



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	71251172
Nama Lengkap	Ang, Joshua Alexander Hartono
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%)

Kenapa Python ?

Python merupakan bahasa pemrograman tingkat tinggi yang dijalankan secara interpreted (dijalankan baris demi baris oleh interpreter saat program dieksekusi). Selain itu, bahasa ini mendukung konsep Object Oriented Programming (OOP) serta menggunakan sistem dynamic typing. Menurut survei yang dilakukan oleh Stack Overflow, Python termasuk salah satu bahasa pemrograman yang paling banyak digunakan setelah JavaScript. Hal ini dikarenakan Python memiliki aturan sintaks yang lebih sederhana dan mudah dipelajari oleh pemula. Kesederhanaan bentuk dan sintaks Python dapat dilihat dari perbandingan source code program Hello World jika dibandingkan dengan Java dan C.

Pada bahasa pemrograman Java:

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

Gambar 1.1: Penulisan source code program “Hello World!” pada bahasa pemrograman Java.

Sumber: <https://sl1nk.com/zlAXQ>

Pada bahasa pemrograman C:

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

Gambar 1.2: Penulisan source code program “Hello World!” pada bahasa pemrograman C.

Sumber: <https://sl1nk.com/zlAXQ>

Pada bahasa pemrograman Python:

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

Gambar 1.3: Penulisan source code program “Hello World!” pada bahasa pemrograman Python.

Sumber: <https://sl1nk.com/zlAXQ>

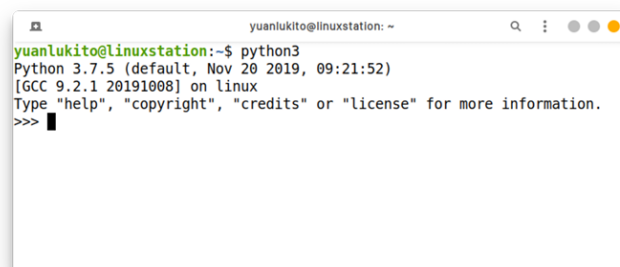
Adapun Python memiliki beberapa keunggulan lainnya yaitu Pustaka bawaan yang dari Python yang beragam dan sudah mencakup banyak aspek yang dibutuhkan oleh programmer seperti dukungan terhadap basis data, pengaksesan jaringan, pengaksesan fitur-fitur di sistem operasi dan masih banyak lainnya. Disisi lain, Python juga memiliki lisensi open source sehingga dapat digunakan secara bebas untuk keperluan komersial tanpa perlu membayar lisensi.

Meskipun demikian, Python tetap memiliki banyak kekurangan. Bahasa ini belum banyak digunakan untuk pengembangan aplikasi mobile, membutuhkan penggunaan memori yang relatif besar, dan memiliki kecepatan eksekusi yang lebih lambat dibandingkan bahasa pemrograman tingkat rendah seperti C.

Menginstal Python 3

Python memiliki dua versi yang umum digunakan, yaitu Python 2 dan Python 3. Pada bagian ini kita akan menginstal Python 3 yang merupakan versi terbaru dari Python.

Pada sistem operasi Linux dan macOS, Python 3 biasanya sudah terpasang dan siap digunakan. Pengguna Ubuntu dapat menjalankan Python 3 melalui terminal dengan perintah Python 3.

A screenshot of a terminal window titled 'yuanlukito@linuxstation: ~'. The terminal shows the command 'python3' being executed, which outputs 'Python 3.7.5 (default, Nov 20 2019, 09:21:52) [GCC 9.2.1 20191008] on linux'. It then prompts the user to 'Type "help", "copyright", "credits" or "license" for more information.' and shows the prompt '>>>' with a cursor.

Gambar 1.4: Python versi 3 di Ubuntu Linux. Terinstal Python Versi 3.7.5 Sumber:

<https://sl1nk.com/zlAXQ>

Untuk pengguna Windows, Python dapat diinstal dengan menggunakan Anaconda, karena proses instalasinya yang lebih mudah. Disarankan memilih Anaconda Individual Edition 64-Bit Graphical

Installer dan menyesuaikan versi dengan sistem operasi. Pada saat modul ini dibuat, Anaconda menyediakan Python versi 3.9.



Gambar 1.5: Distribusi Anaconda Individual Edition. Sumber: <https://sl1nk.com/zlAXQ>

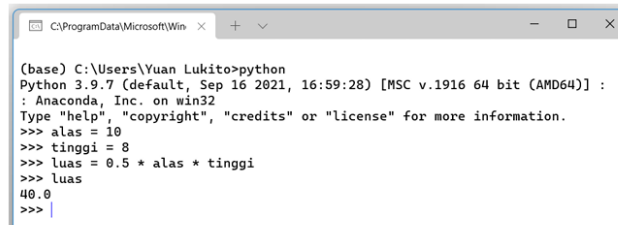
Menjalankan Python Mode Interaktif

Mode interaktif Python dapat dijalankan melalui terminal. Pada sistem Ubuntu, penggunaan cukup menetik python3. Sementara itu, pada Windows, mode ini dapat diakses dengan membuka Anaconda Prompt lalu menetik perintah Python.

```
C:\ProgramData\Microsoft\Win
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :
: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Gambar 1.6: Informasi versi Python akan ditampilkan dan siap menerima perintah. Sumber: <https://sl1nk.com/zlAXQ>

Dalam mode interaktif, perintah Python dapat dimasukkan satu per satu dan langsung diproses oleh interpreter. Sebagai contoh, perhitungan luas segitiga dengan alas 10 cm dan tinggi 8 cm dapat dilakukan dengan mengetikkan perintah berikut secara baris-perbaris dan diakhiri tombol Enter.



```
(base) C:\Users\Yuan Lukito>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :
: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> alas = 10
>>> tinggi = 8
>>> luas = 0.5 * alas * tinggi
>>> luas
40.0
>>>
```

Gambar 1.7: Menghitung Luas Segitiga dengan bantuan Python. *Sumber:*

<https://sl1nk.com/zlAXQ>

Catatan: variable digunakan untuk menyimpan nilai yang dipakai dalam proses selanjutnya. Untuk mengakhiri sesi mode interaktif, pengguna dapat mengetik perintah `exit()`.

Editor untuk Python

Untuk menulis suatu program, diperlukan editor yang berfungsi untuk menuliskan source code yang diinginkan. Beberapa editor yang sering digunakan untuk Python antara lain Visual Studio Code, PyCharm, Spyder, ActivePython, dan IDLE. Editor yang lebih lengkap biasanya menyediakan fitur tambahan yang dapat membantu proses pemrograman menjadi lebih mudah dan efisien.

Beberapa editor seperti PyCharm dan Spyder termasuk dalam kategori Integrated Development Environment (IDE) yang umumnya digunakan untuk proyek berskala menengah hingga besar. Namun, untuk keperluan praktikum, editor sederhana sudah cukup. Pada modul ini digunakan Visual Studio Code sebagai editor utama, dengan PyCharm dan Spyder sebagai editor pendukung.



Gambar 1.8: Python Extension untuk Visual Studio Code. *Sumber:* <https://sl1nk.com/zlAXQ>

Menjalankan Script Python di Terminal/Console

Python dapat dijalankan dalam mode interaktif, Di mana perintah diketik satu per satu dan langsung dieksekusi. Mode ini memudahkan pemula untuk mencoba berbagai fungsi, memeriksa hasil perintah, atau menguji kode secara cepat. Cara ini juga bagus untuk memahami bagaimana Python bekerja secara langsung, sehingga setiap perubahan atau percobaan bisa segera terlihat. Namun, kelemahannya adalah perintah harus diketik ulang jika keluar dari mode interaktif.

Selain itu, Python mendukung mode script, yaitu menjalankan kumpulan perintah yang disimpan dalam file .py. File script dapat dijalankan melalui Terminal (Ubuntu) atau Anaconda Prompt (Windows) menggunakan perintah `python3 namafile.py` atau `namafile.py`. Baris yang diawali tanda # merupakan komentar dan tidak dijalankan. Editor seperti Visual Studio Code mempermudah proses ini dengan tombol Run dan terminal bawaan.



```
yuanlukito@linuxstation: ~/Documents/python-code
yuanlukito@linuxstation:~/Documents/python-code$ python3 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
yuanlukito@linuxstation:~/Documents/python-code$
```

Input dari user (keyboard)

Gambar 1.9: Menjalankan script Python di Terminal Ubuntu. Sumber: <https://sl1nk.com/zlAXQ>

Mencari Bug dan Memperbaikinya (Debugging)

Bug adalah kesalahan dalam program yang umumnya terbagi menjadi dua jenis, yaitu syntax error dan runtime error. Syntax error terjadi karena penulisan kode yang salah, seperti typo pada nama variabel. Biasanya bug ini mudah dideteksi oleh interpreter Python atau editor sebelum program dijalankan. Sedangkan runtime error muncul saat program dijalankan, sering disebabkan oleh kesalahan logika atau algoritma. Sehingga hasil yang diperoleh tidak sesuai dengan yang diharapkan.

```

1  #jumlah belanja (dalam rupiah)
2  belanja = 100000
3
4  #besarnya diskon (dalam persen)
5  diskon = 30
6
7  #besarnya diskon
8  nominal_diskon = (diskon/100) * belanja
9
10 #hitung jumlah yang harus dibayar
11 bayar = nominaldiskon - belanja
12
13 #tampilkan hasilnya

```

Gambar 1.10: Program menghitung jumlah yang harus dibayarkan. Sumber:

<https://sl1nk.com/zlAXQ>

```

yuanlukito@linuxstation: ~/Documents/python-code
yuanlukito@linuxstation:~/Documents/python-code$ python3 contohdebug.py
Traceback (most recent call last):
  File "contohdebug.py", line 11, in <module>
    bayar = nominaldiskon - belanja
NameError: name 'nominaldiskon' is not defined
yuanlukito@linuxstation:~/Documents/python-code$

```

Gambar 1.11: Kesalahan yang muncul saat script dijalankan (syntax error). Sumber:

<https://sl1nk.com/zlAXQ>

```

yuanlukito@linuxstation: ~/Documents/python-code
yuanlukito@linuxstation:~/Documents/python-code$ python3 contohdebug.py
Tidak ada pesan kesalahan lagi!
Anda harus membayar: Rp. -10000.0
yuanlukito@linuxstation:~/Documents/python-code$

```

Gambar 1.12: script sudah bisa dijalankan sampai selesai namun bagian hasil yang harus dibayarkan belum sesuai (runtime error). Sumber: <https://sl1nk.com/zlAXQ>

Kesalahan terjadi pada baris ke-11, yaitu pada bagian jumlah uang yang harus dibayar, seharusnya jumlah yang harus dibayar adalah harga barang dikurangi diskon. Pada script tersebut perhitungan terbalik sehingga menjadi salah.

```

yuanlukito@linuxstation: ~/Documents/python-code
yuanlukito@linuxstation:~/Documents/python-code$ python3 contohdebug.py
Anda harus membayar: Rp. 70000.0
yuanlukito@linuxstation:~/Documents/python-code$

```

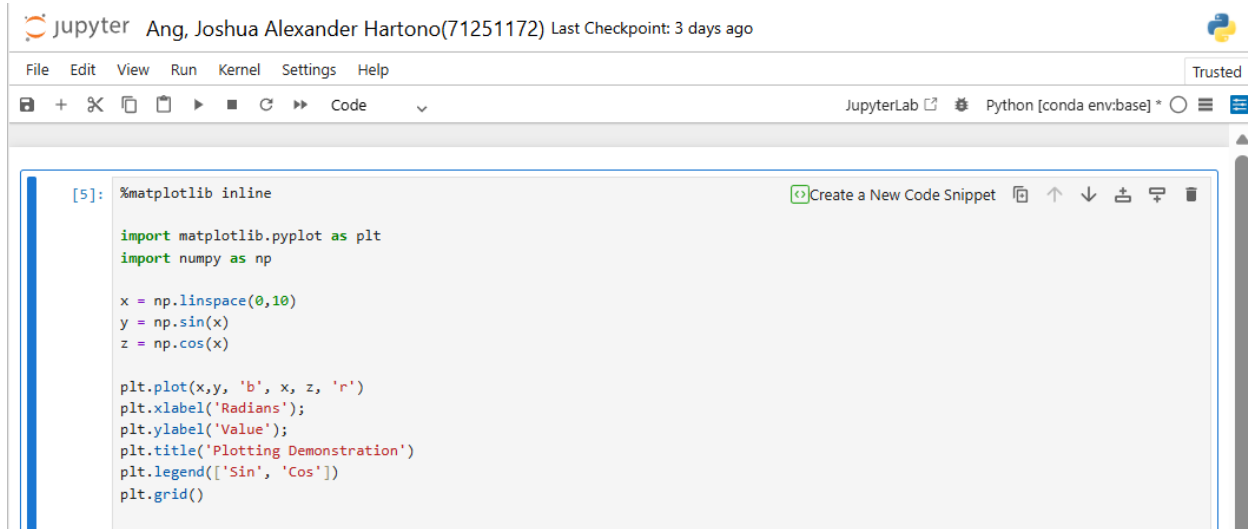
Gambar 1.13: Hasil setelah script diperbaiki. Sumber: <https://sl1nk.com/zlAXQ>

BAGIAN 2: LATIHAN MANDIRI (60%)

Link GitHub: <https://github.com/angioshua-sudo/Prak-Alpro-2026.git>

SOAL 1

Source code:



```
[5]: %matplotlib inline

import matplotlib.pyplot as plt
import numpy as np

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

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

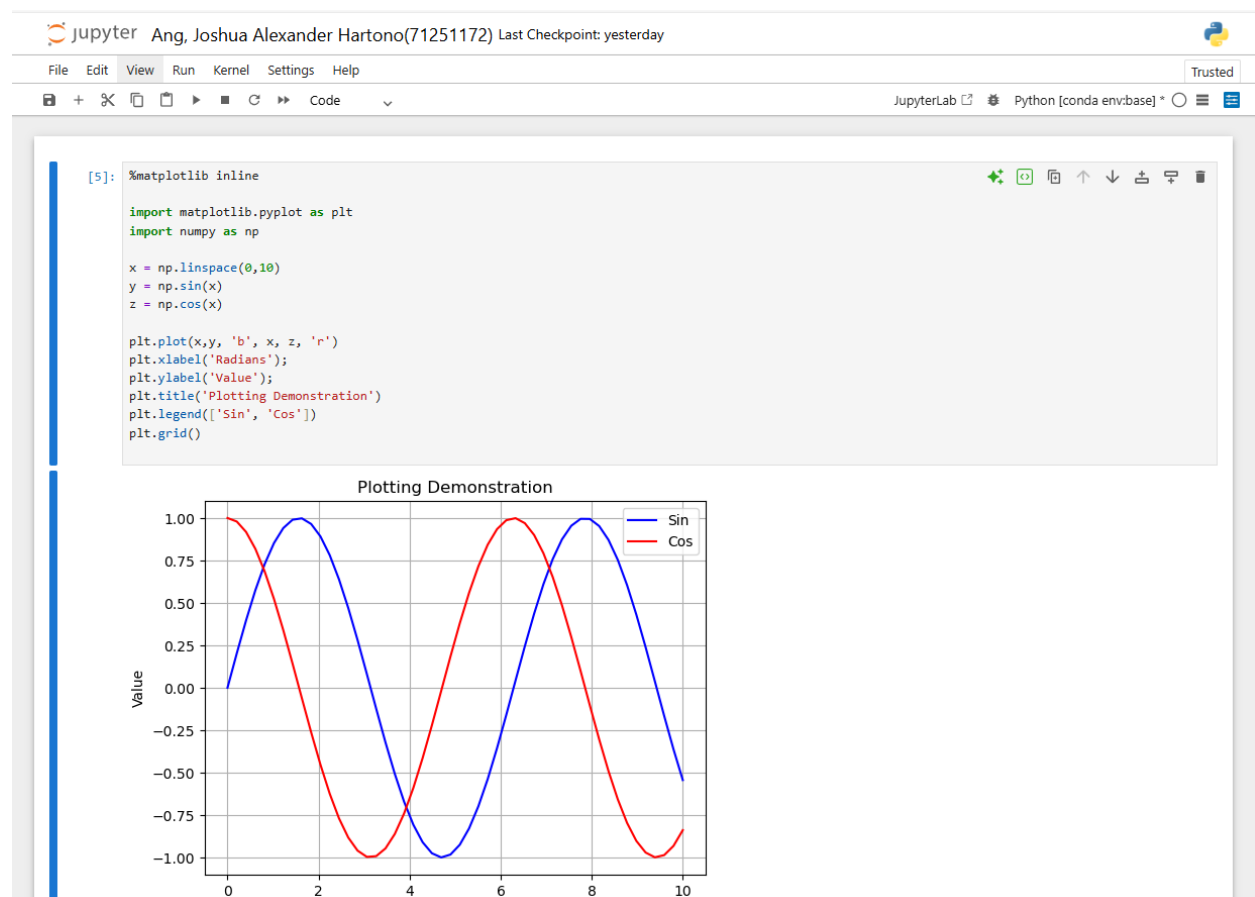
Penjelasan:

Kode ini bertujuan untuk menampilkan grafik sinus dan cosinus menggunakan Python di Jupyter Notebook. Baris pertama, **%matplotlib inline** berfungsi untuk memunculkan grafik secara langsung setelah kode dijalankan bukan di jendela yang terpisah. Setelah itu, ada **import matplotlib.pyplot as plt** yang berfungsi untuk membuat grafik, dalam hal ini **as plt** bertujuan agar fungsi ini lebih mudah dipanggil. Sedangkan, **import numpy as np** berfungsi untuk melakukan perhitungan matematika, di mana **as np** bertujuan agar fungsi ini lebih mudah untuk dipanggil.

Pada bagian **x = np.linspace(0,10)**, program ini membuat sekumpulan angka dari 0 sampai 10 dengan jarak yang sama. Angka-angka ini akan menjadi sumbu horizontal (sumbu x). Setelah itu, **y = np.sin(x)** berfungsi untuk menghitung nilai sinus dari setiap angka dalam x, dan **z = np.cos(x)** digunakan untuk menghitung nilai cos. Sehingga saat ini kita sudah memiliki dua kumpulan data yaitu data sin dan data cos.

Perintah **plt.plot(x, y, 'b', x, z, 'r')** digunakan untuk menampilkan grafik. **Data x** dan **data y** diplot sebagai garis berwarna **biru (ditandai dengan 'b')**, sedangkan **data x** dan **data z** diplot sebagai garis berwarna **merah (ditandai dengan 'r')**. Dengan demikian, dua grafik yaitu fungsi sinus dan cosinus ditampilkan dalam satu gambar yang sama. Selanjutnya, **plt.xlabel('Radians')** memberi label pada sumbu tersebut dinyatakan dalam radian, sedangkan **plt.ylabel('Value')** memberi label pada sumbu vertikal untuk menunjukkan nilai hasil fungsi. Perintah **plt.title('Plotting Demonstration')** menambahkan judul pada grafik, dan **plt.legend(['sin','cos'])** menampilkan bagian legenda untuk membedakan garis sinus dan cosinus sesuai dengan urutan plotting sebelumnya. Terakhir, **plt.grid()** menambahkan garis bantu pada grafik agar pembacaan nilai menjadi lebih mudah dan terlihat rapi.

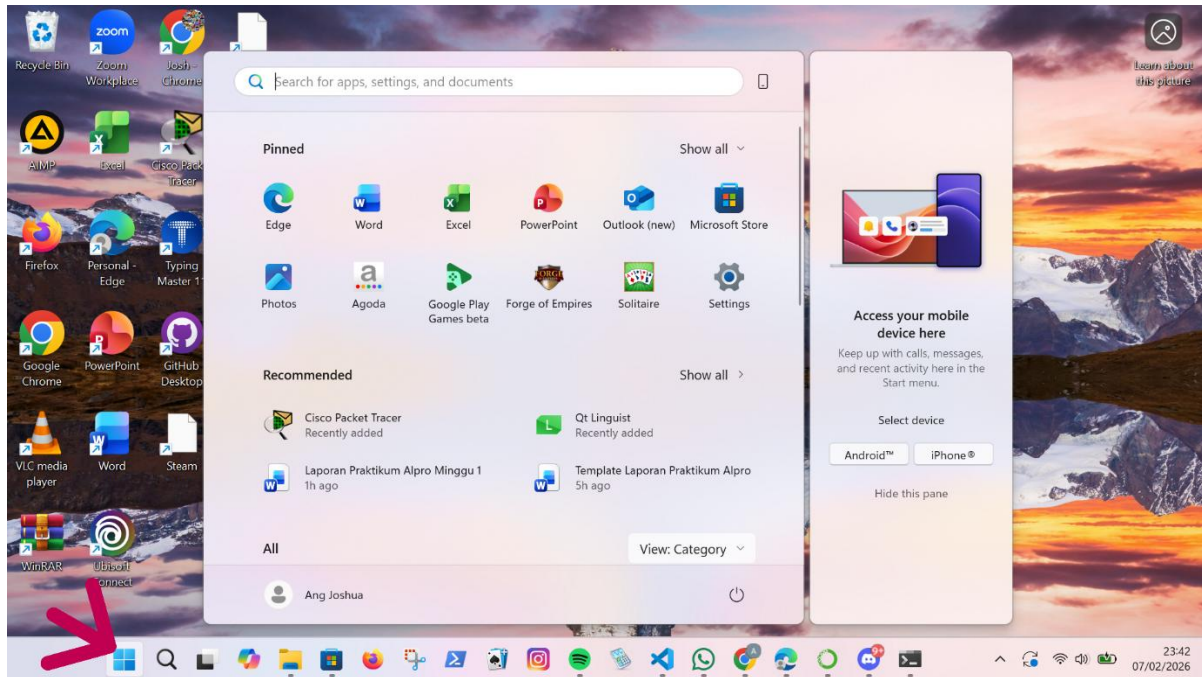
Output:



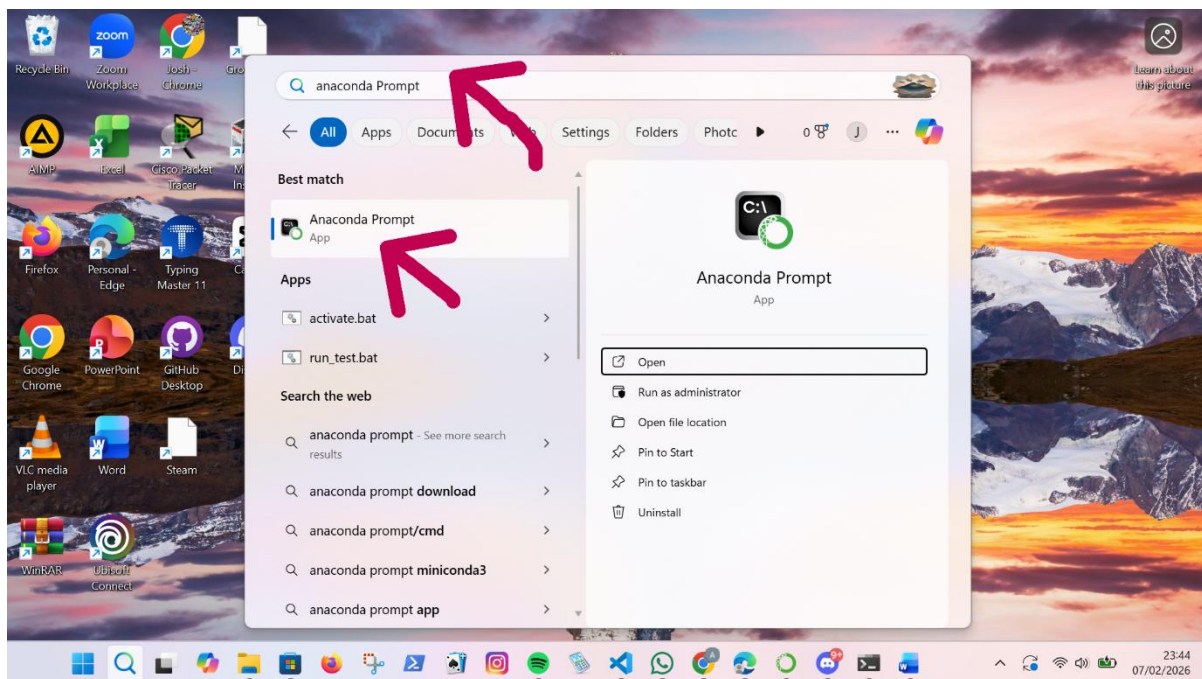
SOAL 2

Langkah membuka Python mode interaktif :

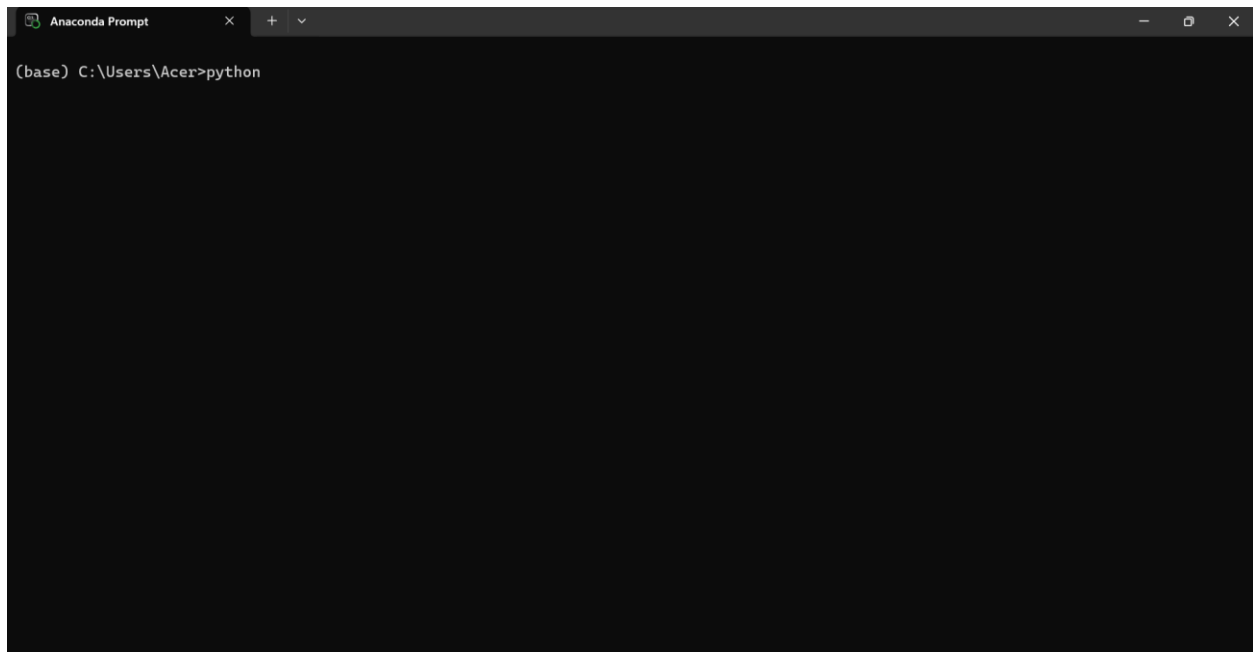
1. Klik tombol Start



2. Cari Anaconda Prompt lalu klik

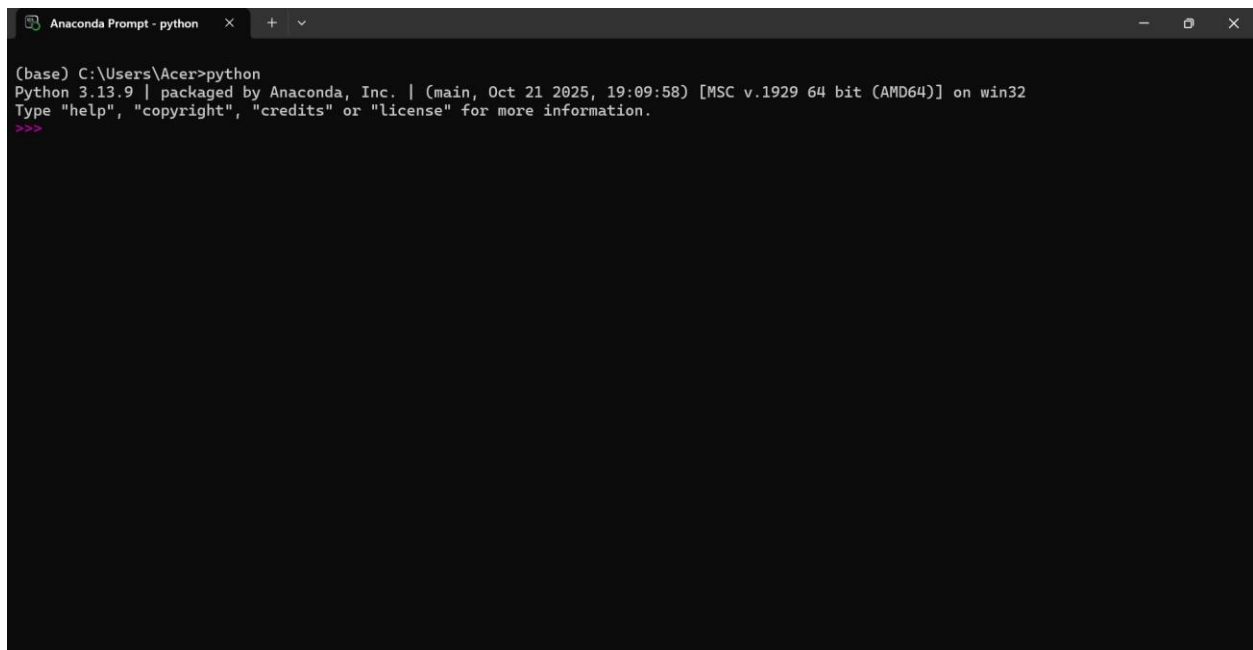


3. Ketikkan python dan tekan enter



```
Anaconda Prompt
(base) C:\Users\Acer>python
```

4. Apabila muncul versi Python serta tand >>> , maka Python mode interaktif siap digunakan



```
Anaconda Prompt - python
(base) C:\Users\Acer>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.
>>>
```

Source code:

```
1 emas_awal = 650000
2 beli = 25
3 emas_naik = 685000
4
5 keuntungan_rupiah = (emas_naik - emas_awal) * beli
6 keuntungan_persen = (emas_naik - emas_awal) / emas_awal * 100
7
8 keuntungan_rupiah
9 keuntungan_persen
10
11 beli_lagi = 15
12 emas_naik_kedua = 715000
13
14 uang1 = emas_awal * beli
15 uang2 = emas_naik * beli_lagi
16 total_uang = uang1 + uang2
17 uang3 = (beli + beli_lagi) * emas_naik_kedua
18 keuntungan_rupiah_kedua = uang3 - total_uang
19
20 keuntungan_persen_kedua = keuntungan_rupiah_kedua / total_uang * 100
21
22 keuntungan_rupiah_kedua
23 keuntungan_persen_kedua
```

Penjelasan:

Pada bagian awal kode terdapat variabel yaitu **emas_awal = 650000** yang menunjukkan harga emas pada awal pembelian. Variabel **beli = 25** menunjukkan jumlah emas yang dibeli di awal (saat pertama kali). Selanjutnya terdapat variabel **emas_naik = 685000** yang menampilkan harga kenaikan emas yang pertama kali.

Bagian selanjutnya adalah menghitung keuntungan yang didapatkan dari pembelian emas di awal. Variabel **keuntungan_rupiah** dihitung dengan menggunakan rumus **(emas_naik - emas_awal) * beli**, yang artinya selisih antara harga kenaikan yaitu Rp 685.000 dikurangi dengan harga awal pembelian emas yaitu Rp 650.000 lalu dikalikan dengan jumlah gram yang dimiliki. Kemudian variabel **keuntungan_persen** dihitung dengan menggunakan rumus **(emas_naik - emas_awal) / emas_awal * 100**. Perhitungan ini dilakukan dengan mencari selisih dari harga awal emas dan harga setelah kenaikan pertama, kemudian perhitungan ini dibagi dengan harga awal untuk melihat perbandingan kenaikannya lalu di kalikan 100 agar menjadi bentuk persen. Setelah itu, variabel **keuntungan_persen** dan **keuntungan_rupiah** dipanggil untuk memunculkan nilai yang tersimpan di variabel tersebut.

Selanjutnya terdapat variabel **beli_lagi = 15** yang menunjukkan jumlah emas tambahan yang dibeli pada pembelian kedua. Variabel **emas_naik_kedua = 715000** menunjukkan harga kenaikan emas kedua serta harga dari pembelian emas kedua per gramnya yaitu Rp 715.000.

Variabel **uang1 = emas_awal * beli** digunakan untuk menghitung total biaya pembelian pertama, yaitu harga emas awal dikalikan dengan jumlah gram yang dibeli pada pembelian pertama kali. Kemudian variabel **uang2 = emas_naik * beli_lagi** digunakan untuk menghitung total biaya pembelian kedua dengan harga saat pembelian emas kedua yaitu Rp 685.000 dikali dengan jumlah gram yang dibeli saat itu yaitu 15 gram. Setelah itu, nilai kedua variabel tersebut ditambah dalam variabel **total_uang = uang1 + uang2** sehingga kita mengetahui modal dari pembelian emas di setiap periode. Kemudian, nilai emas saat ini yaitu Rp 715.000 dikali dengan rumus **uang3 = (beli + beli_lagi) * emas _naik_kedua**. Pada bagian ini, jumlah gram emas yang ada dari hasil pembelian pertama (25 gram) dan pembelian kedua (15 gram) ditotal lalu dikalikan dengan nilai emas saat ini sehingga didapati nilai total dari seluruh emas yang ada.

Selanjutnya keuntungan rupiah setelah pembelian emas yang kedua dapat dihitung dengan variabel **keuntungan_rupiah_kedua = uang3 – total_uang** yaitu dengan cara mengurangkan total modal yang ada dengan nilai emas saat ini. Selisih tersebut menunjukkan jumlah keuntungan dalam rupiah yang didapat setelah dua kali pembelian dan dua kali kenaikan harga. Sedangkan, persentasi keuntungan kedua dihitung dengan rumus **keuntungan_persen_kedua = keuntungan_rupiah_kedua / total_uang * 100** hal ini dilakukan untuk membandingkan total keuntungan terhadap total modal lalu dikalikan 100 agar menjadi persen. Terakhir, **keuntungan_rupiah_kedua** dan **keuntungan_persen_kedua** di panggil untuk menampilkan nilai akhir dari masing-masing variabel.

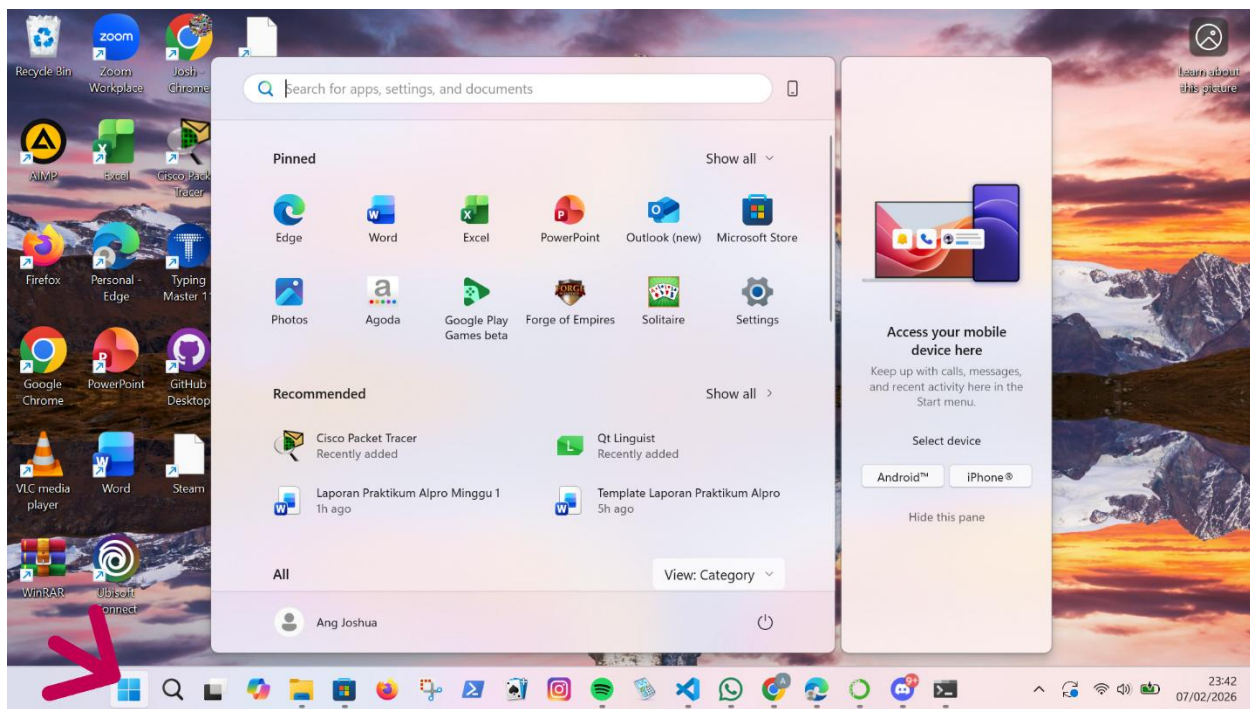
Output:

```
Anaconda Prompt - python
(base) C:\Users\Acer>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.
>>> emas_awal = 650000
>>> beli = 25
>>> emas_naik = 685000
>>>
>>> keuntungan_rupiah = (emas_naik - emas_awal) * beli
>>> keuntungan_persen = (emas_naik - emas_awal) / emas_awal * 100
>>>
>>> keuntungan_rupiah
875000
>>> keuntungan_persen
5.384615384615385
>>>
>>> beli_lagi = 15
>>> emas_naik_kedua = 715000
>>>
>>> uang1 = emas_awal * beli
>>> uang2 = emas_naik * beli_lagi
>>> total_uang = uang1 + uang2
>>> uang3 = (beli + beli_lagi) * emas_naik_kedua
>>> keuntungan_rupiah_kedua = uang3 - total_uang
>>>
>>> keuntungan_persen_kedua = keuntungan_rupiah_kedua / total_uang * 100
>>>
>>> keuntungan_rupiah_kedua
2075000
>>> keuntungan_persen_kedua
7.822808671065033
>>> |
```

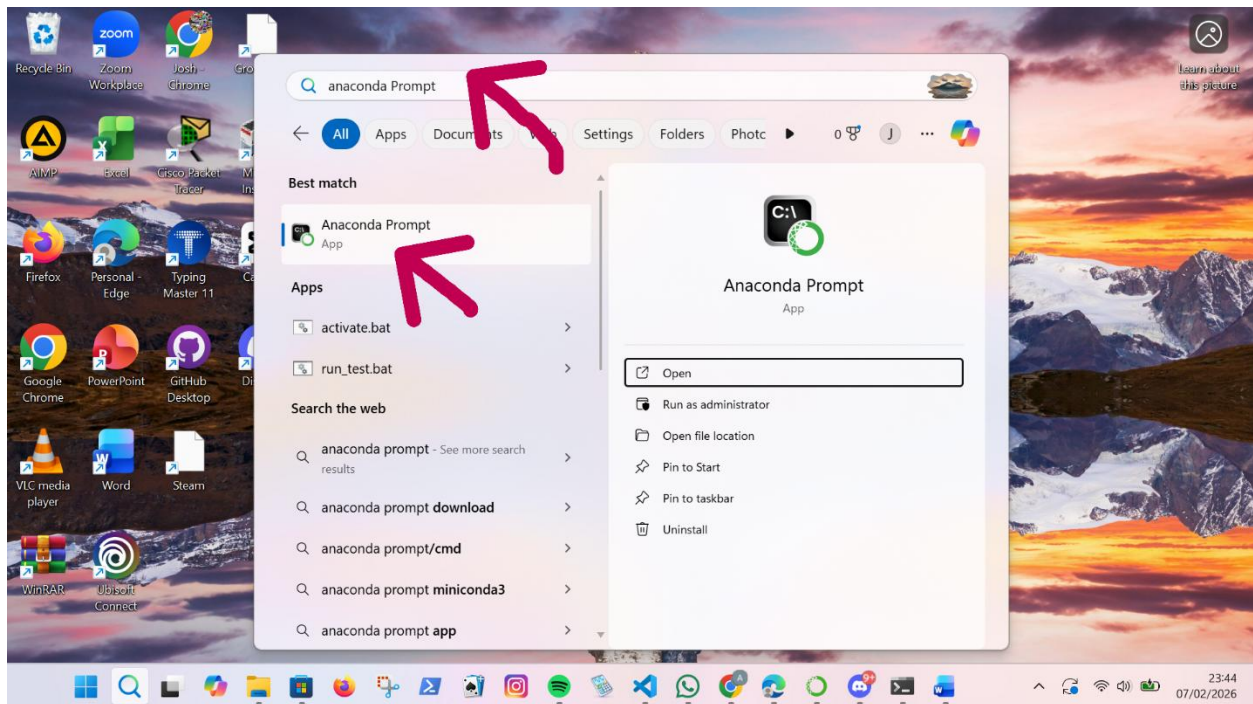
SOAL 3

Langkah membuka Python mode interaktif :

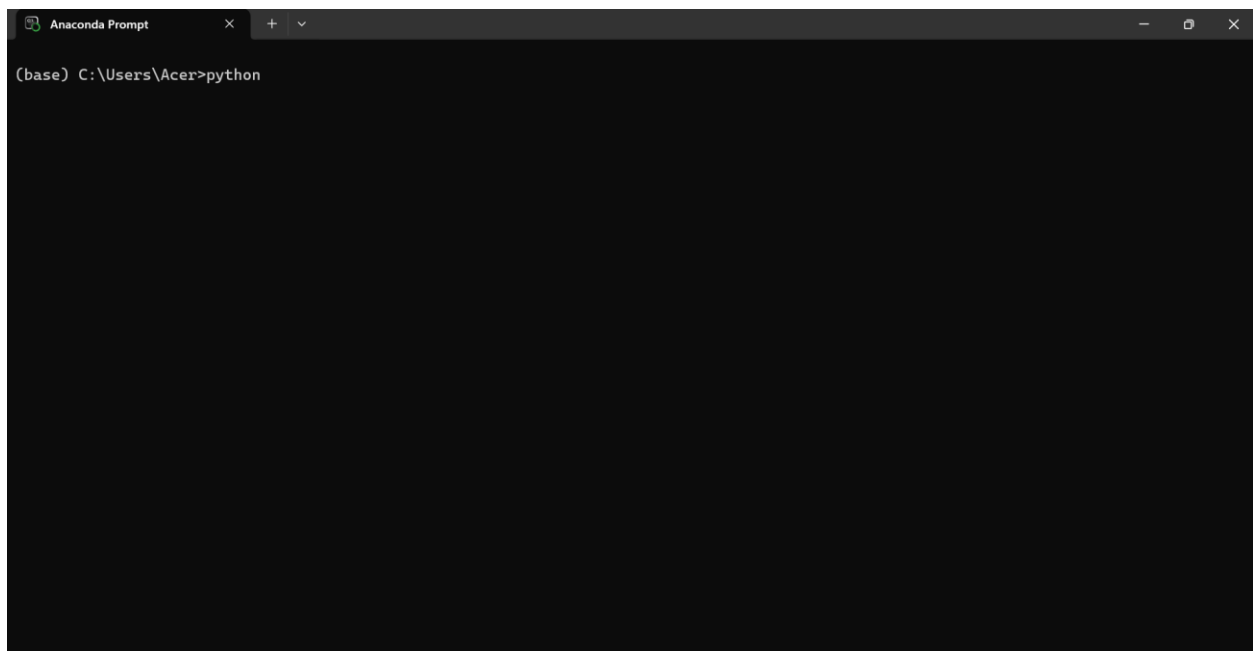
1. Klik tombol Start



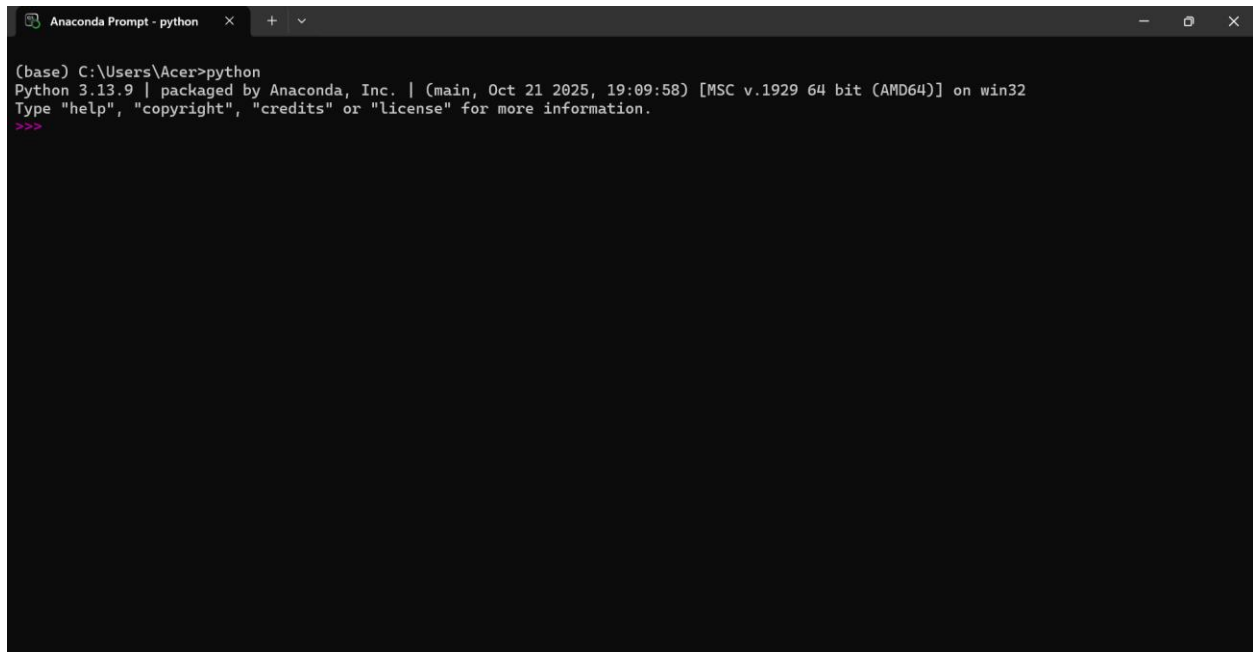
2. Cari Anaconda Prompt lalu klik



3. Ketikkan python dan tekan enter

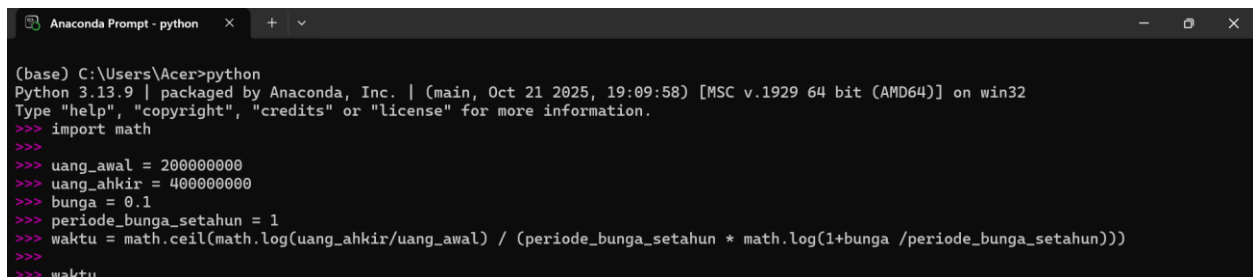


4. Apabila muncul versi Python serta tand >>> , maka Python mode interaktif siap digunakan



```
Anaconda Prompt - python
(base) C:\Users\Acer>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.
>>>
```

Source code:



```
Anaconda Prompt - python
(base) C:\Users\Acer>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.
>>> import math
>>>
>>> uang_awal = 200000000
>>> uang_akhir = 400000000
>>> bunga = 0.1
>>> periode_bunga_setahun = 1
>>> waktu = math.ceil(math.log(uang_akhir/uang_awal) / (periode_bunga_setahun * math.log(1+bunga /periode_bunga_setahun)))
>>>
>>> waktu
```

Penjelasan:

Baris pertama kode terdapat **import math** yang merupakan perintah untuk mengimport modul math yang merupakan modul bawaan Python untuk menyediakan fungsi-fungsi pada matematika (dalam hal ini digunakan untuk logaritma dan pembulatan ke atas). Selanjutnya variabel **uang_awal = 200000000** menunjukkan jumlah uang awal yang dimiliki, sedangkan **uang_akhir = 400000000** menunjukka jumlah uang yang ingin dicapai. Lalu terdapat variabel **bunga = 0.1** yang menjelaskan tingkat bunga sebesar 10% pertahun dalam bentuk decimal.

Selanjutnya terdapat variabel **periode_bunga_setahun = 1** yang menunjukkan bahwa bunga dihitung sebanyak satu kali dalam setahun.

Sebelum masuk ke perhitungan variabel waktu, terlebih dulu perlu dipahami rumus dasar yang digunakan (rumus bunga majemuk):

$$A = P \left(1 + \frac{r}{n}\right)^{nt}$$

Gambar 2.1: Rumus bunga majemuk. Sumber: <https://sl1nk.com/zlAXQ>

Pada rumus tersebut, A merupakan jumlah uang akhir, P adalah modal awal, r adalah tingkat bunga pertahun dalam bentuk decimal, n adalah jumlah periode penggabungan bunga dalam satu tahun dan t adalah lamanya bulan dalam waktu. Dalam kode ini diwakilkan menjadi:

P = uang_awal

A= uang_akhir

r= bunga

n= periode_bunga_setahun

t= waktu

Untuk mencari waktu rumus tersebut dapat diubah menjadi :

$$t = \log(A / P) / (n \cdot \log(1 + r/n))$$

Dan jika kita tuliskan kedalam kode dapat ditulis **waktu = math.ceil (math.log (uang_akhir / uang_awal) / (periode_bunga_setahun * math.log (1 + bunga / periode_bunga_setahun)))**.

Dalam hal ini, fungsi **math.log()** digunakan untuk menghitung logaritma, sedangkan **uang_akhir / uang_awal** digunakan untuk menunjukkan perbandingan antara jumlah uang akhir dan modal awal. Bagian **math.log (1 + bunga / periode_bunga_setahun)** digunakan untuk menghitung faktor pertumbuhan bunga. Fungsi **math.ceil** adalah untuk melakukan pembulatan ke atas. Hal ini diperlukan karena perolehan bunga dilakukan setahun sekali sehingga pembulatan ke atas

dilakukan agar **uang_akhir** dapat mencapai target yaitu Rp 400.000.000. Terakhir variabel **waktu** digunakan untuk nilai yang tersimpan dalam variabel yakni waktu yang dibutuhkan pertahun agar dapat mencapai target.

Output:

```
Anaconda Prompt - python
(base) C:\Users\Acer>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.
>>> import math
>>> 
>>> uang_awal = 200000000
>>> uang_akhir = 400000000
>>> bunga = 0.1
>>> periode_bunga_setahun = 1
>>> waktu = math.ceil(math.log(uang_akhir/uang_awal) / (periode_bunga_setahun * math.log(1+bunga /periode_bunga_setahun)))
>>> 
>>> waktu
8
>>> |
```