

Nama: Ririn Anastasya

NIM: 1227030030

Mata Kuliah: Praktikum Fisika Komputasi

- Tugas 1 A

```
*PRAKFIKOM 2 (CONTOH SOAL 1A).py - C:\Users\Hp\OneDrive\Documents\PRAKFIKOM TUGAS 2\PRAKFIKOM 2 (CONTOH SOAL 1A).py (3.11.9)*
File Edit Format Run Options Window Help

n = 1.50
R1 = 22.0
R2 = 17.5

f_inv = (n - 1)*(1/R1 + 1/R2)
f = 1/f_inv
print("Jarak Fokus Lensa (f) = {:.2f} cm".format(f))

= RESTART: C:\Users\Hp\OneDrive\Documents\PRAKFIKOM TUGAS 2\PRAKFIKOM 2 (CONTO
H SOAL 1A).py
Jarak Fokus Lensa (f) = 19.49 cm
|
```

Kode pemrograman tersebut digunakan untuk menghitung jarak fokus sebuah lensa berdasarkan indeks bias medium serta jari-jari kelengkungan dua permukaan lensa. Pertama, kode mendefinisikan tiga bagian utama, yaitu indeks bias lensa ( $n = 1.50$ ), jari-jari kelengkungan permukaan pertama ( $R1 = 22.0$  cm), dan jari-jari kelengkungan permukaan kedua ( $R2 = 17.5$  cm). Nilai-nilai ini menentukan karakteristik optik lensa, indeks bias menunjukkan seberapa besar pembiasan cahaya yang terjadi saat melewati lensa, dan jari-jari kelengkungan menentukan seberapa cembung atau cekung permukaan lensa tersebut. Selanjutnya, kode menghitung kebalikan dari jarak fokus. Dalam perhitungan ini,  $(n - 1)$  merupakan pengaruh material lensa terhadap perubahan arah cahaya, sementara  $(1/R1 + 1/R2)$  merupakan kelengkungan kedua permukaan lensa. Setelah mendapatkan nilai  $(1/f)$ , kebalikannya diambil untuk menghitung jarak fokus sebenarnya ( $f$ ).

- Tugas 1 B  
(Variasi 1)

File Edit Format Run Options Window Help

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

alpha = np.radians(45)
g = 9.8
v0 = 10

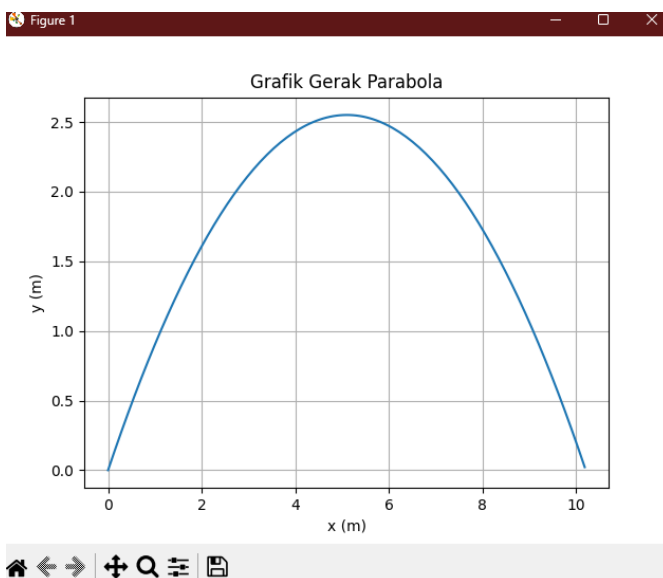
v0x = v0 * np.cos(alpha)
v0y = v0 * np.sin(alpha)

X = ((v0**2)*np.sin(2*alpha))/(2*g)
print("Jarak Horizontal Maksimum = ",X," m")
Y = ((v0**2)*(np.sin(alpha)**2))/(2*g)
print("Jarak Vertikal Maksimum = ",Y," m")
T = (2*v0*np.sin(alpha))/g
print("Waktu Mencapai Jarak Horizontal Maksimum = ",T," s")
print("\n")

t = np.arange(0.0, T, 0.01)
y = v0y*t - 0.5*g*t**2
x = v0x*t

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set(xlabel='x (m)', ylabel='y (m)', title='Grafik Gerak Parabola')
ax.grid()
plt.show()
```

```
>>>
= RESTART: C:\Users\Hp\OneDrive\Documents\PRAKFIKSKOM TUGAS 2\PRAKFIKSKOM 2 GRAFIK
PARABOLA.py
Jarak Horizontal Maksimum = 5.1020408163265305 m
Jarak Vertikal Maksimum = 2.5510204081632657 m
Waktu Mencapai Jarak Horizontal Maksimum = 1.4430750636460152 s
```



(Variasi 2)

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

alpha = np.radians(75)
g = 9.8
v0 = 20

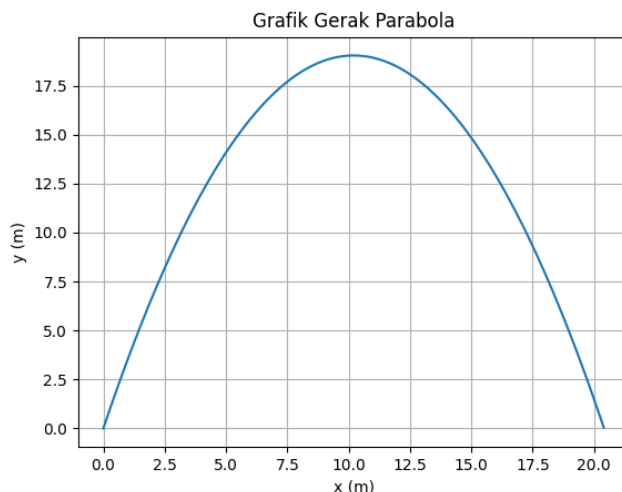
v0x = v0*np.cos(alpha)
v0y = v0*np.sin(alpha)

X = ((v0**2)*np.sin(2*alpha))/(2*g)
print("Jarak Horizontal Maksimum = ",X," m")
Y = ((v0**2)*(np.sin(alpha)**2))/(2*g)
print("Jarak Vertikal Maksimum = ",Y," m")
T = (2*v0*np.sin(alpha))/g
print("Waktu Mencapai Jarak Horizontal Maksimum = ",T," s")
print("\n")

t = np.arange(0.0, T, 0.01)
y = v0y*t - 0.5*g*t**2
x = v0x*t

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set(xlabel='x (m)', ylabel='y (m)', title='Grafik Gerak Parabola')
ax.grid()
plt.show()
```

```
= RESTART: C:/Users/Hp/OneDrive/Documents/PRAKFIKOM TUGAS 2/Contoh Variasi Alpha dan v0 (2).py
Jarak Horizontal Maksimum = 10.20408163265306 m
Jarak Vertikal Maksimum = 19.041075548820803 m
Waktu Mencapai Jarak Horizontal Maksimum = 3.942554393016605 s
```



Kode program ini digunakan untuk menghitung dan menampilkan lintasan dalam gerak parabola. Pertama, kode program menggunakan dua *library* penting, yaitu *numpy* untuk melakukan perhitungan matematika dan *matplotlib.pyplot* untuk menggambar grafik lintasan gerak. Setelah itu, beberapa variabel utama didefinisikan, seperti sudut  $\alpha$  yang diubah ke radian, percepatan gravitasi  $g$ , dan kecepatan awal  $v_0$ .

Selanjutnya, kode program menghitung komponen kecepatan awal, baik kecepatan horizontal ( $v_{0x}$ ) maupun vertikal ( $v_{0y}$ ), menggunakan fungsi cosinus dan sinus berdasarkan sudut yang diberikan. Dengan ini, kode program kemudian menghitung jarak horizontal maksimum yang bisa dicapai (menggunakan rumus trigonometri), tinggi maksimum yang bisa diraih, serta waktu yang dibutuhkan untuk mencapai jarak maksimum tersebut.

Setelah perhitungan selesai, kode program kemudian membuat grafik lintasan gerak parabola dengan menghitung posisi ketinggian pada berbagai waktu hingga mencapai waktu maksimum. Kode program ini menggambarkan bagaimana bergerak dari titik awal hingga kembali ke bawah, melihatkan jalur lengkung gerak parabola.