

Aplikasi Graf pada Sistem Rekomendasi dalam *E-Commerce*

Hafshy Yazid Albisthami 13518051
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518051@std.stei.itb.ac.id

Abstrak—Makalah ini membahas salah satu aplikasi graf yang digunakan pada Sistem Rekomendasi. Sistem Rekomendasi adalah cabang dari Sistem Penyaringan Informasi yang dapat menghasilkan referensi atau rekomendasi yang ditawarkan kepada pengguna pada suatu aplikasi atau platform. Dengan adanya sistem rekomendasi, pengguna dapat dengan mudah mendapatkan sesuatu yang dicari dengan prediksi data *history* yang didapat dengan cara eksplisit dan implisit. Sistem Rekomendasi biasanya terdapat pada media sosial, *search engine*, *E-Commerce*, dan lain-lain. Sistem rekomendasi memiliki metode yang dapat diaplikasikan dengan cara *Collaborative Filtering*, *Content-Based Filtering*, dan *Hybrid Recommendation System*. Ketiga metode tersebut dapat diaplikasikan dengan menggunakan teori graf. Makalah ini akan membahas lebih lanjut mengenai pengaplikasian graf pada sistem rekomendasi.

Kata Kunci—Graf, Sistem Rekomendasi, *E-Commerce*, dan Rekomendasi.

I. LATAR BELAKANG

Sistem rekomendasi (*recommender system* atau *system recommendation*) dapat membuat suatu aplikasi atau platform lebih interaktif dengan memberikan referensi atau rekomendasi yang sekiranya sesuai dengan apa yang dicari atau diinginkan oleh pengguna. Sistem rekomendasi memanfaatkan *history* pencarian pengguna, kesamaan *behavior* dari pengguna itu sendiri dengan pengguna lain, *rating* pengguna dan lain-lain. Kemudian sistem rekomendasi diharapkan dapat membantu memberikan pengalaman yang baik untuk pengguna melalui pengalaman pencarian *item* yang mudah.

Proses penerimaan *input* untuk analisa sistem rekomendasi dapat didapat dengan dua cara yaitu eksplisit dan implisit. *Input* secara eksplisit adalah *input* dengan cara menanyakan langsung kepada pengguna melalui survei, penilaian, atau pencarian yang tersedia di aplikasi atau platform. Sedangkan *input* secara implisit adalah *input* yang memanfaatkan *history* dari pengguna dan pengguna lainnya. *Input* secara implisit dapat berupa analisa frekuensi pencarian pengguna, kesamaan *behavior* dengan pengguna lain, dan jenis-jenis *item* yang pernah dilihat sebelumnya.

Aplikasinya pada *E-Commerce* diantaranya adalah penampilan rekomendasi barang-barang yang sekiranya dibutuhkan atau diinginkan oleh pengguna. Contoh *input* untuk menghasilkan rekomendasi pada *E-Commerce* diantaranya

history pencarian barang, barang-barang yang sudah pernah dilihat, kategori yang sering dikunjungi, *behavior* pencarian yang sama dengan pengguna lain, *rating* yang diberikan pengguna terhadap suatu barang, dan lain-lain.

Judul makalah ini dipilih karena merupakan topik yang menarik untuk dibahas pada saat penulis menyusun makalah ini yaitu saat teknologi, internet, dan minat orang dalam berbelanja *online* sedang meningkat pesat. Selain itu sistem rekomendasi dapat membantu *E-Commerce* dalam menarik minat penggunanya.

II. LANDASAN TEORI

2.1 Teori Graf [1]

2.1.1 Definisi Graf

Misalkan graf $G = (V, E)$, dengan:

V adalah himpunan simpul-simpul tidak kosong.

E adalah himpunan sisi yang menghubungkan simpul.

2.1.2 Jenis-Jenis Graf

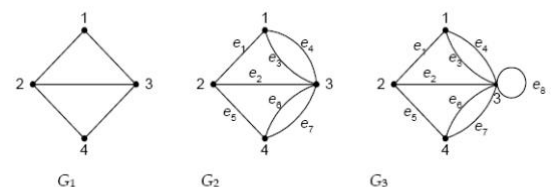
Berdasarkan ada tidaknya gelang atau sisi ganda:

1. Graf Sederhana

Graf yang tidak memiliki gelang dan sisi-ganda (tidak memiliki keduanya).

2. Graf tak-sederhana

Graf yang memiliki gelang atau sisi-ganda (salah satu atau keduanya).



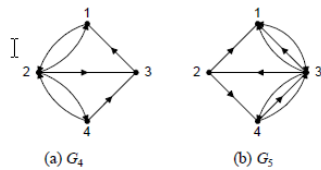
(G₁) graf sederhana, (G₂) multigraf, dan (G₃) multigraf

Gambar 1. Jenis graf berdasarkan sisi ganda

Sumber: <http://athayaniimtinan.blogspot.com/>

Berdasarkan ada tidaknya arah pada sisi:

1. Graf tak-berarah
Graf yang sisi-sisinya tidak memiliki arah.
2. Graf berarah
Graf yang sisi-sisinya memiliki arah.



Gambar 2. Jenis graf berdasarkan arahnya
Sumber: <http://septian-sisinfak.blogspot.com/>

2.1.3 Terminologi Graf

Beberapa terminologi pada graf:

1. Ketetanggaan (*Adjacent*)
Dua buah simpul bertetangga apabila keduanya terhubung secara langsung.
2. Bersisian (*Incidency*)
Dua buah simpul dihubungkan oleh sisi maka sisi tersebut bersisian.
3. Simpul Terpencil (*Isolated Vertex*)
Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya.
4. Graf Kosong (*Null Graph*)
Graf Kosong adalah graf yang himpunan sisinya merupakan himpunan kosong.
5. Derajat (*Degree*)
Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.
6. Lintasan (*Path*)
Lintasan adalah barisan yang berisi nama dari simpul asal kemudian berikutnya dan seterusnya hingga di simpul tujuan. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut.
7. Sirkuit (*Circuit*)
Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.
8. Terhubung (*Connected*)
Dua buah simpul dikatakan terhubung apabila terdapat lintasan diantara keduanya.
9. Upagraf (*Subgraf*) dan Komplemen Upagraf
Graf $G_1 = (V_1, E_1)$ dan graf $G_2 = (V_2, E_2)$ dikatakan upagraf jika V_2 merupakan himpunan bagian dari V_1 dan E_2 merupakan himpunan bagian dari E_1 .

Komplemennya adalah suatu graf G_3 dimana $E_3 = E_1 - E_2$ dan V_3 adalah himpunan simpul yang anggota-anggota E_3 bersisian dengannya.

10. Upagraf Rentang (*Spanning Subgraph*)
Upagraf $G_1 = (V_1, E_1)$ dari $G_2 = (V_2, E_2)$ dikatakan upagraf rentang jika $V_1 = V_2$ (yaitu G_1 mengandung semua simpul dari G_2).
11. *Cut-Set*
Cut-Set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung.
12. Graf Berbobot (*Weighted Graph*)
Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).

2.1.4 Graf Khusus

Beberapa jenis graf khusus:

1. Graf Lengkap (*Complete Graph*)
Graf lengkap adalah graf sederhana yang tiap simpulnya memiliki sisi ke semua simpul lainnya. Jumlah sisi pada graf ini yang terdiri dari n simpul adalah $n(n - 1) / 2$.
2. Graf Lingkaran
Graf Lingkaran adalah graf yang tiap simpulnya berderajat dua.
3. Graf Teratur (*Regular Graph*)
Graf Teratur adalah graf yang tiap simpulnya memiliki derajat yang sama. Jumlah sisi pada graf teratur adalah $nr / 2$.
4. Graf Bipartite (*Bipartite Graph*)
Misalkan graf G yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V_1 dan V_2 , sehingga tiap sisinya menghubungkan sebuah simpul di V_1 ke simpul di V_2 disebut graf bipartite dan dinyatakan sebagai $G(V_1, V_2)$.

2.1.5 Representasi Graf

Graf dapat direpresentasikan dengan beberapa cara, diantaranya:

1. Matriks Ketetanggaan (*adjacency matrix*)
2. Matriks Bersisian (*incidency matrix*)
3. Senarai Ketetanggaan (*adjacency list*)

2.1.6 Graf Isomorfik

Graf dapat dikatakan isomorfik apabila graf tersebut merupakan dua graf sama namun berbeda penggambarannya.

2.1.7 Graf Planar dan Graf Bidang

Graf planar adalah graf yang bisa digambarkan tanpa adanya sisi-sisi yang saling berpotong, sedangkan graf bidang adalah graf planar yang telah

dimodifikasi tanpa adanya sisi yang saling berpotongan.

2.1.8 Teorema Kuratowski

Teorema Kuratowski bertujuan untuk menentukan keplanaran dari suatu graf. Graf G bersifat planar apabila ia tidak mengandung upagraf yang isomorfik dengan salah satu graf Kuratowski atau homeomorfik dengan salah satu dari kedua graf Kuratowski.

2.1.9 Lintasan dan Sirkuit Euler

Lintasan yang melalui setiap sisi dalam graf hanya tepat satu kali. Sedangkan sirkuit Euler adalah sirkuit yang melewati setiap sisi hanya tepat satu kali.

2.1.10 Lintasan dan Sirkuit Hamilton

Lintasan yang melalui setiap simpul dalam graf tepat hanya satu kali. Sedangkan sirkuit Hamilton adalah sirkuit yang melewati setiap simpul tepat hanya satu kali.

2.2 Teori Sistem Rekomendasi [2] [3] [4] [5] [6]

2.2.1 Pengertian Sistem Rekomendasi

Sistem Rekomendasi atau *recommender system* atau *system recommendation* adalah suatu sistem yang memiliki fungsi untuk memprediksi sejumlah *item* atau *data* yang ditujukan untuk pengguna di masa yang akan datang, yang selanjutnya akan disediakan untuk pengguna sebagai rekomendasi paling teratas. Di tengah terlalu banyaknya informasi yang tidak semua dibutuhkan oleh pengguna, pengguna pun terasa hilang kendali untuk menemukan informasi yang diinginkan. Maka lahirlah sistem rekomendasi atau *recommender system* untuk membantu pengguna untuk menemukan informasi atau data-data yang disukai, diinginkan, atau dibutuhkan.

Sistem rekomendasi atau *recommender system* pada umumnya memanfaatkan opini dari pengguna dan *behavior* pengguna untuk mendapatkan hasil rekomendasi yang diinginkan. Biasanya sistem rekomendasi terdapat pada penyedia layanan *e-commerce*. Penerapan dari sistem rekomendasi ini biasanya melakukan prediksi untuk barang, informasi, gambar, artikel, music, film dan lain sebagainya yang sekiranya menarik bagi pengguna. Untuk mendapatkan prediksinya, sistem rekomendasi membutuhkan sejumlah data *history* pengguna.

2.2.2 Metode Pengumpulan Data

Untuk dapat menghasilkan prediksi yang sesuai dengan pengguna diperlukan data *history* untuk diolah dan dicari kesamaannya. Untuk mendapatkan data *history* tersebut dapat dilakukan dengan dua metode, yaitu:

1. Secara Eksplisit atau Langsung

Untuk mendapatkan data agar menghasilkan data prediksi, data dapat ditujukan langsung dan secara terang-terangan ditanyakan kepada

pengguna layanan. Contohnya ialah:

- Meminta pengguna untuk memberikan rating pada sebuah *item*.
- Meminta pengguna untuk mengurutkan *item* (*ranking*) atau daftar favorit dari beberapa atau sebuah *item*.
- Survei dalam layanan atau memberikan beberapa pilihan *item* pada pengguna dan meminta untuk memilih yang terbaik atau yang paling diinginkan.
- Meminta pengguna untuk memisahkan *item* yang paling disukai dan *item* yang paling tidak disukai.

2. Secara Implisit atau Tidak Langsung

Untuk mendapatkan data *history* untuk keperluan prediksi rekomendasi tidak selalu harus didapatkan dengan cara meminta langsung kepada pengguna. Metode pengumpulan data *history* yang lain adalah secara tidak langsung atau implisit. Secara tidak langsung adalah untuk mendapatkan data *history* pengguna, pengguna tidak akan menyadari bahwa ada data yang diambil dari *behavior* pengguna di saat pengguna sedang menggunakan layanan itu sendiri atau layanan lain. Pengumpulan data dengan metode ini dilakukan dengan cara seperti:

- Melihat atau mengamati *item* yang dilihat oleh pengguna.
- Mengumpulkan data transaksi pada sebuah *tool online*.
- Melihat atau mengamati *behavior* pengguna dari layanan lainnya.
- Melihat atau mengamati *history* pencarian *item* atau *tool online*.

2.2.3 Metode *Filtering* pada Sistem Rekomendasi

Setelah mendapatkan data-data *history* yang didapat dari pengguna dengan cara eksplisit, implisit ataupun keduanya, perlu diklasifikasikan kembali untuk memodelkan metode *filtering* yang ingin digunakan untuk menghasilkan prediksi yang sesuai dengan layanan dan pengguna. *Filtering* ini digunakan untuk memisahkan data-data dan diklasifikasikan berdasarkan kategori dari layanan dengan pengguna serta tingkat kemiripan datanya dengan metode tertentu. Pada umumnya metode *Filtering* dapat dibagi menjadi:

1. *Content Based Recommendation*

Pada metode *Content Based Recommendation*, metode ini memanfaatkan informasi berupa *item* ataupun data *history* dari pengguna untuk kembali direkomendasikan kepada pengguna sebagai prediksi dan referensi yang terkait dengan informasi, data, atau *item* yang diinginkan oleh pengguna itu sendiri. Tujuan dari metode *Filtering Content Based Recommendation* itu sendiri adalah untuk memprediksi persamaan-

persamaan dari sejumlah informasi atau data yang didapat dari pengguna secara langsung atau tidak langsung ataupun keduanya.

Salah satu contoh *Content Based Recommendation* adalah seseorang melihat-lihat beberapa *item* dari sebuah toko *online* dalam suatu platform. Layanan yang memiliki sistem rekomendasi akan otomatis memberikan rekomendasi kepada pengguna tentang hal-hal yang sekiranya mirip dengan *item-item* yang ada pada toko *online* tersebut.

Dalam pembuatan *filtering Content Based Recommendation*, pemiliki layanan dapat menggunakan beberapa cara untuk mewujudkannya, diantaranya dengan bantuan:

- a) Konsep Perhitungan *vector*.
- b) Algoritma TF-IDF (*Term Frequency - Inverse Document Frequency*), yang merupakan suatu metode algoritma yang berguna untuk menghitung bobot dari setiap *item* yang umum digunakan. Metode ini menghitung nilai TF (*Term Frequency*) dan nilai IDF (*Inverse Document Frequency*) pada setiap *item* di dalam suatu dokumen. Mudahnya TF-IDF ini menghitung seberapa sering suatu *item* muncul di suatu dokumen. Pada TF, terdapat beberapa jenis rumus yang dapat dipakai, yaitu:
 - TF Biner, hanya memperhatikan ada tidaknya suatu *item* di dalam dokumen, jika ada diberi nilai (1) dan jika tidak ada diberi nilai (0).
 - TF Murni, nilai TF diberikan berdasarkan jumlah kemunculan suatu *item* dalam dokumen.
 - TF Normalisasi, membandingkan frekuensi sebuah *term* dengan nilai maksimum keseluruhan.
 - TF Logaritmik, menghindari dominasi dokumen yang mengandung sedikit *term* dalam *query*, namun memiliki frekuensi yang tinggi.

$$TF = \begin{cases} 1 + \log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases}$$

Gambar 3. Formula *Term Frequency*
Sumber: <https://informatikalogi.com>

Berbeda dengan TF, di dalam IDF, semakin sedikit frekuensi kata muncul dalam dokumen, maka makin besar nilainya. Untuk menghitung nilai IDF dapat menggunakan rumus:

$$IDF_j = \log(D/df_j)$$

Gambar 4. Formula *Inverse Document Frequency*
Sumber: <https://informatikalogi.com>

Dengan demikian rumus umum untuk Term Weighting TF-IDF adalah penggabungan dari formula perhitungan raw TF dengan formula IDF

dengan cara mengalikan nilai TF dengan nilai IDF:

$$w_{ij} = tf_{ij} \times idf_j$$

$$w_{ij} = tf_{ij} \times \log(D/df_j)$$

Gambar 5. Formula *Term Weighting*
Sumber: <https://informatikalogi.com>

Dimana w_{ij} adalah bobot *term* (t_j) terhadap dokumen (d_i). Sedangkan tf_{ij} adalah jumlah kemunculan *term* (t_j) dalam dokumen (d_i). D adalah jumlah semua dokumen yang ada dalam *database* dan df_j adalah jumlah dokumen yang mengandung *term* (t_j) (minimal ada satu kata yaitu *term* (t_j)).

- c) *Cosine Similarity*, yaitu mencari nilai kedekatan dari suatu *item*. Nilai *cosine similarity* didapat dari rumus:

$$Cosine Similarity : Sim(u_i, u_k) = \frac{r_i \cdot r_k}{|r_i| |r_k|} = \frac{\sum_{j=1}^m r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^m r_{ij}^2} \sqrt{\sum_{j=1}^m r_{kj}^2}}$$

Gambar 6. *Cosine Similarity*
Sumber: <https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700>

2. Collaborative Filtering

Pada metode *Collaborative Filtering*, metode ini memanfaatkan transaksi suatu *item* yang didasarkan kepada perilaku atau *behavior* pengguna. Metode ini bertujuan agar pengguna yang sama dan *item* yang mirip dapat disukai oleh pengguna sebagai rekomendasi pilihan. *Collaborative Filtering* pada umumnya dapat digunakan menggunakan dua metode, yaitu:

- a) *User-Based Collaborative Filtering (UB-CF)*, metode ini merekomendasikan *item* dengan menemukan pengguna yang mirip dengan pengguna yang lainnya. Metode ini dapat dikatakan seperti pengguna membantu pengguna lain untuk mendapatkan referensi secara tidak langsung dan tidak disadari. Hal ini seperti saat seseorang merekomendasikan suatu hal kepada temennya dengan asumsi bahwa temannya tersebut akan memiliki selera yang sama. UB-CF ini memiliki 2 metode perhitungan untuk mendapatkan kedekatannya yaitu:
 - *Cosine similarity*
 - *Pearson Correlation*, yang berguna untuk menghitung nilai korelasi linier diantara dua variabel X dan Y. *Pearson correlation* memiliki nilai diantara +1 dan -1, yaitu 1 adalah total linier korelasi positif, 0 adalah tidak ada korelasi linier, dan -1 adalah total linier korelasi negative.

$$\text{Pearson Correlation : } \text{Sim}(u_i, u_k) = \frac{\sum_j (r_{ij} - r_i)(r_{kj} - r_k)}{\sqrt{\sum_j (r_{ij} - r_i)^2 \sum_j (r_{kj} - r_k)^2}}$$

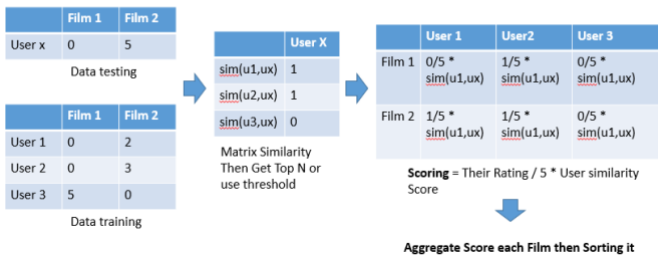
Gambar 7. Formula *Pearson Correlation*

Sumber: [https://medium.com/data-folks-](https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700)

[indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700](https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700)

Kedua metode tersebut dapat digunakan untuk *User-Based Collaborative Filtering*. Perbedaan diantara kedua metode perhitungan tersebut ialah *Pearson correlation* tidak akan berubah saat menambahkan konstanta kepada semua elemen di dalamnya. Pada pembuatannya, umumnya digunakan algoritma *K-Nearest Neighbour* (KNN), yaitu suatu algoritma yang digunakan untuk klasifikasi dan regresi.

User-Based Collaborative Filtering

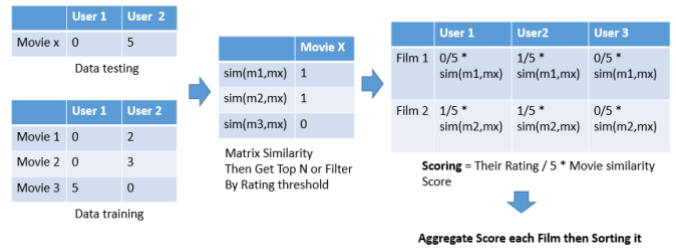


Gambar 8. Skema *User-Based Collaborative Filtering*

Sumber: [https://medium.com/@ranggaantok/bagaimana-](https://medium.com/@ranggaantok/bagaimana-sistem-rekomendasi-berkerja-e749dac64816)

- b) *Item-Based Collaborative Filtering* (IB-CF), metode ini merekomendasikan kesamaan antara *item*, saat seorang pengguna melakukan interaksi dengan pengguna lain dan *item*. Saat pengguna menyukai suatu klub sepakbola, sistem secara otomatis akan merekomendasikan suatu item yang berhubungan dengan klub sepakbola tersebut, seperti bola, poster, *jersey*, dan lain sebagainya yang berhubungan. Pada pembuatannya IB-CF menggunakan metode perhitungan yang sama dengan UB-CF. Yang membedakannya adalah dalam menargetkan suatu rekomendasi yang akan dibuat.

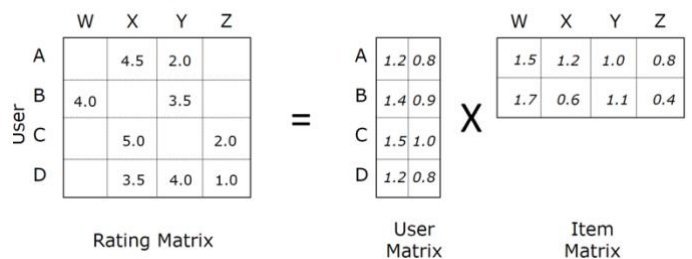
Item-Based Collaborative Filtering



Gambar 9. Skema *Item-Based Collaborative Filtering*

Sumber: [https://medium.com/@ranggaantok/bagaimana-](https://medium.com/@ranggaantok/bagaimana-sistem-rekomendasi-berkerja-e749dac64816)

Selain menggunakan algoritma seperti *K-Nearest Neighbor* (KNN), *collaborative filtering* juga dapat menggunakan metode lain yang sering pula digunakan yaitu algoritma *Matrix Factorization* (MF). *Matrix Factorization* berfungsi untuk memecah atau mendekomposisi interaksi matriks dari *item* dan pengguna menjadi hasil kali dari dua matriks dua dimensi yang lebih rendah. Suatu matriks dapat dilihat sebagai matriks dari pengguna yang di dalamnya terdapat baris yang mewakili pengguna dan kolom mewakili *latent factor*. Matriks selanjutnya adalah matriks *item* yang di dalamnya terdapat baris yang mewakili *latent factor* dan kolom yang mewakili *item*.



Gambar 10. *Matrix Factorization*

Sumber: [https://medium.com/data-folks-](https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700)

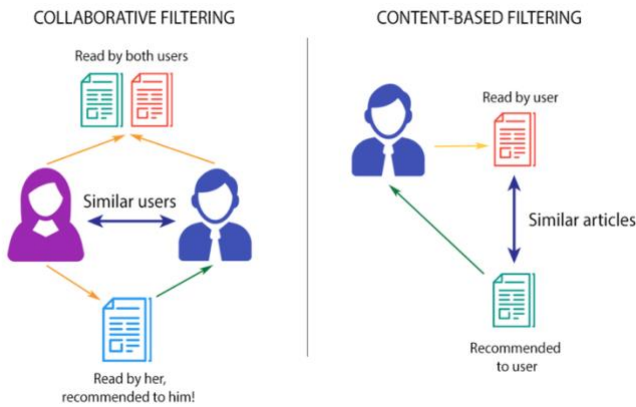
Untuk menentukan nilai *latent factor*, algoritma pada *Matrix Factorization* bisa menggunakan *singular value decomposition* (SVD), yaitu faktorisasi dari matriks ril atau matriks kompleks.

3. Content – Collaborative Recommendation System (Hybrid Recommendation System)

Hybrid Recommendation System adalah gabungan antara metode-metode *filtering* yaitu *content based filtering* dengan *collaborative filtering*. Namun untuk menciptakan *Recommender System* tidak harus terpaku kepada dua metode tersebut. Namun, *Hybrid Recommendation System* dapat dipadukan dengan *recommendation strategy* yang lainnya dengan upaya untuk mendapatkan keuntungan.

III. APLIKASI GRAF PADA SISTEM REKOMENDASI

Sistem Rekomendasi memanfaatkan graf untuk mendapatkan hasil prediksi atau referensi untuk penggunaanya. Graf dapat berfungsi untuk mengetahui tingkat kedekatan atau bobot korelasi antara *item*, pengguna itu sendiri dan pengguna yang lainnya. Dalam pengaplikasian metode *filtering* dapat digunakan graf untuk menemukan korelasi dengan pengguna yang lainnya.



Gambar 11. Graf pada metode *Recommender System*
Sumber: <http://datameetsmedia.com/an-overview-of-recommendation-systems/>

Pada *Collaborative Filtering*, simpul pengguna ada yang terhubung dengan *item* yang sama. Dengan demikian semakin banyak kesamaan *item* yang disukai oleh kedua pengguna, bobot kesamaan dari kedua pengguna semakin meningkat. Dengan kesamaan tersebut dapat diprediksi kedua pengguna memiliki selera yang mirip. Semakin tinggi tingkat kesamaan dari kedua pengguna, semakin banyak pula rekomendasi atau referensi dari pengguna untuk pengguna lainnya yang diprediksi memiliki kesamaan selera.

Pada *Content-Based Filtering*, simpul pengguna terhubung dengan suatu *item*, dan *item* tersebut memiliki tingkat kemiripan yang tinggi dengan *item*. Karena kedua *item* memiliki kemiripan, diprediksi bahwa pengguna memiliki ketertarikan akan suatu *item* yang mirip, maka *item* yang diprediksi mirip akan direkomendasikan kepada pengguna.

Sedangkan pada *Hybrid Recommendation System* yang menggabungkan keduanya, pengguna akan mendapat rekomendasi berdasarkan kolaborasi antara metode *Collaborative Filtering*, dan *Content-Based Filtering*. Sehingga rekomendasi yang akan ditawarkan kepada pengguna akan jauh lebih baik dan presisi.

IV. KESIMPULAN

Teori graf banyak diterapkan di dalam bidang informatika. Salah satu pengaplikasiannya ialah pada Sistem Rekomendasi atau *Recommender System* yang merupakan cabang dari Sistem Penyaringan Informasi. Penerapan graf pada Sistem Rekomendasi adalah pengguna, atau *item* sebagai simpulnya dan korelasi atau interaksi antara pengguna dengan *item*, antara pengguna dengan pengguna, antara *item* dengan *item* terhubung

sebagai sisi dengan sisi yang memiliki bobot berdasarkan frekuensi interaksi antar simpul yang didapatkan dari metode perhitungan seperti *cosine similarity*. Fungsi dari Sistem Rekomendasi adalah menghasilkan referensi dan rekomendasi untuk pengguna agar pengguna nyaman menggunakan suatu layanan karena kemudahan dalam mendapatkan *item* atau data yang dicari. Metode yang digunakan untuk menemukan kemiripan data *history* pengguna diantaranya adalah *Content-Based Recommendation*, *Collaborative Filtering*, dan *Hybrid Recommendation System* yang masing-masing memiliki cara yang berbeda untuk menemukan hasil rekomendasi atau referensi untuk pengguna dengan menggunakan teori graf. Dengan semakin berkembangnya tren berbelanja *online* diharapkan Sistem Rekomendasi dapat membantu seluruh pengguna menemukan data atau informasi yang diinginkan.

V. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan terima kasih dan rasa syukur kepada Tuhan Yang Maha Esa karena oleh anugerah dan karunia-Nya penulis dapat menyelesaikan makalah ini tepat waktu. Penulis mengucapkan terima kasih kepada dosen IF2120 yaitu Pak Rinaldi Munir dan Ibu Fariska Zakhralativa Ruskanda karena tanpa ilmu yang diberikan melalui kelas penulis tidak dapat menyelesaikan makalah ini dengan baik. Serta penulis mengucapkan terima kasih kepada semua teman-teman seperjuangan yang telah membantu penulis untuk menyelesaikan makalah ini. Penulis pun tidak lupa mengucapkan terima kasih kepada semua pembaca makalah ini dan semoga tulisan ini dapat bermanfaat bagi para pembacanya.

REFERENSI

- [1] [informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).ppt](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).ppt).
- [2] <https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-definisi-part-1-71154dc3f700>.
- [3] <https://medium.com/@ranggaantok/bagaimana-sistem-rekomendasi-berkerja-e749dac64816>.
- [4] <https://realpython.com/build-recommendation-engine-collaborative-filtering/>.
- [5] <https://www.twoh.co/2013/05/17/sekilas-tentang-sistem-rekomendasi-recommender-system/2/>.
- [6] <https://medium.com/@dltsierra/algoritma-tf-idf-633e17d10a80>.

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 Desember 2019



Hafshy Yazid Albisthami
13518051