

Struktur Dasar Program Prosedural – Python

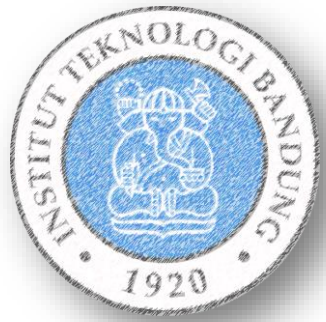
Tim Penyusun Materi Pengenalan Komputasi
Institut Teknologi Bandung © 2019





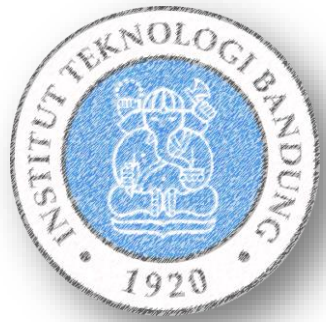
Tujuan

- 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



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

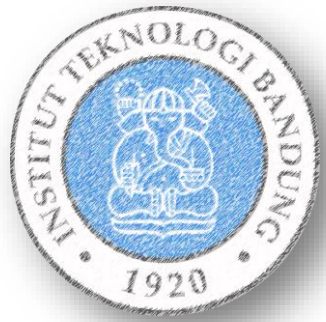


Struktur Dasar Algoritma

Program <JudulProgram>
{ Spesifikasi Program }

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

ALGORITMA
{ Deretan langkah algoritmik untuk penyelesaian
persoalan }
{ 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
```

Program Pertama

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

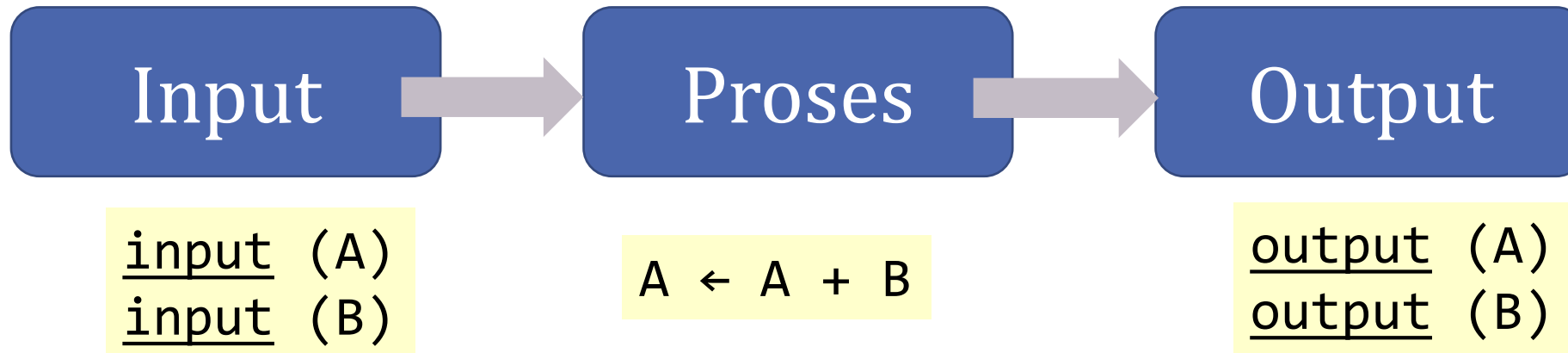
print adalah perintah untuk mencetak teks ke layar/monitor

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

# KAMUS
# belum diperlukan

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

Input – Proses – Output



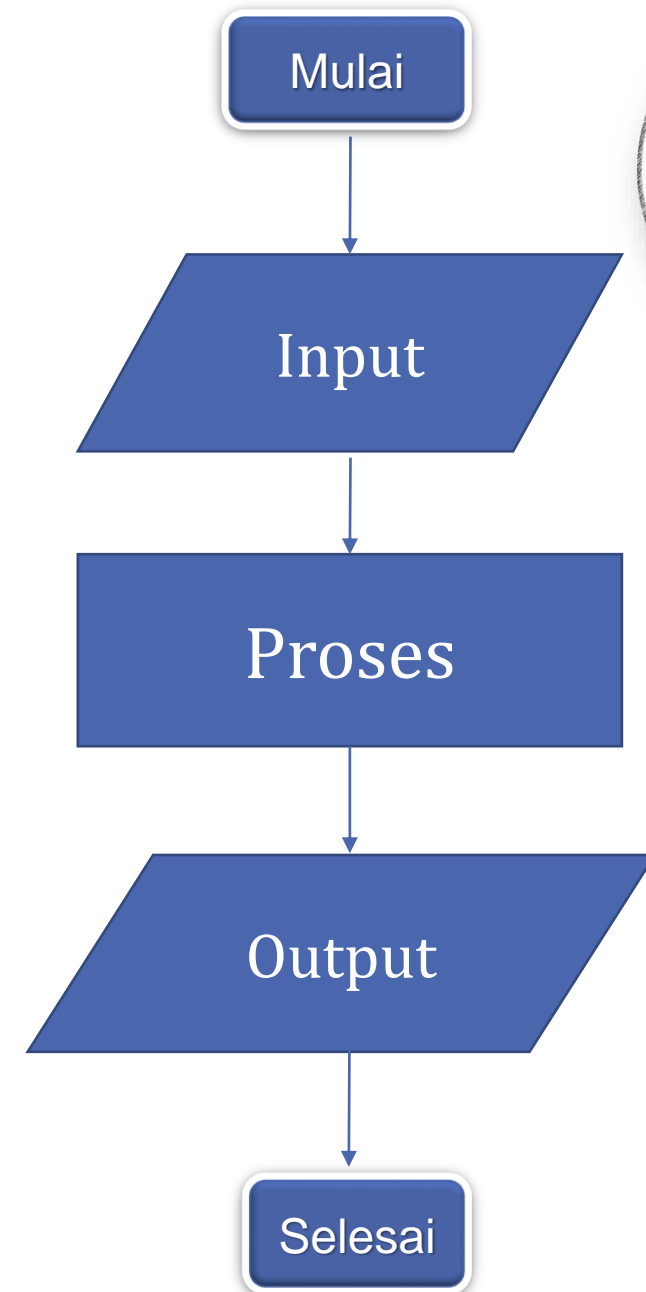
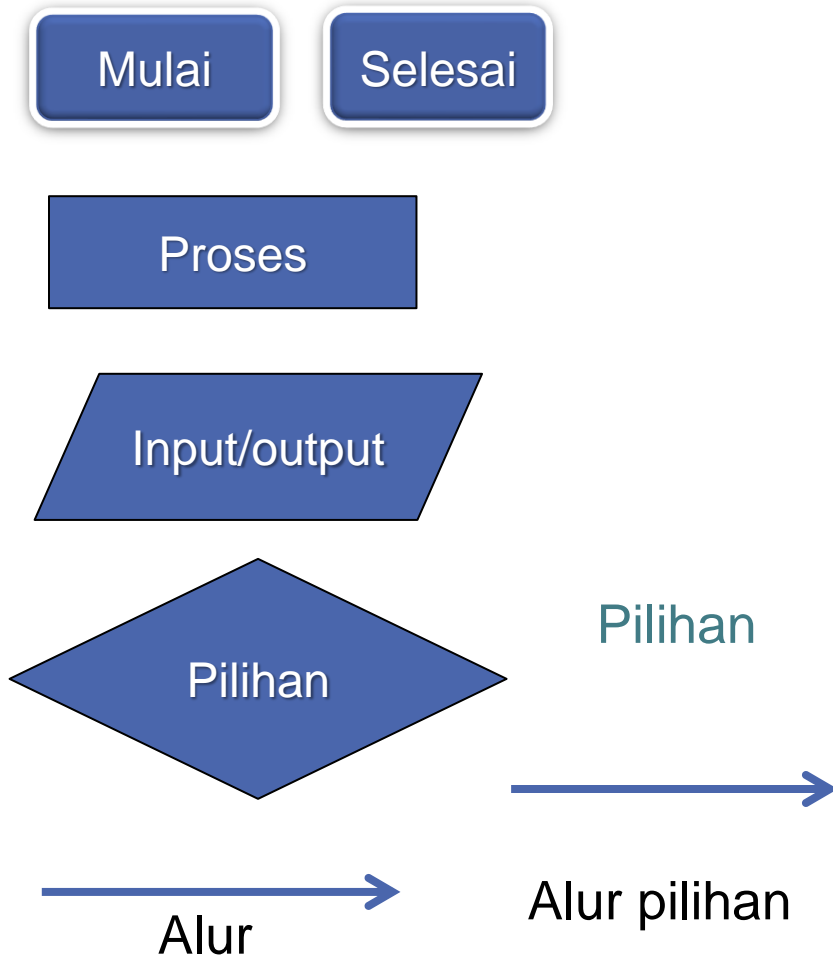
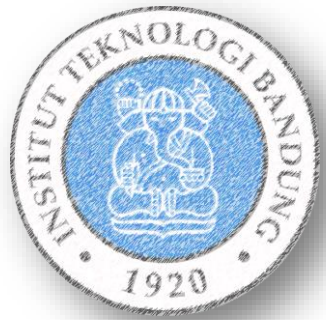
Python

```
A = int(input(''))  
B = int(input(''))
```

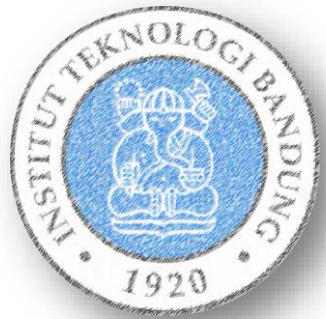
```
A = A + B
```

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

Flow Chart



Struktur Dasar Program



Program Test

{ Spesifikasi Program: menghitung $A + B$ }

KAMUS

{ Deklarasi variabel }

$A, B : \text{integer}$

ALGORITMA - Notasi Algoritmik

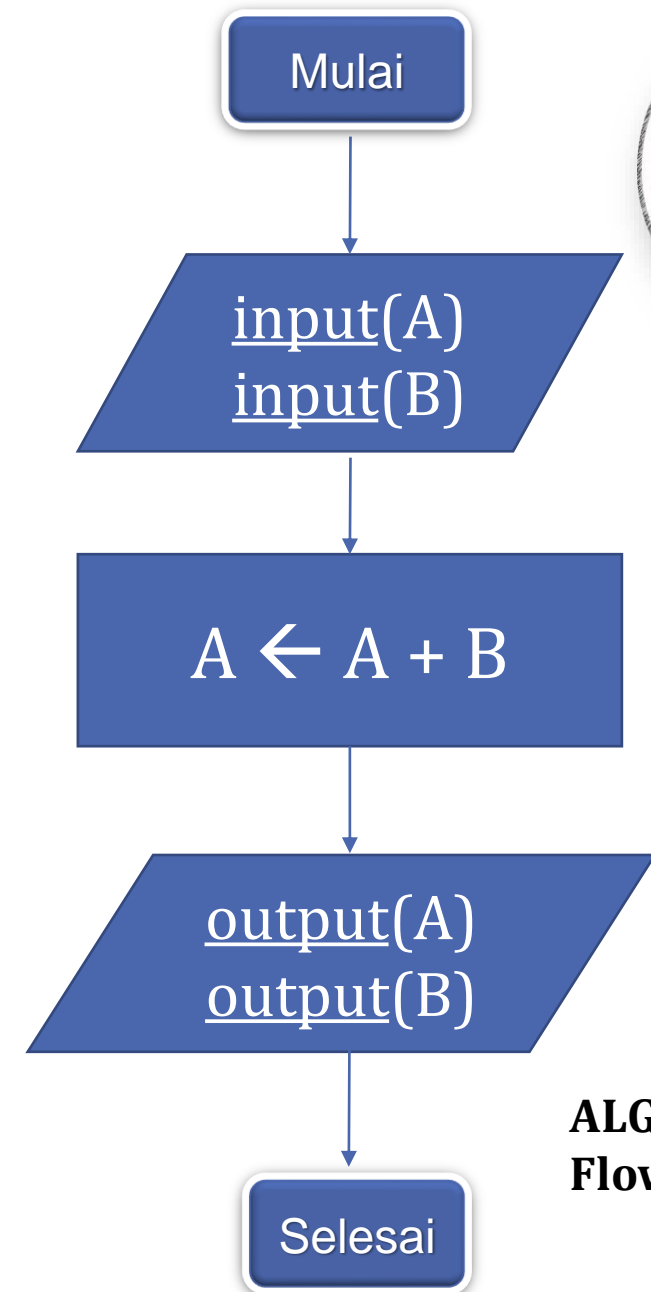
input(A)

input(B)

$A \leftarrow A + B$

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())

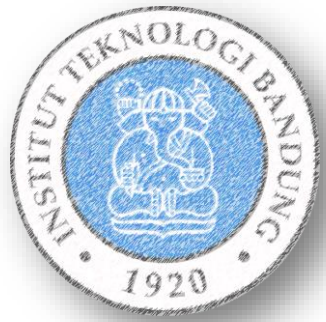
A = A + B        # proses

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

Judul Program + spesifikasi, dituliskan dalam komentar

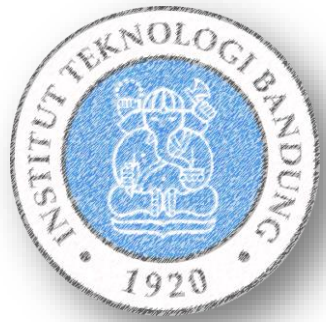
KAMUS: deklarasi variabel A dan B (dalam komentar)

ALGORITMA: Input, Proses, Output



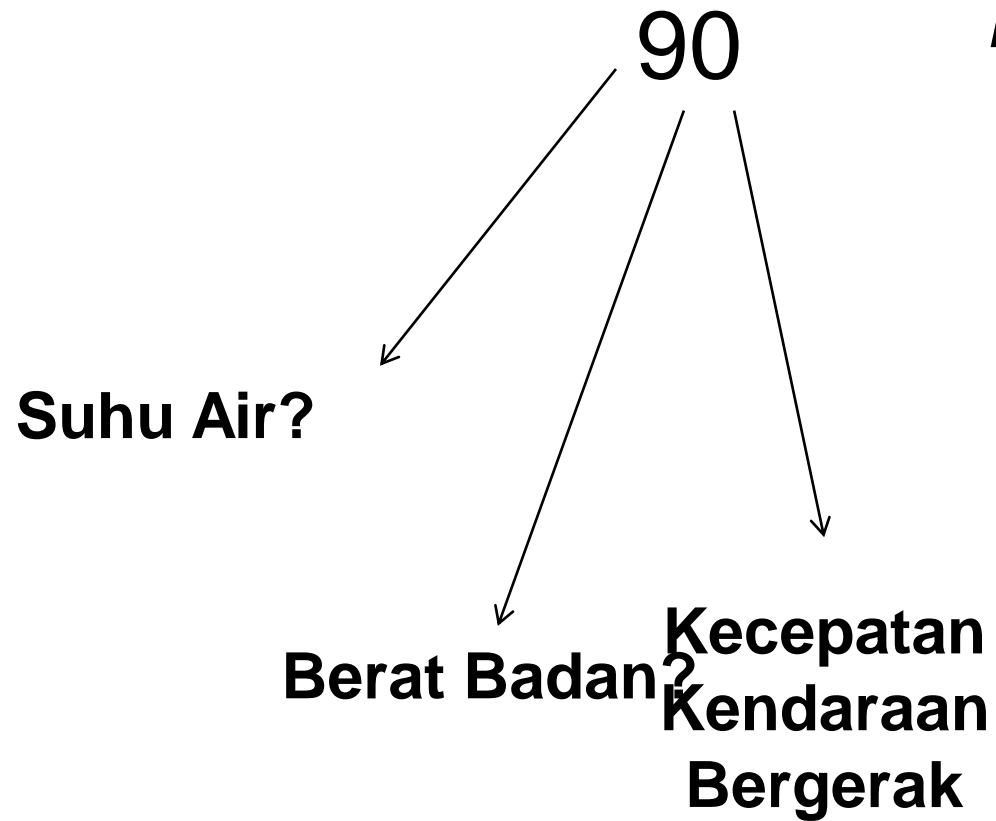
Komentar

- 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



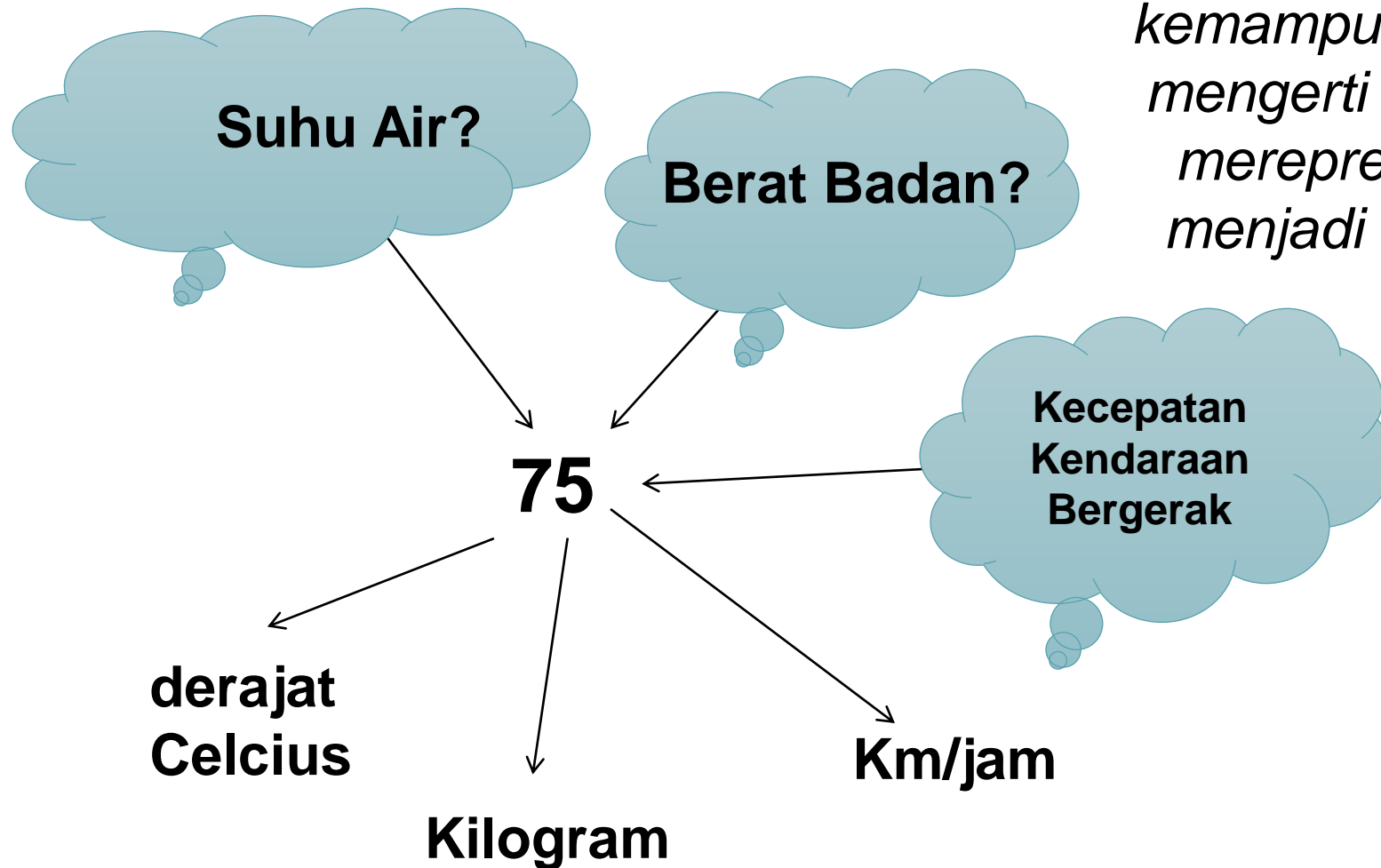
Data

Abstraksi Data



*kemampuan kita untuk
menginterpretasikan
suatu data dengan
konteks masalahnya*

Persoalan Abstraksi Data



*kemampuan kita untuk
mengerti konteks dan
merepresentasikan
menjadi bentuk lain.*

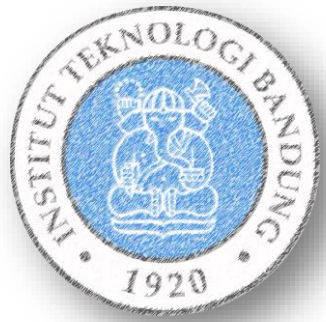


Bagian Kamus

- 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

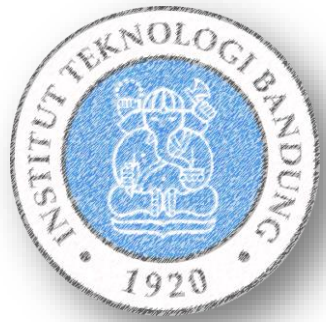
Tipe Data Dasar/Primitif

- Disediakan oleh bahasa pemrograman

Python	Domain Nilai
Bool	Nilai boolean: True ; False
Numbers	Nilai-nilai numerik. Jenis nilai numerik: <ul style="list-style-type: none">• 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)

- **Variabel** 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
- Penggunaan variabel:
 - deklarasi (supaya nama dikenal dan diketahui tipe datanya),
 - inisialisasi dan manipulasi nilai

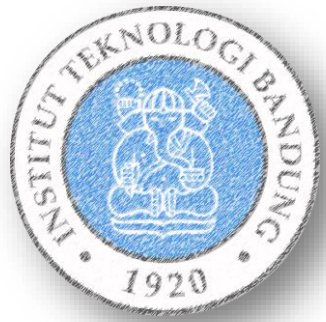
Variabel (2)

- Contoh deklarasi dan inisialisasi variabel:

Python

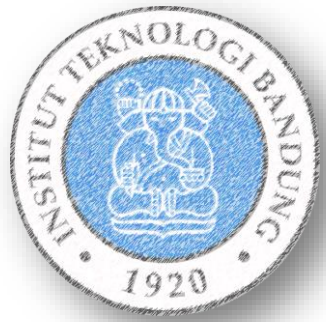
```
# KAMUS
# i : int
# A : int

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

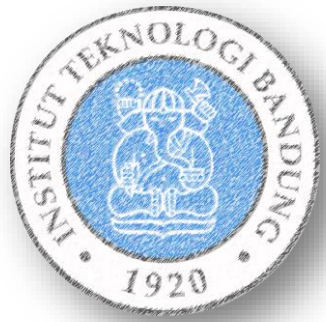


Membuat Nama Variabel yang Benar dan “baik”

- 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, bertipe integer. Jangan gunakan untuk data bertipe 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 (bertipe 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)
```