

# Penerapan Teori Graf pada Sistem Rekomendasi Mobile Games

Shannon Aurellius Anastasya Lie - 13523019<sup>1,2</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>[13523019@std.stei.itb.ac.id](mailto:13523019@std.stei.itb.ac.id) <sup>2</sup>[shannonlie23@gmail.com](mailto:shannonlie23@gmail.com)

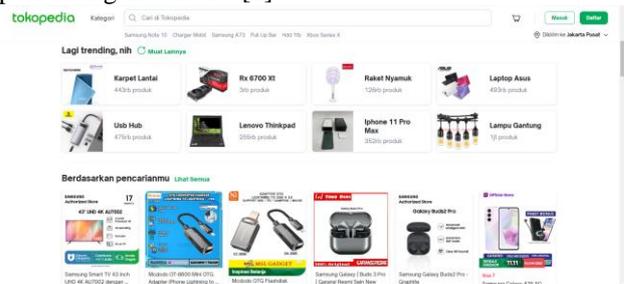
**Abstract**— Games are a type of entertainment enjoyed by everyone. Among various mediums of gaming, one of the most loved forms would probably be mobile games as they are played on smartphones supporting Android, iOS, and other operating systems. Recommendations hold importance for old and new users alike. While newcomers need proper guidance regarding some of the highest-rated games, existing users often tend to look for recommendations according to their favorite genres or games developed by their favorite developers. Graph-based approaches can be used to recommend mobile games by analyzing interrelationships among: developer, genre, and rating. K-Nearest Neighbors could be applied as a recommendation technique for generating such suggestions.

**Keywords**—Graph, recommendation, mobile games, K-Nearest Neighbors (KNN).

## I. PENDAHULUAN

Akhir-akhir ini, semakin banyak orang yang bermain games seiring perkembangan teknologi. Berdasarkan Menteri Komunikasi dan Informatika (Menkominfo) pada tahun 2024, lebih dari 220 juta orang di Indonesia merupakan pengguna internet aktif. Menurut data APJII 2024, angka tersebut mewakili lebih dari 70% populasi Indonesia. [1] Hal tersebut didukung oleh tren data tahun 2024 dari andi.link, Indonesia memiliki total populasi sebanyak 276,4 juta jiwa. Jumlah perangkat mobile yang terhubung mencapai 353,8 juta, yang setara dengan 128% dari total populasi. Pengguna internet di Indonesia tercatat sebanyak 212,9 juta orang, atau sekitar 77% dari total populasi. Sementara itu, pengguna media sosial aktif mencapai 167 juta, atau sekitar 60,4% dari total populasi. Waktu rata-rata setiap hari dalam penggunaan internet adalah 7 jam, 38 menit. Sedangkan, waktu rata-rata setiap hari waktu bermain game adalah 1 jam, 12 menit. Terdapat beberapa alasan utama penggunaan internet bagi orang di Indonesia yaitu untuk mencari informasi, komunikasi dengan teman dan keluarga mencari ide atau inspirasi baru, mengisi waktu luang dan penjelajahan umum, mengikuti berita dan peristiwa terkini, menonton video, acara televisi, atau film aktivitas, mengakses dan mendengarkan musik, mencari tahu cara melakukan sesuatu, mencari produk dan merek banyak orang mencari informasi tentang produk bertemu orang baru dan membuat koneksi baru, mencari informasi tentang tempat, liburan, dan perjalanan, bermain

game, mencari informasi tentang masalah kesehatan dan produk kesehatan menyampaikan pendapat, mengelola keuangan dan tabungan. Sebanyak 40.3% merupakan alasan utama penggunaan internet untuk bermain game pada orang Indonesia. [2]



Gambar 1. Penggunaan Sistem Rekomendasi pada E-Commerce Tokopedia

Sumber: <https://www.tokopedia.com/>

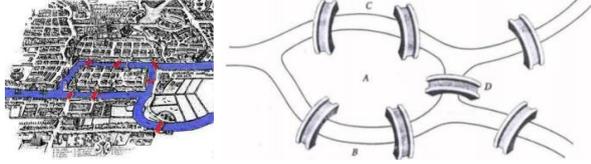
Menurut NVIDIA, sistem rekomendasi merupakan suatu kelas pembelajaran mesin yang menggunakan data untuk membantu memprediksi, mempersempit, dan menemukan apa yang dicari orang di antara jumlah pilihan yang tumbuh secara eksponensial. Komponen penting dari pengalaman pengguna yang dipersonalisasi adalah sistem rekomendasi, yang membantu membuat keputusan yang kuat dalam bidang ritel, hiburan, perawatan kesehatan, keuangan, dan bidang lainnya. Jika kualitas rekomendasi ditingkatkan 1%, rekomendasi dapat menghasilkan miliaran dolar. Hal ini dapat terjadi hingga 30% dari pendapatan pada beberapa platform komersial terbesar. Beberapa alasan penggunaan sistem rekomendasi adalah meningkatkan retensi pengguna, meningkatkan penjualan, membantu membentuk kebiasaan dan tren pelanggan, mempercepat laju kerja, meningkatkan nilai keranjang belanja. [3]

Graf dapat diaplikasikan untuk memberikan rekomendasi mobile games dengan algoritma K-Nearest Neighbors (KNN) yang banyak digunakan dalam sistem rekomendasi. Dalam graf ini, setiap node mewakili seorang pengguna atau sebuah game, sementara sisi atau hubungan antar node merepresentasikan interaksi, seperti nama game yang dimainkan, rating yang diberikan, dan developer game. KNN dapat mengidentifikasi berdasarkan interaksi atau persamaan preferensi pengguna dan menghasilkan rekomendasi pengguna yang lebih personal dan relevan secara efektif.

## II. LANDASAN TEORI

### A. Teori Graf

Graf umumnya digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Teori graf ditemukan pada tahun 1736 oleh seorang matematikawan Swiss, Leonhard Euler, setelah menyelesaikan persoalan jembatan Königsberg. [4]



**Gambar 2. Jembatan Königsberg (kiri)**

**Sumber:**

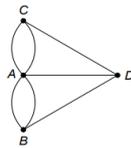
<https://old.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem>

**Gambar 3. Penyederhanaan Jembatan Königsberg**

**Sumber:**

<https://medium.com/@ulfafaudiah99/matematika-diskrit-graf-a23109909cc1>

Berikut merupakan graf persoalan jembatan Königsberg.



**Gambar 4. Graf Persoalan Jembatan Königsberg**

**Sumber:**

<https://medium.com/@ulfafaudiah99/matematika-diskrit-graf-a23109909cc1>

Graf  $G$  didefinisikan sebagai  $G = (V, E)$ .  $V$  merupakan himpunan tidak-kosong dari simpul-simpul (*vertices*) direpresentasikan dalam  $\{v_1, v_2, \dots, v_n\}$  namun himpunan  $V$  tidak boleh kosong, artinya graf tidak boleh tidak mengandung simpul.  $E$  merupakan himpunan sisi (*edges*) yang menghubungkan sepasang simpul direpresentasikan dalam  $\{e_1, e_2, \dots, e_n\}$ , himpunan  $E$  boleh kosong, artinya graf boleh tidak mengandung sisi satu buah pun. [4]

### B. Jenis-Jenis Graf

Graf digolongkan berdasarkan dua hal, yaitu berdasarkan ada tidaknya gelang atau sisi ganda dan orientasi arah pada sisi.

1. Berdasarkan ada tidaknya gelang atau sisi ganda, graf digolongkan menjadi dua macam:

a. Graf sederhana (*simple graph*).

Graf sederhana merupakan graf yang tidak mengandung gelang maupun sisi ganda.



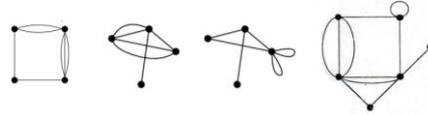
**Gambar 5. Graf Sederhana**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

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

Graf tak-sederhana merupakan graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple-grap*).



**Gambar 6. Graf Tak-Sederhana**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

Graf tak-sederhana dibedakan menjadi :

i. Graf ganda (*multi-graph*) merupakan graf yang mengandung sisi ganda.

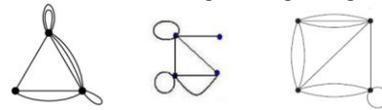


**Gambar 6. Graf Ganda**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

ii. Graf semu (*pseudo-graph*) merupakan graf yang mengandung sisi gelang.



**Gambar 6. Graf Semu**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

2. Berdasarkan orientasi arah pada sisi, graf dibedakan atas dua jenis:

a. Graf tak-berarah (*undirected graph*)

Graf tak-berarah merupakan graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.



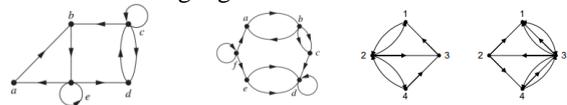
**Gambar 7. Graf Tak-Berarah**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

b. Graf berarah (*directed graph* atau *digraph*)

Graf berarah merupakan graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.



**Gambar 8. Graf Berarah**

**Sumber:**

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

### C. Unsur-Unsur Graf

Graf memiliki beberapa unsur, yaitu :

1. Ketetanggaan (*Adjacent*)  
Dua buah simpul dikatakan bertetangga jika keduanya terhubung langsung.
2. Bersisian (*Incidency*)  
Untuk sembarang sisi  $e = (v_j, v_k)$  dikatakan  $e$  bersisian dengan simpul  $v_j$ , atau  $e$  bersisian dengan simpul  $v_k$ .
3. Simpul Terpencil (*Isolated Vertex*)  
Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya.
4. Graf Kosong (*null graph* atau *empty graph*)  
Graf yang himpunan sisinya merupakan himpunan kosong ( $N_n$ ).
5. Derajat (*Degree*)  
Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Notasi derajat adalah  $d(v)$ .
6. Lintasan (*Path*)  
Lintasan yang panjangnya  $n$  dari simpul awal  $v_0$  ke simpul tujuan  $v_n$  di dalam graf  $G$  adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk  $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$  sedemikian 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  $G$ . Jika graf mengandung sisi ganda, maka sisi  $e_i$  perlu dituliskan di dalam lintasan. Jika graf sederhana (tidak mengandung sisi ganda), sisi  $e_i$  tidak perlu ditulis.
7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)  
Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus. Panjang sirkuit adalah jumlah sisi dalam sirkuit tersebut.
8. Kerterhubungan (*Connected*)  
Dua buah simpul  $v_1$  dan simpul  $v_2$  disebut terhubung jika terdapat lintasan dari  $v_1$  ke  $v_2$ .  $G$  disebut graf terhubung (*connected graph*) jika untuk setiap pasang simpul  $v_i$  dan  $v_j$  dalam himpunan  $V$  terdapat lintasan dari  $v_i$  ke  $v_j$ . Jika tidak, maka  $G$  disebut graf tak-terhubung (*disconnected graph*). Graf berarah  $G$  dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari  $G$  diperoleh dengan menghilangkan arahnya). Dua simpul,  $u$  dan  $v$ , pada graf berarah  $G$  disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari  $u$  ke  $v$  dan juga lintasan berarah dari  $v$  ke  $u$ . Jika  $u$  dan  $v$  tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka  $u$  dan  $v$  dikatakan terhubung lemah (*weakly connected*). Graf berarah  $G$  disebut graf terhubung kuat (*strongly connected graph*) apabila untuk setiap pasang simpul sembarang  $u$  dan  $v$  di  $G$ , terhubung kuat. Kalau tidak,  $G$  disebut graf terhubung lemah.
9. Upagraf (*Subgraph*) dan Komplemen Upagraf  
Misalkan  $G = (V, E)$  adalah sebuah graf.  $G_1 = (V_1, E_1)$  adalah upagraf (*subgraph*) dari  $G$  jika  $V_1 \subseteq V$

dan  $E_1 \subseteq E$ . Komplemen dari upagraf  $G_1$  terhadap graf  $G$  adalah graf  $G_2 = (V_2, E_2)$  sedemikian sehingga  $E_2 = E - E_1$  dan  $V_2$  adalah himpunan simpul yang anggota-anggota  $E_2$  bersisian dengannya. Komponen graf (*connected component*) adalah jumlah maksimum upagraf terhubung dalam graf  $G$ .

10. Upagraf Merentang (*Spanning Subgraph*)  
Upagraf  $G_1 = (V_1, E_1)$  dari  $G = (V, E)$  dikatakan upagraf merentang jika  $V_1 = V$  (yaitu  $G_1$  mengandung semua simpul dari  $G$ ).
11. *Cut-set*  
*Cut-set* dari graf terhubung  $G$  adalah himpunan sisi yang bila dibuang dari  $G$  menyebabkan  $G$  tidak terhubung. *Cut-set* selalu menghasilkan dua buah komponen.
12. Graf Berbobot (*Weighted Graph*)  
Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). [4]

### D. Sistem Rekomendasi

Sistem rekomendasi adalah kecerdasan buatan atau algoritma AI, yang biasanya dikaitkan dengan pembelajaran mesin, yang menggunakan *Big Data* untuk menyarankan atau merekomendasikan produk tambahan kepada konsumen. Rekomendasi ini dapat didasarkan pada berbagai kriteria, termasuk pembelian sebelumnya, riwayat pencarian, informasi demografi, dan faktor lainnya. Sistem rekomendasi sangat bermanfaat karena membantu pelanggan menemukan barang dan jasa yang mungkin tidak dapat mereka temukan sendiri. [3]

Sistem rekomendasi dilatih untuk memahami preferensi, keputusan sebelumnya, dan karakteristik orang dan produk menggunakan data yang dikumpulkan tentang interaksi berupa tayangan, klik, suka, dan pembelian. Sistem rekomendasi mampu memprediksi minat dan keinginan konsumen pada tingkat yang sangat personal, sistem rekomendasi menjadi favorit penyedia konten dan produk. Sistem ini dapat mengarahkan konsumen ke hampir semua produk atau layanan yang menarik minat mereka, mulai dari buku hingga video, kelas kesehatan, hingga pakaian. [3]

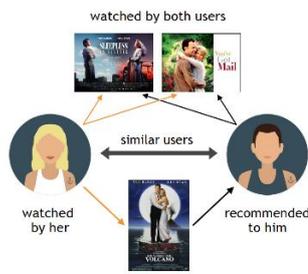
Sistem rekomendasi terbagi menjadi tipe *personalized* dan *non-personalized*. Tipe *non-personalized* tidak membedakan antar pengguna. Semua pengguna akan mendapatkan hasil rekomendasi sama. Tipe *non-personalized* biasanya digunakan untuk menampilkan item favorit pada film, pada *game*, produk terlaris di *e-commerce* atau *top hits music chart* di sistem rekomendasi musik. Terdapat dua cara untuk mendapatkan interaksi dari pengguna ke sistem yaitu dengan cara implisit (*browsing history*) dan eksplisit (*rating*). Pada tipe *personalized*, rekomendasi didasarkan pada data tentang interaksi pengguna dengan sistem sebelumnya. Rekomendasi yang dihasilkan berupa item khusus pada masing-masing pengguna. Tipe *personalized* paling banyak digunakan pada beberapa layanan populer seperti Amazon, Netflix, Youtube, dsb.

[5] Pada sistem rekomendasi *personalized*, terdapat tiga model utama yaitu:

1. *Collaborative Filtering*

Model ini merekomendasikan item (bagian penyaringan) berdasarkan informasi preferensi dari banyak pengguna (bagian kolaboratif). *Collaborative Filtering* merupakan model yang paling banyak digunakan dalam sistem rekomendasi. Pendekatan ini menggunakan kesamaan perilaku preferensi pengguna, mengingat interaksi sebelumnya antara pengguna dan item, algoritma pemberi rekomendasi untuk memprediksi interaksi di masa mendatang.

*Collaborative Filtering*



**Gambar 9. Representasi Model Collaborative Filtering**

Sumber: <https://www.nvidia.com/en-us/glossary/recommendation-system/>

*Collaborative Filtering* terbagi menjadi dua yaitu:

a. *Model based*

Mendefinisikan sebuah model untuk interaksi pengguna-item, di mana representasi pengguna dan item harus dipelajari dari matriks interaksi.

b. *Memory based*

Tidak mendefinisikan model untuk interaksi pengguna-item dan bergantung pada kesamaan antara pengguna atau item berdasarkan interaksi yang diamati.

2. *Content Filtering*

Model ini menggunakan atribut atau fitur suatu item (bagian konten) untuk merekomendasikan item lain yang serupa dengan preferensi pengguna. Pendekatan ini didasarkan pada kesamaan fitur item dan pengguna, dengan informasi tentang pengguna dan item yang pernah berinteraksi dengan mereka (misalnya usia pengguna, kategori masakan restoran, ulasan rata-rata untuk sebuah film), memodelkan kemungkinan interaksi baru.

*Content-based Filtering*



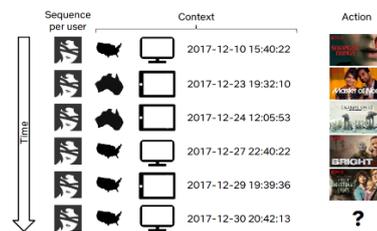
**Gambar 10. Representasi Model Content Filtering**

Sumber: <https://www.nvidia.com/en-us/glossary/recommendation-system/>

3. *Context Filtering*

Model ini menggunakan informasi kontekstual pengguna dalam proses rekomendasi. Sebagai contoh, Netflix membuat rekomendasi yang lebih baik dengan meringkaskan rekomendasi sebagai prediksi urutan kontekstual. Pendekatan ini menggunakan urutan tindakan pengguna kontekstual, ditambah konteks saat ini, untuk memprediksi kemungkinan tindakan berikutnya.

*Contextual sequence data*



**Gambar 11. Representasi Model Context Filtering**

Sumber: <https://www.nvidia.com/en-us/glossary/recommendation-system/>

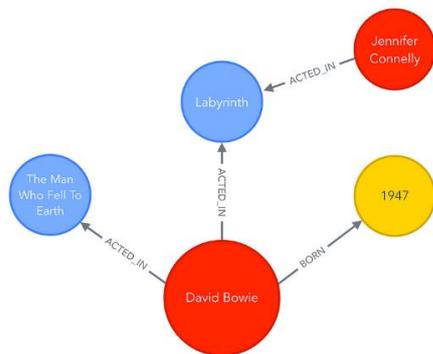
*E. Mobile Game*

*Mobile games* merupakan bentuk jamak dari *mobile game*. *Mobile game* terdiri dari dua kata yaitu ‘*mobile*’ yang berarti perangkat seluler dan ‘*game*’ yang berarti permainan. *Mobile game* merupakan permainan yang dirancang khusus untuk dimainkan pada perangkat seluler, seperti ponsel pintar dan tablet. Permainan tersebut dapat diunduh melalui toko aplikasi yang tersedia pada perangkat seluler, seperti *App Store* untuk perangkat *Apple iOS*, *Google Play Store* untuk perangkat *Android*, dan berbagai toko aplikasi lainnya untuk platform seluler yang lain. Menurut Rollings dan Adams (2006), *game mobile* merupakan sebuah teknologi dibandingkan sebuah genre atau jenis permainan, sebuah mekanisme untuk menghubungkan pemain bersama dibandingkan pola tertentu dalam sebuah permainan. [7]

*F. Knowledge Graph*

*Knowledge Graph* merupakan representasi data yang berbasis graf, yang menghubungkan berbagai entitas (seperti orang, tempat, atau objek) melalui hubungan atau atribut yang terdefinisi secara eksplisit. *Knowledge graph* dirancang untuk menyimpan informasi dalam

struktur yang terorganisasi, memungkinkan pengolahan dan analisis data yang lebih efektif untuk mendapatkan wawasan yang bermakna. Setiap node dalam graf mewakili entitas, sedangkan setiap sisi (*edge*) menghubungkan entitas-entitas tersebut melalui hubungan tertentu. Contoh penerapan *knowledge graph* mencakup sistem pencarian cerdas, asisten virtual, dan rekomendasi produk. Dalam sistem rekomendasi, *knowledge graph* membantu mengidentifikasi hubungan kompleks antara pengguna dan item, sehingga menghasilkan rekomendasi yang lebih akurat dan relevan. *Knowledge graph* memiliki beberapa fungsi utama yaitu meningkatkan kualitas rekomendasi, mendukung sistem pencarian yang semantik, dan sebagai pengintegrasian data. Berikut contoh *knowledge graph*.



**Gambar 12. Knowledge Graph**

**Sumber:** <https://towardsdatascience.com/knowledge-graphs-applied-in-the-retail-industry-ecac4e7baf8>

#### F. K-Nearest Neighbors

*K-Nearest Neighbors* (KNN) merupakan suatu algoritma *machine learning* yang termasuk dalam jenis *supervised learning*, yang dapat diterapkan pada masalah klasifikasi dan regresi. *Supervised learning* memerlukan data berlabel untuk melatih model sehingga mampu membuat prediksi ketika diberikan data baru yang belum memiliki label. Sebaliknya, dalam *unsupervised learning*, algoritma bekerja tanpa menggunakan label dan mencoba menemukan pola atau struktur tersembunyi dalam data untuk memberikan wawasan baru. Masalah klasifikasi biasanya menghasilkan output berupa kategori atau label diskrit, sementara regresi menghasilkan nilai dalam bentuk angka real. [8]

Pada dasarnya, algoritma KNN didasarkan pada asumsi bahwa objek-objek yang serupa akan memiliki jarak yang lebih dekat satu sama lain. Untuk menentukan jarak antara dua titik dalam graf, KNN menggunakan berbagai metode penghitungan jarak, seperti Euclidean, Manhattan, atau lainnya. Proses algoritma KNN dapat dirangkum dalam langkah-langkah berikut:

1. Menerima data masukan.
2. Menentukan nilai *k*, yaitu jumlah tetangga yang akan dipertimbangkan.
3. Menghitung jarak antara setiap data dalam dataset dengan data query.

4. Menyusun daftar jarak dalam urutan menaik.
5. Memilih *k* tetangga terdekat berdasarkan jarak terkecil.
6. Untuk klasifikasi, menentukan kelas mayoritas dari *k* tetangga.
7. Untuk regresi, menghitung rata-rata nilai label dari *k* tetangga

Dalam proses pemilihan tetangga terdekat, penggunaan metode sorting yang tidak efisien seperti brute-force dapat dihindari dengan menggunakan algoritma yang lebih optimal. Alternatif yang dapat digunakan meliputi:

- Partial Heapsort: Membentuk heap dari semua elemen data, kemudian mengekstraksi *m* elemen terkecil. Kompleksitas terburuk dari metode ini adalah  $\Theta(n+m \log n)$
- On-line Selection: Membentuk heap untuk *m* elemen pertama, kemudian memproses sisa elemen secara bertahap dan memperbarui heap jika diperlukan. Kompleksitas terburuknya adalah  $\Theta(n \log m)$ .
- Quickselect dan Quicksort: Menggunakan algoritma Quickselect untuk menemukan elemen terkecil ke-*m* diikuti oleh Quicksort untuk mengurutkan elemen lainnya. Kompleksitas rata-ratanya adalah  $\Theta(n + m \log m)$ .

Algoritma KNN menggunakan metrik jarak. Terdapat 4 cara dalam menemukan metrik jarak yaitu Euclidean, Manhattan, Minkowski, dan Hamming. Euclidean berfungsi dengan cara mengukur jarak garis lurus antara dua vektor. Manhattan mengukur nilai absolut perbedaan antara dua vektor (geometri taksi). Minkowski menggeneralisasi jarak Euclidean dan Manhattan (bergantung pada parameter *p*). Hamming digunakan untuk mengukur ketidakcocokan antara dua vektor. [9]

Sebagai alternatif dari metode sorting tradisional, KD-Trees dapat digunakan. Struktur data ini mendukung partisi ruang (*spatial partitioning*) dan sangat efektif untuk operasi pencarian tetangga terdekat (NN) maupun tetangga terdekat *k* (KNN). [8]

### III. PEMBAHASAN

Sistem rekomendasi telah menjadi bagian integral dari pengalaman pengguna di berbagai platform digital, termasuk industri game seluler. Seiring dengan meningkatnya jumlah game seluler yang tersedia di pasar dengan pesat, terdapat peningkatan kebutuhan akan sistem yang dapat memberikan rekomendasi yang akurat dan relevan. Pendekatan yang efektif untuk mengatasi tantangan ini adalah dengan menggunakan teori grafik ketika merancang sistem pemberi rekomendasi.

Teori graf adalah salah satu cabang matematika yang mempelajari hubungan antar objek yang terhubung. Dalam konteks sistem rekomendasi game seluler, objek tersebut dapat berupa game, pengembang, genre, *rating*, dan berbagai atribut lainnya. Teori graf memungkinkan untuk memodelkan hubungan antar objek dalam bentuk

graf yang terdiri dari *node* dan *edge*. *Node* mewakili entitas, dan tepi menggambarkan hubungan antara entitas ini. Selain itu, dapat memberikan rekomendasi berdasarkan kedekatan atau kemiripan antar game yang ada. Pembahasan akan dijelaskan bagaimana teori graf dapat diterapkan untuk membangun sistem rekomendasi yang mampu memberikan saran game yang relevan kepada pengguna, dengan memanfaatkan struktur graf untuk menghubungkan berbagai informasi yang dimiliki oleh game mobile, seperti genre, developer, dan rating. Selain itu, akan dibahas pula manfaat dan tantangan yang dihadapi dalam penerapan teori graf dalam sistem rekomendasi game mobile.

### A. Dataset

Menurut [dibimbing.id](https://www.kaggle.com/datasets/dem0nking/mobile-games-android-and-ios-rating-dataset), *dataset* merupakan kumpulan data yang disusun dalam format yang terstruktur, seperti tabel atau *file*, dan berisi informasi dari berbagai sumber. *Dataset* dapat berupa data numerik, teks, gambar, atau gabungan dari semuanya. Keberagaman jenis data set ini memungkinkan penerapan dalam berbagai industri dan disiplin ilmu. *Dataset* diperoleh melalui situs kaggle. *Dataset* ini digunakan untuk menghasilkan urutan rekomendasi *mobile game* yang sesuai dengan preferensi konsumen. Dataset berikut berisi 101 nama *mobile game*. Metode *collaborative filtering* akan digunakan dalam menganalisis dan menghasilkan rekomendasi yang sesuai.

Berikut merupakan tabel sampel data yang diperoleh:

	Game Name	Developer	Genre	Rating
0	Candy Crush Saga	King	Puzzle	4.6
1	Clash of Clans	Supercell	Strategy	4.5
2	Among Us	Innersloth	Party	4.4
3	Pokémon GO	Niantic	Augmented Reality	4.3
4	PUBG Mobile	Tencent Games	Battle Royale	4.2
5	Roblox	Roblox Corporation	Sandbox	4.5
6	Fortnite	Epic Games	Battle Royale	4.1
7	Genshin Impact	miHoYo	Action RPG	4.7
8	Call of Duty: Mobile	Activision	Shooter	4.3
9	Subway Surfers	SYBO Games	Endless Runner	4.5
10	Temple Run	Imangi Studios	Endless Runner	4.2
11	Angry Birds	Rovio Entertainment	Arcade	4.3
12	Fruit Ninja	Halfbrick Studios	Arcade	4.4
13	Brawl Stars	Supercell	Action	4.5
14	Minecraft	Mojang	Sandbox	4.8
15	Asphalt 9: Legends	Gameloft	Racing	4.6
16	8 Ball Pool	Miniclip	Sports	4.5
17	Homescapes	Playrix	Puzzle	4.4
18	Gardenscapes	Playrix	Puzzle	4.4
19	Clash Royale	Supercell	Strategy	4.5
20	Mario Kart Tour	Nintendo	Racing	4.7

Gambar 12. Mobile Games Rating Dataset

Sumber:

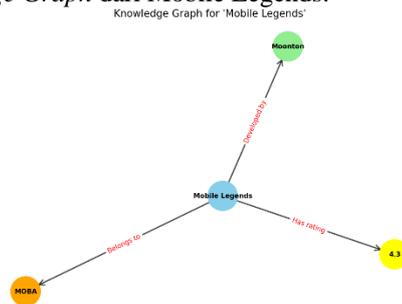
<https://www.kaggle.com/datasets/dem0nking/mobile-games-android-and-ios-rating-dataset>

Kolom Game Name menunjukkan nama dari *game*. Selain itu, kolom Developer menunjukkan pihak pengembang *game*. Kemudian, Genre menunjukkan jenis permainan. Setelah itu, kolom Rating berisi penilaian terhadap *game* yang telah dinilai oleh orang lain. Nilai Rating dalam *dataset* terdiri pada rentang 4.1 – 4.9.

### B. Konstruksi Graf

Konstruksi Graf menggunakan *dataset*, *node* terdiri dari *Game Name*, *Developer*, *Genre*, dan *Rating*. Sedangkan, *edges* menyatakan relasi *Game Name* dengan *node* lainnya seperti “Developed by”, “Belongs to”, dan “Has rating”. *Node*(simpul) merupakan elemen utama dalam graf yang mewakili entitas atau objek tertentu. *Edges*(sisi) merupakan relasi atau hubungan antara simpul-simpul

yang ada dalam graf. Keberadaan sisi menggambarkan adanya hubungan antar entitas. Implementasi berikut menggunakan *Python library*. Berikut merupakan contoh *Knowledge Graph* dari Mobile Legends.



Gambar 13. Knowledge Graph dari Mobile Legends

Sumber: Dokumen Pribadi

Berdasarkan Gambar 13, dapat diketahui bahwa Mobile Legends merupakan entitas utama dalam *dataset* yaitu *game name*. *Game* ini dikembangkan oleh Moonton, merupakan *game* dengan genre MOBA, dan memiliki *rating* sebesar 4.3.

### C. Implementasi Sistem Rekomendasi Mobile Games Menggunakan Bahasa Pemrograman Python

Implementasi dilakukan dengan menggunakan *library* Python yaitu *pandas*, *networkx*, *matplotlib.pyplot*, *sklearn*, dan *scipy*. Secara umum, program akan meminta *input* mengenai apa yang ingin dijadikan dasar rekomendasi, atau ingin melihat daftar *game* yang ada di *dataset*, atau ingin menampilkan *Knowledge Graph* dari suatu nama *game*. Algoritma KNN yang digunakan menerapkan metrik jarak Euclidean

Berikut merupakan implementasi pada bagian utama.

```

PS C:\One Drive\Documents\Sem3\Wakalah Matdis> python recommendation.py
HAIIII, SELAMAT DATANG DI REKOMENDASI MOBILE GAMES !!!!

Apa yang ingin Anda lakukan?
1. Lihat daftar game
2. Rekomendasi Game berdasarkan nama game, developer, atau genre
3. Lihat knowledge graph
4. Rekomendasi Game berdasarkan rating tertinggi
5. Keluar
Masukkan pilihan Anda (1/2/3/4/5): █
  
```

Gambar 14. Implementasi Program Utama

Sumber: Dokumen Pribadi

Pada bagian utama, terdapat ucapan selamat datang di bagian atas, kemudian dilanjutkan dengan pertanyaan. Setelah itu, masukkan pilihan (1/2/3/4/5) untuk mengaksesnya. Sebagai catatan, pertanyaan dan permintaan untuk memasukkan pilihan akan terus-menerus ditanyakan kecuali memilih “5”.

```

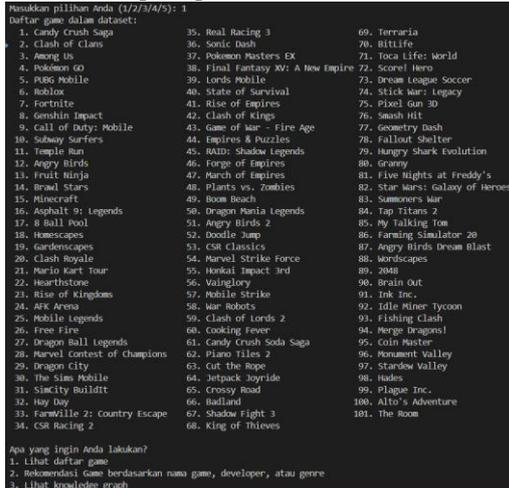
Apa yang ingin Anda lakukan?
1. Lihat daftar game
2. Rekomendasi Game berdasarkan nama game, developer, atau genre
3. Lihat knowledge graph
4. Rekomendasi Game berdasarkan rating tertinggi
5. Keluar
Masukkan pilihan Anda (1/2/3/4/5): 6
Pilihan tidak valid.
  
```

Gambar 15. Validasi Input pada Program Utama

Sumber: Dokumen Pribadi

Terdapat validasi pilihan sehingga karakter atau kata apapun yang dimasukkan akan menghasilkan pesan “Pilihan tidak valid.” dan diminta untuk memasukkan

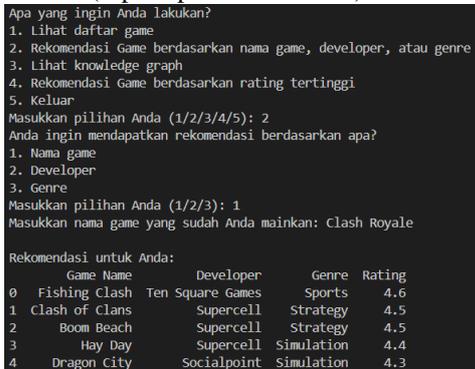
pilihan kembali (seperti pada Gambar 14).



Gambar 16. Lihat Daftar Game

Sumber: Dokumen Pribadi

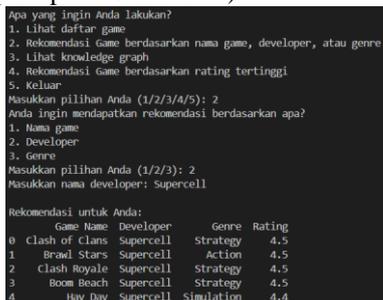
Gambar 16, menghasilkan keluaran yang sesuai dengan pilihan yang dipilih yaitu "1" (Lihat daftar game). Dapat terlihat bahwa terdapat 101 data nama game pada dataset tersebut. Setelah itu, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).



Gambar 17. Kasus Rekomendasi Game berdasarkan Nama Game

Sumber: Dokumen Pribadi

Pada Gambar 17, dimasukkan pilihan 2 (Rekomendasi Game berdasarkan nama game, developer, atau genre) kemudian dilanjutkan dengan memasukkan pilihan 1 (Nama game). Perilaku tersebut menandakan bahwa pengguna ingin mencari rekomendasi game berdasarkan nama game. Setelah itu, diikuti dengan memasukkan nama game yang sudah dimainkan yaitu Clash Royale, lalu program akan mengeluarkan hasil rekomendasinya. Kemudian, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).

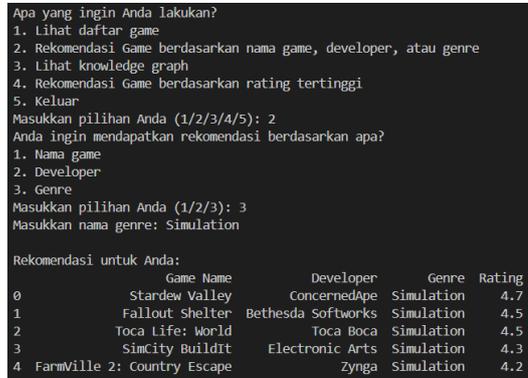


Gambar 18. Kasus Rekomendasi Game berdasarkan

## Developer Game

Sumber: Dokumen Pribadi

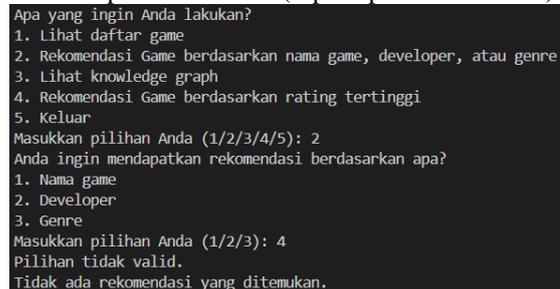
Pada Gambar 18, dimasukkan pilihan 2 (Rekomendasi Game berdasarkan nama game, developer, atau genre) kemudian dilanjutkan dengan memasukkan pilihan 2 (Developer). Perilaku tersebut menandakan bahwa pengguna ingin mencari rekomendasi game berdasarkan nama game. Setelah itu, diikuti dengan memasukkan Supercell sebagai developer, lalu program akan mengeluarkan hasil rekomendasinya. Lalu, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).



Gambar 19. Kasus Rekomendasi Game berdasarkan Genre

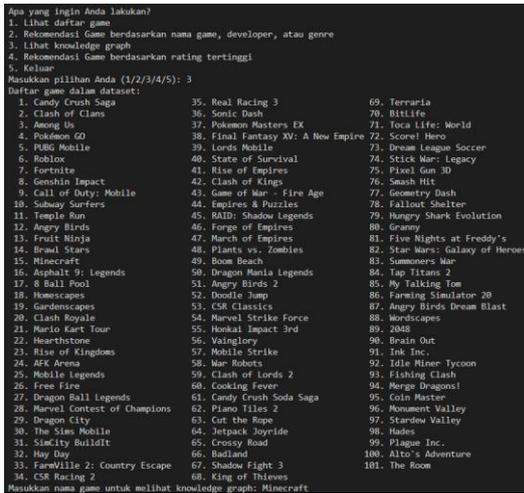
Sumber: Dokumen Pribadi

Berdasarkan Gambar 18, dimasukkan pilihan 2 (Rekomendasi Game berdasarkan nama game, developer, atau genre) kemudian dilanjutkan dengan memasukkan pilihan 3 (Genre). Perilaku tersebut menandakan bahwa pengguna ingin mencari rekomendasi game berdasarkan nama game. Setelah itu, diikuti dengan memasukkan Simulation sebagai Genre yang ingin diketahui rekomendasi gamenya, lalu program akan mengeluarkan hasil rekomendasinya. Kemudian, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).



Gambar 20. Validasi Input Rekomendasi Game Berdasarkan Nama Game, Developer, atau Genre

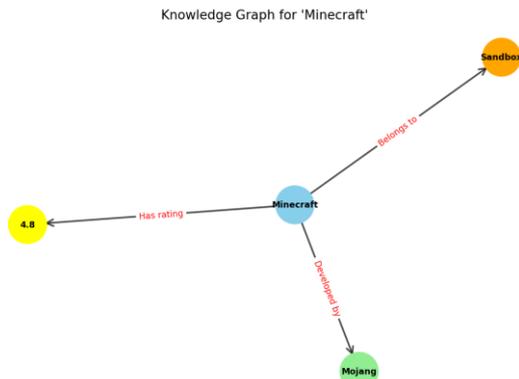
Pada bagian input rekomendasi game berdasarkan nama game, developer, dan genre, terdapat validasi pilihan sehingga karakter atau kata apapun yang dimasukkan akan menghasilkan pesan "Pilihan tidak valid. Tidak ada rekomendasi yang ditemukan" dan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).



**Gambar 21. Kasus Pemilihan Lihat Knowledge Graph**  
**Sumber: Dokumen Pribadi**

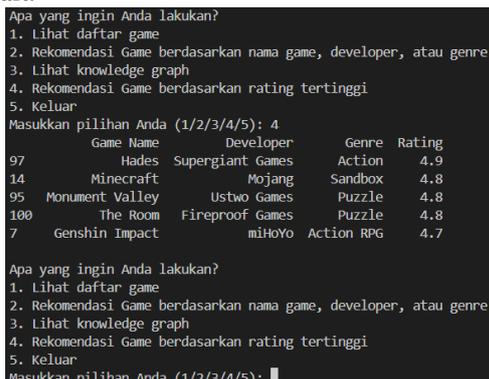
Pada Gambar 21, dimasukkan pilihan 3 (Lihat Knowledge Graph). Setelah itu, diikuti dengan memasukkan Minecraft sebagai nama *game* yang ingin ditampilkan hasil grafnya, lalu program akan mengeluarkan hasil rekomendasinya. Kemudian, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).

Berikut merupakan hasil dari langkah pada Gambar 21.



**Gambar 22. Knowledge Graph dari Minecraft**  
**Sumber: Dokumen Pribadi**

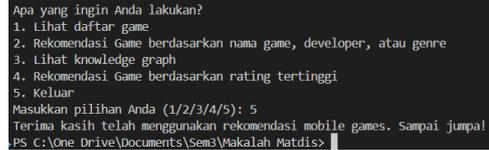
Berdasarkan Gambar 22, dapat diketahui bahwa Minecraft merupakan entitas utama dalam *dataset* yaitu *game name*. *Game* ini dikembangkan oleh Mojang, merupakan *game* dengan genre Sandbox, dan memiliki *rating* sebesar 4.8. *Rating* 4.8 cukuplah tinggi sehingga dapat disimpulkan bahwa banyak orang yang suka bermain Minecraft.



**Gambar 23. Kasus Rekomendasi Game Berdasarkan Rating Tertinggi**

**Sumber: Dokumen Pribadi**

Gambar 23, menghasilkan keluaran yang sesuai dengan pilihan yang dipilih yaitu “4” (Rekomendasi Game berdasarkan rating tertinggi). Lalu, program menampilkan 5 *mobile game* yang memiliki *rating* tertinggi. Setelah itu, akan diminta untuk memasukkan pilihan kembali (seperti pada Gambar 14).



**Gambar 24. Contoh Penggunaan Program Utama**  
**Sumber: Dokumen Pribadi**

Pada Gambar 24, dimasukkan “5” (Keluar) sehingga pengguna telah berhasil keluar dari program.

#### IV. KESIMPULAN

Penerapan teori graf khususnya *Knowledge Graph* dalam sistem rekomendasi *mobile games* terbukti efektif dan mampu mengorganisasi dan menganalisis data pada *dataset* yang melibatkan berbagai elemen, seperti *Game Name*, *Developer*, *Genre*, dan *Rating*. Struktur graf dapat dimanfaatkan untuk memvisualisasikan hubungan antar entitas dengan jelas melalui *node* dan *edge*.

Dalam penulisan makalah ini, penggunaan algoritma *K-Nearest Neighbors* (KNN) telah diterapkan untuk merekomendasikan *game* berdasarkan fitur-fitur seperti *developer*, *genre*, dan *rating*. KNN membantu dalam mengidentifikasi *game-game* yang memiliki kesamaan dengan *game* yang dipilih oleh pengguna, memberikan rekomendasi yang relevan dan personal. Penentuan kedekatan antar *game* berdasarkan jarak Euclidean memungkinkan sistem untuk memberikan saran *game* yang memiliki karakteristik serupa, meningkatkan pengalaman pengguna dalam menemukan *game* yang sesuai dengan preferensi pengguna.

#### REFERENSI

- [1] Komdigi, “Siaran Pers: Menkominfo Budi Arie Tegaskan Indonesia Telah Mengalami Kemajuan Transformasi Digital.” Tersedia: <https://www.komdigi.go.id/berita/siaran-pers/detail/siaran-pers-tentang-menkominfo-budi-arie-tegaskan-indonesia-telah-mengalami-kemajuan-transformasi-digital> [Diakses 5 Januari 2025].
- [2] Andi, “Hootsuite & We Are Social: Data Digital Indonesia 2024.” Tersedia: <https://andi.link/hootsuite-we-are-social-data-digital-indonesia-2024/> [Diakses 5 Januari 2025].
- [3] NVIDIA, “Recommendation System.” Tersedia: <https://www.nvidia.com/en-us/glossary/recommendation-system/> [Diakses 5 Januari 2025].
- [4] Munir, Rinaldi, “Graf: Bagian 1.” Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf> [Diakses 5 Januari 2025].
- [5] Laksito, Arif, “Sistem Rekomendasi: Non-Personalized vs Personalized.” Tersedia: <https://blog.ariflaksito.net/2021/04/sistem-rekomendasi-non-personalized-vs.html> [Diakses 5 Januari 2025].
- [6] Towards Data Science, “Knowledge Graphs Applied in the Retail Industry.” Tersedia: <https://towardsdatascience.com/knowledge-graphs-applied-in-the-retail-industry-ecac4e7baf8> [Diakses 5 Januari 2025].

- [7] Puspitasari, Oktaviani, "Mengenal Game Mobile." Tersedia: <https://www.kompasiana.com/oktavianipuspitasari2770/6513d7d54addee576168cab2/mengenal-game-mobile> [Diakses 5 Januari 2025].
- [8] Medium, "K-Nearest Neighbor." Tersedia: <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4> [Diakses 5 Januari 2025].
- [9] K. Gohrani, "Different Types of Distance Metrics Used in Machine Learning," *Medium*, 10 Februari 2020. Tersedia: [https://medium.com/@kunal\\_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7](https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7) [Diakses 5 Januari 2025].

#### 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, 6 Januari 2025



Shannon Aurellius Anastasya Lie  
13523019