

Analisis Komputasi *Trust-Based* pada metode *Collaborative Filtering* untuk Sistem Rekomendasi Hotel

Azka Apta Afrizal¹, Ibnu Asror², Yusza Reditya³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹knightazka@students.telkomuniversity.ac.id, ²iasror@telkomuniversity.ac.id, ³yuszaa@telkomuniversity.ac.id,

Abstrak

Hotel adalah salah satu dari banyaknya jenis tempat tinggal yang ada hampir dalam setiap kota. Beberapa orang biasanya akan mencari hotel tergantung dari kebutuhan dan juga melihat ulasan dari orang-orang sebelumnya. Hotel dan juga ulasan yang terlalu banyak, terkadang akan membingungkan pencari hotel untuk menemukan hotel yang pas bagi mereka. Dari sinilah dibangun sebuah sistem yang akan mempermudah pencari hotel untuk menemukan hotel yang pas. Sistem ini adalah sebuah sistem rekomendasi. Dimana sistem akan merekomendasikan sebuah *item* yaitu hotel. Hotel yang akan direkomendasikan berdasarkan beberapa kriteria tertentu dan salah satunya yang terpenting yaitu *rating* dan ulasan *user*. Sistem rekomendasi yang dibangun di dalam tugas akhir ini menggunakan metode *Collaborative Filtering* (CF) dan dibantu dengan perhitungan komputasi *Trust-Based*. CF bekerja dengan cara melakukan perhitungan terhadap *rating* dari pengguna ataupun ulasan sebelumnya dan *trust* akan menghitung keaslian nilai yang diberikan suatu *user* terhadap sebuah *item*. Sistem akan merekomendasikan hotel berdasarkan karakteristik *user* dan nilai dari kesamaan (*similarity*) antar *user* lain yang sudah pernah melakukan *review* hotel dengan yang akan direkomendasi. Dari percobaan yang telah dilakukan, didapatkan nilai error sebesar 1.6924 dengan menggunakan metode CF, 0.8746 menggunakan metode CF ditambah *Trust-Based* dengan nilai *threshold trust* >3 dan 0.8716 dengan nilai *threshold trust* >5. Semakin kecil nilai error, maka semakin bagus rekomendasi bekerja. Dilihat dari pengujian yang sudah dijalankan, penggunaan *trust* dapat mengurangi besarnya error pada penggunaan metode *collaborative filtering*.

Kata kunci : sistem rekomendasi, *review*, *collaborative filtering*, *Trust-Based*

Abstract

Hotels are the one of the many types of residences that exist in almost every city. Some people usually will search for hotels depending on their needs and also see reviews from the previous people. Hotels and review that too many will make hotel seekers confuse to find the right hotel for them. From here appointed a system that will facilitate hotel seekers to find the right hotel. This system is called recommendation system. Where the system will recommend an item which in this context is a hotel. Hotels to be recommended are based on certain criteria and one of the most important is user reviews. Recommendation system that was built using the Collaborative Filtering (CF) method. CF works by calculating the rating from users or previous reviews and trust will calculate the authenticity of the user value given by a user to an item. The system will recommend hotels based on the value of the similarity of other users who have rated one of the hotels and who are looking for hotel. From the experiment that has been done, there is an error value 1.6924 obtained using the CF method, 0.8746 with CF methods plus Trust-Based with trust threshold value >3 and 0.8716 with trust threshold value >5. The smaller error value, recommendation will have a better works. From the tests that has been done, the use of trust can reduce error in the use of collaborative filtering methods.

Keywords: recommendation system, *review*, *collaborative filtering*, *Trust-Based*

1. Pendahuluan

Latar Belakang

Era sekarang ini, internet sudah menjadi sumber informasi dari hal-hal yang ada di dunia. Jumlah informasi yang didapatkan dari internet sangatlah berlimpah sehingga terkadang kita sulit untuk menyaring informasi yang banyak terkait dengan informasi yang dibutuhkan. Salah satu sistem yang saat ini sedang berkembang adalah sistem rekomendasi. Sistem rekomendasi akan mempermudah *user* untuk melakukan pencarian berdasarkan dengan hal yang dibutuhkan. Oleh karena itu, dengan adanya sistem rekomendasi, diharapkan kedepannya *user* tidak akan kesulitan untuk menyaring data ataupun mencari data

terkait tentang yang mereka inginkan. Ada beberapa opsi untuk membangun sistem rekomendasi yaitu menggunakan metode *collaborative filtering*, *content-based* dan juga *hybrid*.

Collaborative filtering (CF) adalah salah satu metode yang biasa digunakan untuk pembuatan sebuah sistem rekomendasi. CF secara umum akan memprediksi utilitas *item* untuk *user* tertentu berdasarkan *item* yang sebelumnya sudah dinilai oleh orang ataupun *user* lain [1]. Sebuah sistem rekomendasi yang menggunakan metode CF akan mengandalkan nilai atau *rating* dari *user* sebelumnya dan juga melihat kesamaan karakteristik antar *user* untuk mendapatkan rekomendasi. CF akan mencoba mencari kesamaan hal yang disukai oleh *user* dimasa kini dengan *user* dimasa lalu.

CF mempunyai kekurangan untuk beberapa konteks data tertentu, seperti, hanya sebagian kecil dari *user* yang menilai lebih dari 1 hotel yang berbeda. Kurangnya peringkat ataupun *rating* dari *user* sebelumnya akan mempersulit sistem untuk menemukan jumlah *user* yang serupa dan membuat prediksi *item* yang akurat untuk *user* yang akan direkomendasi [2]. Selain itu ada juga konteks masalah lain yaitu dari penilaian *user*. Ada sebuah kondisi dimana *user* memberikan nilai yang asal atau tidak sesuai terhadap *item* yang sudah dinilai oleh mereka. Sesuai ataupun tidaknya nilai yang diberikan dari suatu *user* ini nanti akan dilihat dengan perbandingan penilaian semua *user* yang lain yang juga sudah menilai *item* yang sama. Pemberian nilai yang asal ini juga akan membuat sistem sulit untuk merekomendasikan *item* yang tepat untuk *user* selanjutnya.

Dari sinilah diangkat sebuah perhitungan bernama komputasi *Trust-Based* untuk mengurangi kekurangan dari CF. *Trust* akan memperhitungkan kepercayaan dari *user*. *Trust* akan dibentuk dalam sebuah perhitungan dimana *trust* akan membandingkan keaslian nilai yang diberikan oleh suatu *user* terhadap *item* yang sudah pernah dinilai. Nilai yang diberikan oleh *user* akan dibandingkan oleh banyak *user* yang juga sudah menilai ataupun memberikan *review* terhadap *item*. Semakin besar nilai *trust* dari suatu *user*, maka semakin besar pula kemungkinan *user* itu akan dijadikan acuan untuk rekomendasi bagi *user* selanjutnya. Penggunaan *trust* ini sendiri juga dipercaya dapat menghasilkan nilai *error* prediksi *item* yang lebih kecil daripada hanya menggunakan metode CF.

Penelitian ini akan mengangkat pembahasan sebuah komputasi model dengan *Trust-Based* untuk menutupi kekurangan dari *collaborative filtering*. Maka dari itu, tugas akhir ini akan membahas bagaimana membangun sebuah sistem rekomendasi dengan menggunakan metode *collaborative filtering* dan juga akan memasukkan sebuah perhitungan menggunakan komputasi *Trust-Based* yang nantinya akan dilihat apakah pembuatan sistem rekomendasi dengan menggunakan *collaborative filtering* ditambah dengan perhitungan *Trust-Based* dapat mengurangi nilai *error* terhadap *item* yang akan direkomendasi untuk suatu *user*.

Topik dan Batasannya

Dalam penelitian ini akan dibahas bagaimana membangun sebuah sistem rekomendasi hotel dengan menggunakan metode *collaborative filtering* ditambah dengan perhitungan *Trust-Based*. Dimana *Trust-Based* ini sendiri dipercaya dapat menghasilkan prediksi *item* yang lebih baik untuk *user* daripada hanya sekedar menggunakan metode *collaborative filtering* itu sendiri. Selain itu akan dilihat perbandingan performansi apakah penggunaan *Trust-Based* akan menghasilkan nilai *error* yang lebih kecil terhadap prediksi *item* untuk *user* daripada hanya sekedar menggunakan *collaborative filtering* saja.

Data yang akan diolah yaitu sebanyak 700 data hotel dan 1300 data *user review* yang didalamnya terdapat beberapa atribut informasi tentang hotel dan penilaian yang diberikan oleh *user* terhadap suatu hotel. Data hotel yang akan direkomendasikan hanyalah untuk region Bandung. Semua dataset yang digunakan adalah hasil *crawling* data dari www.traveloka.com. Batasan pembahasan dari pembuatan tugas akhir ini tidak membahas dan juga membandingkan pembangunan sistem rekomendasi dengan metode lain, hanya berfokus dengan membangun sistem rekomendasi menggunakan metode *collaborative filtering* dan juga *Trust-Based*.

Tujuan

Tujuan dari pembuatan tugas akhir ini yaitu :

1. Membangun sistem rekomendasi hotel dengan menggunakan metode *collaborative filtering* disertai dengan *Trust-Based*.
2. Menganalisis perbandingan performansi apakah penggunaan metode CF ditambah dengan *Trust-Based* akan menutupi kekurangan dari metode CF.

Organisasi Tulisan

Selanjutnya pada bab 2 akan dibahas mengenai studi terkait dengan penelitian yang dilakukan. Bab 3 akan dibahas tentang perancangan sistem yang dibuat. Bab 4 akan membahas evaluasi dari model penelitian dan yang terakhir bab 5 akan membahas kesimpulan.

2. Studi Terkait

2.1 Sistem Rekomendasi

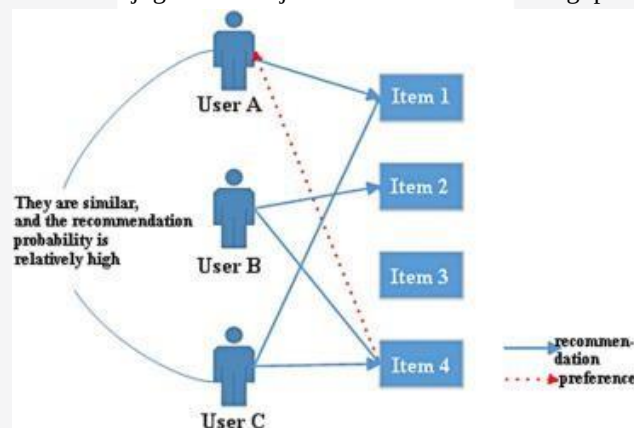
Banyak informasi yang datang silih berganti setiap waktunya tentang hal-hal baru yang ada di belahan dunia ini. Terkait tentang banyaknya informasi yang banyak ini, terkadang akan didapatkan informasi yang benar-benar dibutuhkan maupun yang tidak di perlukan sama sekali. Sistem rekomendasi ini hadir untuk para *user* ataupun pencari yang sedang bingung dengan banyaknya informasi yang tidak tersaring.

Sistem rekomendasi memiliki sejarah yang semakin meningkat popularitasnya karena mereka mempersonalisasi pengalaman *user* dan membantu *user* menemukan konten yang relevan[3]. Sistem rekomendasi akan menyarankan *user* beberapa *item* yang dapat memenuhi kebutuhan dan keinginan mereka[4]. Sistem rekomendasi akan membantu *user* yang memiliki kesulitan untuk menyaring informasi yang banyak terkait tentang konten yang sedang mereka cari. Selain itu, tujuan dari sistem rekomendasi ini sendiri adalah untuk mengurangi informasi yang terlalu banyak dari berbagai sumber dan konten yang berbeda. Sistem rekomendasi mempunyai beberapa metode untuk menghasilkan rekomendasi seperti *collaborative filtering*, *content based* dan juga *hybrid*. Didalam penulisan tugas akhir ini akan difokuskan untuk membangun sebuah sistem rekomendasi dengan menggunakan metode *collaborative filtering*. *Collaborative filtering* adalah sebuah metode yang merekomendasikan *item* yang disukai *user* lain yang sejenis dengan menghitung nilai kesamaan (*similarity*) antar *user*.

2.2 Collaborative Filtering

Collaborative filtering adalah salah satu metode yang biasa digunakan pada sistem rekomendasi. *Collaborative filtering* dalam sistem rekomendasi berfokus untuk memprediksi *item* baru yang menarik untuk *user* berdasarkan korelasi antara *user* lain dengan *user* itu sendiri[5]. Sistem rekomendasi dengan menggunakan metode *collaborative filtering* memiliki berbagai prospek aplikasi di berbagai bidang seperti *e-commerce* dan jejaring sosial[6]. *Collaborative filtering* akan membuat sebuah perhitungan dimana *user* akan dikumpulkan dan dikaitkan karena memiliki keterkaitan yang sama.

Collaborative filtering akan memprediksi pilihan *user* berdasarkan interaksi *item* dengan *user* dalam jumlah yang besar[7]. Kualitas dari rekomendasi dengan metode *collaborative filtering* sangat bergantung dari opini *user* lain terhadap sebuah *item*. Seorang *user* yang mungkin sebelumnya menyukai sebuah konten atau *item*, kemungkinan *item* atau konten itu juga bisa menjadi sebuah ketertarikan bagi pencari selanjutnya.



Gambar 1 Contoh Collaborative Filtering[8]

Collaborative filtering akan memungkinkan *user* baru yang memiliki ketertarikan tertentu untuk mendapatkan sebuah rekomendasi tentang minat dan kesukaan *user* dari *user* yang telah memberikan *rating* ataupun nilai sebelumnya. Dengan menggunakan nilai acuan sebelumnya, dapat dibangun sebuah model produk baru yang mirip ataupun serupa untuk *user* baru. *Collaborative filtering* akan bekerja dengan cara menghitung nilai kesamaan/*similarity* antar *user* untuk dapat menghasilkan sebuah rekomendasi. *Collaborative filtering* akan menggunakan sebuah matriks dua dimensi dari *user* x *item* dan selanjutnya akan menggunakan matriks antar *user* atau *user* x *user*.

2.3 Trust-Based

Trust atau kepercayaan bisa didefinisikan dalam berbagai macam konteks. Bart Noteboom mengatakan bahwa *trust* adalah suatu ekspektasi dari seseorang yang hasilnya tidak akan mengecewakan[9]. *Trust* dalam sistem rekomendasi tidak mempunyai definisi ataupun arti khusus. Penggunaan *trust* untuk memberikan rekomendasi telah muncul sebagai cara baru untuk memberikan rekomendasi yang lebih baik. Dalam sebuah rujukan sumber ditunjukkan bahwa rekomendasi dengan menggunakan pendekatan *trust* berkinerja lebih baik daripada hanya didasarkan pada kesamaan *user* ataupun kesamaan *item*[10]. *Trust* adalah sebuah metode yang akan menghasilkan sebuah kepercayaan dengan melihat nilai penggabungan antara *user* dan juga *item*[11].

Trust digunakan untuk melihat apakah nilai yang diberikan oleh suatu *user* terhadap masing-masing *item* yang sudah di-*rating* adalah nilai yang benar dan bisa diterima. *Trust* akan menghitung jarak antara nilai rata-rata *rating user* dan total rata-rata *rating* yang diberikan oleh *user* lain terhadap suatu *item*. Oleh karena itu, *trust* dapat didefinisikan seperti yang ditunjukkan dalam persamaan (1) dimana r_i adalah nilai rata-rata *rating* yang diberikan suatu *user* terhadap semua *item* yang sudah dinilainya dan t_i untuk total nilai rata-rata *rating* keseluruhan dari semua *user* terhadap suatu *item*.

$$Trust_i = \frac{1}{|r_i - t_i|} \quad (1).$$

Dalam persamaan dibalik perhitungan *trust* ini, *user* akan dikatakan *trustworthy* apabila nilai rata-rata *rating* asli mendekati nilai dari total *rating*. $Trust_i$ akan dialokasikan sebagai nilai yang lebih besar dibandingkan dengan *user* lain. Nilai dari total *rating* t_i akan didapatkan dengan melihat total *rating* yang sudah dinilai oleh banyak *user* terhadap suatu *item*. Nilai total itu akan dianggap sebagai nilai yang lebih jujur karena melihat rata-rata *rating* dari keseluruhan *user*. Untuk mendapatkan nilai t_i dapat menggunakan persamaan (2) dimana I_i adalah jumlah *user* yang menilai suatu *item* dan t_j adalah total nilai yang diberikan *user* terhadap item j . Sistem akan mempertimbangkan *user* dan *item* yang diklaim sebagai *trustworthy* sebagai nilai lebih untuk rekomendasi bagi *user* selanjutnya.

$$t_i = \frac{1}{|I_i|} \sum_{j \in I_i} t_j \quad (2).$$

2.4 Cosine Similarity

Cosine similarity akan di implementasikan setelah didapatkan nilai *trust*. *Cosine similarity* akan mencari *user* yang mempunyai pola karakteristik ataupun kemiripan satu sama lain. *Cosine similarity* digunakan untuk melihat seberapa besar nilai kesamaan (*similarity*) ataupun korelasi antar satu *user* dengan *user* lainnya. Nilai dari *cosine similarity* dapat diambil dengan menggunakan persamaan berikut:

$$Sim(A,B) = \frac{A \cdot B}{|\sqrt{A}| \cdot |\sqrt{B}|} \quad (3).$$

Himpunan data A berisikan data nilai *trust* dari *user* yang akan direkomendasi, dan himpunan data B akan berisikan nilai *trust* dari *user* lain. Sistem akan menyilangkan nilai *trust* dari *user* yang akan direkomendasi dengan *user* lain dengan rumus persamaan 3. Sistem akan melakukan pengulangan rumus sampai didapatkan semua nilai *similarity* dari masing-masing *user*.

2.5 Score Function

Score function adalah sebuah fungsi yang digunakan untuk melihat prediksi *item* yang akan direkomendasikan untuk *user*. Setelah mendapatkan *user* yang serupa dengan menggunakan *cosine similarity* selanjutnya akan dilihat berapakah *rating* yang mungkin akan diberikan oleh *user* yang direkomendasi apabila *user* diminta untuk menilai hotel yang sama dan sudah dinilai oleh *user* lainnya yang serupa. Pengambilan skor untuk masing-masing *item* yang dinilai oleh *user* akan didapatkan dengan menggunakan persamaan berikut:

$$Score = average\ rating + \frac{sum\ score}{total\ correlation} \quad (4).$$

Dimana, *Average rating* adalah nilai rata-rata dari semua *rating* yang sudah diberikan oleh *user* terhadap *item*. *Sum score* adalah jumlah hotel yang sudah dinilai oleh *user*. Dan yang terakhir *total correlation*, nilai dari total korelasi diisi dengan nilai korelasi *user* dengan *user* lain.

2.6 Mean Absolute Error

Mean absolute error (MAE) adalah perhitungan yang digunakan untuk menghitung seberapa besar rata-rata kesalahan ataupun *error* yang bersifat absolut antara nilai prediksi dan nilai sebenarnya [12]. MAE akan menghasilkan rata-rata besarnya *error* dari semua data yang ada. MAE akan bekerja dengan cara menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai sebenarnya. Tiap *user* memiliki nilai *error* nya masing-masing terhadap *item* yang direkomendasi. Dari penggunaan MAE ini juga dapat dilihat seberapa besarkah *error* ataupun ketepatan rekomendasi *item* untuk *user* yang direkomendasi. MAE dapat dihasilkan dengan menggunakan persamaan berikut:

$$MAE = \frac{1}{n} \sum |y - \hat{y}| \quad (5).$$

Keterangan:

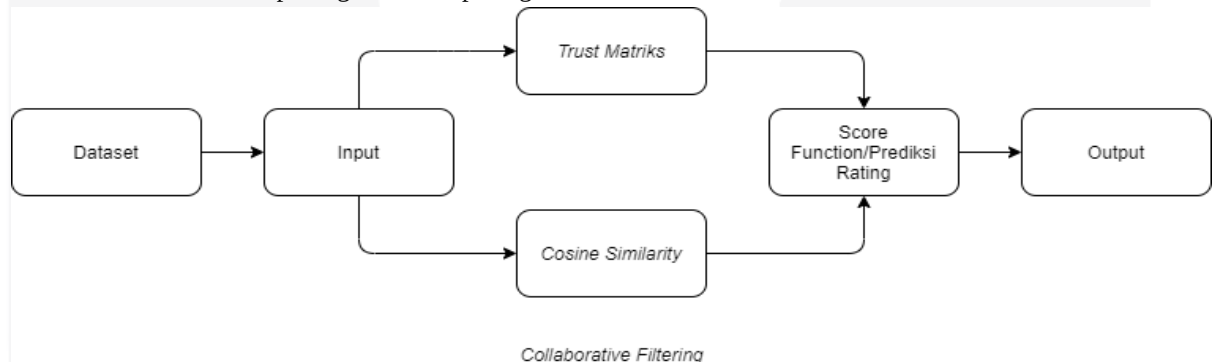
MAE : nilai rata-rata kesalahan hitungan
 y : nilai sebenarnya yang diberikan *user* lain
 $\hat{y}$: nilai prediksi dari *user* yang direkomendasi
 n : jumlah data prediksi

Nilai prediksi didapatkan dari prediksi skor yang dibahas pada pembahasan score function. Sedangkan nilai sebenarnya (Actual score) nilai asli yang didapatkan dengan melihat *rating* yang sudah diberikan oleh *user* lain terhadap *item*.

3. Sistem yang Dibangun

3.1 Perancangan Sistem

Sistem yang dibangun adalah sebuah sistem yang akan merekomendasikan *user* beberapa daftar hotel yang sekiranya memiliki kesamaan dan karakteristik yang paling mendekati dengan *user*. Perancangan sistem untuk sistem rekomendasi hotel dapat digambarkan pada gambar 2.



Gambar 2 Perancangan Sistem

3.2 Crawling Dataset

Dataset untuk penelitian ini diambil dari sebuah website yang menyediakan fitur seperti ulasan dan juga *rating* dari banyak *user*. Semua informasi dataset didapatkan dari sebuah website hotel ternama yaitu traveloka.com. Penelitian ini akan menggunakan 2 buah jenis data yaitu dataset hotel dan juga dataset *user review*. Dari hasil *crawling* hotel beserta informasinya, didapatkan sebanyak kurang lebih 700 hotel dan 1300 *review*. Atribut yang terkait didalam dataset seperti nama hotel, hotelid, lokasi, *rating*, userid dan username.

Tabel 1 Contoh Dataset Hotel

hotelId	hotelName	location	region
1	El Hotel Royale Bandung	Merdeka	Bandung
2	Grand Tjokro Bandung	Cihampelas	Bandung
3	Horison Ultima Bandung	Lengkong	Bandung
4	Best Western Premier La Grande Hotel	Merdeka	Bandung
5	The Trans Luxury Hotel	Buah Batu	Bandung

Tabel 2 Contoh Dataset user

userId	userName	hotelId	rating	timestamp
1	A Cfast	31	8	26 Des 2019
2	Aan S.P	102	8	6 Jan 2020
3	Abdan s	708	6	4 Jan 2020
4	Abdhy G	133	9	4 Jan 2020
5	Abdul H	104	10	30 Des 2019

3.3 Penggunaan Trust-Based

Setelah semua dataset telah terkumpul, selanjutnya adalah mencari nilai *trust* dari masing-masing *user* terhadap *item* yang ada. Nilai dari *trust* masing-masing *user* bisa didapatkan dengan menggunakan persamaan (1). Berikut adalah contoh beberapa *user* yang telah didapatkan nilai *trust*-nya.

Tabel 3 Contoh perhitungan Trust

userId	userName	hotelId	rating	r_i	t_i	trust
1	A Cfast	31	8	7.4	8	1.666667
3	Abdan s.	708	6	8.6	6	0.384615
4	Abdhy G	133	9.4	8.6	9.4	1.25
5	Abdul H	104	10	7.9	10	0.47619
6	Andul R	67	10	7.5	10	0.4
7	Abdurrahman	91	9	7.9	9	0.909091
525	Herman S	50	5	4	5.5	0.666667
525	Herman S	76	6	3.15	5.5	0.425532

3.4 Cosine Similarity

Setelah melakukan perhitungan *trust*, tahap selanjutnya adalah mencari *user* yang mempunyai karakteristik ataupun kemiripan satu sama lain. *Cosine similarity* digunakan untuk melihat nilai kesamaan (*similarity*) ataupun korelasi antar satu *user* dengan *user* lainnya. Sebelumnya akan dibangun sebuah data matriks *user* yang berisikan nilai *trust* masing-masing *user* terhadap masing-masing *item*. Ada beberapa kondisi di dalam matriks ini yaitu:

1. *User* yang sudah pernah menilai suatu hotel nilai *trust* nya akan dimasukkan kedalam matriks.
2. NaN (Not a Number) adalah kondisi apabila *user* tidak melakukan penilaian terhadap hotel yang tertera.

Tabel 4 Matriks Trust User

Matriks Trust User											
hotelId/userId	1	2	3	4	5	6	711	712	713	714	715
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.28169	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN	0.526316	NaN	NaN	NaN	NaN	NaN

Nilai NaN seperti kondisi diatas diartikan dengan *user* tidak melakukan penilaian terhadap *item* yang ada, sedangkan untuk proses menuju ke tahap mencari nilai *similarity* dibutuhkan semua nilai *trust user* terhadap masing-masing *item*. Agar data tetap dapat diolah, nilai *trust* dari *user* yang tidak menilai *item* yang ada akan digantikan dengan nilai rata-rata dari keseluruhan *user* yang telah menilai *item* tersebut.

Tabel 5 Trust Tanpa Nilai NaN

Matriks Trust User											
hotelId/userId	1	2	3	4	5	6	711	712	713	714	715
1	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28
2	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28
3	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28
4	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28
15	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	0.281 69	1.430 775	3.580 011	2.871 28
16	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	4.388 011	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28
17	2.429 864	2.567 275	1.323 947	1.272 016	2.195 15	0.526 316	1.759 52	4.377 559	1.430 775	3.580 011	2.871 28

Setelah nilai *trust* masing-masing *user* terhadap *item* didapatkan dan dimasukkan kedalam matriks, selanjutnya adalah memasukkan nilai *trust* yang sudah didapat kedalam persamaan rumus (3) *cosine similarity* untuk mencari kemiripan atau nilai korelasi antar *user*. *User* yang akan direkomendasi akan dibandingkan *similarity* nya dengan semua *user* yang ada. *Similarity* mempunyai rentang nilai dari 0-1, dimana *similarity* yang bernilai mendekati 0 akan bernilai *low correlation* dan 1 akan bernilai *high correlation*. Contoh hasil *similarity* dapat dilihat pada tabel berikut.

Tabel 6 Hasil Similarity User-User

Similarity User-User						
userId/userId	1	2	3	1362	1363	1364
1	1.000000	0.999709	0.999676	0.997905	0.999709	0.999668
2	0.999709	1.000000	0.999968	0.998188	1.000000	0.999979
3	0.999676	0.999968	1.000000	0.998157	0.999967	0.999947
4	0.999606	0.999898	0.999865	0.998089	0.999898	0.999877
5	0.999708	0.999999	0.999966	0.998186	0.999998	0.999978

Pada tabel 6 ditunjukkan bahwa sudah didapatkan nilai kesamaan (*similarity*) antaruser. Nilai ini didapatkan dengan menggunakan rumus *cosine similarity*. Dari nilai *similarity* ini dapat diketahui seberapa besar nilai korelasi antar *user*. Setelah mengetahui nilai korelasi antar *user*, selanjutnya akan di urutkan *user* yang memiliki tingkat korelasi dengan *user* yang akan direkomendasi dari nilai tertinggi hingga terendah.

Setelah didapatkan daftar *user* yang memiliki korelasi, sistem akan mengecek apa saja daftar *item* yang sudah di *rating* oleh *user* tersebut. Seperti konsepnya *collaborative filtering* yaitu menghasilkan rekomendasi berdasarkan nilai dari kesamaan *user*, nilai kesamaan dan korelasi *user* dari yang tertinggi akan dilihat dan dicek. Apabila *user* yang direkomendasi belum pernah menilai *item* yang sudah pernah dinilai oleh *user* lain yang memiliki korelasi paling tinggi dengan yang direkomendasi, maka *item* itu akan menjadi rekomendasi bagi *user* selanjutnya. Akan dilakukan pengecekan terhadap peringkat *item* yang akan direkomendasi bagi *user*. Peringkat *item* ini diberikan berdasarkan nilai dari prediksi skor *item* dari yang tertinggi hingga yang terendah yang akan dibahas pada sub-bab selanjutnya.

3.5 Score Function

Setelah mendapatkan nilai korelasi dari hubungan antara *user*, selanjutnya adalah melihat prediksi *item* yang akan direkomendasikan untuk *user*. *Item* yang akan direkomendasikan untuk *user* dilihat dari nilai prediksi *rating* yang sekiranya akan diberikan untuk *user* apabila semisalnya *user* diminta untuk menilai hotel yang belum pernah dinilai atau di *rating* oleh *user* sebelumnya tapi sudah pernah di *rating* oleh *user* lain.

Semua hotel yang sudah di *rating* oleh *user* lain yang dilihat dari nilai korelasi paling besar dengan *user* (dilihat dari hasil yang sudah didapatkan dengan melihat kedekatan antar *user* pada pembahasan *cosine similarity*) akan dibuat sebuah prediksi berapakah *rating* yang mungkin akan diberikan oleh *user* apabila *user* diminta untuk menilai *item* yang sudah dinilai oleh *user* lain. *Rating* prediksi yang akan diberikan oleh *user* yang belum pernah menilai hotel itu disebut dengan skor.

Pembobotan skor nilai ini digunakan untuk melihat daftar rekomendasi *item* untuk *user*. Setelah *user* mempunyai nilai skor terhadap masing-masing *item* yang ada, lalu akan dibuatkan kandidat peringkat *item* rekomendasi untuk *user*. *Item* yang memiliki nilai skor yang paling besar akan ditempatkan sebagai rekomendasi yang pertama, begitu pun selanjutnya. Untuk mendapatkan skor dari masing-masing *item* dapat menggunakan rumus persamaan (4).

Seperti yang sudah dijelaskan sebelumnya, skor digunakan untuk membuat sebuah prediksi berapa kira-kira *rating* yang akan diberikan oleh *user* terhadap suatu *item* yang belum pernah dia *rating* tetapi sudah pernah di *rating* oleh *user* lain yang memiliki korelasi paling besar dengan *user*. *Rating* yang akan diprediksi adalah *rating* dari *item* yang belum pernah dinilai oleh *user* sebelumnya. Setelah bobot skor *rating* nilai untuk setiap *item* sudah didapatkan, selanjutnya *item* itu akan diurutkan dari yang terbesar hingga terkecil. *Item* yang memiliki skor paling besar akan menjadi rekomendasi yang paling baik untuk *user* begitu pun seterusnya.

4. Evaluasi

Setelah sistem selesai dibuat, maka tahap selanjutnya adalah pengujian terhadap sistem. Pengujian ini bertujuan untuk mengetahui seberapa baik sistem bekerja dan juga memastikan bahwa hasil yang dikeluarkan sesuai dengan kebutuhan dan tujuan yang sudah dibahas sebelumnya serta melihat perbandingan dengan menambahkan sebuah pengujian metode baru terhadap *collaborative filtering*. Pengujian yang dilakukan adalah pengujian sistem rekomendasi dengan menggunakan metode *collaborative filtering* serta penggunaan *collaborative filtering* dipadukan dengan komputasi *Trust-Based*.

4.1 Skenario 1 Sistem Rekomendasi Hotel dengan Collaborative Filtering

Pada pengujian ini akan dilakukan percobaan untuk menghasilkan rekomendasi dengan hanya menggunakan metode *collaborative filtering*. Pengujian ini dilakukan untuk melihat seberapa banyak error yang dihasilkan

untuk rekomendasi dengan metode *collaborative filtering*. Cara yang diambil untuk melihat banyaknya *error* dari sistem rekomendasi yang dibahas dalam tugas akhir ini yaitu dengan menggunakan persamaan MAE (*Mean Absolute Error*)(5).

Pada perhitungan MAE, semakin kecil nilai MAE maka prediksi *item* untuk *user* akan semakin tepat, begitu pun sebaliknya. Oleh karena itu, dengan menggunakan rumus *Mean Absolute Error* pada metode *collaborative filtering* maka diperoleh MAE sebesar **1.692400452365117**. Perhitungan MAE diambil dari rata-rata 1300 *user*.

Tabel 7 Contoh MAE User dengan metode CF

userId	MAE	userId	MAE
1	0.069043642	11	0.740149
2	0.740149	12	0.74453
3	0.73443	13	0.772329
4	0.845884	14	0.934925
5	0.740149	15	0.15905623
6	0.772329	16	0.740149
7	0.40149	17	0.77344
8	0.740149	18	0.772329
9	0.772329	19	0.740149
10	0.740149	20	0.740149

Pada uji coba dengan hanya menggunakan menggunakan metode *collaborative filtering* menghasilkan nilai rata-rata MAE yang cukup besar. Nilai MAE yang cukup besar ini dikarenakan sistem sulit untuk mencari *user* yang mempunyai pola yang sama dikarenakan minimnya *user* menilai *item*. Akibatnya rekomendasi *item* dengan melihat skor nilai yang diprediksi dengan nilai *rating* asli yang didapatkan dengan melihat nilai yang diberikan semua *user* terhadap suatu *item* berbanding jauh. Apabila nilai rentang MAE terlampaui jauh, maka dapat disimpulkan bahwa terdapat sedikit kesalahan terkait dengan *item* yang direkomendasikan untuk *user*.

4.2 Skenario 2 Sistem Rekomendasi Hotel dengan *Collaborative filtering* dan *Trust-Based* dengan *threshold trust* >3

Pada pengujian ini akan dilakukan pengujian untuk menghasilkan rekomendasi dengan metode *collaborative filtering* dan juga ditambah dengan komputasi *Trust-Based* dengan nilai *threshold trust* >3. Disini rekomendasi yang dihasilkan hanya dilihat dari *user* yang memiliki nilai *trust* diatas 3. Nilai dari *user* dengan rentang nilai *trust* diatas 3 akan dikumpulkan untuk dilihat apakah penggunaan *trust* pada metode *collaborative filtering* dapat mengecilkan nilai *error* pada rekomendasi. Dengan menggunakan rumus yang sama yaitu MAE (*Mean Absolute Error*) maka, sistem rekomendasi dengan menggunakan metode *collaborative filtering* dan juga digabung dengan perhitungan *Trust-Based* dengan *threshold trust* >3 menghasilkan nilai rata-rata *error* sebesar **0.8746169677134139**.

Tabel 8 Contoh MAE User dengan trust threshold >3

userId	Trust	MAE
1006	3.278689	0.7265836589
1350	3.333333	0.75967746698
628	3.389831	0.7487676496
1084	3.703704	0.827596774
801	3.846154	0.846758271

Apabila dibandingkan dengan nilai *error* yang dihasilkan dengan hanya menggunakan metode *collaborative filtering*, dapat dilihat bahwa rata-rata nilai MAE dengan penggunaan *trust* dapat menghasilkan nilai *error* yang

lebih kecil daripada hanya menggunakan metode *collaborative filtering*. Ini dikarenakan sistem telah mengumpulkan *user* yang hanya dinilai *trustworthy* atau layak dijadikan sebagai acuan untuk rekomendasi bagi *user* selanjutnya.

Sistem akan mencari nilai korelasi hanya dari *user* yang bisa dikatakan *trustworthy*. Sistem akan mempertimbangkan semua *user* yang dapat dipercaya untuk dijadikan rekomendasi bagi *user* selanjutnya. Penggunaan *trust* pada metode *collaborative filtering* juga berguna untuk menyaring *user* agar didapatkan korelasi yang sama walaupun *user* hanya menilai sebagian kecil dari *item* yang ada.

4.3 Skenario 3 Sistem Rekomendasi Hotel dengan *Collaborative filtering* dan *Trust-Based* dengan *threshold trust* >5

Pada pengujian ini akan dilakukan pengujian untuk menghasilkan rekomendasi dengan metode *collaborative filtering* dan juga ditambah dengan komputasi *Trust-Based* dengan nilai *threshold* >5. Masih sama dengan sebelumnya yaitu mencari berapa besarnya nilai error dengan MAE dari penggunaan metode *collaborative filtering* ditambah dengan *Trust-Based* dengan *threshold* >5. *Trust-Based* dengan menggunakan *threshold trust* >5 menghasilkan nilai rata-rata *error* sebesar **0.8716576776977597**.

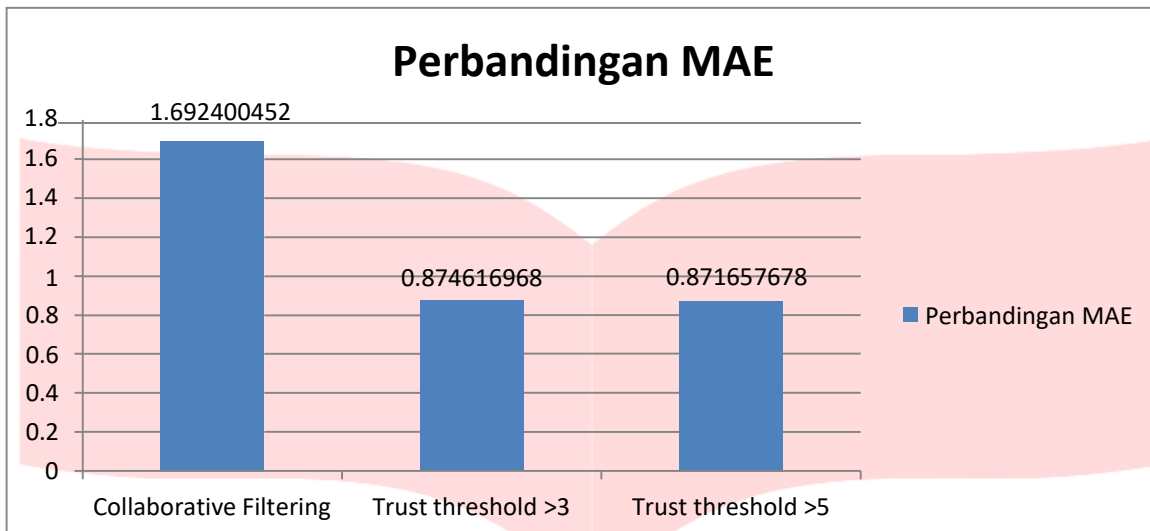
Tabel 9 Contoh MAE User dengan *trust threshold* >5

UserId	Trust	MAE
881	5.128205128	0.856171561659
1258	5.555555556	0.816158166428
145	5.882352941	0.71546285676
1087	6.52173913	0.81514756726
1364	7.240424	1.26181576

Jika dilihat dari 3 skenario yang telah dijalankan, berdasarkan perhitungan dengan melihat nilai *error* dengan perhitungan MAE, dapat dilihat penggunaan *Trust* pada metode *collaborative filtering* menghasilkan hasil yang lebih baik daripada hanya sekedar penggunaan *collaborative filtering* itu sendiri untuk sebuah sistem rekomendasi jika dilihat dari perhitungan MAE.

MAE tidak bisa disamakan dengan melihat akurasi, apabila semakin kecil *error* nya berarti akurasi nya semakin besar. MAE yang dihasilkan adalah rentang atau jarak dari hasil prediksi yang dibuat untuk *user* yang akan direkomendasi apabila sekiranya *user* diminta untuk menilai *item* yang serupa dengan *user* yang lainnya. Semakin kecil selisih nilai prediksi dengan nilai asli yang diberikan oleh *user* lain maka, akan semakin tepat rekomendasi yang diberikan untuk *user*.

Nilai MAE yang besar tidak bisa dikatakan sebagai rekomendasi yang buruk. Hanya saja, MAE digunakan untuk melihat seberapa besarnya kesalahan rekomendasi *item* yang didapatkan dari rekomendasi yang dihasilkan jika dibandingkan dengan data asli. Hasil rekomendasi dengan menggunakan *Trust-Based* menghasilkan rekomendasi yang lebih spesifik. Spesifik disini dikarenakan sistem hanya mengumpulkan *user* dengan nilai *trust* tertentu berdasarkan *threshold* yang di gunakan. *Trust-Based* sangat berperan untuk mencari *user* mana yang dapat dijadikan acuan rekomendasi bagi *user* selanjutnya dengan melihat nilai *trustworthy* yang dimilikinya.



Gambar 3 Perbandingan MAE

Sebaliknya, saat membangun sebuah sistem rekomendasi dengan hanya menggunakan nilai kesamaan dan karakteristik *user* atau menggunakan metode CF menghasilkan nilai MAE yang lebih besar, ini dikarenakan sistem akan kesulitan untuk menentukan *user* mana yang akan dijadikan acuan untuk rekomendasi karena *user* lain pun hanya menilai sebagian *item* yang tersedia. Dengan kata lain, sistem tidak mendapatkan *user* yang memiliki *similarity* yang lumayan dekat dengan *user* yang akan direkomendasikan. Atau juga sebagai opsi sistem akan memberikan rekomendasi hanya berdasarkan nilai korelasi terdekat dengan *user* tanpa mempertimbangkan apakah satu *user* dengan *user* lain mempunyai pola atau ketertarikan yang sama.

5. Kesimpulan dan Saran

Penulisan tugas akhir ini berfokus pada beberapa aspek. Pertama, yaitu penulis mengumpulkan data asli yang berisikan informasi *user review* dan juga *rating* hotel dari sebuah website ternama yaitu traveloka.com. Kedua, diangkat sebuah metode *collaborative filtering* untuk membangun sebuah sistem rekomendasi dan juga ditambahkan perhitungan *cosine similarity* untuk menghitung nilai kesamaan antar-*user* dan juga fungsi skor untuk menghitung nilai prediksi *rating* dari *item* yang ada di dataset, tidak lupa pula ditambahkan juga perhitungan Trust-Based untuk melihat perbandingan apakah hasil dari pembangunan sistem rekomendasi dengan menggunakan metode *collaborative filtering* akan menutupi kekurangan yang ada pada metode *collaborative filtering*.

Dari pengujian yang sudah dijalankan, didapatkan hasil seperti yang ada pada bab 4. Penambahan perhitungan Trust-Based pada metode *collaborative filtering* menghasilkan hasil MAE ataupun error yang lebih kecil daripada hanya menggunakan metode *collaborative filtering*. Konsep dari Trust-Based ini sendiri dimana *user* akan semakin bernilai *trustworthy* atau menjadi nilai lebih untuk menjadi acuan bagi *user* lain jika nilai yang diberikan *user* terhadap suatu *item* itu benar adanya dan bukan nilai asal. Nilai yang diberikan oleh *user* itu akan dibandingkan dengan suatu set dari penilaian *rating* yang diberikan oleh *user* lain terhadap suatu *item*.

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Hasil rekomendasi hotel dengan menggunakan metode CF ditambah dengan Trust-Based dapat mengurangi kekurangan dari metode CF itu sendiri.
2. Penggunaan Trust-Based bisa digunakan untuk kondisi apabila hanya sebagian *user* yang menilai lebih dari satu *item* yang tersedia. Dengan bantuan Trust-Based, sistem tetap dapat mencari nilai korelasi antar *user* berdasarkan nilai kepercayaan dari *rating* yang diberikan oleh satu *user* terhadap *item* yang sudah dinilai olehnya.

Saran yang dapat dipertimbangkan untuk penelitian selanjutnya adalah menambahkan lagi dataset yang ada sehingga sistem akan lebih mempunyai banyak pembandingan untuk menghasilkan ragam rekomendasi yang lebih baik.

daftar pustaka

- [1] J. Zhou and T. Luo, "Towards an introduction to collaborative filtering," *Proc. - 12th IEEE Int. Conf. Comput. Sci. Eng. CSE 2009*, vol. 4, pp. 576–581, 2009.
- [2] Q. Wu, A. Forsman, Z. Yu, and W. W. Song, "A Computational Model for Trust-Based Collaborative Filtering An Empirical Study on Hotel Recommendations," pp. 266–279, 2014.
- [3] A. N. Maus and A. K. Atwood, "Surveying Older Adults About a Recommender System for a Digital Library," pp. 41–44, 2015.
- [4] S. Parvatikar and B. Joshi, "Online book recommendation system by using collaborative filtering and association mining," *2015 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2015*, pp. 6–9, 2016.
- [5] N. Mittal, R. Nayak, M. C. Govil, and K. C. Jain, "Recommender system framework using clustering and collaborative filtering," *Proc. - 3rd Int. Conf. Emerg. Trends Eng. Technol. ICETET 2010*, pp. 555–558, 2010.
- [6] D. Zeng and H. Chen, "for E-commerce."
- [7] H. Li, X. Diao, J. Cao, and Q. Zheng, "Collaborative Filtering Recommendation Based on All-Weighted Matrix Factorization and Fast Optimization," *IEEE Access*, vol. 6, no. c, pp. 25248–25260, 2018.
- [8] J. Li *et al.*, "Category Preferred Canopy–K-means based Collaborative Filtering algorithm," *Futur. Gener. Comput. Syst.*, vol. 93, pp. 1046–1054, 2019.
- [9] S. Publications, "Review Author (s): Peter Smith Ring Review by : Peter Smith Ring Source : Administrative Science Quarterly , Vol . 48 , No . 4 (Dec . , 2003), pp . 720-722 Published by : Sage Publications , Inc . on behalf of the Johnson Graduate School of Management," vol. 48, no. 4, pp. 720–722, 2016.
- [10] M. G. Ozsoy and F. Polat, "Trust based recommendation systems," *Proc. 2013 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2013*, pp. 1267–1274, 2013.
- [11] N. Lathia, S. Hailes, and L. Capra, "Trust-based collaborative filtering Trust-Based Collaborative Filtering," no. July, pp. 2650–2654, 2014.
- [12] S. Cleger-Tamayo, J. M. Fernández-Luna, and J. F. Huete, "On the use of weighted mean absolute error in recommender systems," *CEUR Workshop Proc.*, vol. 910, pp. 24–26, 2012.