



Class of Statistics Research 1



Dasar-Dasar Python

I Putu Agus Wahyu Dupayana, S.Tr.Stat.

Pranata Komputer Ahli Pertama BPS Provinsi Jawa Timur



Kamis, 9 Juli 2026



/OUTLINE



/01 Tipe Data dan Variabel

/02 Operator

/03 Percabangan

/04 Perulangan

/05 Fungsi

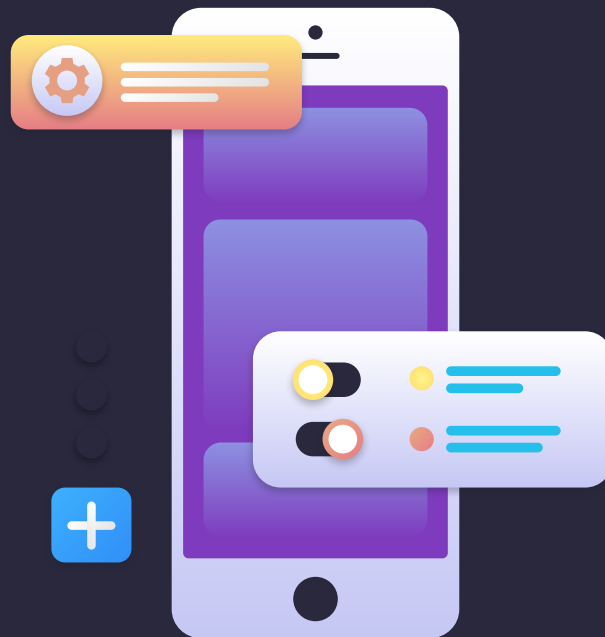
/06 Tugas Praktik





/00

PENDAHULUAN



/APA ITU PYTHON



- Python merupakan bahasa pemrograman tingkat tinggi.
- Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari , karena struktur sintaknya rapi dan mudah dipahami.
- Python adalah bahasa pemrograman yang populer.



/KEGUNAAN PYTHON



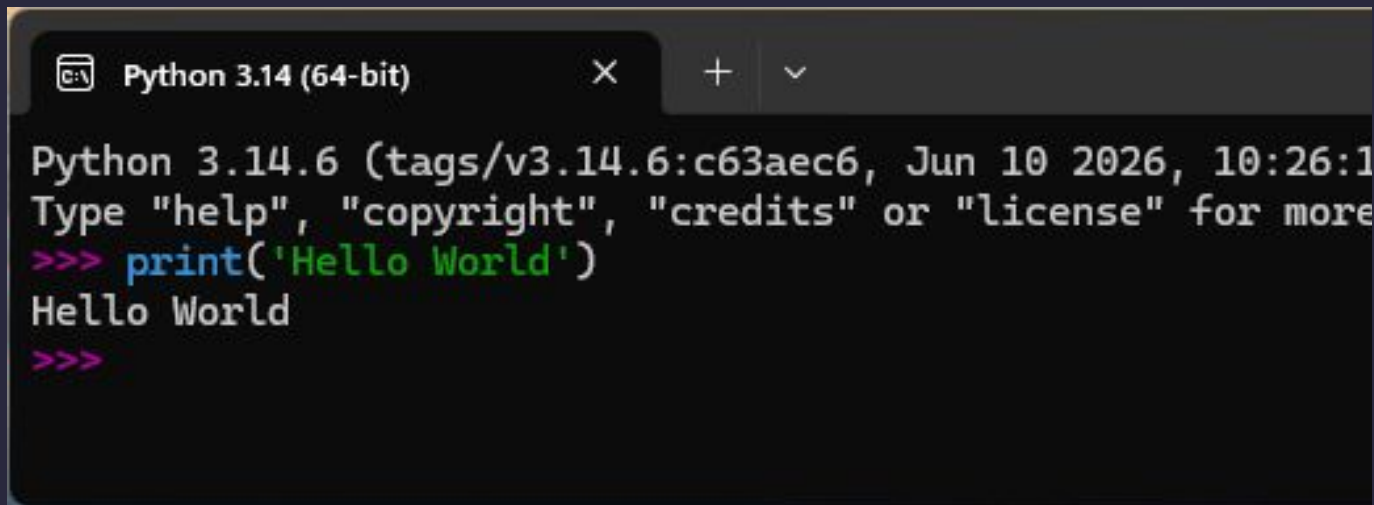
- Kecerdasan Buatan (AI) dan Machine Learning - Library seperti TensorFlow, PyTorch, dan scikit-learn menjadikan Python pilihan utama untuk pengembangan AI.
- Data Science dan Analisis Data - Pandas, NumPy, dan Matplotlib adalah tools standar industri.
- Pengembangan Web - Framework seperti Django, Flask, dan FastAPI.
- Otomasi dan Scripting - Mengotomasi tugas-tugas repetitif dengan mudah.
- Internet of Things (IoT) - Digunakan di Raspberry Pi dan perangkat embedded.
- Desktop Applications: Tkinter, PyQt
- Game Development: Pygame

/INSTALASI PYTHON

- Buka browser, kunjungi halaman download Python untuk Windows
<https://www.python.org/downloads/windows>
- Download installer Python 3.14.6 Rilis 10 Juni 2026 (pilih versi 64-bit untuk Windows)
- Buka (klik 2x) file installer python yang baru saja di download
- Ikuti langkah instalasi sampai selesai



/INTERACTIVE SHELL



```
Python 3.14 (64-bit) × + v
Python 3.14.6 (tags/v3.14.6:c63aec6, Jun 10 2026, 10:26:1
Type "help", "copyright", "credits" or "license" for more
>>> print('Hello World')
Hello World
>>>
```



/FILE PYTHON

- Pada kenyataannya, saat membuat program menggunakan Python, kita biasanya tidak mungkin jarang menggunakan Python Interactive Shell, biasanya kita akan simpan kode program Python kita dalam file
- File Python menggunakan ekstensi / akhiran `.py`, menggunakan huruf kecil semua, jika butuh lebih dari satu kata, pisahkan kata dengan underscore (`_`), dan terakhir hindari spasi dan karakter khusus pada nama file



/MENJALANKAN FILE PYTHON



- Untuk menjalankan kode program yang sudah kita buat, kita bisa menggunakan perintah : `python nama_file.py`
- Pastikan kita mengetikkan perintah `python` di lokasi folder yang terdapat `nama_file.py` tersebut



/FILE JUPYTER NOTEBOOK



- File Jupyter Notebook (sebelumnya, IPython Notebook) digunakan untuk komputasi interaktif di berbagai bidang. File ini berisi catatan lengkap sesi pengguna dan mencakup : Code, Narrative text, Equations & Rich output.
- Untuk mempermudah pembelajaran saat ini kita akan membuat file notebook yang menggunakan ekstensi .ipynb

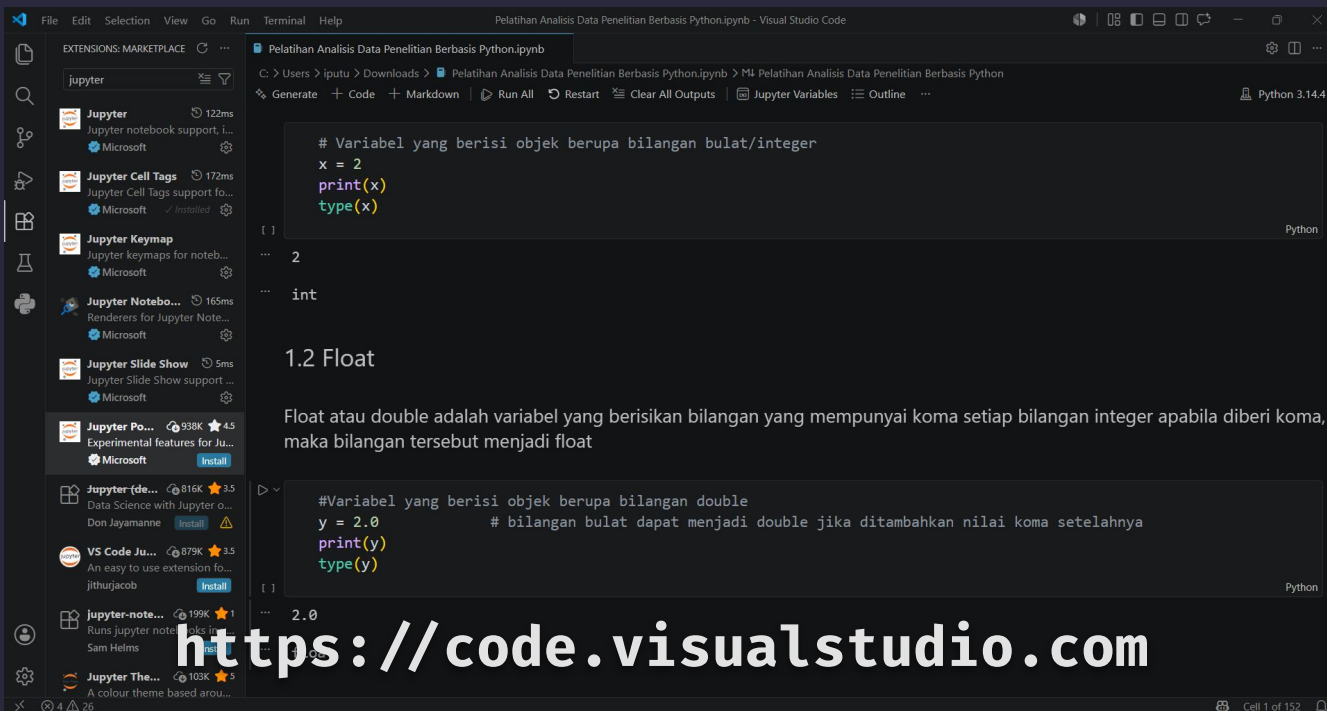


/IDE DAN CODE EDITOR UNTUK PYTHON





/VSCODE



<https://code.visualstudio.com>





/JUPYTER NOTEBOOK



The screenshot shows a Jupyter Notebook window titled "Pelatihan Analisis Data Penel...". The interface includes a file browser on the left, a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), and a code editor. The code is written in Python 3 (ipykernel) and demonstrates basic data types:

```
1.1 Integer

Integer adalah variabel yang berisikan bilangan bulat. Nilai dari variabel integer tidak boleh diikuti tanda koma, karena akan mengubah tipe data tersebut

[6]: # Variabel yang berisi objek berupa bilangan bulat/integer
x = 2
print(x)
type(x)

2
[6]: int

1.2 Float

Float atau double adalah variabel yang berisikan bilangan yang mempunyai koma setiap bilangan integer apabila diberi koma, maka bilangan tersebut menjadi float

[ ]: #Variabel yang berisi objek berupa bilangan double
# bilangan bulat dapat menjadi double jika ditambahkan nilai koma setelahnya
y = 2.0
print(y)
type(y)

2.0
[5]: float

1.3 Boolean

Boolean adalah variabel yang menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0

[ ]: #Variabel yang berisi objek berupa boolean
nilai = True
print(nilai)
type(nilai)

True
[6]: bool
nilai = 'string'
```

At the bottom of the notebook, the status bar shows "Python 3 (ipykernel) | Idle", "Mode: Command", "Cell 5/152", "Ln 3, Col 9", and the file name "Pelatihan Analisis Data Penelitian Berbasis Python.ipynb".

<https://jupyter.org/install>





/GOOGLE COLAB



Pelatihan Analisis Data Penelitian Berbasis Python.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connected

Table of contents

- 6.2 Menginstal Paket
- 6.3 Menggunakan Pak...
- 6.4 Menghapus Paket
- Pengenalan NumPy dan...
- 7.1 Instalasi dan Inisialis...
- 7.2 Membuat Array
- 7.3 Tipe Data NumPy
- 7.4 Dimensi dan Bentuk...
- 7.5 Reshape: Mengubah...
- 7.6 Operasi Matematika
- 7.7 Fungsi Agregasi N...
- 7.8 Operasi pada Arra...
- 7.9 Series: Array 1 Dim...
- 7.10 Membuat Datafra...
- Import dan Ekspor Data
- 8.1 Import Data dari URL
- 8.2 Import Data dari G...
- 8.3 Import Data dari Git
- 8.5 Ekspor Data
- Analisis Data Eksploratif

1.1 Integer

Integer adalah variabel yang berisikan bilangan bulat. Nilai dari variabel integer tidak boleh diikuti tanda koma, karena akan mengubah tipe data tersebut

```
[ ]  
# Variabel yang berisi objek berupa bilangan bulat/integer  
x = 2  
print(x)  
type(x)  
  
2  
int
```

1.2 Float

Float atau double adalah variabel yang berisikan bilangan yang mempunyai koma setiap bilangan integer apabila diberi koma, maka bilangan tersebut menjadi float

```
[ ]  
# Variabel yang berisi objek berupa bilangan double  
y = 2.0 # bilangan bulat dapat menjadi double jika ditambahkan nilai koma setelahnya  
print(y)  
type(y)  
  
2.0  
float
```

1.3 Boolean

Boolean adalah variabel yang berisikan nilai logika, yaitu benar (True) atau salah (False).

Variables Terminal

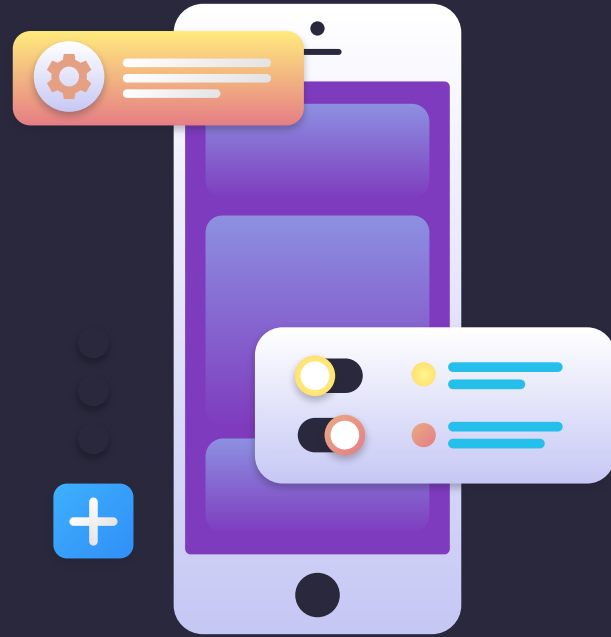
20:30 Python 3

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



/01

/TIPE DATA DAN VARIABEL

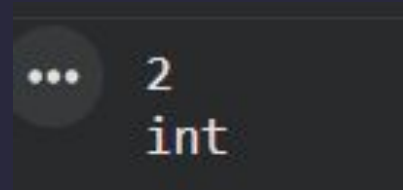


/TIPE DATA INTEGER

Integer adalah variabel yang berisikan bilangan bulat. Nilai dari variabel integer tidak boleh diikuti tanda koma, karena akan mengubah tipe data tersebut



```
x = 2
print(x)
type(x)
```



```
... 2
    int
```

/TIPE DATA INTEGER

Float adalah variabel yang berisikan bilangan yang mempunyai koma

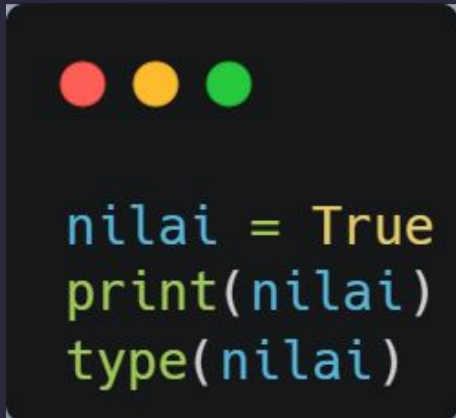


```
y = 2.0  
print(y)  
type(y)
```

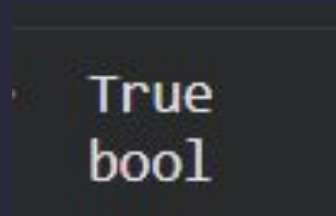
```
... 2.0  
float
```

/TIPE DATA BOOLEAN

Boolean adalah variabel yang menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0.



```
nilai = True  
print(nilai)  
type(nilai)
```



```
True  
bool
```

/TIPE DATA STRING



String adalah variabel yang menyatakan karakter/kalimat bisa berupa huruf angka, dan lain lain.



```
pesan = 'Halo'  
print(pesan)  
type(pesan)
```

```
.. Halo  
str
```

/STRUKTUR DATA LIST

List adalah kumpulan data yang bisa menyimpan banyak nilai dalam satu variabel. Data pada list juga dapat dirubah-rubah. List ditulis dengan menggunakan kurung siku [].



```
x = ["Bobo", "Andi", 1]
print(x)
type(x)
```

```
['Bobo', 'Andi', 1]
list
```

/STRUKTUR DATA TUPLE

Tuple adalah variabel yang berisikan data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah. Dalam membuat Tuple harus ditulis dengan tanda kurung biasa "(..)"



```
x = ("Baba", "Adi", 2)
print(x)
type(x)
```

```
.. ('Baba', 'Adi', 2)
tuple
```

/STRUKTUR DATA DICTIONARY

Dictionary adalah variabel yang berisikan data untaian yang menyimpan berbagai tipe data berupa pasangan key dan value. Key dan value dibatasi oleh titik dua ":" dan ditulis dengan tanda kurung kurawal {}



```
x = {'nama':"Eko", 'umur':36, 1:2}  
print(x)  
type(x)
```

```
{'nama': 'Eko', 'umur': 36, 1: 2}  
dict
```

/STRUKTUR DATA SET

Set adalah kumpulan data yang tidak berurutan dan tidak memiliki elemen duplikat (harus unik). Set ditulis dengan kurung kurawal {} atau fungsi set().



```
x = {"Baba", "Adi", 2}  
print(x)  
type(x)
```

/INPUT DARI PENGGUNA

`input()` selalu menghasilkan string (teks).



```
nama = input("Masukkan nama Anda: ")  
print("Halo, " + nama + "!")
```

```
Masukkan nama Anda: Agus  
Halo, Agus!
```



```
umur_teks = input("Masukkan umur Anda: ")  
print("Umur Anda adalah", umur_teks, "tahun.")  
print("Tipe data: ", type(umur_teks))
```

```
Masukkan umur Anda: 23  
Umur Anda adalah 23 tahun.  
Tipe data: <class 'str'>
```

/KONVERSI TIPE DATA

Secara umum konversi tipe data dalam Python dapat dilakukan dengan cara: `tipe_data(variabel)`



```
x = 20.5
y = int(x)
print(y)
Output:
20
```



```
x = '10'
y = int(x)
print(y)
Output:
10
```



```
x = 10
y = float(x)
print(y)
Output:
10.0
```



```
x = '20.5'
y = float(x)
print(y)
Output:
20.5
```

/VARIABEL



Variabel adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini berarti bahwa ketika Anda membuat sebuah variabel Anda memesan beberapa ruang di memori. Variabel menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel tersebut dapat diubah oleh operasi - operasi tertentu pada program yang menggunakan variabel.

Variabel dapat menyimpan berbagai macam tipe data. Di dalam pemrograman Python, variabel mempunyai sifat yang dinamis, artinya variabel Python tidak perlu dideklarasikan tipe data tertentu dan variabel Python dapat diubah saat program dijalankan.



/VARIABEL

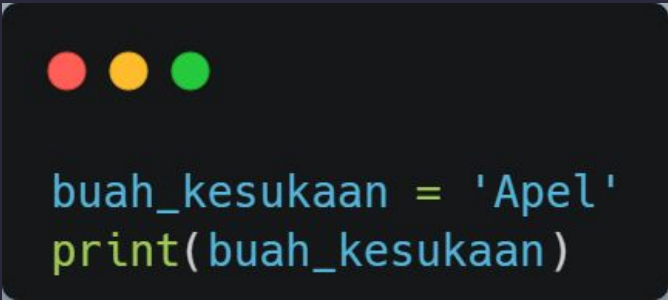


Penulisan variabel Python sendiri juga memiliki aturan tertentu, yaitu

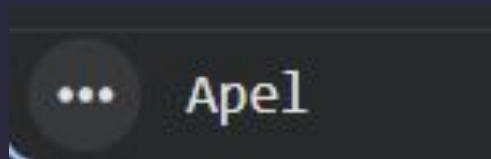
- Karakter pertama harus berupa huruf atau garis bawah/underscore (A-z dan _)
- Karakter selanjutnya dapat berupa huruf, garis bawah/underscore atau angka (A-z, 0-9, dan _)
- Karakter khusus (@, #, %, ^, &, *) tidak boleh digunakan menjadi nama variabel.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf kecil dan huruf besar dibedakan. Sebagai contoh, variabel namaDepan dan namadepan adalah variabel yang berbeda.
- Reserve words tidak dapat dideklarasikan sebagai variabel

/VARIABEL

Untuk mulai membuat variabel di Python caranya sangat mudah, Anda cukup menuliskan variabel lalu mengisinya dengan suatu nilai dengan cara menambahkan tanda sama dengan = diikuti dengan nilai yang ingin dimasukan



```
buah_kesukaan = 'Apel'  
print(buah_kesukaan)
```



```
...
```



/02

/OPERATOR



/OPERATOR ARITMATIKA



Operator	Keterangan
+	Penjumlahan/Addition
-	Pengurangan/Subtraction
/	Pembagian/division
%	modulo/mod
*	Perkalian/multiplication
//	floor division
**	Pangkat/ to the power of



/PENJUMLAHAN



```
print (2 + 10)
print ('Hello' +" " + 'World') #pada string akan digabung
print (([2,3,4,5]) + ([4,6,7,8])) #array tidak dijumlahkan namun digabungkan
```

```
... 12
    Hello World
    [2, 3, 4, 5, 4, 6, 7, 8]
```





/PENGURANGAN



```
print (50 - 20.5)
```

29.5





/PERKALIAN



```
print (0.3 * 3.14)  
print (('Hello' + 'World')*4)
```



```
0.942  
HelloWorldHelloWorldHelloWorldHelloWorld
```





/PEMBAGIAN



```
print (99/11)
```

9.0





/MODULUS / SISA BAGI



```
print (49 % 6)  
print (10.5 % 2 )
```

1

0.5





/PEMBAGIAN BULAT



```
print (30.1 // 15)  
print (22/7 // 3)
```

```
2.0  
1.0
```





/PANGKAT



```
print (8**2)  
print (3/4 ** 2)
```

```
... 64  
    0.1875
```





/OPERATOR PERBANDINGAN



```
# SAMA DENGAN
print(1 == 1) # Hasilnya akan bernilai True karena satu sama dengan satu

# TIDAK SAMA DENGAN
print(2 != 2) # Hasilnya akan bernilai False karena dua seharusnya sama

# LEBIH BESAR DARI
print(5 > 3) # Hasilnya akan bernilai True karena lima lebih besar dari tiga

# LEBIH KECIL DARI
print(5 < 3) # Hasilnya akan bernilai False karena lima tidak lebih besar dari tiga

# LEBIH BESAR DARI SAMA DENGAN
print(5 >= 3) # Hasilnya akan bernilai True karena lima lebih besar dari sama dengan tiga

# LEBIH KECIL DARI SAMA DENGAN
print(5 <= 3) # Hasilnya akan bernilai False karena lima tidak lebih besar dari sama dengan tiga
```

```
... True
False
False
True
True
False
True
False
```





/OPERATOR LOGIKA



```
umur = 25
print (umur > 18 and umur < 25)
hari = "Sabtu"
print (hari == "Sabtu" or hari == "Minggu")
aktif = True
print (not aktif)
```

False
True
False

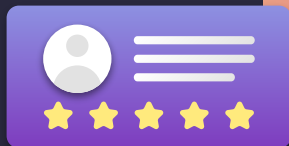
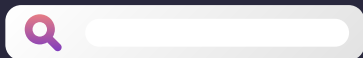


/PRIORITAS OPERATOR



Operator memiliki urutan prioritas (precedence). Operator dengan prioritas lebih tinggi akan dieksekusi terlebih dahulu
Urutan Prioritas (dari tinggi ke rendah):

- `**` (pangkat)
- `*`, `/`, `//`, `%` (perkalian, pembagian)
- `+`, `-` (penjumlahan, pengurangan)
- `=`, `≠`, `<`, `>`, `≤`, `≥` (perbandingan)
- `not` (logika not)
- `and` (logika and)
- `or` (logika or)



/03

/PERCABANGAN





/IF STATEMENT



```
#menggunakan simple if-else statement
```

```
if (5>4): print("5 lebih besar daripada 4")
```

```
↳ 5 lebih besar daripada 4
```



/IF-ELSE STATEMENT



```
nilai = float(input("Nilai: "))  
def HitungNilai(nilai):  
    if (nilai >= 75 ):  
        print("Anda lulus")  
    else:  
        print("Anda tidak lulus")  
  
HitungNilai(nilai)
```



```
Nilai: 57  
Anda tidak lulus
```

/ELIF STATEMENT

```
hari_ini = "Minggu"

if(hari_ini == "Senin"):
    print("Saya akan kuliah")
elif(hari_ini == "Selasa"):
    print("Saya akan kuliah")
elif(hari_ini == "Rabu"):
    print("Saya akan kuliah")
elif(hari_ini == "Kamis"):
    print("Saya akan kuliah")
elif(hari_ini == "Jumat"):
    print("Saya akan kuliah")
elif(hari_ini == "Sabtu"):
    print("Saya akan kuliah")
elif(hari_ini == "Minggu"):
    print("Saya akan libur")
```

Saya akan libur

/MATCH CASE



```
hari_ini = "Minggu"
```

```
match hari_ini:
```

```
    case "Senin" | "Selasa" | "Rabu" | "Kamis" | "Jumat":
```

```
        print("Saya akan kuliah")
```

```
    case "Sabtu":
```

```
        print("Saya akan mengerjakan tugas")
```

```
    case "Minggu":
```

```
        print("Saya akan libur")
```

```
    case _:
```

```
        print("Hari tidak valid")
```

Saya akan libur



/04 /PERULANGAN





/FOR

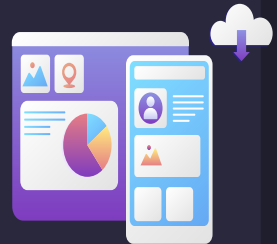


```
#For akan menampilkan variabel "i" mulai dari angka 1 hingga 5 dengan jarak 1.  
for i in range(1, 6, 1):  
    print(i)
```



```
1  
2  
3  
4  
5
```

range(x) maksudnya mulai dari 0 hingga (x-1)
range(x,y) maksudnya mulai dari x hingga (y-1)
range(x,y,z) maksudnya mulai dari x hingga (y-1)
dengan interval z. x,y,z harus bernilai integer.



/FOR



```
#For juga dapat digunakan didalam for (nested loops).  
a = list([[1,3,5],[2,4,6]])  
for i in range(0, 2, 1): # nested loop  
    for j in range(0, 3, 1):  
        print(a[i][j], end='  ')  
    print()
```



```
1  3  5  
2  4  6
```

range(x) maksudnya mulai dari 0 hingga (x-1)
range(x,y) maksudnya mulai dari x hingga (y-1)
range(x,y,z) maksudnya mulai dari x hingga (y-1)
dengan interval z. x,y,z harus bernilai integer





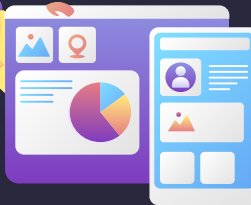
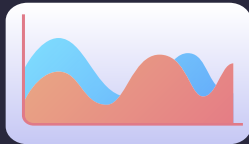
/WHILE



```
#While akan digunakan untuk memanggil angka 1 sampai 5.  
i = 1  
while i <= 5:  
    print(i)  
    i += 1
```



```
1  
2  
3  
4  
5
```





/WHILE



```
# input password sampai benar
password = ""
while password != "secret":
    password = input("Masukkan password: ")
    if password != "secret":
        print("Password salah. Silakan coba lagi.")
print("Password benar. Selamat datang!")
```

```
Masukkan password: ssss
Password salah. Silakan coba lagi.
Masukkan password: asasa
Password salah. Silakan coba lagi.
Masukkan password: fsfdsfds
Password salah. Silakan coba lagi.
Masukkan password: secret
Password benar. Selamat datang!
```





/CONTINUE



```
# Mencetak angka ganjil  
for i in range(20):  
    if i % 2 == 0:  
        continue  
    print(i)
```

```
... 1  
     3  
     5  
     7  
     9  
    11  
    13  
    15  
    17  
    19
```



/BREAK



```
angka_rahasia = 7
```

```
while True:
```

```
    tebakan = int(input("Masukkan tebakan Anda: "))
```

```
    if tebakan == angka_rahasia:
```

```
        print("Selamat! Anda menebak angka rahasia dengan benar.")
```

```
        break
```

```
    else:
```

```
        print("Tebakan Anda salah. Coba
```

```
Masukkan tebakan Anda: 1
```

```
Tebakan Anda salah. Coba lagi.
```

```
Masukkan tebakan Anda: 54
```

```
Tebakan Anda salah. Coba lagi.
```

```
Masukkan tebakan Anda: 7
```

```
Selamat! Anda menebak angka rahasia dengan benar.
```



/PASS



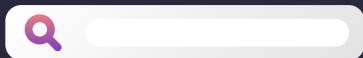
#Pass digunakan sebagai tanda tidak menghasilkan keluaran apapun.
#Pass berguna ketika kita belum menetapkan statement untuk sebuah fungsi, karena di dalam #Python, sebuah fungsi harus memiliki statement agar tidak error.

```
for i in range(5):  
    pass
```

```
for i in range(5):  
    pass  
    print(i, end=" ")
```

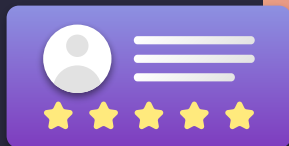
• 0 1 2 3 4





/05

/FUNGSI





Built-in Function

Fungsi yang langsung difasilitasi oleh python.



User-defined Function

Fungsi yang perlu ada didefinisikan sendiri oleh pengguna



/BUILT-IN FUNCTION

Fungsi yang langsung difasilitasi oleh python.



```
a=(1,5,3,9)
n=max(a)
m=min(a)
# menjumlahkan semua pada a dan ditambah 7
s=sum(a,7)
print("max : ",n)
print("min : ",m)
print("sum : ",s)
```

```
➞ max : 9
   min : 1
   sum : 25
```

/STRUKTUR USER- DEFINED FUNCTION



```
def nama_function():  
    print("Hello dari function")
```

```
nama_function()  
nama_function()  
nama_function()
```

```
.. Hello dari function  
   Hello dari function  
   Hello dari function
```

1. Blok fungsi diawali dengan “def” yang diikuti oleh nama fungsi dan tanda kurung ().
2. Input dari parameter atau argumen ditempatkan di dalam tanda kurung
3. Pernyataan pertama dari fungsi dapat berupa docstring (OPTIONAL)
4. Baris kode tiap fungsi dimulai dengan titik dua (:) dan menjorok
5. “Return” nilai yang dikembalikan oleh fungsi setelah eksekusi tugas selesai dilakukan. Jika return tanpa argumen maka akan dikembalikan None

/FUNCTION DENGAN PARAMETER



- Parameter memungkinkan kita mengirim data ke function
- Caranya kita bisa tambahkan nama parameter diantara kurung buka dan kurung tutup di function
- Jika kita ingin menambahkan lebih dari satu parameter, kita bisa gunakan karakter koma sebagai pemisah parameter tersebut
- Saat memanggil function nya, maka kita wajib mengisi data pada parameter function nya



/FUNCTION DENGAN PARAMETER



```
#Function dengan Parameter
```

```
def sapa_nama(nama):  
    print("Hello", nama)  
    print("Senang bertemu dengan Anda")
```

```
sapa_nama("Agus")
```

```
sapa_nama("Budi")
```

```
... Hello Agus  
    Senang bertemu dengan Anda  
    Hello Budi  
    Senang bertemu dengan Anda
```



/FUNCTION DENGAN RETURN VALUE



- Saat membuat function, mungkin kita ingin function tersebut melakukan suatu kegiatan dan hasil kegiatan tersebut ingin kita dapatkan.
- Function bisa mengembalikan nilai dari kegiatan yang dilakukan di dalam function
- Function bisa mengembalikan nilai menggunakan kata kunci return
- Setelah kata kunci return, diikuti dengan nilai yang akan dikembalikan di function tersebut



/FUNCTION DENGAN RETURN VALUE



```
# Function dengan Return Value
def hitung_luas_lingkaran(radius):
    pi = 3.14159
    luas = pi * radius * radius
    return luas
```

```
luas1 = hitung_luas_lingkaran(5)
luas2 = hitung_luas_lingkaran(10)
```

```
print("Luas lingkaran radius 5", luas1)
print("Luas lingkaran radius 10", luas2)
```

```
Luas lingkaran radius 5 78.53975
Luas lingkaran radius 10 314.159
```



/FUNCTION DENGAN DEFAULT PARAMETER



- Saat kita menambahkan parameter di function, maka kita wajib mengisi nilai para parameter tersebut ketika memanggil function nya
- Namun, kita juga bisa menambahkan default parameter, yaitu nilai awal untuk parameter tersebut
- Hal ini menjadikan jika kita tidak mengisi nilai pada parameter tersebut, maka parameter tersebut akan menggunakan nilai awal
- Posisi default parameter, tidak boleh di depan parameter biasa, harus di belakang parameter biasa
- Untuk membuat default parameter, cukup tambahkan = (sama dengan) dan diikuti dengan nilai awalnya



/FUNCTION DENGAN DEFAULT PARAMETER



```
def sapa(nama, sapaan="Hello"):  
    print(sapaan, nama)
```

```
sapa("Eko")  
sapa("Budi", "Hi")  
sapa("Joko", "Hey")
```

```
... Hello Eko  
    Hi Budi  
    Hey Joko
```



/KEYWORD ARGUMENT



- Argument adalah nilai yang kita kirim ke parameter function ketika kita memanggil function
- Saat memanggil function, posisi argument harus sama dengan posisi parameter di function
- Namun, jika kita ingin mengubah posisi argument, sehingga posisinya tidak harus sama dengan posisi parameter, maka kita harus menyebutkan nama parameter nya saat memanggil function
- Fitur ini disebut Keyword Argument
- Saat menggunakan Keyword Argument, kita juga bisa mengkombinasikan dengan Argument biasa

/KEYWORD ARGUMENT

```
def perkenalan(nama, umur, kota):  
    print("Nama", nama)  
    print("Umur", umur)  
    print("Kota", kota)  
    print("-----")
```

```
perkenalan("Eko", 30, "Jakarta")  
perkenalan(kota="Bandung", umur=25, nama="Rani")  
perkenalan(umur=17, kota="Surabaya", nama="Budi")
```

```
Nama Eko  
Umur 30  
Kota Jakarta  
-----  
Nama Rani  
Umur 25  
Kota Bandung  
-----  
Nama Budi  
Umur 17  
Kota Makasar  
-----
```

/VARIABEL GLOBAL & LOCAL



- Saat kita membuat variabel di dalam function, maka variabel tersebut bersifat local di function itu, artinya tidak bisa digunakan di luar function
- Variabel yang kita buat di luar function, maka bisa digunakan di dalam function, atau bersifat global
- Namun kita tidak bisa mengubah nilai dari global variable di dalam function, jika kita ingin mengubah nilai dari global variable, maka kita harus gunakan kata kunci global di dalam function nya

/VARIABEL GLOBAL & LOCAL



```
angka_keberuntungan = 5

def buat_angka_keberuntungan():
    angka_keberuntungan = 10
    print("Angka keberuntungan dalam function:",
    angka_keberuntungan)
    buat_angka_keberuntungan()
print("Angka keberuntungan luar function:", angka_keberuntungan)
```

```
Angka keberuntungan dalam function: 10
Angka keberuntungan luar function: 5
```

/TUGAS PRAKTIK



Buatlah sebuah program kalkulator sederhana berbasis teks. Program ini harus menampilkan menu utama yang memungkinkan pengguna memilih salah satu dari 4 operasi matematika dasar (Penjumlahan, Pengurangan, Perkalian, Pembagian) atau memilih untuk Keluar dari program. Program harus memiliki fungsi terpisah untuk setiap operasi matematika. Kalkulator hanya melakukan operasi untuk dua angka saja. Program akan terus kembali ke menu utama setelah selesai melakukan satu perhitungan, sampai pengguna memilih angka 5 (Keluar).



/TUGAS PRAKTIK

Langkah-langkah :

- 1. Membuat Fungsi Penjumlahan
- 2. Membuat Fungsi Operasi Lainnya
- 3. Membuat Kerangka Menu Utama
- 4. Menambahkan Logika Pemilihan Menu
- 5. Menjalankan Program

```
PowerShell 7.6.3
PS C:\Users\iputu\Downloads\CoStar1> python kalkulator.py
=== PROGRAM KALKULATOR SEDERHANA ===
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Keluar
=====
Pilihan: 3
=== PROGRAM PERKALIAN ===
Masukkan angka 1: 20
Masukkan angka 2: 4.5
Hasil perkalian: 90.0
Tekan Enter untuk lanjut...
```



<Thank u>



DO YOU HAVE ANY
QUESTIONS?

