

# Friend Recommendation System Using Social Network Graph and Community Detection

Nadhif Al Rozin 13523076

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[13523076@std.stei.itb.ac.id](mailto:13523076@std.stei.itb.ac.id), [2nadhifalrozin@gmail.com](mailto:2nadhifalrozin@gmail.com)

**Abstract**—Sistem rekomendasi teman yang dibuat menggunakan metode Friend Of Friend dengan membaca data jejaring sosial dalam bentuk Graf, Sedangkan Deteksi komunitas dibuat dengan Algoritma Greedy Modularity yang berfokus pada menghasilkan nilai Modularity setinggi mungkin dengan membagi bagi graf jejaring sosial menjadi sub graf (komunitas)

**Keywords**—Graf, Rekomendasi, Friend Of Friend, Greedy Modularity

## I. PENDAHULUAN

Dengan semakin cepatnya teknologi berkembang, jejaring sosial di masyarakat mulai berubah mengikuti jaman menjadi jejaring sosial di dunia maya, di social media berupa facebook, Instagram, X dan yang lainnya. Hal ini menyebabkan munculnya suatu kebutuhan untuk mendesai sosial media yang dapat dengan akurat menentukan “kepentingan” user lain dengan dirinya berdasarkan hubungan saling mengikuti atau saling mengenal orang yang sama, atau mengkonsumsi konten yang sama (komunitas). Hal ini menunjukkan jika sistem rekomendasi teman yang baik dan akurat dibutuhkan.

Rekomendasi teman dalam jaringan sosial, sangatlah penting karena ini sebagai koneksi personal seseorang. Hal ini membantu pengguna untuk memperluas jaringan sosialnya sendiri berdasarkan minat atau ketertarikan yang sama atau memiliki koneksi yang serupa. Dengan sistem seperti ini, ini memudahkan pengguna untuk mencari teman baru. Selain dari segi pengguna, pemilik platform juga diuntungkan dengan semakin banyak pengguna yang terhubung satu sama lain, karena hal ini mendorong interaksi antar pengguna di dalam aplikasi menjadi lebih tinggi. Tidak hanya itu, tingkat retensi pengguna juga dapat ditingkatkan dengan memberikan personalisasi layanan jika diketahui komunitas / minat yang digemari untuk dikonsumsi oleh user, rekomendasi yang sesuai ini yang akan membuat pengguna tetap pada aplikasi dengan waktu yang lebih lama

Jaringan sosial sebagai graf adalah konsep yang mengibaratkan entitas entitas yang saling berhubungan, simpul pada graf atau disebut node adalah representasi dari entitas tunggal, entitas ini dapat terhubung dengan entitas lain dan membentuk koneksi atau hubungan, hal ini direpresentasikan dalam bentuk sisi atau edge. Hal ini diperluas lagi menjadi banyak hal seperti bobot pada sisi, graf yang dinamis dan keterhubungan. Yang akan digunakan lebih lanjut untuk membuat sistem rekomendasi ini.

Hal ini memiliki tantangan yang cukup besar dalam

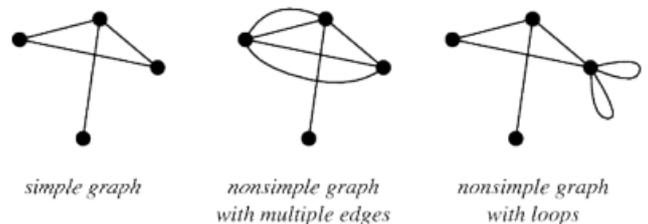
pengembangannya, terutama dalam volume data yang sangat besar, karena ini menangani jutaan pengguna yang terhubung pada jaringan. Hal ini mengarah ke masalah berikutnya yaitu keakuratan dari rekomendasi. Ini harus diatur dengan membuat algoritma yang dapat menyeimbangkan antara akurasi dan efisiensi. Dikarenakan data yang dinamis dan terus berubah, dibutuhkan algoritma yang dapat dengan cepat memperbaharui kondisi pengguna.

## II. LANDASAN TEORI

### A. Teori Graf

Graf terdiri atas 2 komponen utama, simpul (nodes) dan sisi (edges). Simpul adalah titik yang merepresentasikan suatu objek dan sisi adalah garis yang menghubungkan simpul ke simpul, sisi bisa diartikan untuk menyatakan hubungan antar simpul. Pada konteks ini simpul adalah pengguna dan sisi adalah ikatan pertemanan.

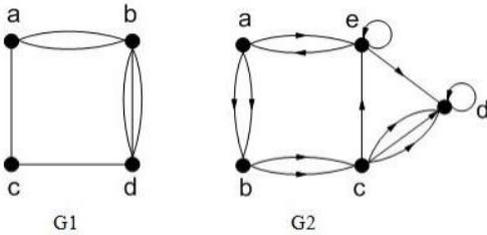
Graf dibagi menjadi 2 jenis, berdasarkan ada tidaknya gelang atau sisi ganda di dalam graf. Ini digolongkan menjadi 2 jenis, Graf sederhana dan tak-sederhana. Graf sederhana adalah graf yang tidak mengandung gelang ataupun sisi ganda, sedangkan Graf tak-sederhana adalah Graf yang memiliki sisi ganda atau gelang.



Gambar 1 Contoh gambar Graf sederhana dan tak-sederhana  
Sumber : Wolfram

Pada konteks ini hanya akan digunakan graf sederhana, artinya tidak ada gelang dan tidak ada sisi ganda. Karena dalam jejaring sosial yang digunakan di implementasi tidak ada hubungan dengan diri sendiri dan 2 hubungan berbeda antar 2 orang yang sama.

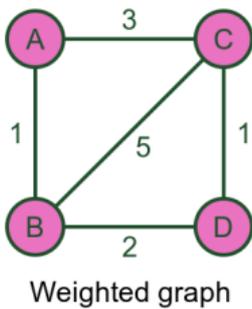
Selain itu Graf juga ada yang berarah, Graf berarah adalah Graf yang sisi sisinya memiliki orientasi arah. Ditunjukkan dengan tanda panah pada gambar dibawah.



G1 : graf tak-berarah; G2 : Graf berarah

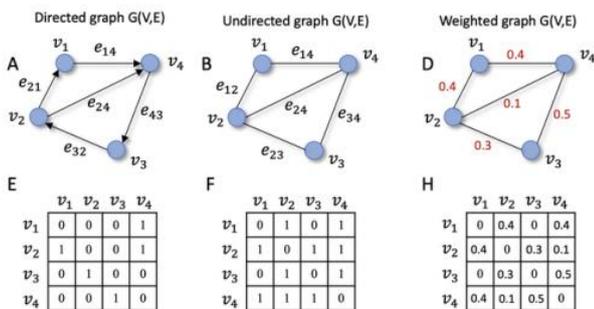
Gambar 2 Contoh gambar graf berarah dan tak-berarah  
Sumber : ppt pak Rinaldi Munir Graf 1

Graf berarah bisa digunakan untuk mengklasifikasi lebih lanjut hubungan yang ada berdasarkan arahnya (masuk dan keluar). Sisi dari Graf juga bisa memiliki berat untuk mengkuantisasi hubungan dari 2 pengguna tersebut, hal ini dinamakan Graf berbobot (Weighted Graph)



Gambar 3 Contoh Graf berbobot  
Sumber : <https://graphicmaths.com/computer-science/graph-theory/adjacency-matrices/>

Matriks Adjacency adalah representasi matriks dari suatu graf di mana elemen-elemen yang ada pada matriks menunjukkan ada atau tidaknya (0/1) hubungan (sisi) antara dua buah simpul, ini digunakan untuk merepresentasikan hubungan antara 1 pengguna dengan pengguna lain. Nilainya bisa berubah jika termasuk dalam weighted graf (bisa lebih dari 1).



Gambar 4 Contoh matriks adjacency untuk tiap jenis graf  
Sumber : ppt pak Rinaldi Munir Graf 2

### B. Deteksi Komunitas

Deteksi komunitas adalah proses untuk mengidentifikasi dan membagi bagi sekelompok pengguna menjadi komunitas komunitas yang ada dalam graf. Dikelompokkan berdasarkan keterhubungan yang lebih erat antar simpul simpul dalam suatu komunitas dengan komunitas lain. Hal ini digunakan dalam jejaring sosial adalah untuk membantu dalam memahami bagaimana grup atau kelompok yang dibentuk oleh individu. Selain itu hal ini juga dapat mengungkapkan dan menganalisis struktur komunitas pada jejaring sosial yang tersembunyi, hal ini digunakan juga untuk melihat kecenderungan konten yang akan diberikan oleh aplikasi media sosial pada pengguna.

Modularitas digunakan dalam deteksi komunitas, metode ini digunakan untuk mengukur kualitas dari pembagian jaringan yang ada menjadi komunitas komunitas. Apabila modularitas tinggi hal ini menunjukkan jika komunitas yang sudah dibagi adalah komunitas yang jelas dan padat. Jika nilai modularitas positif artinya jumlah komunitas yang dibentuk lebih banyak dari yang diharapkan. Teknik ini digunakan untuk menilai kualitas pembagian graf menjadi komunitas. Algoritma deteksi komunitas berfokus pada memaksimalkan nilai modularitas sehingga pembagian komunitas menjadi lebih baik. Modularitas ini didapat dengan membandingkan kepadatan sisi di dalam komunitas dengan kepadatan sisi yang diharapkan jika graf dalam keadaan acak. Nilai ini berkisar antara -1 hingga 1

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Gambar 5 Modularity Formula  
Sumber : [8]

$A_{ij}$  adalah matrix adjacency,  $k_i$  adalah derajat dari node  $i$ . sedangkan  $m$  adalah jumlah total sisi dalam graf. Dan  $\delta(c_i, c_j)$  adalah 1 jika  $i$  dan  $j$  ada pada komunitas yang sama, jika tidak hasilnya 0.

Untuk melakukan deteksi komunitas ada beberapa algoritma yang dapat digunakan, salah satunya adalah Algoritma Louvain, Louvain adalah algoritma yang sangat populer digunakan untuk deteksi komunitas karena efektif dalam meningkatkan Tingkat modularitas. Algoritma ini bekerja secara iterative, ini dimulai dengan menganggap setiap simpul sebagai komunitas individual terlebih dahulu, lalu saling menggabungkan komunitas yang paling dekat, jika penggabungan itu meningkatkan modularitasnya. Akan diulang terus sampai tidak ada peningkatan lagi.

Selain itu ada juga algoritma spectral clustering, algoritma ini menggunakan eigenvector dari matriks adjacency atau Laplacian untuk memisahkan graf yang ada menjadi komunitas komunitas. Yang dilakukan adalah dengan mengurangi dimensi data dari graf menjadi bentuk bentuk yang bisa dipisahkan dengan lebih mudah. Hal ini dilakukan dengan metode clustering lain seperti k-means. Dibandingkan algoritma lain, algoritma ini cocok digunakan untuk mengidentifikasi struktur komunitas secara matematis dan dengan struktur spektral yang kuat dan jelas.

Algoritma lain adalah Greedy Modularity, Algoritma ini menggunakan pendekatan secara heuristic, untuk

memaksimalkan nilai modularitas yang dibentuk oleh pembagian komunitas. Algoritma ini dimulai dengan membuat setiap simpul sebagai suatu komunitas independent, sama halnya dengan Louvarian, namun yang membedakan adalah komunitas yang digabungkan dipilih dari yang memberikan peningkatan terbesar dalam modularitas. Keuntungan dari metode ini adalah algoritma ini sangat mudah untuk di implementasikan dan memiliki efisiensi yang tinggi untuk mendapat nilai modularitas tertinggi.

Selain itu Algoritma lain seperti Edge Betweenness juga dapat digunakan, algoritma ini menghapus sisi dengan nilai betweenness yang mempunyai nilai paling tinggi secara iteratif. Nilai between ini adalah nilai dari sisi yang paling banyak dilalui oleh jalur terpendek. Setelah penghapusan ini, struktur komunitas akan mulai muncul karena sisi yang menghubungkan antar komunitas adalah sisi yang punya nilai betweenness tinggi. Algoritma ini cocok untuk ukuran graf yang kecil dan menengah.

Algoritma Hierarchical Clustering juga bisa digunakan untuk deteksi komunitas, algoritma ini membentuk hirarki dari komunitas dimulai dari simpul individu sampai seluruh graf menjadi satu komunitas besar, Algoritma ini bersifat agglomerative atau divisive. Menggabungkan dan membagi komunitas yang ada. Keunggulannya adalah ini memberikan pandangan secara menyeluruh tentang struktur komunitas dari berbagai Tingkat granularitas, ini cocok jika data yang dipakai memiliki struktur hubungan hirarkis yang alami.

### C. Rekomendasi

Rekomendasi Teman didasari dengan bagaimana cara sistem menilai / memprediksi koneksi baru yang mungkin akan terjadi berdasarkan pola hubungan suatu user dengan user lain. Hal ini memanfaatkan data hubungan banyak hal seperti teman, minat, aktivitas, tempat dan lain lain. Pada makalah ini akan difokuskan ke hubungan teman.

Pendekatan algoritma yang digunakan memiliki beberapa cara, salah satunya adalah Friends of Friends. Pendekatan ini dilakukan dengan merekomendasikan hubungan baru berdasarkan Tingkat dua dalam garf. Jika ada dua individu yang memiliki teman Bersama, mereka memiliki kemungkinan yang tinggi untuk saling terhubung. Kelebihan dari cara ini adalah sederhana dan mudah di implementasikan, namun kekurangannya adalah car aini tidak memperhitungkan bobot hubungan atau faktor tambahan lainnya.

Cara lain adalah dengan menggunakan Jaccard Coefficient, Konsep ini mengukur kesamaan antara 2 simpul dengan membandingkan ukuran irisan dan gabungan dari tetangga simpul tersebut. Rumus yang digunakan adalah:

$$J(A, B) = \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|}$$

Di mana  $N(A)$  dan  $N(B)$  adalah tetangga tetangga dari node A dan B. Sama dengan FOF, kekurangannya ada pada bobot hubungan yang tidak dipertimbangkan, Cara ini lebih cocok untuk mencari hubungan baru jika banyak hubungan yang tumpang tindih.

Cara lain adalah Preferential Attachment. Konsep ini didasarkan pada teori bahwa simpul yang populer cenderung

akan semakin populer. Artinya, simpul dengan derajat yang tinggi memiliki kemungkinan lebih besar untuk menarik hubungan baru. Keuntungan dari metode ini adalah ini sederhana dan mudah diterapkan di graf dengan ukuran besar, dan cara ini juga mencerminkan pola yang biasa terjadi di dunia nyata, kekurangannya adalah bisa memperburuk ketimpangan dalam jaringan, dan sama seperti metode lainnya, ini tidak mempertimbangkan kualitas dari hubungan atau konteks hubungan itu.

Cara lainnya adalah Resource Allocation, metode ini melakukan rekomendasi dengan dasar konsep distribusi sumber daya melalui sisi sisi yang ada dalam graf. Skor dari dua simpul dihitung berdasarkan kontribusi tetangga bersama yang mereka miliki. Dapat dilihat seperti berikut:

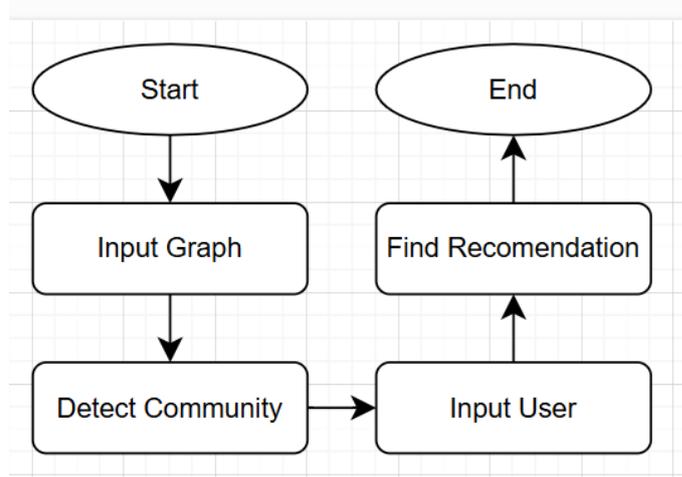
$$s_{ij} = \sum_{k \in N(i) \cap N(j)} \frac{1}{\text{deg}(k)}$$

I dan J adalah 2 node yang akan dihitung,  $s_{ij}$  adalah skor dari hubungan node I dan J.  $\text{deg}(k)$  adalah derajat dari node k, Dimana k adalah tetangga I yang merupakan tetangga J. Keuntungan dari metode ini adalah cara ini memperhitungkan kontribusi dari setiap tetangga bersama, namun ini hanya efektif di graf skala kecil hingga menengah. Ini tidak efisien untuk graf yang sangat besar, dan cara ini juga sangat rentan error jika dengan node dengan derajat yang sangat tinggi

Hybrids Model adalah kombinasi dari beberapa pendekatan, seperti dari algoritma graf lain seperti Jaccard dan Resource Allocation dengan penggunaan AI untuk meningkatkan akurasi prediksi, metode ini digunakan oleh Netflix untuk menggabungkan algoritma berbasis hubungan dengan data preferensi pengguna untuk rekomendasi film. Namun kekurangannya adalah ini jauh lebih kompleks dan membutuhkan daya komputasi yang besar, dan sulit di implementasikan tanpa data yang cukup dan berkualitas.

### III. IMPLEMENTASI

Implementasi Algoritma PageRank dapat dilihat sebagai berikut:



Gambar 6 Flowchart Main

```
def detect_communities(graph):
    """Mendeteksi komunitas dalam graf menggunakan algoritma modularitas."""
    communities = list(greedy_modularity_communities(graph))
    community_mapping = {node: f"Community {i + 1}" for i, community in enumerate(communities) for node in community}
    return community_mapping

def recommendation(graph, user):
    """Merekendasikan teman berdasarkan koneksi teman dari teman."""
    if user not in graph:
        return f"User {user} tidak ditemukan dalam jaringan sosial."
    # teman dari user
    friends = set(graph.neighbors(user))
    # teman dari teman
    friends_of_friends = {fof for friend in friends for fof in graph.neighbors(friend)} - friends - {user}
    return list(friends_of_friends)
```

Gambar 7 Source Code Implementasi

Algoritma yang digunakan dalam implementasi ini adalah Algoritma Greedy Modularity, Algoritma ini dipilih digunakan karena mudah untuk di implementasikan dan berfungsi dengan baik di data kecil sampai besar. Algoritma ini di import dari `network.algorithms.community` sehingga bisa langsung digunakan dalam kode. Untuk algoritma rekomendasi menggunakan metode friend of friends, awalnya dicek terlebih dahulu tetangga node dari user yang dimasukkan, setelah itu mencari Friend of Friend (tetangga dari teman teman user kecuali user dan teman user saat ini). Dengan 2 Algoritma ini, sistem dapat menunjukkan dengan baik komunitas yang ada pada graf, dan siapa saja orang didalamnya, dan juga memberikan rekomendasi teman kepada seorang user berdasarkan hubungan teman dari teman.

```
# Visualisasi
try:
    import matplotlib.pyplot as plt
    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, node_color="lightblue", edge_color="gray", node_size=2000, font_size=10)
    plt.show()
except ImportError:
    print("Matplotlib tidak diinstal. Lewati visualisasi graf.")
```

Gambar 8 Visualisasi Graf

Ditambahkan juga algoritma untuk visualisasi graf, visualisasi menggunakan matplotlib dengan bentuk `spring_layout`.

Dapat dilihat lebih lanjut disini:

<https://github.com/Narr21/Friend-Recomendation-Matdis>

#### IV. HASIL DAN PEMBAHASAN

Data yang akan dimasukkan adalah user user yang ada pada graf dalam bentuk representasi sebagai nodes dan hubungan teman yang ada sebagai edges. Yang akan dilakukan adalah, kita akan membandingkan graf yang sudah degenerate lewat matplotlib dengan hasil yang didapat oleh deteksi komunitas dan rekomendasi teman, rekomendasi akan diminta dari salah satu user

```
# node (pengguna)
G.add_nodes_from(["Alice", "Bob", "Charlie", "David", "Eve", "Frank", "Grace", "Hannah"])

# edge (hubungan antara pengguna)
G.add_edges_from([
    ("Alice", "Bob"), ("Alice", "Charlie"), ("Bob", "David"),
    ("Charlie", "David"), ("Eve", "Frank"), ("Frank", "Grace"),
    ("Grace", "Hannah"), ("Alice", "Eve")
])
```

Gambar 9 Data 1

Dapat dilihat jika data 1 adalah data yang berisi 8 user dan 8 edge, ini dibuat sebagai perbandingan awal untuk graf dengan skala yang sangat kecil

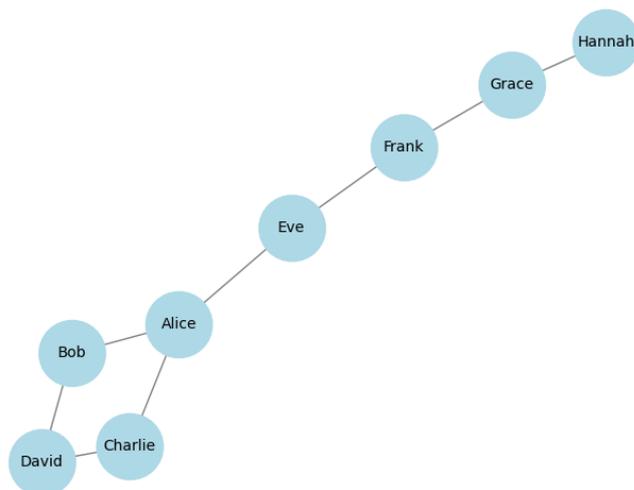
```
Komunitas yang terdeteksi:
Charlie -> Community 1
David -> Community 1
Bob -> Community 1
Alice -> Community 1
Frank -> Community 2
Grace -> Community 2
Eve -> Community 2
Hannah -> Community 2

Masukkan nama pengguna untuk merekomendasikan teman: Charlie

Rekomendasi teman untuk Charlie: ['Eve', 'Bob']
```

Gambar 10 Hasil Data 1

Dari hasil, dapat dilihat jika Charlie, David, Bob dan Alice berada pada komunitas yang sama yaitu komunitas 1 dan Frank, Grace, Eve dan Hannah berada pada komunitas yang berbeda yaitu komunitas 2. Saat memasukkan Charlie sebagai user, sistem memberikan rekomendasi teman Eve dan Bob yang mana adalah teman dari teman Charlie. Contohnya Alice adalah teman dari Charlie dan Alice memiliki teman Bob, Eve dan Charlie, sehingga yang direkomendasikan adalah Bob dan Eve karena mereka berdua tidak berteman dengan Charlie dan bukan Charlie itu sendiri. Hal yang sama juga berlaku pada teman charlie yang lain David, David punya teman Bob dan Charlie, sehingga ia merekomendasikan Bob sebagai teman. Meski bob direkomendasikan 2 kali, hanya dihitung 1 kali saja. Hal ini dapat dipastikan dengan melihat hasil graf dibawah.



Gambar 11 Graf data 1

Dapat dilihat jika komunitas 1 dan 2 direpresentasikan dengan baik, hal ini membagi graf menjadi 2 sub graf, subgraf pertama adalah Alice, Charlie, Bob dan David, sub graf ke 2 adalah Eve, Frank, Grace dan Hannah. Dengan Hubungan Alice dan Eve yang menjadi sisi yang menghubungkan 2 komunitas ini. Dapat dilihat juga hubungan antara Charlie, Alice dan David yang merekomendasikan Bob dan Eve kepada Charlie sebagai teman.

```
# Menambahkan 30 nodes
nodes = [
    "Node1", "Node2", "Node3", "Node4", "Node5", "Node6", "Node7", "Node8", "Node9", "Node10",
    "Node11", "Node12", "Node13", "Node14", "Node15", "Node16", "Node17", "Node18", "Node19", "Node20",
    "Node21", "Node22", "Node23", "Node24", "Node25", "Node26", "Node27", "Node28", "Node29", "Node30"
]
G.add_nodes_from(nodes)

# Menambahkan edges (hubungan antar pengguna)
edges = [
    ("Node1", "Node2"), ("Node1", "Node3"), ("Node2", "Node4"), ("Node3", "Node4"),
    ("Node5", "Node6"), ("Node5", "Node7"), ("Node6", "Node8"), ("Node7", "Node8"), ("Node9", "Node10"),
    ("Node9", "Node11"), ("Node10", "Node11"), ("Node11", "Node12"), ("Node12", "Node13"),
    ("Node13", "Node14"), ("Node14", "Node15"), ("Node15", "Node16"), ("Node16", "Node17"),
    ("Node17", "Node18"), ("Node18", "Node19"), ("Node19", "Node20"), ("Node20", "Node21"),
    ("Node21", "Node22"), ("Node22", "Node23"), ("Node23", "Node24"), ("Node24", "Node25"),
    ("Node25", "Node26"), ("Node26", "Node27"), ("Node27", "Node28"), ("Node28", "Node29"),
    ("Node29", "Node30"), ("Node1", "Node30")
]
G.add_edges_from(edges)
```

Gambar 12 Data 2

Data 2, sama halnya dengan data 1, data 2 terdiri dari simpul (nodes) yang merepresentasikan user dan sisi (edges) yang merepresentasikan hubungan, jumlah nodes ditingkatkan menjadi 30 simpul dan 30 sisi.

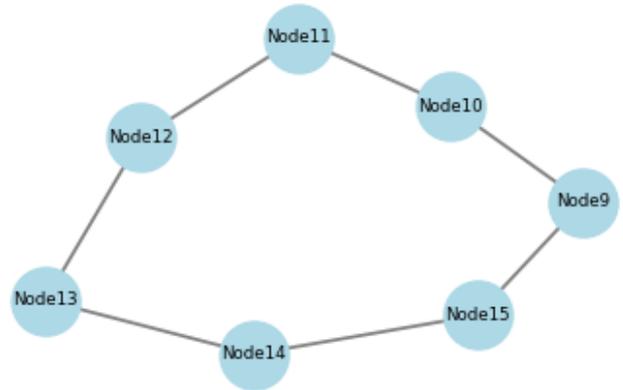
```
Komunitas yang terdeteksi:
Node16 -> Community 1
Node21 -> Community 1
Node20 -> Community 1
Node22 -> Community 1
Node18 -> Community 1
Node19 -> Community 1
Node17 -> Community 1
Node23 -> Community 1
Node12 -> Community 2
Node13 -> Community 2
Node11 -> Community 2
Node15 -> Community 2
Node10 -> Community 2
Node14 -> Community 2
Node9 -> Community 2
Node24 -> Community 3
Node29 -> Community 3
Node30 -> Community 3
Node28 -> Community 3
Node25 -> Community 3
Node26 -> Community 3
Node27 -> Community 3
Node2 -> Community 4
Node3 -> Community 4
Node1 -> Community 4
Node4 -> Community 4
Node6 -> Community 5
Node5 -> Community 5
Node8 -> Community 5
Node7 -> Community 5

Masukkan nama pengguna untuk merekomendasikan teman: Node3
Rekomendasi teman untuk Node3: ['Node5', 'Node2', 'Node30']
```

Gambar 13 Hasil Data 2

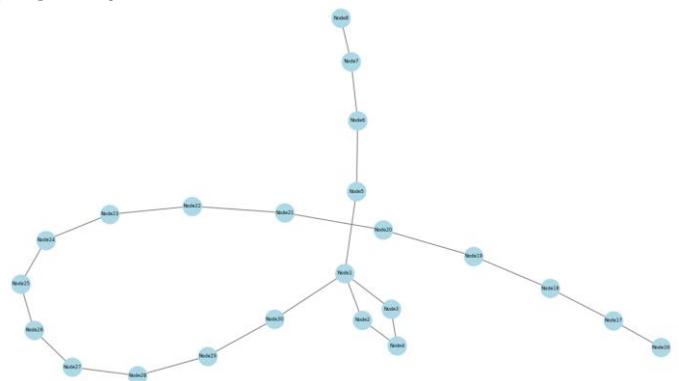
Dari hasil, dapat dilihat jika Node 16 – 23 ada pada komunitas yang sama yaitu komunitas 1 Sama halnya dengan Node 9 – 15 yang ada pada komunitas 2, Node 24 – 30 yang ada pada komunitas 3, Node 1 – 4 yang ada pada komunitas 4 dan Node 5 – 8 yang ada pada komunitas 5. Saat memasukkan Node 3 sebagai user, sistem memberikan rekomendasi teman Node 5, Node 2 dan Node 30 yang mana adalah teman dari teman Node 3. Contohnya Node 1 adalah teman dari Node 3 dan Node 1 memiliki teman Node 3, Node 5, Node 2 dan Node 30, sehingga yang direkomendasikan adalah Node 5, Node 2 dan Node 30 karena mereka tidak berteman dengan Node 3 dan bukan Node

3 itu sendiri. Hal yang sama juga berlaku pada teman Node 3 yang lain Node 4, Node 4 punya teman Node 3 dan Node 2, sehingga ia merekomendasikan Node 2 sebagai teman. Meskipun Node 2 direkomendasikan sebanyak 2 kali, sama halnya dengan kondisi pada data 1 hal ini hanya dihitung 1 kali saja. Hal ini dapat dipastikan dengan melihat hasil graf data 2 dibawah.



Gambar 14 Graf data 2 bagian 1

Bagian Subgraf ini terpisah dari Node lainnya karena tidak memiliki koneksi dengan simpul lain, Dapat dilihat jika node yang ada disini adalah node 9 – 15 yang merupakan satu komunitas yaitu komunitas 2, hal ini sudah sesuai dengan apa yang ditunjukkan oleh sistem



Gambar 15 Graf data 2 bagian 2

Bagian graf ini adalah Bagian subgraf lain yang berisi 4 komunitas berbeda, Node 1 yang merupakan derajat paling tinggi pada graf ini adalah yang menghubungkan 3 komunitas sekaligus, komunitas 4 yang berisi Node 1 – 4 ada dekat dengan Node 1 di kedua sisi nya. Sisi lain yang menjulur ke atas adalah komunitas 5 yang berisi Node 5 – 8. Untuk sisi yang menjulur ke bawah adalah rantai panjang yang terdiri dari 2 komunitas yaitu Node 24 – 30 yang merupakan komunitas 3 dan komunitas 1 yang berisi node 16 – 23 dimulai dari ujung kanan bawah node dan menjalar sampai pertengahan. Dapat dilihat juga hubungan antara Node 3, Node 1 dan Node 4 yang merekomendasikan Node 2, Node 5 dan Node 30 kepada Node 3 sebagai teman.

## V. KESIMPULAN

Dari hasil, dapat diperhatikan jika graf dapat dibentuk dengan akurat baik dari data 1 dan data 2. Meskipun data yang lebih besar (10000 nodes) yang merupakan ukuran medium dalam jejaring sosial pada aplikasi saat ini masih belum bisa di implementasikan karena keterbatasan data. Tapi pembagian komunitas bisa lebih terlihat di data dengan 30 nodes, komunitas juga dibagi dengan baik, dan sesuai seperti representasi pada graf, jumlah tiap komunitas juga tidak berbeda jauh.

Untuk Rekomendasi teman dapat dilakukan dengan akurat, hal ini bisa dikembangkan lebih jauh lagi jika dibutuhkan, menggunakan parameter lain seperti bobot hubungan, minat, kecenderungan komunitas dll. Pada makalah ini bisa dibuktikan, jika Penggunaan algoritma Friend Of Friend untuk membuat rekomendasi teman pada jejaring sosial yang dirubah menjadi representasi graf, berfungsi dengan efektif. Algoritma Greedy Modularity juga yang digunakan untuk Mendeteksi komunitas yang ada pada Graf juga dapat dengan efektif dan akurat memisahkan graf menjadi sub graf sub graf dengan modularity tertinggi.

## VI. UCAPAN TERIMAKASIH

Pertama-tama, penulis memanjatkan puji dan Syukur kepada Tuhan YME, atas berkat dan nikmatnya diberikan kelancaran untuk menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada kedua orang tua, serta teman teman yang sudah mendukung selama pembuatan makalah ini.

Kemudian penulis juga memberikan rasa terimakasih khusus kepada Bapak Rila Mandala selaku dosen pengajar mata kuliah IF1220 Matematika Diskrit kelas K2, yang telah memberikan penulis ilmu dan kesempatan untuk mengerjakan makalah ini.

## REFERENSI

- [1] Munir, Rinaldi. 2024. Graf (Bagian 1), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>, diakses pada 5 Januari 2025.
- [2] Munir, Rinaldi. 2024. Graf (Bagian 2), <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>, diakses pada 5 Januari 2025.
- [3] D. Liu and L. Lü, "Link prediction in complex networks: A survey," *Physics Reports*, vol. 531, no. 2, pp. 1-57, 2010.
- [4] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167-256, 2003.
- [5] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75-174, 2010.
- [6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, Jun. 2005.
- [7] C. C. Aggarwal, *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2016.
- [8] T. Petric et al., "The modularity formula identifies cohesive communities in a network by measuring the density of edges within communities compared to between them," *ResearchGate*. [Online]. Available: [https://www.researchgate.net/figure/The-modularity-formula-identifies-cohesive-communities-in-a-network-by-measuring-the\\_fig2\\_374303714](https://www.researchgate.net/figure/The-modularity-formula-identifies-cohesive-communities-in-a-network-by-measuring-the_fig2_374303714). [Accessed: Jan. 05, 2025].
- [9] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Oct. 2008. doi: 10.1088/1742-5468/2008/10/P10008.
- [10] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395-416, Dec. 2007. doi: 10.1007/s11222-007-9033-z.

- [11] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Sep. 2006. doi: 10.1103/PhysRevE.74.036104.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Januari 2025



Nadhif Al Rozin 13523076