1. Fathina Munifah
2. Yasmine khairatun
3. Gladiva Warouw
4. Budi Triadi
5. Ilham Saputra
6. Afrizal

1.Transformasi Koordinat

Pendahuluan Koordinat Silindris: Cara untuk merepresentasikan titik dalam ruang 3D menggunakan triplet terurut (r, θ, z). Koordinat Sferis: Representasi menggunakan ((\rho, \theta, \phi)). Rumus Konversi Dari Silindris ke Kartesian:

$x = r\cos\theta$ $y = r\sin\theta$ $z = z$ Dari Kartesian ke Silindris:

$r = \sqrt{x^2 + y^2}$ $\theta = \tan^{-1}(\frac{y}{x})$ $z = z$ Dari Sferis ke Kartesian:

$x = \rho\sin\phi\cos\theta$ $y = \rho\sin\phi\sin\theta$ $z = \rho\cos\phi$ Dari Kartesian ke Sferis:

$\rho = \sqrt{x^2 + y^2 + z^2}$ $\theta = \tan^{-1}(\frac{y}{x})$ $\phi = \cos^{-1}(\frac{z}{\rho})$

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

# Fungsi untuk mengonversi silindris ke kartesian
def silindris_ke_kartesian(r, theta, z):
    x = r * np.cos(theta)
    y = r * np.sin(theta)
    return x, y, z
```

[14]

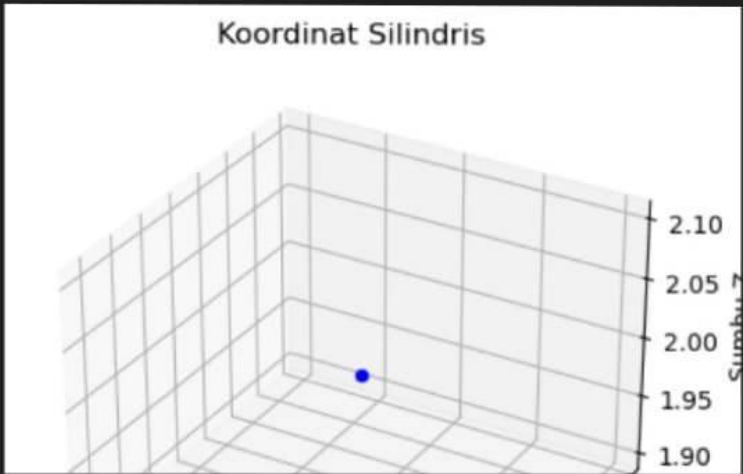
tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > M↓ 1.Transformasi Koordinat

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

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, color='b')
ax.set_xlabel('Sumbu X')
ax.set_ylabel('Sumbu Y')
ax.set_zlabel('Sumbu Z')
plt.title('Koordinat Silindris')
plt.show()
```

[15]



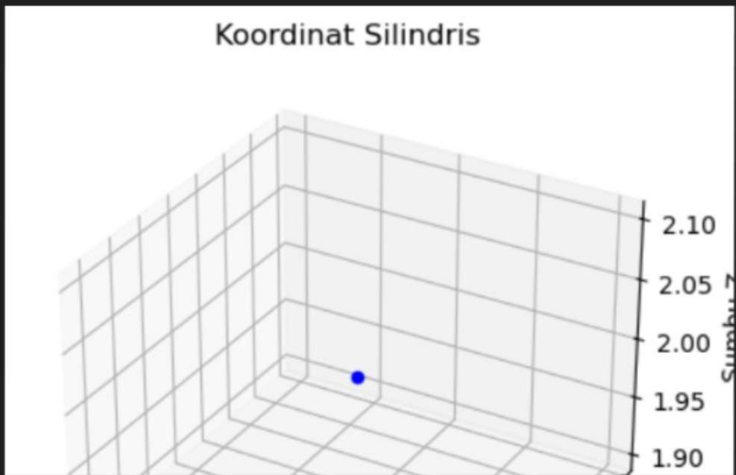
tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > 1.Transformasi Koordinat

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

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, color='b')
ax.set_xlabel('Sumbu X')
ax.set_ylabel('Sumbu Y')
ax.set_zlabel('Sumbu Z')
plt.title('Koordinat Silindris')
plt.show()
```

[15] Python

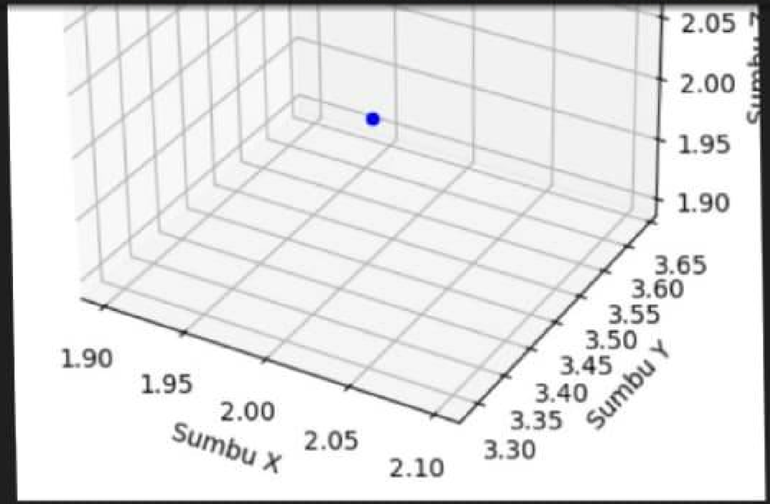


File Edit Selection View Go Run ··· Search

tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > 1.Transformasi Koordinat > import numpy as np

Generate + Code + Markdown | Run All Clear All Outputs | Outline ··· Select Kernel



Generate + Code + Markdown

2.Orthogonalitas Gram-Schmidt

Gram–Schmidt Orthogonalization

Prosedur Gram–Schmidt adalah algoritma dalam aljabar linear yang digunakan untuk mengubah himpunan vektor linier independen menjadi himpunan vektor ortonormal. Proses ini penting dalam berbagai aplikasi, seperti dekomposisi QR dan pemrosesan sinyal.

0 7 Spaces: 4 {} Cell 3 of 6

2.Orthogonalitas Gram-Schmidt

Gram–Schmidt Orthogonalization

Prosedur Gram–Schmidt adalah algoritma dalam aljabar linear yang digunakan untuk mengubah himpunan vektor linier independen menjadi himpunan vektor ortonormal. Proses ini penting dalam berbagai aplikasi, seperti dekomposisi QR dan pemrosesan sinyal.

🎯 Tujuan

Diberikan himpunan vektor linier independen $\{v_1, v_2, \ldots, v_m\}$ dalam ruang hasil kali dalam V, prosedur Gram–Schmidt menghasilkan himpunan vektor ortonormal $\{e_1, e_2, \ldots, e_m\}$ yang membentang pada subruang yang sama dengan himpunan awal.

🔁 Langkah-langkah

1. **Inisialisasi**: Set $u_1 = v_1$
2. **Iterasi**: Untuk $k = 2$ hingga m:

File Edit Selection View Go Run ···

tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > 1.Transformasi Koordinat > import numpy as np

Generate | + Code | + Markdown | Run All | Clear All Outputs | Outline ··· | Select Kernel

Langkah-langkah

1. **Inisialisasi**: Set $u_1 = v_1$
2. **Iterasi**: Untuk $k = 2$ hingga m:
   - Hitung proyeksi v_k ke setiap u_j sebelumnya:

$$\mathrm{proj}_{u_j}(v_k) = \frac{\langle v_k, u_j \rangle}{\langle u_j, u_j \rangle} u_j$$

   - Kurangi semua proyeksi dari v_k untuk mendapatkan u_k:

$$u_k = v_k - \sum_{j=1}^{k-1} \mathrm{proj}_{u_j}(v_k)$$

3. **Normalisasi**: Untuk setiap u_k, hitung:

$$e_k = \frac{u_k}{\|u_k\|}$$

Hasilnya adalah himpunan $\{e_1, e_2, \ldots, e_m\}$ yang ortonormal dan membentang pada subruang yang sama dengan himpunan awal.

Spaces: 4 | Cell 3 of 6

28°C Berawan | 9:27 AM 5/9/2025

```python
import numpy as np


def gram_schmidt(vectors):
    """
    Melakukan ortonormalisasi Gram-Schmidt pada himpunan vektor.
    """
    orthonormal_basis = []
    for v in vectors:
        w = v.copy()
        for u in orthonormal_basis:
            proj = np.dot(w, u) * u
            w = w - proj
        norm = np.linalg.norm(w)
        if norm < 1e-10:
            raise ValueError("Vektor linier dependen terdeteksi.")
        u = w / norm
        orthonormal_basis.append(u)
    return orthonormal_basis

# Contoh penggunaan
v1 = np.array([3, 1])
v2 = np.array([2, 2])

vectors = [v1, v2]
orthonormal_vectors = gram_schmidt(vectors)
```

File Edit Selection View Go Run ···

tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > M↓ 1.Transformasi Koordinat > import numpy as np

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

```python
# Contoh penggunaan
v1 = np.array([3, 1])
v2 = np.array([2, 2])

vectors = [v1, v2]
orthonormal_vectors = gram_schmidt(vectors)

for i, vec in enumerate(orthonormal_vectors, start=1):
    print(f"e{i} = {vec}")
```

Python

```
e1 = [0.9486833  0.31622777]
e2 = [-0.31622777  0.9486833 ]
```

```python
import numpy as np
import matplotlib.pyplot as plt
# Define input vectors (columns of matrix A)
A = np.array([[1, 1, 0], [1, 0, 1], [0, 1, 1]], dtype=float).T  # 3x3 matrix

# Define Gram-Schmidt function
def gram_schmidt(V):
    U = []
    for v in V.T:
        u = v.copy()
```

⊗ 0 ⚠ 7 Spaces: 4 Cell 3 of 6

28°C Berawan Search 9:27 AM 5/9/2025

```python
import numpy as np
import matplotlib.pyplot as plt
# Define input vectors (columns of matrix A)
A = np.array([[1, 1, 0], [1, 0, 1], [0, 1, 1]], dtype=float).T  # 3x3 matrix

# Define Gram-Schmidt function
def gram_schmidt(V):
    U = []
    for v in V.T:
        u = v.copy()
        for prev_u in U:
            proj = np.dot(u, prev_u) / np.dot(prev_u, prev_u) * prev_u
            u = u - proj
        U.append(u)
    return np.column_stack(U)

# Apply Gram-Schmidt process
orthogonal_vectors = gram_schmidt(A)

# Display original and orthogonalized vectors
print("Original Vectors (columns):\n", A)
print("\nOrthogonalized Vectors (columns):\n", orthogonal_vectors)

# Visualize in 3D
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
```

```python
# Visualize in 3D
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

origin = np.zeros(3)
colors = ['r', 'g', 'b']

for i in range(3):
    ax.quiver(*origin, *A[:, i], color=colors[i], label=f'Original v{i+1}', alpha=0.5)
    ax.quiver(*origin, *orthogonal_vectors[:, i], color=colors[i], linestyle='dashed', label=f'Ortho u{i+1}')

ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
ax.set_zlim([-2, 2])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Gram-Schmidt Orthogonalization')
ax.legend()
plt.show()
```

Python

```
Original Vectors (columns):
 [[1. 1. 0.]
 [1. 0. 1.]
 [0. 1. 1.]]
```

Spaces: 4 Cell 3 of 6

```python
import numpy as np


def gram_schmidt(vectors):
    """
    Melakukan ortonormalisasi Gram-Schmidt pada himpunan vektor.
    """
    orthonormal_basis = []
    for v in vectors:
        w = v.copy()
        for u in orthonormal_basis:
            proj = np.dot(w, u) * u
            w = w - proj
        norm = np.linalg.norm(w)
        if norm < 1e-10:
            raise ValueError("Vektor linier dependen terdeteksi.")
        u = w / norm
        orthonormal_basis.append(u)
    return orthonormal_basis

# Contoh penggunaan
v1 = np.array([3, 1])
v2 = np.array([2, 2])

vectors = [v1, v2]
orthonormal_vectors = gram_schmidt(vectors)
```

File Edit Selection View Go Run ··· Search

tugas geomath.ipynb

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > 1.Transformasi Koordinat > import numpy as np

Generate + Code + Markdown | Run All Clear All Outputs | Outline ··· Select Kernel

```python
# Contoh penggunaan
v1 = np.array([3, 1])
v2 = np.array([2, 2])

vectors = [v1, v2]
orthonormal_vectors = gram_schmidt(vectors)

for i, vec in enumerate(orthonormal_vectors, start=1):
    print(f"e{i} = {vec}")
```

Python

```
e1 = [0.9486833  0.31622777]
e2 = [-0.31622777  0.9486833 ]
```

```python
import numpy as np
import matplotlib.pyplot as plt
# Define input vectors (columns of matrix A)
A = np.array([[1, 1, 0], [1, 0, 1], [0, 1, 1]], dtype=float).T  # 3x3 matrix

# Define Gram-Schmidt function
def gram_schmidt(V):
    U = []
    for v in V.T:
        u = v.copy()
```

Spaces: 4 Cell 3 of 6

```python
import numpy as np
import matplotlib.pyplot as plt
# Define input vectors (columns of matrix A)
A = np.array([[1, 1, 0], [1, 0, 1], [0, 1, 1]], dtype=float).T  # 3x3 matrix

# Define Gram-Schmidt function
def gram_schmidt(V):
    U = []
    for v in V.T:
        u = v.copy()
        for prev_u in U:
            proj = np.dot(u, prev_u) / np.dot(prev_u, prev_u) * prev_u
            u = u - proj
        U.append(u)
    return np.column_stack(U)

# Apply Gram-Schmidt process
orthogonal_vectors = gram_schmidt(A)

# Display original and orthogonalized vectors
print("Original Vectors (columns):\n", A)
print("\nOrthogonalized Vectors (columns):\n", orthogonal_vectors)

# Visualize in 3D
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
```

```python
# Visualize in 3D
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

origin = np.zeros(3)
colors = ['r', 'g', 'b']

for i in range(3):
    ax.quiver(*origin, *A[:, i], color=colors[i], label=f'Original v{i+1}', alpha=0.5)
    ax.quiver(*origin, *orthogonal_vectors[:, i], color=colors[i], linestyle='dashed', label=f'Ortho u{i+1}')

ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
ax.set_zlim([-2, 2])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Gram-Schmidt Orthogonalization')
ax.legend()
plt.show()
```

Python

```
Original Vectors (columns):
 [[1. 1. 0.]
 [1. 0. 1.]
 [0. 1. 1.]]
```

Spaces: 4 Cell 3 of 6

```python
# Visualize in 3D
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

origin = np.zeros(3)
colors = ['r', 'g', 'b']

for i in range(3):
    ax.quiver(*origin, *A[:, i], color=colors[i], label=f'Original v{i+1}', alpha=0.5)
    ax.quiver(*origin, *orthogonal_vectors[:, i], color=colors[i], linestyle='dashed', label=f'Ortho u{i+1}')

ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
ax.set_zlim([-2, 2])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Gram-Schmidt Orthogonalization')
ax.legend()
plt.show()
```

Python

```
Original Vectors (columns):
 [[1. 1. 0.]
 [1. 0. 1.]
 [0. 1. 1.]]
```

Spaces: 4 Cell 3 of 6

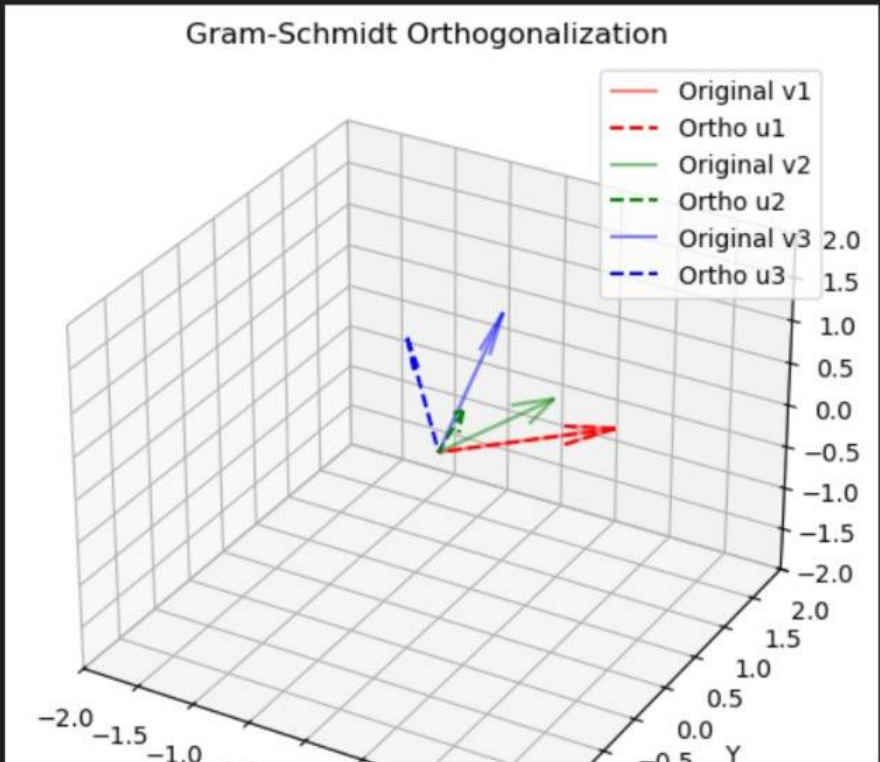
File Edit Selection View Go Run ··· Search

tugas geomath.ipynb ×

C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > M↓ 1.Transformasi Koordinat > import numpy as np

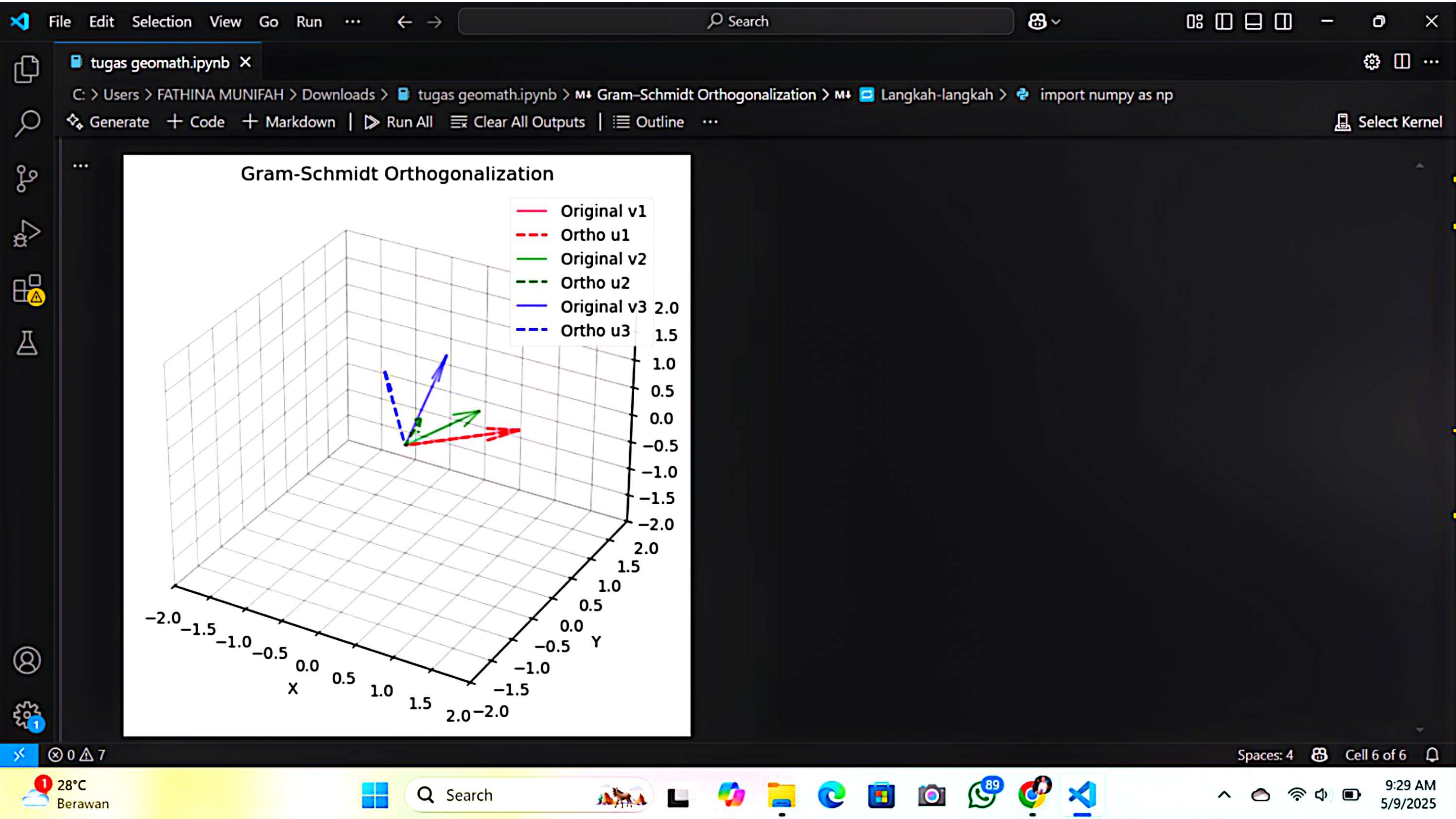
Generate + Code + Markdown | Run All Clear All Outputs | Outline ··· Select Kernel

```
Orthogonalized Vectors (columns):
 [[ 1.          0.5        -0.66666667]
 [ 1.         -0.5         0.66666667]
 [ 0.          1.          0.66666667]]
```



⊗ 0 ⚠ 7 Spaces: 4 Cell 3 of 6

28°C Berawan Search 9:29 AM 5/9/2025


File Edit Selection View Go Run ···
Search
tugas geomath.ipynb
C: > Users > FATHINA MUNIFAH > Downloads > tugas geomath.ipynb > Gram-Schmidt Orthogonalization > Langkah-langkah > import numpy as np
Generate + Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Gram-Schmidt Orthogonalization
Original v1
Ortho u1
Original v2
Ortho u2
Original v3
Ortho u3
X
Y
−2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0
2.0 1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5 −2.0
0 7
Spaces: 4
Cell 6 of 6
28°C
Berawan
Search
9:29 AM
5/9/2025

No.

Date

exercise 114

2. $\begin{pmatrix} 3 & 6 \\ 1 & 4 \end{pmatrix}$

Jawab

Mencari Eigenvalues

$|A - \lambda I| = 0$

$\begin{vmatrix} 3-\lambda & 6 \\ 1 & 4-\lambda \end{vmatrix} = (3-\lambda)(4-\lambda) - 6(1)$

$= (12 - 3\lambda - 4\lambda + \lambda^2) - 6 = \lambda^2 - 7\lambda + 6 = 0$

$(\lambda - 6)(\lambda - 1) = 0$

$\lambda_1 = 6$

$\lambda_2 = 1$

Mencari Eigenvector

~~8.~~ untuk $\lambda = 6$

Subtitusi ke $(A - 6I)$

$\begin{pmatrix} -3 & 6 \\ 1 & -2 \end{pmatrix}$ $-3x + 6y = 0 \Rightarrow x = 2y$

$y = 1, \quad x = 2$

Eigenvector : $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$

~~tam~~ $\lambda = 1$

$(A - I) = \begin{pmatrix} 2 & 6 \\ 1 & 3 \end{pmatrix}$

$2x + 6y = 0 \rightarrow x = -3$

$y = 1, \quad x = -3$

eigenvector : $\begin{pmatrix} -3 \\ 1 \end{pmatrix}$

Soal Nomor 2

Tentukan nilai eigen dan vektor eigen dari matriks berikut:

[

$$\begin{pmatrix} 3 & 6 \\ 1 & 4 \end{pmatrix}$$

]

```python
import numpy as np

A2 = np.array([[3, 6], [1, 4]])
vals2, vecs2 = np.linalg.eig(A2)
print(vals2)
print(vecs2)
```

[3] ✓ 0.0s

```
[1. 6.]
[[-0.9486833  -0.89442719]
 [ 0.31622777 -0.4472136 ]]
```

Gladiva Warouw
H061241061

Tugas Geo Matematika

25.5 Eigenvalues and eigenvectors

practice Exercise 114

For each of the following matrices, determine their (a) eigenvalues (b) eigenvectors

3.) $\begin{pmatrix} 3 & 1 \\ -2 & 0 \end{pmatrix}$ 7.) $\begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 2 \\ -1 & 1 & 3 \end{pmatrix}$

Answer

3.) $\begin{bmatrix} 3 & 1 \\ -2 & 0 \end{bmatrix}$

• eigenvalue (λ)

$\det(A - \lambda I) : 0$ Pers. karakteristik

$A - \lambda I : \begin{bmatrix} 3-\lambda & 1 \\ -2 & -\lambda \end{bmatrix}$

$\det(A - \lambda I) : (3-\lambda)(-\lambda) - (-2)(1) = -3\lambda + \lambda^2 + 2$

$= \lambda^2 - 3\lambda + 2$

$= \lambda^2 - 3\lambda + 2 = 0$

$(\lambda - 1)(\lambda - 2) = 0$

$\lambda_1 : 1 \quad \lambda_2 = 2$

• eigenvektor

$* \lambda : 1$

$(A - I)\vec{v} = 0$

$\begin{bmatrix} 3-1 & 1 \\ -2 & 0-1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix}$

$2x + y = 0 \rightarrow y : -2x$

$\vec{V}_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

$* \lambda = 2$

$A - 2I = \begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix}$

$x + y = 0 \rightarrow y : -x$

$\vec{V}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

EXERCISE 114 NOMOR 3

```python
import numpy as np

# Definisikan matriks
A = np.array([[3, 1], [-2, 0]])

# Hitung eigenvalues dan eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

print("Eigenvalues:")
print(eigenvalues)
print("\nEigenvectors:")
print(eigenvectors)
```

[2] ✓ 0.0s

```
Eigenvalues:
[2. 1.]

Eigenvectors:
[[ 0.70710678 -0.4472136 ]
 [-0.70710678  0.89442719]]
```

No.

Date

5. a.

$$A \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

$|A - \lambda I| = 0$

$$= \begin{vmatrix} 1-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix}$$

$$= (1-\lambda) \begin{vmatrix} 2-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} - (-1) \begin{vmatrix} -1 & -1 \\ 0 & 1-\lambda \end{vmatrix}$$

Minor pertama :

$$(2-\lambda)(1-\lambda) - (-1)(-1) = (\lambda^2 - 3\lambda + 1)$$

Minor kedua :

$$(-1)(1-\lambda) = -(1-\lambda)$$

Maka :

$$(1-\lambda)(\lambda^2 - 3\lambda + 1) + (1-\lambda) = (1-\lambda)(\lambda^2 - 3\lambda + 2)$$

$$= (1-\lambda)(\lambda - 2)(\lambda - 1)$$

Jadi $\lambda = 0$

$\lambda = 1$

$\lambda = 3$

No.

Date

mencari Eigenvector

$\lambda = 0$

$$A = \begin{pmatrix} 1 & -1 & 0 \\ -2 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

- $x - y = 0 \longrightarrow x = y$
- $-x + 2y - z = 0$
- $-y + z = 0 \longrightarrow z = y$

$y = 1$

eigenvector: $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

$\lambda = 1$

$$(A - I) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$-y = 0 \longrightarrow y = 0$

$-x - z = 0 \longrightarrow z = -x$

eigenvector $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$

$\lambda = 3$

$$(A - 3I) = \begin{pmatrix} -2 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -2 \end{pmatrix}$$

$-2x - y = 0 \longrightarrow y = -2x$

$-x - y - z = 0$

$-x + 2x - z = 0 \longrightarrow z = x$

$x = 1$

Eigenvector: $\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$

File Edit Selection View Go Run ··· geomath

EXPLORER ···

OPEN EDITORS 1 unsaved

tugas.ipynb

GEOMATH

geomath

helloworld.ipynb

sinh_cosh_plot.png

tugas.ipynb

UTS GEOMATH.ipynb

uts.ipynb

OUTLINE

TIMELINE

tugas.ipynb

tugas.ipynb > Exercise 114 > Soal Nomor 2

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· geomath (Python 3.13.2)

Soal nomor 5

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

```python
import numpy as np

A5 = np.array([[1, -1, 0], [-1, 2, -1], [0, -1, 1]])
vals5, vecs5 = np.linalg.eig(A5)
print(vals5)
print(vecs5)
```

[2] 0.0s Python

```
[ 3.00000000e+00  1.00000000e+00 -1.64111068e-16]
[[-4.08248290e-01 -7.07106781e-01  5.77350269e-01]
 [ 8.16496581e-01  4.24983316e-16  5.77350269e-01]
 [-4.08248290e-01  7.07106781e-01  5.77350269e-01]]
```

7. $\begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 2 \\ -1 & 1 & 3 \end{bmatrix}$

• eigenvalue

$\det(A - \lambda I) = 0$

$$(A - \lambda I) = \begin{bmatrix} 1-\lambda & 1 & 2 \\ 0 & 2-\lambda & 2 \\ -1 & 1 & 3-\lambda \end{bmatrix}$$

$$\det(A - \lambda I) = \begin{vmatrix} 1-\lambda & 1 & 2 \\ 0 & 2-\lambda & 2 \\ -1 & 1 & 3-\lambda \end{vmatrix}$$

ekspansi baris pertama

$$= (1-\lambda)\begin{vmatrix} 2-\lambda & 2 \\ 1 & 3-\lambda \end{vmatrix} - 1\begin{vmatrix} 0 & 2 \\ -1 & 3-\lambda \end{vmatrix} + 2\begin{vmatrix} 0 & 2-\lambda \\ -1 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} 2-\lambda & 2 \\ 1 & 3-\lambda \end{vmatrix} = (2-\lambda)(3-\lambda) - (2)(1) = (2-\lambda)(3-\lambda) - 2$$

$$= \begin{vmatrix} 0 & 2 \\ -1 & 3-\lambda \end{vmatrix} = 0(3-\lambda) - (2)(-1) = 2$$

$$= \begin{vmatrix} 0 & 2-\lambda \\ -1 & 1 \end{vmatrix} = 0(1) - (-1)(2-\lambda) = 2 - \lambda$$

$$\det = (1-\lambda)\left[(2-\lambda)(3-\lambda) - 2\right] - 2 + 2(2-\lambda)$$

$$= (2-\lambda)(3-\lambda) = 6 - 5\lambda + \lambda^2 \; -2 = 4 - 5\lambda + \lambda^2$$

$$\det = (1-\lambda)(\lambda^2 - 5\lambda + 4) - 2 + 4 - 2\lambda$$

$$= (1-\lambda)(\lambda^2 - 5\lambda + 4) + 2 - 2\lambda$$

$$(1-\lambda)(\lambda^2 - 5\lambda + 4) = \lambda^2 - 5\lambda + 4 - \lambda^3 + 5\lambda^2 - 4\lambda$$

$$= -\lambda^3 + 6\lambda^2 - 9\lambda + 4$$

$$-\lambda^3 + 6\lambda^2 - 9\lambda + 4 + 2 - 2\lambda = -\lambda^3 + 6\lambda^2 - 11\lambda + 6$$

$\rightarrow -\lambda^3 + 6\lambda^2 - 11\lambda + 6 = 0$ atau $\lambda^3 - 6\lambda^2 + 11\lambda - 6 = 0$

faktorkan $\lambda = 1$

$1^3 - 6(1)^2 + 11(1) - 6 = 0$

$(\lambda - 1)(\lambda^2 - 5\lambda + 6) = 0 \rightarrow (\lambda-1)(\lambda-2)(\lambda-3) = 0$

• eigenvektor

* $\lambda = 1$

$A - I \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \\ -1 & 1 & 2 \end{bmatrix}$ $y + 2z = 0 \rightarrow y = -2z$

$-x + y + 2z = 0$

$y = -2z \quad = -x - 2z + 2z = 0 \Rightarrow x = 0$

$x = 0 \quad y = -2 \quad z = 1$

$\vec{V}_1 = \begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix}$

* $\lambda = 2$

$A - 2I = \begin{bmatrix} -1 & 1 & 2 \\ 0 & 0 & 2 \\ -1 & 1 & 1 \end{bmatrix}$ $-x + y + 2z = 0$

$2z = 0 \rightarrow z = 0$

$-x + y = 0 \rightarrow x = y$

$x = 1 \quad y = 1 \quad z = 0$

$\vec{V}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

* $\lambda = 3$

$A - 3I = \begin{bmatrix} -2 & 1 & 2 \\ 0 & -1 & 2 \\ -1 & 1 & 0 \end{bmatrix}$ $-2x + y + 2z = 0$

$-y + 2z = 0 \rightarrow y = 2z$

$-x + y = 0 \rightarrow y = x$

$x = y = 2z \Rightarrow x = 2z$

$x = 2 \quad y = 2 \quad z = 1$

$\vec{V}_3 = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$

EXERCISE 114 NOMOR 7

```
import numpy as np

# Definisikan matriks
A = np.array([[1, 1, 2], [0, 2, 2], [-1, 1, 3]])

# Hitung eigenvalues dan eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

print("Eigenvalues:")
print(eigenvalues)
print("\nEigenvectors:")
print(eigenvectors)
```

[6] ✓ 0.0s

```
Eigenvalues:
[3. 2. 1.]

Eigenvectors:
[[ 6.66666667e-01 -7.07106781e-01 -2.76850919e-16]
 [ 6.66666667e-01 -7.07106781e-01  8.94427191e-01]
 [ 3.33333333e-01 -1.12262114e-16 -4.47213595e-01]]
```