



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

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	71251221
Nama Lengkap	RIZKY FEBRIANTO KRISTIANSYAH
Minggu ke / Materi	01 / Bahasa Pemrograman Python

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026

BAGIAN 1: BAHASA PEMROGRAMAN PYTHON

Kenapa Python?

Python adalah bahasa pemrograman tingkat tinggi yang interpreted, mendukung pemrograman berbasis objek dan bersifat dynamic semantics. Menurut survey dari Stackoverflow, Python merupakan bahasa yang paling banyak dipakai setelah Javascript (JS) karena syntax yang sederhana dan beginner friendly. Kesederhanaan syntax ini dapat dilihat dari perbandingan program “Hello World” antara Java, C, dan Python.

Pada Bahasa pemrograman Java:

```
1 public class Main {  
2     public static void main(String[] args){  
3         System.out.println("Hello World!");  
4     }  
5 }
```

Pada bahasa pemrograman C:

```
1 #include<stdio.h>  
2  
3 int main(){  
4     printf("Hello World");  
5     return 0;  
6 }
```

Pada bahasa pemrograman Python:

```
1 print("Hello World!");
```

Kelebihan dari bahasa pemrograman Python antara lain:

- Library bawaan Python sudah mencakup aspek dasar yang dibutuhkan programmer, seperti basis data, jaringan, sistem operasi, dan AI.
- Third-party library yang kaya dan beragam, seperti pandas, numpy, dan tensorflow.
- Python berlisensi open source dengan komunitas yang aktif mengembangkan fitur baru.
- Syntax yang sederhana dan menyerupai bahasa Inggris, beginner friendly.

Walaupun memiliki banyak kelebihan, Python juga memiliki kekurangan sebagai berikut:

- Saat ini Python belum mendukung pembuatan aplikasi di platform mobile.

- Konsumsi memory yang relatif besar dan processing yang lebih lambat dibanding C.

Menginstall Python 3

Untuk keperluan mata kuliah ini digunakan Python versi 3. Biasanya Python 3 sudah terinstall dan siap digunakan bila menggunakan OS Linux maupun MacOS. Bagi pengguna Windows bisa menginstall Anaconda Distribution. Pilih Anaconda Individual Edition 64-bit Graphical Installer, seperti pada Gambar 1.3.



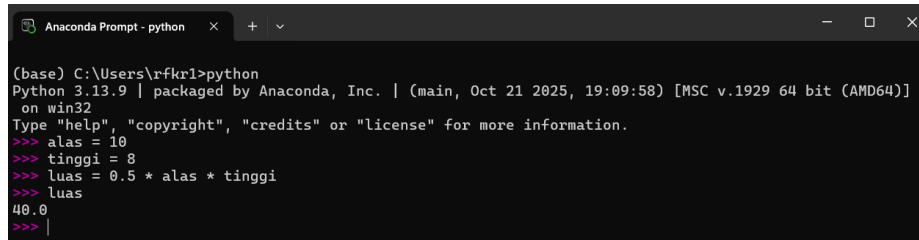
Gambar 1.1: Anaconda Distribution Individual Edition

Menjalankan Python Mode Interaktif

Untuk masuk mode interaktif di Ubuntu, jalankan perintah **python3** pada terminal. Jika menggunakan Windows, jalankan Anaconda Prompt, kemudian jalankan perintah **python**. Penggunaan mode interaktif memungkinkan programmer memasukkan perintah satu-persatu dan langsung diproses oleh interpreter Python. Contohnya, kita dapat menghitung luas segitiga dengan contoh perintah sebagai berikut:

- `alas = 10`
- `tinggi = 8`
- `luas = 0.5 * alas * tinggi`
- `luas`

Hasilnya akan seperti yang ditunjukkan pada Gambar 1.2.

A screenshot of the Anaconda Prompt terminal window. The window title is 'Anaconda Prompt - python'. The prompt shows the command 'python' being executed, which starts the Python 3.13.9 interpreter. The user enters several lines of code: 'alas = 10', 'tinggi = 8', 'luas = 0.5 * alas * tinggi', and 'luas'. The interpreter outputs '40.0' for the 'luas' variable. The prompt is currently at the next line '>>> |'.

Gambar 1.2: Menghitung luas segitiga dengan bantuan Python

Untuk mengakhiri sesi interaktif, ketik perintah **exit()** lalu klik Enter.

Editor untuk Python

Untuk menulis source code program Python, dibutuhkan editor. Beberapa editor yang biasa dipakai untuk membuat program dalam bahasa Python antara lain:

- Visual Studio Code + Python Extension for VSCode (<https://code.visualstudio.com/>)
- PyCharm (<https://www.jetbrains.com/pycharm/>).
- Spyder (<https://www.spyder-ide.org/>).
- ActivePython (<https://www.activestate.com/products/activepython/>).
- IDLE (<https://docs.python.org/3/library/idle.html>).

Selain editor, ada juga Integrated Development Environment (IDE). IDE adalah aplikasi aplikasi yang dapat digunakan untuk mengembangkan aplikasi-aplikasi yang seluruh fasilitasnya sudah tersedia dan terintegrasi. Biasanya IDE dipakai untuk project-project tingkat mid to advance. Contoh IDE adalah Pycharm dan Spyder.

Menjalankan Script Python di Terminal/Console

Python Interpreter mendukung mode script, yaitu menjalankan perintah yang sudah disimpan dalam file .py secara langsung. Script dapat dijalankan di Terminal (Ubuntu) atau Anaconda Prompt (Windows) dengan mengetik perintah **python3 namafile.py** (Ubuntu) atau **python namafile.py** (Windows). Untuk mencoba, buat file baru bernama **nilaitukardollar.py** dan ketik source code berikut di dalam file, kemudian simpan di suatu tempat.

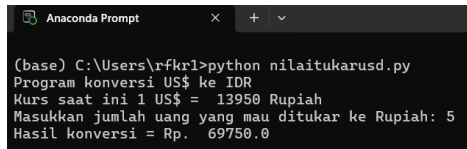
```

1 # nilai kurs 1 US$ ke IDR
2 kursud = 13950
3 # informasi program
4 print('Program konversi US$ ke IDR')
5 print('Kurs saat ini 1 US$ = ', kursud, 'Rupiah')
6 # input jumlah US$ yang mau ditukar
7 jumlahusd = float(input('Masukkan jumlah uang yang mau ditukar ke Rupiah: '))
8 # hitung nilainya dalam Rupiah
9 dalamrupiah = jumlahusd * kursud
10 # tampilkan hasilnya
11 print('Hasil konversi = Rp. ', dalamrupiah)

```

Gambar 1.3: Kode untuk nilaitukardollar.py

Untuk menjalankan script, gunakan perintah **python3 nilaitukardollar.py** (Ubuntu) atau **python nilaitukardollar.py** (Windows). Hasil eksekusi dapat dilihat pada Gambar 1.4.



```

Anaconda Prompt
(base) C:\Users\rfrkl>python nilaitukardollar.py
Program konversi US$ ke IDR
Kurs saat ini 1 US$ = 13950 Rupiah
Masukkan jumlah uang yang mau ditukar ke Rupiah: 5
Hasil konversi = Rp. 69750.0

```

Gambar 1.4: Menjalankan script Python di Anaconda Prompt

Mencari Bug dan Memperbaikinya (Debugging)

Bug berarti kesalahan dalam program. Jenis bug yang umum adalah:

- Syntax error = kesalahan dalam menulis program. Biasanya terjadi ketika ada typo terhadap nama variabel atau syntax pada program.
- Runtime error = kesalahan saat program berjalan. Biasanya terjadi karena logika atau algoritma yang salah.

Python menggunakan sistem interpreter, di mana Python akan membaca program baris demi baris dan akan berhenti jika menemukan kesalahan. Umumnya terdapat pesan error yang menyatakan di baris mana yang terjadi error.

Berikut adalah contoh kasus dari debugging.

```

belanja = 10000 #1
diskon = 30 #2

nominalDiskon = (diskon / 100) * belanja #3

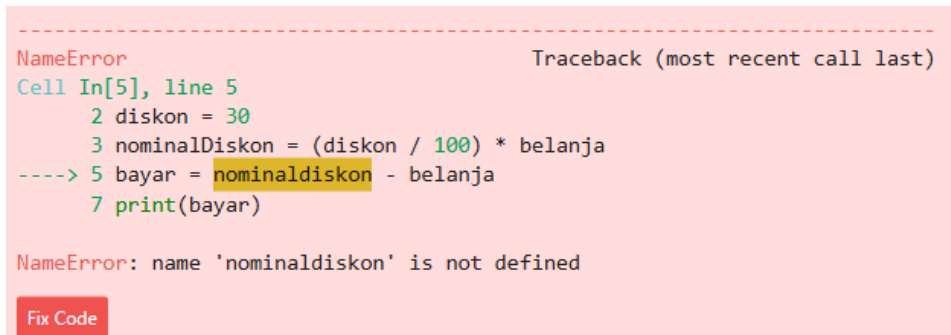
bayar = nominaldiskon - belanja #4

print(bayar) #5

```

Gambar 1.5: Test code untuk debugging

Program tersebut seharusnya dapat menghitung jumlah yang perlu dibayarkan ketika belanja Rp. 10.000 mendapat diskon 30%. Diketahui ada variabel belanja, diskon, nominalDiskon, dan bayar. Algoritmanya sudah benar, diberikan nilai variabel belanja sebesar 10000 dan diskon 30. Kemudian variabel nominalDiskon diberikan assignment yaitu $(\text{diskon}/100) * \text{belanja}$, yang akan menghasilkan nilai 3000. Selanjutnya, variabel bayar diberikan assignment $\text{nominaldiskon} - \text{belanja}$, seharusnya memiliki nilai 7000. Yang terakhir adalah perintah print untuk memberikan output berupa nilai dari variabel bayar, yaitu 7000. Tetapi ketika dirun/execute, akan muncul error message seperti ini:



```
-----  
NameError                                Traceback (most recent call last)  
Cell In[5], line 5  
      2 diskon = 30  
      3 nominalDiskon = (diskon / 100) * belanja  
----> 5 bayar = nominaldiskon - belanja  
      7 print(bayar)  
  
NameError: name 'nominaldiskon' is not defined  
  
Fix Code
```

Gambar 1.6: Error message pada Jupyter Notebook

Kenapa terjadi demikian? Dapat diamati pada error message tersebut diberikan keterangan jenis kesalahan yaitu NameError dan highlight kesalahannya. NameError merupakan **syntax error** karena ada aturan penulisan yang tidak terpenuhi. Dalam kasus ini adalah kesalahan penamaan variabel. Python mengadopsi aturan case sensitive, yaitu penggunaan huruf besar dan huruf kecil sebagai karakter sendiri. Pada line 3, variabel nominalDiskon dideklarasikan bernilai $(\text{diskon} / 100) * \text{belanja}$. Sedangkan pada line 5, variabel bayar dideklarasikan bernilai $\text{nominaldiskon} - \text{belanja}$. NameError mengatakan bahwa 'nominaldiskon' itu tidak terdefiniskan, karena sistem menganggap bahwa nominalDiskon dan nominaldiskon adalah 2 variabel berbeda. Sehingga untuk memperbaiki program ini kita perlu mengubah nama variabel nominaldiskon pada line 5 menjadi nominalDiskon.

```
belanja = 10000 #1
diskon = 30 #2

nominalDiskon = (diskon / 100) * belanja #3

bayar = nominalDiskon - belanja #4

print(bayar) #5

-7000.0
```

Gambar 1.7: Output debugging pertama

Script sudah berhasil dijalankan, tetapi outputnya masih belum sesuai harapan. Jika membeli barang seharga Rp. 10.000 dengan diskon 30% seharusnya nominal bayar sebanyak Rp. 7000. Tetapi output memberikan nilai -7000. Ini merupakan contoh kasus dari **runtime error**, yaitu kesalahan yang terjadi ketika program dijalankan. Dalam kasus ini, kesalahan logika terjadi pada line 4. Seharusnya untuk menghitung nominal pembayaran, dilakukan pengurangan antara belanja dengan nominalDiskon. Pada program, logika yang terjadi terbalik sehingga menghasilkan nilai -7000. Untuk memperbaikinya, bisa menulis ulang line 4 dengan **bayar = belanja – nominalDiskon**.

```
belanja = 10000 #1
diskon = 30 #2

nominalDiskon = (diskon / 100) * belanja #3

bayar = belanja - nominalDiskon #4

print(bayar) #5

7000.0
```

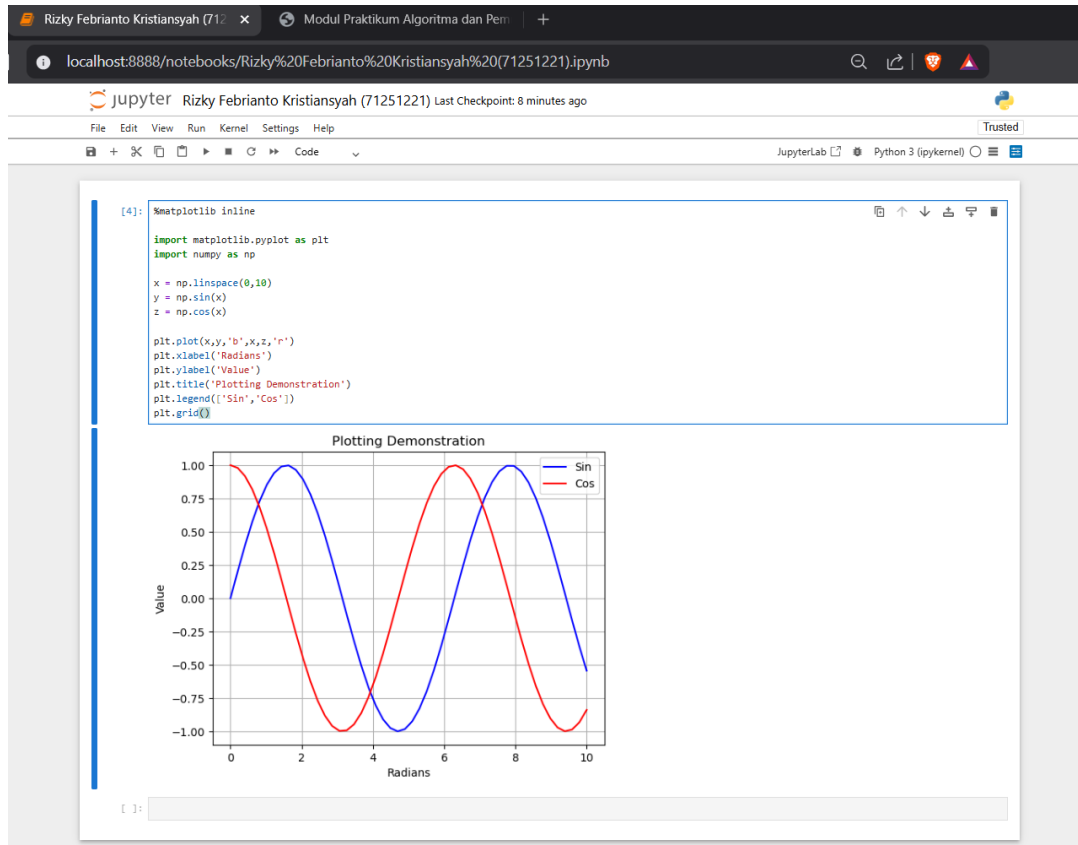
Gambar 1.8: Output debugging kedua

Setelah diperbaiki, sekarang program menghasilkan output yang diharapkan. Program berjalan dengan sukses dan tidak ada kesalahan logika maupun algoritma.

BAGIAN 2: LATIHAN MANDIRI

Link Github repository: <https://github.com/ifgodwillsit/PrakAIProsy/issues/1>

SOAL 1



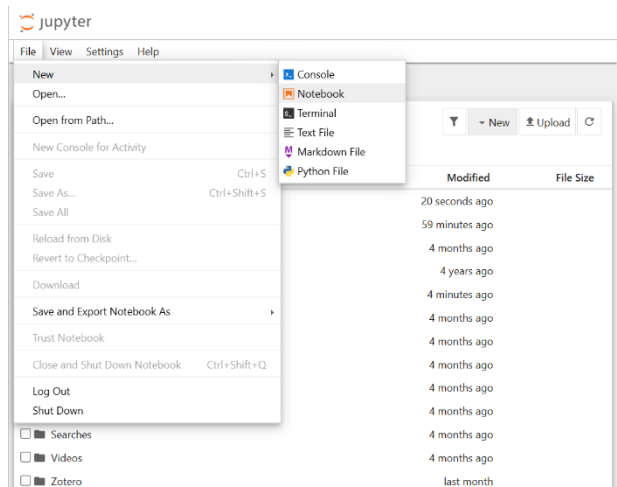
Gambar 2.1: Hasil Latihan Mandiri soal 1.

Langkah pertama adalah dengan membuka Jupyter Notebook. Pertama-tama, buka Anaconda Prompt kemudian ketik **jupyter notebook**. Setelah beberapa saat, prompt akan meredirect ke browser dan membuka Jupyter Notebook.

```
ngm prefix
(base) C:\Users\rfrkl>jupyter notebook
[2020-02-08 02:31:42.153 ServerApp] Extension package aext.assistant took 3.3708s to import
[2020-02-08 02:31:43.150 ServerApp] ***** ENVIRONMENT [Panels] Environment: PRODUCTION *****
[2020-02-08 02:31:43.161 ServerApp] ***** ENVIRONMENT [Panels] Environment: PRODUCTION *****
[2020-02-08 02:31:43.600 ServerApp] Database not initialized by migration_upgrade(). Running it again
Database not initialized by migration_upgrade(). Running it again
--> Alembic Config: {'db_conn_url': 'sqlite:///C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db'}
\\db\\project_filebrowser.db', 'db_path': 'C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db'}
Default dir: C:\\Users\\rfrkl. Running chdir...
[2020-02-08 02:31:43.682 ServerApp] Checking for database file @ C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db
er.db
[2020-02-08 02:31:43.683 ServerApp] Creating database file...
[2020-02-08 02:31:43.690 ServerApp] Migrations executed
[2020-02-08 02:31:43.732 ServerApp] --> Alembic Config: {'db_conn_url': 'sqlite:///C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db'}
\\db\\project_filebrowser.db', 'db_path': 'C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db'}
[2020-02-08 02:31:43.733 ServerApp] Checking for database file @ C:\\Users\\rfrkl\\anaconda_projects\\db\\project_filebrowser.db
er.db
[2020-02-08 02:31:43.734 ServerApp] Database file already exists
[2020-02-08 02:31:43.736 ServerApp] Migrations executed
[2020-02-08 02:31:43.758 ServerApp] Checking project configuration: C:\\Users\\rfrkl. Running chdir...
[2020-02-08 02:31:43.751 ServerApp] Initializing a new watchdog thread: []
[2020-02-08 02:31:44.125 ServerApp] Successfully initialized a watchdog thread: [16592]
[2020-02-08 02:31:44.137 ServerApp] Extension package aext.toolbox took 0.9658s to import
[2020-02-08 02:31:45.901 ServerApp] Extension package panel.io.jupyter_server_extension took 1.2251s to import
[2020-02-08 02:31:45.901 ServerApp] aext.assistant | extension was successfully linked.
[2020-02-08 02:31:45.902 ServerApp] aext.core | extension was successfully linked.
[2020-02-08 02:31:45.902 ServerApp] aext.panels | extension was successfully linked.
[2020-02-08 02:31:45.902 ServerApp] aext.share_notebook | extension was successfully linked.
```

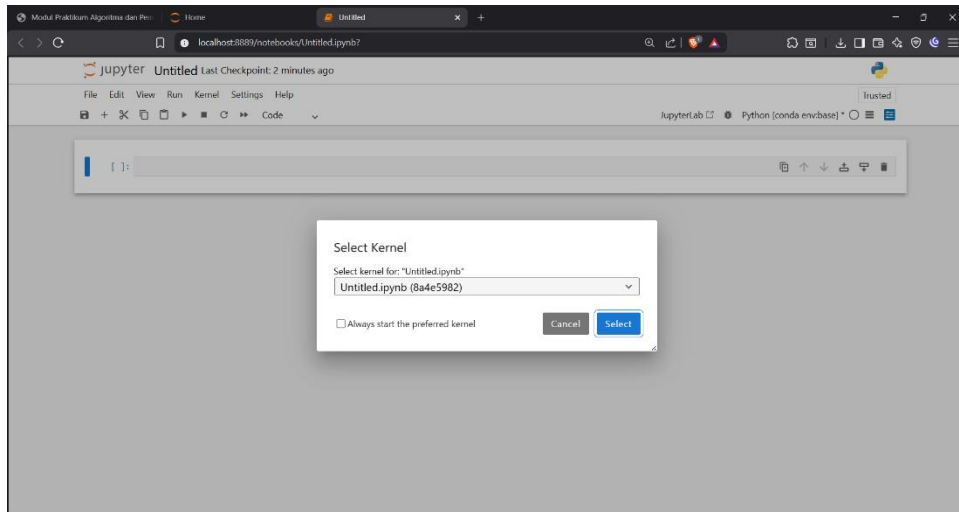
Gambar 2.1: Tampilan Anaconda Prompt setelah mengeksekusi perintah jupyter notebook.

Pada halaman Home Jupyter Notebook, klik menu File, kemudian New, kemudian klik Notebook, untuk membuat Notebook baru.



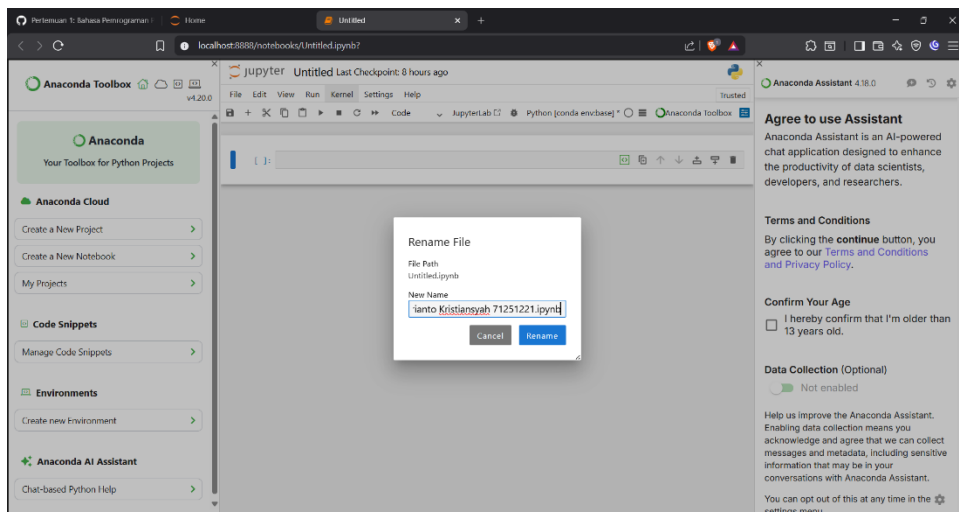
Gambar 2.2: Tampilan Home Jupyter Notebook.

Setelah itu akan redirect menuju Notebook yang kosong. Jupyter Notebook akan meminta user untuk memilih kernel, pilih **Python [Conda env: base]** kemudian klik Select.



Gambar 2.3: Pemilihan kernel pada Jupyter Notebook

Untuk mengganti nama file, klik pada Untitled, kemudian ketik Nama (NIM) seperti ini:



Gambar 2.4: Tampilan Rename file

Kemudian, ketikkan perintah berikut:

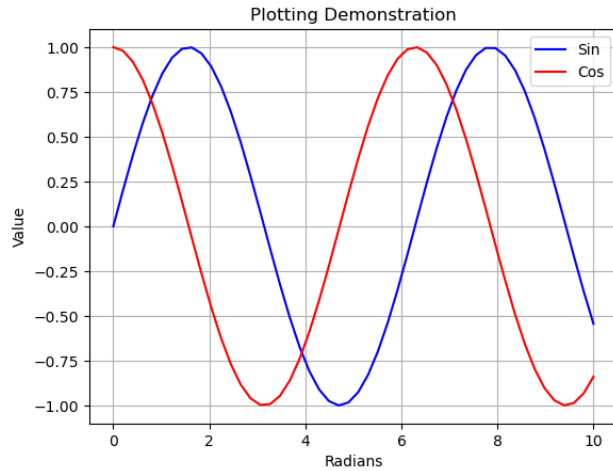
```
%matplotlib inline #1
import matplotlib.pyplot as plt #2
import numpy as np #3
x = np.linspace(0,10) #4
y = np.sin(x) #5
z = np.cos(x) #6
plt.plot(x,y,'b',x,z,'r') #7
plt.xlabel('Radians') #8
plt.ylabel('Value') #9
plt.title('Plotting Demonstration') #10
plt.legend(['Sin','Cos']) #11
plt.grid() #12
```

Gambar 2.5: Source code grafik sinus dan cosinus

Baris pertama berfungsi untuk menampilkan grafik yang dibuat ditampilkan langsung di bawah cell code (inline). Baris kedua berfungsi untuk mengimport library matplotlib.pyplot dengan alias plt. Library tersebut digunakan untuk membuat visualisasi data seperti grafik, chart, dan lain sebagainya. Baris ketiga berfungsi untuk mengimport library numpy dengan alias np. Library numpy digunakan untuk komputasi numerik dan manipulasi array.

Baris keempat membuat array nilai x dari 0 sampai 10, array ini akan menjadi sumbu x pada grafik. Baris kelima membuat array nilai y yang berisi nilai sin. Baris keenam membuat array nilai z yang berisi nilai cosinus. Baris ketujuh membuat grafik dengan 2 kurva sekaligus. Kurva pertama berwarna biru ('b') dengan x sebagai sumbu horizontal dan y sebagai sumbu vertikal sehingga membentuk kurva sinus. Sementara kurva kedua berwarna merah ('r') dengan x sebagai sumbu horizontal dan z sebagai sumbu vertikal sehingga membentuk kurva cosinus.

Baris kedelapan memberi label "Radian" di bawah sumbu x (horizontal). Sedangkan baris kesembilan memberi label "Values" di samping sumbu y (vertikal). Baris kesepuluh memberikan judul "Plotting Demonstration" di atas grafik. Baris kesebelas menambahkan legenda pada grafik, dengan informasi kurva pertama bernama sin berwarna biru dan kurva kedua bernama cos berwarna merah. Baris keduabelas menambahkan grid/kotak-kotak pada grafik. Hasil output dari kode tersebut akan menghasilkan grafik sebagai berikut:



Gambar 2.6: Output grafik sinus dan cosinus.

SOAL 2

```
[25]: emasAwal = 25
asetAwal = emasAwal * 650000
asetNaik1 = emasAwal * 685000

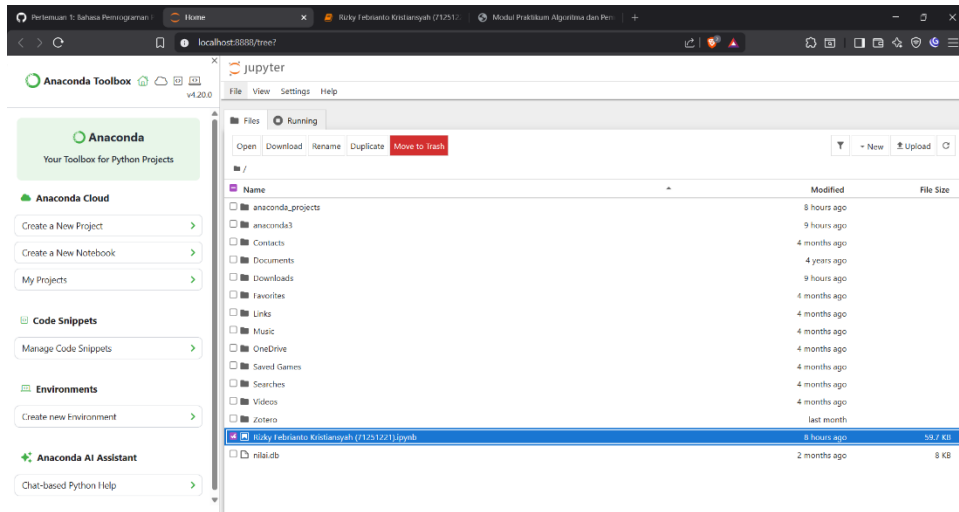
untungRp1 = asetNaik1 - asetAwal
untungPrsn1 = (untungRp1 / asetAwal) * 100
print('Keuntungan yang didapat oleh Gerard sebesar Rp.', untungRp1, 'atau', round(untungPrsn1,2), '%')

dcaEmas = asetAwal + 15 * 685000
asetNaik2 = 40 * 715000
untungRp2 = asetNaik2 - dcaEmas
untungPrsn2 = (untungRp2 / dcaEmas) * 100
print('Keuntungan yang didapat Gerard setelah dca emas di harga Rp.685000 adalah Rp.', untungRp2, 'atau', round(untungPrsn2,2), '%')

Keuntungan yang didapat oleh Gerard sebesar Rp. 875000 atau 5.38 %
Keuntungan yang didapat Gerard setelah dca emas di harga Rp.685000 adalah Rp. 2075000 atau 7.82 %
```

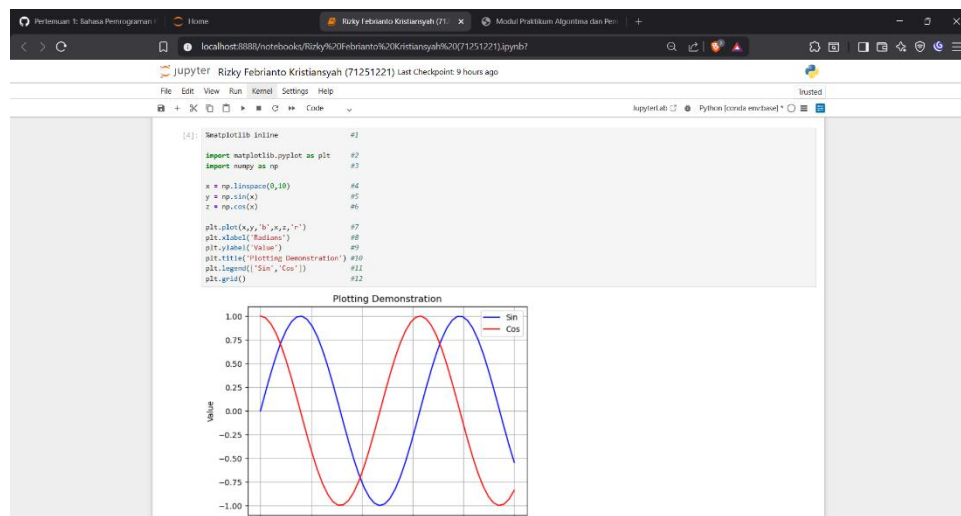
Gambar 2.7: Hasil Latihan Mandiri Soal 2

Pertama-tama buka Jupyter Notebook lewat Anaconda Prompt dan menetik **jupyter notebook**. Setelah redirect ke halaman Home, klik file latihan mandiri sebelumnya yang dinamai Nama (NIM).ipynb seperti pada gambar berikut.



Gambar 2.8: Tampilan Home Jupyter Notebook

Setelah redirect ke file notebook tersebut, akan muncul tampilan seperti ini:



Gambar 2.9: Tampilan file Nama(NIM).ipynb di Jupyter Notebook

Pada code cell selanjutnya, ketik kode berikut:

```
emasAwal = 25 #1
asetAwal = emasAwal * 650000 #2
asetNaik1 = emasAwal * 685000 #3

untungRp1 = asetNaik1 - asetAwal #4
untungPrsn1 = (untungRp1 / asetAwal) * 100 #5
print('Keuntungan yang didapat oleh Gerard sebesar Rp.', untungRp1, 'atau', round(untungPrsn1,2), '%') #6

dcaEmas = asetAwal + 15 * 685000 #7
asetNaik2 = 40 * 715000 #8
untungRp2 = asetNaik2 - dcaEmas #9
untungPrsn2 = (untungRp2 / dcaEmas) * 100 #10
print('Keuntungan yang didapat Gerard setelah dca emas di harga Rp.685000 adalah Rp.', untungRp2, 'atau', round(untungPrsn2,2), '%') #11
```

Gambar 2.10: Source code untuk soal 2

Diketahui bahwa Gerard membeli emas sebanyak 25 gram di harga Rp. 650.000. Sehingga dapat diketik dalam kode bahwa variabel `emasAwal` senilai 25, dengan variabel `asetAwal` senilai $\text{emasAwal} (25\text{g}) * 650000$ (harga kala itu) dengan total aset Rp. 16.250.000. Kemudian ketika emas di harga Rp. 685.000, dibuat variabel `asetNaik1` senilai $\text{emasAwal} (25\text{g}) * 685000$ (harga naik pertama) yaitu Rp. 17.125.000. Untuk menghitung keuntungan dalam rupiah, dibuat variabel `untungRp1` yang bernilai $\text{asetNaik1} - \text{asetAwal}$ dengan hasil Rp. 875.000. Untuk menghitung keuntungan dalam persentase, dibutuhkan variabel baru yaitu `untungPrsn1` senilai $(\text{untungRp1} / \text{asetAwal}) * 100$ yang menghasilkan nilai 5.38.

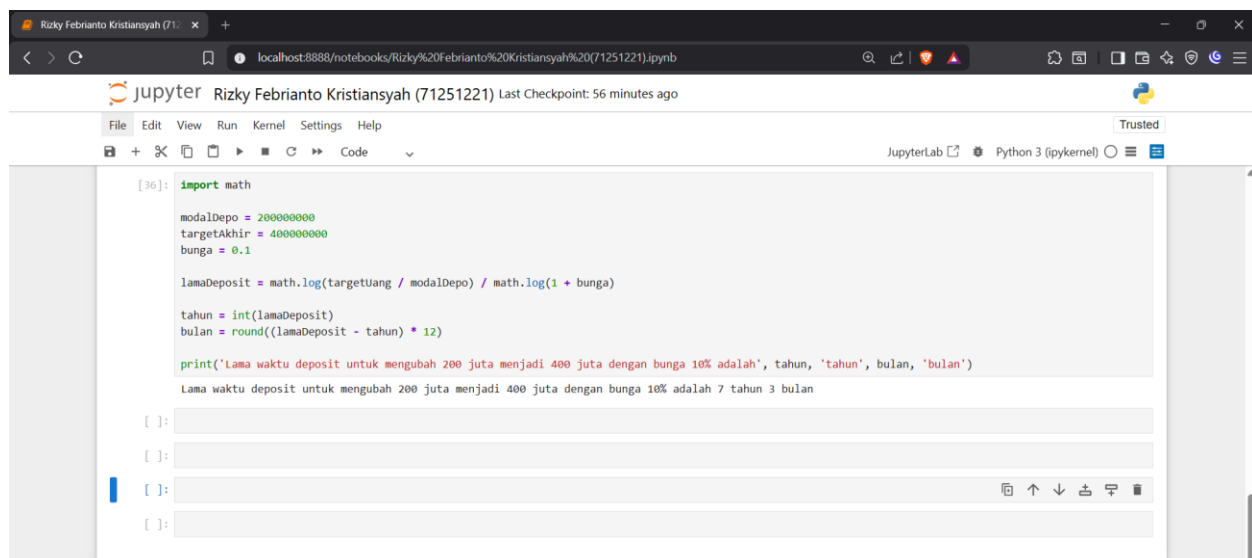
Kasus kedua adalah Gerard membeli emas lagi sebanyak 15 gram di harga Rp. 850.000. Untuk menghitung nilai aset emas Gerard saat ini bisa dibuat variabel baru dengan nama `dcaEmas` yang berisi $\text{asetAwal} (\text{senilai Rp. 16.250.000}) + 15 * 685000$, menghasilkan nilai aset Rp. 26.525.000. Untuk menghitung nilai aset emas di harga Rp. 715000 bisa dibuat variabel `asetNaik2` yang berisi nilai $40 (\text{total emas}) * 715000$ yaitu Rp. 28.600.000. Untuk menghitung keuntungan dalam rupiah, dibutuhkan variabel baru yang diberi nama `untungRp2` yang bernilai $\text{asetNaik2} (\text{Rp. 28.600.000}) - \text{dcaEmas} (\text{Rp. 26.525.000})$, menghasilkan nilai Rp. 2.075.000. Untuk menghitung persentase keuntungan, dibutuhkan variabel baru yaitu `untungPrsn2` yang diisi $(\text{untungRp2} / \text{dcaEmas}) * 100$, menghasilkan nilai 7,82%. Untuk menampilkan nilai-nilai tersebut, tidak lupa untuk melakukan perintah `print`, seperti pada line 6 dan line 11.

Keuntungan yang didapat oleh Gerard sebesar Rp. 875000 atau 5.38 %

Keuntungan yang didapat Gerard setelah dca emas di harga Rp.685000 adalah Rp. 2075000 atau 7.82 %

Gambar 2.11: Output kode

SOAL 3



```
[36]: import math

modalDepo = 200000000
targetAkhir = 400000000
bunga = 0.1

lamaDeposit = math.log(targetUang / modalDepo) / math.log(1 + bunga)

tahun = int(lamaDeposit)
bulan = round((lamaDeposit - tahun) * 12)

print('Lama waktu deposit untuk mengubah 200 juta menjadi 400 juta dengan bunga 10% adalah', tahun, 'tahun', bulan, 'bulan')
Lama waktu deposit untuk mengubah 200 juta menjadi 400 juta dengan bunga 10% adalah 7 tahun 3 bulan
```

Gambar 2.12: Hasil Latihan soal 3

Pertama-tama buka Jupyter Notebook lewat Anaconda Prompt dan mengetik **jupyter notebook**. Setelah redirect ke halaman Home, klik file latihan mandiri sebelumnya yang dinamai Nama (NIM).ipynb. Pada cell code berikutnya, ketik kode berikut:

```
import math #1

modalDepo = 200000000 #2
targetAkhir = 400000000 #3
bunga = 0.1 #4

lamaDeposit = math.log(targetUang / modalDepo) / math.log(1 + bunga) #5

tahun = int(lamaDeposit) #6
bulan = round((lamaDeposit - tahun) * 12) #7

print('Lama waktu deposit untuk mengubah 200 juta menjadi 400 juta dengan bunga 10% adalah', tahun, 'tahun', bulan, 'bulan') #8
```

Gambar 2.13: Source code untuk soal 3

Diketahui bahwa Erika memiliki modal untuk deposit senilai Rp. 200 juta dengan bunga 10% APY. Erika ingin target uang yang dia depositkan menjadi minimal Rp. 400 juta. Untuk mencari lama deposit (tahun), dapat menggunakan formula compound interest $A = P(1 + r/n)^{nt}$, dengan keterangan sebagai berikut: A adalah saldo akhir, P adalah modal, r adalah besar bunga, n adalah jumlah perhitungan bunga (1 tahun sekali), dan t adalah jumlah periode. Diketahui untuk A adalah Rp. 400 juta dan P adalah Rp. 200 juta, dengan bunga 10% setiap tahun. Karena dicari jumlah periode atau lama depositnya, dapat mengubah formula tersebut menjadi $t = \log(P/A) / \log(1+r)$, untuk bentuk kodenya ada pada line 5. Karena membutuhkan fungsi logaritma, maka di baris 1 diperlukan untuk mengimport library math. Setelah memproses lamaDeposit dan mendapatkan hasil yang bernilai 7.2725..., line 6 berfungsi untuk mengubah nilai float tersebut menjadi integer sebagai nilai dari variabel tahun. Line 7 berfungsi untuk mencari 7,2725... tahun itu 7 tahun berapa bulan, menghasilkan nilai 3 sebagai nilai untuk variabel bulan. Line ke-8 berfungsi untuk memberikan output dari program ini.

```
Lama waktu deposit untuk mengubah 200 juta menjadi 400 juta dengan bunga 10% adalah 7 tahun 3 bulan
```

Gambar 2. 14: Output kode