

# Struktur Data Program Prosedural - Python (Bag. 2)

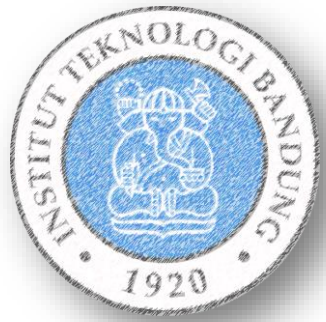
Tim Penyusun Materi Pengenalan Komputasi  
Institut Teknologi Bandung © 2019



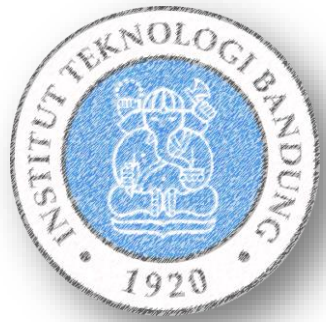


# Tujuan

- Mahasiswa dapat
  - menjelaskan dan menggunakan ekspresi dan operasi yang berkaitan dengan type data
  - menjelaskan dan menggunakan aksi sekuensial dengan benar dalam program prosedural
  - membuat program kecil dengan menggunakan Bahasa Python

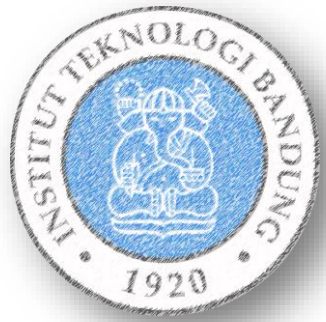


# Ekspresi



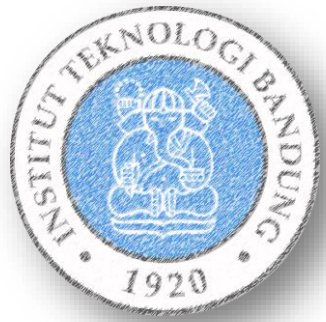
# Ekspresi

- **Ekspresi** adalah kombinasi dari satu atau lebih variabel, konstanta, operator, dan fungsi yang bermakna menurut aturan suatu bahasa pemrograman dan menghasilkan suatu nilai dalam suatu type tertentu
- **Operator** adalah suatu fungsi standar yang disediakan dalam bahasa pemrograman untuk melakukan beberapa hal dasar seperti perhitungan aritmatika, logika, dan relasional.
- Struktur umum ekspresi [biner]: `<operan1> <operator> <operan2>`
- Hasil dari operasi bergantung pada tipe data operan
- Operan dapat berupa nilai, variable, konstanta, atau ekspresi lain



# Jenis Ekspresi

- Jenis ekspresi menurut *arity* dari operator:
  - Ekspresi **biner**: bentuk dasarnya adalah operasi dengan 2 operan
    - Contoh:  $A + 5$
  - Ekspresi **uner**: bentuk dasarnya adalah operasi dengan 1 operan
    - Contoh: `not (found)`
- Jenis ekspresi menurut tipe data yang dihasilkan:
  - Ekspresi **aritmatika**: operan bertipe numerik (int/float) dan menghasilkan nilai numerik
  - Ekspresi **relasional**: operan bertipe numerik (int/float) dan menghasilkan nilai bool/logika
  - Ekspresi **logika**: operan bertipe bool/logika dan menghasilkan nilai bool/logika

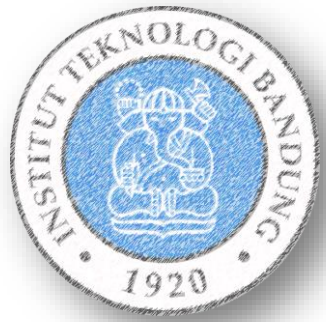


# Operator Tipe Dasar (1)

## Operator Aritmatika

Jika **a = 10** dan **b = 21**, maka:

Operator	Description	Type Operan	Example
+	<b>Penjumlahan:</b> menambahkan nilai kedua operan	int, float	$a + b = 31$
-	<b>Pengurangan:</b> mengurangi nilai operan kiri dengan nilai operan kanan	int, float	$a - b = -11$
*	<b>Perkalian:</b> mengalikan nilai kedua operan	int, float	$a * b = 210$
//	<b>Pembagian bulat:</b> Jika operan adalah int, maka hasil operasi adalah pembagian bulat	int	$b // a = 2$
/	<b>Pembagian riil:</b> Jika operan adalah float, maka hasil operasi adalah pembagian bilangan float	int, float	$b / a = 2.1$
%	<b>Modulo:</b> sisa hasil pembagian bulat	int	$b \% a = 1$
**	<b>Pangkat:</b> mengangkat operan kiri dengan operan kanan	int, float	$10 ** 2 = 100$

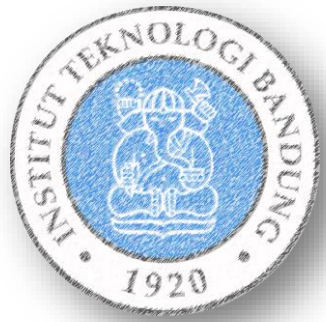


# Operator Tipe Dasar (2)

## Operator Relasional

Jika **a = 10** dan **b = 21**, maka:

Operator	Description	Type Operan	Example
==	Jika nilai kedua operan sama, maka menghasilkan true (tidak berlaku untuk bilangan riil)	int, char, string, bool	(a == b) menghasilkan false
!=	Jika nilai kedua operan tidak sama, maka menghasilkan true	int, float, char, string, bool	(a != b) menghasilkan true
>	Jika nilai operan kiri lebih besar dari operan kanan, maka menghasilkan true	int, float, char, string	(a > b) menghasilkan false
<	Jika nilai operan kiri lebih kecil dari operan kanan, maka menghasilkan true	int, float, char, string	(a < b) menghasilkan true
>=	Jika nilai operan kiri lebih besar dari atau sama dengan operan kanan, maka menghasilkan true	int, float, char, string	(a >= b) menghasilkan false
<=	Jika nilai operan kiri lebih kecil dari atau sama dengan operan kanan, maka menghasilkan true	int, float, char, string	(a <= b) menghasilkan true



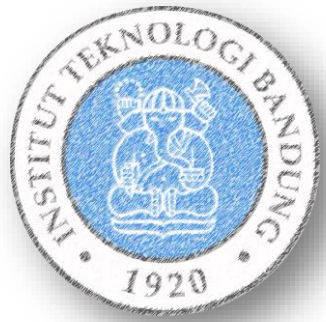
# Operator Tipe Dasar (3)

## Operator Logika

Jika **a = true** dan **b = false**, maka:

Operator	Description	Type Operan	Example
and	Logika AND: Jika kedua operan bernilai true, maka menghasilkan true.	bool	(a and b) menghasilkan false
or	Logika OR: Jika setidaknya salah satu dari kedua operan bernilai true, maka menghasilkan true.	bool	(a or b) menghasilkan false
not	Logika NOT/negasi: Untuk membalik nilai logika dari operannya.	bool	not(a) menghasilkan false





# Operator Tipe Dasar (4)

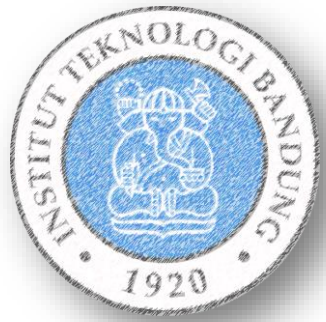
## Operator Assignment

Jika **a = 10** dan **b = 21**, maka:

Operator	Description	Type Operan	Example
<op>=	<op> adalah + - * / % Meringkas operasi: A = A <op> B menjadi A <op>= B	int, float	a+=b; maka a = 31 (setara a = a + b) a*=b; maka a = 210 (setara a = a * b)



# Aksi Sekuensial

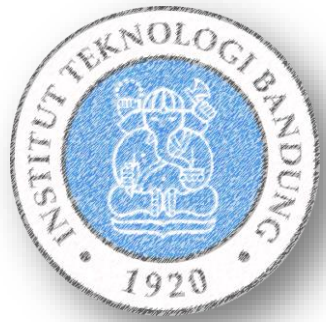


# Struktur Dasar Algoritma

**Program** <JudulProgram>  
{ Spesifikasi Program }

**KAMUS**  
{ Deklarasi type, variabel, konstanta, fungsi,  
prosedur }

**ALGORITMA**  
{ Deretan langkah algoritmik untuk penyelesaian  
persoalan }  
{ Ditulis dengan pseudocode atau flowchart }

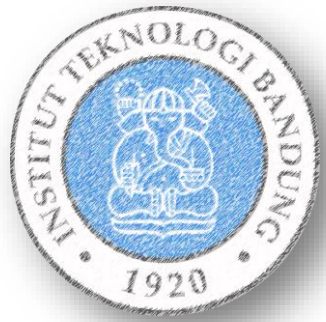


# Struktur Dasar Program Python

```
# Program <JudulProgram>
# Spesifikasi Program

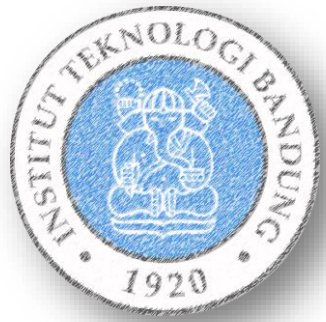
# KAMUS
# Penjelasan dalam bentuk komentar
# Deklarasi type, variabel, konstanta, fungsi, prosedur

# ALGORITMA
# Deretan langkah algoritmik untuk penyelesaian # persoalan
```



# Bagian Algoritma dari Program

- Adalah bagian program dalam bentuk teks algoritmik yang berisi instruksi atau pemanggilan aksi
- Teks algoritmik tsb. dapat berupa:
  - Perintah dasar: Input/Output, assignment
  - Perintah perintah yang berurutan
  - Analisis kasus (jika-maka)
  - Pengulangan... dll.
- Dalam Bahasa Python, setiap instruksi ditulis per baris
  - Jika lebih dari 1 instruksi dituliskan pada satu baris, maka setiap instruksi dipisahkan oleh titik koma (;)
  - Contoh: `nama = input(); print(nama)`



# Aksi Sekuensial

- **Aksi sekuensial:** sederetan instruksi primitif dan/atau aksi yang akan dilaksanakan (dieksekusi) oleh komputer berdasarkan urutan penulisannya
- Setiap aksi akan mengubah status dari program
  - Jadi setiap aksi sekuensial harus ada awal dan akhir.
  - Dengan kata lain, suatu program harus dimulai dan suatu ketika harus berakhir
- Instruksi ditulis terurut sesuai penulisan per baris
- Perhatikan bahwa:
  - ada program yang akan berubah jika urutan baris instruksinya berubah
  - dan ada juga program yang tidak berubah jika urutan baris instruksinya berubah

# Urutan instruksi tidak mengubah hasil eksekusi...

```
# Program Test
# KAMUS
# i : int
# x : float
```

```
# ALGORITMA
i = int(input())
x = 100.75
```

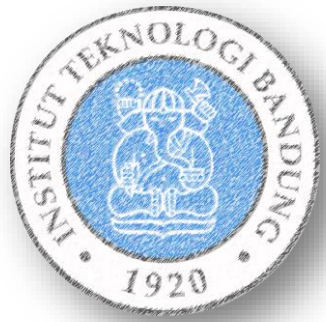
```
print(x)
print(i*2)
```

```
# Program Test
# KAMUS
# i : int
# x : float
```

```
# ALGORITMA
x = 100.75
i = int(input())
```

```
print(x)
print(i*2)
```

Hasil eksekusi **tidak berubah**, walaupun urutan instruksi diubah



# Urutan instruksi mengubah hasil eksekusi...

```
# Program Test
# KAMUS
# i : int
# x : float

# ALGORITMA
i = int(input())
x = 100.75
```

```
print(x)
print(i*2)
```

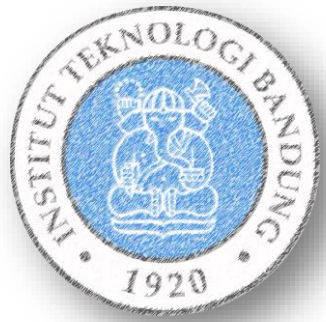
```
# Program Test
# KAMUS
# i : int
# x : float

# ALGORITMA
i = int(input())
x = 100.75
```

```
print(i*2)
print(x)
```

Hasil eksekusi **berubah** karena urutan instruksi diubah





# Blok Program (1)

- Sederetan instruksi yang dieksekusi secara sekuensial dikelompokkan dalam blok program
- Dalam Python, satu blok program ditandai dengan indentasi yang semakin menjorok ke dalam
- Dalam 1 blok program dimungkinkan ada blok program lain yang berada lebih di dalam (*inner block*)
- Jika instruksi berada dalam 1 blok, maka indentasi harus rapi. Jika tidak, akan *error*.

# Blok Program (2)

```
a = int(input("Masukkan angka = "))
if (a > 50):
    print ("Hello World!")
    print ("bye")
else: # a <= 50
    print ("Hello Darling!")
    print ("bye bye")
```

OK!

```
a = int(input("Masukkan angka = "))
if (a > 50):
    print ("Hello World!")
    print ("bye")
else: # a <= 50
    print ("Hello Darling!")
    print ("bye bye")
```

Error!

Contoh-2:

- Baris ke-7 s.d. 13 dalam 1 blok (*outer block*)
- Baris ke-9 s.d. 10 dalam 1 blok (*inner block*)
- Baris ke-12 s.d. 13 dalam 1 blok (*inner block*)

```
1  # Program Test
2
3  # KAMUS
4  # a : int
5
6  # ALGORITMA
7  a = int(input("Masukkan angka = "))
8  if (a > 50):
9      print ("Hello World!")
10     print ("bye")
11 else: # a <= 50
12     print ("Hello Darling!")
13     print ("bye bye")
14
```

Instruksi if-then-else... coming soon

# Contoh-1. Roda Pak Pit

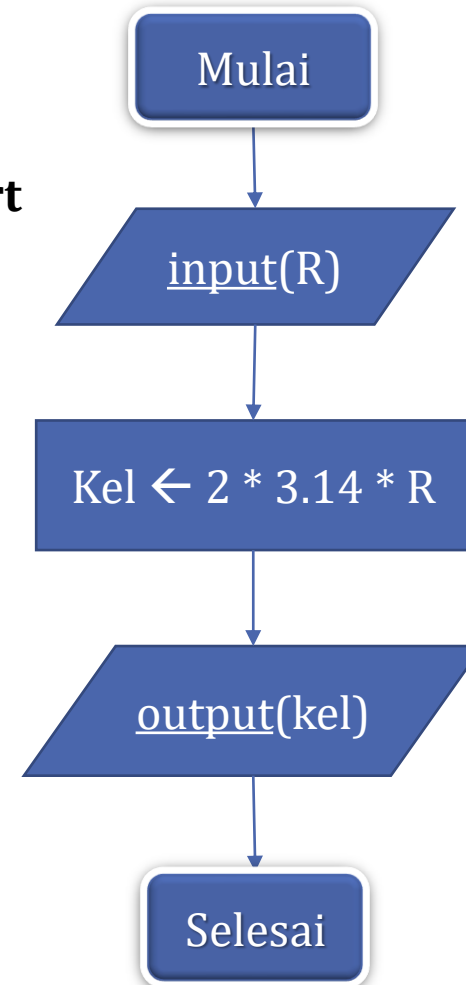
- Pak Pit, seorang pengusaha bengkel sepeda, memberikan tarif untuk setiap roda sepeda yang diperbaikinya berdasarkan keliling dari roda sepeda.
- Untuk itu, ia mengukur jari-jari sepeda, yaitu panjang dari pusat roda sampai tepi roda.
- Buatlah program yang menampilkan hasil perhitungan keliling lingkaran berdasarkan masukan nilai jari-jari.
- Rumus menghitung keliling lingkaran:  $2 \Pi r$ 
  - $r$  adalah panjang jari-jari

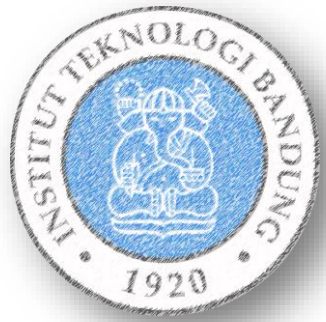
# Contoh-1: Pseudocode + Flowchart

## Pseudocode

```
input(R)  
Kel  $\leftarrow$  2 * 3.14 * R  
output(R)
```

## Flowchart





# Contoh-1: Python

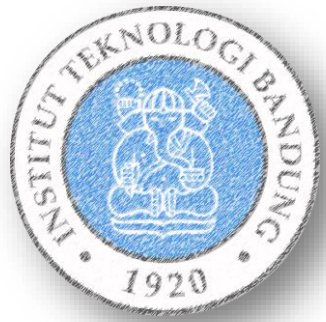
```
# Program KelilingLingkaran
# Menghitung keliling lingkaran berdasarkan masukan jari-jari

# KAMUS
# R : float
# Kel : float

# ALGORITMA
R = float(input())

Kel = 2 * 3.14 * R

print(Kel)
```



## Contoh-2. Tinggi Rata-Rata

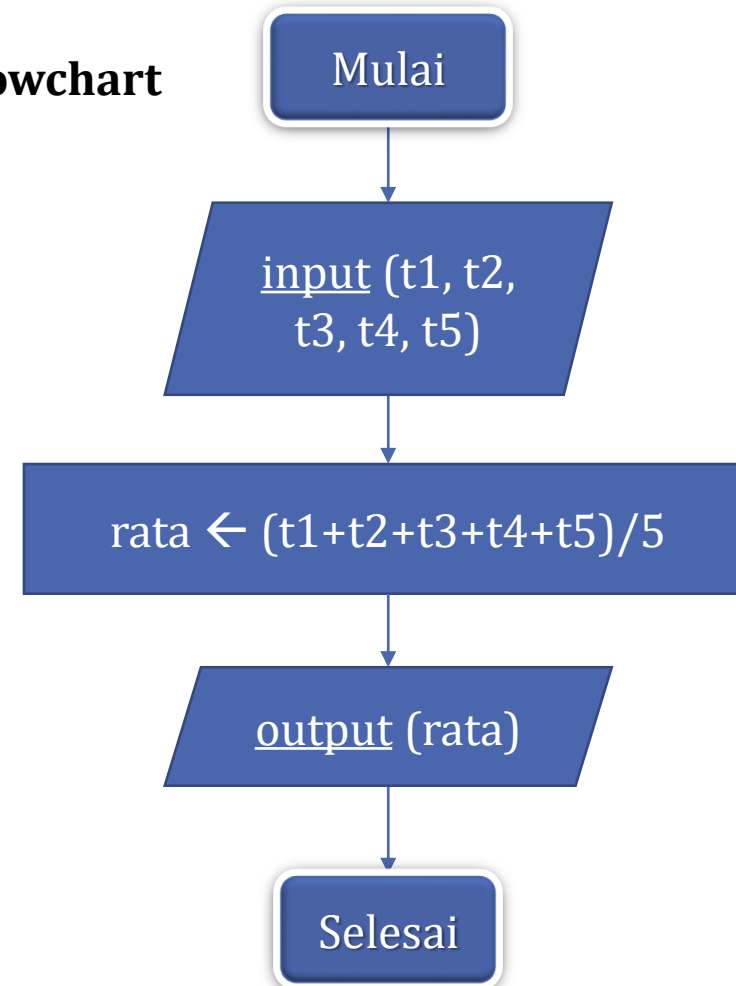
- Pak Guru menyeleksi 5 orang anak yang akan masuk ke tim basket sekolah. Ia ingin mengetahui tinggi badan rata-rata mereka.
- Buat program menghitung rata-rata dari tinggi badan 5 anak
  - Program akan menerima masukan data tinggi badan untuk 5 orang anak
  - Selanjutnya program menampilkan tinggi rata-rata dari ke lima anak tersebut

# Contoh-2: Pseudocode + Flowchart

## Pseudocode

```
input(t1, t2, t3, t4, t5)  
rata ← (t1+t2+t3+t4+t5)/5  
output(rata)
```

## Flowchart





# Contoh-2: Python

```
# Program TinggiRataRata
# Menerima tinggi 5 siswa dan menghitung rata-ratanya

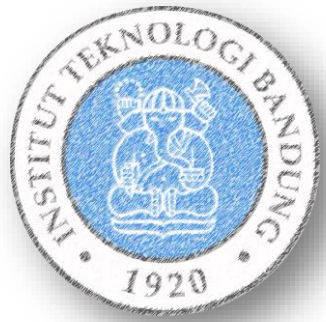
# KAMUS
# t1, t2, t3, t4, t5 : float
# rata : float

# ALGORITMA
t1 = float(input())
t2 = float(input())
t3 = float(input())
t4 = float(input())
t5 = float(input())

rata = (t1 + t2 + t3 + t4 + t5)/5

print (rata)
```





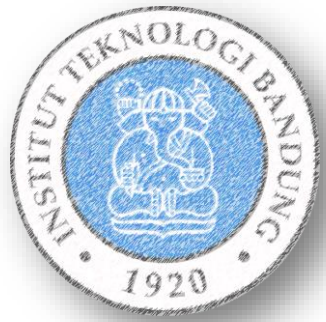
# Latihan

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



# Latihan-1: Hitung Jarak

- Dalam Fisika, jarak (s) dapat dihitung berdasarkan kecepatan (v) dan waktu tempuh (t), yaitu:  $s = v * t$
- Buatlah program untuk menghitung jarak (dalam m) berdasarkan masukan kecepatan (dalam m/s) dan waktu (dalam s)



## Latihan-2. Umbul-Umbul Segitiga

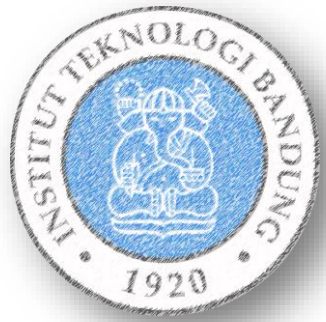
- Bu Tuti adalah seorang pengusaha umbul-umbul yang terkenal di kotanya. Dia membuat berbagai umbul-umbul dari berbagai bentuk, termasuk segitiga.
- Untuk setiap umbul-umbul segitiga, Bu Tuti menetapkan harga umbul-umbul berdasarkan luasnya. Untuk bisa menghitung luas umbul-umbul, Bu Tuti memerlukan tinggi dan alas umbul-umbul.
- Buatlah program yang menerima masukan tinggi dan alas dan menghasilkan luas umbul-umbul segitiga.
- Rumus luas segitiga:  $\text{luas} = \frac{1}{2} * \text{alas} * \text{tinggi}$

## Latihan-3. Toko Kelereng

- Sebuah toko menjual kelereng. Berikut adalah tabel harga kelereng berdasarkan warnanya:

Warna kelereng	Harga 1 butir (dalam ratusan rupiah)
Merah	10
Hijau	15
Kuning	20

- Seorang anak membeli kelereng sejumlah  $m$  kelereng merah,  $h$  kelereng hijau, dan  $k$  kelereng kuning. Asumsikan  $m \geq 0$ ,  $h \geq 0$ ,  $k \geq 0$ .
- Hitunglah berapa yang harus dibayarkan anak itu.



# Alternatif Solusi Latihan

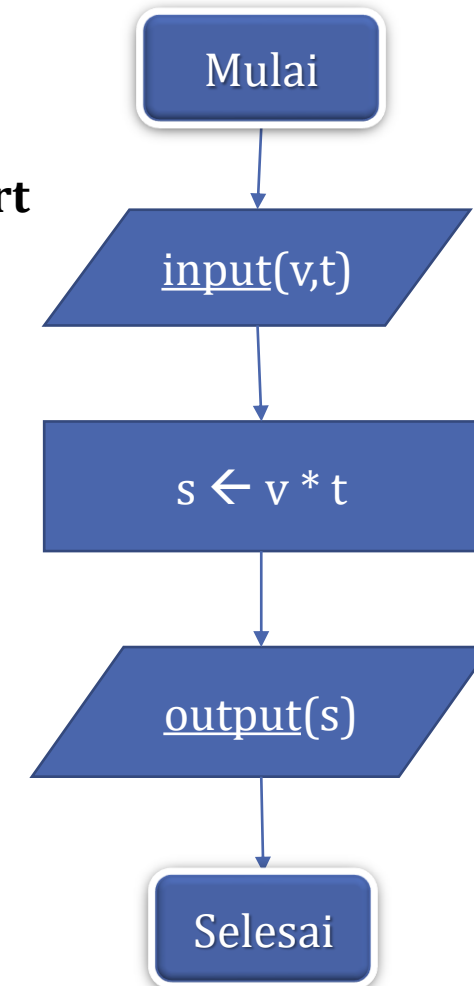
Hanya untuk dosen

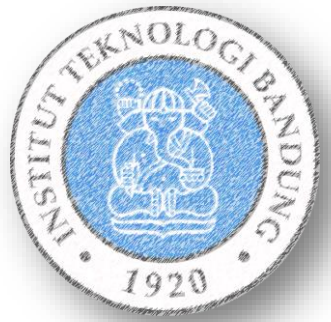
# Latihan-1: Pseudocode + Flowchart

## Pseudocode

```
input(v, t)  
S  $\leftarrow$  v * t  
output(s)
```

## Flowchart





# Latihan-1: Python

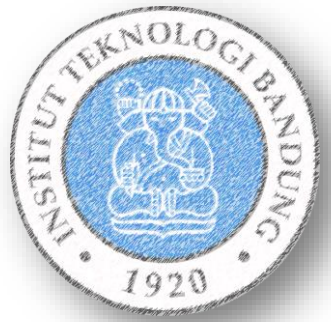
```
# Program HitungJarak
# Menghitung jarak berdasarkan masukan kecepatan (v) dan waktu (t)

# KAMUS
# v, t, s : float

# ALGORITMA
v = float(input())
t = float(input())

s = v * t

print(s)
```



# Latihan-1: Python - tambahan tampilan

```
# Program HitungJarak
# Menghitung jarak berdasarkan masukan kecepatan (v) dan waktu (t)

# KAMUS
# v, t, s : float

# ALGORITMA
v = float(input("Masukkan kecepatan = "))
t = float(input("Masukkan waktu = "))

s = v * t

print("Jarak = ", s)
```

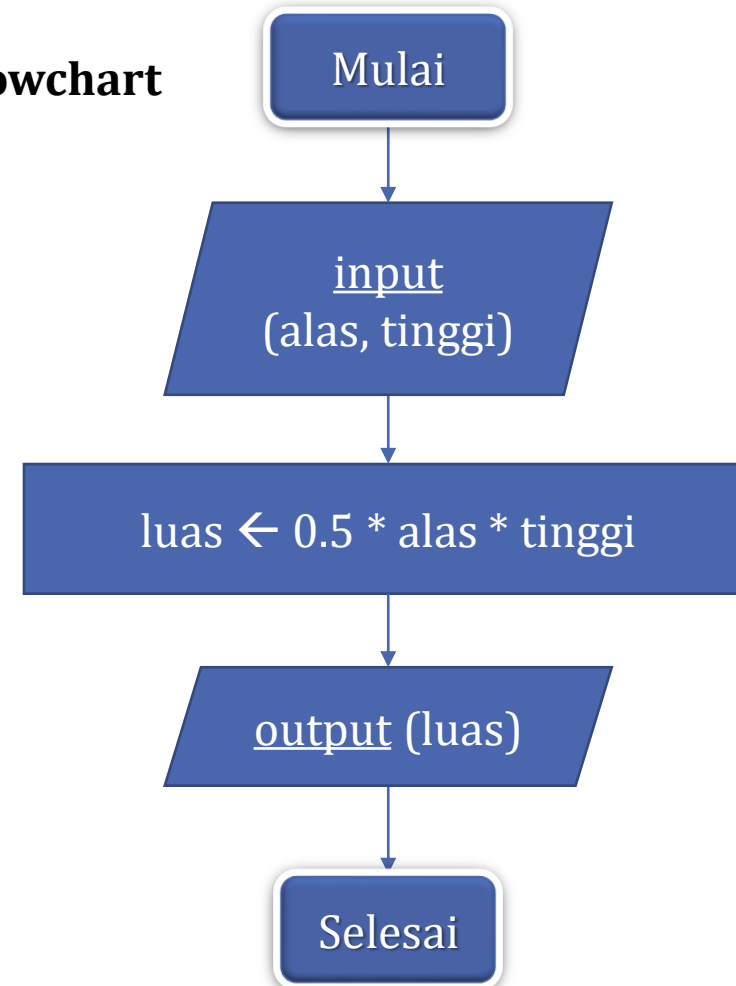


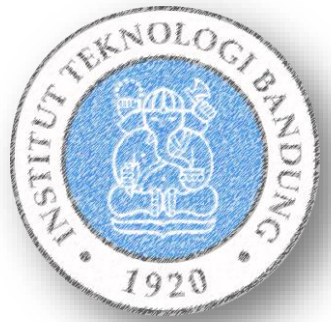
# Latihan-2: Pseudocode + Flowchart

## Pseudocode

```
input(alas, tinggi)  
luas ← 0.5 * alas * tinggi  
output(luas)
```

## Flowchart





# Latihan-2: Python

```
# Program MenghitungLuasSegitiga
# Menghitung luas segitiga berdasarkan masukan alas dan tinggi

# KAMUS
# luas, alas, tinggi : float

# ALGORITMA
alas = float(input())
tinggi = float(input())

luas = 0.5 * alas * tinggi

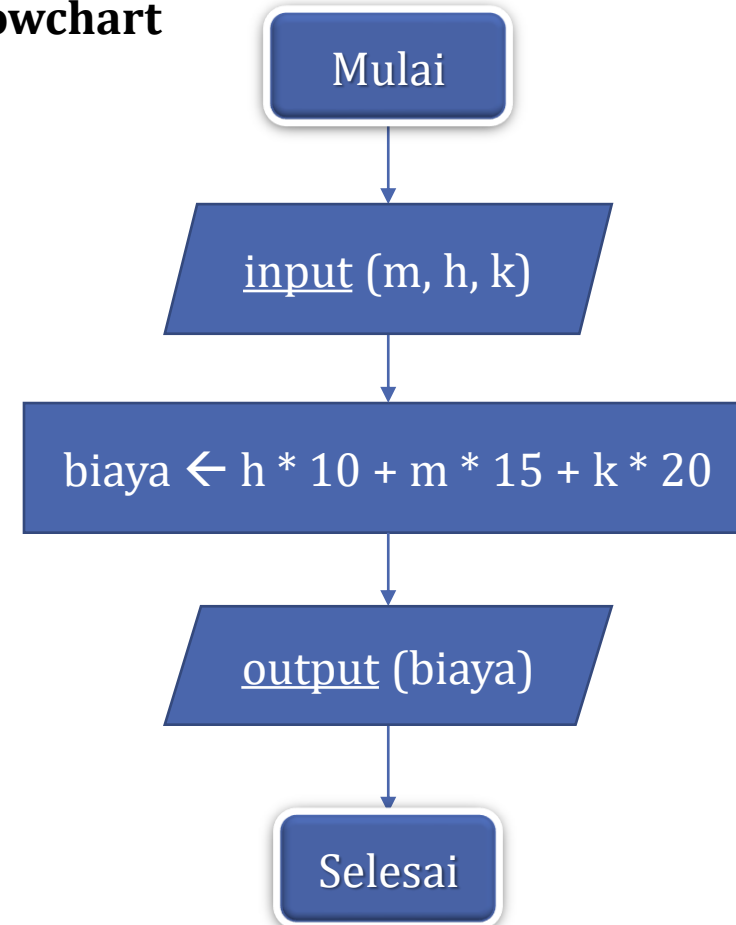
print(luas)
```

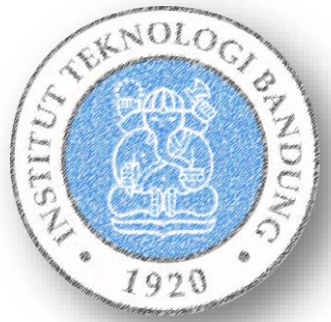
# Latihan-3: Pseudocode + Flowchart

## Pseudocode

```
input(m, h, k)  
biaya  $\leftarrow$  h * 10 + m * 15 + k * 20  
output(biaya)
```

## Flowchart





# Latihan-3: Python

```
# Program MenghitungBiayaBeliKelereng;  
# Menghitung biaya beli kelereng tergantung jumlah kelereng tiap warna  
  
# KAMUS  
# biaya, m, h, k : int  
  
# ALGORITMA  
m = int(input()); h = int(input()); k = int(input())  
  
biaya = m * 10 + h * 15 + k * 20  
  
print(biaya)
```