

Tugas Pendahuluan Praktikum Dasar Kecerdasan Artifisial

Modul 4 : Pengantar Dasar NetworkX

INSTALLASI LIBRARY MK DKA

Jika belum terinstal, jalankan perintah berikut pada terminal atau notebook:

```
pip install jupyter ipykernel numpy matplotlib networkx sympy
scikit-fuzzy scipy
```

Petunjuk Umum

1. Jangan ubah struktur kode yang diberikan, cukup isi bagian yang kosong.
2. Jalankan setiap sel satu per satu agar tidak ada error.
3. Gunakan modul praktikum atau dokumentasi NetworkX
(<https://networkx.org/documentation/stable/>) jika membutuhkan referensi tambahan.
4. Kerjakan soal yang ada secara berurutan, dikarenakan soal akan meneruskan code soal sebelumnya

```
In [1]: print("Gavin Benjiro Ramadhan") # Tuliskan nama lengkap Anda
print("103012300452") # Tuliskan NIM Anda
print("IF-47-08") # Tuliskan kelas Anda
print("ABH (Klo ga salah)") # Tuliskan kode asprak anda
```

Gavin Benjiro Ramadhan
103012300452
IF-47-08
ABH (Klo ga salah)

Pengenalan NetworkX

NetworkX adalah pustaka Python yang digunakan untuk bekerja dengan **graf dan jaringan**.

Dengan NetworkX, kita dapat membuat **struktur graf**, menambahkan **simpul (nodes)** dan **sisi (edges)**, serta melakukan berbagai **analisis graf**.

Sebelum memulai, pastikan Anda telah mengimpor pustaka berikut:

```
In [3]: import networkx as nx # Import library NetworkX
import matplotlib.pyplot as plt # Import library Matplotlib
```

Soal 1: Membuat dan Menampilkan Graf Sederhana

1. Buat sebuah graf kosong menggunakan NetworkX.

2. Tambahkan simpul (node) "A", "B", "C", dan "D".
3. Tambahkan sisi (edge) antara:
 - "A" dan "B"
 - "A" dan "C"
 - "B" dan "D"
 - "C" dan "D"
4. Tampilkan graf yang telah dibuat dengan `nx.draw()`.

 Clue: Gunakan fungsi `add_node()`, `add_nodes_from()`, `add_edge()`, dan `add_edges_from()`.

Jangan lupa menampilkan graf dengan `plt.show()`.

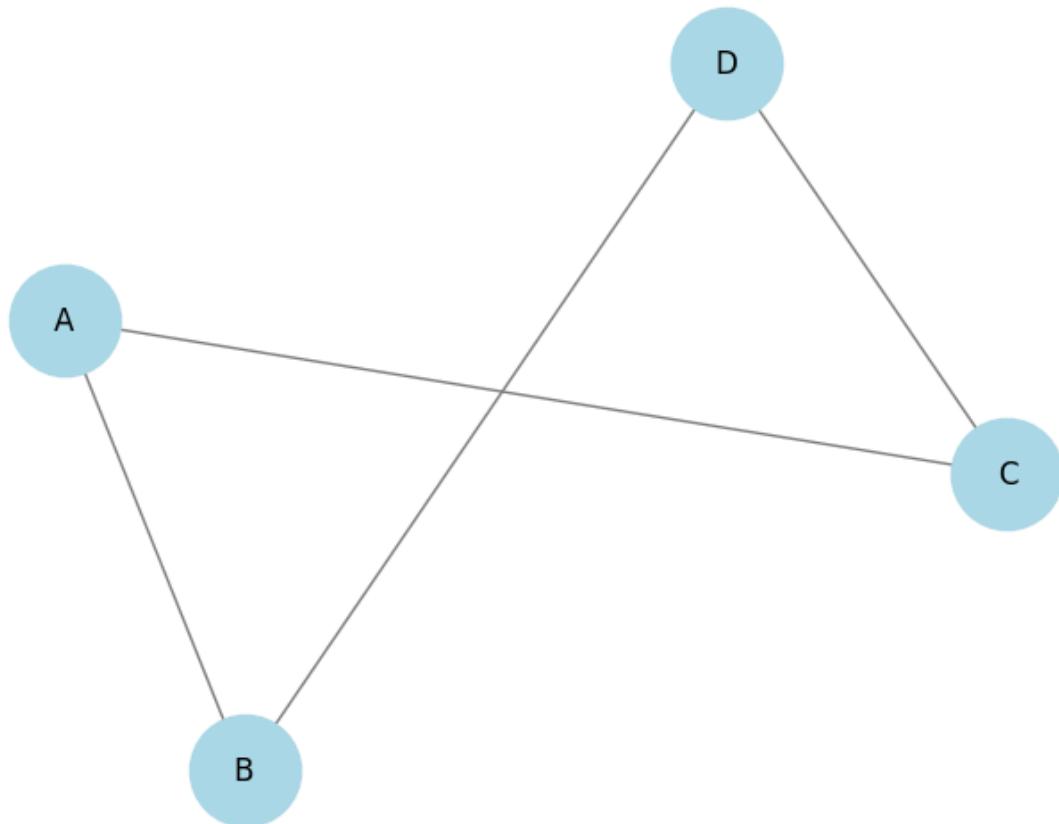
Isi bagian kosong di bawah ini:

```
In [8]: G = nx.Graph() # Buat graf kosong

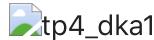
# Tambahkan simpul (node)
G.add_node("A") # Tambahkan satu simpul
G.add_nodes_from(["B", "C", "D"]) # Tambahkan beberapa simpul sekaligus

# Tambahkan sisi (edge)
G.add_edge("A", "B") # Tambahkan satu sisi
G.add_edges_from([('A', 'C'), ('B', 'D'), ('C', 'D')]) # Tambahkan beberapa sisi

# Tampilkan graf
nx.draw(G, with_labels=True, node_color="lightblue", edge_color="gray", n
plt.show()
```



Contoh output:



Soal 2: Memodifikasi Graf

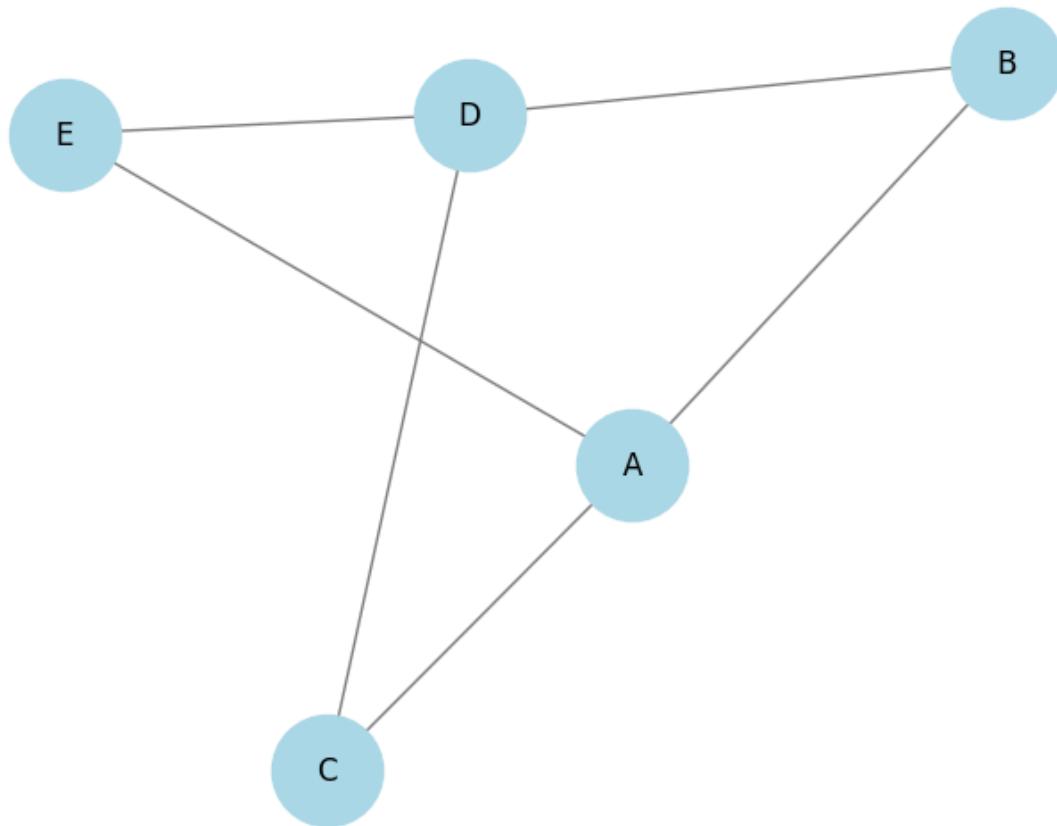
1. **Tambahkan simpul baru** bernama "E" ke dalam graf.
2. **Hubungkan simpul "E"** ke "A" dan "D".
3. **Tampilkan kembali graf yang telah diperbarui.**

Isi bagian kosong di bawah ini:

In [10]:

```
# Soal 2: Menambahkan simpul dan sisi baru
G.add_node("E") # Tambahkan simpul E
G.add_edges_from([('E', "A"), ("E", "D")]) # Hubungkan E ke A dan D

# Tampilkan graf yang diperbarui
nx.draw(G, with_labels=True, node_color="lightblue", edge_color="gray", n
plt.show()
```



Contoh *output*:



Soal 3: Membuat Graf Baru dengan 5 Simpul dan 6 Sisi

1. **Buat graf baru** yang berisi 5 simpul bernama X, Y, Z, W, V.
2. **Tambahkan 6 sisi** sesuai kebutuhan.

3. Tampilkan graf dengan warna berbeda untuk simpul dan sisi.

 Clue: Gunakan warna berbeda di `nx.draw()` seperti:

- `node_color="pink"` untuk simpul berwarna merah muda
- `edge_color="blue"` untuk sisi berwarna biru

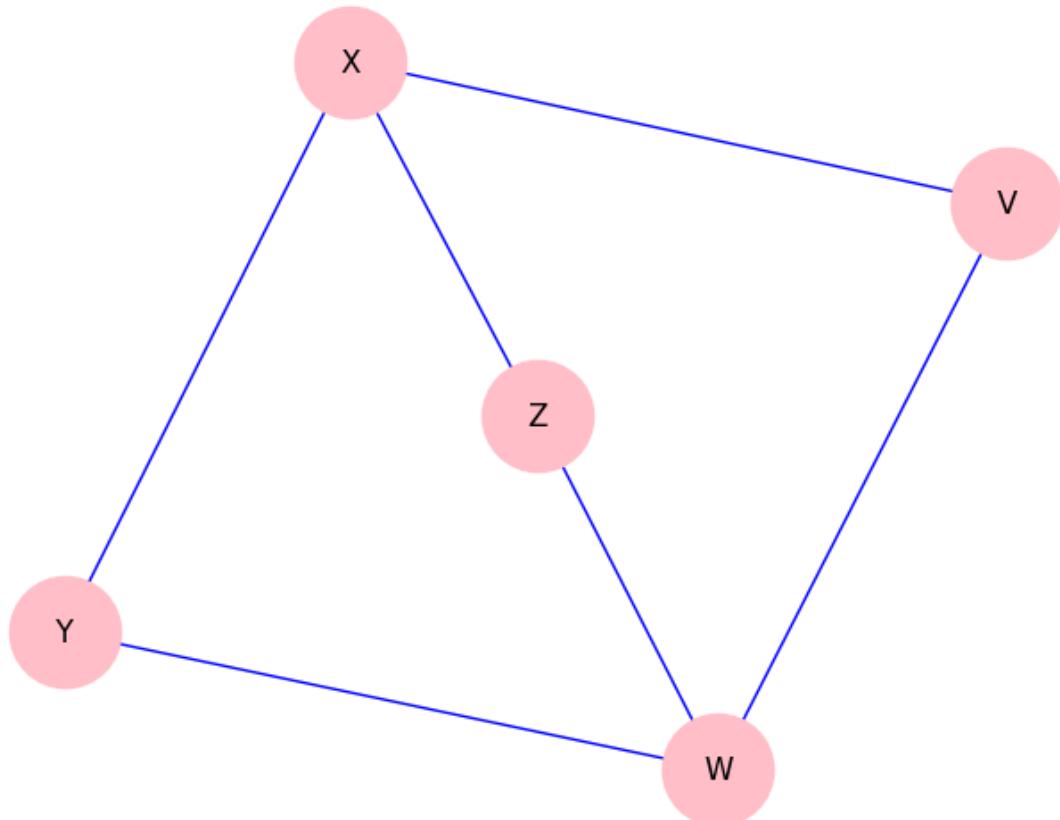
Isi bagian kosong di bawah ini:

```
In [14]: # Soal 3: Membuat graf baru
H = nx.Graph() # Buat graf baru

# Tambahkan 5 simpul
H.add_nodes_from(["X", "Y", "Z", "W", "V"])

# Tambahkan 6 sisi
H.add_edges_from([( "V", "W"), ( "V", "X"), ( "W", "Y"), ( "W", "Z"), ( "X", "Y"), ( "Y", "Z")])

# Tampilkan graf dengan warna berbeda dan sesuaikan dengan parameter yang
nx.draw(H, with_labels=True, node_color="pink", edge_color="blue", node_s
plt.show()
```



Contoh *output*:

