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	**71251230**
Nama Lengkap	**Okky Alexander**
Minggu ke / Materi	**01 / Bahasa Pemrograman Python**

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

MATERI PYTHON

Python didefinisikan sebagai bahasa pemrograman tingkat tinggi yang bersifat interpreted, mendukung Object Oriented Programming (OOP), serta memiliki dynamic semantics. Python merupakan salah satu bahasa yang paling banyak digunakan di dunia, berada di posisi kedua setelah Javascript menurut survei Stackoverflow 2019. Aturan sintaks Python sangat sederhana dibandingkan bahasa lain (seperti Java atau C), sehingga sangat ramah bagi pemula.

Gambar 1.1: Logo Python (diambil dari https://www.python.org/).

Contoh Source code Python Program Hello World :

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

Gambar 1.2 : Source code Python

```
PS E:\Koolyeah\github\PrakAlPro-71251230> python -u "e:\Koolyeah\github\PrakAlPro-71251230\Minggu 1\Latihan1.py"
Hello, World!
```

Gambar 1.3 : Output Source code python Program Hello World.

Keunggulan Python:

- Dukungan pustaka (library) pihak ketiga yang sangat kaya, terutama di bidang Data Science (contoh: Pandas, Numpy, TensorFlow).
- Pustaka bawaan yang mencakup basis data, jaringan, dan fitur sistem operasi.
- Lisensi Open Source sehingga bebas digunakan untuk keperluan komersial.

Kekurangan Python:

- Belum mendukung pembuatan aplikasi mobile asli (Android/iOS).

- Konsumsi memori relatif besar.
- Kecepatan proses cenderung lebih lambat dibanding bahasa C.

MATERI MENINSTALL PYTHON 3

Python saat ini tersedia dalam versi 2 dan 3, namun Python 3 sebagai versi terbaru. Pengguna Linux atau macOS biasanya sudah memiliki Python 3 secara bawaan. Python 3 bisa diakses lewat terminal dengan perintah python3.

```
C:\Users\Lenovo LOQ>python3
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Gambar 1.4 : Python Versi 3 di Windows. Terinstall Python versi 3.14.

Bagi pengguna Windows, Anda bisa juga menggunakan distribusi Anaconda Individual Edition karena proses instalasinya relatif mudah. Cukup pilih 64-bit Graphical Installer dengan versi Python 3.9 atau yang terbaru sesuai dengan spesifikasi sistem operasi Anda.

Gambar 1.5 : Distribusi Anaconda Individual Edition. (Gambar di ambil dari Modul Materi Bahasa Pemrograman Python.)

MATERI MENJALANKAN PYTHON MODE INTERAKTIF

Untuk masuk ke mode interaktif, Anda dapat menjalankan perintah python3 pada terminal, namun bisa juga menggunakan Anaconda Prompt jika Anda pengguna Windows

dengan mengetikkan perintah python saja.

```
C:\Users\Lenovo LOQ>python3
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Gambar 1.6 : Informasi versi Python akan ditampilkan dan siap menerima perintah.

Mode interaktif memungkinkan Anda memasukkan perintah satu per satu untuk langsung diproses oleh interpreter. Sebagai contoh, untuk menghitung luas segitiga dengan alas 10 dan tinggi 8, Anda namun bisa juga menggunakan urutan perintah berikut: cukup ketik

alas = 10

tinggi = 8

luas = 0.5 * alas * tinggi

lalu ketik luas dan tekan Enter untuk melihat hasilnya.

```
C:\Users\Lenovo LOQ>python3
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 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
>>>
```

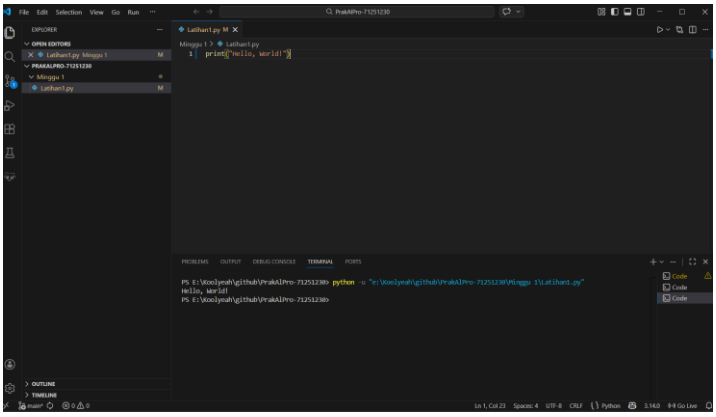
Gambar 1.6 : Menghitung Luas Segitiga dengan bantuan Python.

Dua baris awal merupakan perintah *assignment* untuk menyimpan nilai pada variabel panjang dan tinggi. Selanjutnya, variabel luas diisi dengan hasil perhitungan rumus segitiga, namun bisa juga menggunakan perintah terakhir hanya dengan mengetik nama variabelnya untuk menampilkan hasil akhir. Sebagai catatan, variabel berfungsi sebagai penyimpan nilai untuk proses berikutnya, dan Anda cukup mengetik exit() lalu Enter untuk mengakhiri sesi interaktif.

MATERI EDITOR UNTUK PYTHON

Pemilihan perangkat lunak yang tepat sangat menentukan kemudahan dalam menulis program Python. Secara umum, tersedia berbagai pilihan populer mulai dari editor teks seperti

Visual Studio Code (VS Code), ActivePython, dan IDLE, hingga IDE (Integrated Development Environment) yang lebih kompleks seperti PyCharm dan Spyder. Perbedaan utamanya terletak pada kelengkapan fitur, IDE menyediakan seluruh fasilitas pengembangan yang terintegrasi untuk proyek skala besar, sementara editor lebih menonjolkan kesederhanaan tampilan yang sangat cocok untuk kebutuhan praktikum.



Gambar 1.7 : Tampilan Visual Studio Code saat menjalankan script Python.

MATERI MENJALANKAN SCRIPT PYTHON DI TERMINAL/CONSOLE

Dalam pemrograman Python, terdapat dua metode utama untuk menjalankan kode, yaitu mode interaktif dan mode script. Mode interaktif memungkinkan pengguna untuk mengetik perintah satu per satu dan langsung melihat hasilnya melalui interpreter, sehingga sangat ideal untuk bereksperimen dengan fungsi baru secara cepat. Namun, mode ini memiliki keterbatasan karena perintah tidak tersimpan secara permanen dan harus diketik ulang jika sesi berakhir.

Sebagai solusinya, Python menyediakan mode script yang memungkinkan kumpulan perintah disimpan dalam file berekstensi .py untuk dijalankan sekaligus tanpa pengetikan ulang. Untuk mengeksekusi script tersebut, pengguna cukup membuka Terminal atau Command Prompt dan mengetikkan perintah python namafile.py. Metode ini jauh lebih efisien untuk pengembangan program yang kompleks dan permanen, seperti pada contoh pembuatan program konversi nilai tukar dolar yang disimpan dalam sebuah file.

```python
# nilai kurs 1 US$ ke IDR
kursusd = 13950
# informasi program
print('Program konversi US$ ke IDR')
print('Kurs saat ini 1 US$ = ',kursusd, 'Rupiah')
# input jumlah US$ yang mau ditukar
jumlahusd = float(input('Masukkan jumlah uang yang mau ditukar ke Rupiah: '))
# hitung nilainya dalam Rupiah
dalamrupiah = jumlahusd * kursusd
# tampilkan hasilnya
print('Hasil konversi = Rp. ', dalamrupiah)
```

Gambar 1.8: Code Program kursusd.

Jalankan script tersebut gunakan perintah python namafile.py pada terminal. Tampilan hasil dapat dilihat pada gambar 1.9.

```
PS E:\Koolyeah\github\PrakAlPro-71251230> python -u "e:\Koolyeah\github\PrakAlPro-71251230\Minggu 1\Latihan1.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
PS E:\Koolyeah\github\PrakAlPro-71251230>
```

Gambar 1.9: Menjalankan script Python di Terminal.

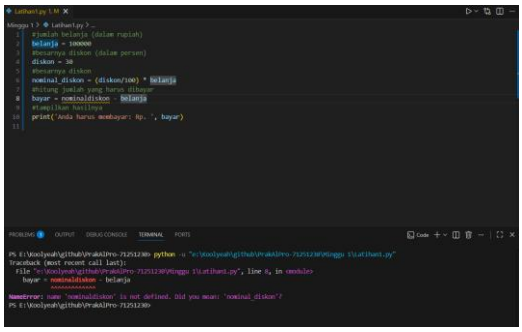
Dalam pemrograman Python, tanda pagar (#) digunakan untuk menulis komentar yang berfungsi sebagai pemberitahuan bagi pemrogram namun diabaikan oleh sistem. Penggunaan editor modern seperti Visual Studio Code semakin mempermudah proses ini karena pengguna tidak perlu mengetik perintah manual di terminal eksternal. Cukup dengan menekan tombol **Run**, program akan langsung dieksekusi dan hasilnya dapat dipantau serta diinteraksi melalui terminal yang sudah terintegrasi di dalam aplikasi.

MATERI MENCARI BUG DAN MEMPERBAIKINYA (debugging)

Dalam pemrograman, kesalahan atau *bug* secara umum terbagi menjadi dua: **syntax error** yang terjadi akibat kesalahan penulisan (seperti *typo*) dan **runtime error** yang terjadi saat program sedang berjalan. Karena Python menggunakan sistem interpreter, kode akan diperiksa dan dijalankan baris demi baris. Jika ditemukan kesalahan pada satu baris, program akan langsung berhenti seketika.

Sebagai contoh, sebuah program bisa gagal berjalan hanya karena perbedaan kecil pada nama variabel, seperti menuliskan nominal_diskon di awal namun memanggilnya sebagai nominaldiskon di baris berikutnya. Kesalahan jenis ini memicu **NameError**, di mana interpreter

menganggap variabel tersebut belum pernah didefinisikan, sehingga proses perhitungan terhenti sebelum sempat menampilkan hasil apa pun.

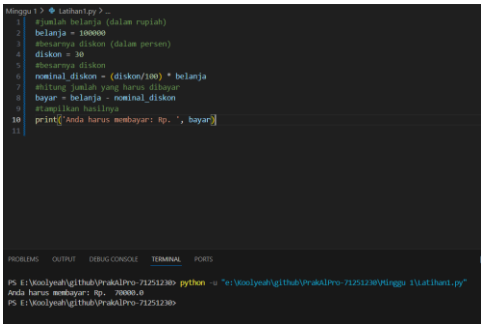


Gambar 1.9 : Kesalahan yang muncul saat script dijalankan.

Selain syntax error, terdapat jenis kesalahan lain yang lebih sulit dideteksi yaitu logic error (sering disebut sebagai bagian dari runtime error). Pada jenis ini, program berhasil berjalan hingga selesai tanpa berhenti, namun memberikan hasil yang salah. Hal ini biasanya dipicu oleh algoritma atau logika perhitungan yang keliru, seperti pada contoh perhitungan diskon di mana rumusnya terbalik sehingga menghasilkan angka negatif.

Kesalahan pada baris ke-11 tersebut menunjukkan bahwa meskipun penulisan kodenya sudah benar secara sistem, urutan pengurangannya salah. Untuk memperbaikinya, variabel belanja harus dikurangi dengan nominal diskon agar menghasilkan nilai bayar yang tepat sebesar Rp. 70.000.

Setelah diperbaiki, script tersebut akan menghasilkan hasil yang benar sesuai dengan yang diharapkan. Hasilnya dapat dilihat pada Gambar 1.10.



Gambar 1.10 : Hasil sudah sesuai dengan yang diharapkan.

Source Code Github : https://github.com/tuangkeman/PrakAlPro-71251230.git

LATIHAN 1.1

Membuka anaconda navigator dan lauch Jupyter dapat dilihat pada gambar 1.11. Setelah jupyter berhasil dibuka clik new dan pilih python 3 untuk membuat file python pada jupyter dapat dilihat pada gambar 1.12. Lanjut menuliskan code program. Gunakan import matplotlib.pyplot (as plt) untuk library utama membuat grafik. Gunakan import numpy (as np) untuk library untuk pengolahan angka dan array numerik. Ketik %matplotlib inline untuk Perintah khusus di Jupyter agar grafik muncul langsung di bawah baris kode (bukan di jendela terpisah).

x = np.linspace(0, 10): Membuat 50 titik angka yang berjarak sama, dimulai dari 0 hingga 10. Ini digunakan sebagai sumbu horizontal (X).

y = np.sin(x): Menghitung nilai sinus untuk setiap titik pada x.

z = np.cos(x): Menghitung nilai kosinus untuk setiap titik pada x.

plt.plot(x, y, 'b', x, z, 'r'): Menginstruksikan Python untuk menggambar dua garis sekaligus.

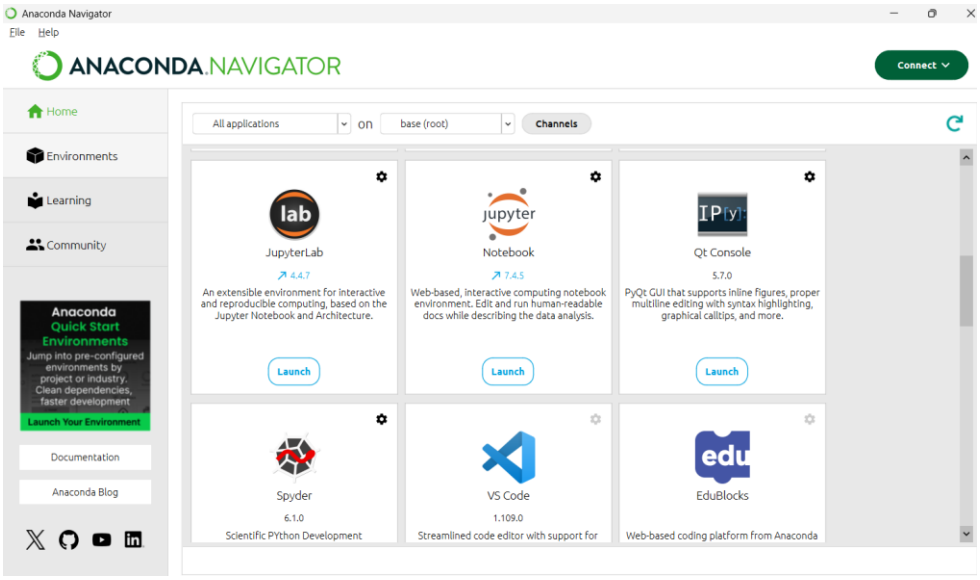
- Garis biru ('b') mewakili fungsi Sin.
- Garis merah ('r') mewakili fungsi Cos.

plt.xlabel & plt.ylabel: Memberikan label pada sumbu X ("Radians") dan sumbu Y ("Value").

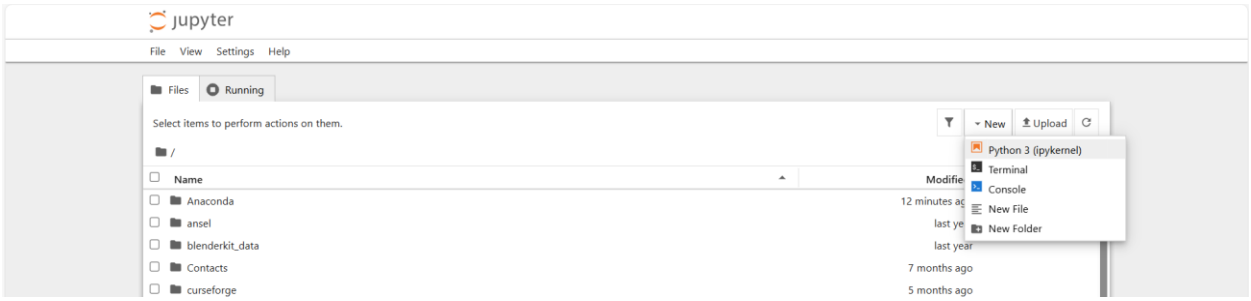
plt.title: Memberikan judul grafik "Plotting Demonstration".

plt.legend(['Sin', 'Cos']): Menampilkan kotak keterangan di pojok kanan atas untuk membedakan garis biru dan merah.

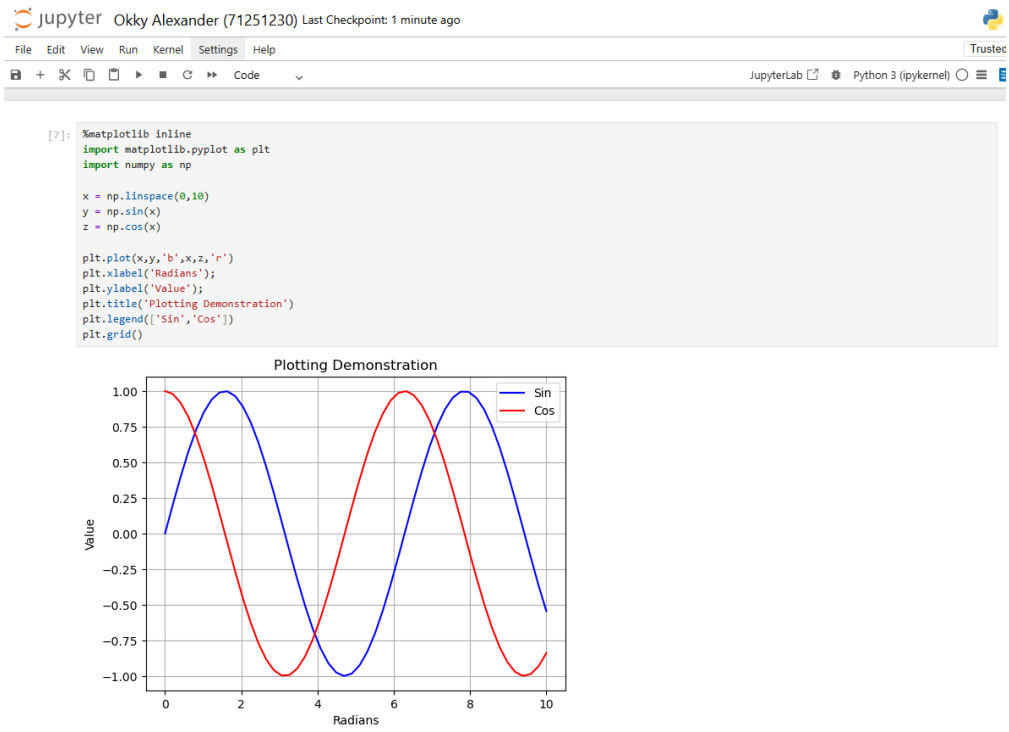
plt.grid(): Menampilkan garis-garis kotak (kisi-kisi) di latar belakang agar nilai lebih mudah dibaca.



Gambar 1.11 : Tampilan Awal Anaconda Navigator.



Gambar 1.12 : Tampilan Jupyter membuat file baru.



Gambar 1.13 : Menggunakan Jupyter Notebook untuk menghasilkan grafik.

LATIHAN 1.2

Untuk menghitung keuntungan dan presentasenya maka kita perluh menuliskan code didalam Terminal CMD. Ketik python3 untuk masuk ke python dan menuliskan code.

harga_awal = 650000: Menentukan harga beli emas per gram (Rp 650.000).

berat_emas = 25: Menentukan jumlah emas yang dibeli (25 gram).

modal_beli = harga_awal * berat_emas: Menghitung total uang yang dikeluarkan (Harga * Berat).

harga_kenaikan = 685000: Menentukan harga emas saat ini yang sudah naik.

nilai_saatini = harga_kenaikan * berat_emas: Menghitung nilai total emas jika dijual dengan harga baru.

untung_rp_1 = nilai_saatini - modal_beli: Menghitung selisih keuntungan dalam bentuk Rupiah.

untung_persen_1 = (untung_rp_1 / modal_beli) * 100: Menghitung persentase keuntungan dari modal awal.

print(): Baris ini berfungsi untuk menampilkan hasil perhitungan ke layar dengan format ribuan (:,) dan dua angka di belakang koma (:.2f).

harga_pembelian_kedua = 685000: Menentukan harga beli emas untuk transaksi kedua.

berat_pembelian_kedua = 15: Menentukan jumlah emas tambahan yang dibeli.

modal_pembelian_kedua = harga_pembelian_kedua * berat_pembelian_kedua: Menghitung uang yang dikeluarkan untuk pembelian kedua.

total_berat = berat_emas + berat_pembelian_kedua: Menjumlahkan stok emas lama (25g) dan baru (15g) menjadi 40 gram.

total_modal = modal_beli + modal_pembelian_kedua: Menjumlahkan modal pertama dan kedua untuk mendapatkan total investasi.

harga_akhir = 715000: Menentukan harga jual emas terbaru (setelah naik lagi).

nilai_akhir = total_berat * harga_akhir: Menghitung total nilai uang dari seluruh emas (40g) dengan harga terbaru.

untung_rp_total = nilai_akhir - total_modal: Menghitung total keuntungan bersih dari seluruh transaksi.

untung_persen_total = (untung_rp_total / total_modal) * 100: Menghitung persentase keuntungan total dibandingkan total modal yang keluar.

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
... untung_rp_1 = nilai_saatini - modal_beli
... untung_persen_1 = (untung_rp_1 / modal_beli) * 100
...
... print("Pembelian Gerard Pertama (25 gram):")
... print(f"Modal Awal: Rp {modal_beli:,}")
... print(f"Keuntungan: Rp {untung_rp_1:,}")
... print(f"Persentase Keuntungan: {untung_persen_1:.2f}%")
... print()
...
...
... harga_pembelian_kedua = 685000
... berat_pembelian_kedua = 15
... modal_pembelian_kedua = harga_pembelian_kedua * berat_pembelian_kedua
...
...
... total_berat = berat_emas + berat_pembelian_kedua
... total_modal = modal_beli + modal_pembelian_kedua
...
... harga_akhir = 715000
... nilai_akhir = total_berat * harga_akhir
...
... untung_rp_total = nilai_akhir - total_modal
... untung_persen_total = (untung_rp_total / total_modal) * 100
...
... print("Pembelian Gerard Kedua (15 gram):")
... print(f"Total Modal (25g + 15g): Rp {total_modal:,}")
... print(f"Nilai Emas Akhir: Rp {nilai_akhir:,}")
... print(f"Total Keuntungan: Rp {untung_rp_total:,}")
... print(f"Total Persentase Keuntungan: {untung_persen_total:.2f}%")
...
Pembelian Gerard Pertama (25 gram):
Modal Awal: Rp 16,250,000
Keuntungan: Rp 875,000
Persentase Keuntungan: 5.38%

Pembelian Gerard Kedua (15 gram):
Total Modal (25g + 15g): Rp 26,525,000
Nilai Emas Akhir: Rp 28,600,000
Total Keuntungan: Rp 2,075,000
Total Persentase Keuntungan: 7.82%
```

Gambar 1.14 : Tampilan python 3 berisi code program menghitung presentase keuntungan.

LATIHAN 1.3

Menggunakan perulangan untuk menghitung lama waktu yang dibutuhkan Erika. Kita perluh membuka Terminal CMD dan menuliskan kode programnya.

saldo = 200000000: Menentukan modal awal Erika sebesar 200 juta.

target = 400000000: Menentukan angka sasaran yang ingin dicapai (minimal 400 juta).

bunga = 0.10: Menentukan bunga deposito sebesar 10% per tahun.

tahun = 0: Variabel pencatat waktu yang dimulai dari titik nol.

while saldo < target:: Ini adalah perintah "selama". Python akan terus menjalankan kode di bawahnya selama saldo Erika belum menyentuh angka 400 juta.

saldo += saldo * bunga: Ini adalah inti dari compound interest (bunga majemuk). Saldo baru dihitung dari saldo tahun berjalan ditambah bunga dari saldo tersebut.

tahun += 1: Menambah hitungan tahun setiap kali siklus bunga selesai dihitung.

print(...): Menampilkan proses pertumbuhan uang Erika tahun demi tahun agar kita bisa melihat progresnya.

```
C:\Users\Lenovo LOQ>python3
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> saldo = 200000000
... target = 400000000
... bunga = 0.10
... tahun = 0
...
... while saldo < target:
...     saldo += saldo * bunga
...     tahun += 1
...     print(f"Tahun ke-{tahun}: Rp {saldo:,.0f}")
... print(f"\nWaktu yang dibutuhkan adalah {tahun} tahun.")
...
Tahun ke-1: Rp 220,000,000
Tahun ke-2: Rp 242,000,000
Tahun ke-3: Rp 266,200,000
Tahun ke-4: Rp 292,820,000
Tahun ke-5: Rp 322,102,000
Tahun ke-6: Rp 354,312,200
Tahun ke-7: Rp 389,743,420
Tahun ke-8: Rp 428,717,762

Waktu yang dibutuhkan adalah 8 tahun.
>>>
```

Gambar 1.15 : Tampilan python 3 berisi code Menghitung waktu yang dibutuhkan.