

Programming I:

Introduction to Python with
Google Colab Environment



Hello, everyone!



Sebelum kita memulai kelas, kita awali dengan:

1. Berdoa
2. siapkan diri



Pertemuan lalu...



Let's Recall

pertemuan sebelumnya kita sudah membahas lanjutan command SQL. Ada yang inget apa aja?



Target



1. Introduction to Condition and Control Flow
2. Flowchart of Conditions
3. If statement
4. Nested if
5. Switch-case
6. Introduction to Python
7. Structure Data in Python



Tools



<https://colab.research.google.com/>

1. Introduction to Condition and Control Flow

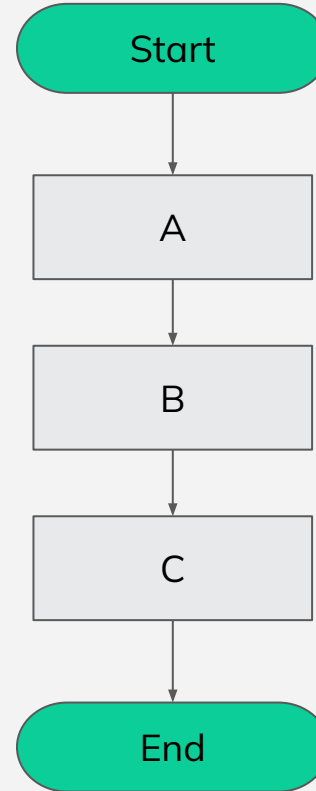
control flow di python seperti apa ya?

Misalnya kita akan memasak. sebelum memasak pasti kita menyiapkan keperluan untuk memasak itu. Tapi sebelum memasak pasti kita mikir mau masak apa, apa saja yang akan dibeli.

Nah alur pengambilan keputusan itu mirip dengan konsep flow control.



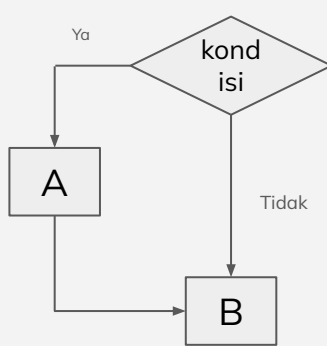
Jadi control flow adalah sebuah metode yang mengontrol alur kerja dari sebuah program.
Berfungsi untuk mengontrol keputusan apa saja yang akan diambil ketika sedang membuat sebuah program.



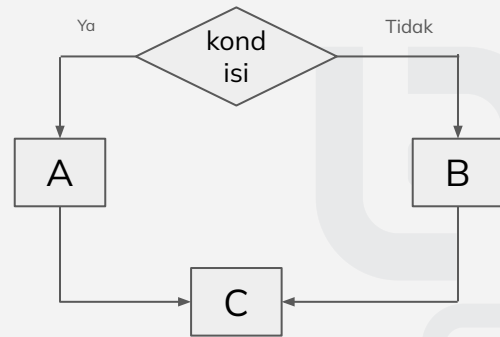
Jenis control flow di Python

1. IF ... ELSE statement
2. While Loop statement
3. For Loop Statement
4. Break statement
5. Continu statement
6. Pass statement

2. Flowchart of Conditions



Tunggal

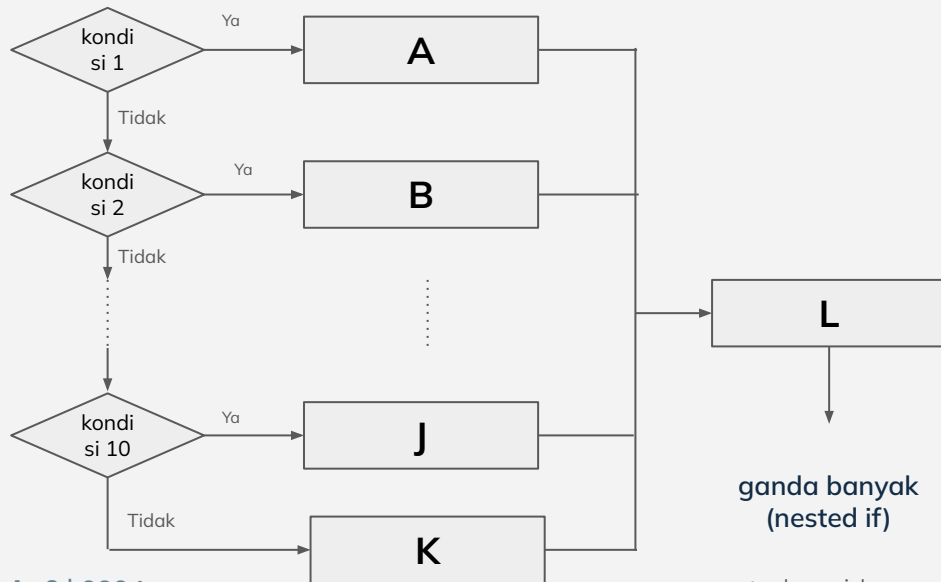


Ganda

Urutan langkah penyelesaian secara
beurutan dengan adanya pengendalian
berdasarkan persyaratan tertentu.

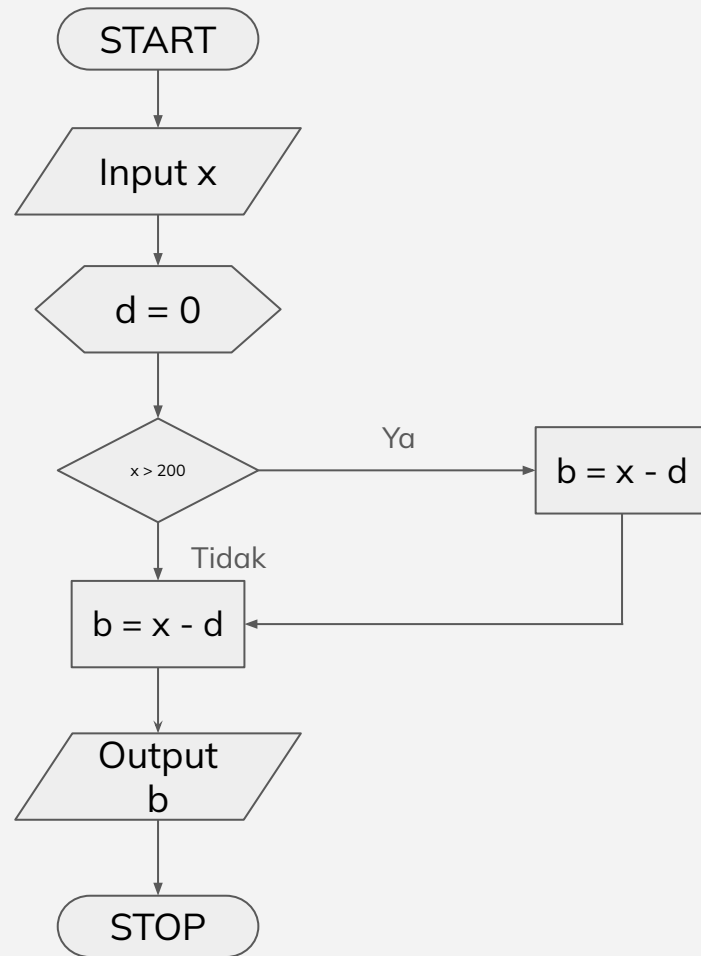
Jenisnya:

1. Tunggal
2. Ganda
3. ganda banyak (nested if)



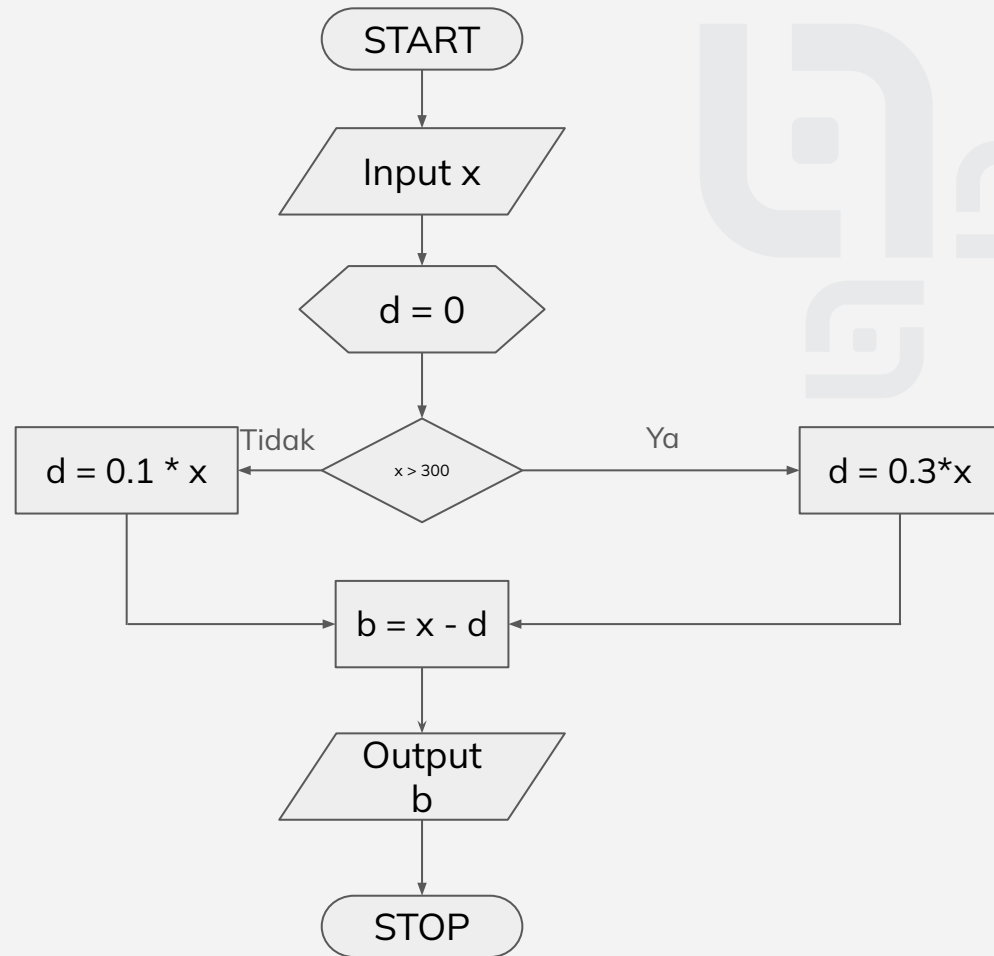
Tunggal

1. input x
2. $d = 0$
3. if $x > 0$ then $d = 0.2 * x$
4. $b = x - d$
5. output b
6. stop



Ganda

1. input x
2. if $x > 300$ then
3. $d = 0.3 * x$
4. ELSE IF $x > 300$ then
5. $d = 0.1 * x$
6. ENDIF
7. $b = x - d$
8. STOP



Ganda Banyak (nested if)

1. input x
2. if x > 500 then
3. d = 0.5*x
4. ELSE IF x > 300 then
5. d = 0.3*x
6. ELSE
7. d = 0
8. ENDIF
9. b = x - d
10. STOP

3. IF statement

IF ... ELSE statement

- control flow ini digunakan ketika terdapat kondisi tertentu pada suatu kejadian/ketentuan. Output yang akan dikeluarkan sesuai dengan aturan kejadian suatu kondisi.
- IF : Jika/untuk boleh
- ELIF : untuk tidak
- ELSE : mengulang/lainnya

IF statement Python

- Pada contoh di atas, kondisi pengujiannya adalah $\text{num} > 0$.
- Blok kode if dijalankan hanya jika kondisi pengujian bernilai True.
- Ketika variabel num sama dengan 5, kondisi pengujian benar dan pernyataan di dalamnya jika blok kode dijalankan.
- Ketika variabel num sama dengan -8, kondisi pengujian adalah False dan pernyataan di dalamnya jika blok kode dilewati.
- Pernyataan print() berada di luar blok if. Oleh karena itu, ini dijalankan terlepas dari ekspresi pengujiannya.

```
if test condition:  
    statement(s)
```

```
[1] num = 5  
    if num > 0:  
        print(num, "adalah bilangan positif.")  
    print("Default.")
```

5 adalah bilangan positif.
Default.

```
[2] num = -8  
    if num > 0:  
        print(num, "adalah bilangan positif.")  
    print("Default.")
```

Default.

IF ... ELSE Python

- Pada contoh di atas, ada 3 kondisi pengujian.
- Ketika variabel `var=300`, semua kondisi pengujian adalah False, pernyataan di dalam blok if/elif dilewati dan pernyataan di dalam blok kode else dijalankan.

```
▶ if test condition:  
    statement(s)  
elif test condition:  
    elif 1 statement(s)  
elif test condition:  
    elif 2 statement(s)  
else:  
    else statement(s)
```

```
[3] var = 300  
if var == 200:  
    print("condition 1 got true value")  
    print(var)  
elif var == 150:  
    print("condition 2 got true value")  
    print(var)  
elif var == 100:  
    print("condition 3 got true value")  
    print(var)  
else:  
    print("no condition satisfied")  
    print(var)
```

```
no condition satisfied  
300
```

4. Nested IF

Nested IF

Kita dapat memiliki pernyataan if...elif...else di dalam pernyataan if...elif...else lainnya. Ini disebut bersarang dalam pemrograman komputer. Sejumlah pernyataan ini dapat disarangkan satu sama lain. Dalam python nested if bisa dilakukan. Contoh sebagai berikut

▼ Nested IF

```
❏ if (condition1):  
    # Executes when condition1 is true  
    if (condition2):  
        # Executes when condition2 is true  
        # if Block is end here  
    # if Block is end here
```

```
❏ i = 30  
  
if (i == 30):  
    # First if statement  
    if (i < 20):  
        print ("i is smaller than 20")  
  
    # Nested - if statement  
    # Will only be executed if statement above  
    # it is true  
    if (i < 5):  
        print ("i is smaller than 5 too")  
    else:  
        print ("i is greater than 5 and smaller than 20")
```

i is greater than 5 and smaller than 20

5. Switch-case

Switch-case

- Switch case adalah metode kode membuat keputusan berdasarkan kondisi tertentu.
- Python tidak memiliki konstruksi switch case bawaan. Sebagai gantinya, kita bisa meniru fungsi switch case menggunakan beberapa pendekatan.
- Dengan kata lain, kita masih bisa menerapkan konsep switch case untuk membuat kode yang lebih efisien dan terstruktur di Python.
- Pendekatan yang bisa dilakukan untuk mengimplementasikan konsep switch case di Python:
 - if-elif-else
 - dictionary
 - class dan metode

Dictionary untuk mengimplementasikan Switch-case

- Dictionary di Python dapat mengimplementasikan switch case. Dictionary, kita menetapkan kunci dan nilai yang berhubungan dengan kondisi serta tindakan yang sesuai.

▼ Dictionary untuk implementasi switch case

```
✓ [13] def Januari():  
      return "Januari"  
      def Februari():  
          return "februari"  
      def maret():  
          return "maret"  
      def april():  
          return "april"  
      def mey():  
          return "mey"  
      def juni():  
          return "juni"  
      def default():  
          return "H2"  
  
      switcher = {  
          1: Januari,  
          2: Februari,  
          3: maret,  
          4: april,  
          5: mey,  
          6: juni  
      }  
  
      def switch(h1):  
          return switcher.get(h1, default)()  
  
      print(switch(1))  
      print(switch(2))  
      print(switch(3))  
      print(switch(4))  
      print(switch(5))  
      print(switch(6))  
      print(switch(7))  
      print(switch(8))  
  
      Januari  
      Februari  
      maret  
      april  
      mey  
      juni  
      H2  
      H2
```

IF-ELIF-ELSE untuk mengimplementasikan Switch-case

Struktur if-elif-else di Python dapat digunakan sebagai pengganti switch case. Fungsinya mirip dengan switch case, yaitu kita bisa mengevaluasi beberapa kondisi dan melakukan tindakan berbeda berdasarkan kondisi tersebut.

✓ IF-ELIF-ELSE untuk implementasi switch case

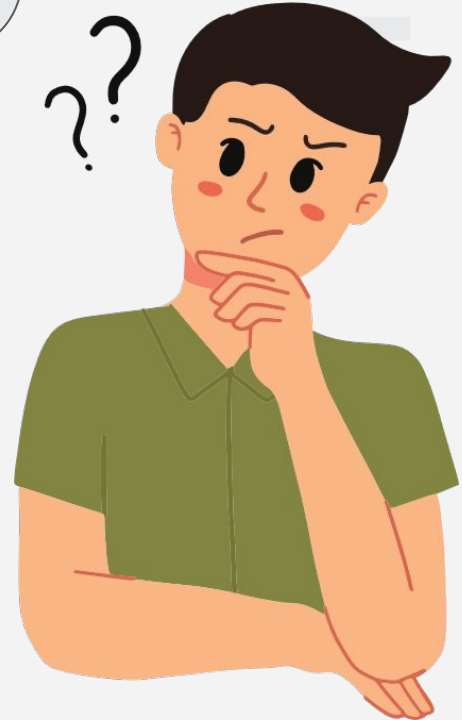
```
▶ bulan = 7
if bulan == 1:
    print("Januari")
elif bulan == 2:
    print("februari")
elif bulan == 3:
    print("maret")
elif bulan == 4:
    print("april")
elif bulan == 5:
    print("mei")
elif bulan == 6:
    print("juni")
else:
    print("h2")
```

h2

6. Introduction to Python

Apaitu Python?

- Python adalah bahasa pemrograman. Salah satu fungsinya untuk menganalisis data dan melakukan modelling.
- python cukup populer di kalangan data scientist. karena dapat melakukan analisis statistik biasa sampai yang paling rumit seperti modelling data
- Python bisa digunakan untuk machine learning, membangun situs web, sampai pengembangan aplikasi.
- Python punya sintaks yang gampang dipahami, dibaca dan ditulis
- Python punya perpustakaan (library)
- python bisa diinstall di berbagai macam sistem operasi (operation system/OS) komputer seperti microsoft, linux, macOS



Library Python

- libraries pada python menyimpan kode-kode yang dapat diambil dan digunakan
- libraries berisi kumpulan modul dan kode yang bisa kamu gunakan berulang-ulang untuk berbagai kebutuhan
- libraries python memudahkan mengambil kode tertentu yang kita butuhkan.
- Kita akan bahas detail di programming III



PIP

- Dapat dikatakan sebagai asisten pribadi python
- menginstall, menghapus, maupun memperbaharui package python



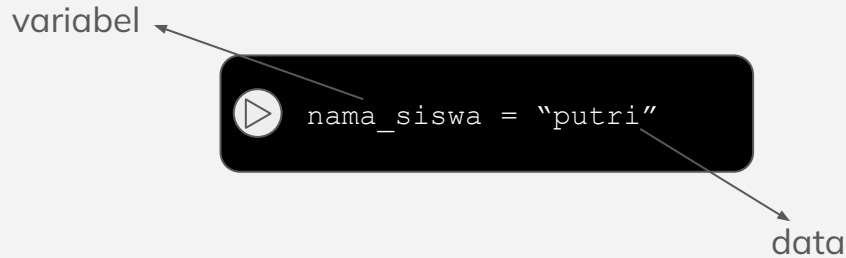
7. Structure Data in Python

Atribut Data

Atribut data-data di python dibagi menjadi dua, yaitu:

- Variabel: Variabel adalah wadah yang menyimpan suatu informasi. Bisa berupa angka, string teks, atau bahkan struktur data yang kompleks.
- tipe data: Tipe data adalah suatu media atau memori pada komputer yang digunakan untuk menampung informasi.

Variabel



- kita masukkan ke python variabel data “nama_siswa” dan datanya adalah “putri”, maka kita bisa tulis di python seperti di gambar atas.
- Jika penamaan variabel lebih dari 1 suku kata maka dipisah dengan underscore atau garis bawah

Tipe Data

Tipe Data	Contoh	Penjelasan
Boolean	<code>True</code> atau <code>False</code>	Menjelaskan benar <code>True</code> yang bernilai 1 atau <code>False</code> bernilai 0
String	"apel"	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda <code>"</code> atau <code>'</code>)
Integer	17 atau 1996	Menyatakan bilangan bulat
Float	1.75	Menyatakan bilangan yang mempunyai koma
Hexadecimal	8a	menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	1+3j	Menyatakan pasangan angka real dan imajiner
List	[<code>'abc'</code> , 891, 2.34]	data untaian yang penyimpanan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	(<code>'abc'</code> , 891, 2.34)	data untaian yang penyimpanan berbagai tipe data dan isinya tidak bisa diubah-ubah
Dictionary	{ <code>'nama'</code> : "tasya", <code>'id'</code> : 1}	data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Tipe Data List

- List adalah kumpulan data yang sifatnya terurut
- List bisa berupa tipe data yang berbeda-beda
- List bisa menambahkan data baru ke dalam list tersebut.
- List bisa mengubah isi datanya tapi tidak bisa merubah urutannya
- harus menulis menggunakan simbol kurung siku [], di awal dan di akhir data
- Ketika membentuk list di dalam python, ada aturan penulisannya. Kita harus menulis menggunakan simbol kurung siku [], di awal dan di akhir data. Seperti yang ada di gambar samping

```
✓ 0d [7] [50, 130, 300]
      [50, 130, 300]

✓ 0d [8] ['halo', 1, 2.3, [10,6]]
      ['halo', 1, 2.3, [10, 6]]

✓ 0d [9] data_saya = ['a','b','c']

✓ 0d [10] data_saya.append('d')

✓ 0d [11] data_saya
      ['a', 'b', 'c', 'd']

✓ 0d [12] data_saya[0]
      'a'

✓ 0d [13] data_saya[0] = 'cendol'

✓ 0d [14] data_saya[0]
      'cendol'
```

Tipe Data Tuple

- data yang sudah dimasukkan tidak bisa diubah (immutable)
- tuple dituliskan menggunakan kurung biasa ()

```
▼ Tuple

✓ [1] nama_ = ('naura', 'cindi', 'roy')

✓ [2] nama_
      ('naura', 'cindi', 'roy')

✗ [3] nama_.add('tika')
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-3-721ab8c80a45> in <cell line: 1>()
----> 1 nama_.add('tika')

AttributeError: 'tuple' object has no attribute 'add'

TELUSURI STACK OVERFLOW

✗ [4] nama_.append('tika')
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-4-2d5f7f5cf6aa> in <cell line: 1>()
----> 1 nama_.append('tika')

AttributeError: 'tuple' object has no attribute 'append'

TELUSURI STACK OVERFLOW

✓ [5] nama_[0]
      'naura'

✗ [6] nama_[0] = 'cinta'
-----
TypeError                                    Traceback (most recent call last)
<ipython-input-6-941558821129> in <cell line: 1>()
----> 1 nama_[0] = 'cinta'

TypeError: 'tuple' object does not support item assignment

TELUSURI STACK OVERFLOW
```

Type Data Dictionary

- tipe data yang memungkinkan untuk menyimpan pasangan key-value.
- Bisa menambah, mengubah, atau menghapus entri tanpa harus mengubah seluruh struktur data.

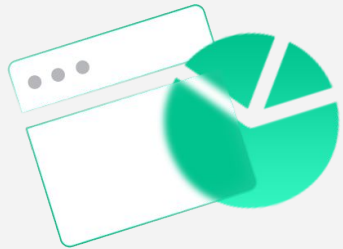
```
Dictionary
[15] user_ = {"nama": "Tika", "umur": 18, "pekerjaan": "Guru"}
[16] user_
{'nama': 'Tika', 'umur': 18, 'pekerjaan': 'Guru'}
[18] user_["nama"] = "Ticka"
[19] user_
{'nama': 'Ticka', 'umur': 18, 'pekerjaan': 'Guru'}
[20] user_["domisili"] = "Jogja"
[21] user_
{'nama': 'Ticka', 'umur': 18, 'pekerjaan': 'Guru', 'domisili': 'Jogja'}
```

Latihan

1. Buatlah struktur if statement dengan output Budi lulus karena nilainya 85
2. Buatlah dictionary bertemakan makanan

NEXT!!

Programming II:
Intermediate Python (loop, while,dll)



Terima Kasih

