, Negatif, atau Nol?

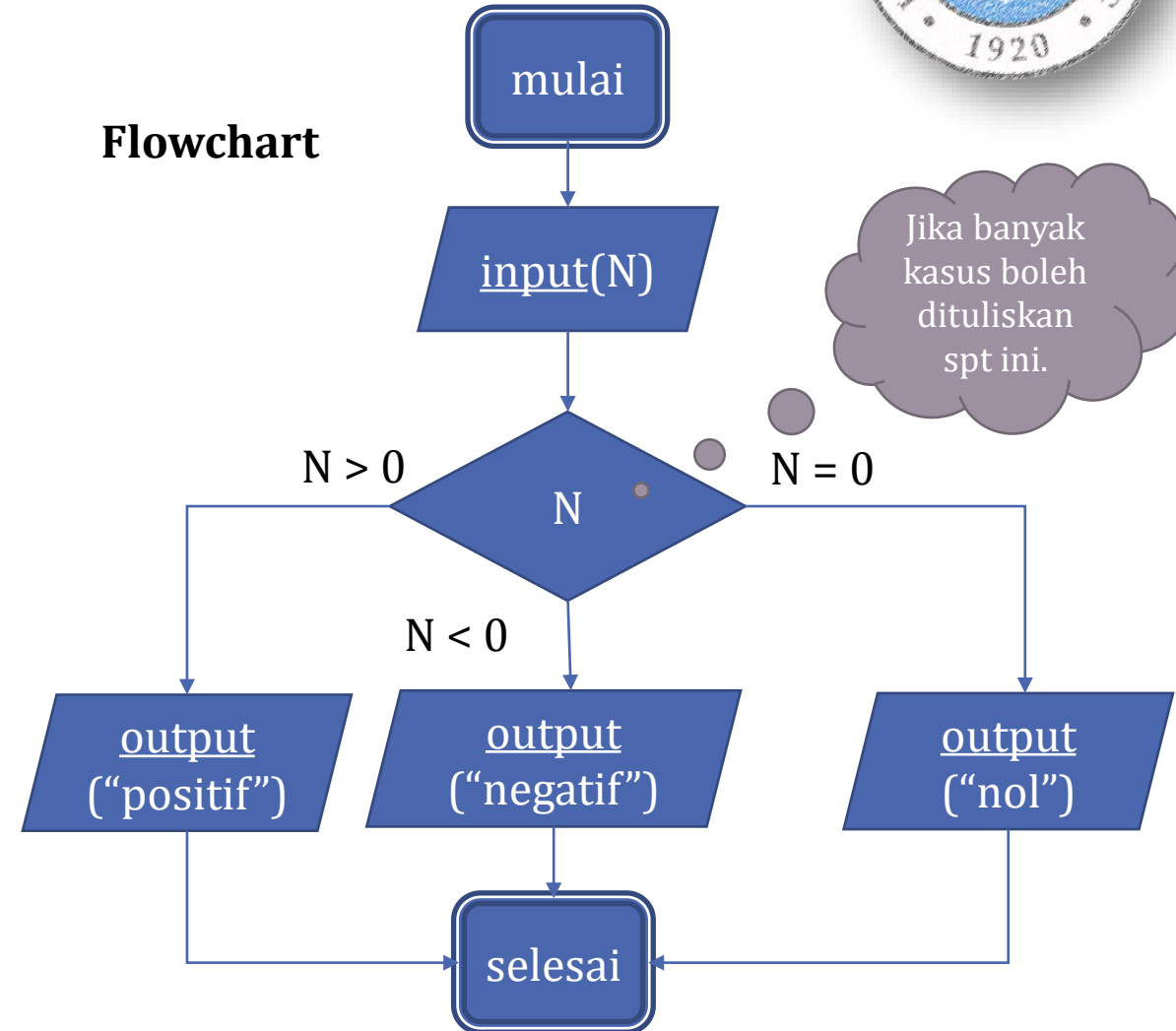
- Buatlah program yang menerima masukan sebuah integer, misalnya N, dan menentukan apakah N adalah bilangan bulat positif, negatif, atau nol
- Kasus:
  - Jika  $N > 0$ ; cetak “positif”
  - Jika  $N < 0$ , cetak “negatif”
  - Jika  $N = 0$ ; cetak “nol”

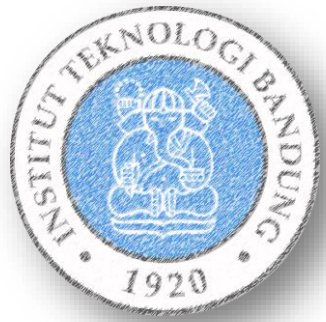
# Contoh-5: Pseudocode + Flowchart

## Pseudocode

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

## Flowchart





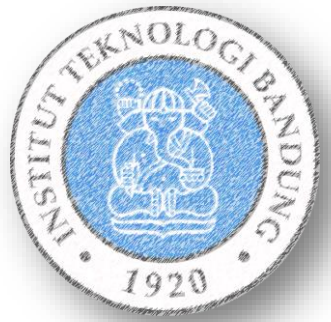
# Contoh-5: Python

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

# KAMUS
# N : float

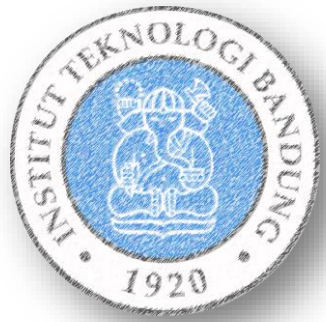
# ALGORITMA
N = int(input())

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



# Latihan Soal

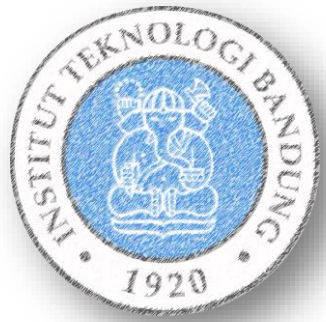
- Untuk soal-soal berikut berlatihlah untuk membuat:
  - Flowchart atau Pseudocode (silakan pilih, atau ditentukan oleh dosen kelas)
  - Program Python yang bersesuaian



# 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)
- Apakah ini 3 kasus atau hanya 2 kasus?





## Latihan-2. Wujud Air

- Buatlah sebuah program yang menerima suhu air (dalam derajat celcius) dan menuliskan wujud air ke layar yaitu **beku**, **cair**, atau **uap**.
  - Jika suhu air  $\leq 0$  derajat, maka tuliskan “beku”
  - Jika suhu air  $> 0$  dan  $< 100$  derajat, maka tuliskan “cair”
  - Jika suhu air  $\geq 100$ , maka tuliskan “uap”

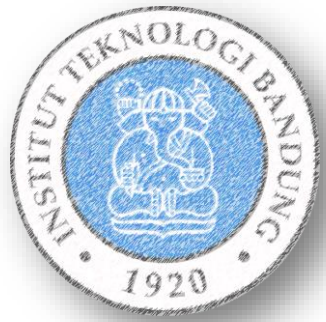


## Latihan-3. Total Hambatan Seri

- Buatlah program yang menerima 3 buah hambatan ( $R_1$ ,  $R_2$ ,  $R_3$ ) dan menghasilkan hambatan total ( $R_T$ ) jika dirangkai seri.

$$R_T = R_1 + R_2 + R_3$$

- $R_1$ ,  $R_2$ , dan  $R_3$  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”.



## Latihan-4. Ranging 3 Bilangan

- Diberikan 3 buah integer yang dibaca dari keyboard, misalnya A, B, C. Asumsikan bahwa ketiga bilangan tersebut **berbeda**.
- Tuliskan ke layar ketiga bilangan tersebut dalam urutan dari yang terbesar sampai yang terkecil.
- Contoh:  $A = 1, B = -1, C = 2$   
Maka tertulis di layar: 2 1 -1