

Membangun Sistem Keputusan Otomatis untuk Rekomendasi Film Menggunakan Relasi, Rekursi, dan Graf

Muhammad Timur Kanigara - 13523055

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

timurkanigara21@gmail.com, 13523055@std.stei.itb.ac.id

Abstract—Recommendation systems have become an essential part of modern applications, such as movie streaming services. These systems can improve user experience by recommending relevant content based on user preferences. In this paper, the authors build an automated decision system for movie recommendations that is processed into a weighted graph that represents the relationship between user preferences and the movies they watch. A recursive algorithm is also used to recommend movies to users based on the similarity of preferences between users. The simulation results obtained show that this approach is effective in providing movie recommendations based on simple data.

Keywords—recommendation system, graph theory, relation, recursion.

I. PENDAHULUAN

Sistem rekomendasi adalah hal penting dalam suatu layanan digital, seperti platform streaming film, musik, dan *e-commerce*, dan sebagainya. Dengan meningkatnya jumlah konten yang tersedia, pengguna sering kali merasa kewalahan dalam memilih apa yang ingin mereka tonton atau beli. Oleh karena itu, sistem rekomendasi berfungsi untuk menyaring informasi dan memberikan saran yang relevan berdasarkan preferensi dan perilaku pengguna. Dalam konteks film, sistem rekomendasi bisa meningkatkan pengalaman pengguna dan membantu suatu platform dalam meningkatkan retensi pengguna dan kepuasannya.

Sistem rekomendasi bekerja dengan cara mempelajari pola perilaku pengguna, seperti film yang telah ditonton, rating yang telah diberikan, dan interaksi dengan konten lainnya. Dengan memanfaatkan data-data ini, sistem dapat mengidentifikasi kesamaan antara pengguna dan merekomendasikan film yang mungkin disukai. Dalam penelitian ini, penulis berfokus pada pengembangan sistem rekomendasi otomatis yang menggunakan kombinasi konsep relasi, rekursi, dan teori graf. Pendekatan tersebut memungkinkan penulis untuk membangun model sederhana namun efektif dalam memberikan suatu rekomendasi film yang relevan.

Teori graf memberikan kerangka kerja yang kuat untuk merepresentasikan hubungan antara pengguna dan film. Dalam

graf, simpul dapat mewakili pengguna atau film, sementara sisi yang menghubungkan simpul-simpul tersebut menunjukkan hubungan preferensi. Dengan menggunakan jenis graf berbobot, penulis dapat melihat serta menilai hubungan antar simpul berdasarkan intensitas preferensi pengguna terhadap film tertentu. Selain itu, algoritma rekursif digunakan untuk mengeksplorasi graf dan menemukan film yang belum ditonton oleh pengguna, yang didasarkan dengan kesamaan preferensi terhadap pengguna lain.

II. LANDASAN TEORI

A. Relasi

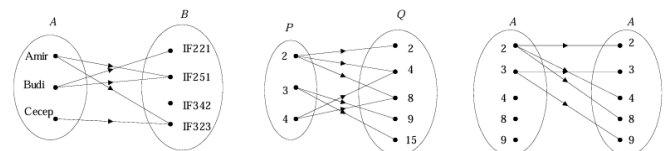
Relasi biner R himpunan A dan B adalah himpunan bagian dari $A \times B$, notasinya adalah $R \subseteq (A \times B)$.

- Bentuk notasi dari $(a, b) \in R$, yaitu a dihubungkan dengan b oleh R , adalah $a R b$
- Bentuk notasi dari $(a, b) \notin R$, yaitu a tidak dihubungkan dengan b oleh R , adalah $a \not R b$
- Himpunan A dapat disebut sebagai daerah asal (*domain*) dari R , dan Himpunan B dapat disebut sebagai daerah tujuan (*kodomain*) dari R .

Pada sebuah himpunan, Relasinya adalah relasi yang khusus. Relasi yang ada pada himpunan A adalah himpunan bagian dari $A \times A$, notasinya adalah $R \subseteq (A \times A)$

Relasi dapat direpresentasikan menjadi sebuah diagram panah (gambar 2.1), tabel, matriks, graf berarah, ataupun diagram kartesian.

1. Representasi relasi menggunakan diagram panah



Lingkaran kiri: daerah asal (*domain*)

Lingkaran kanan: daerah tujuan (*kodomain*)

Gambar 2.1 Representasi relasi dengan diagram panah (Diambil dari Rinaldi Munir Homepage)

2. Representasi relasi menggunakan tabel

Lingkaran kiri menyatakan daerah asal (*domain*) dan lingkaran kanan menyatakan daerah tujuan (*kodomain*).

Tabel 1		Tabel 2		Tabel 3	
A	B	P	Q	A	A
Amir	IF251	2	2	2	2
Amir	IF323	2	4	2	4
Budi	IF221	4	4	2	8
Budi	IF251	2	8	3	3
Cecep	IF323	4	8	3	3
		3	9		
		3	15		

Gambar 2.2 Representasi relasi dengan tabel (Diambil dari Rinaldi Munir Homepage)

Kolom pertama dalam tabel menyatakan sebuah daerah asal (*domain*) dan kolom kedua dalam tabel menyatakan sebuah daerah tujuan (*kodomain*).

3. Representasi relasi menggunakan Matriks

Misalkan R merupakan relasi dari $A = \{a_1, a_2, \dots, a_m\}$ serta $B = \{b_1, b_2, \dots, b_n\}$, maka relasi R dapat ditulis sebagai matriks $M = [m_{ij}]$.

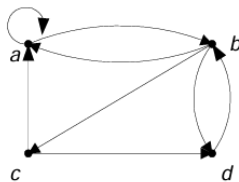
$$M = \begin{matrix} & \begin{matrix} b_1 & b_2 & \dots & b_n \end{matrix} \\ \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{matrix} & \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \dots & m_{mn} \end{bmatrix} \end{matrix}$$

Gambar 2.3 Representasi relasi dengan matriks (Diambil dari Rinaldi Munir Homepage)

dengan $m_{ij} = 1$ jika $(a, b) \in R$ dan $m_{ij} = 0$ jika $(a, b) \notin R$

4. Representasi Relasi menggunakan Graf Berarah

Terdapat $(a, b) \in R$ sehingga sebuah sisi dapat dibuat dari simpul a menuju ke simpul b . Simpul a dapat disebut sebagai simpul asal (*initial vertex*) dan simpul b dapat disebut sebagai simpul tujuan (*terminal vertex*). Pasangan terurut (a, a) bisa dinyatakan dengan sebuah sisi dari simpul a ke simpul a itu sendiri. Sisi ini disebut dengan gelang (*loop*).

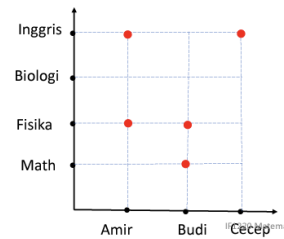


Gambar 2.4 Representasi Relasi menggunakan Graf Berarah (Diambil dari Rinaldi Munir Homepage)

5. Representasi Relasi menggunakan Diagram Kartesian.

Sumbu x menyatakan daerah asal (*domain*) dan sumbu y menyatakan daerah tujuan (*kodomain*). Elemen relasi

dinyatakan sebagai titik di dalam diagram kartesian



Gambar 2.5 Representasi Relasi dengan Diagram Kartesian (Diambil dari Rinaldi Munir Homepage)

Dengan menggunakan konsep relasi, kita bisa dengan mudah mengidentifikasi film yang telah ditonton oleh pengguna tertentu dan menganalisis preferensi dari masing-masing pengguna.

B. Teori Graf

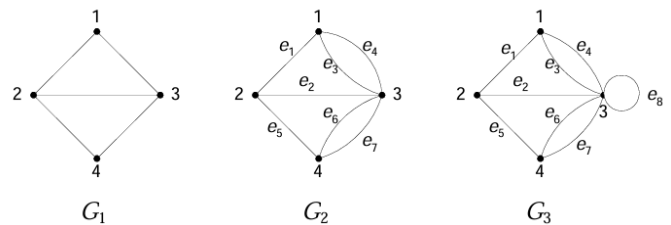
Graf biasanya dipakai untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.

Graf G bisa didefinisikan sebagai $G = (V, E)$, yang di mana: V adalah himpunan tidak kosong dari simpul-simpul (*vertices*), $V = \{v_1, v_2, \dots, v_n\}$.

Himpunan V tidak boleh kosong, yang berarti graf **tidak boleh** tidak mengandung simpul, atau bisa dikatakan bahwa graf harus memiliki simpul di dalamnya..

E adalah himpunan sisi (*edges*) yang menghubungkan sepasang simpul, $E = \{e_1, e_2, \dots, e_n\}$

Himpunan E boleh kosong, yang berarti graf **boleh** untuk tidak memiliki satupun sisi di dalamnya..



Gambar 2.6 Contoh Graf (Diambil dari Rinaldi Munir Homepage)

Dapat dilihat dari gambar 2.6,

G_1 merupakan graf dengan $V = \{1,2,3,4\}$, sedangkan $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$,

G_2 merupakan graf dengan $V = \{1,2,3,4\}$, sedangkan $E = \{(1,2), (2,3), (1,3), (1,3), (2,4), (3,4), (3,4)\}$, atau $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$,

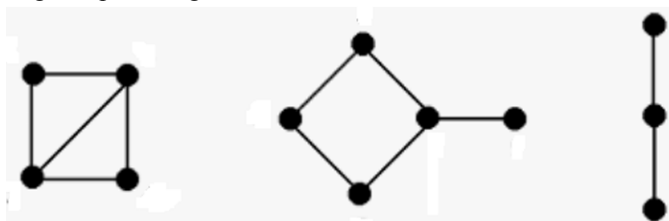
G_3 merupakan graf dengan $V = \{1,2,3,4\}$, sedangkan $E = \{(1,2), (2,3), (1,3), (1,3), (2,4), (3,4), (3,4), (3,3)\}$, atau $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

Jika kita melihat dari ada atau tidaknya suatu gelang atau sisi

ganda di dalam graf, graf dapat diklasifikasikan menjadi dua jenis:

1. Graf sederhana (*simple graph*)

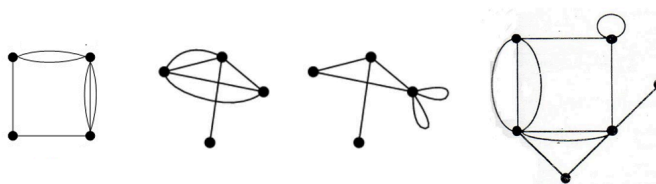
Graf sederhana adalah sebuah graf yang tidak mengandung gelang maupun sisi ganda.



Gambar 2.7 Contoh graf sederhana (Diambil dari Rinaldi Munir Homepage)

2. Graf tak-sederhana (*unsimple graph*)

Graf tak sederhana adalah sebuah graf yang mengandung sisi ganda atau gelang.



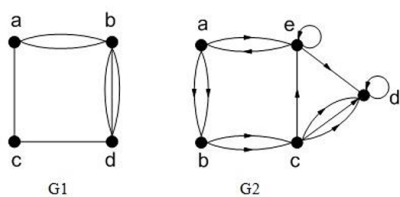
Gambar 2.8 Contoh graf tak-sederhana (Diambil dari Rinaldi Munir Homepage)

Graf tak sederhana dapat dibedakan lagi menjadi:

1. Graf ganda (*multi-graph*), yaitu graf yang mengandung sisi ganda, dan
2. Graf semu (*pseudo-graph*), yaitu graf yang mengandung sisi gelang.

Jika didasarkan terhadap orientasi arah pada sisi, graf dibedakan atas dua jenis:

1. Graf tak berarah, yaitu graf yang sisinya tidak memiliki orientasi arah, dan
2. Graf berarah, yaitu graf yang setiap sisinya diberikan orientasi arah tertentu.



G1 : graf tak-berarah; G2 : Graf berarah

Gambar 2.9 Contoh graf tak berarah dan graf berarah (Diambil dari Rinaldi Munir Homepage)

Terminologi dalam Graf

Graf juga memiliki terminologi di dalamnya, beberapa diantaranya:

1. Ketetanggaan (*Adjacent*)

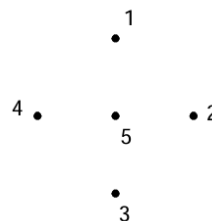
Dua simpul dapat dikatakan sebagai bertetangga apabila kedua simpul tersebut terhubung secara langsung.

2. Bersisian (*incidency*)

Sebuah sembarang sisi $e = (v_j, v_k)$ bisa dikatakan dengan e bersisian dengan simpul v_j , atau e bersisian dengan simpul v_k .

3. Graf Kosong (*Empty Graph*)

Graf kosong adalah suatu graf yang himpunan sisinya adalah himpunan yang kosong (N_n)



Gambar 2.10 Contoh graph kosong (Diambil dari Rinaldi Munir Homepage)

4. Derajat (*Degree*)

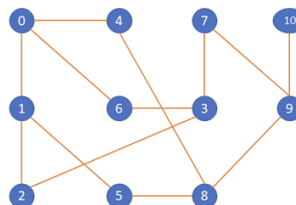
Derajat suatu simpul merupakan jumlah sisi yang bersisian dengan simpul tersebut. Notasi dari derajat suatu simpul adalah $d(v)$

5. Lintasan (*Path*)

Lintasan yang memiliki panjang n dari simpul awal v_n di dalam graf G merupakan barisan selang-seling dari simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, e_3, \dots, v_{n-1}, e_n, v_n$ sedemikian rupa sehingga

$$e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots,$$

$e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G . Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Dapat dilihat pada gambar 2.11 bahwa lintasan 0, 6, 3, 7, 9, 10 adalah lintasan dari simpul 0 ke simpul 10 dengan panjang lintasannya adalah 5.



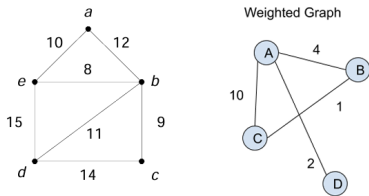
Gambar 2.11 Contoh Graf dengan Lintasan (Diambil dari Rinaldi Munir Homepage)

6. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Lintasan yang memiliki dan akhir pada satu simpul yang sama dapat disebut **sirkuit** atau **siklus**. Panjang suatu sirkuit dapat ditentukan dari jumlah sisi dalam sirkuit tersebut. Pada gambar 2.11, lintasan 0, 4, 8, 5, 1, 0 merupakan sebuah sirkuit yang memiliki panjang sirkuit yaitu 5.

7. Graf Berbobot (*Weighted Graph*)

Graf berbobot merupakan suatu graf yang setiap sisinya diberi sebuah harga (bobot).



Gambar 2.12 Contoh graf berbobot (Diambil dari Rinaldi Munir Homepage)

Dengan menggunakan algoritma dan konsep dari graf, kita bisa melakukan analisis yang lebih mendalam, seperti mencari jalur terpendek ataupun mengidentifikasi komunitas pada pengguna dengan preferensi yang sama.

C. Rekursi

Rekursi adalah suatu teknik pemrograman yang di mana suatu fungsi memanggil dirinya sendiri untuk menyelesaikan masalah yang ada. Sebuah objek dikatakan sebagai rekursif jika ia didefinisikan dalam terminologinya sendiri. Dalam konteks sistem rekomendasi, rekursi bisa digunakan untuk menganalisis rekomendasi film berdasarkan kesamaan preferensi antar pengguna. Algoritma dari rekursif ini memungkinkan suatu sistem untuk menjelajahi sebuah graf dan menemukan film yang belum ditonton oleh pengguna.

Sebagai contoh, jika kita ingin merekomendasikan sebuah film kepada pengguna X, algoritma rekursif akan mencari pengguna lain yang memiliki kesamaan dalam film yang telah ditonton. Dengan cara ini, sistem dapat memberikan rekomendasi yang lebih personal dan relevan berdasarkan pola perilaku pengguna lain yang serupa dengan pengguna X.

III. METODOLOGI

A. Dataset Sederhana

Dataset yang digunakan dalam penelitian ini merupakan representasi hubungan antar pengguna dan film yang disukai. Tabel berikut menunjukkan daftar 10 pengguna, film yang disukai, serta nilai (*rating*) dari film tersebut:

User	Film yang Disukai	Rating (1-5)
A	Nusa dan Rara, Harry Potter	4, 3
B	Harry Potter, Doraemon	5, 4
C	Nusa dan Rara, Doraemon, Spongebob	5, 3, 4
D	Spongebob, Upin dan Ipin	4, 5
E	Nusa dan Rara, Upin dan Ipin	3, 4
F	Harry Potter, Spongebob	4, 5
G	Doraemon, Upin dan Ipin	5, 3

H	Nusa dan Rara, Harry Potter, Doraemon	4, 3, 5
I	Spongebob, Upin dan Ipin	4, 5
J	Harry Potter, Doraemon, Upin dan Ipin	5, 4, 3

Tabel 3.1 Contoh Dataset Rekomendasi Film

Dataset ini dipilih karena kemudahannya untuk dianalisis serta kemampuannya untuk merepresentasikan hubungan preferensi secara intuitif. Bobot rating menunjukkan tingkat kesukaan pengguna terhadap film tertentu. Hubungan ini kemudian bisa dimodelkan ke dalam bentuk graf untuk analisis lebih lanjut.

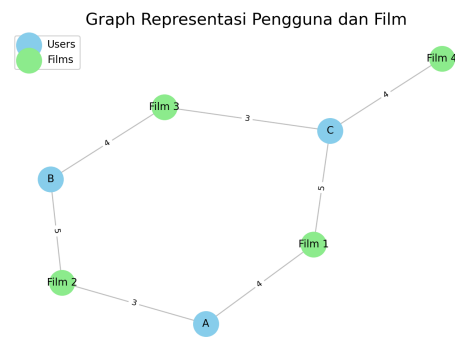
B. Representasi Graf

Graf yang dibentuk dari dataset di atas memiliki simpul (*node*) yang merepresentasikan pengguna dan filmnya, sementara itu sisi (*edges*) berbobot menghubungkan pengguna dengan film yang diberi rating. Bobot pada setiap sisi menunjukkan tingkat kesukaan pengguna terhadap film tersebut.

Langkah-langkah membentuk graf:

1. Identifikasi simpul-simpul pada graf, simpul terdiri dari dua jenis yaitu pengguna (User A, User B, dst.) dan film (Nusa dan Rara, Harry Potter, dst.).
2. Hubungkan simpul pengguna dengan simpul film berdasarkan data preferensi.
3. Tetapkan bobot pada setiap sisi sesuai dengan nilai rating yang diberikan oleh pengguna terhadap film.

Contoh graf:



Gambar 3.1 Contoh graf rekomendasi film

Graf ini memvisualisasikan hubungan langsung antara pengguna dengan film yang mereka sukai. Struktur ini memungkinkan identifikasi film yang relevan berdasarkan hubungan tidak langsung melalui pengguna lain.

C. Algoritma Rekomendasi

Pendekatan algoritmik dapat digunakan untuk menganalisis graf dan menghasilkan rekomendasi film. Algoritma ini akan memanfaatkan relasi antar pengguna dan bobot pada graf untuk

menentukan film yang relevan bagi pengguna target.

Langkah-langkah algoritma:

1. Identifikasi Simpul Awal : Pilih simpul pengguna target serta daftar film yang telah mereka tonton.
2. Analisis Relasi : Cari pengguna lain yang memiliki hubungan dengan pengguna yang diinginkan melalui film yang sama.
3. Telusuri Simpul Baru : Untuk setiap pengguna, identifikasi film yang belum ditonton oleh pengguna.
4. Pilih Rekomendasi : Film yang ditemukan akan ditambahkan ke daftar rekomendasi dengan mempertimbangkan bobot pada sisi yang relevan.

Implementasi Python:

Berikut ini adalah implementasi algoritma rekomendasi menggunakan Python berdasarkan dataset di atas:

```
# Dataset pengguna dan film yang disukai
users = {
    "A": {"Nusa dan Rara": 4, "Harry Potter": 3},
    "B": {"Harry Potter": 5, "Doraemon": 4},
    "C": {"Nusa dan Rara": 5, "Doraemon": 3,
    "Spongebob": 4},
    "D": {"Spongebob": 4, "Upin dan Ipin": 5},
    "E": {"Nusa dan Rara": 3, "Upin dan Ipin": 4},
    "F": {"Harry Potter": 4, "Spongebob": 5},
    "G": {"Doraemon": 5, "Upin dan Ipin": 3},
    "H": {"Nusa dan Rara": 4, "Harry Potter": 3,
    "Doraemon": 5},
    "I": {"Spongebob": 4, "Upin dan Ipin": 5},
    "J": {"Harry Potter": 5, "Doraemon": 4, "Upin
    dan Ipin": 3},
}

# Fungsi untuk mencari rekomendasi film
def rekomendasi(user, users):
    film_tonton = set(users[user].keys())
    rekomendasi = {}

    for pengguna_lain, film_lain in users.items():
        if pengguna_lain != user:
            film_sama =
            film_tonton.intersection(film_lain.keys())
            if film_sama:
                film_baru = set(film_lain.keys()) -
                film_tonton

                for film in film_baru:
                    if film not in rekomendasi:
                        rekomendasi[film] = 0
                    rekomendasi[film] +=
                    sum(film_lain[film_sama_film] for film_sama_film in
                    film_sama)

    return sorted(rekomendasi.items(), key=lambda
    x: x[1], reverse=True)
```

```
# Contoh penggunaan
user_target = input("Masukkan User: ")
hasil_rekomendasi = rekomendasi(user_target, users)
print(f"Rekomendasi untuk User {user_target}:
{hasil_rekomendasi}")
```

Penjelasan:

- Input : Dataset berisi pengguna dan film yang mereka beri rating
- Proses:
 - Cari pengguna lain dengan film yang sama seperti yang ditonton pengguna.
 - Identifikasi film yang belum ditonton oleh pengguna.
 - Hitung skor rekomendasi berdasarkan bobot dari film yang sama.
- Output : Daftar film yang direkomendasikan beserta skor prioritasnya.

Contoh saat program dijalankan :

```
Masukkan User: A
Rekomendasi untuk User A: [('Film 3', 22), ('Film 4', 9), ('Film 5', 8)]
```

Gambar 3.1 Contoh hasil eksekusi program

D. Pengembangan Lebih Lanjut

Pendekatan ini dapat diperluas dan dikembangkan lebih lanjut lagi dengan mempertimbangkan metadata tambahan seperti genre, aktor, durasi, atau tahun rilis film. Hal ini memungkinkan sistem untuk menghasilkan rekomendasi yang lebih personal dan akurat. Selain itu, metode ini juga dapat diintegrasikan dengan algoritma pembelajaran mesin untuk memanfaatkan dataset yang lebih besar.

IV. HASIL DAN DISKUSI

A. Visualisasi Graf

Kita dapat menggunakan implementasi Python untuk membentuk graf dari dataset, kode Python tersebut berupa:

```
import networkx as nx
import matplotlib.pyplot as plt

# Dataset pengguna dan film yang disukai
users = {
    "A": {"Nusa dan Rara": 4, "Harry Potter": 3},
    "B": {"Harry Potter": 5, "Doraemon": 4},
    "C": {"Nusa dan Rara": 5, "Doraemon": 3,
    "Spongebob": 4},
    "D": {"Spongebob": 4, "Upin dan Ipin": 5},
    "E": {"Nusa dan Rara": 3, "Upin dan Ipin": 4},
    "F": {"Harry Potter": 4, "Spongebob": 5},
    "G": {"Doraemon": 5, "Upin dan Ipin": 3},
    "H": {"Nusa dan Rara": 4, "Harry Potter": 3,
    "Doraemon": 5},
    "I": {"Spongebob": 4, "Upin dan Ipin": 5},
    "J": {"Harry Potter": 5, "Doraemon": 4, "Upin
    dan Ipin": 3},
}
```



```

# Membuat graf
G = nx.Graph()

# Menambahkan node pengguna dan film
for user, films in users.items():
    for film, rating in films.items():
        G.add_edge(user, film, weight=rating)

# Mengatur posisi node untuk visualisasi
pos = nx.spring_layout(G, seed=42)
# Memisahkan pengguna dan film untuk pewarnaan
user_nodes = [node for node in G.nodes if
node.startswith("User") or len(node) == 1]
film_nodes = [node for node in G.nodes if
node.startswith("Film")]

# Menggambar graf
plt.figure(figsize=(15, 10))
nx.draw_networkx_nodes(G, pos, nodelist=user_nodes,
node_color="skyblue", node_size=600, label="Users")
nx.draw_networkx_nodes(G, pos, nodelist=film_nodes,
node_color="lightgreen", node_size=600,
label="Films")
nx.draw_networkx_edges(G, pos, width=1, alpha=0.5,
edge_color="gray")
nx.draw_networkx_labels(G, pos, font_size=10,
font_color="black")

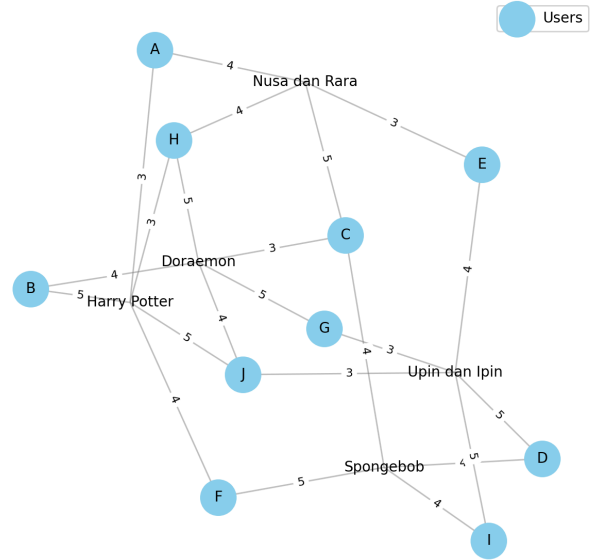
# Menambahkan label bobot pada edge
edge_labels = nx.get_edge_attributes(G, "weight")
nx.draw_networkx_edge_labels(G, pos,
edge_labels=edge_labels, font_size=8)

# Menyelesaikan visualisasi
plt.title("Graph Representasi Pengguna dan Film",
font_size=16)
plt.legend()
plt.axis("off")
plt.show()

```

Graf yang dihasilkan dari analisis dataset di atas dapat divisualisasikan sebagai:

Graph Representasi Pengguna dan Film



Gambar 4.1 Hasil Graf Dataset

Pada graf ini, hubungan antara pengguna dan film ditunjukkan oleh sisi yang berbobot. Bobot pada sisi mencerminkan nilai rating yang diberikan oleh pengguna terhadap film. Misalnya, hubungan User A dan Nusa dan Rara memiliki bobot 4, yang berarti User A sangat menyukai film tersebut sehingga memberikan rating 4.

Visualisasi graf ini memberikan gambaran intuitif tentang bagaimana pengguna saling terhubung melalui preferensi mereka terhadap film tertentu. Hubungan ini akan menjadi dasar untuk menghasilkan rekomendasi yang relevan.

B. Hasil Simulasi

Simulasi algoritma dilakukan untuk menghasilkan rekomendasi bagi setiap pengguna. Berikut merupakan hasil rekomendasi yang didapat menggunakan algoritma yang telah dibuat:

User	Film yang Disukai	Rekomendasi Film
A	Nusa dan Rara, Harry Potter	Spongebob, Upin dan Ipin
B	Harry Potter, Doraemon	Nusa dan Rara, Spongebob, Upin dan Ipin
C	Nusa dan Rara, Doraemon, Spongebob	Harry Potter, Upin dan Ipin
D	Spongebob, Upin dan Ipin	Nusa dan Rara, Doraemon
E	Nusa dan Rara, Upin dan Ipin	Harry Potter, Spongebob

F	Harry Potter, Spongebob	Nusa dan Rara, Upin dan Ipin
G	Doraemon, Upin dan Ipin	Harry Potter, Spongebob
H	Nusa dan Rara, Harry Potter, Doraemon	Spongebob, Upin dan Ipin
I	Spongebob, Upin dan Ipin	Harry Potter, Doraemon
J	Harry Potter, Doraemon, Upin dan Ipin	Nusa dan Rara, Spongebob

Tabel 4.1 Hasil Rekomendasi Film dari Dataset

Penjelasan Hasil:

1. User A : Spongebob dan Upin dan Ipin direkomendasikan berdasarkan preferensi pengguna lain dengan hubungan ke Nusa dan Rara dan Harry Potter.
2. User B : Nusa dan Rara, Spongebob, Upin dan Ipin, direkomendasikan karena pengguna lain menyukai film-film tersebut.
3. User H : Spongebob dan Upin dan Ipin direkomendasikan karena kesamaan preferensi terhadap Nusa dan Rara, Harry Potter, dan Doraemon.

C. Diskusi

Algoritma yang digunakan dalam penelitian ini dapat menunjukkan efektivitas dalam menghasilkan rekomendasi berbasis relasi sederhana. Beberapa poin diskusi yang penting untuk diketahui adalah:

1. Keuntungan :
 - a. Pendekatan berbasis graf memberikan representasi visual yang intuitif.
 - b. Algoritma rekursif memungkinkan eksplorasi hubungan yang lebih dalam antar pengguna.
2. Keterbatasan :
 - a. Tidak mempertimbangkan metadata seperti genre atau aktor.
 - b. Tidak menangani dataset besar secara efisien.
3. Potensi Pengembangan :
 - a. Menambahkan metadata untuk meningkatkan akurasi rekomendasi.
 - b. Mengintegrasikan metode *clustering* untuk mengelompokkan pengguna berdasarkan preferensi mereka masing-masing.

Implikasi

Sistem ini dapat diterapkan pada platform kecil dengan dataset terbatas, seperti komunitas berbagi film yang bersifat lokal. Untuk skala besar, integrasi dengan pembelajaran mesin menjadi sebuah langkah penting untuk meningkatkan performa.

V. KESIMPULAN

Dalam makalah ini, penulis telah membahas mengenai pengembangan sistem rekomendasi otomatis untuk film menggunakan kombinasi relasi, rekursi, dan teori graf. Pendekatan ini dapat memungkinkan sistem untuk memberikan rekomendasi yang relevan berdasarkan preferensi pengguna dengan memanfaatkan graf berbobot dan algoritma rekursif. Hasil simulasi menunjukkan bahwa sistem ini efektif dalam menghasilkan rekomendasi yang sesuai dengan pola perilaku pengguna.

Meskipun hasil yang diperoleh sudah cukup memuaskan, masih terdapat peluang untuk pengembangan lebih lanjut, seperti mempertimbangkan faktor tambahan dalam rekomendasi. Penelitian ini memberikan dasar atau fondasi yang kuat untuk pengembangan sistem rekomendasi yang lebih canggih di masa depan.

VI. ACKNOWLEDGMENT

Penulis menyampaikan rasa syukur kepada Tuhan Yang Maha Esa yang atas kemampuan yang diberikan sehingga penulis dapat menyelesaikan makalah ini dengan baik. Penulis juga berterima kasih kepada orang tua penulis dan teman-teman yang selalu memberikan dukungan dan nasihat sehingga dapat menjadi motivasi bagi penulis dalam menyelesaikan tugas perkuliahan di ITB.

Ucapan terima kasih juga ditujukan kepada Bapak Dr. Ir. Rinaldi Munir dan Bapak Ir. Rila Mandala, M.Eng., Ph.D. selaku dosen mata kuliah Matematika Diskrit, atas tugas dan materi yang berkaitan dengan Matematika Diskrit yang sangat berguna dalam penulisan makalah ini. Tak lupa, penulis juga berterima kasih kepada seluruh yang telah membantu, baik secara langsung maupun tidak langsung, dalam menyelesaikan makalah ini.

REFERENCES

- [1] R. Munir, *Relasi dan Fungsi (Bagian 1)*, Bahan Kuliah IF1220 Matematika Diskrit, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2024. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-\(2024\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-(2024).pdf). diakses pada 5 Januari 2025.
- [2] R. Munir, *Deretan, Rekursi, dan Relasi Rekurens (Bagian 1)*, Bahan Kuliah IF1220 Matematika Diskrit, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2024. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-\(2024\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-(2024).pdf). diakses pada 5 Januari 2025.
- [3] R. Munir, *Graf (Bagian 1)*, Bahan Kuliah IF1220 Matematika Diskrit, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2024. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-\(2024\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/05-Relasi-dan-Fungsi-Bagian1-(2024).pdf). diakses pada 5 Januari 2025.
- [4] Aggarwal, Charu C., "Recommender Systems: The Textbook," Springer, 2016. ISBN 9783319296579. [Online]. Available: <https://link.springer.com/book/10.1007/978-3-319-29658-6>.
- [5] S. K. S. Gupta, A. K. Gupta, "A Survey on Recommender Systems: Techniques and Applications," *International Journal of Computer Applications*, vol. 975, no. 8887, 2016. [Online]. Available: <https://www.ijcaonline.org/archives/volume975/number8887/17480-2016>.
- [6] Jannach, D., & Adomavicius, G., "Recommendation Systems: Challenges and Opportunities," *Computer Science Review*, vol. 6, no. 2, pp. 85-97,

2012. [Online]. Available:

<https://www.sciencedirect.com/science/article/pii/S1574013712000050>.

- [7] S. K. S. Gupta, A. K. Gupta, "A Survey on Recommender Systems: Techniques and Applications," International Journal of Computer Applications, vol. 975, no. 8887, 2016. [Online]. Available: <https://www.ijcaonline.org/archives/volume975/number8887/17480-2016>

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, 26 Desember 2024

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke at the end.

Muhammad Timur Kanigara 13523055