



# PYTHON

KLOTER 5

ALESANDORA  
RAMIZA  
HUSNA



# PENGERTIAN

Python adalah bahasa pemrograman **interpretatif multiguna**. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar **lebih mudah** untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.



```
# Python 3: Contoh Output Sempel  
print("Hello world! I'm Python!")
```

✓ 0.0s

```
Hello world! I'm Python!
```

Sumber: – "BUKU PANDUAN PEMROGRAMAN PYTHON"





**WHY**  
**PYTHON?**

# KEUNTUNGAN

1. MUDAH DIBACA, DIPELAJARI, DAN DITULIS
2. BERSIFAT OPEN SOURCE
3. MERUPAKAN BAHASA PEMROGRAMAN TERPOPULER DI DUNIA
4. MEMILIKI BANYAK APLIKASI

Sumber: <https://www.genius.education/blog>



# KEUNTUNGAN

## 1. MUDAH DIBACA, DIPELAJARI, DAN DITULIS

```
//Pemrograman dalam Bahasa C
#include <stdio.h>
int main(){
    printf("Ini Bahasa C!");
    return 0;
}
```

```
#Pemrograman dalam Bahasa Python
print("Ini Bahasa Python!")
```



# KEUNTUNGAN

## 2. BERSIFAT OPEN SOURCE

All Python releases are Open Source (see <https://opensource.org/> for the Open Source Definition).

Find, install and publish Python packages  
with the Python Package Index



Or [browse projects](#)

453,484 projects

4,454,455 releases

8,190,355 files

699,672 users

# KEUNTUNGAN

## 3. MERUPAKAN BAHASA PEMROGRAMAN TERPOPULER DI DUNIA





# KEUNTUNGAN

## 4. MEMILIKI BANYAK APLIKASI



Unity®





# KEUNTUNGAN

**5. IMPROVED PRODUCTIVITY**

**6. GRATIS!!**

**7. SUPPORT LIBRARY LUAS**

**8. PORTABILITY**

**DAN MASIH BANYAK LAGI...**



Sumber: <https://www.genius.education/blog>

# SYNTAX DASAR

Dalam dunia pemrograman, sintaks adalah rangkaian aturan atas susunan kode atau ekspresi secara terstruktur.

- **CASE SENSITIVE**
- **HUMAN FRIENDLY**
- **TITIK KOMA OPSIONAL**
- **PENGGUNAAN PETIK SATU (' '), PETIK DUA (" "), ATAU TANDA PETIK TIGA ('''' ''')** DIPERBOLEHKAN
- **INDENTASI SEBAGAI PEMBENTUK STRUKTUR**
- **TIDAK KETAT TERHADAP TIPE DATA**



# INDENTATION

Indentasi adalah penggunaan spasi atau tab untuk menandai blok kode dalam Python. Dalam python, indentasi adalah hal yang super-super penting, karena ia bertugas untuk mendefinisikan struktur blok kode program.

```
#Contoh pentingnya indentasi
print('Aku')
print('Python')
```

✓ 0.0s

Aku  
Python

```
#Contoh pentingnya indentasi
print('Aku')
print('Python')
```

⊗ 0.0s

Cell In[10], line 3

```
print('Python')
```

^

**IndentationError:** unexpected indent



# STATEMENT

Statement adalah instruksi atau kalimat perintah yang diberikan dalam beberapa baris untuk dieksekusi oleh komputer.

```
#Contoh statement satu baris
```

```
a="Henlo!"
```

```
print(a)
```

✓ 0.1s

Henlo!

```
#Contoh lain statement
```

```
a=5; b=7; c=a+b; print(c)
```

✓ 0.0s

12

```
#Contoh statement multibaris
```

```
a=5
```

```
b=7
```

```
c=a+b
```

```
print(c)
```

✓ 0.0s

12



# COMMENTS

Comments merupakan sekumpulan teks yang dituliskan dalam sebuah program dan tidak akan mempengaruhi hasil dari program tersebut. Secara umum terdapat dua jenis comment.

```
1  #Contoh Single Line Comment
```

```
2
```

```
3
```

```
a, b = 1, 3  # Comments juga bisa  
c = a + b   # diletakkan disamping kode  
print(c)    # seperti ini
```

```
✓ 0.0s
```

```
1  '''
```

```
2  Contoh Multi Line Comment
```

```
3  Apapun yang ada didalam comment
```

```
4  tidak mempengaruhi kode
```

```
5  '''
```

```
6
```

4



**STRUKTUR**

**DATA &**

**TIPE DATA**





# STRUKTUR DATA (BASIC)

```
..: a = "ini adalah string"  
..: print(a)
```

✓ 0.0s

```
ini adalah string
```

```
L = [1, 'ini', 'adalah', 'list']  
print(L)  
print(L[3])
```

✓ 0.0s

```
[1, 'ini', 'adalah', 'list']  
list
```





# STRUKTUR DATA (BASIC)

```
L = (1, 'ini', 'adalah', 'tuple')  
print(L)  
print(L[3])
```

✓ 0.0s

```
(1, 'ini', 'adalah', 'tuple')  
tuple
```

```
L = ([1, 'ini', 'adalah', 'set'])  
print(L)  
print(L[3])
```

✓ 0.0s

```
[1, 'ini', 'adalah', 'set']  
set
```



# STRUKTUR DATA (BASIC)

```
# Contoh Dictionary
# menggunakan Integer Keys
Dict = {1: 'Ini', 2: 'adalah', 3: 'Contoh'}
print("\nDictionary menggunakan Integer Keys: ")
print(Dict)
```

```
# Contoh Dictionary
# menggunakan Mixed keys
Dict = {'Ini': 'contoh', 1: [1, 2, 3, 4]}
print("\nDictionary menggunakan Mixed Keys: ")
print(Dict)
```

```
Dictionary menggunakan Integer Keys:
{1: 'Ini', 2: 'adalah', 3: 'Contoh'}
```

```
Dictionary menggunakan Mixed Keys:
{'Ini': 'contoh', 1: [1, 2, 3, 4]}
```



# TIPE DATA

Tipe data adalah jenis dari suatu data. Setiap data memiliki nilai, dan setiap nilai memiliki jenis. Ada data-data yang bertipe angka, ada pula yang bertipe huruf/karakter, ada juga yang bertipe benar/salah dan sebagainya.

```
# Integer (Bilangan Bulat)
a=100
b=13
```

```
# Boolean
Lab_adaptive_keren=True
lab_adaptive_notkeren=False
```

```
# Float (Bilangan Koma)
a=896.12
b=3.14
```

```
# String
a="Ini contoh string!"
b='Ini juga string!'
```

Sumber: <https://jagongoding.com/>



# VARIABLE

- Tempat penyimpanan data yang bersifat dinamis.
- Dapat berisi teks maupun bilangan.
- **nama\_variabel = <nilai>**
- Terdapat beberapa aturan penulisan variabel, yaitu :
  - Karakter pertama harus berupa huruf atau garis bawah "\_"
  - Karakter selanjutnya dapat berupa huruf, garis bawah "\_" atau angka.
  - Karakter pada nama variabel bersifat sensitif (case-sensitive). Artinya huruf kecil dan huruf besar dibedakan.



# INPUT & OUTPUT

```
#Input  
var = input("Statement: ")  
#Output  
print ("Statement", var)
```

```
#Integer Input  
var = int(input("Statement: "))
```



# OPERATOR

Operator adalah konstruksi yang dapat memanipulasi nilai dari operan.

```
a + b #Penjumlahan  
a - b #Pengurangan  
a * b #Perkalian  
a / b #Pembagian  
a % b #Modulo/Sisa Bagi  
a ** b #Pangkat  
a // b #Pembagian Bulat
```

```
a == a #Sama dengan  
a != b #Tidak sama dengan  
a > b #Lebih dari  
a < b #Kurang dari  
a >= b #Lebih dari sama dengan  
a <= b #Kurang dari sama dengan
```



# IF ELSE

If Else merupakan sebuah kondisi pengambilan keputusan yang digunakan untuk menentukan tindakan apa yang sesuai kondisi dan yang tidak sesuai kondisi.

```
if(condition):  
    statement  
else:  
    statement
```

```
nilai = 10  
if(nilai > 8):  
    print("Anda lulus :)")  
else:  
    print("Anda tidak lulus :(")
```

```
C:\Users\USER\Pycharm  
Anda lulus :)  
  
Process finished with
```





# WHILE LOOP

Mengeksekusi statement berkali-kali selama kondisi bernilai benar atau True.

```
initialization  
while (condition):  
    statement  
    iteration
```

```
num = 0  
while (num <= 3):  
    print (num)  
    num = num + 1
```

```
C:\Users\USER\Pycha  
0  
1  
2  
3
```



# FOR LOOP

Pengulangan For pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti list atau string.

```
for i in iterable:  
    statement  
    ...
```

```
data = 20, 25, 15, 10, 5  
for i in data:  
    print (i)
```

```
C:\Users\USER\Py  
20  
25  
15  
10  
5
```



# NESTED LOOP

Perulangan bersarang (nested loop) adalah struktur perulangan yang berada di dalam perulangan lainnya.

```
for i in range(3):  
    for j in range(5):  
        print (i,j)
```

```
C:\Users\USER\PycharmProje  
0 0  
0 1  
0 2  
0 3  
0 4  
1 0
```



# NUMBER

- Python mendukung beberapa tipe data Number diantaranya :
  - Int
  - Float
  - Complex
- Konversi tipe data menggunakan fungsi:
  - `int (x)`: konversi x menjadi plain integer.
  - `float (x)` : konversi x menjadi floating.



```
#Mengubah float ke integer  
print(int(5.6))  
#Mengubah integer ke float  
print(float(5))
```

```
C:\Users\USER\PycharmProjects\py  
Output yang dihasilkan adalah:  
5  
5.0
```



# STRING

Di dalam bahasa pemrograman Python, "string" merujuk pada tipe data yang digunakan untuk merepresentasikan teks atau urutan karakter. String diapit oleh tanda kutip tunggal atau tanda kutip ganda.

```
print("Hello")  
print('Hello')
```

✓ 0.0s

```
Hello  
Hello
```

```
a = "Hello"  
print(a)
```

✓ 0.0s

```
Hello
```

# FUNGSI

Fungsi adalah blok kode yang hanya berjalan saat dipanggil dan dapat meneruskan data, yang dikenal sebagai parameter, ke dalam suatu fungsi.

Suatu fungsi dapat mengembalikan data sebagai hasilnya.

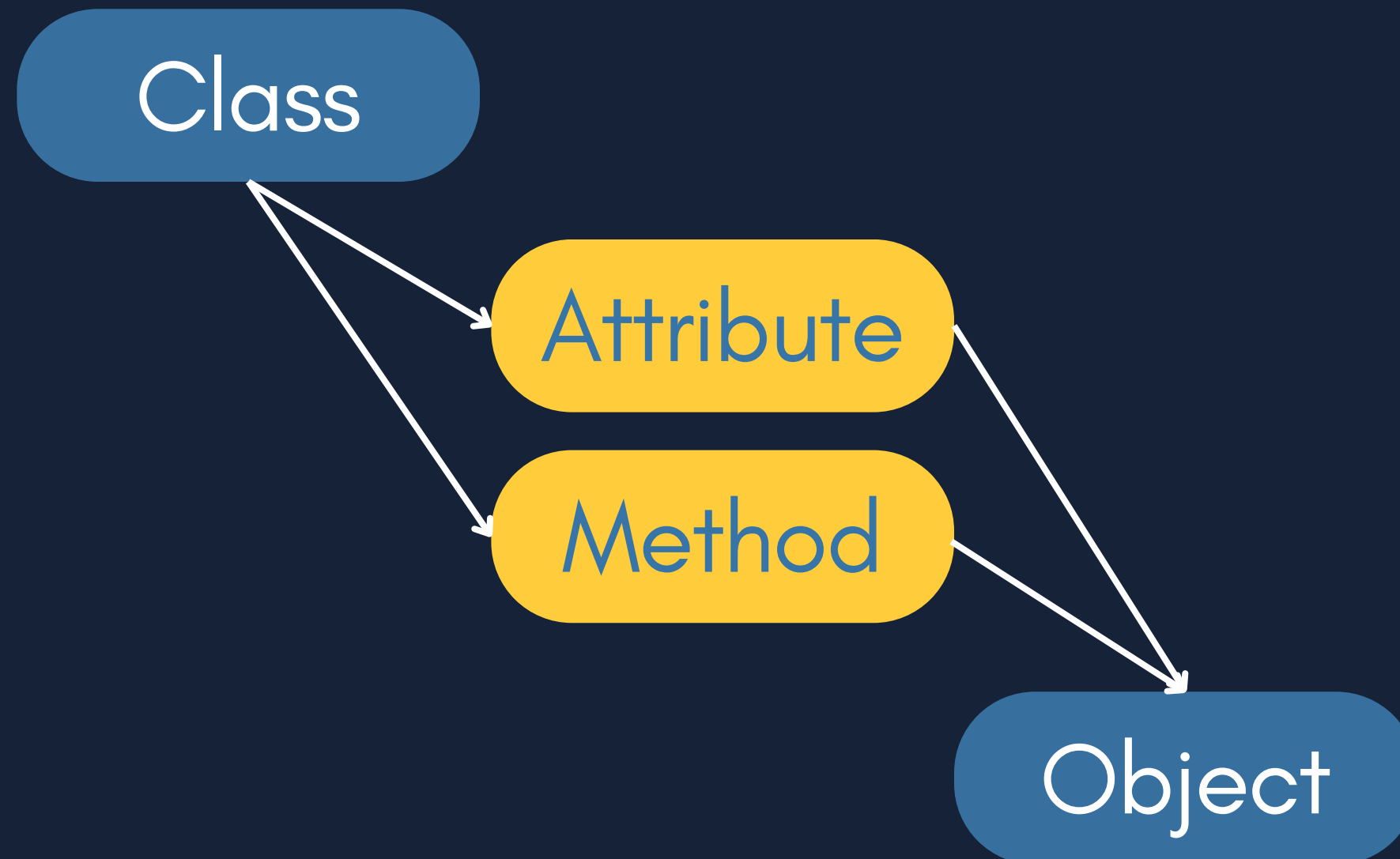
```
1 def alexa():  
2     print("hello")  
3 alexa()
```

```
1 def orang(fname):  
2     print(fname + "cerdas")  
3 orang("alexa ")
```



# CLASS

Class seperti konstruktor object, atau blueprint untuk membuat object.



```
class dataOrang:
    #fungsi __init__ untuk menetapkan nilai ke object
    #self merupakan parameter untuk tiap class variable
    def __init__(self, nama, jurusan):
        self.nama = nama
        self.jurusan = jurusan

    #membuat metode/fungsi untuk class
    def halo(self):
        print("Halo " + self.nama + " " + self.jurusan)

#membuat objek dari className
orang1 = dataOrang("Kuma", "Telco")
orang2 = dataOrang("Lala", "Sisfo")

#memanggil metode halo() dengan object
orang1.halo()
orang2.halo()
```

Output:

```
Halo Kuma Telco
Halo Lala Sisfo
```

# DICTIONARY

Dictionary merupakan tipe data pada Python yang berfungsi untuk menyimpan kumpulan data atau nilai, yang setiap urutanya berisi key dan value. Jika biasanya kita ingin mengakses nilai pada list menggunakan indeks, di dictionary ini kita perlu kata kunci (key) untuk mengakses nilainya.

```
1  datadiri = {"nama": "Alesandora", "Jurusan"  
              : "Teknik", "Usia": 19}  
2  
3  print(datadiri)  
4
```

```
{'nama': 'Alesandora', 'Jurusan': 'Teknik',  
  'Usia': 19}
```

```
> |
```

# LIBRARY

Library python adalah kumpulan modul terkait berisi kumpulan kode yang dapat digunakan berulang kali dalam program yang berbeda.





**PYTHON IS POWERFUL... AND FAST;  
PLAYS WELL WITH OTHERS;  
RUNS EVERYWHERE;  
IS FRIENDLY & EASY TO LEARN;  
IS OPEN.**



Sumber: <https://www.python.org/about/>

