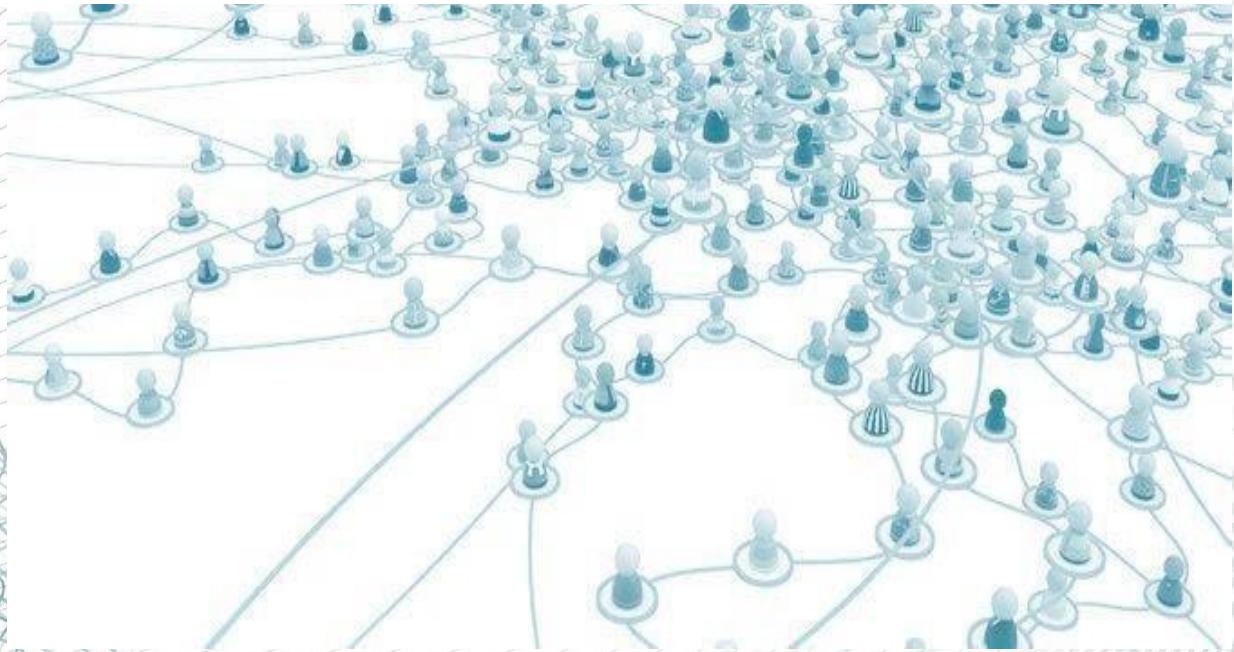


MODUL PRAKTIKUM

SD25-31001 DATA MINING



**Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera**

2025

MODUL 5

**Deteksi Komunitas dengan NetworkX
dan Algoritma Girvan-Newman**

1. PENDAHULUAN

Meskipun manusia sangat mahir dalam mendeteksi pola yang berbeda atau berulang di antara beberapa komponen, sifat jaringan besar yang saling terhubung membuat tugas dasar tersebut hampir mustahil dilakukan secara manual. Kelompok node yang terhubung secara padat mungkin mudah dilihat secara visual, tetapi metode yang lebih canggih diperlukan untuk melakukan tugas-tugas tersebut secara terprogram. Algoritma deteksi komunitas digunakan untuk menemukan kelompok node yang terhubung secara padat dalam berbagai jenis jaringan.

M. Girvan dan M. E. J. Newman telah mengusulkan salah satu algoritma deteksi komunitas yang paling banyak digunakan, yaitu algoritma **Girvan–Newman**. Menurut mereka, kelompok node dalam suatu jaringan terhubung erat di dalam komunitas, tetapi hanya terhubung lemah antar komunitas.

Dalam modul ini, Anda akan mempelajari prinsip dasar dibalik algoritma deteksi komunitas, implementasi spesifiknya, dan bagaimana menjalankannya menggunakan Python dan NetworkX.

2. CONTOH APLIKASI DETEKSI KOMUNITAS

Karena jaringan merupakan bagian integral dari banyak permasalahan dunia nyata, algoritma deteksi komunitas telah digunakan di berbagai bidang, mulai dari analisis jejaring sosial hingga inisiatif kesehatan masyarakat.

- Salah satu kasus penggunaan yang dikenal adalah pendekripsi kelompok teroris dalam jaringan sosial dengan menelusuri aktivitas dan interaksi mereka. Demikian pula, kelompok bot berbahaya dapat dideteksi pada platform sosial daring.
- Deteksi komunitas dapat digunakan untuk mempelajari dinamika kelompok tertentu yang rentan terhadap penyakit epidemik. Jenis penyakit lain dapat dipelajari dengan cara serupa untuk menemukan keterkaitan umum antar pasien.
- Salah satu kasus penggunaan terbaru adalah *community evolution prediction*, yaitu prediksi perubahan struktur jaringan.

3. TEKNIK DETEKSI KOMUNITAS

Ada dua jenis utama teknik deteksi komunitas: **agglomerative** dan **divisive**.

1) Agglomerative

Metode agglomerative umumnya dimulai dengan jaringan yang hanya berisi node dari graf asli. Edge kemudian ditambahkan satu per satu ke graf, di mana edge yang lebih kuat

diprioritaskan daripada yang lebih lemah. Kekuatan (atau bobot) sebuah edge dihitung berbeda tergantung implementasi algoritma.

2) **Divisive**

Di sisi lain, metode divisive bergantung pada proses menghapus edge dari graf asli secara iteratif. Edge yang lebih kuat dihapus sebelum edge yang lebih lemah. Pada setiap langkah, perhitungan bobot edge diulang, karena bobot edge yang tersisa berubah ketika sebuah edge dihapus. Setelah sejumlah langkah, graf terpecah menjadi kelompok node yang terhubung secara padat, yaitu komunitas.

4. NETWORKX

NetworkX adalah sebuah paket Python untuk membuat, memanipulasi, dan mempelajari struktur, dinamika, serta fungsi dari jaringan kompleks. Paket ini menyediakan:

1. Alat untuk mempelajari struktur dan dinamika jaringan sosial, biologis, dan infrastruktur;
2. Antarmuka pemrograman standar serta implementasi graf yang cocok untuk berbagai aplikasi;
3. Lingkungan pengembangan yang cepat untuk proyek kolaboratif dan multidisiplin;
4. Dukungan percepatan algoritma dan fitur tambahan melalui backend pihak ketiga;
5. Antarmuka ke algoritma numerik yang sudah ada serta kode yang ditulis dalam C, C++, dan FORTRAN; dan
6. Kemampuan untuk bekerja dengan mudah pada kumpulan data besar yang tidak standar.

Dengan NetworkX, Anda dapat memuat dan menyimpan jaringan dalam format data standar maupun tidak standar, menghasilkan berbagai jenis jaringan acak dan klasik, menganalisis struktur jaringan, membangun model jaringan, merancang algoritma jaringan baru, menggambar jaringan, dan masih banyak lagi.

5. DETEKSI KOMUNITAS DI NETWORKX

Beberapa algoritma deteksi komunitas yang tersedia di NetworkX antara lain:

- **Girvan–Newman algorithm**
Menghapus edge dengan betweenness tertinggi secara progresif.
- **Fluid Communities algorithm**
Berdasarkan interaksi fluida dalam suatu lingkungan yang saling mendorong.
- **Label Propagation algorithm**
Algoritma semi-supervised untuk memberi label pada data yang sebelumnya tidak berlabel.

- **Clique Percolation algorithm**

Menemukan komunitas k-clique menggunakan metode perkolasii.

- **Kernighan-Lin algorithm**

Mempartisi jaringan menjadi dua set dengan menukar pasangan node untuk mengurangi *edge cut*.

Algoritma Girvan–Newman

Untuk mendekripsi dan menganalisis struktur komunitas, algoritma Girvan–Newman melakukan eliminasi iteratif terhadap edge yang memiliki jumlah jalur terpendek terbanyak yang melewati edge tersebut. Dengan menghapus edge satu per satu, jaringan akan terpecah menjadi bagian-bagian kecil, yaitu komunitas. Algoritma ini diperkenalkan oleh Michelle Girvan dan Mark Newman.

Bagaimana cara kerjanya?

Intinya adalah mencari edge yang paling sering dilalui oleh jalur terpendek antar node, yaitu dengan menghitung **edge betweenness centrality**. Edge yang menghubungkan komunitas biasanya memiliki nilai betweenness tinggi. Ketika edge-edge tersebut dihapus, struktur komunitas menjadi lebih jelas.

Empat langkah utama algoritma:

1. Hitung betweenness centrality untuk setiap edge di graf.
2. Hapus edge dengan betweenness tertinggi.
3. Hitung ulang betweenness untuk edge yang tersisa.
4. Ulangi langkah 2–3 hingga tidak ada edge yang tersisa.

Contoh menunjukkan bahwa edge antara node C dan D dihapus terlebih dahulu karena memiliki betweenness tertinggi; edge tersebut kemungkinan merupakan penghubung antar komunitas.

6. BETWEENNESS CENTRALITY

Betweenness centrality mengukur sejauh mana node atau edge berada pada jalur antara node-node lain. Node atau edge dengan betweenness tinggi dapat memiliki pengaruh besar karena mengontrol aliran informasi pada jaringan.

Perhitungan betweenness tidak seragam dan dapat dihitung melalui beberapa metode. Umumnya didefinisikan sebagai jumlah jalur terpendek yang melewati node/edge dibagi total jalur terpendek.

REPEAT

LET n BE number of edges in the graph

```

FOR i=0 to n-1
    LET B[i] BE betweenness centrality of edge i
    IF B[i] > max_B THEN
        max_B = B[i]
        max_B_edge = i
    ENDIF
ENDFOR
REMOVE edge i FROM graph
UNTIL number of edges in graph is 0

```

7. ALGORITMA DETEKSI KOMUNITAS DI NETWORKX

girvan_newman(G, most_valuable_edge=None)

Input

- G: graf NetworkX
- most_valuable_edge: fungsi opsi untuk memilih edge yang akan dihapus. Jika tidak diberikan, NetworkX memilih edge dengan betweenness tertinggi.

Output

Iterator berisi tuple kumpulan node (komunitas) pada setiap level iterasi.

Contoh

```

import matplotlib.pyplot as plt
import networkx as nx
from      networkx.algorithms.community import girvan_newman

G = nx.karate_club_graph()
communities = girvan_newman(G)

node_groups = []
for com in next(communities):
    node_groups.append(list(com))

print(node_groups)

color_map = []
for node in G:
    if node in node_groups[0]:
        color_map.append('blue')
    else:
        color_map.append('green')
nx.draw(G, node_color=color_map, with_labels=True)
plt.show()

```

Output contoh:

```
[[0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 19, 21],
```

```
[2, 8, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,  
32, 33]]
```

Graf terbagi menjadi dua komunitas yang sangat berbeda.

8. IMPLEMENTASI LAINNYA

```
import networkx as nx  
from networkx.algorithms.community.centrality import  
girvan_newman  
from networkx.algorithms.community import modularity  
import matplotlib.pyplot as plt  
  
# 1. Memuat Graph  
G = nx.karate_club_graph()  
  
# 2. Menjalankan Girvan-Newman  
communities_generator = girvan_newman(G)  
  
all_partitions = []  
all_modularity = []  
  
# Ambil semua level komunitas  
for communities in communities_generator:  
    communities = [list(c) for c in communities]  
    all_partitions.append(communities)  
    Q = modularity(G, communities)  
    all_modularity.append(Q)  
    print(f"Level {len(all_partitions)}: {len(communities)} komunitas, modularity={Q:.4f}")  
  
# 3. Menentukan komunitas terbaik  
best_index = all_modularity.index(max(all_modularity))  
best_communities = all_partitions[best_index]  
  
print("\n Komunitas terbaik berdasarkan modularity:")  
print(best_communities)  
  
# 4. Visualisasi komunitas terbaik  
colors = ["red", "blue", "green", "orange", "purple", "cyan"]  
  
color_map = {}  
for i, comm in enumerate(best_communities):  
    for node in comm:  
        color_map[node] = colors[i % len(colors)]  
  
node_colors = [color_map[n] for n in G.nodes()]  
  
plt.figure(figsize=(8, 6))  
nx.draw(G, node_color=node_colors, with_labels=True,  
node_size=500)  
plt.title("Deteksi Komunitas dengan Girvan-Newman")
```

```
plt.show()
```

Output:

Deteksi Komunitas dengan Girvan-Newman

