



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	Raffiel Gideon Nyolo Nyolo
Nama Lengkap	71251214
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 level tinggi yang interpreted, mendukung Object Oriented Programming (OOP) dan memiliki sifat dynamic semantics. Menurut sebuah survey yang diselenggarakan oleh Stackoverflow. Survey dapat dilihat pada: (<https://insights.stackoverflow.com/survey/2019#technology>), Python merupakan salah satu bahasa pemrograman yang paling banyak dipakai setelah Javascript. Secara umum Python merupakan salah satu bahasa pemrograman yang paling populer di dunia karena aturan dan sintaksnya yang sederhana untuk dipelajari bagi pemula.

Kesederhanaan bentuk dan sintaks Python dapat dilihat pada program berikut yang menampilkan Hello World jika dibandingkan dengan java dan C:

Java:

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

C:

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

Python:

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

Berikut merupakan logo dari Python:



Gambar 1.1: Logo Python

Berikut ini merupakan kelebihan dari bahasa pemrograman Python:

- Python memiliki dukungan pustaka pihak ketiga yang sangat lengkap, sehingga dapat digunakan di berbagai bidang, terutama data science dengan adanya pustaka seperti Pandas, NumPy, TensorFlow, Keras, dan library machine learning lainnya.
- Python memiliki pustaka bawaan yang lengkap untuk mendukung kebutuhan dasar programmer, seperti basis data, jaringan, dan fitur sistem operasi.
- Python bersifat open source, bebas digunakan termasuk untuk keperluan komersial, dan terus dikembangkan sehingga fiturnya selalu bertambah.
- Python mudah dipelajari oleh pemula karena sintaksnya sederhana, mirip bahasa Inggris, dan didukung banyak dokumentasi serta sumber belajar.

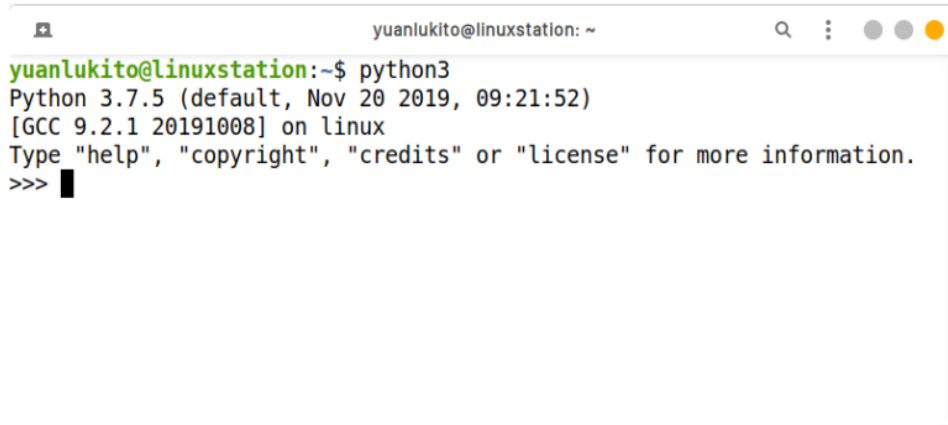
Python juga memiliki beberapa kekurangan, yaitu:

- Saat ini Python belum mendukung untuk pembuatan aplikasi di platform mobile seperti Android atau iOS.
- Konsumsi memory yang relatif besar sehingga tidak cocok digunakan untuk kasus-kasus yang membutuhkan memory dalam jumlah sangat besar.
- Kecepatan proses dari Python relatif lebih lambat jika dibandingkan dengan bahasa pemrograman seperti C.

Menginstall Python 3

Untuk saat ini terdapat dua macam versi Python yang banyak digunakan, yaitu Python versi 2 dan versi 3. Untuk keperluan mata kuliah ini digunakan Python 3 yang merupakan versi

yang lebih baru. Bagi anda yang menggunakan sistem operasi Linux maupun macOS, biasanya Python versi 3 sudah terinstall dan siap digunakan. Pada Ubuntu, Python dapat dijalankan pada terminal menggunakan perintah `python3` seperti yang ditunjukkan pada Gambar 1.2.



```
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.  
>>>
```

Gambar 1.2: Python versi 3 di Ubuntu Linux. Terinstall Python versi 3.7.5

Untuk perangkat windows, menginstall distribusi Anaconda yang relatif mudah untuk diinstall. Pilihlah Anaconda Individual Edition 64-Bit Graphical Installer, seperti pada Gambar 1.3.



Gambar 1.3: Distribusi Anaconda Individual Edition.

Menjalankan Python Mode Interaktif

Untuk masuk mode interaktif, jalankan perintah `python3` pada terminal di Ubuntu. Jika anda menggunakan Windows, jalankan Anaconda Prompt, kemudian ketikkan perintah `python`.

Editor untuk Python

Semakin canggih suatu editor, biasanya akan sangat mempermudah dalam pembuatan program. Beberapa editor yang biasanya dipakai untuk membuat program dalam bahasa pemrograman Python antara lain: Visual Studio Code + Python Extension for Visual Studio Code, PyCharm, Spyder, ActivePython dan IDLE.

Menjalankan Script Python di Terminal/Console

Mode interaktif Python menjalankan perintah satu per satu dan langsung menampilkan hasil, cocok untuk mencoba fungsi atau perintah baru. Kekurangannya, perintah harus diulang jika keluar dari mode ini. Selain itu ada mode script, yaitu menjalankan kumpulan perintah dalam file .py melalui Terminal atau Command Prompt dengan perintah python atau python3.

Mencari Bug dan Memperbaikinya (debugging)

Bug merupakan istilah dalam pemrograman yang berarti ada kesalahan dalam program. Kesalahan dalam membuat program secara umum dapat digolongkan menjadi dua macam, yaitu:

- Kesalahan dalam menulis program. Biasa disebut sebagai syntax error atau compile error. Kesalahan jenis ini biasanya diakibatkan oleh kesalahan dalam mengetik (typo). Jenis kesalahan ini mudah sekali untuk ditemukan dengan bantuan interpreter Python maupun editor yang dipakai.
- Kesalahan saat program berjalan. Biasa disebut sebagai runtime error. Jenis kesalahan ini lebih sulit ditemukan dan diperbaiki dibanding jenis kesalahan yang pertama.

Python merupakan bahasa pemrograman yang menggunakan sistem interpreter, artinya kode program anda akan dibaca baris-perbaris untuk dicek apakah ada kesalahan dalam penulisan perintah. Jika baris yang sedang dicek memenuhi aturan penulisan program (sintaks), maka baris tersebut langsung dijalankan. Jika terjadi kesalahan pada baris tersebut saat dijalankan, program Python anda akan langsung berhenti dan tidak melanjutkan ke baris berikutnya. Sebagai contoh, perhatikan kode program berikut ini:

```

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
14 print('Anda harus membayar: Rp. ', bayar)

```

Program tersebut menghitung jumlah yang harus dibayarkan jika belanja sebesar Rp. 100.000 dan mendapat diskon 30%. Jika script tersebut dijalankan, hasilnya akan seperti pada Gambar 1.4. Saat script dijalankan, tidak ada hasil yang dikeluarkan karena terjadi kesalahan pada baris ke-11. Ada keterangan jenis kesalahan, yaitu **NameError**. Kesalahan ini merupakan kesalahan sintaks karena ada aturan penulisan yang tidak terpenuhi. Kesalahan terjadi pada variabel bernama **nominaldiskon** yang belum didefinisikan tetapi nilainya sudah digunakan. Jika anda perhatikan kode programnya, pada baris **ke-8** ada variabel bernama **nominal_diskon**, tetapi pada baris **ke-11** dituliskan sebagai **nominaldiskon**. Terjadi kesalahan pengetikan (typo) yang menyebabkan script tidak bisa dijalankan.

```

yuanyukito@linuxstation: ~/Documents/python-code
yuanyukito@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
yuanyukito@linuxstation:~/Documents/python-code$

```

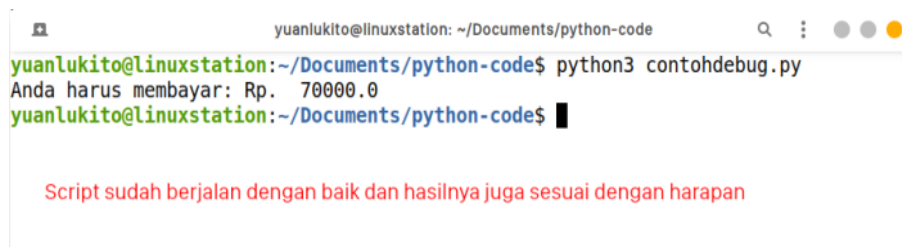
Gambar 1.4: Kesalahan yang muncul saat script dijalankan

Jika diperhatikan, script tersebut berjalan dengan hasil yang tidak sesuai dengan yang diharapkan. Jika kita membeli barang Rp. 100.000 dan mendapat diskon 30%, maka seharusnya yang harus dibayarkan adalah Rp. 70.000. Kesalahan ini adalah kesalahan jenis **runtime error**, yaitu kesalahan yang muncul saat program dijalankan. Kesalahan jenis ini biasanya diakibatkan oleh kesalahan algoritma, kesalahan logika atau kesalahan-kesalahan dalam mengatur langkah-langkah dalam program. Kesalahan terjadi pada baris ke-11, yaitu seharusnya jumlah yang harus

dibayar adalah harga barang dikurangi diskon. Perbaiki script tersebut menjadi seperti berikut ini:

```
1  #jumlah belanja (dalam rupiah)
2  belanja = 100000
3
4  #besarnya diskon (dalam persen)
5  diskon = 30
6
7  #besarnya diskon
8  nominaldiskon = (diskon/100) * belanja
9
10 #hitung jumlah yang harus dibayar
11 # INI SALAH: bayar = nominaldiskon - belanja
12 bayar = belanja - nominaldiskon
13
14 #tampilkan hasilnya
15 print('Anda harus membayar: Rp. ', bayar)
```

Setelah diperbaiki, script tersebut akan menghasilkan hasil yang benar sesuai dengan yang diharapkan. Hasilnya dapat dilihat pada Gambar 1.5.



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

Script sudah berjalan dengan baik dan hasilnya juga sesuai dengan harapan

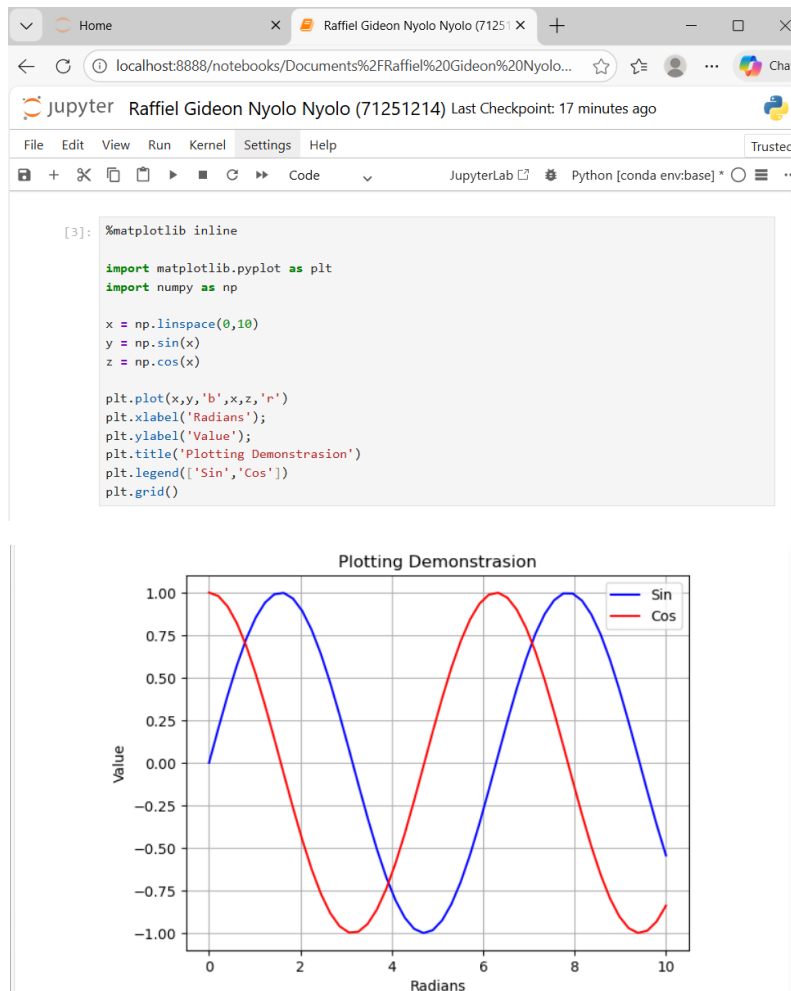
Gambar 1.5: Hasil sudah sesuai dengan yang diharapkan.

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.

Link Github: <https://github.com/fieliel-star/tugas-prakAlpro-minggu1-71251214.git>

SOAL 1



Gambar 2.1

Kode tersebut digunakan untuk membuat dan menampilkan grafik fungsi sinus dan cosinus pada Jupyter Notebook dengan memanfaatkan library NumPy dan Matplotlib, di mana perintah `%matplotlib inline` memastikan grafik langsung muncul di dalam notebook, kemudian NumPy digunakan untuk membangkitkan data x berupa sekumpulan nilai dari 0 hingga 10 dengan

jarak yang sama, selanjutnya nilai $\sin(x)$ dan $\cos(x)$ dihitung untuk setiap elemen x , hasil perhitungan tersebut kemudian dipetakan ke dalam satu grafik menggunakan Matplotlib dengan warna biru untuk grafik sinus dan warna merah untuk grafik cosinus agar mudah dibedakan, serta dilengkapi dengan label sumbu X “Radians” yang menunjukkan satuan sudut, label sumbu Y “Value” yang menunjukkan nilai fungsi, judul grafik “Plotting Demonstration” sebagai keterangan umum grafik, legenda untuk menjelaskan masing-masing kurva, dan garis grid untuk membantu pembacaan nilai grafik secara lebih jelas dan terstruktur.

SOAL 2

```
1  import math
2
3  berat_awal = 25
4  harga_awal = 650000
5  harga_naik = 685000
6  modal_awal = berat_awal * harga_awal
7  keuntungan = (berat_awal * harga_naik) - modal_awal
8
9  persen_keuntungan = (keuntungan / modal_awal) * 100
10
11 berat_akhir = 15
12 harga_akhir = 685000
13 harga_sekarang = 715000
14 total_berat = berat_awal + berat_akhir
15 total_modal = modal_awal + (berat_akhir * harga_akhir)
16 nilai_sekarang = total_berat * harga_sekarang
17 keuntungan2 = nilai_sekarang - total_modal
18
19 persen_keuntungan2 = (keuntungan2 / total_modal) * 100
20
21 print("Modal awal:" "Rp",modal_awal)
22 print("Keuntungan awal:" "Rp",keuntungan)
23 print("Persentase keuntungan awal:", round (persen_keuntungan),"%")
24
25 print("Total modal:" "Rp",total_modal)
26 print("Keuntungan sekarang:" "Rp",keuntungan2)
27 print("Persentase keuntungan sekarang:", round (persen_keuntungan2),"%")
```

Gambar 2.1: program menghitung modal, keuntungan, dan persentase keuntungan dari transaksi jual beli berdasarkan perubahan berat dan harga.

Penjelasan:

Program ini digunakan untuk menghitung modal, keuntungan, dan persentase keuntungan dari transaksi jual beli berdasarkan perubahan berat dan harga. Pada bagian awal, didefinisikan **berat_awal** sebesar 25 dan **harga_awal** 650000 yang kemudian digunakan untuk menghitung **modal_awal** sebagai hasil perkalian berat dan harga awal, lalu **harga_naik** sebesar 685000 dipakai untuk menghitung keuntungan, yaitu selisih antara nilai jual awal (berat awal dikali harga naik) dengan modal awal, setelah itu dihitung **persen_keuntungan** untuk mengetahui persentase keuntungan awal terhadap modal awal.

Selanjutnya program mensimulasikan kondisi lanjutan dengan menetapkan **berat_akhir** sebesar 15 dan **harga_akhir** 685000, kemudian menentukan **harga_sekarang** sebesar 715000, di mana **total_berat** merupakan penjumlahan berat awal dan berat akhir, sedangkan **total_modal** adalah akumulasi modal awal ditambah biaya pembelian berat akhir, lalu **nilai_sekarang** dihitung sebagai total berat dikali harga sekarang untuk mengetahui nilai jual keseluruhan saat ini. Dari nilai tersebut dihitung **keuntungan2** sebagai selisih antara nilai sekarang dan total modal, kemudian ditentukan **persen_keuntungan2** untuk mengetahui persentase keuntungan saat ini. Terakhir, program menampilkan seluruh hasil perhitungan ke layar menggunakan fungsi print, mulai dari modal awal, keuntungan awal, persentase keuntungan awal, total modal, keuntungan sekarang, hingga persentase keuntungan sekarang yang dibulatkan menggunakan "round" agar lebih mudah dibaca.

Jika dijalankan, program tersebut akan mengeluarkan hasil sebagai berikut:

```
Keuntungan awal:Rp 875000
Persentase keuntungan awal: 5 %
Total modal:Rp 26525000
Keuntungan sekarang:Rp 2075000
Persentase keuntungan sekarang: 8 %
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman\minggu1> █
```

Gambar 2.2: hasil dari program menghitung modal, keuntungan, dan persentase keuntungan dari transaksi jual beli berdasarkan perubahan berat dan harga.

SOAL 3

```
latuhanMandiriNo3 > mandiri3.py > ...
1 import math
2 modalAwal = 200000000
3 targetUang = 400000000
4 bungaPerTahun = 0.1
5 waktuYangDiButuhkan = math.log(targetUang / modalAwal) / math.log(1 + bungaPerTahun)
6
7 print ("Waktu yang dibutuhkan =", round(waktuYangDiButuhkan), "Tahun")
```

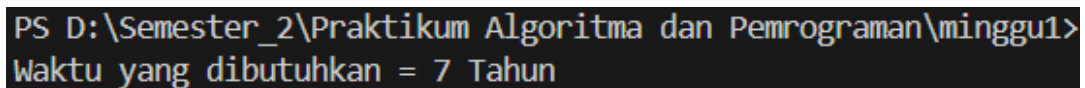
Gambar 2.3: menghitung perkiraan waktu yang dibutuhkan.

Penjelasan:

Program ini digunakan untuk menghitung perkiraan waktu yang dibutuhkan agar suatu modal awal mencapai target uang tertentu dengan asumsi pertumbuhan bunga majemuk per tahun. Pertama, modul math diimpor untuk menyediakan fungsi matematika, khususnya fungsi logaritma yang diperlukan dalam perhitungan. Variabel modalAwal diisi dengan nilai 200.000.000 sebagai jumlah uang awal, targetUang diisi 400.000.000 sebagai jumlah uang yang ingin dicapai, dan bungaPerTahun bernilai 0,1 yang berarti tingkat bunga 10% per tahun. Perhitungan waktu dilakukan menggunakan rumus logaritma

dari bunga majemuk, yaitu dengan membagi logaritma perbandingan antara target uang dan modal awal dengan logaritma dari $(1 + \text{bunga per tahun})$, sehingga diperoleh waktu yang dibutuhkan dalam satuan tahun. Nilai waktu tersebut kemudian dibulatkan menggunakan fungsi `round()` agar hasilnya lebih sederhana dan mudah dipahami, lalu ditampilkan ke layar dengan perintah `print` dalam kalimat “Waktu yang dibutuhkan = ... Tahun”, sehingga pengguna dapat mengetahui estimasi lama waktu yang diperlukan untuk mencapai target uang berdasarkan modal awal dan tingkat bunga yang telah ditentukan.

Jika dijalankan, program tersebut akan mengeluarkan hasil sebagai berikut:



```
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman\minggu1>  
Waktu yang dibutuhkan = 7 Tahun
```

Gambar 2.4: menghitung perkiraan waktu yang dibutuhkan.