



## Tutorial Menggunakan Jupyter Notebook

Tempat untuk menuliskan atau menjalankan code disebut dengan **Cell**

Pada Jupyter Notebook terdapat 2 jenis **Cell** yaitu:

1. Code Cell
2. Markdown Cell

Untuk masuk kedalam mode **Cell** bisa menggunakan command **Y** dan command **M** untuk mode **Markdown**

## Menjalankan Cell

Jika ingin menjalankan **Cell**, kita dapat menggunakan **Ctrl+Enter** atau dengan **Shift+Enter**

Command	Keterangan
Ctrl+Enter	Eksekusi & Tetap di cell
Shift+Enter	Eksekusi & Fokus ke next cell

### Latihan

Cetaklah "Selamat Belajar" pada Code 1 dan "Pemrograman python" pada Code 2, kemudian jalankan cell nya dengan Ctrl+Enter atau Shift+Enter

In [1]: *#Code 1*

Selamat Belajar

In [2]: *#Code 2*

Pemrograman python

# Menambahkan Cell

Jika kita kehabisan **Cell** maka dapat ditambahkan menggunakan :

- command A (above) -> Menambah diatas
- command B (below) -> Menambah dibawah

Tetapi perlu **diperhatikan** jika ingin menambahkan **Cell** warna garis **Cell** disebelah kiri harus

---

## Shortcut

- **Delete Cell** menggunakan command **D+D**
  - **Cut Cell** menggunakan command **X**
  - **Paste below** menggunakan command **V**
  - **Paste above** menggunakan command **Shift+V**
  - **Undo** menggunakan command **Z**
- 

## Markdown

Markdown adalah (lightweight markup language) bahasa markup yang lebih ringan dari HTML yang digunakan untuk

- Membuat websites
- Documents
- Notes
- Presentation
- Technical Documentation
- Math Notations

## Outline

1. Print
  2. Variabel & Data Type
  3. More Print
  4. Comment
  5. Error & Problem Solving
-

## How to print a text

- Sebagai pengalaman ngoding awal, kita bisa menampilkan sesuatu lewat Python.
- Pada bagian ini, kita akan coba menampilkan sebuah text.
- Caranya adalah dengan menggunakan perintah print melalui syntax

```
print "<teks yang ditampilkan>"
```

- Misalkan kita mau menampilkan kalimat Hello World!
- Kita dapat melakukannya dengan cara seperti berikut

```
print "Hello World!"
```

- Selain menggunakan tanda petik dua ("..."), kita juga dapat menggunakan tanda petik satu ('...')

```
print('Hello world!')
```

- Sekarang ayo latihan, coba print kalimat **Halo Dunia!**

In [4]: `#Code Here`

```
Hallo Dunia!
```

## How to print a number

Selain melakukan printing berupa text, kita juga dapat melakukan printing berupa angka dengan menggunakan syntax

```
print(<ANGKA YANG INGIN DITAMPILKAN>)
```

Bedanya dengan text adalah apabila ingin melakukan printing berupa angka tidak perlu diikuti dengan tanda petik satu (") maupun tanda petik dua (").

Jika ingin mencetak angka dalam jumlah besar, bisa menggunakan **underscore** ( \_ ) untuk memudahkan pembacaan kode oleh orang lain.

Sekarang ayo latihan, cobalah print tanggal hari ini!

In [8]: `#code here`

```
23
```

## Variables

- Variabel merupakan tempat menyimpan sebuah nilai ataupun data.
- Kita dapat membuat variabel dengan menggunakan syntax sebagai berikut:

```
nama_variabel = <nilai>
```

- Nilai dapat berupa teks maupun angka/bilangan.

## Why Use Variables?

Lalu kenapa kita harus menggunakan variabel?

Perhatikan contoh di bawah ini!

```
angka_1 = 100000000  
angka_2 = 50000000
```

Code di atas menunjukkan bahwa terdapat 2 variabel yaitu angka\_1 dan angka\_2.

Kedua variabel tersebut menyimpan nilai angka yang sangat besar. Bayangkan apabila kita tidak menggunakan variabel dan ingin menjumlahkan kedua angka tersebut dan menulis ulang kembali angkanya, maka akan ada kemungkinan terjadinya kesalahan. Untuk itulah kita perlu menyimpannya di dalam sebuah variabel, yang mana variabel ini nantinya dapat dipanggil.

### Contoh

Sekarang ayo latihan, coba buat sebuah variabel yang menyimpan text "Saya sedang belajar membuat variabel"

Nama Variabel = text

Isi dari Variabel = "Saya sedang belajar membuat variabel"

## How to Write a Proper Variables

Di dalam variabel kita akan mengenal Bad and Good Variables

good name	good alternative	avoid
Max_temp_C	MaxTemp	Maximum Temp (°C)
Precipitation_mm	Precipitation	precmm
Mean_year_growth	MeanYearGrowth	Mean growth/year
sex	sex	M/F
weight	weight	w.
cell_type	CellType	Cell type
Observation_01	first_observation	1st Obs.

### Latihan

```
In [30]: #Code here
#menghitung diameter lingkaran
print("menghitung diameter lingkaran")
r=float(input("masukkan jari-jari: "))
diameter=2*r
print(diameter)

#menghitung keliling lingkaran
print("MENGHITUNG KELILING LINGKARAN")
r2=float(input("masukkan jari-jari: "))
pi=3.14
keliling=2*pi*r2
print(keliling)
```

```
menghitung diameter lingkaran
masukkan jari-jari: 4
8.0
MENGHITUNG KELILING LINGKARAN
masukkan jari-jari: 4
25.12
```

Python juga dapat mengisi banyak nilai ke dalam banyak variabel

```
x, y, z = nilai_x, nilai_y, nilai_z
```

## Latihan

Sebuah segitiga sama kaki mempunyai 2 sudut yang sama yaitu 30 derajat, carilah nilai dari 1 sudut lainnya!

```
In [11]: #code here
#rumus maka:
#x+x+30=180
#2x+30=180
# 2x = 180-30
# 2x=150
# x=75
sudut1=int(30)
sudut2=int(30)
totalSudut=int(180)
#mencari sudut 3:
hasil1 = totalSudut-sudut1
print(f"Besar sudut 3 adalah",hasil1/2,"Derajat")
```

```
Besar sudut 3 adalah 75.0 Derajat
```

Selain itu untuk penulisan sebuah konstanta disarankan menggunakan HURUF KAPITAL

```
PHI = 3.14
```

Buatlah konstanta untuk nilai dari gravitasi yaitu 9.8

```
In [8]: #Code here
G=9.8
```

Konstanta gravitasi di tulis 9.8

## More Printing

1. Cara lama menggunakan %, %s untuk teks dan %d untuk angka
2. Cara baru menggunakan f-string

```
nama = "alun"
umur = 40
print("Nama : %s, Umur : %d" %(nama,umur))
print(f>Nama anda adalah {nama}, umur anda adalah {umur})
```

## Latihan

Sebuah kedai kopi setarbug mempunyai menu yaitu **expresso** dengan harga **35.000**, **Ice Latte Blended** dengan harga **55.000** dan **Croissant** dengan harga **15.000** Ubahlah kedalam bentuk variabel dan carilah berapa harga total jika seseorang membeli semua menu dengan jumlah masing-masing 2?

```
In [13]: #Code here
#diketahui:
#expresso : 35.000
#ice latte blended : 55.000
#croissant : 15000

expresso=3500
icl=55000
croissant=15000
jumlah=2

#beli
Jexpresso=expresso*jumlah
Jicl=icl*jumlah
Jcroissant=croissant*2
rp="Rp. "
rp1=", -"

#total harga
total=Jexpresso+Jicl+Jcroissant
print(f>Jumlah total harga: {rp}",total,rp1)
```

Jumlah total harga: Rp. 147000 , -

## Latihan

Tomi mempunyai 2 kendaraan dengan nopol H 2134 FG dan F 1234 DD. Pajak kendaraan tersebut adalah 100.000. Peraturan pajak adalah pajak progressive yaitu tambahan sebanyak

20% dari pajak kendaraan pertama. Jika Tomi ingin menabung untuk membayar pajak kendaraannya selama 1 tahun, berapakah jumlah yang harus dikumpulkan perbulannya?

```
In [26]: #Code here
progresif=int(20*100000/100000)
prgsf=progresif
print(f"progresif pajak adalah sebesar: ",prgsf)
kendaraan1=int (100000)
kendaraan2=int (100000)
progresif=prgsf

#total yang harus dibayar
print(f"total yang harus dibayar adalah :",kendaraan1+kendaraan2+prgsf)
#menabung 1 tahun untuk mendapatkan
satuthn=12
menabung=int(kendaraan1/satuthn)
print()
print("Tomi harus menabung per bulan nya adalah: ",menabung)
```

progresif pajak adalah sebesar: 20  
total yang harus dibayar adalah : 200020

Tomi harus menabung per bulan nya adalah: 8333

## Commenting

Komentar pada python dituliskan dengan menggunakan awalan tanda #

1. Commenting merupakan sebuah komentar yang digunakan untuk memperjelas maksud dari sebuah proses ataupun sebuah perintah.
2. Apabila dijalankan commenting tidak akan melakukan eksekusi apapun.
3. Commenting diperlukan untuk mempermudah kita mengerti suatu program.
4. Commenting juga akan mempermudah orang lain yang hendak membaca program kita kelak.

Terdapat 3 macam bentuk komentar, yakni:

- 1.Single Line - (#) - digunakan untuk menjelaskan arti sebuah proses
- 2.Inline - (#) - biasanya berada satu baris dengan perintahnya
- 3.Multi Line - ("""") - digunakan apabila memiliki isi/konten yang banyak

```
In [ ]: '''
        Multiple Line
        '''
#Code here
```

## Expression Evaluation

Sebuah ekspresi adalah kombinasi dari nilai, variabel dan operator.

Contoh : 5 / 2

Bedakan antara mencetak menggunakan print dengan mengevaluasi ekspresi

In [ ]: `#Code Here`

## Error and How to solve

- Pada sesi ini, kita akan belajar mengenai error dan bagaimana cara mengatasinya.
- Error merupakan hal yang wajar ditemui saat sedang membuat sebuah pemrograman.
- Melalui error jugalah kita semakin banyak belajar dan semakin memahami code yang telah kita buat.

Ketika mendapati suatu error, ada beberapa langkah yang dapat kita lakukan untuk dapat mengatasinya, yaitu:

- Perhatikan error outputnya
- Cobalah untuk googling mengenai error tersebut
- Kemudian cobalah code yang telah digoogle sebelumnya
- Apabila masih belum solve, cobalah untuk googling kembali

Ada beberapa jenis error yang sering dialami, yakni:

1. Syntax Error
2. Indentation Error
3. Name Error

In [ ]: `#code here`