



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	<ISI DENGAN NIM ANDA>
Nama Lengkap	<ISI DENGAN NAMA LENGKAP ANDA>
Minggu ke / Materi	01 / Bahasa Pemrograman Python

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

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

Kenapa Python?

Python adalah bahasa pemrograman level tinggi yang interpreted, mendukung Object Oriented Programming (OOP) dan memiliki sifat dynamic semantics. Berdasarkan survei yang dilakukan oleh Stackoverflow pada 2019

(<https://insights.stackoverflow.com/survey/2019#technology>), Python menempati posisi kedua sebagai salah satu Bahasa paling banyak digunakan setelah JavaScript, dan secara umum termasuk bahasa yang sangat populer di dunia industri maupun akademik. Popularitas ini terutama didorong oleh aturan penulisan dan sintaks Python yang sederhana, sehingga mudah dipahami bagi para pemula yang baru belajar bahasa pemrograman.

Kesederhanaan Python tampak jelas jika dibandingkan dengan bahasa seperti Java atau C saat menulis program sederhana, misalnya "Hello World". Pada bahasa pemrograman Java, programmer perlu menulis deklarasi kelas, fungsi main, dan pemanggilan `System.out.println`. Pada bahasa pemrograman C, harus menuliskan `#Include`, fungsi main, dan pemanggilan `printf` beserta penanganan nilai kembalian. Pada Python, program yang sama cukup ditulis dengan satu baris `printf("Hello World!")`, tanpa deklarasi yang panjang. Perbedaan yang sangat jelas pada Python, Java dan C yang menunjukkan bahwa Python dibuat agar tidak banyak aturan yang rumit, sehingga programmer dapat lebih fokus memikirkan apa yang ingin dilakukan program, bukan sibuk dengan penulisannya.

Kelebihan dari Python:

- Dukungan pustaka atau ekosistem pustaka yang kaya, Python memiliki dukungan pustaka pihak ketiga yang sangat beragam. Ketersediaan pustaka seperti *pandas*, *numpy*, *tensorflow*, dan *keras* membuat Python menjadi pilihan utama di bidang *data science* dan *machine learning*.

- Pustaka bawaan yang lengkap, Python telah dilengkapi dengan pustaka standar yang mencakup berbagai aspek pemrograman fundamental, mulai dari manipulasi string, pengaksesan basis data, hingga operasi jaringan.
- *Lisensi Open Source*, Python dapat digunakan secara bebas untuk berbagai keperluan, termasuk proyek komersial, tanpa biaya lisensi. Komunitas pengembangannya yang aktif juga memastikan bahasa pemrograman Python terus berkembang.
- Mudah dipelajari, sintaks Python yang sederhana dan mirip dengan bahasa Inggris membuatnya relatif mudah dipelajari, terutama bagi pemula. Dokumentasi yang lengkap, tutorial yang melimpah, serta berbagai *online course* tentang Python yang tersedia di Internet sangat memudahkan siapa saja yang ingin mempelajari Python, baik secara otodidak maupun melalui jalur formal.

Kekurangan dari Python:

- Belum mendukung *mobile development*, saat ini Python belum mendukung untuk pembuatan aplikasi *mobile native* di platform seperti Android atau iOS. Untuk pengembangan aplikasi mobile, bahasa seperti Java/Kotlin (Android) atau Swift (iOS) masih menjadi pilihan utama.
- Konsumsi memori yang besar, Python cenderung mengkonsumsi memori yang relatif besar dibandingkan bahasa pemrograman lain. Hal ini membuat Python kurang cocok untuk aplikasi-aplikasi yang membutuhkan efisiensi memori tinggi atau bekerja dengan data dalam jumlah yang sangat besar dengan resource terbatas.
- Kecepatan Proses, Python umumnya lebih lambat dibanding bahasa seperti C, terutama untuk perhitungan yang sangat mendalam.

Namun untuk pemula, kelemahan-kelemahan ini biasanya tidak terlalu relevan karena fokus utamanya masih pada pemahaman konsep algoritma dan logika pemrograman. Python justru dipilih karena dapat mempercepat proses belajar dibanding bahasa lain.

Menginstall Python 3

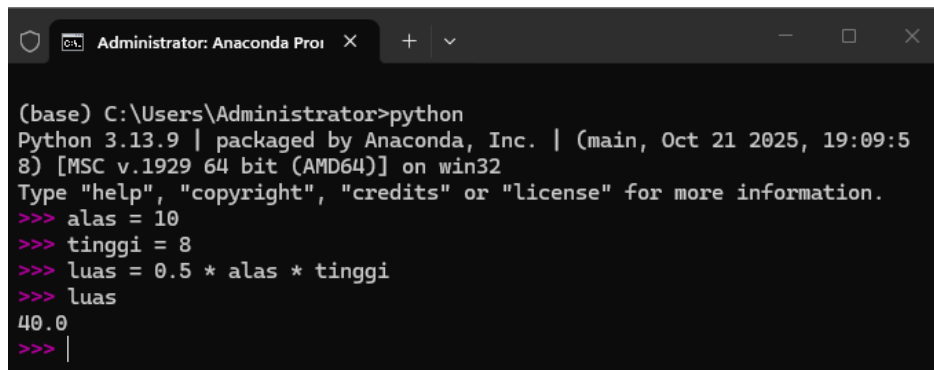
Python memiliki dua versi utama yang masih digunakan, yaitu Python 2 dan Python 3. Untuk pembelajaran dan pengembangan aplikasi modern, disarankan menggunakan Python 3 karena merupakan versi yang lebih baru dan terus dikembangkan, sementara Python 2 sudah tidak lagi didukung secara resmi.

Pada sistem operasi Linux dan macOS, Python umumnya sudah tersedia secara bawaan dan dapat langsung dipanggil dari terminal. Di Ubuntu, Python 3 dapat dijalankan dengan perintah `python3`. Jika Python sudah terinstal, akan muncul informasi versi Python yang terpasang beserta prompt interaktif yang siap menerima perintah.

Untuk pengguna Windows, pendekatan yang disarankan adalah menggunakan distribusi Anaconda. Anaconda Individual Edition menyediakan *installer grafis* 64-bit yang memudahkan instalasi Python beserta paket-paket pendukung yang sering dipakai untuk data science. Pada saat modul disusun, versi Python yang tersedia di Anaconda adalah Python 3.9, dan pengguna diminta menyesuaikan unduhan dengan sistem operasi yang digunakan. Dengan Anaconda, banyak paket penting yang sudah terinstal sejak awal, sehingga mahasiswa tidak perlu mengatur Python dari nol.

Menjalankan Python Mode Interaktif

Mode interaktif merupakan salah satu fitur Python yang sangat berguna untuk pembelajaran dan eksperimen cepat. Dalam mode interaktif dapat menjalankan perintah Python satu persatu dan langsung melihat hasilnya. Pada Ubuntu, mode interaktif dicapai dengan menjalankan `python3` di terminal. Sedangkan pada Windows, diminta membuka Anaconda Prompt lalu mengetik `python` untuk masuk ke antarmuka interpreter Python. Setelah perintah dijalankan, layar akan menampilkan informasi versi Python serta prompt siap pakai yang menandakan interpreter siap menerima instruksi baris demi baris. Mode interaktif memungkinkan pengguna menulis satu persatu dan langsung mendapatkan hasil eksekusi. Sebagai contoh menghitung luas segitiga dengan alas 10 cm dan tinggi 8 cm.



```
(base) C:\Users\Administrator>python
Python 3.13.9 | packaged by Anaconda, Inc. | (main, Oct 21 2025, 19:09:58) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> alas = 10
>>> tinggi = 8
>>> luas = 0.5 * alas * tinggi
>>> luas
40.0
>>> |
```

Gambar 1.1: Menghitung Luas Segitiga dengan bantuan Python

Baris Pertama dan kedua adalah operasi *assignment* (penugasan nilai), yaitu proses memberi nilai ke variabel panjang dengan nilai 10 dan variabel tinggi diisi dengan nilai 8. Sedangkan baris ketiga menghitung hasil berdasarkan variabel luas dengan hasil perhitungan $0.5 * \text{panjang} * \text{tinggi}$. Baris terakhir memanggil nama variabel untuk melihat isi nilai luas.

Untuk mengakhiri sesi mode interaktif, cukup mengetik `exit()` dan tekan enter. Python akan menutup sesi interaktif dan kembali ke command prompt sistem operasi.

Editor untuk Python

Untuk menulis program Python yang lebih kompleks, kita memerlukan editor atau Integrated Development Environment (IDE) yang menyediakan berbagai fitur untuk mempermudah proses coding. Semakin canggih editor yang digunakan, akan semakin mudah dan efisien proses pembuatan program.

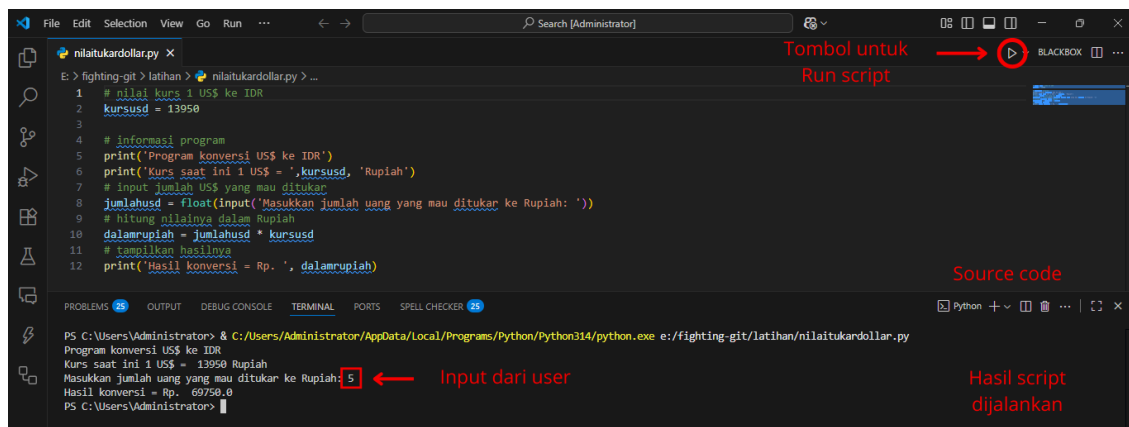
Pilihan editor yang sering digunakan untuk pengembangan Python:

- Visual studio Code (VS Code): editor ringan dengan Python Extension yang menyediakan fitur *syntax highlighting*, *code completing*, dan *debugging*.
- PyCharm: IDE lengkap khusus untuk Python dengan fitur yang sangat lengkap.
- Spyder: IDE untuk scientific computing yang umumnya sudah terinstal dengan Anaconda.
- IDLE: editor bawaan Python yang sederhana dan cocok untuk pemula.

PyCharm dan Spyder adalah Integrated Development Environment (IDE) yang menyediakan semua fasilitas pengembangan dalam satu paket terintegrasi. Untuk keperluan praktikum, Visual Studio Code sudah cukup memadai karena tampilan yang sederhana dan fiturnya mencukupi untuk pembelajaran.

Menjalankan Script Python di Terminal/Console

Selain mode interaktif, Python dapat dijalankan dalam mode script, yaitu menjalankan perintah-perintah yang sudah disimpan di file .py. Dengan mode script, pengguna dapat menulis program sekali dan menjalankannya berulang kali tanpa harus mengetik ulang setiap perintah. Sebagai contoh mencari nilai tukar dollar ke rupiah:



Gambar 1.2: Menjalankan script Python di Visual Studio Code

Jika menggunakan editor seperti Visual Studio Code, pengguna tidak harus selalu membuka terminal secara manual. VS Code menyediakan tombol Run yang dapat langsung mengeksekusi script Python, sekaligus menampilkan input dan output melalui terminal internal editor. Ini membuat alur kerja lebih efisien, terutama saat sedang sering melakukan perubahan kecil dan pengujian ulang kode.

Mencari Bug dan Memperbaikinya (debugging)

Bug adalah istilah dalam dunia pemrograman yang merujuk pada kesalahan dalam proses. Proses menemukan dan memperbaiki bug disebut debugging, yang merupakan keterampilan penting yang harus dikuasai setiap programmer.

Kesalahan dalam program dapat dikategorikan menjadi dua jenis:

1. Kesalahan penulisan (syntax error atau compile error). Kesalahan dalam penulisan sintaks program yang melanggar aturan bahasa pemrograman seperti lupa tanda baca, salah menulis nama fungsi atau keyword, indentasi yang salah. Kesalahan pada penulisan relatif mudah ditemukan karena interpreter atau editor akan memberikan pesan error yang jelas, bahkan sering kali menandai baris yang salah.
2. Kesalahan saat program berjalan (runtime error). Kesalahan yang terjadi saat program berjalan atau kesalahan logika yang membuat program menghasilkan output yang salah. Kesalahan ini lebih sulit ditemukan dan diperbaiki karena hasil yang salah tidak selalu jelas terlihat, program dapat berjalan tanpa *error message*, memerlukan pemahaman logika program, dan perlu testing yang menyeluruh.

Python sebagai bahasa yang diinterpretasi mengeksekusi kode baris demi baris. setiap baris perlu dicek terlebih dahulu sintaksnya, jika sudah valid maka baris tersebut akan langsung dijalankan. Jika terjadi error saat eksekusi, maka interpreter akan menghentikan program dan menampilkan pesan kesalahan. Setelah kesalahan sintaks diperbaiki, program dapat berjalan hingga selesai.

Debugging tidak sekadar memperbaiki kesalahan ketik, tetapi juga memeriksa apakah langkah-langkah algoritma dan relasi antar variabel sudah benar. Editor modern seperti Visual Studio Code dapat membantu mendeteksi syntax error sebelum program dijalankan, namun untuk runtime error, programmer tetap harus menguji dan menalar logika programnya sendiri. Dengan demikian, kemampuan membaca pesan error, menelusuri baris kode yang bermasalah, dan menguji ulang perhitungan menjadi keterampilan penting dalam belajar pemrograman.

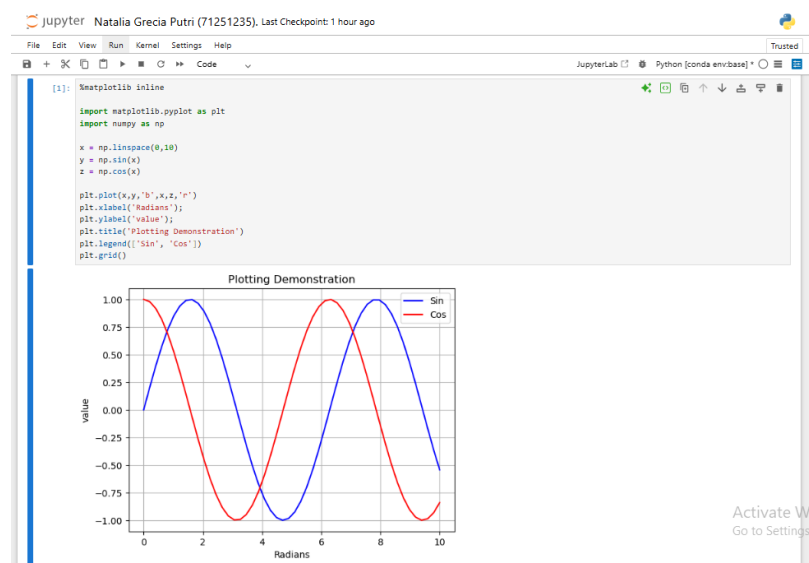
BAGIAN 2: LATIHAN MANDIRI (60%)

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.

SOAL 1

<https://github.com/grec12/PrAIPro-Minggu-1.git>

Notebook pada Jupyter



Gambar 1.3: Menggunakan Jupyter Notebook untuk menghasilkan grafik.

Penjelasan:

1. Instalasi Package Pendukung

Untuk membuat grafik, dibutuhkan dua pustaka (*library*) utama: Matplotlib untuk visualisasi data dan NumPy untuk operasi matematika dan pembuatan array. Meski umumnya Anaconda sudah menyertakannya, kita akan menunjukkan cara instalasi menggunakan pip, merupakan manajer package untuk Python

```
(base) C:\Users\Administrator>pip install matplotlib numpy
Requirement already satisfied: matplotlib in c:\anaconda\lib\site-packages (3.10.6)
Requirement already satisfied: numpy in c:\anaconda\lib\site-packages (2.3.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\anaconda\lib\site-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\anaconda\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\anaconda\lib\site-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\anaconda\lib\site-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in c:\anaconda\lib\site-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in c:\anaconda\lib\site-packages (from matplotlib) (12.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\anaconda\lib\site-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\anaconda\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\anaconda\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```


Gambar 1.4: Instal Matplotlib dan NumPy di Anaconda Prompt

2. Mengimpor Library ke dalam Notebook

```
%matplotlib inline  
  
import matplotlib.pyplot as plt  
import numpy as np
```

Gambar 1.5: Import Matplotlib dan NumPy ke dalam Notebook

Sebelum menggunakan fungsi-fungsi dari Matplotlib dan NumPy, pengguna harus memanggilnya (mengimpor) terlebih dahulu ke dalam notebook. Gunakan `plt` untuk `matplotlib.pyplot` dan `np` untuk `numpy` agar penulisannya lebih singkat dan umum digunakan.

3. Data untuk Grafik

```
x = np.linspace(0,10)  
y = np.sin(x)  
z = np.cos(x)
```

Gambar 1.5: input data grafis

- Data sumbu X akan membuat serangkaian angka dari 0 hingga 2π (kira-kira 6.28). NumPy memiliki fungsi *linspace* yang berguna untuk keperluan membuat grafik.
- Data sumbu Y akan dihitung dengan menerapkan fungsi sinus pada setiap nilai data yang ada pada sumbu X. Fungsi `np.sin` dari *library* NumPy akan melakukan perhitungan matematis secara efisien.
- Data Z: Selain fungsi sinus, grafik juga akan menampilkan gelombang kosinus. Cara pembuatannya mirip dengan perhitungan sinus. Menggunakan fungsi `np.cos()` dari pustaka NumPy untuk menghitung nilai kosinus dari setiap nilai pada array `x`. Hasil perhitungan tersebut disimpan dalam sebuah variabel baru, `z`.

4. Membuat dan Menampilkan Grafik

```
plt.plot(x,y,'b',x,z,'r')
plt.xlabel('Radians');
plt.ylabel('value');
plt.title('Plotting Demonstration')
plt.legend(['Sin', 'Cos'])
plt.grid()
```

Gambar 1.6: Input membuat dan menampilkan grafik

Dengan data yang telah disiapkan, langkah selanjutnya adalah menggunakan Matplotlib untuk menggambar grafik. Untuk menampilkan lebih dari satu garis pada grafik yang sama, fungsi plt.plot() dapat dipanggil beberapa kali secara berurutan.

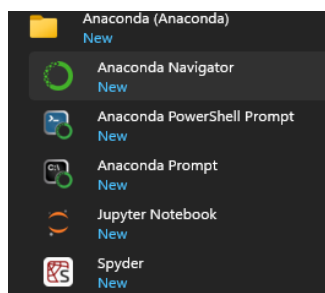
- plt.plot(x,y,'b',x,z,'r'): fungsi inti untuk menggambar garis yang menghubungkan titik-titik.
- plt.xlabel('Radians'), plt.ylabel('value'), plt.title('Plotting Demonstration'): fungsi-fungsi ini untuk memberi judul dan label pada sumbu agar grafik lebih mudah dipahami.
- plt.legend(['Sin', 'Cos']): berfungsi untuk menambahkan keterangan(legenda) pada grafik. Garis plot yang pertama kali dibuat akan diberi label 'Sin'.Garis plot yang kedua dibuat akan diberi label 'Cos'.
- plt.grid(): berfungsi untuk menampilkan grafik yang telah dibuat.

SOAL 2

Perhitungan Keuntungan Jual Beli Emas

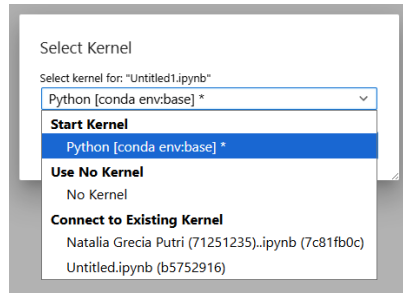
1. Membuka Python di Jupyter Notebook

Pertama, buka aplikasi Jupyter Notebook

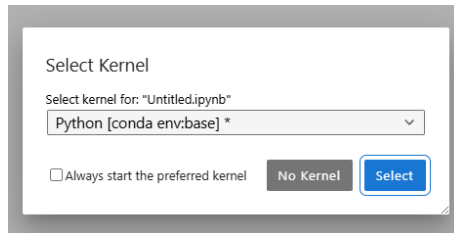


Gambar 1.7: Anaconda—Jupyter Notebook

Setelah menekan Jupyter Notebook akan muncul terminal Jupyter



Gambar 1.10: Select Kernel di Jupyter Notebook



Gambar 1.11: Select Kernel Python

2. Menyelesaikan Permasalahan A

```
#Definisi variabel awal
gram1 = 25
harga_beli1 = 650000
harga_sekarang = 685000

#hitung keuntungan dalam rupiah
total_beli1 = gram1 * harga_beli1
total_sekarang = gram1 * harga_sekarang
untung_rp1 = total_sekarang - total_beli1

#hitung keuntungan dalam presentase
untung_persen1 = (untung_rp1 / total_beli1) * 100

print(f"untung rp: {untung_rp1}")
print(f"untung %: {untung_persen1}%")

untung rp: 875000
untung %: 5.384615384615385%
```

Gambar 1.12: Penyelesaian masalah A

Penjelasan code:

- Mendefinisikan semua data yang diketahui ke dalam variable

```
#Definisi variabel awal
gram1 = 25
harga_beli1 = 650000
harga_sekarang = 685000
```

Gambar 1.13: Memasukkan data

Masukkan semua data yang diberikan dengan variabel yang jelas.

- Menghitung total beli 1, total sekarang, keuntungan dalam rupiah 1

```
#hitung keuntungan dalam rupiah
total_beli1 = gram1 * harga_beli1
total_sekarang = gram1 * harga_sekarang
untung_rp1 = total_sekarang - total_beli1
```

Gambar 1.14: mencari keuntungan dalam rupiah

Untuk mencari total beli pertama menggunakan rumus gram pertama dikalikan dengan harga beli pertama. Untuk mencari total sekarang menggunakan rumus gram pertama dikalikan dengan harga emas sekarang. Untuk mencari keuntungan dalam Rupiah pertama menggunakan rumus total sekarang dikurang dengan total beli pertama.

- Menghitung keuntungan dalam persen dan menampilkan hasil

```
#hitung keuntungan dalam presentase
untung_persen1 = (untung_rp1 / total_beli1) * 100

print(f"untung rp: {untung_rp1}")
print(f"untung %: {untung_persen1}%")
```

Gambar 1.15: mencari keuntungan dan menampilkan hasil

Gunakan rumus untung dalam rupiah pertama dibagi dengan total beli pertama, lalu kalikan dengan seratus.

Selanjutnya, untuk menampilkan hasil gunakan perintah print, dan f-string untuk mencetak teks dan variable.

- Output

```
untung rp: 875000
untung %: 5.384615384615385%
```

Gambar 1.16: Output

Setelah dijalankan akan mendapat outputnya. Untungan dalam Rupiah adalah Rp 875.000, keuntungan yang didapat dalam persen adalah 5,38%.

3. Menyelesaikan Permasalahan B

Menghitung keuntungan setelah membeli emas lagi.

```
#Definisi variabel awal
gram1 = 25
gram2 = 15
harga_beli1 = 650000
harga_beli2 = 685000
harga_sekarang = 715000

#hitung keuntungan dalam rupiah
total_beli1 = gram1 * harga_beli1
total_beli2 = gram2 * harga_beli2
total_modal = total_beli1 + total_beli2
total_sekarang = (gram1 + gram2) * harga_sekarang
untung_rp2 = total_sekarang - (total_modal)

#hitung keuntungan dalam presentase
untung_persen2 = (untung_rp2 / total_modal) * 100

print(f"total modal : {total_modal}")
print(f"untung dalam rupiah : {untung_rp2}")
print(f"untung dalam persentase : {untung_persen2}%")

total modal : 26525000
untung dalam rupiah : 2075000
untung dalam persentase : 7.822808671065033%
```

Gambar 1.17: Permasalahan B

Penjelasan code:

- Mendefinisikan semua data yang diketahui ke dalam variable

```
#Definisi variabel awal
gram1 = 25
gram2 = 15
harga_beli1 = 650000
harga_beli2 = 685000
harga_sekarang = 715000
```

Gambar 1.18: Memasukkan data

Masukkan semua data yang diberikan dengan variabel yang jelas.

- Mencari keuntungan

```
#hitung keuntungan dalam rupiah
total_beli1 = gram1 * harga_beli1
total_beli2 = gram2 * harga_beli2
total_modal = total_beli1 + total_beli2
total_sekarang = (gram1 + gram2) * harga_sekarang
untung_rp2 = total_sekarang - (total_modal)
```

Gambar 1.19: mencari keuntungan kedua

Gambar 1.19 adalah rumus untuk mencari total beli pertama dan kedua, mencari total modal, total sekarang dan mencari keuntungan dalam bentuk rupiah

- Mencari keuntungan dalam persen dan hasil

```
#hitung keuntungan dalam presentase
untung_persen2 = (untung_rp2 / total_modal) * 100

print(f"total modal : {total_modal}")
print(f"untung dalam rupiah : {untung_rp2}")
print(f"untung dalam persentase : {untung_persen2}%")
```

Gambar 1.20: keuntungan persen dan untuk memunculkan hasil

Gunakan rumus untung dalam rupiah pertama dibagi dengan total beli pertama, lalu kalikan dengan seratus.

Selanjutnya, untuk menampilkan hasil gunakan perintah print, dan f-string untuk mencetak teks dan variable.

- Output

```
total modal : 26525000
untung dalam rupiah : 2075000
untung dalam persentase : 7.822808671065033%
```

Gambar 1.21: Output

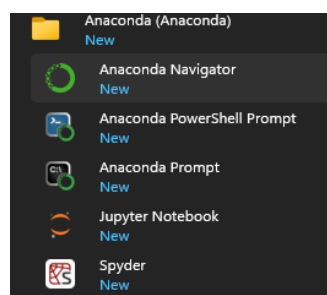
Setelah dijalankan akan mendapat outputnya. Total modalnya 2.652.500, keuntungan dalam Rupiah adalah Rp 2.075.000, keuntungan yang didapat dalam persen adalah 7,82%.

SOAL 3

Menghitung deposito

1. Membuka Python di Jupyter Notebook

Pertama, buka aplikasi Jupyter Notebook



Gambar 1.22: Anaconda—Jupyter Notebook

Setelah menekan Jupyter Notebook akan muncul terminal Jupyter

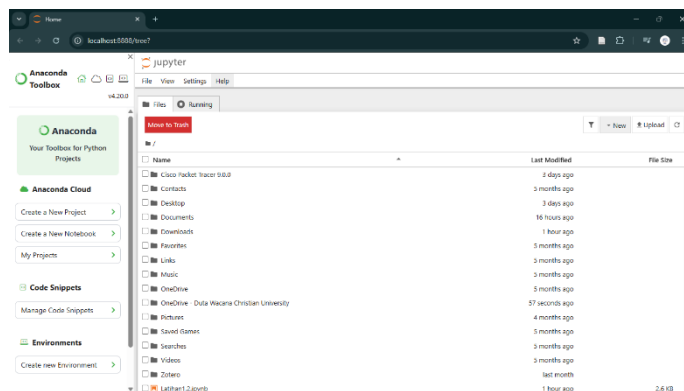
```

Jupyter Notebook
1 2020-02-08 11:59:27.532 ServerApp
2 2020-02-08 11:59:27.622 ServerApp
3 2020-02-08 11:59:27.631 ServerApp
...
Extension package ast_toolbox took 1.1622s to import
Extension package jupyter_lap took 0.2231s to import
Extension package jupyter_server_terminals took 0.2135s to import
Extension package panel.io-jupyter_server_extension took 3.3909s to import
ast_assistant | extension was successfully linked.
ast_core | extension was successfully linked.
ast_panels | extension was successfully linked.
ast_share_notebook | extension was successfully linked.
ast_toolbox | extension was successfully linked.
jupyter_lap | extension was successfully linked.
jupyter_server_terminals | extension was successfully linked.
jupyterlab | extension was successfully linked.
notebook | extension was successfully linked.
notebook_shim | extension was successfully linked.
panel.io-jupyter_server_extension | extension was successfully linked.
nb_conda_kernels | enabled, 1 kernels found.
Registered ast_assistant server extension
ast_assistant | extension was successfully loaded.
Registered ast_core server extension
ast_core | extension was successfully loaded.
Registered ast_panels server extension
ast_panels | extension was successfully loaded.
Registered ast_share_notebook server extension
ast_share_notebook | extension was successfully loaded.

```

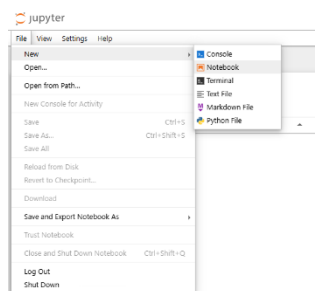
Gambar 1.23: Jupyter

Setelah muncul seperti gamabr 1.8, akan otomatis masuk ke Jupyter Notebook web



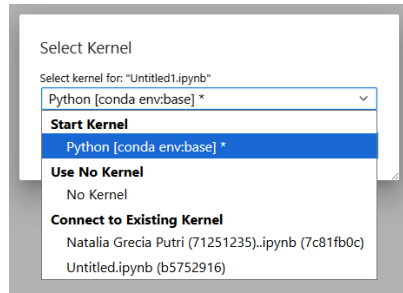
Gambar 1.24: Jupyter Notebook Web

Selanjutnya, jika sudah masuk ke Jupyter Notebook web dan ingin membuat lembar baru klik menu File > New > Notebook

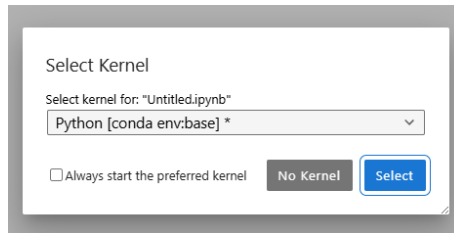


Gambar 1.25: Membuat lembar kerja baru

Setelah menekan Notebook akan muncul Select Kernel, pilih yang Python.



Gambar 1.26: Select Kernel di Jupyter Notebook



Gambar 1.27: Select Kernel Python

2. Analisis masalah dan Menyimpan data

```
#definisi variabel awal
uang_awal = 200000000
uang_akhir = 400000000
bunga_tahun = 0.1

bunga_pertahun = bunga_tahun * uang_awal
waktu = uang_akhir / bunga_pertahun

print(f"lama waktu yang dibutuhkan {waktu} tahun")
```

lama waktu yang dibutuhkan 20.0 tahun

Gambar 1.28: Penyelesaian Latihan 1.3

Penjelasan code:

- Mendefinisikan variabel

```
#definisi variabel awal
uang_awal = 200000000
uang_akhir = 400000000
bunga_tahun = 0.1
```

Gambar 1.29: memasukkan data yang

Masukkan data yang diberikan ke dalam variabel

- Proses mencari bunga

```
bunga_pertahun = bunga_tahun * uang_awal
waktu = uang_akhir / bunga_pertahun

print(f"lama waktu yang dibutuhkan {waktu} tahun")
```

Gambar 1.30: Perhitungan bunga

Prosenya mencari bunga pertahun dan lama waktu untuk mendapat 400 juta

- Output

`lama waktu yang dibutuhkan 20.0 tahun`

Gambar 1.31: Output

Output yang dapat adalah 20 tahun untuk mendapat hasil 400 juta.