

NAMA : PUTRI SALWA RAMADHANI

NPM : 233210216

KELAS : 2 E

1. TUGAS ARRAY

```
In [ ]: #perkalian matrix array

import numpy as np

A = np.array([[2,2],[1,0],[0,4]])
B = np.array([[9,1],[6,5],[0,5]])
C = np.array([[1,4],[7,0],[3,9]])

perkalian = A * B * C
print(perkalian)

[[ 18  8]
 [ 42  0]
 [  0 100]]
```

```
In [ ]: #pengurangan matrix array

import numpy as np

A = np.array([[2,2],[1,0],[0,4]])
B = np.array([[9,1],[6,5],[0,5]])
C = np.array([[1,4],[7,0],[3,9]])

pengurangan = A - B - C
print(pengurangan)

[[ -8 -3]
 [-12 -5]
 [ -3 -10]]
```

```
In [ ]: # Transpose Matrix

import numpy as np

A = np.array([[2,2],[1,0],[0,4]])
B = np.array([[9,1],[6,5],[0,5]])
C = np.array([[1,4],[7,0],[3,9]])

transpose_A = np.transpose(A)
transpose_B = np.transpose(B)
transpose_C = np.transpose(C)

print(transpose_A)
print(transpose_B)
print(transpose_C)

[[2 1 0]
 [2 0 4]]
[[9 6 0]
 [1 5 5]]
[[1 7 3]
 [4 0 9]]
```

```
In [ ]: # Nilai Invers

import numpy as np

A = np.array([[2,2],[1,0]])
B = np.array([[9,1],[6,5]])
C = np.array([[1,4],[7,0]])

inverse_A = np.linalg.inv(A)
inverse_B = np.linalg.inv(B)
inverse_C = np.linalg.inv(C)

print("\nInvers matrix A:")
print("\nInvers matrix B:")
print("\nInvers matrix C:")
print(inverse_A)
print(inverse_B)
print(inverse_C)

Invers matrix A:

Invers matrix B:

Invers matrix C:
[[ 0.  1. ]
 [ 0.5 -1. ]]
[[ 0.12820513 -0.02564103]
 [-0.15384615  0.23076923]]
[[ 0.  0.14285714]
 [ 0.25 -0.03571429]]
```

```
In [ ]: # Nilai Determinan

import numpy as np

A = np.array([[2,2],[1,0]])
B = np.array([[9,1],[6,5]])
C = np.array([[1,4],[7,0]])

determinant_A = np.linalg.det(A)
determinant_B = np.linalg.det(B)
determinant_C = np.linalg.det(C)
print("Determinan dari matrix A:", determinant_A)
print("Determinan dari matrix B:", determinant_B)
print("Determinan dari matrix C:", determinant_C)

Determinan dari matrix A: -2.0
Determinan dari matrix B: 38.99999999999999
Determinan dari matrix C: -27.999999999999996
```

2. TUGAS LIBRARY

```
In [ ]: import matplotlib.pyplot as plt

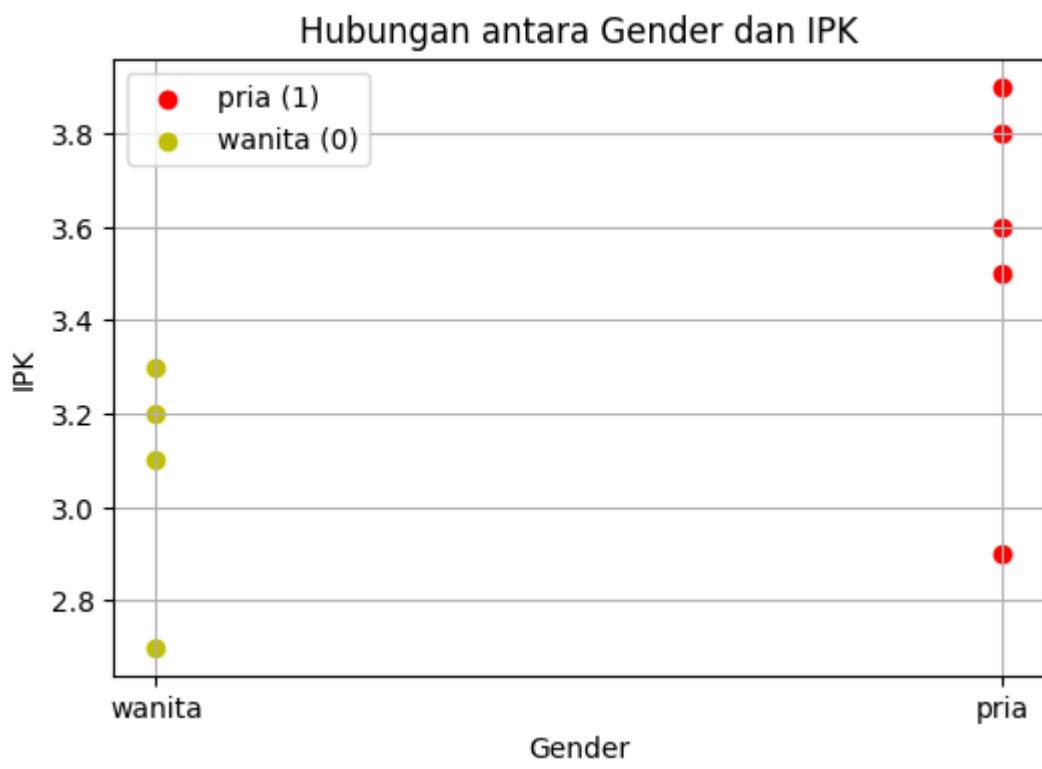
# Data
usia = [20, 22, 21, 23, 20, 22, 21, 24, 23]
ipk = [3.5, 3.2, 3.8, 2.9, 3.1, 3.6, 2.7, 3.9, 3.3]
gender = [1, 0, 1, 1, 0, 1, 0, 1, 0] # 1 for pria, 0 for wanita

plt.figure(figsize=(6, 4))

# Filtering IPK based on gender
pria = [ipk[i] for i in range(len(gender)) if gender[i] == 1]
wanita = [ipk[i] for i in range(len(gender)) if gender[i] == 0]

# Plotting scatter plot
plt.scatter([1]*len(pria), pria, color='r', label='pria (1)')
plt.scatter([0]*len(wanita), wanita, color='y', label='wanita (0)')

# Visualisasi
plt.xlabel('Gender')
plt.ylabel('IPK')
plt.title('Hubungan antara Gender dan IPK')
plt.legend()
plt.xticks([0, 1], ['wanita', 'pria'])
plt.grid(True)
plt.show()
```



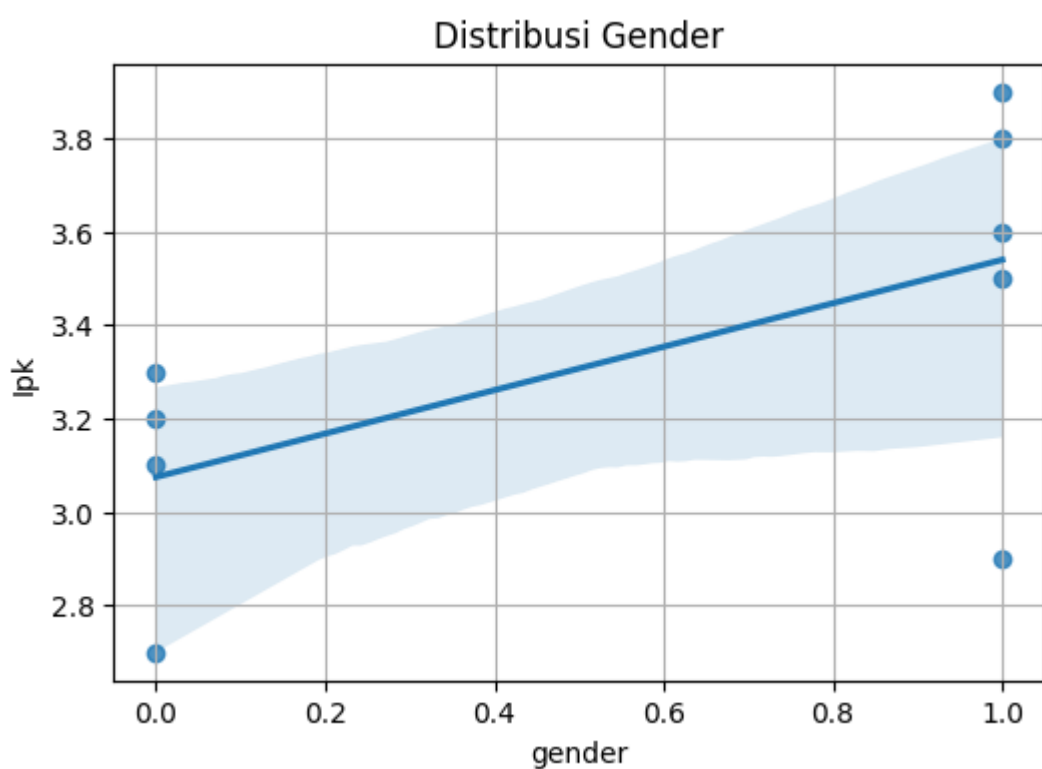
PENJELASAN CODING

- matplotlib.pyplot as plt, yang digunakan untuk membuat berbagai jenis plot dan grafik.
- Data ini mendefinisikan tiga daftar (list) yaitu usia, ipk, dan gender. Daftar usia berisi usia individu, daftar ipk berisi nilai IPK masing-masing individu, dan daftar gender berisi informasi gender (1 untuk pria dan 0 untuk wanita).
- plt.figure(figsize=(6, 4)) ini membuat sebuah figure (gambar) dengan ukuran 6x4 inci menggunakan plt.figure.
- Filtering IPK based on gender, Bagian ini membuat dua daftar baru, pria dan wanita, yang masing-masing berisi nilai IPK dari individu yang berjenis kelamin pria dan wanita. Baris ini menggunakan list comprehension untuk menyaring nilai-nilai IPK berdasarkan gender.
- plt.scatter([1]*len(pria), pria, color='r', label='pria (1)') membuat titik-titik merah di posisi x=1 untuk setiap nilai IPK pria. 6. plt.scatter([0]*len(wanita), wanita, color='y', label='wanita (0)') membuat titik-titik kuning di posisi x=0 untuk setiap nilai IPK wanita. 7. plt.xlabel('Gender') menambahkan label "Gender" pada sumbu x. 8. plt.ylabel('IPK') menambahkan label "IPK" pada sumbu y. 9. plt.title('Hubungan antara Gender dan IPK') menambahkan judul "Hubungan antara Gender dan IPK". 10. plt.legend() menampilkan legenda yang menjelaskan warna titik-titik. 11. plt.xticks([0, 1], ['wanita', 'pria']) mengganti tanda-tanda (ticks) pada sumbu x dari 0 dan 1 menjadi "wanita" dan "pria". 12. plt.grid(True) menambahkan garis kisi (grid) pada plot. 13. plt.show() menampilkan plot.

```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt

# Data Gender dan Ipk
gender = [1, 0, 1, 1, 0, 1, 0, 1, 0]
ipk = [3.5, 3.2, 3.8, 2.9, 3.1, 3.6, 2.7, 3.9, 3.3]

# Visualisasi
plt.figure(figsize=(6, 4))
sns.regplot(x=gender, y= ipk)
plt.title('Distribusi Gender')
plt.xlabel('gender')
plt.ylabel('Ipk')
plt.grid(True)
plt.show()
```



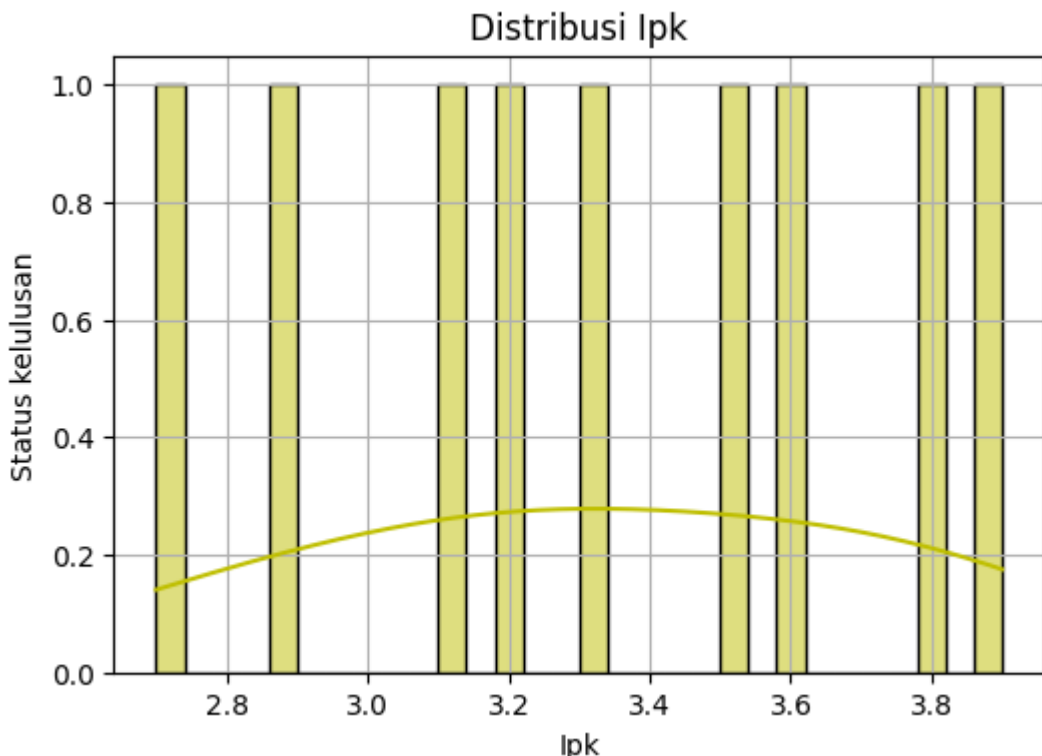
PENJELASAN CODING

- seaborn as sns dan matplotlib.pyplot as plt. Seaborn adalah pustaka visualisasi data berbasis matplotlib yang memberikan antarmuka tingkat tinggi untuk menggambar grafik statistik yang menarik, sementara matplotlib.pyplot adalah modul dasar untuk membuat berbagai jenis grafik.
- gender dan ipk. Daftar gender berisi informasi gender (1 untuk pria dan 0 untuk wanita), dan daftar ipk berisi nilai IPK masing-masing individu.
- plt.figure(figsize=(6, 4)). Baris ini membuat sebuah figure (gambar) dengan ukuran 6x4 inci menggunakan plt.figure.
- x=gender: Nilai pada sumbu x berasal dari daftar gender.
- y=ipk: Nilai pada sumbu y berasal dari daftar ipk.
- plt.title('Distribusi Gender') menambahkan judul "Distribusi Gender".
- plt.xlabel('gender') menambahkan label "gender" pada sumbu x.
- plt.ylabel('ipk') menambahkan label "ipk" pada sumbu y.
- plt.grid(True) menambahkan garis kisi (grid) pada plot.
- plt.show() menampilkan plot.

```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt

status_kelulusan = [1, 0, 1, 1, 0, 1, 0, 1, 0]
ipk = [3.5, 3.2, 3.8, 2.9, 3.1, 3.6, 2.7, 3.9, 3.3]

# Histogram
plt.figure(figsize=(6, 4))
sns.histplot(ipk, bins=30, kde=True, color='y')
plt.title('Distribusi Ipk')
plt.xlabel('Ipk')
plt.ylabel('Status kelulusan')
plt.grid(True)
plt.show()
```



PENJELASAN CODING

- seaborn as sns dan matplotlib.pyplot as plt. Seaborn adalah pustaka visualisasi data berbasis matplotlib yang memberikan antarmuka tingkat tinggi untuk menggambar grafik statistik yang menarik, sementara matplotlib.pyplot adalah modul dasar untuk membuat berbagai jenis grafik.
- status_kelulusan dan ipk. Daftar status_kelulusan berisi informasi status kelulusan (1 untuk lulus dan 0 untuk tidak lulus), dan daftar ipk berisi nilai IPK masing-masing individu.
- plt.figure(figsize=(6, 4)). Membuat sebuah figure (gambar) dengan ukuran 6x4 inci menggunakan plt.figure.
- sns.histplot(ipk, bins=30, kde=True, color='y'). Membuat histogram dari data ipk dengan 30 bin, menambahkan kernel density estimate (KDE) untuk menunjukkan distribusi data secara lebih halus, dan menggunakan warna kuning (color='y').
- plt.title('Distribusi Usia berdasarkan umur'). Menambahkan judul "Distribusi Ipk" pada plot.
- plt.xlabel('Ipk'). Menambahkan label "Ipk" pada sumbu x.
- plt.ylabel('Status kelulusan'). Menambahkan label "Status kelulusan" pada sumbu y. Label ini juga tidak sesuai dengan plot histogram yang menampilkan distribusi IPK, seharusnya lebih sesuai seperti "Frekuensi".
- plt.grid(True). Menambahkan garis kisi (grid) pada plot.
- plt.show(). Menampilkan plot.