



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

NIM	71241119
Nama Lengkap	Antonius Kiya Ananda Derron
Minggu ke / Materi	01 / Bahasa Pemrograman Python

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA**



MATERI 1

A. (Kenapa Python) gambar 1.1: logo Python

Secara garis besar, python adalah bahasa pemrograman yang bersifat dinamis secara semantik dan mendukung *Object Oriented Programming*. Bahasa ini merupakan bahasa terpopuler nomor dua setelah *javascript* menurut survei yang didapat dari stackoverflow (<https://insights.stackoverflow.com/survey/2019#technology>). Alasannya terdapat pada kesederhanaan sintaks yang bersifat intuitif dan menyerupai bahasa inggris yang dimiliki dan juga karena sifatnya yang *open source* membuat dukungan dari komunitas sangat besar terutama ketika sedang belajar (melalui video, dokumentasi, roadmap, dll). Selanjutnya, dukungan ekosistem *library* yang sangat besar membuat bahasa pemrograman ini sangat multiguna atau *versatile*. Selain itu, *library* bawaan python sudah sangat beragam dan sesuai dengan kebutuhan aspek dasar seperti untuk pengembangan web, analisis data, AI (*artificial intelligence* atau kecerdasan buatan), pengembangan perangkat lunak dan masih banyak lagi.

Kekurangan dari bahasa ini adalah belum mendukung mobile app development (meski ada beberapa framework yang bisa digunakan seperti Kivy dan BeeWare) seperti di Android dan IOS. Selain itu, dari aspek kecepatan tidak bisa secepat *compiled programming language* seperti C++ atau Java. Selain itu, penggunaan memori python juga lebih besar, yang mungkin menjadi masalah untuk program yang membutuhkan kinerja dan efisiensi memori.

```
1
2 #contoh kemudahan dan kesederhanaan python adalah banyaknya build-in function yang tersedia
3 #contoh pemrograman kalkulator kesederhanaan
4 while True: print(eval(input(">>>")))
```

gambar 1.2 : program python sederhana menggunakan fungsi eval().

Meskipun nampak bahwa python bahkan dapat membuat program kalkulator bahkan dengan 1 baris kode, ada masalah keamanan dan efisiensi memori yang rendah. Contohnya adalah karena program tersebut meminta *input string*, pengguna dapat merusak sistem atau bahkan mengakses data sensitif jika memberi *input* berbahaya yang akan dieksekusi berikutnya jika tidak hati-hati dalam mengolah *source code*. Namun, ada alternatif *hard code* untuk membuat program ini sama seperti bahasa lain namun lebih sederhana secara sintaksis menggunakan "switch case".

Mari kita lihat perbandingan yang memperlihatkan kesederhanaan sintaks c dan python :

```
9 #include <stdio.h>
10
11 int main()
12 {
13     printf("Hello World");
14
15     return 0;
16 }
```

gambar 1.3 : program "hello world" pada c.

- `#include` : Memanggil *library* untuk menggunakan fungsi print.
- `int main()` : Menyatakan fungsi utama yang akan mencetak teks.

- `return 0` : Menyatakan bahwa program akan dijalankan tanpa adanya masalah.



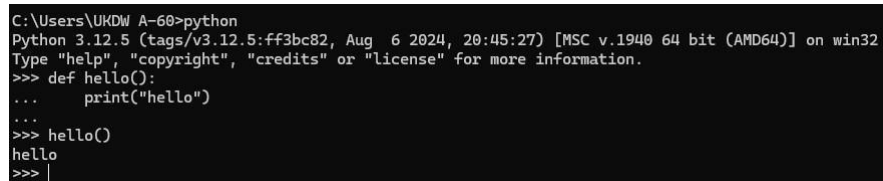
```
main.py
1
2 #hello world
3 print("hello world")
```

gambar 1.4 : program “hello world” pada Python.

di python, kita hanya perlu menggunakan *built - in* function bernama “`print()`” untuk mencetak “hello world”.

B. Instalasi python

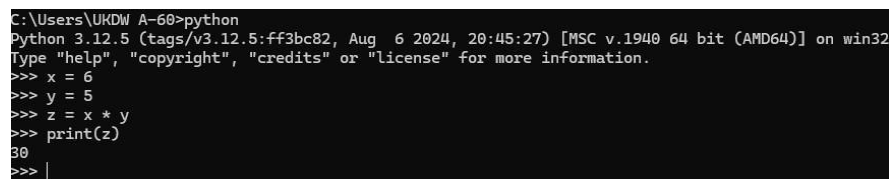
python sendiri dapat diinstal melalui <https://www.python.org/downloads> atau jika OS yang digunakan adalah mac OS atau linux biasanya sudah tersedia versi 3 yang siap untuk dipergunakan. Program python sendiri dapat ditulis dimana saja, termasuk di notebook, terminal, Visual Studio (atau IDE lainnya), dan VS Code (atau code editor manapun).



```
C:\Users\UKDW A-60>python
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> def hello():
...     print("hello")
...
>>> hello()
hello
>>>
```

gambar 1.5 : Contoh program sederhana untuk mengeluarkan output “hello” dengan fungsi melalui terminal.

Seperti yang kita lihat pada *source code* di atas, python memang tidak menggunakan tanda khusus untuk mengakhiri barisnya seperti bahasa C (“;”). Namun, bahasa ini menggunakan indentasi ke dalam untuk membedakan akhir baris dengan baris lain.



```
C:\Users\UKDW A-60>python
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 6
>>> y = 5
>>> z = x * y
>>> print(z)
30
>>>
```

gambar 1.6: Program perkalian sederhana di python.

- Variabel : Suatu kontainer dalam dunia pemrograman dan matematika yang digunakan untuk menampung suatu nilai.
- Dalam python, tidak perlu inisialisasi tipe data variabel seperti beberapa bahasa lain.
- Tipe data dapat berupa *integer, float, bool, list*, dll.

MATERI 2



A. Anaconda dan Python Interaktif

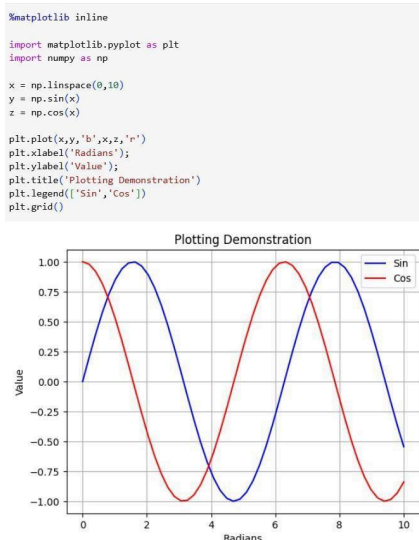
ANACONDA

sumber : <https://idmetafora.com/news/read/1281/Mengenal-Lebih-Jauh-Mengenai-Anaconda-Python.html>

Python interaktif adalah *environment* python yang dapat dijalankan di terminal atau command prompt serta dapat menulis program dan melihat hasilnya langsung. Salah satu python interaktif adalah jupyter notebook yang dapat dibuka melalui anaconda prompt. Anaconda adalah Anaconda merupakan platform ilmu data gratis. memungkinkan untuk menginstal tergantung pada sistem operasi Windows, Linux, MacOS. Terdiri dari distribusi Python dan R , manajer paket yang disebut dengan conda. Anaconda prompt adalah terminal bawaan yang dimiliki oleh packet distributor anaconda. Sama halnya dengan PIP atau “Packet Installer for Python”, anaconda prompt menyediakan dependensi dan library yang lebih banyak karena terdapat dependensi python dan non python sekaligus. Instalasi anaconda sudah termasuk dengan anaconda prompt yang dapat digunakan untuk menginstall jupyter notebook sebagai salah satu python interaktif yang diakses dengan perintah : `conda install -c conda-forge notebook`.

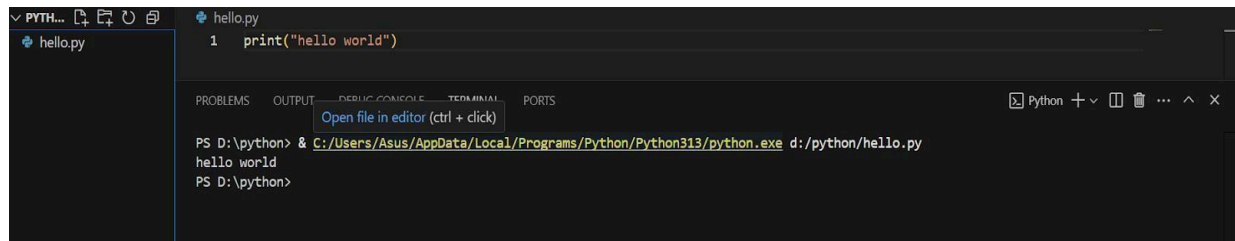
Jupyter bagus sebagai media pembelajaran karena :

- Debugging yang lebih mudah,
- Mendukung beberapa bahasa pemrograman, dan
- Menampilkan visualisasi data dengan lebih mudah.



gambar 1.7 : contoh visualisasi grafik di jupyter

Seperti yang materi sebelumnya sampaikan, python memiliki beberapa IDE atau code editor yang cukup terkenal seperti Visual Studio, PYCharm, Spyder, dan VS Code. Berikut adalah contoh tampilan python di VS Code.

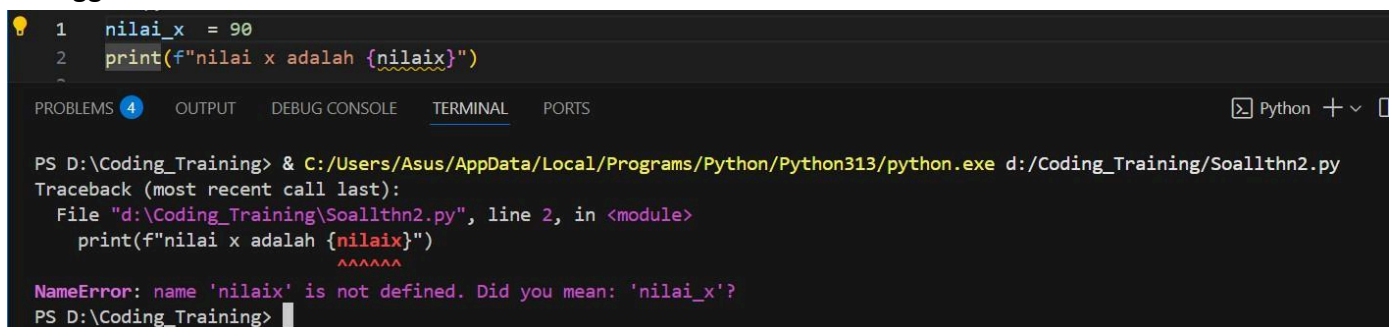


Gambar 1.7 : Python di dalam VSCode.

Kita bisa menjalankan script python melalui terminal yang terdapat di VSCode. Kita juga bisa menjalankan program dengan mengetikkan “python3 namafile”. kenapa “python3”? karena versi python yang saya gunakan adalah versi 3.12. Anda dapat mengeceknya melalui command prompt : `python3 --version`.

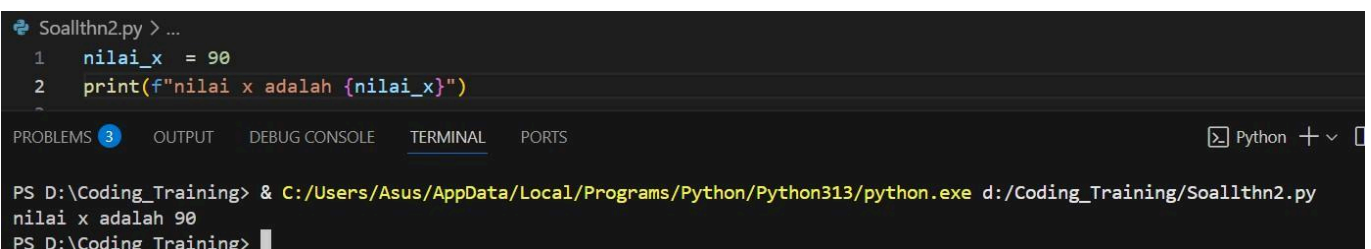
B. Debugging

Debugging adalah proses mencari letak kesalahan pada program kita. Entah itu berupa *syntax error*, *compile error* (kesalahan berupa *typo* biasanya), hingga *runtime error* (ketika program berjalan). Salah satu keuntungan bahasa interpreter adalah eksekusi kode yang baris perbaris sehingga ketika ditemukan masalah pada kode, maka proses eksekusi akan dihentikan. Berbeda dengan compiler yang akan mengubah semua baris dikompilasi terlebih dahulu sehingga lebih sulit mencari letak kesalahan di dalam *source code*.



gambar 1.8 : Contoh jenis *error* pada program python

Gambar tersebut adalah contoh *nameError* yang terjadi karena typo saat penulisan nama variabel. Seharusnya adalah “nilai_x” sesuai nama inisialisasi yang diberikan di awal.



gambar 1.9 : Contoh program setelah proses debug

Gambar tersebut adalah ketika kita sudah melaksanakan proses debugging atau memperbaiki kesalahan kita. Output akan sesuai dengan yang kita mau untuk menampilkan nilai x.

C. Jupyter Notebook

Salah satu environment python interaktif adalah Jupyter notebook yang dapat diakses melalui anaconda prompt dan sudah menjadi paket instalasi bersama anaconda. Berikut cara instalasi anaconda dan cara membuat server untuk menjalankan jupyter notebook.

1. Install package distributor anaconda dari website : <https://www.anaconda.com/download> .
2. Masuk ke lokasi penyimpanan file tersebut.
3. Pilih file dengan ekstensi .exe.
4. Konfigurasi dan install anaconda di perangkat.
5. Buka Anaconda prompt.
6. Pilih lokasi dimana kamu ingin menyimpan directory file anaconda (cd namafile atau cd..).
7. Kamu bisa pilih ingin membuat directory baru dengan (mkdir nama_directory) atau kamu tinggal menuju directory yang sudah ada.



```
Administrator: Anaconda Prompt
(base) C:\Users>cd..
(base) C:\>d:
(base) D:\>mkdir anaconda_
```

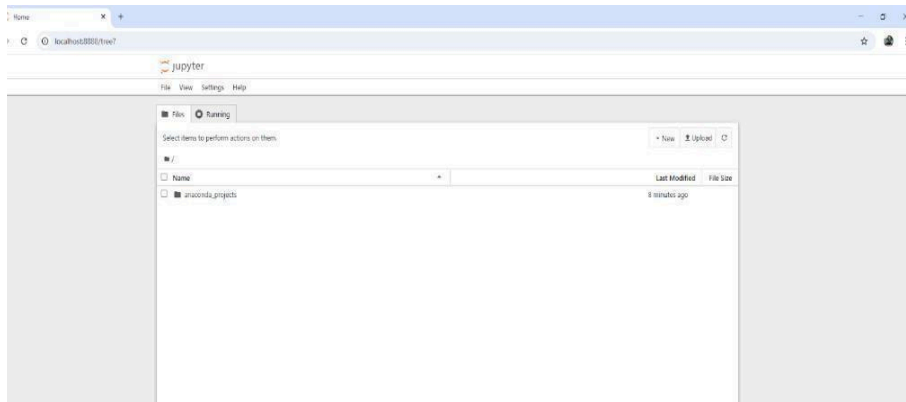
gambar 2.1 : Command untuk membuat directory baru.

8. Di sini saya beralih dari penyimpanan C ke D tempat saya membuat directory bernama "anaconda".
9. Jalankan server lokal dengan mengetikkan command seperti berikut ini.

```
Administrator: Anaconda Prompt
(base) C:\Users>cd..
(base) C:\>d:
(base) D:\>cd anaconda
(base) D:\anaconda>jupyter notebook_
```

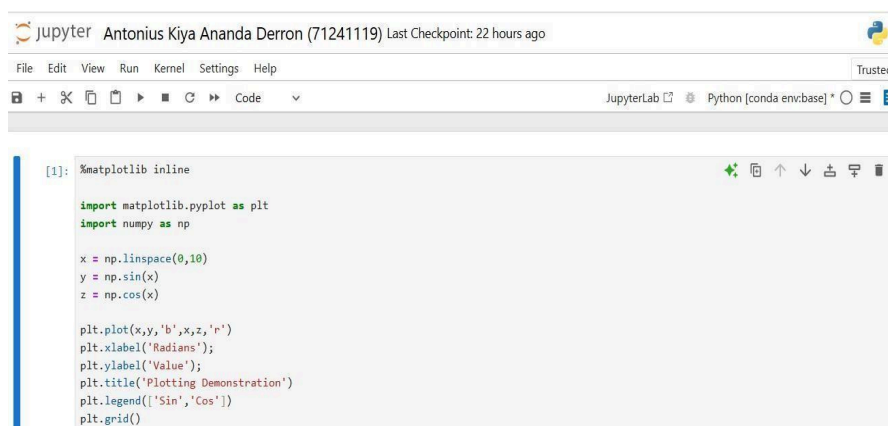
gambar 2.2 : Command untuk memulai server lokal jupyter.

10. Setelah itu Jupyter akan menginisialisasi server lokal komputer dan membukanya melalui web browser. Buka web browser tersebut.
11. Setelah itu Jupyter akan menginisialisasi server lokal komputer dan membukanya melalui web browser. Buka web browser tersebut.



gambar 2.3 : Tampilan awal jupyter notebook

12. Pada tampilan awal ini kita dapat membuka file baru yang sudah ada di *repository* kita yang memiliki ekstensi `.ipynb` atau `.pynb`.
13. Setelah itu, anda dapat memulai projek python dengan memilih bagian new dan pilih kernel seperti ipykernel atau sejenisnya.



gambar 2.4 : Tampilan awal python interaktif.

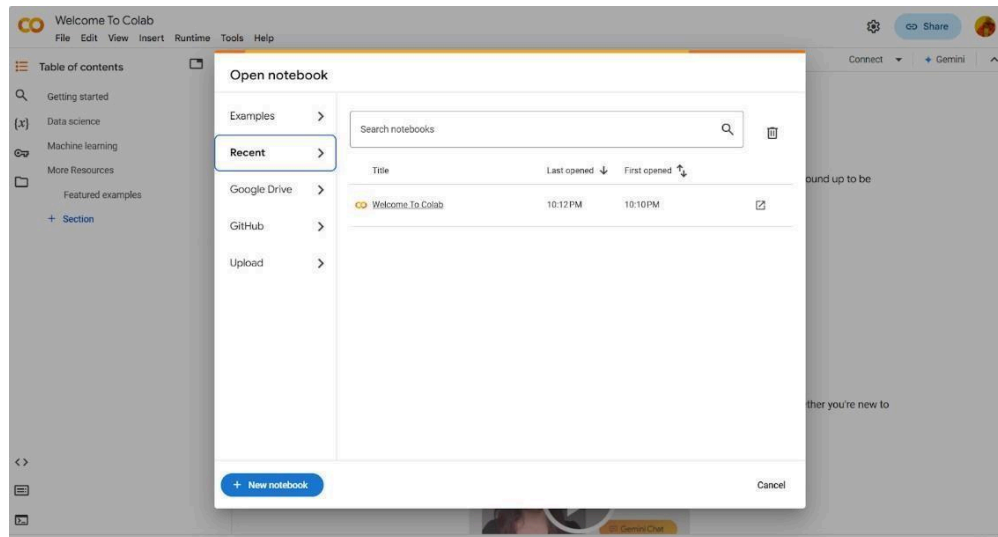
Default name di sini adalah “Untitled”. Anda dapat mengubahnya dengan menekan kotak yang berisi judul. File python yang ingin dijalankan disini harus menggunakan ekstensi `*.ipynb` atau `.pynb`.

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan *screenshot output*.

Penting : Saya menggunakan google collab karena terjadi masalah pada ipykernel saya yang tidak dapat menggunakan magic command untuk menampilkan grafik. Saya sudah coba mencocokkan versi matplotlib dengan jupyter dan python tetapi tidak berhasil.

1. Buka google collab melalui link berikut : <https://colab.google/> .
2. Login atau daftarkan akun anda lalu tekan “Open Collab”
3. Akan muncul tampilan seperti ini :

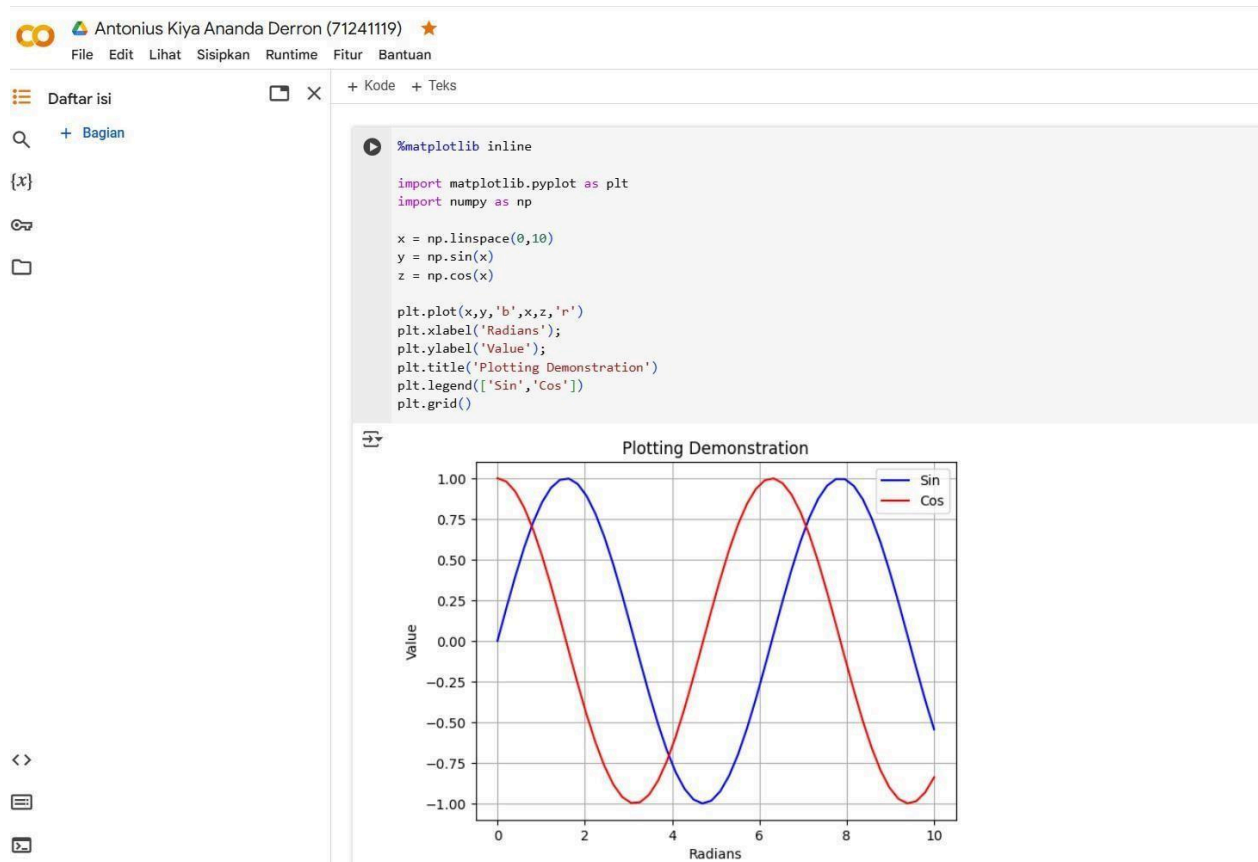


Gambar 3.1 : Tampilan dashboard google collab.

Tekan tombol “New Notebook” dan namai notebook tersebut.

4. Mulai membuat kode.
Link github : <https://github.com/leutenantKiya/PrAIPro-Minggu1>.

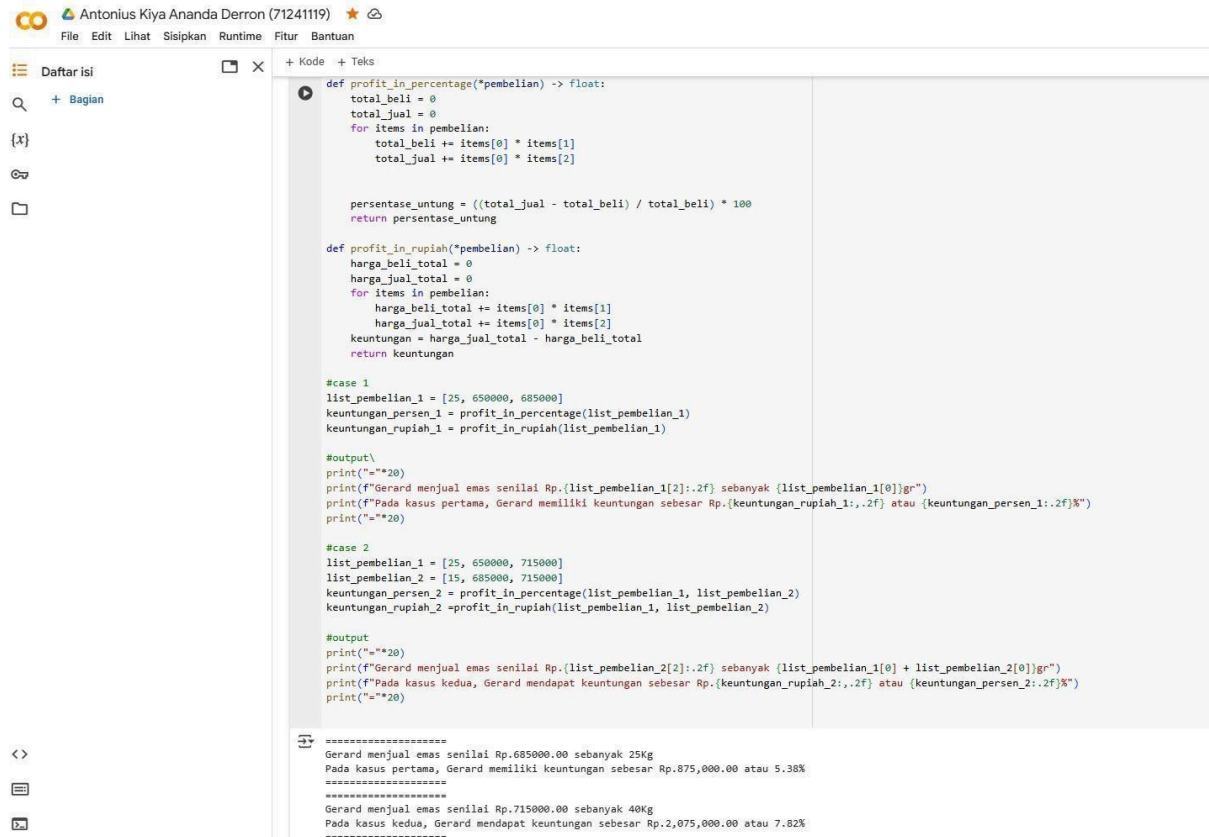
SOAL 1 (<https://github.com/leutenantKiya/PrAIPro-Minggu1.git>)



gambar 3.2 : program plot grafik pada python interaktif sumber : <https://matplotlib.org/stable/index.html> , <https://numpy.org/doc/>

1. Menggunakan magic command “%” matplotlib dengan *backend inline* guna menampilkan grafik pada *cell*.
2. Mengimport library matplotlib sebagai “plt” untuk membuat grafik dan numpy sebagai “np” untuk menentukan plot nya dalam *array*.
3. Membuat variabel x untuk menentukan sumbu horizontal grafik.
4. Membuat variabel “y” dan “z” untuk menghitung nilai sinus dan cosinus.
5. Membuat plot (plt.plot(x, y, 'b', x, z, 'r')) yang berarti akan membuat grafik dengan y sinus terhadap x menggunakan warna biru. Selanjutnya z cosinus terhadap x dengan warna merah.
6. Memberi label terhadap plot sinus dan cosinus serta memberi judul grafik.
7. plt.legend() digunakan untuk menambahkan legenda pembeda antara plot sinus dan cosinus.
8. plt.grid() digunakan untuk menambahkan grid pada grafik supaya mudah dibaca.

SOAL 2 (<https://github.com/leutenantKiya/PrAIPro-Minggu1.git>)



```
def profit_in_percentage(*pembelian) -> float:
    total_beli = 0
    total_jual = 0
    for items in pembelian:
        total_beli += items[0] * items[1]
        total_jual += items[0] * items[2]

    persentase_untung = ((total_jual - total_beli) / total_beli) * 100
    return persentase_untung

def profit_in_rupiah(*pembelian) -> float:
    harga_beli_total = 0
    harga_jual_total = 0
    for items in pembelian:
        harga_beli_total += items[0] * items[1]
        harga_jual_total += items[0] * items[2]
    keuntungan = harga_jual_total - harga_beli_total
    return keuntungan

#case 1
list_pembelian_1 = [25, 650000, 685000]
keuntungan_persen_1 = profit_in_percentage(list_pembelian_1)
keuntungan_rupiah_1 = profit_in_rupiah(list_pembelian_1)

#output\
print("\n*20")
print(f"Gerard menjual emas senilai Rp.{list_pembelian_1[2]:.2f} sebanyak {list_pembelian_1[0]}gr")
print(f"Pada kasus pertama, Gerard memiliki keuntungan sebesar Rp.{keuntungan_rupiah_1:,.2f} atau {keuntungan_persen_1:.2f}%")
print("\n*20")

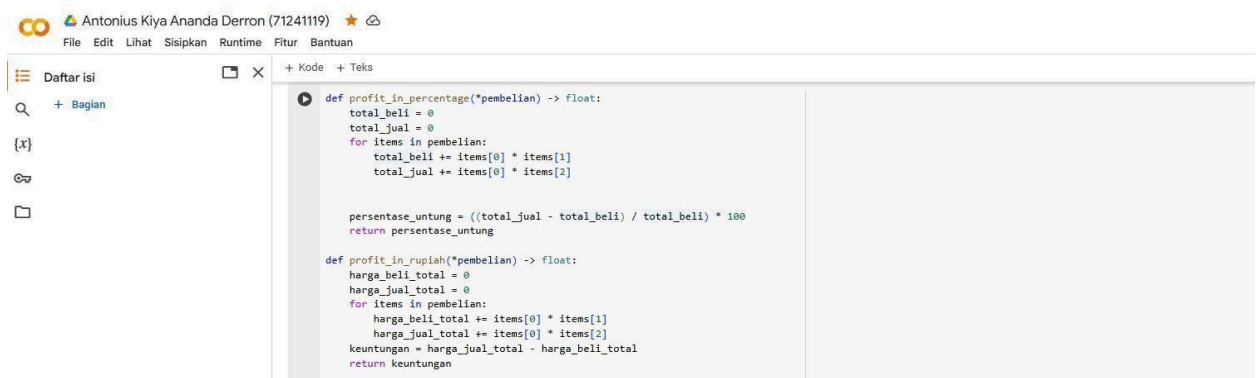
#case 2
list_pembelian_1 = [25, 650000, 715000]
list_pembelian_2 = [15, 685000, 715000]
keuntungan_persen_2 = profit_in_percentage(list_pembelian_1, list_pembelian_2)
keuntungan_rupiah_2 = profit_in_rupiah(list_pembelian_1, list_pembelian_2)

#output
print("\n*20")
print(f"Gerard menjual emas senilai Rp.{list_pembelian_2[2]:.2f} sebanyak {list_pembelian_1[0] + list_pembelian_2[0]}gr")
print(f"Pada kasus kedua, Gerard mendapat keuntungan sebesar Rp.{keuntungan_rupiah_2:,.2f} atau {keuntungan_persen_2:.2f}%")
print("\n*20")

=====
Gerard menjual emas senilai Rp.685000.00 sebanyak 25Kg
Pada kasus pertama, Gerard memiliki keuntungan sebesar Rp.875,000.00 atau 5.38%
=====
Gerard menjual emas senilai Rp.715000.00 sebanyak 40Kg
Pada kasus kedua, Gerard mendapat keuntungan sebesar Rp.2,075,000.00 atau 7.82%
=====
```

Gambar 3.3 : Soal mandiri 1.1.

1. Membuat 2 fungsi, yaitu “profit_in_percentage” dan “profit_in_rupiah” yang menerima parameter berupa list dan mengembalikan nilai berupa float.



```
def profit_in_percentage(*pembelian) -> float:
    total_beli = 0
    total_jual = 0
    for items in pembelian:
        total_beli += items[0] * items[1]
        total_jual += items[0] * items[2]

    persentase_untung = ((total_jual - total_beli) / total_beli) * 100
    return persentase_untung

def profit_in_rupiah(*pembelian) -> float:
    harga_beli_total = 0
    harga_jual_total = 0
    for items in pembelian:
        harga_beli_total += items[0] * items[1]
        harga_jual_total += items[0] * items[2]
    keuntungan = harga_jual_total - harga_beli_total
    return keuntungan
```

2. fungsi “profit_in_precentage” akan menghitung keuntungan dalam persentase, sedangkan “profit_in_rupiah” untuk keuntungan dalam rupiah.
3. Inisialisasi total harga beli dan jual sebagai 0.
4. *For loop* menjalankan perulangan dalam list untuk mengalikan index berat emas dengan harga per gram nya. Kenapa dalam loop? Karena saya memberikan dua list pada *case* kedua sehingga “pembelian” akan berupa tuple yang berisi dua list.
5. Nilai baru akan diberikan pada variabel harga beli dan jual.
6. *Return* atau kembalikan nilai ke variabel pemanggil fungsi. Dan lakukan hal yang sama untuk *case* kedua. Namun, kedua fungsi mengembalikan nilai yang berbeda

dimana fungsi satu akan mengembalikan nilai persentasenya sedangkan fungsi kedua mengembalikan pengurangan nilai jual dari nilai belinya.

```

#case 1
list_pembelian_1 = [25, 650000, 685000]
keuntungan_persen_1 = profit_in_percentage(list_pembelian_1)
keuntungan_rupiah_1 = profit_in_rupiah(list_pembelian_1)

#output\
print("\n*20)
print(f"Gerard menjual emas senilai Rp.{list_pembelian_1[2]:.2f} sebanyak {list_pembelian_1[0]}Kg")
print(f"Pada kasus pertama, Gerard memiliki keuntungan sebesar Rp.{keuntungan_rupiah_1:.2f} atau {keuntungan_persen_1:.2f}%")
print("\n*20)

#case 2
list_pembelian_1 = [25, 650000, 715000]
list_pembelian_2 = [15, 685000, 715000]
keuntungan_persen_2 = profit_in_percentage(list_pembelian_1, list_pembelian_2)
keuntungan_rupiah_2 = profit_in_rupiah(list_pembelian_1, list_pembelian_2)

#output
print("\n*20)
print(f"Gerard menjual emas senilai Rp.{list_pembelian_2[2]:.2f} sebanyak {list_pembelian_1[0] + list_pembelian_2[0]}Kg")
print(f"Pada kasus kedua, Gerard mendapat keuntungan sebesar Rp.{keuntungan_rupiah_2:.2f} atau {keuntungan_persen_2:.2f}%")
print("\n*20)

=====
Gerard menjual emas senilai Rp.685000.00 sebanyak 25Kg
Pada kasus pertama, Gerard memiliki keuntungan sebesar Rp.675,000.00 atau 5.36%
=====
Gerard menjual emas senilai Rp.715000.00 sebanyak 40Kg
Pada kasus kedua, Gerard mendapat keuntungan sebesar Rp.2,075,000.00 atau 7.82%

```

Gambar 3.2 : Soal mandiri 1.2.

7. Di luar fungsi, buat 2 list untuk pembelian pertama dan kedua. Setelah kasus pertama dipanggil dan ditampilkan pada *output*, *update* list pembelian pertama supaya harga jual menjadi Rp.715,000.00.
8. Buat 4 variabel, yaitu “keuntungan_persen_1”, “keuntungan_persen_2”, “keuntungan_rupiah_1”, dan “keuntungan_rupiah_2” untuk masing masing melakukan pemanggilan fungsi keuntungan persen dan rupiah.
9. Keluarkan *output* dengan melakukan format seperti untuk bagian keuntungan rupiah “:,.2f” dan :.2f untuk keuntungan persen dan rupiah.

SOAL 3 (<https://github.com/leutenantKiya/PrAIPro-Minggu1.git>)

```

def compound_interest(*compound_list) -> int:
    saldo_akhir = saldo_awal
    waktu = 0
    while saldo_akhir < target:
        waktu += 1
        saldo_akhir = saldo_awal * ((1 + (bunga / perhitungan_per_tahun)) ** (perhitungan_per_tahun * waktu))
    return waktu, saldo_akhir

saldo_awal = 200000000
target = 400000000
bunga = 0.1
perhitungan_per_tahun = 1
compound_list = [saldo_awal, target, bunga, perhitungan_per_tahun]

waktu, saldo_akhir = compound_interest(compound_list)
print(f"Erika menyimpan uang di deposito Pasti Cuan senilai Rp.{saldo_akhir:.2f} dengan bunga sebesar {bunga * 100}% hingga saldonya menjadi Rp.{waktu[1]:.2f} setelah {waktu[0]} tahun")

=====
Erika menyimpan uang di deposito Pasti Cuan senilai Rp.200,000,000.00
dengan bunga sebesar 10.0% hingga saldonya menjadi Rp.428,717,762.00 setelah 8 tahun

```

gambar 3.3 : Soal mandiri 1.3.

1. Inisialisasi semua variabel untuk menyimpan nilai yang dibutuhkan perhitungan.
2. Buat fungsi menghitung bunga pertahun yang menerima parameter berupa list “compound_list”.
3. Pernyataan “*compound_list” mengartikan bahwa fungsi itu menerima semua argumen dan menjadikannya sebuah tuple untuk di simpan di variabel “compound_list”.
4. Inisialisasi waktu target menjadi 0 karena kita akan mencari berapa lama waktu

yang dibutuhkan hingga saldo deposito kita menjadi target yang diinginkan.

5. "while saldo_akhir < target" adalah sebuah perulangan di dalam fungsi yang akan menghitung berapa saldo setelah waktu tertentu. Perulangan while akan berjalan selama saldo akhir belum mencapai atau melebihi target, dan akan berhenti ketika saldo akhir mencapai atau melebihi target.
6. Karena kondisi yang diinginkan adalah 400 juta secara minimum, maka perulangan terus berjalan selama saldo akhirnya belum melebihi 400 juta.
7. Di dalam perulangan, variabel waktu akan di tambahkan nilainya sebanyak satu.
8. Contoh pada perulangan pertama, saldo akhir adalah 200 juta dengan waktu 0. Maka saldo akhir akan tetap 200 juta karena dikalikan dengan 1. Lalu, waktu bertambah menjadi 1. Saldo akhir akan diperbaharui nilainya sesuai rumus yang sudah diberikan dan seterusnya hingga saldo akhir mencapai target dan waktu yang dicari pun didapat.
9. Nilai akhir waktu dikembalikan.
10. Fungsi compound interest dipanggil oleh variabel waktu di luar fungsi yang sudah diberikan argumen berupa list di variabel "compound_list".