



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	71251225
Nama Lengkap	RIFKI WAYU SHABAN RIZALDI
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%)

MATERI 1 Kenapa Python?

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpretatif dan mendukung paradigma berorientasi objek. Bahasa ini memiliki semantik dinamis serta menekankan kemudahan dalam membaca kode. Python banyak digunakan dalam pengembangan web, pembuatan perangkat lunak, analisis data, dan berbagai jenis aplikasi lainnya.



Source image: <https://images.app.goo.gl/geEsrNjQ6BB8Ehad9>

Python memiliki kesederhanaan bentuk dan sintaks yang dapat dilihat pada source code program hello word jika dibandingkan dengan Java dan C

Java:

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

C:

```
#include <stdio.h>

int main(){
    printf("Hello World");
    return 0;
}
```

Python:

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

Python memiliki kelebihan:

1. **Dukungan Pihak Ketiga yang Kaya dan Beragam:** Python memiliki ekosistem pustaka yang sangat lengkap dan beragam. Berbagai library tersedia untuk mendukung banyak kebutuhan, seperti komputasi ilmiah (NumPy, SciPy), analisis data (Pandas), pembelajaran mesin (Scikit-learn, TensorFlow), pengembangan web (Django, Flask), dan lainnya. Keberadaan pustaka ini membantu pengembang menyelesaikan tugas secara efisien tanpa harus membuat kode dari awal.
2. **Fitur Bawaan yang Komprehensif** Selain dukungan pustaka eksternal, Python juga dilengkapi dengan fitur bawaan yang kuat. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk berorientasi objek, fungsional, dan imperatif. Python juga menyediakan fasilitas untuk pengelolaan basis data, jaringan, file, serta interaksi dengan sistem operasi.
3. **Sumber Terbuka (Open Source):** Python adalah bahasa pemrograman sumber terbuka, yang berarti kode sumbernya tersedia untuk umum dan dapat dimodifikasi oleh siapa pun. Hal ini memungkinkan komunitas pengembang untuk terus mengembangkan dan meningkatkan Python. Selain itu, Python juga gratis untuk digunakan tanpa perlu membayar lisensi.
4. **Mudah Dipelajari:** Python memiliki sintaks yang sederhana, jelas, dan mudah dibaca sehingga cocok bagi pemula. Bentuk penulisannya yang menyerupai bahasa manusia

memudahkan proses pembelajaran. Ditambah lagi, komunitas Python yang besar menyediakan banyak sumber belajar dan dukungan.

5. **Mudah Diintegrasikan dengan Bahasa Lain** Python dapat diintegrasikan dengan bahasa pemrograman lain seperti C dan C++. Kemampuan ini memungkinkan Python digunakan dalam proyek yang membutuhkan performa tinggi atau kerja sama dengan sistem yang dibangun menggunakan bahasa lain.

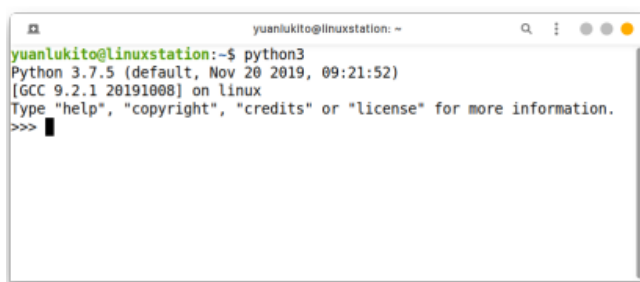
6. **Kekurangan Python:**

1. **Kurangnya Dukungan untuk Platform Mobile:** Python belum memiliki dukungan yang kuat untuk pengembangan aplikasi mobile di platform seperti iOS dan Android. Meskipun ada beberapa kerangka kerja (framework) yang memungkinkan pengembangan aplikasi mobile dengan Python, namun popularitas dan kinerja platform ini masih belum setara dengan bahasa pemrograman lain seperti Java (untuk Android) dan Swift (untuk iOS).
2. **Konsumsi Memori yang Besar:** Python cenderung menggunakan memori yang lebih besar dibandingkan dengan bahasa pemrograman lain seperti C atau C++. Hal ini dapat menjadi masalah dalam proyek-proyek yang membutuhkan penggunaan memori yang efisien.

MATERI 2 Menginstal Python

Python adalah bahasa pemrograman yang mudah ditemukan di berbagai sistem operasi. Saat ini, terdapat dua versi utama yang umum digunakan, yaitu Python 2 dan Python 3. Untuk memudahkan pembelajaran, mata kuliah ini akan menggunakan Python 3. Jika Anda menggunakan Linux atau macOS, Python kemungkinan besar sudah terpasang. Pada sistem operasi Ubuntu, Anda dapat langsung menggunakan Python 3 di terminal hanya dengan mengetikkan perintah `python3`.

Python versi 3 pada Ubuntu Linux:



```
yuanlukito@linuxstation: ~  
yuanlukito@linuxstation:~$ python3  
Python 3.7.5 (default, Nov 20 2019, 09:21:52)  
[GCC 9.2.1 20191008] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

Python versi 3 pada Windows:



MATERI 3 Python Mode Interaktif

```
NameError: name 'cls' is not defined
>>> alas = 10
>>> tinggi = 8
>>> luas = 0.5 * alas * tinggi
>>> luas
40.0
>>> D
```

Serangkaian perintah diberikan untuk menghitung luas sebuah segitiga. Pertama, nilai 10 diberikan ke variabel bernama "panjang", yang kemungkinan mewakili panjang alas segitiga. Kemudian, nilai 8 diberikan ke variabel "tinggi", yang mungkin merupakan tinggi segitiga. Selanjutnya, variabel "luas" dihitung menggunakan rumus luas segitiga, yaitu $0.5 * \text{panjang} * \text{tinggi}$. Hasil perhitungan ini kemudian disimpan dalam variabel "luas". Akhirnya, nilai dari variabel "luas", yang merupakan luas segitiga, ditampilkan.

MATERI 4 Editor Untuk Python

Saat menulis program Python, editor kode sangat penting. Editor yang canggih dapat mempermudah proses pembuatan program. Beberapa editor yang umum digunakan untuk Python adalah Visual Studio Code dengan Python Extension, PyCharm, Spyder, ActivePython, dan IDLE. PyCharm dan Spyder adalah IDE, yang merupakan aplikasi lengkap untuk pengembangan aplikasi dengan fasilitas terintegrasi. IDE biasanya digunakan untuk proyek skala

menengah ke atas. Untuk praktikum, editor sederhana sudah memadai karena tampilannya sederhana dan fasilitasnya cukup untuk kebutuhan praktikum.

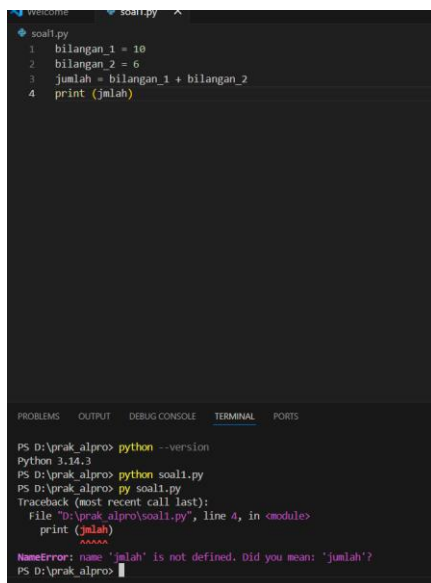
MATERI 4 Menjalankan Script Python di Terminal/Console

Terdapat dua cara menjalankan kode Python:

1. **Mode Interaktif:** Mengetik perintah satu per satu, cocok untuk mencoba perintah baru, namun kurang efisien untuk program panjang.
2. **Mode Skrip:** Menulis kode dalam file .py, lalu menjalankan file tersebut melalui terminal/command prompt dengan perintah `python3 namafile.py` (Linux/macOS) atau `python namafile.py` (Windows). Mode ini lebih efisien untuk program yang lebih besar.

MATERI Mencari Bug dan Memperbaikinya(Debugging)

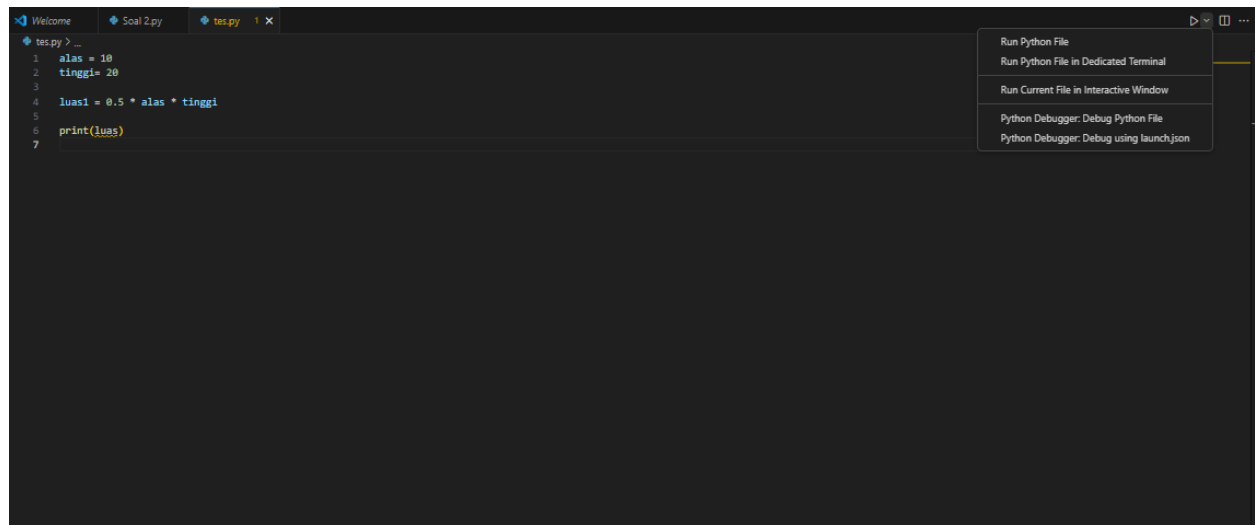
Fitur debugging dapat digunakan dengan vs code seperti dibawah:



```
soali.py
1  bilangan_1 = 10
2  bilangan_2 = 6
3  jumlah = bilangan_1 + bilangan_2
4  print (jumlah)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\prak_alpro> python --version
Python 3.14.3
PS D:\prak_alpro> python soal1.py
PS D:\prak_alpro> py soal1.py
Traceback (most recent call last):
  File "D:\prak_alpro\soal1.py", line 4, in <module>
    print (jumlah)
          ^^^^^
NameError: name 'jumlah' is not defined. Did you mean: 'jumlah'?
PS D:\prak_alpro>
```

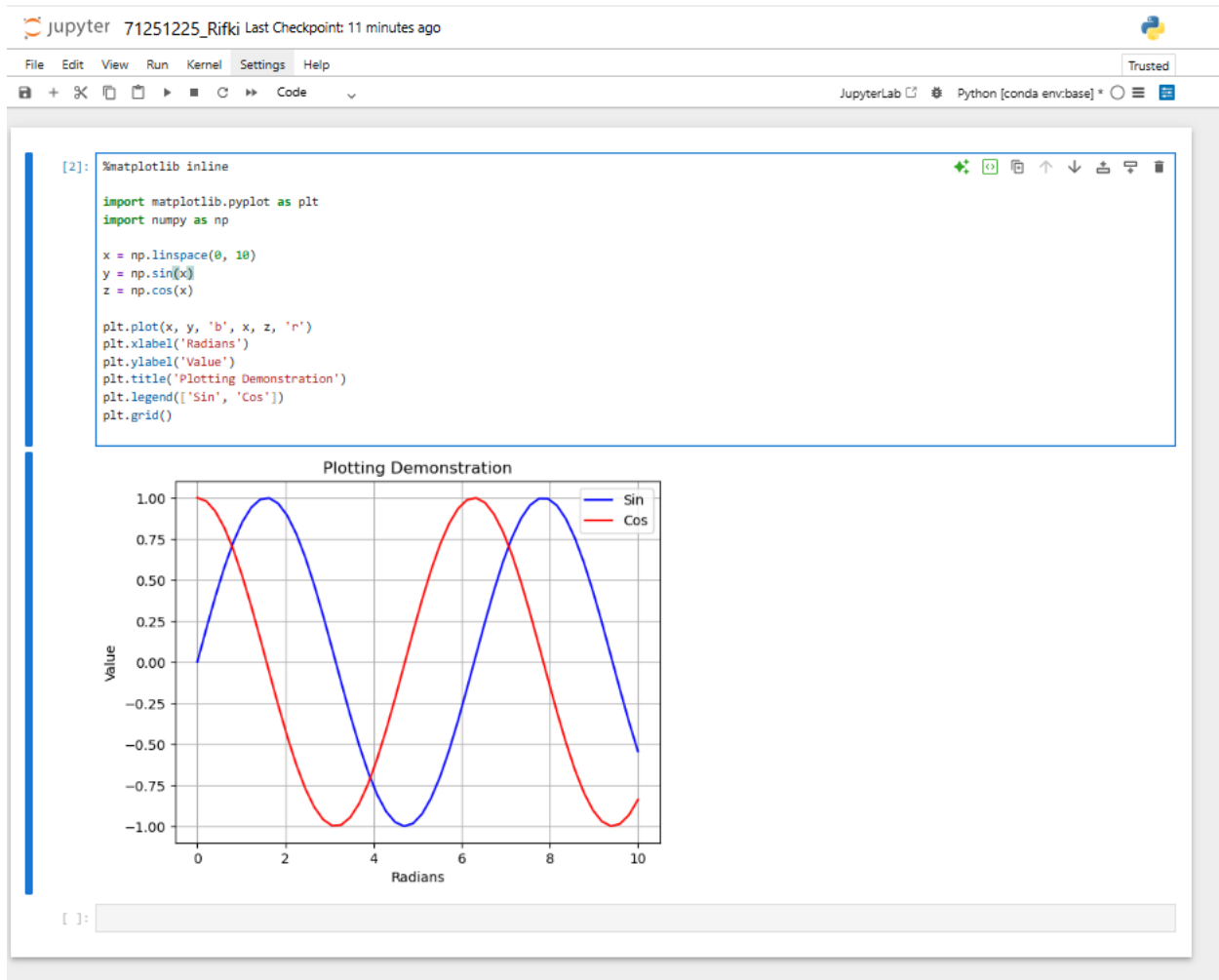


BAGIAN 2: LATIHAN MANDIRI (60%)

Link Github: <https://github.com/rifkiwayushabanR/week1>

SOAL 1

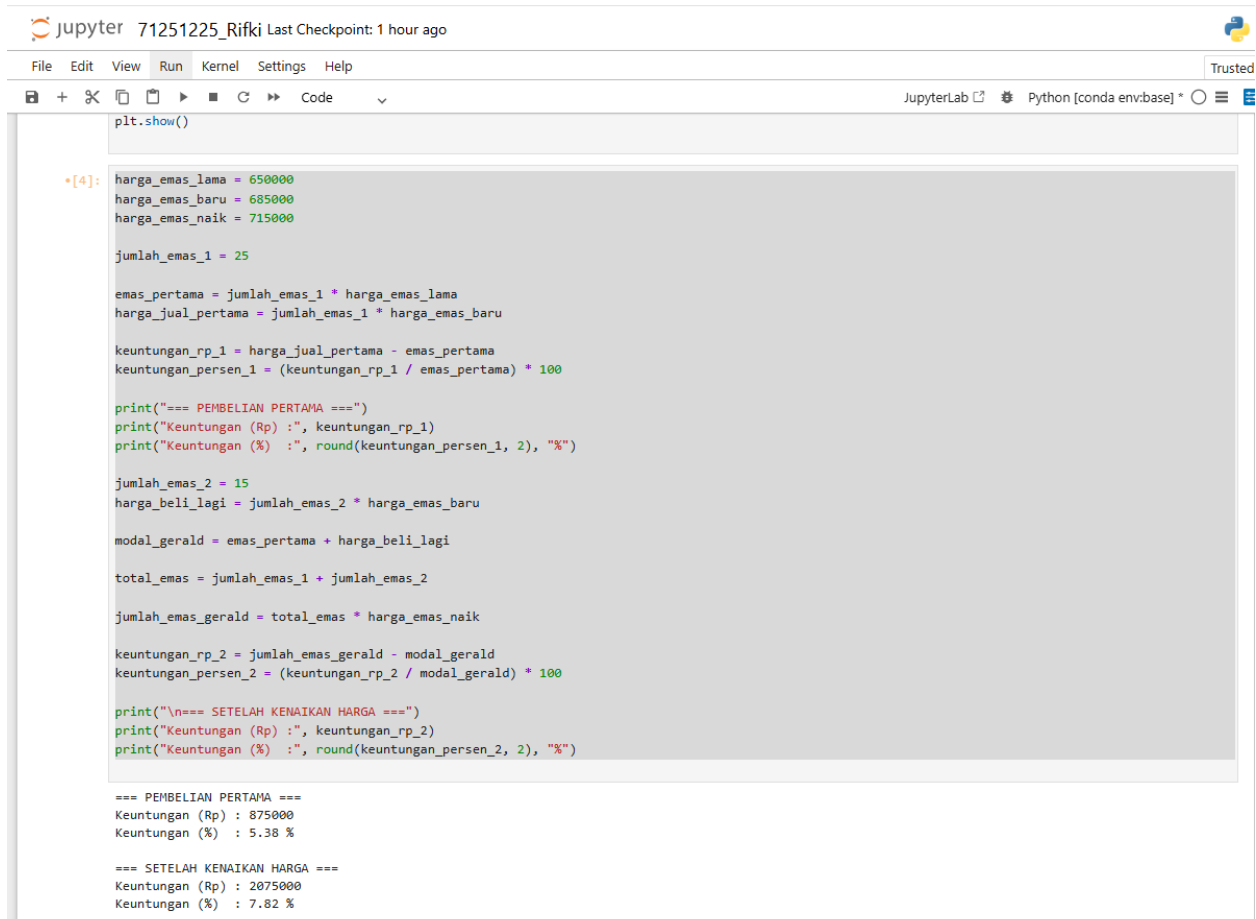
Code:



- Pernyataan *import matplotlib.pyplot as plt* dan *import numpy as np* berfungsi untuk memanggil pustaka yang digunakan dalam pembuatan grafik dan pengolahan data numerik pada Python.
- Baris *x = np.linspace(0, 10)* digunakan untuk menghasilkan sebuah array yang berisi sejumlah nilai yang tersebar merata dari 0 hingga 10, yang kemudian dimanfaatkan sebagai nilai pada sumbu x grafik.

- Variabel y digunakan untuk menyimpan hasil perhitungan fungsi sinus dari setiap nilai yang terdapat pada array x.
- Variabel z digunakan untuk menyimpan hasil perhitungan fungsi kosinus berdasarkan nilai-nilai pada array x.
- Grafik dibuat dengan memetakan nilai x terhadap y sebagai grafik pertama dengan warna biru, sedangkan grafik kedua memetakan nilai x terhadap z dengan garis berwarna merah.

SOAL 2



The screenshot shows a JupyterLab window with a Python kernel. The code calculates the profit percentage for two different gold trading scenarios. The first scenario involves buying 25 units of gold at 650,000 and selling them at 685,000. The second scenario involves buying 15 units at 685,000 and 10 units at 715,000. The final output shows the profit percentages for both scenarios.

```
plt.show()

* [4]: harga_emas_lama = 650000
      harga_emas_baru = 685000
      harga_emas_naik = 715000

      jumlah_emas_1 = 25

      emas_pertama = jumlah_emas_1 * harga_emas_lama
      harga_jual_pertama = jumlah_emas_1 * harga_emas_baru

      keuntungan_rp_1 = harga_jual_pertama - emas_pertama
      keuntungan_persen_1 = (keuntungan_rp_1 / emas_pertama) * 100

      print("=== PEMBELIAN PERTAMA ===")
      print("Keuntungan (Rp) :", keuntungan_rp_1)
      print("Keuntungan (%) :", round(keuntungan_persen_1, 2), "%")

      jumlah_emas_2 = 15
      harga_beli_lagi = jumlah_emas_2 * harga_emas_baru

      modal_gerald = emas_pertama + harga_beli_lagi

      total_emas = jumlah_emas_1 + jumlah_emas_2

      jumlah_emas_gerald = total_emas * harga_emas_naik

      keuntungan_rp_2 = jumlah_emas_gerald - modal_gerald
      keuntungan_persen_2 = (keuntungan_rp_2 / modal_gerald) * 100

      print("\n=== SETELAH KENAIKAN HARGA ===")
      print("Keuntungan (Rp) :", keuntungan_rp_2)
      print("Keuntungan (%) :", round(keuntungan_persen_2, 2), "%")

      === PEMBELIAN PERTAMA ===
      Keuntungan (Rp) : 875000
      Keuntungan (%) : 5.38 %

      === SETELAH KENAIKAN HARGA ===
      Keuntungan (Rp) : 2075000
      Keuntungan (%) : 7.82 %
```

Penjelasan

- Membuat variabel untuk menyimpan harga emas awal, harga emas setelah kenaikan pertama, dan harga emas setelah kenaikan kedua agar mudah digunakan dalam perhitungan.
- Membuat variabel untuk menampung jumlah emas yang dibeli Gerald pada pembelian pertama.
- Menghitung total harga beli emas pada pembelian pertama dengan mengalikan jumlah emas dengan harga emas awal.
- Menghitung total harga jual emas pada pembelian pertama dengan mengalikan jumlah emas dengan harga emas setelah kenaikan pertama.
- Menghitung keuntungan pembelian pertama dalam bentuk rupiah dengan cara mengurangkan total harga jual dengan total harga beli.
- Menghitung keuntungan pembelian pertama dalam bentuk persentase dengan membagi keuntungan rupiah dengan modal awal, kemudian dikalikan 100.
- Menampilkan hasil perhitungan keuntungan pembelian pertama menggunakan fungsi `print()`.
- Membuat variabel untuk menyimpan jumlah emas yang dibeli pada pembelian kedua.
- Menghitung total biaya pembelian emas kedua dengan mengalikan jumlah emas dengan harga emas terbaru.
- Menghitung total modal Gerald dengan menjumlahkan biaya pembelian emas pertama dan pembelian emas kedua.
- Menghitung total jumlah emas yang dimiliki Gerald setelah dua kali pembelian.
- Menghitung total nilai emas setelah harga emas mengalami kenaikan dengan mengalikan total emas dengan harga emas terbaru.
- Menghitung total keuntungan Gerald dalam rupiah dengan mengurangkan total nilai emas terbaru dengan total modal.
- Menghitung total keuntungan Gerald dalam bentuk persentase dengan membagi keuntungan total dengan modal, kemudian dikalikan 100.
- Menampilkan hasil akhir keuntungan setelah kenaikan harga emas menggunakan fungsi `print()`.

SOAL 3

 jupyter 71251225_Rifki Last Checkpoint: 1 hour ago

```
File Edit View Run Kernel Settings Help

[5]: modal_awal = 200_000_000
      batas_target = 400_000_000
      suku_bunga = 10 / 100

      lama_waktu = 0
      jumlah_uang = modal_awal

      while jumlah_uang < batas_target:
          jumlah_uang *= (1 + suku_bunga)
          lama_waktu += 1

      print("Modal awal      : Rp", modal_awal)
      print("Jumlah akhir    : Rp", round(jumlah_uang, 2))
      print("Waktu diperlukan :", lama_waktu, "tahun")

      Modal awal      : Rp 200000000
      Jumlah akhir    : Rp 428717762.0
      Waktu diperlukan : 8 tahun

[ ]:
```

Penjelasan

- Tentukan nilai modal awal, batas target, dan suku bunga dalam bentuk desimal.
- Inisialisasi variabel waktu (lama_waktu) dengan 0 dan jumlah uang saat ini dengan modal awal.
- Buat perulangan yang berjalan selama jumlah uang masih kurang dari batas target.
- Di dalam perulangan, tambahkan bunga ke jumlah uang dengan mengalikan jumlah uang dengan (1 + suku_bunga).
- Tambahkan nilai lama_waktu sebanyak 1 setiap kali perulangan terjadi (menghitung bertambahnya tahun).

- Setelah perulangan selesai, tampilkan modal awal, jumlah akhir, dan lama waktu yang dibutuhkan untuk mencapai target.