

PRAKTIKUM FISIKA KOMPUTASI VISUALISASI DATA

Eneng Yulia Pebryanti

NIM1227030013

File Edit Format Run Options Window Help

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

1. Pertama memasukan library **numpy** untuk perhitungan dan **matplotlib.pyplot** dimasukkan supaya pembuatan grafik bisa dilakukan.

```
g = 9.8          # percepatan gravitasi (m/s^2)
h0 = 42          # ketinggian awal (meter)
```

2. Memasukan nilai yang diketahui dalam soal dengan nilai gravitasi bumi sebesar 9,8 meter per detik kuadrat (m/s²), dan ketinggian awal benda ditentukan 42 meter.

```
# 1. Hitung waktu jatuh
t_fall = (2 * h0 / g) ** 0.5
print("Waktu untuk benda mencapai tanah: {:.2f} detik".format(t_fall))
```

3. **t_fall** untuk menyimpan hasil perhitungan dan ditampilkan dalam bentuk detik dengan dua angka di belakang koma menggunakan **print()**.

Bagian kode ini untuk menghitung waktu yang diperlukan oleh benda untuk mencapai tanah, menggunakan rumus

$$t = \sqrt{\frac{2h_0}{g}}$$

```
# 2. Fungsi untuk kecepatan dan posisi
def hitung_kecepatan(g, t):
    return g * t
```

```
def hitung_posisi(h0, g, t):
    return h0 - 0.5 * g * t**2
```

4. Dengan membuat fungsi **def**, bisa menghitung kecepatan untuk waktu yang berbeda-beda tanpa menulis ulang rumus setiap kali. Cukup fungsi ini dengan memasukkan nilai **g** dan **t**, maka hasil kecepatan akan didapatkan.

Sedangkan fungsi **return** digunakan untuk mengembalikan hasil dari perhitungan di dalam fungsi. Setelah fungsi melakukan tugasnya, return mengirimkan hasilnya kembali ke tempat di mana fungsi tersebut dipanggil.

Kode ini dibuat untuk menghitung posisi atau ketinggian benda selama waktu **t** tertentu.

Rumusnya $v(t) = g \cdot t$

yang berarti menghitung posisi benda dari ketinggian awal **h0** sampai akhirnya benda menyentuh tanah.

```
# Buat array waktu mulai dari 0 hingga t_fall
t_values = np.linspace(0, t_fall, 500)
```

5. Membuat array atau daftar waktu mulai dari 0 detik hingga waktu jatuh benda (**t_fall**), yang dibagi menjadi 500 titik waktu. Ini digunakan untuk menghitung kecepatan dan posisi pada setiap titik waktu
np.linspace, fungsi dari NumPy yang digunakan untuk membuat array (deret) angka dengan jarak yang sama.
t_values menyimpan hasil dari **np.linspace** dan akan berisi deret nilai waktu yang terdistribusi merata dari 0 hingga **t_fall** (waktu benda jatuh).

```
# Hitung kecepatan dan posisi untuk setiap waktu
kecepatan_values = hitung_kecepatan(g, t_values)
posisi_values = hitung_posisi(h0, g, t_values)
```

6. fungsi yang telah dibuat sebelumnya (**hitung_kecepatan** dan **hitung_posisi**) untuk menghitung kecepatan dan posisi benda pada setiap waktu dalam **t_values**.
Hasil kecepatan disimpan dalam **kecepatan_values**.
Hasil posisi disimpan dalam **posisi_values**.

```
# 3. Plot Grafik Kecepatan dan Posisi
plt.figure(figsize=(10, 5))
```

7. Menyiapkan area untuk menampilkan grafik dengan ukuran lebar 10 dan tinggi 5.

```
# Grafik kecepatan
plt.subplot(1, 2, 1)
plt.plot(t_values, kecepatan_values, color="blue")
plt.title("Kecepatan terhadap Waktu")
plt.xlabel("Waktu (detik)")
plt.ylabel("Kecepatan (m/s)")
plt.grid(True)
```

8. Membuat grafik pertama yang menunjukkan hubungan antara waktu dan kecepatan:
plt.subplot(1, 2, 1) berarti grafik ini berada di posisi pertama dari dua grafik yang ditampilkan berdampingan.
plt.plot(t_values, kecepatan_values) menampilkan grafik dengan sumbu X adalah waktu **t_values**, dan sumbu Y adalah kecepatan **kecepatan_values**. Dan colors diberi warna biru untuk menunjukkan warna hasil grafik
Grafik ini diberi judul "**Kecepatan terhadap Waktu**", dengan sumbu X menunjukkan "**Waktu (detik)**" dan sumbu Y menunjukkan "**Kecepatan (m/s)**".

```
# Grafik posisi
plt.subplot(1, 2, 2)
plt.plot(t_values, posisi_values, color="red")
plt.title("Posisi terhadap Waktu")
plt.xlabel("Waktu (detik)")
plt.ylabel("Posisi (meter)")
plt.grid(True)
```

9. Membuat grafik kedua yang menunjukkan hubungan antara waktu dan posisi:
plt.subplot(1, 2, 2) berarti grafik ini berada di posisi kedua dari dua grafik yang akan ditampilkan berdampingan. 1, 2 artinya gambar ini akan dibagi menjadi 1 baris dengan 2 kolom. Angka terakhir (2) menunjukkan bahwa ini adalah grafik kedua yang akan ditempatkan di kolom kedua (sebelah kanan).
plt.plot(t_values, posisi_values) menampilkan grafik dengan sumbu X adalah waktu **t_values**, dan sumbu Y adalah posisi **posisi_values**.
color="red": Garis grafik akan berwarna merah, untuk membedakan dengan grafik lain. Grafik ini diberi judul "**Posisi terhadap Waktu**", dengan sumbu X menunjukkan "**Waktu (detik)**" dan sumbu Y menunjukkan "**Posisi (meter)**".
plt.grid(True), garis grid (kotak-kotak) di dalam grafik agar lebih mudah melihat nilai posisi dan waktu pada titik-titik tertentu. True berarti grid akan ditampilkan.

```
# Tampilkan grafik
plt.tight_layout()
plt.show()
```

10. **plt.tight_layout()**: Mengatur agar grafik tersusun dengan rapi tanpa ada elemen yang bertumpukan.
plt.show(): Menampilkan grafik kecepatan dan posisi di layar setelah semuanya siap.

Lalu di run dan Saat kode dijalankan:

1. Nilai waktu yang diperlukan benda untuk mencapai tanah dihitung dan ditampilkan.
2. Kecepatan dan posisi benda dihitung untuk setiap titik waktu dari awal benda jatuh hingga mencapai tanah.
3. Dua grafik dibuat:
 - o Grafik pertama menunjukkan bagaimana kecepatan benda bertambah seiring waktu selama jatuh.
 - o Grafik kedua menunjukkan bagaimana ketinggian benda menurun hingga mencapai tanah.
4. Hasil akhirnya adalah dua grafik ini ditampilkan di layar.

