

# *Introduction to Python and Programming*

## *Basic Programming 1 (Condition)*

*Learning Progress Review Week 5*

By:  
**MARVEL TEAM**

Fikrie | Natalia | Satria

## Materi Week 5

- ☐ Apa itu Python?
- ☐ Kenapa kita menggunakan Python?
- ☐ Penggunaan Python pada Data Science
- ☐ Python vs R
- ☐ Jupyter Notebook (Google Colab) Introduction
- ☐ Tipe data Python
- ☐ Arithmetic Programming
- ☐ Pengenalan Control Flow dan Conditional
- ☐ Flowchart of conditions
- ☐ If statement
- ☐ If – elif – else statement
- ☐ Nested if

# **1.** *Introduction to Python and Programming*

Python termasuk *high-level programming language*

# Apa itu Python?



**Python** adalah bahasa yang ditujukan untuk *General-Purpose Programming* dan termasuk dalam kategori *high-level programming language*.

**Python** merupakan bahasa pemrograman tingkat tinggi yang di ciptakan oleh Guido Van Rossum.

**Python** banyak digunakan untuk membuat berbagai macam program, seperti: Program CLI, GUI, Aplikasi Mobilie, Web, IOT, Game, dsb.

# Kenapa kita menggunakan Python?

## **Mudah di pelajari dan di aplikasikan**

Python merupakan Bahasa pemrograman yang mudah dipelajari. Karena python memiliki struktur yang sederhana serta keyword yang sedikit.

## **Memiliki banyak *Library***

*Library* merupakan istilah untuk kode tambahan yang disediakan python dengan tujuannya masing-masing. *Library* mempermudah kita dalam membuat program tanpa harus menuliskan kode mulai dari dasar.

## **Bersifat *open source***

Python merupakan salah satu Bahasa pemrograman yang bersifat open source. Artinya siapa saja bisa menciptakan dan mengembangkan sebuah program menggunakan python

# Penggunaan Python pada *Data Science*

## **Data Collection dan Cleansing**

Impor data mentah dengan format csv ke Python memakai library yang ada.

## **Data Exploration**

Lakukan eksplorasi data dan pre-processing memakai library numpy dan sklearn. Disini Python akan mulai memproses 'melihat' dan serta memanipulasi data.

## **Data Visualisation dan Interpretation**

Visualisasi persebaran data dapat dilakukan memakai library matplotlib.

## **Data Modelling**

Buat model data mining memakai fitur machine learning. Evaluasi model yang sudah dibuat. Evaluasi yang dilakukan seperti mengecek akurasi, presisi, sensitivitas dan spesifitas dari model.

## **Deploying**

Merepresentasikan hasil model ke dalam beragam bentuk, seperti web atau plot memakai library khusus seperti graphviz.

# Python vs R

Mungkin banyak pertanyaan di analisis data, apakah R atau Python yang paling baik digunakan?.

- R diciptakan oleh Ross Ihaka dan Robert Gentleman pada tahun 1995 sebagai implementasi dari Bahasa program S. Tujuannya adalah untuk mengembangkan Bahasa yang fokus pada analisis data, statistic, dan model grafis.
- Bahasa R baik dalam visualisasi data karena ada banyak sekali *package* yang mendukung R untuk membangun visualisasi menarik. R dibangun oleh praktisi statistic, sehingga siapapun yang tidak memiliki keahlian programming dalam dengan mudah beradaptasi dengan R.
- Python digunakan pada pekerjaan data analisis harus disambungkan dengan aplikasi web atau pembuatan model machine learning tingkat lanjut.
- Merupakan Bahasa general yang mudah dipelajari dan dimengerti oleh banyak developer. Oleh karena itu Python bisa digunakan untuk keperluan yang lebih luas seperti visualisasi data, machine dan deep learning, dan masih banyak lagi.

# Jupyter Notebook vs Google Colab

**Jupyter Notebook** adalah aplikasi berbasis *web open source* yang dapat digunakan untuk memuat dan membagikan dokumen. Pada penggunaan Jupyter Notebook kita tidak perlu menggunakan internet, dikarenakan Notebook berjalan di Localhost pc pribadi, kelemahan yang di dapatkan menggunakan Jupyter adalah *Install Package* dan *Library* sendiri.

**Google colab** memungkinkan Anda menggabungkan kode yang dapat dijalankan dan *rich text* dalam satu dokumen, beserta gambar, HTML, LaTeX, dan lainnya.

**Google Colab** dapat berkolaborasi dengan pengguna lainnya melalui berbagi *coding* secara *online*. Adapun kelebihan yang didapatkan adalah tak perlu *install package* atau *library* yang diperlukan, karena sudah di sediakan oleh Google Colab. Dan kelemahan nya yaitu harus menggunakan Internet di setiap penggunaanya.



# Type Data Python

Data	Tipe Data
str	Teks
int, float, complex	Angka (int: bulat; float : desimal)
list, tuple, ranges	Sequence
dict	Mapping
set, frozenset	Set
bool	Boolean (True or False)
bytes, bytearray, memoryview	Binary

# Arithmetic Programming

Operator	Fungsi	Contoh: x = 5 dan y = 2
Penjumlahan (+)	Menambahkan nilai	$x + y = 7$
Pengurangan (-)	Mengurangkan nilai	$x - y = 3$
Perkalian (*)	Mengalikan nilai	$x * y = 10$
Pembagian (/ atau //)	/ hasil pembagian float // hasil pembagian integer	$x / y = 2.5$ $x // y = 2$
Modulus (%)	Sisa pembagian	$x \% y = 1$
Pangkat (**)	Perhitungan pangkat	$5 ** 2 = 25$

## 2. Basic Programming I : Conditions

*Conditions* dalam Python menggunakan *if statement*.

# Control Flow

**Control flow** adalah suatu urutan program fungsi perintah, instruksi dan *statement* yang dieksekusi oleh komputer ketika program itu dijalankan.

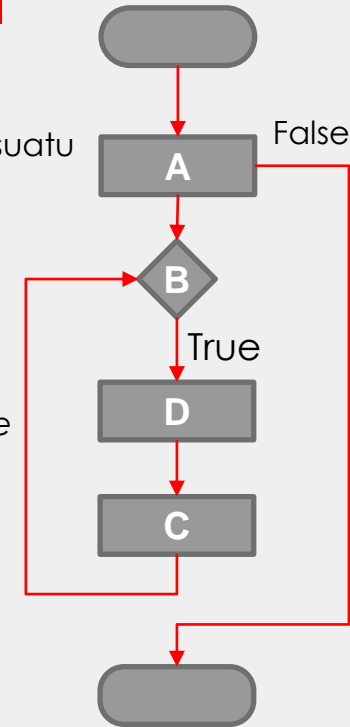
Ada beberapa tipe sistem *control flow* dalam *programming*. Tetapi, tipe yang paling dikenal adalah :

- Conditions
- Loops
- Functions

## Flow Chart

**A.** Ada suatu kondisi

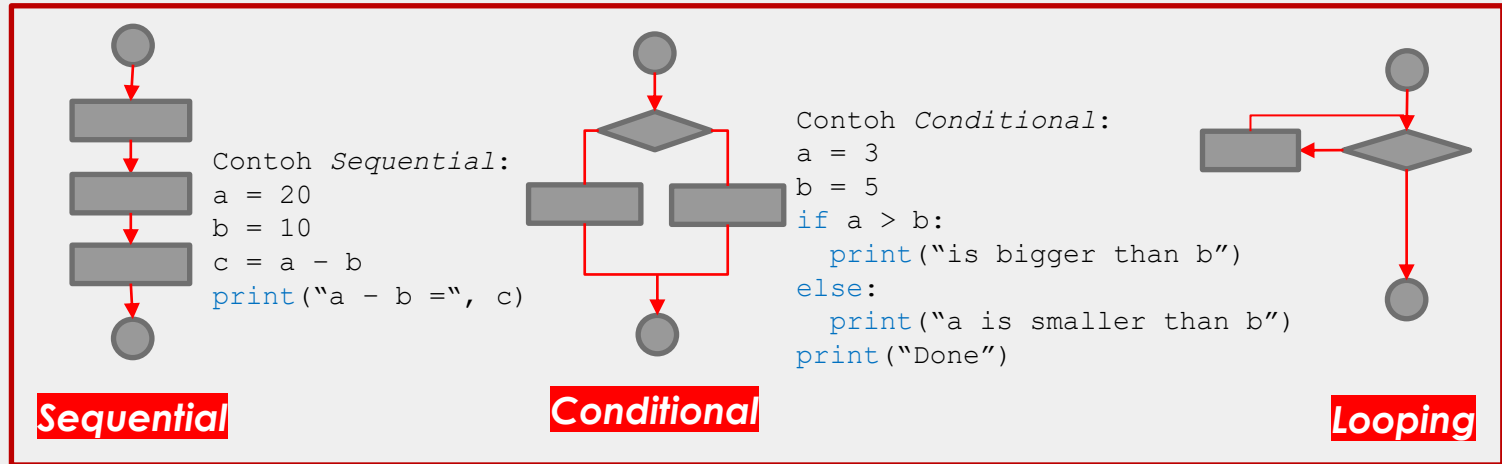
**B.** Masuk *conditional statement* untuk diuji apakah *True* atau *False*



# Control Flow

**Control flow** dalam program Python diatur oleh *conditional statemens*, *loops*, dan *function calls*.

Secara umum, Python memiliki tiga tipe struktur control, yaitu :



# Conditional

**Conditional** adalah ekspresi yang mengevaluasi suatu *statement* yang kita buat di Python dan menghasilkan *True* atau *False*.

Tipe datanya adalah *Boolean / Bool* dimana nilai suatu data ada dua kemungkinan, *True* atau *False*.

# Conditional

**Conditions dalam Python** menggunakan **if statement**.

Python mendukung **comparison operators** dari matematika, antara lain :

- Sama dengan : **a == b**
- Tidak sama dengan : **a != b**
- Kurang dari : **a < b** atau kurang dari sama dengan : **a <= b**
- Lebih dari : **a > b** atau lebih dari sama dengan : **a >= b**

Dan juga, ada **logical operators** yang dilakukan untuk menggabungkan dua nilai *comparison operators*, antara lain :

- *True* jika kedua ekspresi adalah benar: **and**
- *True* jika salah satu ekspresi adalah benar: **or**
- *True* jika ekspresi salah (melengkapi ekspresi yang satunya) : **not**

## *If Statement*

**If Statement** membuat suatu sistem urutan *decision making*.

**If Statement** Mengandung suatu urutan kode yang akan dijalankan ketika suatu kondisi yang diberikan bersifat *True*.

**If Statement** Jika kondisi *False*, maka urutan kode yang akan dijalankan adalah pada kondisi yang bersifat *False*.



# If Statement

```
[1] a = 7
    b = 9
    if a > b:
        print('a bigger than b')
    else:
        print('a smaller than b')
```

Diagram illustrating the execution of the first code block:

- `if a > b:` is labeled as **Ekspresi 1** (Expression 1).
- `print('a bigger than b')` is labeled as **Statement 1**.
- `else:` is labeled as **Ekspresi 2** (Expression 2).
- `print('a smaller than b')` is labeled as **Statement 2**.
- The condition `a > b` is evaluated as **Kondisi 1** (Condition 1).
- The condition `a < b` is evaluated as **Kondisi 2** (Condition 2).

The output of the code is: `a smaller than b`.

```
[4] Apple = 'Fruit'
    if Apple == 'Fruit':
        print('Buah')
    else:
        print('Gadget')
```

Diagram illustrating the execution of the second code block:

- `if Apple == 'Fruit':` is labeled as **Ekspresi 1** (Expression 1).
- `print('Buah')` is labeled as **Statement 1**.
- `else:` is labeled as **Ekspresi 2** (Expression 2).
- `print('Gadget')` is labeled as **Statement 2**.
- The condition `Apple == 'Fruit'` is evaluated as **Kondisi 1** (Condition 1).
- The condition `Apple != 'Fruit'` is evaluated as **Kondisi 2** (Condition 2).

The output of the code is: `Buah`.

# If Statement

```
[6] A = 1
    B = 2
    C = 3
    if C > A and A + B == C:
        print(A + B + C)
```

6

Ekspresi

Statement

```
[9] x = 4
    if x % 2 == 0:
        print('x is even number')
    else:
        print('x is odd number')
```

x is even number

Ekspresi 1

Statement 1

Kondisi 1

Ekspresi 2

Statement 2

Kondisi 2

## *If Statement (Single, Two and Multiple Condition)*

```
[10] a = 7  
     if a == 7:  
         print('a is seven')
```

a is seven

**Single Condition**

} 1 Kondisi

**Two Condition**

```
[11] x = 14  
     y = 21  
     if x != y:  
         print('x dan y tidak sama')  
     else:  
         print('x dan y sama')
```

x dan y tidak sama

} Kondisi 1  
}  
} Kondisi 2

# *If Statement (Single, Two and Multiple Condition)*

## **Multiple Condition**

```
[12] a = 9
      b = 11
      if a > b:
          print('a lebih besar dari b')
      elif a < b:
          print('a lebih kecil dari b')
      else:
          print('a dan b bernilai sama')
```

} Kondisi 1

} Kondisi 2

} Kondisi 3

a lebih kecil dari b

## *If Statement (Single, Two and Multiple Condition)*

```
[13] a = 13
      b = 17
      if a <= b:
          print('a lebih kecil dari b')
```

1 ekspresi

a lebih kecil dari b

```
[14] a = 23
      b = 27
      c = 33
      if a < b and c > b:
          print('c adalah angka paling besar')
```

'and' menunjukkan 2 ekspresi

c adalah angka paling besar

# If-Elif-Else Statement

**If – Elif – Else Statement** digunakan pada kasus yang memiliki banyak kondisi (multiple condition)

```
[15] a = 15
      b = 15
      if a > b:
          print('a lebih besar dari b')
      elif a < b:
          print('a lebih kecil dari b')
      else:
          print('a dan b bernilai sama')
```

Kondisi 1

Kondisi 2

Kondisi 3

a dan b bernilai sama

# Nested If

**Nested if** digunakan pada kasus dimana terdapat kondisi di dalam kondisi. Jika kondisi 1 terpenuhi, maka kondisi a dan b akan berjalan. Jika tidak, maka akan langsung berjalan pada kondisi 2.

```
[25] x = 20
      if x > 17:
          print('x lebih dari 17')
          if x > 20:
              print('x lebih besar dari 20')
          else:
              print('x sama dengan 20')
      else:
          print('kurang dari 20')
```

} Kondisi 1  
} Kondisi a  
} Kondisi b  
} Kondisi 2

```
➤ x lebih dari 17
  x sama dengan 20
```

Special Thanks to :



Slide template by SlideCarnival