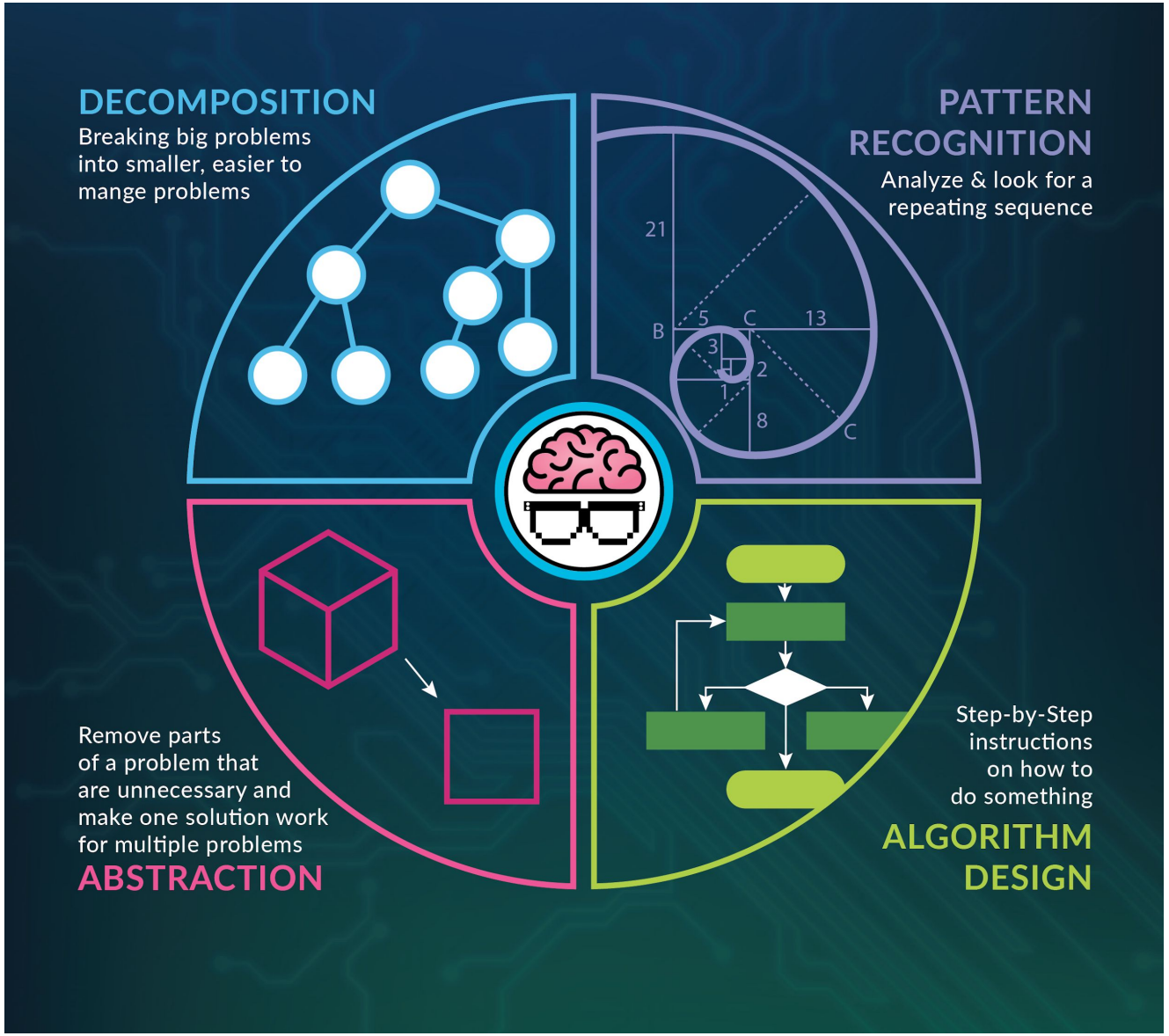
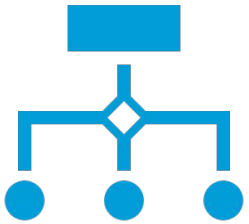


Overview

4 Pillars of Computational Thinking

HTTPS://WWW.COMPUTATIONALTHINKERS.COM/PRODUCT/COMPUTATIONALTHINKING/

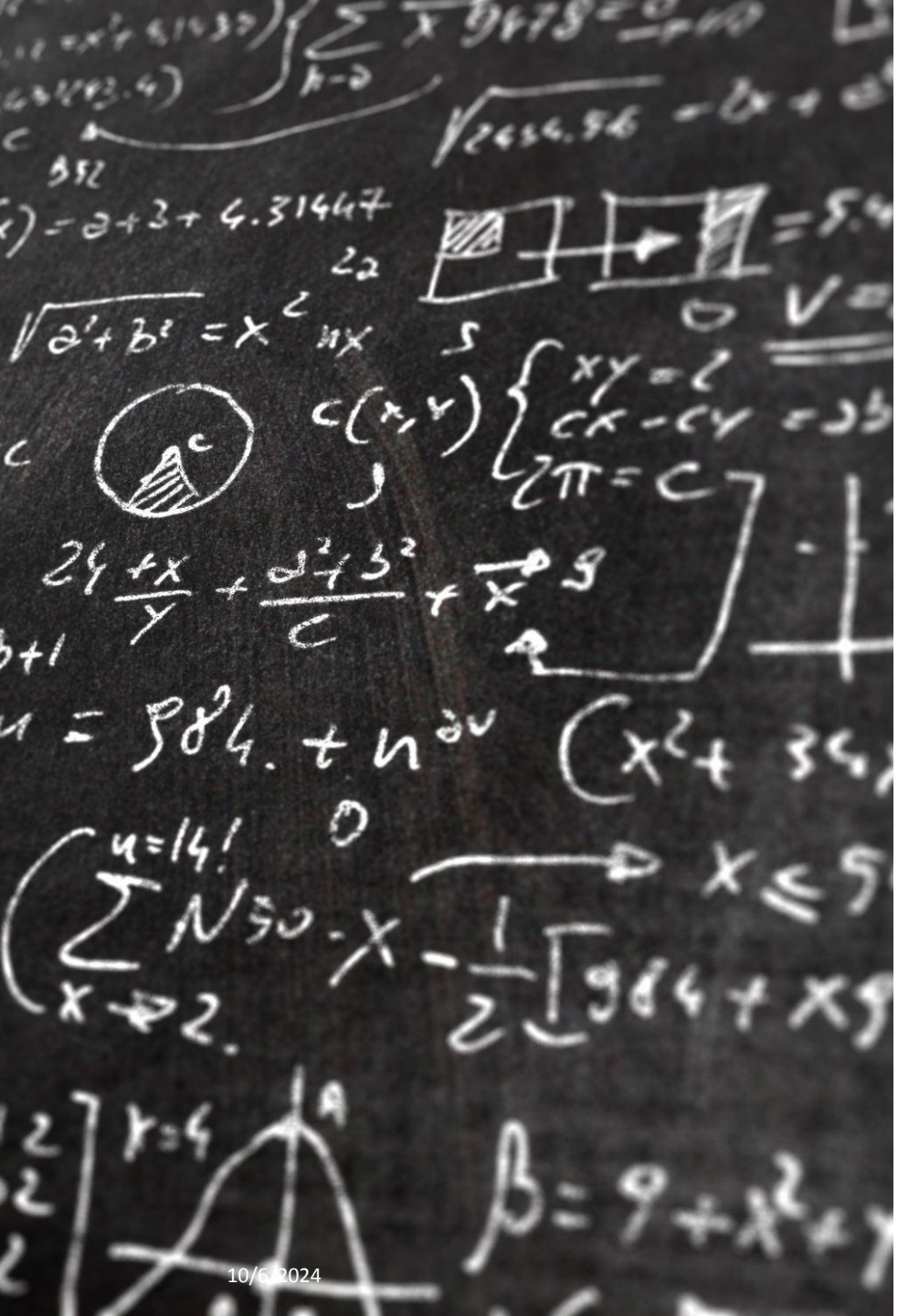
Problem Solving dengan Pemrograman



Pemrograman (*programming*): adalah salah satu bentuk penyelesaian persoalan (*problem solving*) di mana persoalan serta solusinya direpresentasikan dalam bentuk yang bisa diproses oleh **komputer**

Secara umum terdiri atas langkah-langkah sbb.:

- Memahami persoalan
- Menyusun rencana untuk menyelesaikan persoalan **algoritma**
- Menyusun solusi berdasarkan rencana **program komputer**
- Mengevaluasi solusi



Dari Algoritma Menjadi Program

- *Programmer* mengubah **algoritma** menjadi **kode program** komputer dengan menggunakan **bahasa pemrograman**
- Proses untuk menuliskan kode program berdasarkan algoritma disebut sebagai ***coding***
  - File hasil menuliskan kode program: ***source code*** (kode sumber)
- Setiap pernyataan dalam algoritma ditranslasikan secara detil ke dalam kode program
- ***Compiler/interpreter*** akan mentranslasi kode program dalam bahasa pemrograman tertentu menjadi bentuk yang dipahami oleh komputer

Struktur Dasar Program Prosedural – Python

Tim Penyusun Materi WI1102 Berpikir Komputasional
Institut Teknologi Bandung

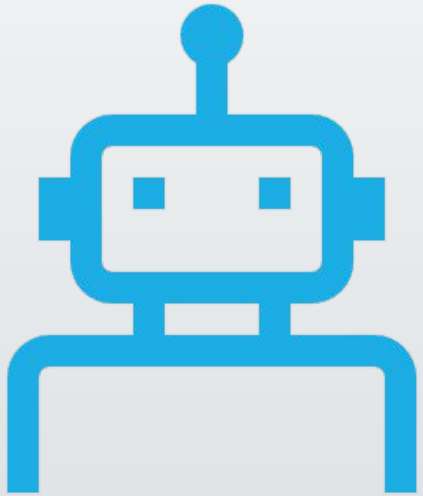
Tujuan

Mahasiswa dapat:

- Menjelaskan struktur dasar program prosedural
- Menjelaskan abstraksi data dan jenis-jenis type data dasar
- Menjelaskan makna dan menggunakan variabel dan konstanta
- Menjelaskan dan menggunakan:
  - perintah assignment dan input/output
  - ekspresi dan operasi yang berkaitan dengan type data
  - aksi sekuensial dengan benar dalam program prosedural
- Membuat program kecil dengan menggunakan Bahasa Python

Python

- Bahasa programming tingkat tinggi, di-*release* oleh Guido van Rossum pada tahun 1991
- Mendukung berbagai paradigma pemrograman. Dalam kuliah ini, hanya akan menggunakan paradigma prosedural.
- Interpreter yg tersedia pada beragam sistem operasi:
  - Indentasi untuk menandai blok program
  - Case sensitive perbedaan huruf besar dan kecil berpengaruh
- Python adalah bahasa pemrograman yang ***strong and dynamically typed***
  - *Strong typed*: Tipe data variabel ditentukan oleh nilai yang di-*assign* pertama dan selanjutnya hanya bisa dioperasikan sesuai tipe tersebut
  - *Dynamically typed*: Tipe data variabel dapat diubah



Struktur Dasar Algoritma

Program <JudulProgram>
{ Spesifikasi Program }

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

ALGORITMA
{ Deretan langkah algoritmik untuk penyelesaian persoalan }

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
```

Program Pertama

- Buatlah program untuk menuliskan "Hello, World!" ke layar.

print adalah perintah untuk mencetak teks ke layar/monitor

```
# Program HelloWorld
# Mencetak Hello, World! ke layar

# KAMUS
# belum diperlukan

# ALGORITMA
print('Hello, World!')
```

Input – Proses – Output


```
input (A)
input (B)
```

```
A ← A +
B
```

```
output (A)
output (B)
```

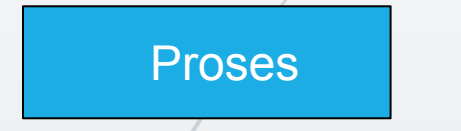
Python

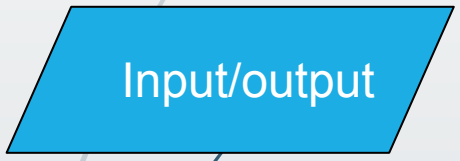
```
A = int(input(‘’))
B = int(input(‘’))
```

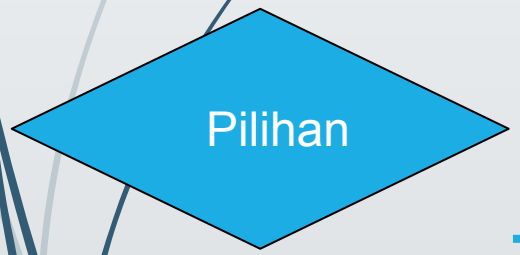
```
A = A + B
```

```
print(A)
print(B)
```

Flow Chart



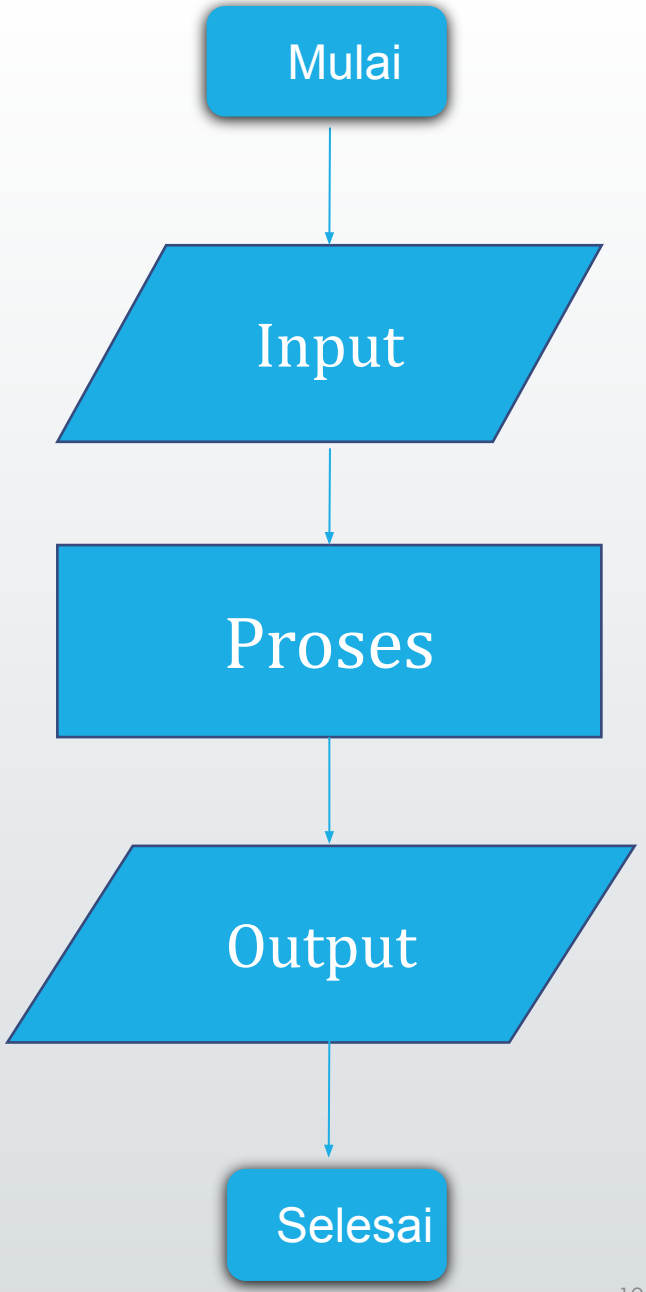




Pilihan

Alur pilihan

Alur



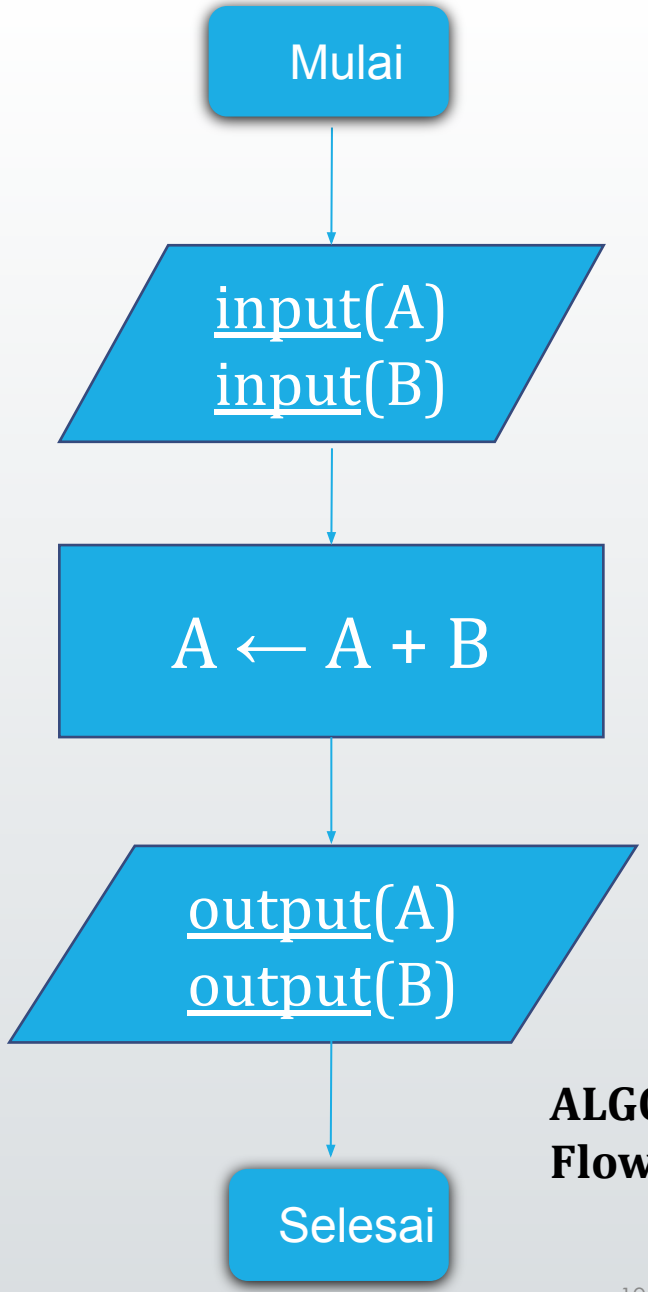
Struktur Dasar Program

Program Test
{ Spesifikasi Program: menghitung A + B }

KAMUS
{ Deklarasi variabel }
    A, B : integer

ALGORITMA - Notasi Algoritmik

```
input(A)
input(B)
A ← A + B
output(A)
output(B)
```



ALGORITMA - Flowchart

Contoh Program Python

```
# Program Test
# Spesifikasi : Menghitung nilai A dan B

# KAMUS
# A : int
# B : int

# ALGORITMA
A = int(input())  # input
B = int(input())

A = A + B         # proses

print(A)          #output
print(B)
```

Judul Program + spesifikasi, dituliskan dalam komentar

KAMUS: deklarasi variabel A dan B (dalam komentar)

ALGORITMA: Input, Proses, Output

Komentar

Dalam bahasa pemrograman komentar adalah bagian program yang tidak dieksekusi

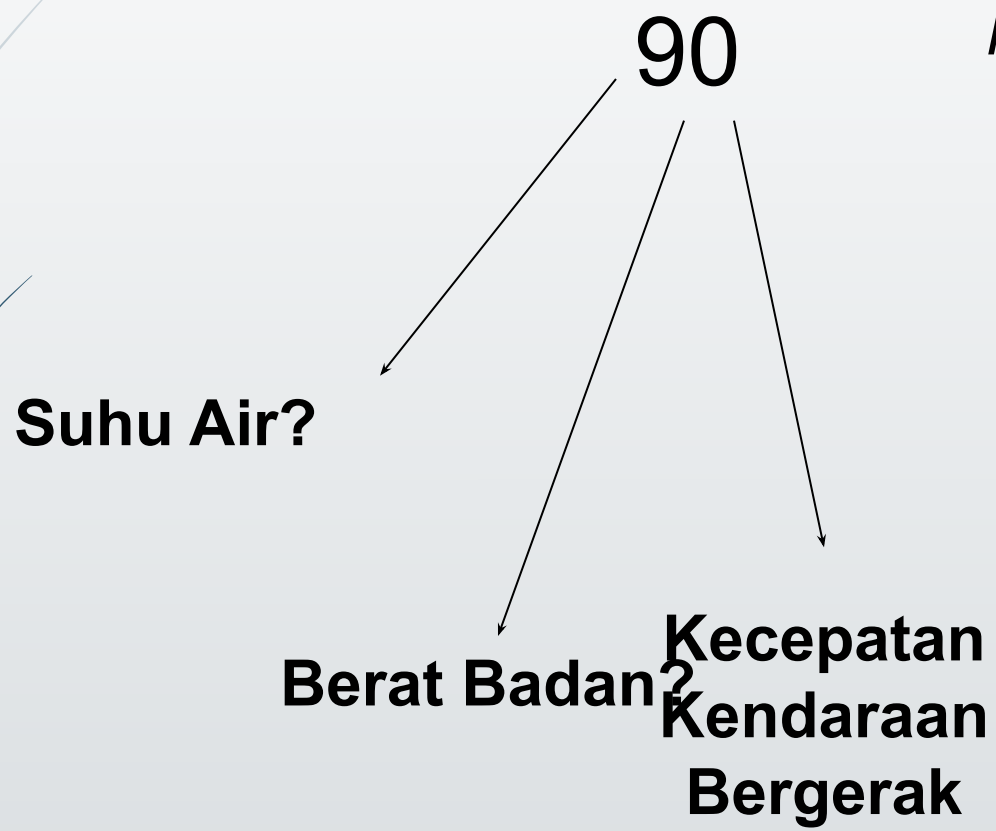
- Bagian ini hanya digunakan untuk memberikan penjelasan suatu langkah, rumus ataupun bisa hanya berupa keterangan

Dalam Python komentar dituliskan per baris diawali dengan **#**

Contoh:
 # ini komentar

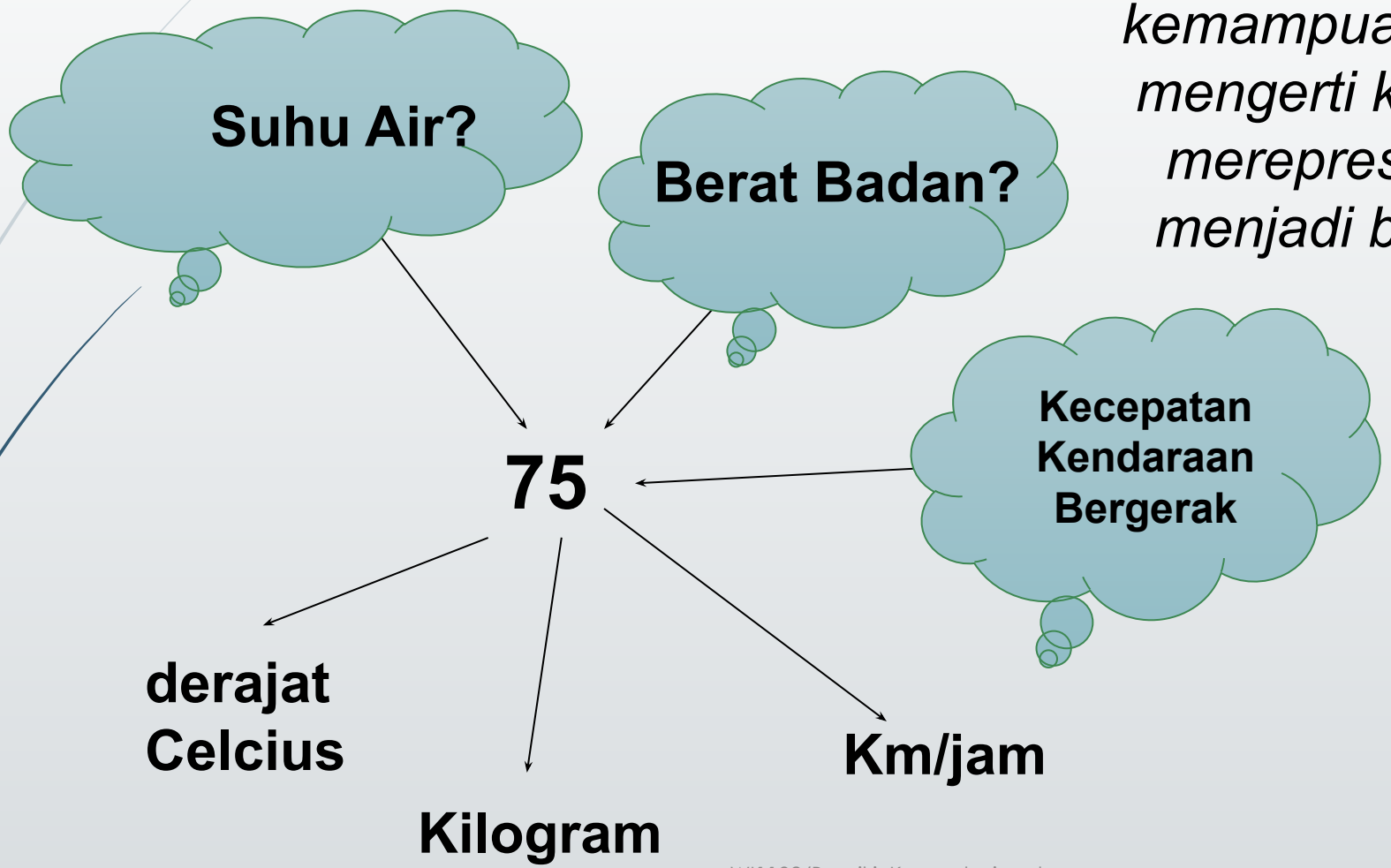
Data

Abstraksi Data



kemampuan kita untuk menginterpretasikan suatu data dengan konteks masalahnya

Persoalan Abstraksi Data



kemampuan kita untuk mengerti konteks dan merepresentasikan menjadi bentuk lain.

Bagian **Kamus**

Bagian **Kamus** dipakai untuk mendeklarasikan nama-nama yang digunakan dalam program

Nama-nama merepresentasikan **data** yang digunakan dalam program

Python adalah bahasa pemrograman yang ***strong and dynamically typed***

Strong typed: Tipe data variabel ditentukan oleh nilai yang di-*assign* pertama dan selanjutnya hanya bisa dioperasikan sesuai tipe tersebut

Dynamically typed: Tipe data variabel dapat diubah

Konvensi kuliah: 1 variabel hanya dipakai untuk 1 tipe data

Dalam menggunakan variabel harus diketahui dengan baik tipe data apa didefinisikan terhadap variabel tersebut

Tuliskan bagian KAMUS dalam bentuk komentar type suatu variabel

Tipe Data (1)

Setiap data memiliki jenis yang berbeda-beda

- Data **umur** seseorang berbeda dengan data **nama**
  - Data umur dibentuk dari kumpulan angka
  - Data nama dibentuk dari serangkaian huruf
- Untuk setiap jenis data juga memiliki rentang (range) yang berbeda
  - Data umur rentangnya antara 1 sampai 100 (bila diasumsikan bahwa umur seseorang tidak lebih dari 100).
  - Data nama rentangnya mulai dari 1 sampai 50 (bila di anggap nama tidak ada yang melebihi 50 huruf

Tipe Data (2)

Nilai yang diperbolehkan untuk variabel tergantung pada tipe data-nya

Tipe data mendefinisikan himpunan nilai-nilai tertentu, misalnya:

Tipe data integer : himpunan nilai yang terdiri atas bilangan bulat (negatif, 0, positif)	Tipe data boolean: himpunan nilai yang terdiri atas nilai true dan false

Tipe Data Dasar/Primitif

Disediakan oleh bahasa pemrograman

Python	Domain Nilai
Bool	Nilai boolean: **True**; **False**
Numbers	Nilai-nilai numerik. Jenis nilai numerik: • **int** : integer/bilangan bulat bertanda (+/-). Contoh: 1; -144; 999; 0 • **float** : floating point (real). Contoh: 3.14; 4.01E+1 • **complex** : bilangan kompleks tidak akan digunakan di kelas ini
string	Kumpulan karakter/huruf, ditandai dengan kutip tunggal atau kutip ganda. Contoh: 'xcxcx'
char	Character: karakter/huruf, ditandai dengan kutip tunggal; Contoh: 'A'; '#'; 'b'

Contoh Penentuan Tipe Data Variabel

- Umur → **Integer** contoh: 25; 44; 35
- Kota → **String**, contoh: “Jakarta”; “Bandung”
- Nama → **String**, contoh: “Budi”; “Ali”
- Suhu → **Integer** atau float, contoh: 37.5; 100
- Luas → **Integer** atau float, contoh: 400; 43.5
- BeratBadan → **Integer** atau float, contoh: 60.5; 75
- NIM → **Integer** atau string?, contoh: 15812001

Variabel (1)

Variables digunakan menyimpan suatu nilai yang ber-"tipe data" tertentu sesuai dengan deklarasi

Merepresentasikan suatu makna di dunia nyata yang ingin diolah dalam program, misalnya:

Sum : jumlah beberapa angka

Max : nilai maksimum

Min : nilai minimum

Penggunaan variabel:

Deklarasi (supaya nama dikenal dan diketahui tipe datanya),

inisialisasi dan manipulasi nilai

Variabel (2)

Contoh deklarasi dan inisialisasi variabel:

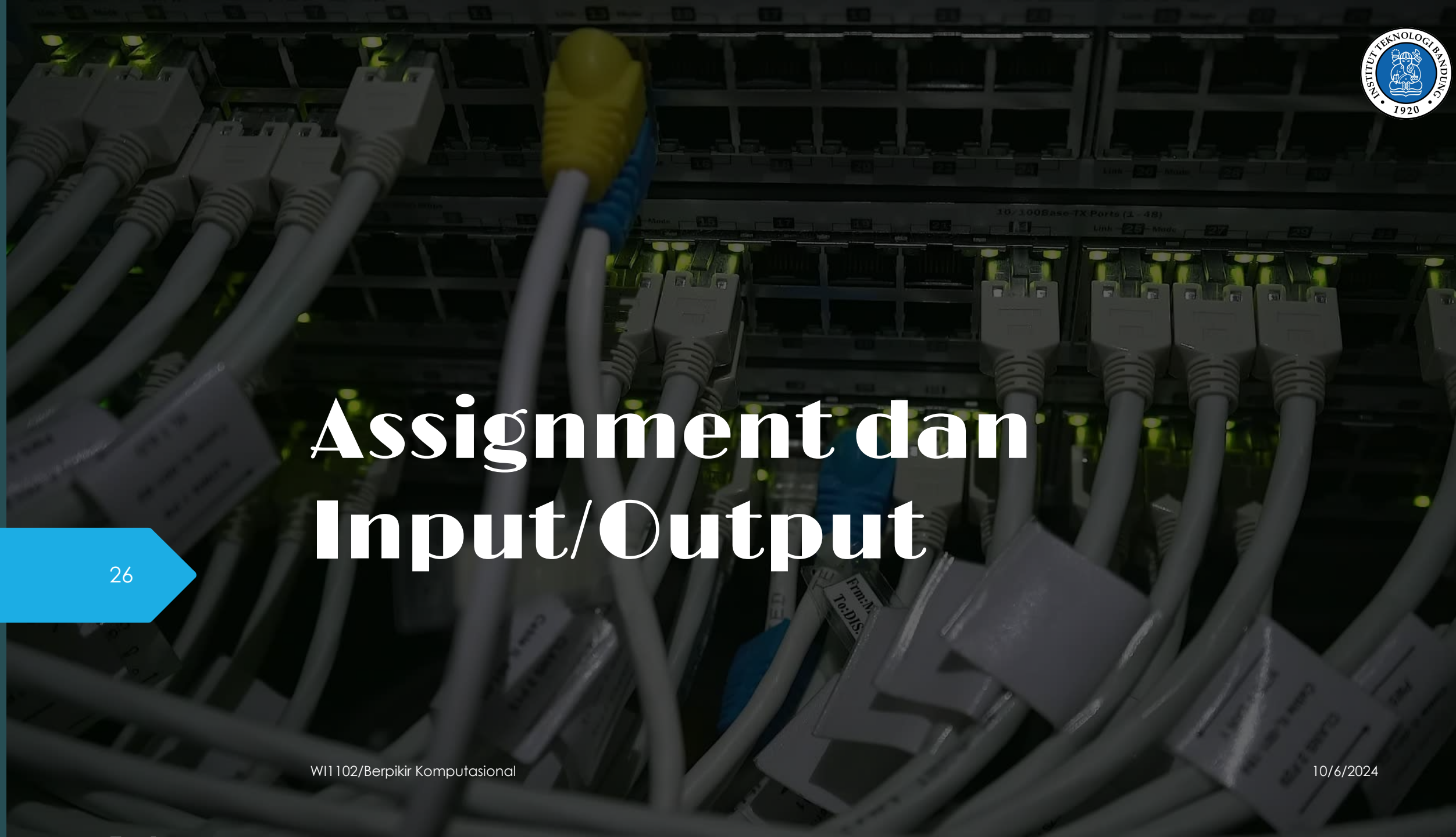
Python

```
# KAMUS
# i : int
# A : int

# ALGORITMA
...
i = 100
A = i * 50
....
```

Membuat Nama Variabel yang Benar dan "baik"

- Nama variabel harus dimulai dengan huruf dan dapat diikuti dengan huruf lagi dan angka
  - Tidak boleh ada karakter lain, kecuali: *underscore* (_)
- Dalam nama variabel tidak boleh dipisahkan oleh spasi
- Cari nama variabel yang bisa dimengerti dan tidak membingungkan
  - Contoh: **sum** adalah untuk jumlah, bertype integer. Jangan guankan untuk data bertype lain
- Python adalah bahasa yang ***case sensitive***: Kesalahan penulisan huruf besar dan kecil menyebabkan error



Assignment dan Input/Output

Pemberian Nilai

Suatu besaran (dengan tipe tertentu), misalnya variabel, yang telah dikenal dapat diberi **nilai/harga**

Pemberian nilai:

- Pemberian nilai langsung atau disebut sebagai ***assignment***
  - Contoh: **A = 10**
- Dibaca dari piranti masukan (perintah input)
  - Contoh: **A = input()**

Assignment

- **Assignment**: Pemberian nilai suatu variabel
- Ruas kiri harus **variable**
- Ruas kanan harus **ekspresi/nilai/variabel** yang sudah jelas nilainya

Python

```
<RuasKiri> = <RuasKanan>
```

Contoh:

```
i = 10
Nama = "Maya"
X = i + 10
```

Nilai X di-assign dengan ekspresi

Input/Output (1)

- Perintah **input**: pemberian nilai **variabel** dari piranti masukan, misal: keyboard dibaca atas masukan dari pengguna
- Perintah di Python: **input(**‘<perintah>’**)**

  <perintah> dapat diganti dengan kalimat pengantar input
- Contoh:

```
A = input()                              # A bertipe string
B = input('Masukkan angka =‘)            # B bertipe string
C = int(input())                         # C bertipe integer
D = float(input('Masukkan angka =‘))     # D bertipe float
```

Type checking: memastikan nilai yang dimasukkan dalam type yang tepat (gunakan type conversion)

Type Conversion

Beberapa fungsi *type conversion* yang penting diketahui:

No.	Function & Description
1	**int(x)** Mengkonversi **x** menjadi integer
2	**float(x)** Mengkonversi **x** menjadi nilai floating point (real)
3	**str(x)** Mengkonversi objek **x** menjadi representasi stringnya
4	**chr(x)** Mengkonversi sebuah integer **x** menjadi character
5	**ord(x)** Mengkonversi sebuah character **x** menjadi nilai integernya

Input/Output (2)

Perintah **output**: penulisan nilai (variabel/konstanta/hasil ekspresi) ke piranti keluaran, misal: monitor

Perintah di python: **print**

Contoh:

```
print(A)                          # menulis isi variabel A ke layar
print('Hello')                    # menulis Hello ke layar
print(A * 4)                      # menulis hasil perkalian A*4
print("Hello World!" + str(a))    # menulis Hello World! <nilai a>
```

Mengkonversi nilai a (bertype lain) menjadi string
+ adalah operator konkatenasi string

Latihan

Tentukan untuk setiap baris (yang diberikan nomor dalam komentar) dari potongan program Python berikut, manakah yang merupakan assignment yang tepat.

Jika tidak tepat, berikan alasannya.

```
# Program Latihan
# Latihan type data dan assignment

# KAMUS
# IA : int
# FA, FB : float
# SA, SB : string
# BA : bool
# CA, CB : char

# ALGORITMA
IA = 10             # (1)
FA = 3.45           # (2)
FB = 4.567          # (3)
FB = IA             # (4)

SA = "ITBJAYA"      # (5)
SA = SB             # (6)

CA = 'C'            # (7)
CA = "MAJUTERUS"    # (8)

BA = True           # (9)
BA = "#"            # (10)
```

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), +5, -10

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
+	**Penjumlahan**: menambahkan nilai kedua operan	int, float	a + b = 31
-	**Pengurangan**: mengurangi nilai operan kiri dengan nilai operan kanan	int, float	a – b = -11
*	**Perkalian**: mengalikan nilai kedua operan	int, float	a * b = 210
//	**Pembagian bulat**: Jika operan adalah int, maka hasil operasi adalah pembagian bulat	int	b // a = 2
/	**Pembagian riil**: Jika operan adalah float, maka hasil operasi adalah pembagian bilangan float	int, float	b / a = 2.1
%	**Modulo**: sisa hasil pembagian bulat	int	b % a = 1
**	**Pangkat**: memangkatkan 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

- Merupakan bagian dari program berbentuk 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 Π r
  - r adalah panjang jari-jari

Contoh-1: Pseudocode + Flowchart

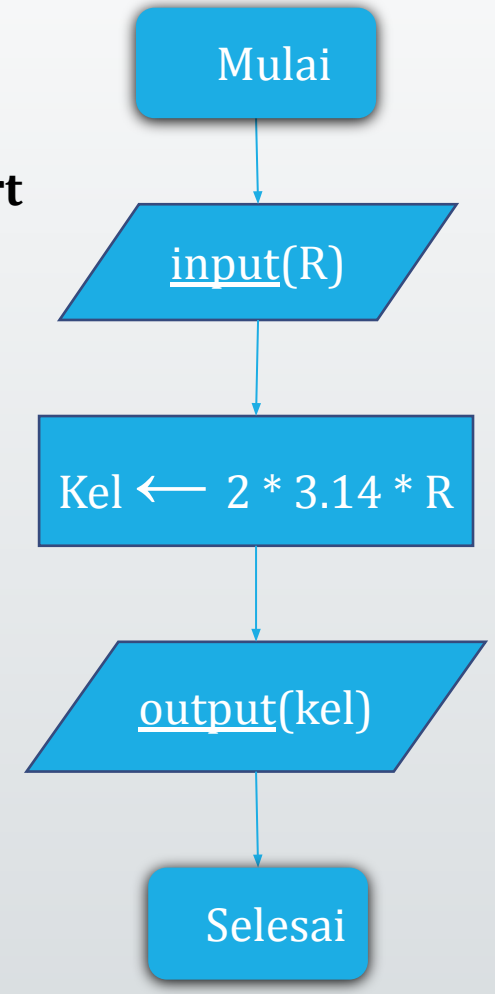
Pseudocode

```
#Program KelilingLingkaran
#Mengitung keliling roda

#KAMUS
R, Kel: Float

#ALGORITMA
input(R)
Kel ⟵ 2 * 3.14 * R
output(Kel)
```

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

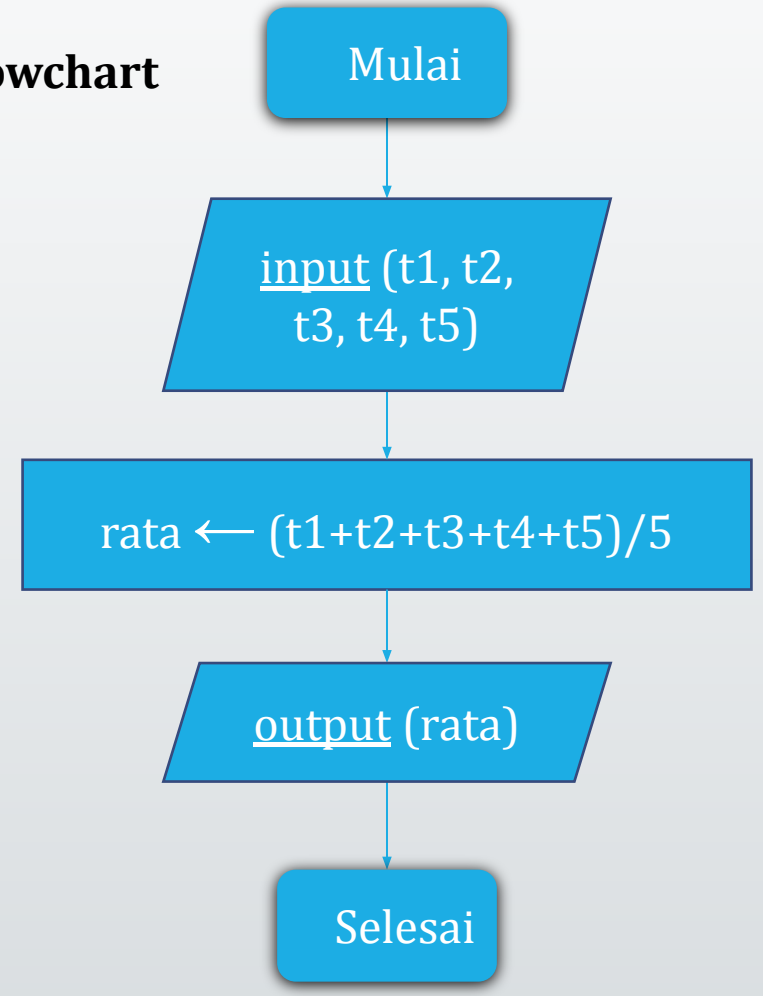
Pseudocode

```
#Program MenghitungRata2
#Menghitung rata2 tinggi 5 anak

#KAMUS
#t1,t2,...,: float

#ALGORITMA
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: luas = ½ * alas * tinggi

Latihan-3. Toko Kelereng

- Sebuah toko menjual kelereng. Tabel di samping adalah tabel harga kelereng berdasarkan warnanya.
- Seorang anak membeli kelereng sejumlah m kelereng merah, h kelereng hijau, dan k kelereng kuning. Asumsikan m >= 0, h >= 0, k >= 0.
- Hitunglah berapa yang harus dibayarkan anak itu.

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