

DOKUMENTASI PROGRAM PYTHON

"CENTER OF MASS + VELOCITY"



Disusun oleh:
Yohanes Dimas Pratama
A11.2021.13254

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

DAFTAR ISI

BAB I – DASAR TEORI.....	3
1.1 Pusat Massa.....	3
1.2 Kecepatan Pusat Massa.....	3
BAB II - DEMO APLIKASI.....	4
2.1 Link Video Demo Aplikasi	4
2.2 Link Google Colab.....	4
2.3 Link GitHub.....	4
2.4 Code Program Menggunakan Google Collab	4
2.5 Hasil Menjalankan Menu Perhitungan Pusat Massa.....	5
2.5 Hasil Menjalankan Menu Perhitungan Kecepatan Pusat Massa	5
BAB III – UJI COBA PROGRAM.....	6
3.1 Perhitungan Pusat Massa	6
3.1.1 Contoh Soal.....	6
3.1.2 Perhitungan Secara Manual	6
3.1.3 Perhitungan Menggunakan Program.....	6
3.2 Perhitungan Kecepatan Pusat Massa.....	9
3.2.1 Contoh Soal.....	9
3.2.2 Perhitungan Secara Manual	9
3.2.3 Perhitungan Menggunakan Program.....	9
BAB IV - SOURCE CODE	12
BAB V – KESIMPULAN	15

BAB I – DASAR TEORI

1.1 Pusat Massa

Pusat massa adalah titik hipotetis dalam sebuah benda atau sistem benda di mana total massa dianggap terkonsentrasi. Dalam fisika, pusat massa sering kali digunakan untuk menyederhanakan analisis gerak, karena gerakan seluruh benda dapat direpresentasikan dengan gerakan titik ini.

Rumus persamaan pusat massa:

$$\vec{R} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i}$$

Keterangan:

- m_i adalah massa partikel ke-i
- \vec{r}_i adalah posisi partikel ke-i
- $\sum_i m_i$ adalah total massa sistem

1.2 Kecepatan Pusat Massa

Kecepatan pusat massa adalah vektor kecepatan yang menggambarkan kecepatan gerakan dari pusat massa suatu sistem benda atau partikel. Ini adalah vektor yang menunjukkan seberapa cepat dan dalam arah mana pusat massa sistem tersebut bergerak.

Rumus persamaan kecepatan pusat massa:

$$\vec{V}_{\text{cm}} = \frac{\sum_i m_i \vec{v}_i}{\sum_i m_i}$$

Keterangan:

- m_i adalah massa partikel ke-i
- \vec{v}_i adalah kecepatan partikel ke-i
- $\sum_i m_i$ adalah total massa sistem

BAB II - DEMO APLIKASI

2.1 Link Video Demo Aplikasi

<https://youtu.be/vcoB9M6bHXw>

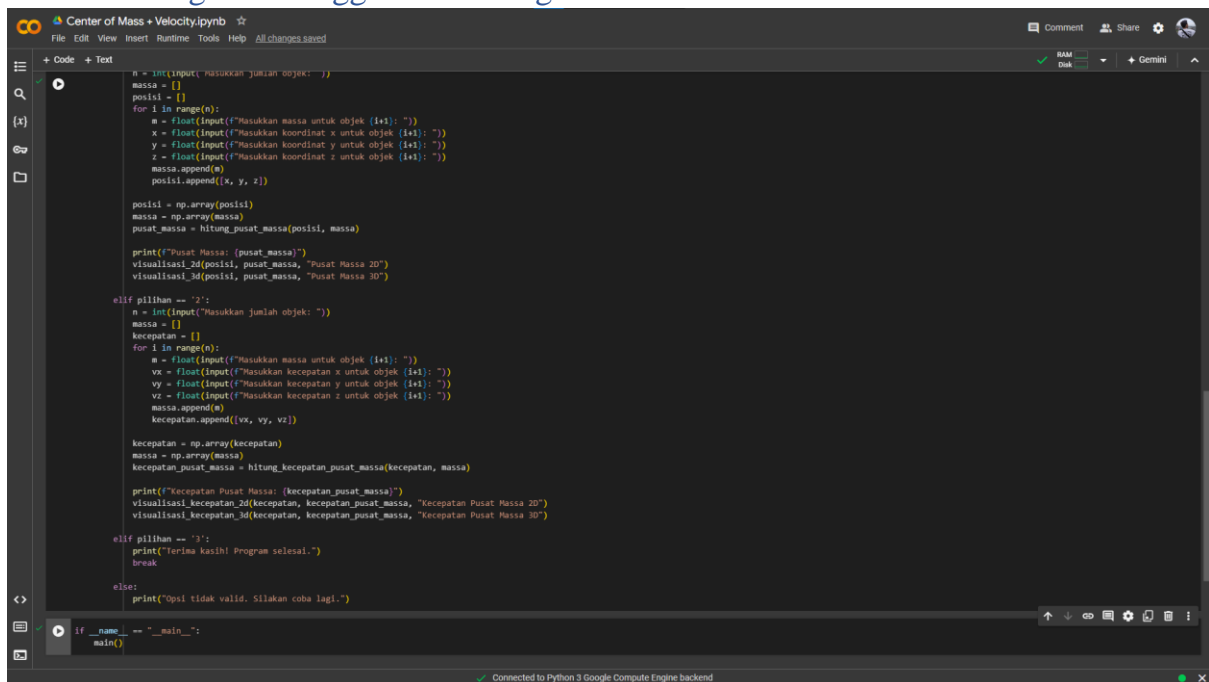
2.2 Link Google Colab

https://colab.research.google.com/drive/1fT-p60rKa2hoE9BJUU1_uBJVyTv30OWV?usp=sharing

2.3 Link GitHub

<https://github.com/weztcy/Program-Python-Center-of-Mass-Velocity>

2.4 Code Program Menggunakan Google Collab



```
Center of Mass + Velocity.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
n = int(input("Masukkan jumlah objek: "))
massa = []
posisi = []
for i in range(n):
    m = float(input(f"Masukkan massa untuk objek ({i+1}): "))
    x = float(input(f"Masukkan koordinat x untuk objek ({i+1}): "))
    y = float(input(f"Masukkan koordinat y untuk objek ({i+1}): "))
    z = float(input(f"Masukkan koordinat z untuk objek ({i+1}): "))
    massa.append(m)
    posisi.append([x, y, z])

posisi = np.array(posisi)
massa = np.array(massa)
pusat_massa = hitung_pusat_massa(posisi, massa)

print(f"Pusat Massa: {pusat_massa}")
visualisasi_2d(posisi, pusat_massa, "Pusat Massa 2D")
visualisasi_3d(posisi, pusat_massa, "Pusat Massa 3D")

elif pilihan == '2':
    n = int(input("Masukkan jumlah objek: "))
    massa = []
    kecepatan = []
    for i in range(n):
        m = float(input(f"Masukkan massa untuk objek ({i+1}): "))
        vx = float(input(f"Masukkan kecepatan x untuk objek ({i+1}): "))
        vy = float(input(f"Masukkan kecepatan y untuk objek ({i+1}): "))
        vz = float(input(f"Masukkan kecepatan z untuk objek ({i+1}): "))
        massa.append(m)
        kecepatan.append([vx, vy, vz])

    kecepatan = np.array(kecepatan)
    massa = np.array(massa)
    kecepatan_pusat_massa = hitung_kecepatan_pusat_massa(kecepatan, massa)

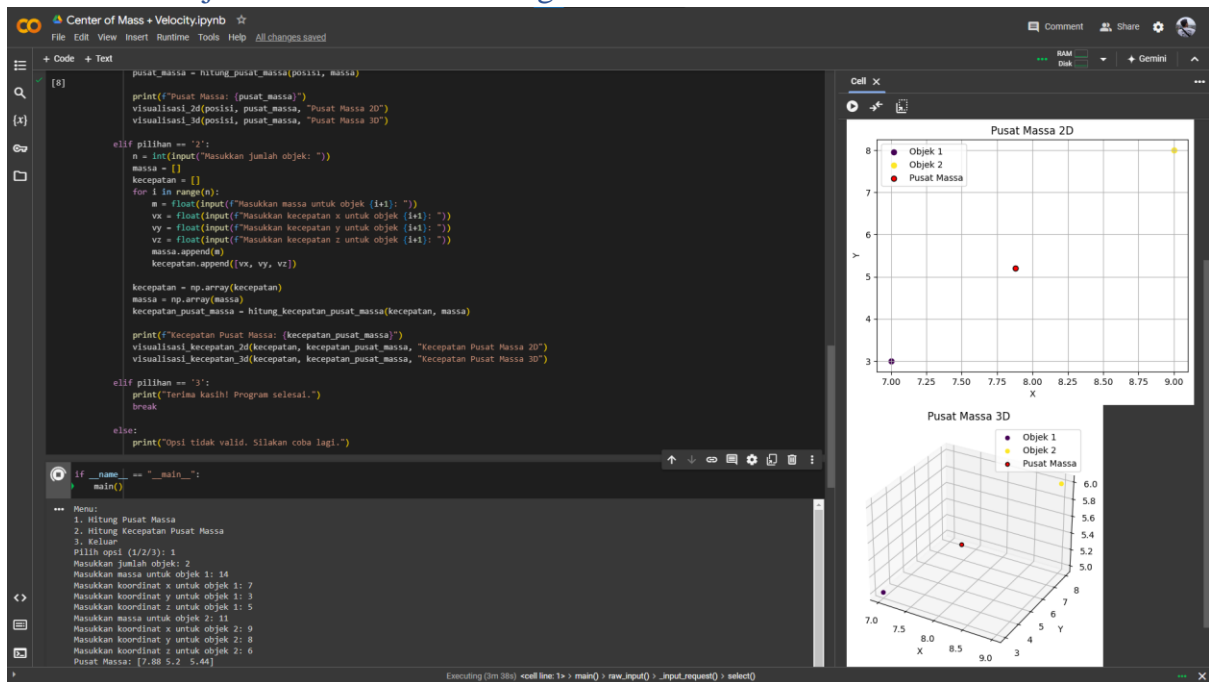
    print(f"Kecepatan Pusat Massa: {kecepatan_pusat_massa}")
    visualisasi_kecepatan_2d(kecepatan, kecepatan_pusat_massa, "Kecepatan Pusat Massa 2D")
    visualisasi_kecepatan_3d(kecepatan, kecepatan_pusat_massa, "Kecepatan Pusat Massa 3D")

elif pilihan == '3':
    print("Terima kasih! Program selesai.")
    break
else:
    print("Opsi tidak valid. Silakan coba lagi.")

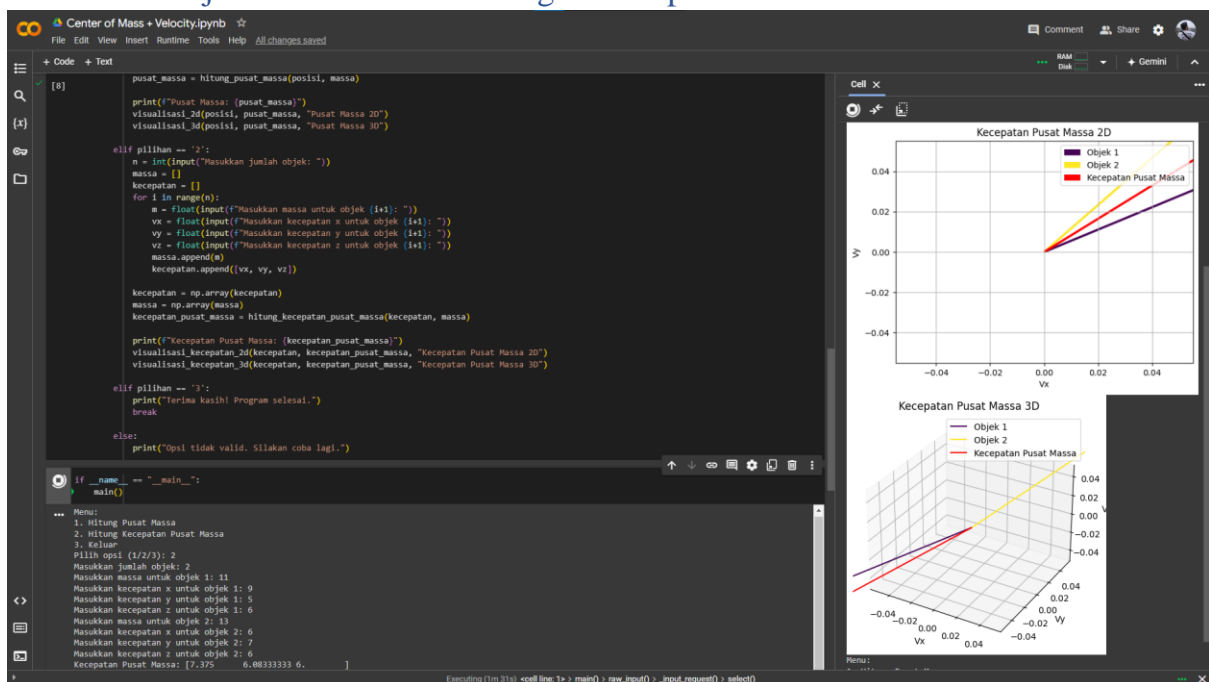
if __name__ == "__main__":
    main()
```

Connected to Python 3 Google Compute Engine backend

2.5 Hasil Menjalankan Menu Perhitungan Pusat Massa



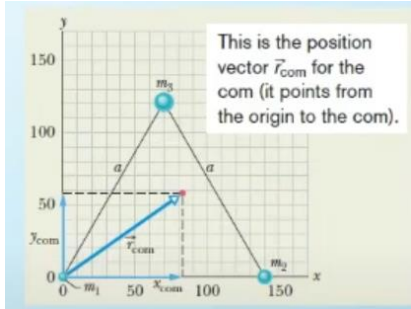
2.5 Hasil Menjalankan Menu Perhitungan Kecepatan Pusat Massa



BAB III – UJI COBA PROGRAM

3.1 Perhitungan Pusat Massa

3.1.1 Contoh Soal



Three particles of masses $m_1 = 1.2 \text{ kg}$, $m_2 = 2.5 \text{ kg}$, and $m_3 = 3.4 \text{ kg}$ form an equilateral triangle of edge length $a = 140 \text{ cm}$. Where is the center of mass of this system?

3.1.2 Perhitungan Secara Manual

Tabel Data

Objek	Massa (KG)	Koordinat X (CM)	Koordinat Y (CM)	Koordinat Z (CM)
1	1.2	0	0	0
2	2.5	140	0	0
3	3.4	70	120	0

Karena ketiga objek tersebut membentuk segitiga sama sisi maka dapat disimpulkan kalau ketiga objek tersebut berada pada dua dimensi saja yaitu X dan Y.

Masukkan ke dalam rumus, dan melakukan perhitungan sebagai berikut:

$$\begin{aligned}x_{com} &= \frac{1}{M} \sum_{i=1}^n m_i x_i = \frac{m_1 x_1 + m_2 x_2 + m_3 x_3}{M} \\&= \frac{(1.2 \text{ kg})(0) + (2.5 \text{ kg})(140 \text{ cm}) + (3.4 \text{ kg})(70 \text{ cm})}{7.1 \text{ kg}} \\&= 83 \text{ cm}\end{aligned}$$
$$\begin{aligned}y_{com} &= \frac{1}{M} \sum_{i=1}^n m_i y_i = \frac{m_1 y_1 + m_2 y_2 + m_3 y_3}{M} \\&= \frac{(1.2 \text{ kg})(0) + (2.5 \text{ kg})(0) + (3.4 \text{ kg})(120 \text{ cm})}{7.1 \text{ kg}} \\&= 58 \text{ cm}\end{aligned}$$

3.1.3 Perhitungan Menggunakan Program

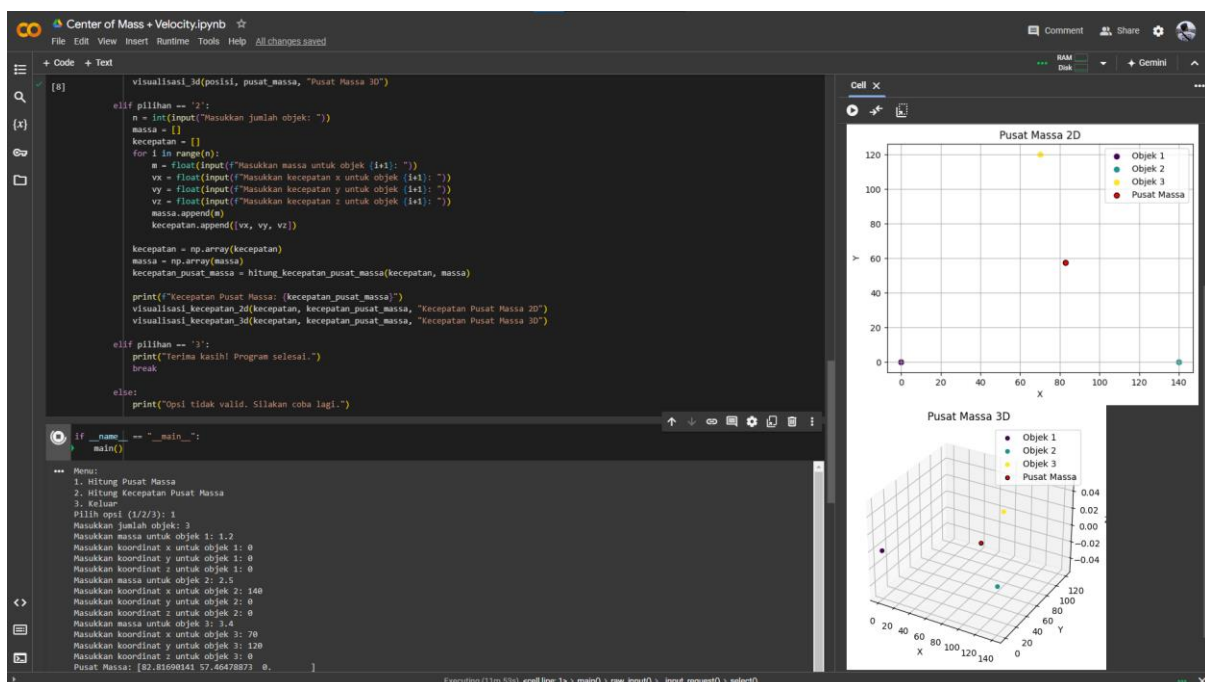
Menjalankan program dan memilih menu nomor 1 yaitu "Hitung Pusat Massa".

```
... Menu:
1. Hitung Pusat Massa
2. Hitung Kecepatan Pusat Massa
3. Keluar
Pilih opsi (1/2/3): 1
Masukkan jumlah objek: 
```

Mengisi input sesuai data yang ingin dihitung.

```
... Menu:
1. Hitung Pusat Massa
2. Hitung Kecepatan Pusat Massa
3. Keluar
Pilih opsi (1/2/3): 1
Masukkan jumlah objek: 3
Masukkan massa untuk objek 1: 1.2
Masukkan koordinat x untuk objek 1: 0
Masukkan koordinat y untuk objek 1: 0
Masukkan koordinat z untuk objek 1: 0
Masukkan massa untuk objek 2: 2.5
Masukkan koordinat x untuk objek 2: 140
Masukkan koordinat y untuk objek 2: 0
Masukkan koordinat z untuk objek 2: 0
Masukkan massa untuk objek 3: 3.4
Masukkan koordinat x untuk objek 3: 70
Masukkan koordinat y untuk objek 3: 120
Masukkan koordinat z untuk objek 3: 0
```

Menjalankan program.



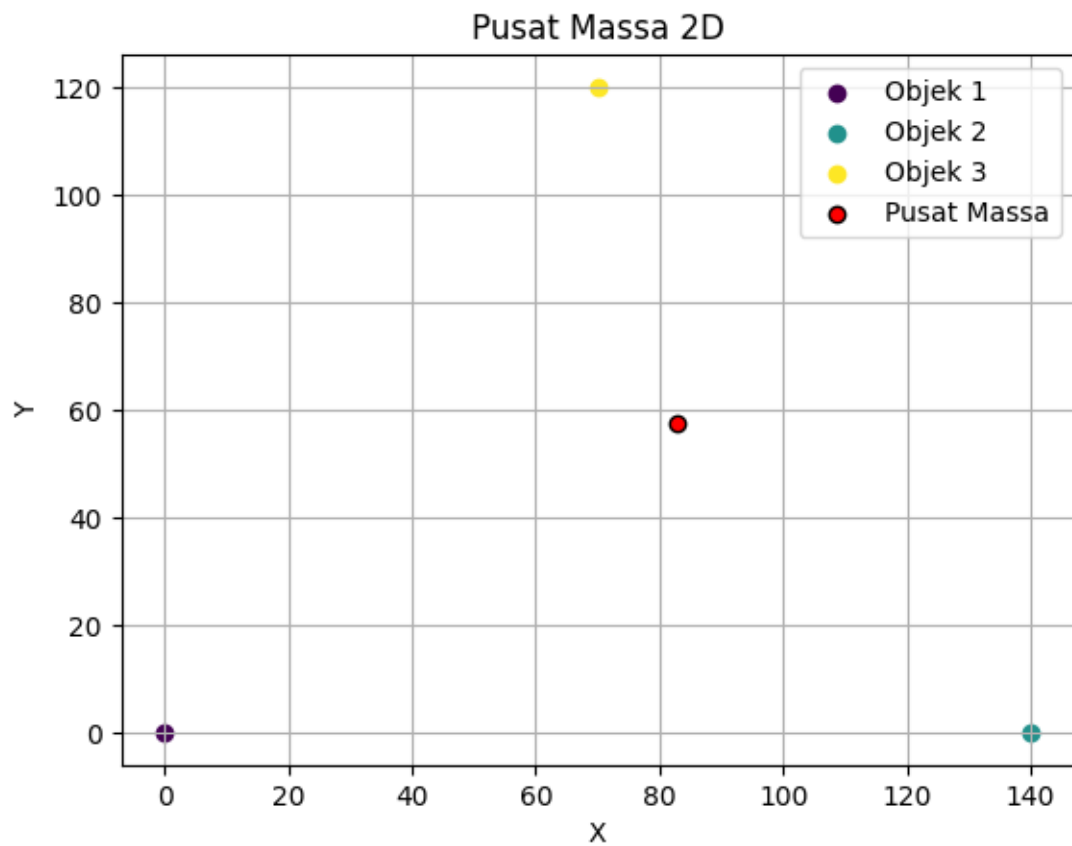
Hasil perhitungan pusat massa:

- Sumbu X = 82.81690141
- Sumbu Y = 57.46478873
- Sumbu Z = 0

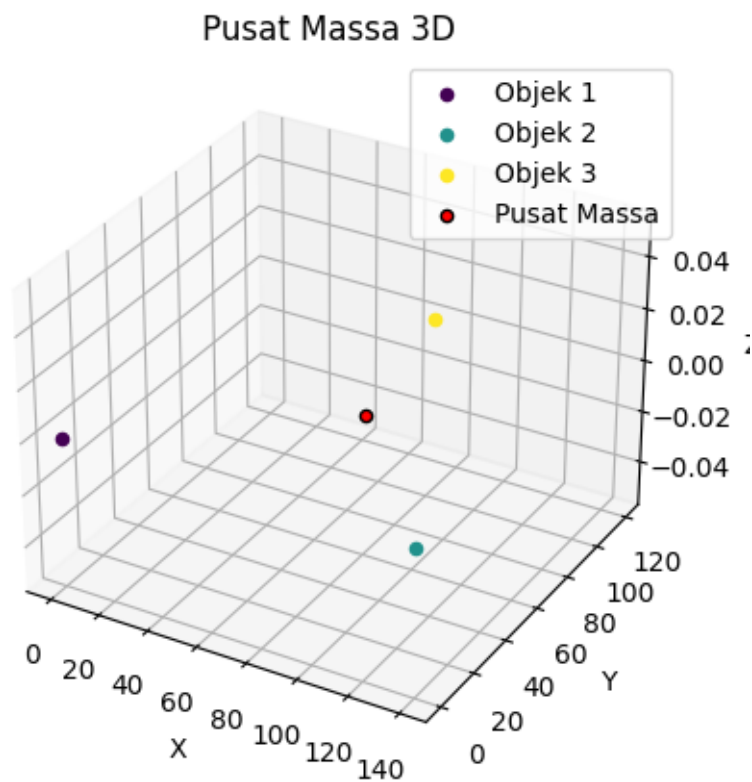
Dapat dilihat pada nilai Sumbu X = 83cm (82.8169014084507 digenapkan), Sumbu Y = 58cm (57.46478873239437 digenapkan), Sumbu Z = 0cm (karena segitiga sama sisi hanya dua dimensi). Maka dari itu disimpulkan hasil pengujian perhitungan pusat massa sudah memiliki akurasi yang tinggi.

Hasil visualisasi sumbu:

- Visualisasi 2D



- Visualisasi 3D



3.2 Perhitungan Kecepatan Pusat Massa

3.2.1 Contoh Soal

Dalam sebuah eksperimen di laboratorium, seorang mahasiswa fisika mengamati tiga objek yang bergerak sepanjang sumbu x. Objek pertama dengan massa 4 kg bergerak ke kanan dengan kecepatan 3 m/s. Objek kedua, yang beratnya hanya 2 kg, bergerak ke kiri dengan kecepatan 1 m/s. Sementara itu, objek ketiga, dengan massa 3 kg, bergerak ke kanan dengan kecepatan 2 m/s. Mahasiswa tersebut ingin menghitung kecepatan pusat massa dari ketiga objek tersebut untuk memahami dinamika gerakan mereka dalam satu dimensi.

3.2.2 Perhitungan Secara Manual

Tabel Data

Objek	Massa (KG)	Kecepatan X (m/s)	Kecepatan Y (m/s)	Kecepatan Z (m/s)
1	4	3	0	0
2	2	-1	0	0
3	3	2	0	0

Pada soal, kecepatan objek-objek hanya dinyatakan sepanjang sumbu x, dan tidak ada Sumbu kecepatan yang disebutkan untuk sumbu y atau z. Oleh karena itu, untuk kecepatan y dan z, nilai-nilainya diasumsikan sebagai 0 m/s, mengindikasikan bahwa objek-objek tersebut tidak memiliki gerakan vertikal atau dalam arah kedalaman.

Masukkan ke dalam rumus, dan melakukan perhitungan sebagai berikut:

Hitung total massa M_{total} :

$$M_{total} = m_1 + m_2 + m_3 = 4 \text{ kg} + 2 \text{ kg} + 3 \text{ kg} = 9 \text{ kg}$$

Hitung V_x :

$$V_x = \frac{m_1 \cdot v_{1x} + m_2 \cdot v_{2x} + m_3 \cdot v_{3x}}{M_{total}} = \frac{4 \cdot 3 + 2 \cdot (-1) + 3 \cdot 2}{9} = \frac{12 - 2 + 6}{9} = \frac{16}{9} \approx 1.78 \text{ m/s}$$

Hitung V_y dan V_z (diketahui semua kecepatan y dan z adalah 0):

$$V_y = \frac{m_1 \cdot v_{1y} + m_2 \cdot v_{2y} + m_3 \cdot v_{3y}}{M_{total}} = \frac{4 \cdot 0 + 2 \cdot 0 + 3 \cdot 0}{9} = 0 \text{ m/s}$$

$$V_z = \frac{m_1 \cdot v_{1z} + m_2 \cdot v_{2z} + m_3 \cdot v_{3z}}{M_{total}} = \frac{4 \cdot 0 + 2 \cdot 0 + 3 \cdot 0}{9} = 0 \text{ m/s}$$

3.2.3 Perhitungan Menggunakan Program

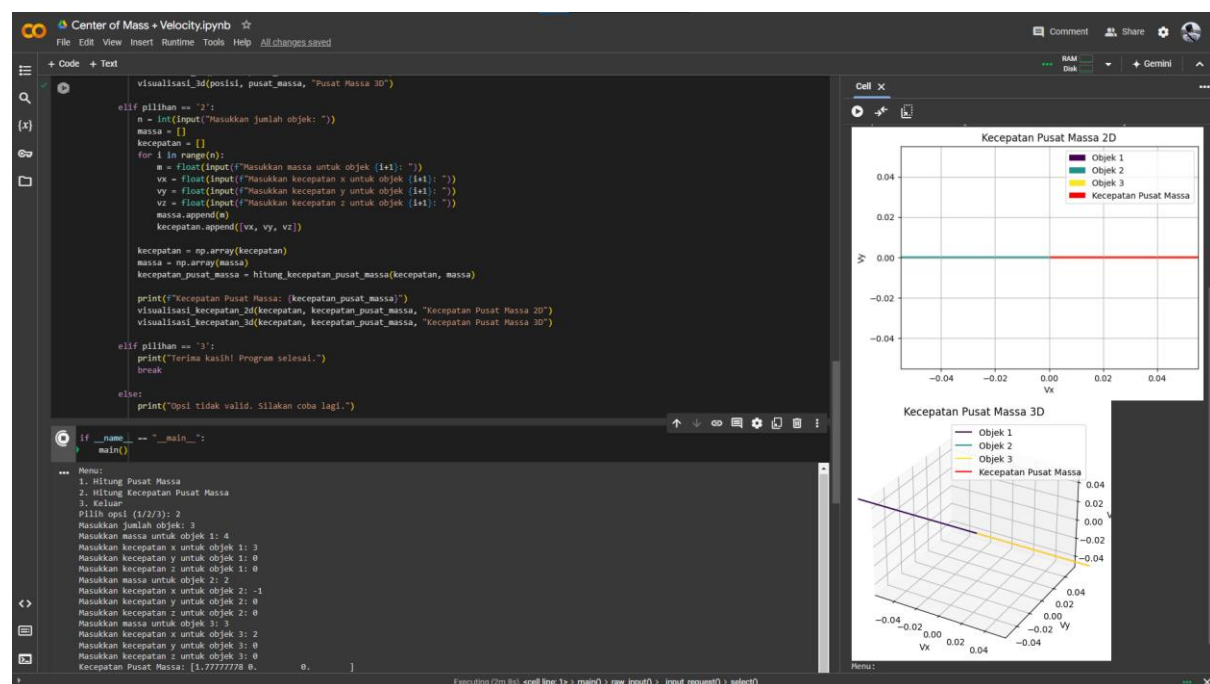
Menjalankan program dan memilih menu nomor 2 yaitu "Hitung Kecepatan Pusat Massa".

```
... Menu:
  1. Hitung Pusat Massa
  2. Hitung Kecepatan Pusat Massa
  3. Keluar
Pilih opsi (1/2/3): 2
Masukkan jumlah objek: 
```

Mengisi input sesuai data yang ingin dihitung.

```
... Menu:
1. Hitung Pusat Massa
2. Hitung Kecepatan Pusat Massa
3. Keluar
Pilih opsi (1/2/3): 2
Masukkan jumlah objek: 3
Masukkan massa untuk objek 1: 4
Masukkan kecepatan x untuk objek 1: 3
Masukkan kecepatan y untuk objek 1: 0
Masukkan kecepatan z untuk objek 1: 0
Masukkan massa untuk objek 2: 2
Masukkan kecepatan x untuk objek 2: -1
Masukkan kecepatan y untuk objek 2: 0
Masukkan kecepatan z untuk objek 2: 0
Masukkan massa untuk objek 3: 3
Masukkan kecepatan x untuk objek 3: 2
Masukkan kecepatan y untuk objek 3: 0
Masukkan kecepatan z untuk objek 3: 0
```

Menjalankan program.



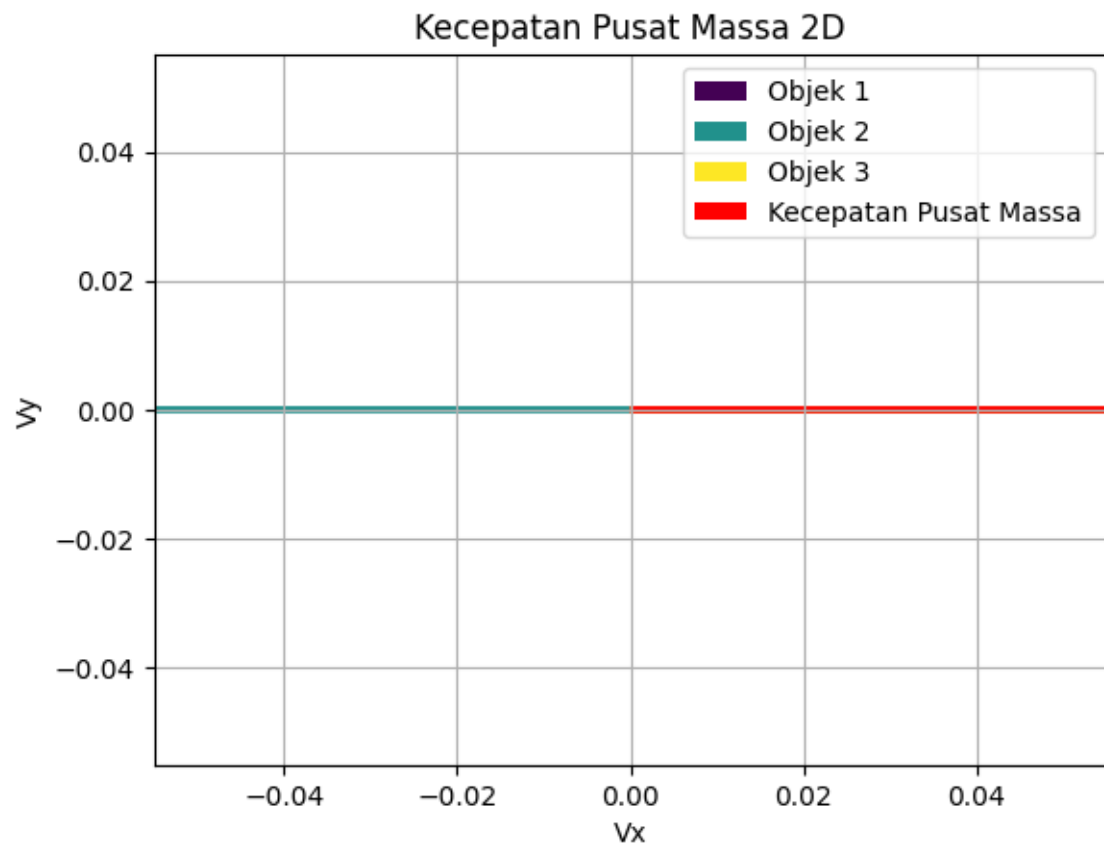
Hasil perhitungan kecepatan pusat massa:

- Sumbu X = 1.78
- Sumbu Y = 0
- Sumbu Z = 0

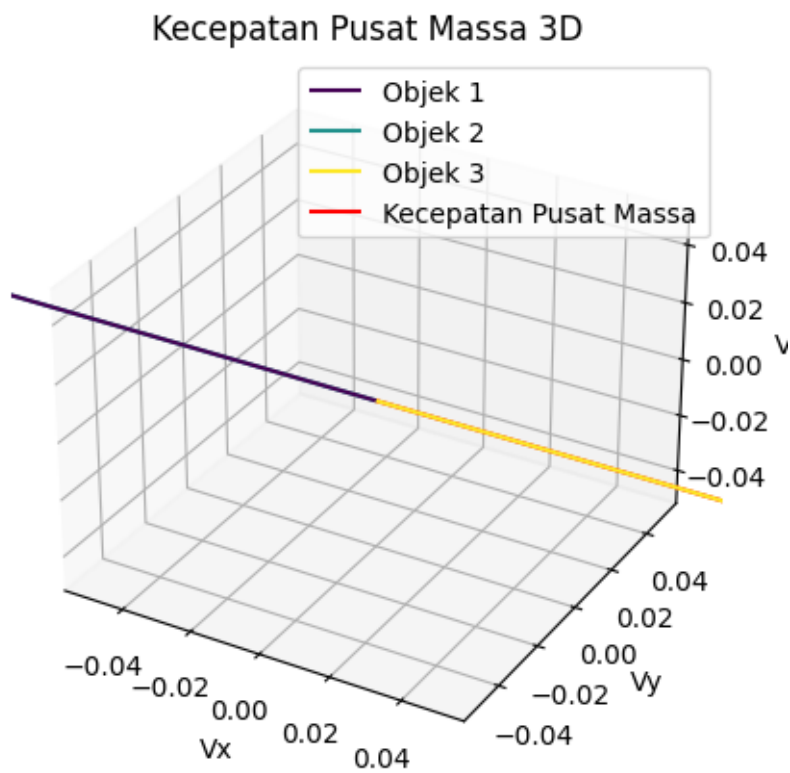
Dapat dilihat pada nilai Sumbu X = 1.78cm (1.77777778 digenapkan), Sumbu Y = 0cm, Sumbu Z = 0cm (karena kecepatan hanya pada satu dimensi). Maka dari itu disimpulkan hasil pengujian perhitungan kecepatan pusat massa sudah memiliki akurasi yang tinggi.

Hasil visualisasi sumbu:

- Visualisasi 2D



- Visualisasi 3D



BAB IV - SOURCE CODE

Center of Mass + Velocity.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```
def hitung_pusat_massa(posisi, massa):
    total_massa = np.sum(massa)
    pusat_massa = np.sum(posisi.T * massa, axis=1) / total_massa
    return pusat_massa
```

```
def hitung_kecepatan_pusat_massa(kecepatan, massa):
    total_massa = np.sum(massa)
    kecepatan_pusat_massa = np.sum(kecepatan.T * massa, axis=1) / total_massa
    return kecepatan_pusat_massa
```

```
def visualisasi_2d(posisi, pusat_massa, judul):
    plt.figure()
    warna = plt.cm.viridis(np.linspace(0, 1, len(posisi)))
    for i, pos in enumerate(posisi):
        plt.scatter(pos[0], pos[1], color=warna[i], label=f'Objek {i+1}')
    plt.scatter(pusat_massa[0], pusat_massa[1], color='red', label='Pusat Massa', edgecolor='black')
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title(judul)
    plt.legend()
    plt.grid(True)
    plt.show()
```

```
def visualisasi_3d(posisi, pusat_massa, judul):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    warna = plt.cm.viridis(np.linspace(0, 1, len(posisi)))
    for i, pos in enumerate(posisi):
        ax.scatter(pos[0], pos[1], pos[2], color=warna[i], label=f'Objek {i+1}')
    ax.scatter(pusat_massa[0], pusat_massa[1], pusat_massa[2], color='red', label='Pusat Massa',
edgecolor='black')
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    ax.set_title(judul)
    ax.legend()
    plt.show()
```

```
def visualisasi_kecepatan_2d(kecepatan, kecepatan_pusat_massa, judul):
    plt.figure()
    warna = plt.cm.viridis(np.linspace(0, 1, len(kecepatan)))
    for i, kecep in enumerate(kecepatan):
        plt.quiver(0, 0, kecep[0], kecep[1], color=warna[i], scale=1, scale_units='xy', angles='xy',
label=f'Objek {i+1}')
        plt.quiver(0, 0, kecepatan_pusat_massa[0], kecepatan_pusat_massa[1], color='red', scale=1,
scale_units='xy', angles='xy', label='Kecepatan Pusat Massa')
    plt.xlabel('Vx')
    plt.ylabel('Vy')
    plt.title(judul)
    plt.legend()
    plt.grid(True)
    plt.show()
```

```
def visualisasi_kecepatan_3d(kecepatan, kecepatan_pusat_massa, judul):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    warna = plt.cm.viridis(np.linspace(0, 1, len(kecepatan)))
    for i, kecep in enumerate(kecepatan):
        ax.quiver(0, 0, 0, kecep[0], kecep[1], kecep[2], color=warna[i], label=f'Objek {i+1}')
    ax.quiver(0, 0, 0, kecepatan_pusat_massa[0], kecepatan_pusat_massa[1],
kecepatan_pusat_massa[2], color='red', label='Kecepatan Pusat Massa')
    ax.set_xlabel('Vx')
    ax.set_ylabel('Vy')
    ax.set_zlabel('Vz')
    ax.set_title(judul)
    ax.legend()
    plt.show()
```

```
def main():
    while True:
        print("Menu:")
        print("1. Hitung Pusat Massa")
        print("2. Hitung Kecepatan Pusat Massa")
        print("3. Keluar")
        pilihan = input("Pilih opsi (1/2/3): ")

        if pilihan == '1':
            n = int(input("Masukkan jumlah objek: "))
            massa = []
            posisi = []
            for i in range(n):
                m = float(input(f"Masukkan massa untuk objek {i+1}: "))
                x = float(input(f"Masukkan koordinat x untuk objek {i+1}: "))
                y = float(input(f"Masukkan koordinat y untuk objek {i+1}: "))
                z = float(input(f"Masukkan koordinat z untuk objek {i+1}: "))
```

```

        massa.append(m)
        posisi.append([x, y, z])

posisi = np.array(posisi)
massa = np.array(massa)
pusat_massa = hitung_pusat_massa(posisi, massa)

print(f"Pusat Massa: {pusat_massa}")
visualisasi_2d(posisi, pusat_massa, "Pusat Massa 2D")
visualisasi_3d(posisi, pusat_massa, "Pusat Massa 3D")

elif pilihan == '2':
    n = int(input("Masukkan jumlah objek: "))
    massa = []
    kecepatan = []
    for i in range(n):
        m = float(input(f"Masukkan massa untuk objek {i+1}: "))
        vx = float(input(f"Masukkan kecepatan x untuk objek {i+1}: "))
        vy = float(input(f"Masukkan kecepatan y untuk objek {i+1}: "))
        vz = float(input(f"Masukkan kecepatan z untuk objek {i+1}: "))
        massa.append(m)
        kecepatan.append([vx, vy, vz])

    kecepatan = np.array(kecepatan)
    massa = np.array(massa)
    kecepatan_pusat_massa = hitung_kecepatan_pusat_massa(kecepatan, massa)

    print(f"Kecepatan Pusat Massa: {kecepatan_pusat_massa}")
    visualisasi_kecepatan_2d(kecepatan, kecepatan_pusat_massa, "Kecepatan Pusat Massa 2D")
    visualisasi_kecepatan_3d(kecepatan, kecepatan_pusat_massa, "Kecepatan Pusat Massa 3D")

elif pilihan == '3':
    print("Terima kasih! Program selesai.")
    break

else:
    print("Opsi tidak valid. Silakan coba lagi.")

```

```

if __name__ == "__main__":
    main()

```

BAB V – KESIMPULAN

Dokumentasi program "Center of Mass + Velocity" ini menjelaskan tentang sebuah program Python yang dirancang untuk menghitung pusat massa dan kecepatan pusat massa dari sejumlah objek, serta memvisualisasikan hasil perhitungannya. Program ini memanfaatkan pustaka NumPy untuk perhitungan numerik dan Matplotlib untuk visualisasi. Tujuan utama dari program ini adalah untuk menyederhanakan analisis gerak sistem benda dengan memusatkan massa dan kecepatan ke dalam satu titik hipotetis yang disebut pusat massa. Kecepatan pusat massa adalah vektor yang menunjukkan seberapa cepat dan dalam arah mana pusat massa sistem tersebut bergerak.

Program ini menawarkan dua fungsi utama, yaitu menghitung pusat massa dan menghitung kecepatan pusat massa. Pengguna diminta untuk memasukkan data massa serta posisi atau kecepatan objek-objek, kemudian program melakukan perhitungan dan menampilkan hasilnya dalam bentuk angka dan grafik 2D serta 3D. Dokumentasi ini juga mencakup dasar teori tentang pusat massa dan kecepatan pusat massa, yang membantu pengguna memahami konsep di balik perhitungan yang dilakukan.

Uji coba program dilakukan dengan beberapa contoh soal yang mencakup perhitungan manual dan perhitungan menggunakan program. Hasil dari uji coba ini menunjukkan bahwa program memiliki akurasi tinggi dalam menghitung pusat massa dan kecepatan pusat massa. Hasil perhitungan yang divisualisasikan dalam grafik 2D dan 3D membantu pengguna memahami distribusi massa dan gerakan dari sistem benda yang dianalisis. Dokumentasi ini secara keseluruhan menunjukkan bahwa program ini merupakan alat yang bermanfaat untuk menganalisis pusat massa dan kecepatan pusat massa dalam sistem multi-objek, serta memastikan bahwa hasil yang diberikan akurat dan dapat diandalkan.