

Diversity Balancing pada Two-stage Collaborative Filtering untuk Mengatur Less Popular Item dalam Domain Film

1st Fajar Widhi Ardiyanto

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

fajarwidhi@student.telkomuniversity.ac.id

2nd Dade Nurjanah

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

dadnurjanah@telkomuniversity.ac.id

3rd Selly Meliana

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

sellym@telkomuniversity.ac.id

Abstrak—Sistem pemberi rekomendasi film adalah sistem yang bertanggung jawab untuk memberikan rekomendasi kepada penonton film tentang judul film lain yang relevan, menarik, dan mungkin disukai oleh penonton. Salah satu metode yang paling sering digunakan pada sistem pemberi rekomendasi adalah *Collaborative Filtering* (CF). CF unggul dalam tingkat *accuracy* terhadap *item* yang direkomendasikan kepada pengguna meskipun dalam keadaan konten yang sulit dianalisis sekalipun. Kelemahannya, CF hanya merekomendasikan *item* yang populer saja. Maka karena itu, dibutuhkan sistem pemberi rekomendasi yang dapat memberikan rekomendasi *item* yang kurang populer dengan tingkat *accuracy* yang masih dapat diterima. Tugas akhir ini mengusulkan sistem pemberi rekomendasi film berbasis *Two-stages CF* dengan menggunakan metode *Diversity Balancing*. Sistem ini bekerja dengan memprediksi *rating* pengguna kemudian menghasilkan daftar kandidat untuk selanjutnya dilakukan *Diversity Balancing*. *Diversity Balancing* dilakukan dengan cara mengganti *item* yang populer dengan *item* yang kurang populer namun relevan. Hasil pengujian dengan metode *precision*, *recall*, *personal diversity* dan *aggregate diversity* menunjukkan nilai tertinggi masing-masing adalah 0.8000, 0.5366, 0.5301, 0.3709. Berdasarkan hasil tersebut didapatkan keseimbangan antara *accuracy* dan *diversity* di parameter $k = 7$ dan *threshold rating* sebesar 3.0.

Kata kunci—sistem pemberi rekomendasi, *collaborative filtering*, *diversity*, *less popular item*

Abstract—A movie recommendation system is responsible for providing recommendations to moviegoers about other movie titles that are relevant, interesting, and maybe liked by the audience. One of the most frequently used methods in recommender systems is *Collaborative Filtering* (CF). CF excels in the level of accuracy for items recommended to users, even in situations where the content is challenging to analyze. The downside is that CF only recommends popular items. Therefore, a recommendation system that can provide recommendations for less popular items with an acceptable level of accuracy is needed. This final project proposes a Two-stages based CF movie recommendation system using the *Diversity Balancing* method. This system works by predicting user ratings, followed by generating a list of candidates to be utilized for *Diversity Balancing*. *Diversity Balancing* is done by replacing popular items with less popular ones yet still relevant. The test results with *precision*, *recall*, *personal diversity*, and *aggregate diversity* methods showed that the highest values are 0.8000, 0.5366, 0.5301, and 0.3709, respectively. These results

show a balance between accuracy and diversity at parameter $k = 7$ and a threshold rating of 3.0.

Keywords—recommender systems, collaborative filtering, diversity, less popular item

I. PENDAHULUAN

A. Latar Belakang

Sistem pemberi rekomendasi adalah sistem yang bertanggung jawab untuk memberikan sebuah informasi atau konten untuk pengguna (*user*) berdasarkan penyaringan informasi dari komunitas atau *user* itu sendiri [1]. Sistem tersebut digunakan untuk memberikan rekomendasi kepada penonton film tentang judul film lain yang mungkin disukai oleh penonton. Sistem pemberi rekomendasi yang banyak dipakai saat ini adalah sistem yang didesain memberikan rekomendasi yang relevan dan menarik terhadap *user*. Sehingga sistem tersebut seringkali hanya merekomendasikan *item* yang populer saja. Kondisi ini menyebabkan *item* yang masih relevan namun kurang mendapatkan popularitas gagal untuk muncul ke dalam rekomendasi. Maka dari itu dibutuhkan sistem pemberi rekomendasi yang dapat memberikan rekomendasi *item* yang kurang populer dengan tingkat *accuracy* yang masih dapat diterima [2].

Collaborative Filtering (CF) merupakan sistem pemberi rekomendasi yang paling umum digunakan [3]. CF bekerja dengan memprediksi *item* untuk *user* tertentu berdasarkan *item* yang telah diberi *rating* sebelumnya oleh *user* lain [4]. Hal tersebut menimbulkan beberapa kekurangan dari sistem CF. Salah satu kekurangan dari CF adalah *cold start*. *Cold start* adalah masalah yang berkaitan dengan jarangya informasi yang tersedia dalam sistem pemberi rekomendasi [5]. Solusi umum dari permasalahan tersebut adalah dengan memberikan tambahan informasi seperti relasi sosial *user* atau konteks informasi [2].

Masalah lain yang muncul dalam sistem pemberi rekomendasi adalah *Long Tail Recommendation Problem* (LTRP). LTRP muncul karena sistem pemberi rekomendasi cenderung tidak merekomendasikan *item* yang tidak banyak diberi *rating*, *item* baru, atau jarang dicari oleh *user* [6]. Sebagian besar sistem pemberi rekomendasi mengabaikan *item* yang kurang populer atau

item baru dengan *rating* yang masih sedikit dan berfokus pada *item* yang memiliki *rating* cukup untuk direkomendasikan [7]. *Long tail* meliputi *item* yang sebenarnya bagus untuk direkomendasikan, tetapi seringkali algoritma sistem pemberi rekomendasi gagal untuk memasukkan *item* tersebut. Sedangkan *distant tail* adalah *item* yang menerima sangat sedikit *rating* yang berguna, sehingga *item* tersebut menjadi kurang dapat diandalkan untuk direkomendasikan [8].

Oleh karena itu, pada penelitian ini diusulkan sistem pemberi rekomendasi film berbasis *Two-stages Collaborative Filtering* dengan menggunakan metode *Diversity Balancing* [2]. *Two-stages Collaborative Filtering* adalah metode CF yang terdiri dari dua tahap. Tahap pertama yaitu pembentukan daftar kandidat *item* yang direkomendasikan kepada *user*. Tahap kedua adalah penerapan *Diversity Balancing* dengan mengganti *item* yang memiliki *rating* tertinggi dengan *item* diluar daftar kandidat yang memiliki kemiripan paling sedikit dari *item* tersebut. Metode *Two-stages Collaborative Filtering* dipilih karena satu tahap pembentukan rekomendasi dinilai hanya akan menghasilkan rekomendasi yang berisi *item* yang populer saja. Maka dari itu dibutuhkan tahap selanjutnya yaitu penerapan *Diversity Balancing*. Metode *Diversity Balancing* dengan penggantian *item* populer dipilih karena metode tersebut dapat meningkatkan nilai *diversity* tanpa mengorbankan terlalu banyak *accuracy*. Sehingga rekomendasi yang diberikan dapat memuat *item* yang kurang populer namun masih relevan dengan *user*. Dataset yang digunakan adalah dataset *rating* dan *trust* dari FilmTrust¹.

B. Topik dan Batasannya

Topik permasalahan yang dibahas pada penelitian ini adalah bagaimana nilai *accuracy* dan *diversity* dari sistem pemberi rekomendasi berbasis *Two-stages Collaborative Filtering* dengan menggunakan metode *Diversity Balancing* dalam merekomendasikan judul film kepada *user*. Nilai *accuracy* diukur berdasarkan nilai *precision* dan *recall*. Sedangkan nilai *diversity* diukur dengan nilai *personal diversity* (PD) dan nilai *aggregate diversity* (AD). Adapun batasan-batasan dari penelitian ini adalah data yang diolah adalah dataset *ratings* dan *trust* dari dataset FilmTrust. Dikarenakan keterbatasan waktu pengerjaan, hanya akan dipakai nilai *rating* dari 60 *user* pertama dari dataset *ratings*.

C. Tujuan

Tujuan dari penelitian tugas ini adalah untuk menghasilkan sistem pemberi rekomendasi yang dapat memberikan rekomendasi terhadap *item* yang kurang populer namun masih relevan. Selain itu dilakukan analisa nilai *accuracy* dan *diversity* dari sistem pemberi rekomendasi *Two-stages Collaborative Filtering* dengan *Diversity Balancing* dalam mengatasi masalah LTRP atau *less popular item*.

D. Organisasi Tulisan

¹<https://guoguibing.github.io/librec/datasets.html>

Penelitian ini disusun dengan urutan sebagai berikut: Bab I merupakan pendahuluan penelitian ini. Bab II merupakan studi terkait penelitian. Bab III membahas metode yang digunakan dalam penelitian ini. Bab IV membahas tentang evaluasi performa dari sistem yang diusulkan. Bab V merupakan kesimpulan dan *future work*.

II. KAJIAN TEORI

Beberapa studi telah dilakukan untuk mengatasi masalah *less popular item*. Penelitian [6] berusaha meningkatkan *personal diversity* dengan gabungan metode *content-based* dan *collaborative filtering*. Penelitian tersebut mengukur nilai *accuracy* dengan *precision* dan nilai *diversity* dengan *personal diversity*. Dalam domain film, penelitian [9] merancang sistem pemberi rekomendasi menggunakan metode *Latent Dirichlet Allocation* (LDA) *based* untuk merekomendasikan *item* yang kurang populer. Sedangkan penelitian [10] menggunakan metode *matrix completion framework* untuk menyeimbangkan nilai *diversity* dan *accuracy*. Penelitian [2] menggunakan metode *diversity balancing* dan mendapatkan hasil *accuracy* sebesar 0.35 untuk *precision*, 0.51 untuk *recall*, 0.59 untuk nilai *personal diversity*, dan 0.43 untuk nilai *aggregate diversity*.

A. Collaborative Filtering

Collaborative Filtering merupakan metode dalam sistem pemberi rekomendasi yang sampai saat ini banyak diteliti sejak pertama kali ditemukan [1]. *Collaborative Filtering* memprediksi *rating* atau nilai dari *item* terhadap seorang *user* tertentu berdasarkan *item* yang sebelumnya sudah diberi *rating* atau nilai dari *user* yang lain [4].

Sistem pemberi rekomendasi *Collaborative Filtering* dapat dibagi ke dalam 2 kelas: *memory-based* dan *model-based* [11]. *Memory-based* menggunakan data yang sebelumnya sudah dikumpulkan untuk memprediksi *rating* berdasarkan aturan tertentu. *Memory-based* mengoperasikan semua data dalam *database* untuk membuat prediksi [11][12].

Untuk mencari *similarity*, *Memory-based CF* menggunakan dua pendekatan: *User-Similarity Based Approach* dan *Item-Similarity Based Approach*. Setelah dicari nilai *similarity* antara *item* atau *user* kemudian digunakan untuk memprediksi *rating* dari target *item* terhadap target *user* [13].

Pendekatan dengan *model-based CF* menggunakan algoritma *machine learning* untuk membangun model dataset *user-item rating* dan merekomendasikannya kepada *user* [13]. *Model-based CF* menggunakan data *rating* murni untuk memperkirakan atau mempelajari sebuah model untuk membuat prediksi [14].

Dalam penelitian [2], proses prediksi *rating* dan pembuatan *list* rekomendasi untuk *user* menggunakan *collaborative filtering* dilakukan dengan menghitung *user similarity* terlebih dahulu. Nilai *user similarity* antara *user* *u* dan *v* didapatkan dari persamaan *trust-based similarity* sebagai berikut:

$$TS_{u,v} = \begin{cases} \frac{2 \times \text{sim}(u,v) \times T_{u,v}}{\text{sim}(u,v) \times T_{u,v} + T_{u,v}} & \text{sim}(u,v) + T_{u,v} \neq 0 \text{ dan } \text{sim}(u,v) \times T_{u,v} \neq 0 \\ T_{u,v} & \text{sim}(u,v) = 0 \text{ dan } T_{u,v} \neq 0 \\ \text{sim}(u,v) & \text{sim}(u,v) \neq 0 \text{ dan } T_{u,v} = 0 \end{cases} \quad (1)$$

Dengan $T_{u,v}$ merupakan relasi kepercayaan antara $user$ u dan v yang dihitung dengan persamaan:

$$T_{u,v} = \frac{H - N_{u,v} + 1}{H} \quad (2)$$

H menunjukkan jarak maksimum penyebaran kepercayaan antara $user$. $N_{u,v}$ menunjukkan jarak penyebaran kepercayaan antara $user$ u dan v . Sedangkan untuk $\text{sim}(u,v)$ yang merupakan nilai *similarity* antara $user$ u dan v dapat dihitung dengan persamaan sebagai berikut:

$$\text{sim}(u,v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (R_{v,i} - \bar{R}_v)^2}} \times \omega_u \quad (3)$$

$R_{u,i}$ dan $R_{v,i}$ merupakan *rating* dari $user$ u dan v terhadap *item* i . Sedangkan $\bar{R}_u$ dan $\bar{R}_v$ merupakan rata-rata *rating* yang diberikan $user$ u dan $user$ v . $I_{u,v}$ merupakan set *item* yang diberi *rating* oleh kedua $user$ u dan v . ω_u merupakan *mutual rating factor* yang didapatkan dari persamaan $\omega_u = |I_{u,v}| / |I_u|$. $|I_{u,v}|$ merupakan *mutual item* yang diberi *rating* oleh kedua $user$ u dan v . $|I_u|$ merupakan *item* yang diberi *rating* oleh $user$ u .

Setelah itu, berdasarkan nilai *trust-based similarity* $TS_{u,v}$, $user$ dapat dibagi ke dalam *high-trust based* ($TS_{u,v} \geq \theta$) atau *low-trust based* ($TS_{u,v} < \theta$) berdasarkan nilai

$$ID_{u,c} = \frac{M_{u,c}}{M_u} \text{ and } \bar{R}'_{u,c} = \frac{R_{u,c}}{R_{max}} \quad (4)$$

Dimana $ID_{u,c}$ menunjukkan *interest distribution* $user$ u terhadap *cluster item* c . $M_{u,c}$ menunjukkan jumlah *cluster item* c yang diberi *rating* oleh $user$ u dan M_u menunjukkan jumlah semua *item* yang diberi *rating* oleh $user$ u . $\bar{R}'_{u,c}$ merepresentasikan preferensi *rating* untuk

trust threshold (θ). *Trust threshold* (θ) adalah sebuah nilai yang ditentukan untuk membagi $user$ ke dalam kategori *high-trust based* dan *low-trust based*. *High-trust based* cenderung untuk percaya dengan rekomendasi dari $user$ lain meskipun memiliki preferensi yang berbeda [2]. Dalam penelitian ini nilai θ di-set bernilai 0.4.

Nilai *rating* dari $user$ dan jumlah *item* yang diberi *rating* digunakan untuk menghitung *user interest distribution*. *Interest distribution* dan rata-rata *rating* $user$ u terhadap *cluster item* c dapat dihitung dengan persamaan berikut:

kategori c . $R_{u,c}$ merepresentasikan rata-rata *rating* yang diberikan $user$ u terhadap *item* c dan R_{max} merepresentasikan *rating* tertinggi yang diberikan $user$ u untuk setiap *item*. Kemudian untuk $user$ u , preferensi untuk c dihitung dengan:

$$\text{pref}_{u,c} = \frac{2 \times ID_{u,c} \times \bar{R}'_{u,c}}{ID_{u,c} \times \bar{R}'_{u,c}} \quad (5)$$

$\text{pref}_{u,c} = \{\text{pref}_{u,c1}, \text{pref}_{u,c2}, \dots, \text{pref}_{u,ck}\}$ merupakan preferensi $user$ u untuk semua kategori. $user$ u hanya akan masuk ke dalam penghitungan *interest weight*

hanya jika preferensi $user$ u lebih besar daripada $user$ v . *Mutual preference* antara $user$ u dan v dihitung dengan persamaan:

$$\text{pref}_{u,v} = \frac{\sum_{c=c}^C |\text{pref}_{v,c} \geq \text{pref}_{u,c}|}{|\text{pref}_u|} \quad (6)$$

Dalam proses penyebaran kepercayaan, penghitungan *similarity weight* antara $user$ yang masuk kedalam kelompok *low-trust based* harus mempertimbangkan tidak hanya hubungan mereka, tetapi juga ketertarikan yang

sama antara kedua $user$ [2]. Maka dari itu dua *threshold* harus dimasukkan kedalam penghitungan. *Aggregate weight* dapat didapatkan dari persamaan:

$$W_{u,v} = \begin{cases} TS_{u,v} & TS_{u,v} \geq \theta \\ \gamma TS_{u,v} + (1 + \gamma) \text{pref}_{u,v} & 0 < TS_{u,v} < \theta \\ 0 & \text{else} \end{cases} \quad (7)$$

Dimana *interest threshold* (γ) merupakan pembatas untuk $user$ yang akan diikuti ke dalam penghitungan *weight*. Hal ini diakibatkan karena jika kedua $users$ merupakan *low-trust based*, mereka mempunyai sebagian kecil area minat yang beririsan [2]. Dalam hal ini γ di-set bernilai 0.4.

Setelah menghitung *aggregate weight*, lima $user$ teratas yang memiliki *weight* terbesar dipilih sebagai *based* yang berpartisipasi dalam prediksi *rating*. *Rating* diprediksi dengan:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{v \in U_u} W(u,v) \times (R_{v,i} - \bar{R}_v)}{\sum_{v \in U_u} W(u,v)} \quad (8)$$

Dimana $P_{u,i}$ merupakan prediksi *rating* dari $user$ u terhadap *item* i dan $W(u,v)$ merupakan *aggregate weight* antara $user$ u dan v . Berdasarkan prediksi *rating* $user$ aktif,

maka sepuluh *item* dengan *rating* paling tinggi akan masuk ke daftar kandidat untuk direkomendasikan.

B. Diversity

Diversity dalam sistem pemberi rekomendasi adalah ketidakmiripan diantara daftar *item* yang direkomendasikan [6]. *Diversity* dibagi ke dalam dua kriteria: *Individual Diversity* (ID) dan *Aggregate Diversity* (AD). ID mengacu kepada luasnya atau beragamnya *item* yang direkomendasikan kepada *user*. Sedangkan AD adalah luasnya rekomendasi yang diberikan antar *user* dan AD lebih melihat masalah dari sudut pandang sistem [10].

C. Diversity Balancing

Accuracy prediksi *rating* adalah salah satu metode pengukuran yang paling banyak digunakan untuk menentukan efektifitas sistem pemberi rekomendasi. Desain sistem pemberi rekomendasi yang ada saat ini hanya berfokus pada meningkatkan *accuracy* pada rekomendasi, sehingga hanya merekomendasikan *item* yang mirip dengan yang sudah diberi *rating* sebelumnya [10].

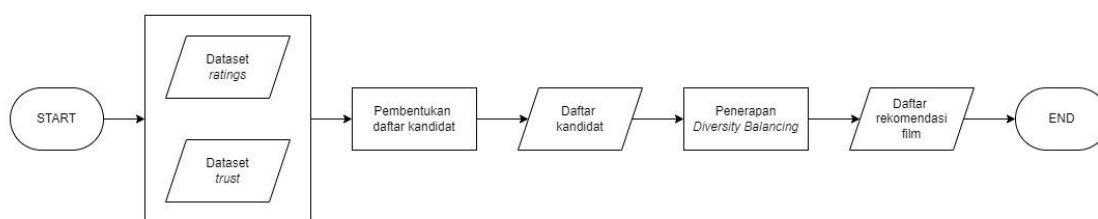
Sedangkan menurut [2] sistem pemberi rekomendasi yang efektif menuntut untuk tidak hanya memiliki *accuracy* yang tinggi, tetapi juga harus memiliki *diversity* yang tinggi. Saat *diversity* naik maka *accuracy* prediksi akan turun, maka pertukaran yang optimal antara *diversity* dan *accuracy* perlu ditentukan [2]. Fokus yang bersamaan terhadap *accuracy* dan *diversity* dapat meningkatkan

kontribusi *item* yang kurang populer dalam rekomendasi sehingga masalah LTRP dapat teratasi [6]. Dengan demikian *diversity balancing* dapat disimpulkan yaitu metode untuk mengatasi LTRP dengan menyeimbangkan antara *accuracy* dan *diversity* dalam suatu sistem pemberi rekomendasi.

Ada beberapa cara dalam menerapkan *diversity balancing*, seperti pada penelitian [6] *diversity balancing* diterapkan dengan menghitung popularitas *item* dan *entropy* dari *user*. Proses *diversity balancing* dilakukan dengan menggunakan *entropy* dari *user* untuk mendapatkan daftar rekomendasi yang non-dominated untuk kemudian dihitung kembali dengan mempertimbangkan popularitas *item*. Hasil dari penghitungan tersebut menjadi rekomendasi final untuk *user*. Sedangkan dalam penelitian [10] menggunakan *rating* yang tersedia dan metadata dari *item* yang ada untuk meningkatkan *diversity* dari sistem pemberi rekomendasi.

III. METODE

Alur kerja dari sistem pemberi rekomendasi *Two-stage Collaborative Filtering* dengan *Diversity Balancing* digambarkan pada diagram alir sebagai berikut:



GAMBAR 1
ALUR KERJA SISTEM

A. Dataset

Pada penelitian ini menggunakan dataset FilmTrust dari Guo et al. FilmTrust terdiri dari dua dataset, yaitu dataset *ratings* yang berisi *rating* dari *user* terhadap film dan dataset *trust* yang berisi nilai kepercayaan antar-*user*. Dataset *ratings* berisi 35497 baris data dengan 3 kolom *feature*. *Rating* diberikan oleh *user* terhadap *item* dengan rentang 1.0 – 5.0. Sedangkan dataset *trust* berisi 1853 baris

data dengan 3 kolom *feature*. Trust disini didasarkan pada kepercayaan yang diberikan oleh *user* kepada *user* lain. Nilai trust antar *user* diberikan dengan nilai 1 sebagai percaya dan 0 sebagai tidak percaya. Dataset ini tersedia secara gratis dan dapat diunduh bebas untuk penelitian. Tabel 1 menampilkan contoh dari dataset *ratings*. Sedangkan Tabel 2 menampilkan contoh dari dataset *trust*.

TABEL 1
CONTOH DATA RATINGS

userid	itemid	rating
1	1	2.0
1	2	4.0
1	3	3.5

TABEL 2
CONTOH DATA TRUST

trustor	trustee	value
2	966	1
2	104	1
5	1509	1

B. Pembentukan Daftar Kandidat

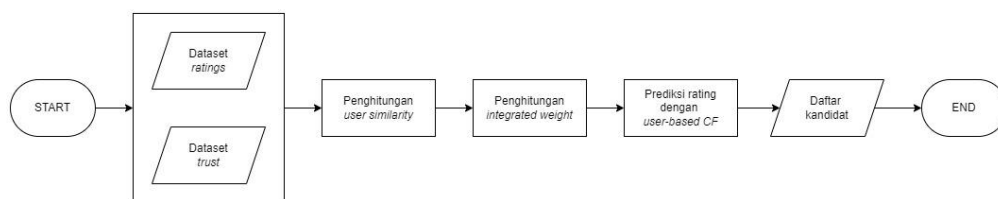
Dari dataset *ratings* akan dibuat daftar kandidat *item* yang akan direkomendasikan kepada *user* aktif menggunakan algoritma *user-based Collaborative Filtering (user-based CF)*. Untuk mengatasi masalah *cold start* yang sering dijumpai dalam algoritma *user-based CF*, terlebih dahulu data akan diberi tambahan konteks informasi yang didapat dari dataset *trust*. Setelah diberi tambahan konteks informasi, prediksi *rating* dilakukan untuk membentuk daftar kandidat *item*. Berikut merupakan langkah-langkah dari tahap pembentukan daftar kandidat:

Langkah 1. Hitung nilai kemiripan antar-*user* (*user similarity*) dari *user* yang akan diberi rekomendasi (*user* aktif) dengan *user* lain menggunakan *trust-based similarity* atau persamaan (1).

Langkah 2. Kelompokkan *item* menggunakan K-Means, kemudian hitung *user interest distribution* dari *user* aktif dan *user* lain dengan (6), kemudian integrasikan *user interest distribution* dengan *user similarity* untuk mendapatkan *aggregate weight* dengan (7).

Langkah 3. Memilih lima *user* teratas berdasarkan *aggregate weight*, kemudian melakukan prediksi *rating* dari *user* aktif terhadap *item* berdasarkan kemiripan dengan lima *user* teratas tersebut dengan menggunakan algoritma *user-based CF*. Setelah itu, sepuluh *item* dengan *rating* paling tinggi dipilih menjadi kandidat *item* yang akan direkomendasikan.

Gambar 2 menampilkan alur dari tahap pembentukan daftar kandidat:



GAMBAR 2
ALUR KERJA TAHAP PEMBENTUKAN DAFTAR KANDIDAT

C. Penerapan Diversity Balancing

Rekomendasi konvensional hanya menggunakan *user rating* untuk memberi peringkat *item* dan tujuannya hanya untuk meningkatkan *accuracy*. Tetapi dalam sistem pemberi rekomendasi ini faktor *diversity* ditambahkan ke dalam pertimbangan. *Diversity balancing* disini diterapkan dengan mengganti *item* yang memiliki *rating* tertinggi dengan *item* di luar daftar kandidat yang memiliki kemiripan paling sedikit dari *item* tersebut. Dalam proses penggantian *item* ini, sistem membandingkan 4 *item* teratas dari daftar kandidat masing-masing dengan top-30 *item* dengan *rating* tertinggi diluar daftar kandidat (*neglected list*). *Item* dengan *similarity* terendah dipilih sebagai *item* pengganti *item* tersebut. Perbandingan *item similarity* dilakukan dengan persamaan berikut:

$$\frac{R_i - R_j}{P_i - P_j} \quad (9)$$

Dimana R_i dan R_j adalah nilai *rating* yang diberikan *user* u berikan ke *item* i dan j , dan P_i dan P_j adalah popularitas *item* i dan j . Popularitas *item* dihitung dengan menghitung jumlah *rating* dari *user* terhadap *item* tersebut. Setelah *item* diganti, maka akan didapatkan top-10 daftar rekomendasi film.

Untuk lebih memahami algoritma dari *Diversity Balancing* untuk *Two-stage Collaborative Filtering (DBTS)*, langkah terperinci tertera dalam Algoritma 1 sebagai berikut:

Algoritma 1 : <i>Diversity Balancing</i> untuk <i>Two-stage Collaborative Filtering (DBTS)</i>
Input : <i>ratings R, trust T, user aktif U</i>
Output: top-10 recommendation list
Begin.
Tahap 1 pembentukan daftar kandidat
1. Menghitung <i>similarity user</i> aktif dengan <i>user</i> lain menggunakan persamaan (1)
2. Membagi <i>user</i> kedalam kelompok <i>low-trust based</i> dan <i>high-trust based</i> berdasarkan θ
3. Menggunakan K-means untuk klusterisasi <i>item</i>
4. Menghitung <i>interest weight</i> antar <i>user</i> aktif dan <i>user</i> lain menggunakan (6)
5. Menyatakan <i>interest weight</i> dan <i>similarity</i> menggunakan (7)
6. Mengurutkan <i>user</i> berdasarkan <i>weight</i> secara menurun
7. Memilih top n <i>user</i> sebagai <i>based</i> dari active <i>user</i>
8. Memprediksi <i>rating</i> yang belum diketahui dari active <i>user</i>
Tahap 2 penggantian <i>item</i> dan pembentukan daftar rekomendasi

9. Memilih n item dengan prediksi *rating* tertinggi
10. Menghitung item pengganti dengan (9)
11. Memasukkan item dengan *rating* terendah kedalam daftar rekomendasi
12. Membuat top-n recommendation list

End.

D. Metriks Evaluasi

Sistem pemberi rekomendasi ini akan diuji dengan dua pengukuran yaitu *accuracy metric* untuk mengukur performansi dari sistem pemberi rekomendasi, dan *diversity metric* untuk mengukur distribusi konten populer atau keberagaman dari sistem pemberi rekomendasi. *Accuracy* diukur berdasarkan *precision* dan *recall*. *Precision* adalah perbandingan antara true positif dari semua yang positif.

$$precision = \frac{t_p}{t_p + f_p} \quad (10)$$

Sedangkan *recall* adalah perbandingan antara true positif dari semua yang seharusnya positif, dengan kata lain *recall* adalah seberapa besar algoritma dapat mendeteksi true positif.

$$recall = \frac{t_p}{t_p + f_n} \quad (11)$$

Dalam sistem pemberi rekomendasi, *precision* juga dapat diartikan sebagai perbandingan antara jumlah item yang direkomendasikan @k yang relevan dan jumlah item yang direkomendasikan @k. Serta *recall* adalah perbandingan antara jumlah item yang direkomendasikan @k yang relevan dan jumlah item yang relevan. Item yang relevan adalah item dengan *rating* sebenarnya (*actual rating*) atau *rating* yang terdapat dalam dataset melebihi batas *threshold* yang telah ditentukan sebelumnya. Sedangkan item yang direkomendasikan adalah item yang memiliki *rating* hasil prediksi oleh sistem pemberi rekomendasi yang melebihi batas *threshold* [15].

Diversity metric diukur berdasarkan *Personal Diversity* (PD) dan *Aggregate Diversity* (AD). PD adalah ketidakmiripan antara item dalam suatu daftar rekomendasi. PD dapat digambarkan dengan:

$$PD = \frac{1}{|U|} \sum_{u \in U} \left(1 - \frac{\sum_{i,j \in L(u), i \neq j} (1 - sim(i,j))}{\frac{1}{2} |L(u)| (|L(u)| - 1)} \right) \quad (12)$$

Dimana $|U|$ adalah jumlah pengguna, $sim(i,j)$ merupakan kemiripan item i dan j . $L(u)$ merupakan daftar rekomendasi dan $|L(u)|$ merupakan jumlah item di dalam daftar rekomendasi $L(u)$.

Selanjutnya *Aggregate Diversity* adalah total item unik yang direkomendasikan kepada seluruh user. AD dapat dihitung dengan persamaan:

$$AD = \frac{\sum_{u \in U} |L(u)|}{|I|} \quad (13)$$

Dimana $|I|$ adalah jumlah seluruh item yang direkomendasikan kepada user [2].

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

Pengujian dilakukan dengan menjalankan model sistem pemberi rekomendasi terhadap semua user yang ada dalam dataset yang dipakai. Sesuai batasan yang digunakan, terdapat total 60 user yang diujikan dari 1508 user yang ada. Pengujian ini dilakukan terhadap user 1 sampai dengan 60. Pengujian dijalankan untuk kemudian didapatkan list rekomendasi item untuk semua user. Setelah itu akan dihitung nilai *accuracy* dan *diversity* sesuai dengan metriks evaluasi yang digunakan. Pengujian dilakukan dengan menghitung pengaruh jumlah k . k adalah jumlah item teratas dalam list rekomendasi yang akan diikutkan ke dalam penghitungan. Penghitungan ini dilakukan dengan *threshold rating* sebesar 3.0. Maka akan didapat hasilnya dalam Tabel 3

TABEL 3
PERFORMA ACCURACY SISTEM UNTUK SETIAP K

K (recommendation list size)	Metrics	
	precision	recall
6	0.8000	0.5366
7	0.7531	0.4803
8	0.7266	0.4398
9	0.7137	0.4102
10	0.7046	0.3874

Dari hasil di Tabel 3 tersebut, terlihat bahwa nilai k sebesar 6 memiliki nilai *precision* sebesar 0.8000 dan *recall* sebesar 0.5366. Hal ini menjadikan nilai $k=6$ merupakan k dengan *accuracy* tertinggi. Terlihat dari tabel tersebut bahwa semakin banyak item yang masuk ke dalam penghitungan maka nilai *precision* dan *recall* akan turun meskipun nilainya tidak turun begitu jauh.

Setelah itu dilakukan pengujian pengaruh *threshold rating* terhadap nilai *accuracy* yang dihasilkan untuk memastikan *threshold* yang optimal. Pengujian ini dilakukan dengan jumlah k yang sama untuk setiap *threshold*. Dimisalkan nilai k adalah 8, maka akan didapat performa model berdasarkan *threshold rating* seperti pada Tabel 4 berikut:

TABEL 4
PERFORMA ACCURACY SISTEM UNTUK SETIAP NILAI THRESHOLD

Rating threshold	Metrics	
	precision	recall
1.5	0.5301	0.4437
2.0	0.6117	0.4495
2.5	0.6894	0.4493
3.0	0.7266	0.4398
3.5	0.7971	0.4449

Tabel 4 menunjukkan bahwa *threshold* 3.5 memiliki nilai *precision* paling tinggi dengan nilai 0.7971. Meskipun nilai *recall* tidak jauh berbeda bahkan bisa dibilang sama. Dari hasil tersebut juga dapat terlihat bahwa semakin besar *threshold* maka *accuracy* semakin tinggi.

Kemudian setelah itu dilakukan pengujian terhadap *diversity metric*. Pengujian dilakukan dengan menggunakan persamaan (12) untuk mengukur *personal diversity* (PD) dan persamaan (13) untuk mengukur *aggregate diversity* (AD).

TABEL 5
PERFORMA DIVERSITY UNTUK SETIAP K

Recommendation list size	Metrics	
	PD	AD
4	0.5006	0.2789
5	0.5009	0.2937
6	0.5085	0.3145
7	0.5249	0.3264
8	0.5270	0.3382
9	0.5245	0.3560
10	0.5301	0.3709

Dari penghitungan tersebut didapatkan nilai tertinggi 0.5301 untuk PD dan 0.3709 untuk nilai AD. Nilai PD dan AD terpengaruh oleh *k* atau jumlah *item* yang direkomendasikan ke *user*. Dari Tabel 5 dapat terlihat bahwa nilai PD dan AD cenderung naik ketika jumlah *item* yang direkomendasikan bertambah. Hal ini bertolak belakang dengan *accuracy* yang dimana ketika jumlah *item* bertambah maka *accuracy* akan berkurang.

Pengujian kemudian dilakukan untuk *user* 61 sampai dengan 120 dengan metode yang sama. Nilai *threshold rating* diatur sebesar 3.0 dan nilai *k* diatur sebesar 8. Setelah itu hasilnya akan dibandingkan dengan hasil pengujian yang sama terhadap *user* 1-60. Hasil perbandingan dapat dilihat di tabel 6 sebagai berikut:

TABEL 6
PERBANDINGAN PERFORMA ACCURACY DAN DIVERSITY SISTEM

User set	Metrics			
	precision	recall	PD	AD
1 - 60	0.7266	0.4398	0.5270	0.3382
61 - 120	0.7957	0.5942	0.5582	0.2009

Dari hasil di Tabel 6 tersebut, terlihat bahwa nilai dari metrik *accuracy* dan *diversity* dari pengujian kedua *user set* tidak jauh berbeda. Pengujian terhadap *user* 61-120 memiliki hasil *accuracy* yang sedikit lebih tinggi dari *user* 1-60. Tetapi hasil pengujian terhadap metrik *diversity* dari *user* 61-120 didapatkan hasil PD yang lebih tinggi dan hasil AD yang lebih rendah.

B. Analisis Hasil Pengujian

Berdasarkan Tabel 3 dapat diamati bahwa jumlah *item* yang ikut dalam penghitungan (*k*) akan

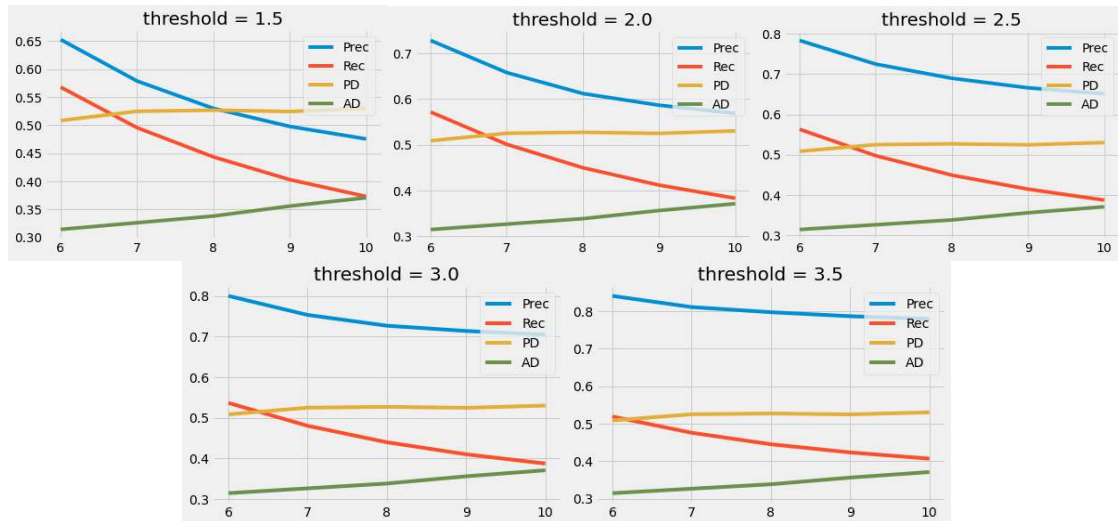
mempengaruhi nilai *precision* dan *recall*. Semakin banyak *item* yang masuk, maka nilai *precision* dan *recall* akan semakin kecil. Terbukti dari Tabel 3 dimana jumlah *item* sebanyak 6 memiliki *precision* sebesar 0.8000 dan *recall* sebesar 0.5366. Kemudian berangsur-angsur turun sampai *k*=10 dengan *precision* 0.7046 dan *recall* sebesar 0.3874.

Hal ini juga berlaku untuk *threshold rating*. Semakin besar nilai *threshold* maka akan semakin besar nilai *precision* nya. Nilai *threshold* yang besar mengakibatkan semakin banyak *item* yang tidak masuk ke dalam penghitungan nilai *precision* dan *recall*. Begitu pula sebaliknya, *threshold* yang kecil akan mengakibatkan

banyak *item* yang masuk ke dalam penghitungan. Tetapi walaupun begitu, *threshold* memiliki pengaruh yang kecil terhadap nilai *recall* untuk kondisi dataset yang demikian. Terlihat bahwa nilai *recall* cenderung tetap dan hanya mengalami sedikit perubahan.

Tabel 5 menerangkan nilai PD dan AD yang didapatkan dari pengujian. Nilai PD dan AD cenderung

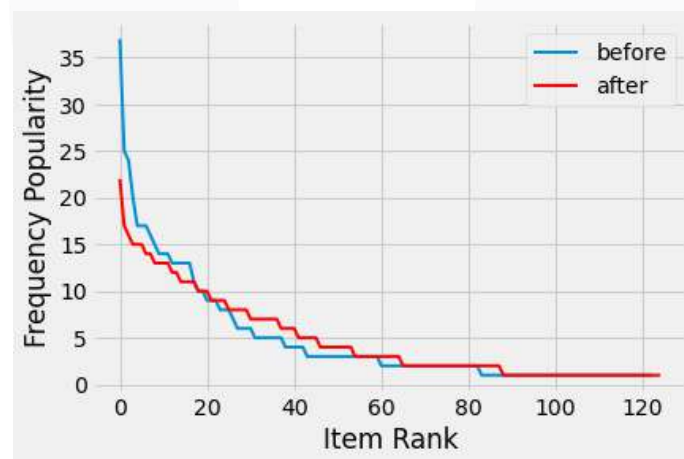
berbanding lurus dengan kenaikan jumlah *item* yang direkomendasikan. Hal ini tentu saja diakibatkan karena semakin banyak *item* yang direkomendasikan maka semakin besar kesempatan untuk *item* yang berbeda akan direkomendasikan.



GAMBAR 3
PERBANDINGAN NILAI *PRECISION*, *RECALL*, PD, DAN AD

Gambar 3 menunjukkan perbandingan antara *precision*, *recall*, PD, dan AD. Dari gambar tersebut bisa disimpulkan bahwa jumlah *item* optimal yang dapat direkomendasikan kepada *user* adalah 7 *item*. Angka tersebut dipilih karena dalam grafik perbandingan terlihat bahwa 7 memiliki loncatan *diversity* yang lumayan signifikan dibandingkan dengan jumlah lain. Selain itu nilai *accuracy* juga tidak turun secara drastis di titik tersebut. Maka dari itu *diversity* dapat meningkat tanpa mengorbankan banyak *accuracy*.

Metode *Diversity Balancing* ini terbukti mampu memberikan rekomendasi terhadap *item* yang kurang populer dan mengurangi masalah *less popular item*. Gambar 4 menunjukkan perbandingan hasil rekomendasi algoritma *Two-stages Collaborative Filtering* dengan menggunakan metode *Diversity Balancing* sebelum dilakukan *Diversity Balancing* (*before*) dengan algoritma yang sama sesudah dilakukan *Diversity Balancing* (*after*).



GAMBAR 4
HASIL REKOMENDASI SEBELUM DAN SESUDAH *DIVERSITY BALANCING*

Dari Gambar 4 terlihat bahwa grafik *before* memiliki *item* populer yang cukup banyak masuk ke dalam rekomendasi. Grafik *before* memiliki kemiringan yang cukup curam menandakan bahwa distribusi rekomendasi terhadap *item* populer dan yang kurang populer tidak merata dan cenderung banyak merekomendasikan *item* yang populer.

Sedangkan dalam grafik *after*, kemiringan grafik tidak begitu curam dan cenderung landai. Hal ini menandakan distribusi rekomendasi terhadap *item* yang populer dan yang kurang populer lebih merata.

V. KESIMPULAN

Pada penelitian kali ini, kami telah mengimplementasikan sistem pemberi rekomendasi berbasis *Two-stages Collaborative Filtering* dengan menggunakan metode *Diversity Balancing* di domain film. Performa sistem dievaluasi menggunakan *accuracy metric* dan *diversity metric*. *Accuracy* diukur dengan *precision* dan *recall*, sedangkan *diversity* diukur dengan *personal diversity* dan *aggregate diversity*. Berdasarkan hasil evaluasi, nilai *diversity* terlihat meningkat ketika *item* yang direkomendasikan lebih banyak. Walaupun nilai *accuracy* akan terlihat menurun karena dibandingkan terhadap dataset *item* yang populer.

Berdasarkan hasil dari metode *Diversity Balancing*, untuk saat ini dapat ditemukan parameter yang optimal untuk kondisi dataset tertentu. Parameter optimal hasil dari menyeimbangkan nilai *accuracy* dan *diversity* didapatkan nilai $k = 7$ dan nilai *threshold rating* sebesar 3.0. Metode tersebut juga terbukti dapat menghasilkan sistem pemberi rekomendasi yang merekomendasikan *item* yang kurang populer namun masih relevan terhadap *user* sekaligus mengurangi masalah *less popular item*.

Kedepannya penelitian ini dapat dikembangkan lagi dengan mengatasi batasan komputasi supaya dataset yang digunakan menjadi lebih kaya. Dengan dataset yang semakin kaya juga, diharapkan dapat dilakukan tuning parameter yang lebih baik sehingga ditemukan parameter yang paling optimal untuk seluruh dataset yang digunakan. Selain itu perlu juga perbandingan dengan sistem pemberi rekomendasi lain yang populer seperti *User-based Collaborative Filtering*, *Singular Value Decomposition* (SVD) dengan K-Means, dsb.

REFERENSI

- [1] Asanov, D., (2011). Algorithms and methods in recommender systems. Berlin Institute of Technology, Berlin, Germany
- [2] Zhang, Liang et al. (2020) 'Diversity Balancing for Two-stage Collaborative Filtering in Recommender Systems', Applied Sciences, 10(4).
- [3] Sebastia, L. et al. (2009) 'E-Tourism: A tourist recommendation and planning application', International Journal of Artificial Intelligence Tools, 18, pp. 717–738.
- [4] Adomavicius, G. and Tuzhilin, A. (2005) 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', IEEE Transactions on Knowledge and Data Engineering, 17(6), pp. 734–749
- [5] Lika, B., Kolomvatsos, K. and Hadjiefthymiades, S. (2014) 'Facing the cold start problem in recommender systems', Expert Systems with Applications, 41(4, Part 2), pp. 2065–2073.
- [6] Hamedani, E. M. and Kaedi, M. (2019) 'Recommending the long tail items through personalized diversification', Knowledge-Based Systems, 164, pp. 348–357.
- [7] Park, yoon-joo (2013) 'The Adaptive Clustering Method for the Long Tail Problem of Recommender Systems', Knowledge and Data Engineering, IEEE Transactions on, 25, pp. 1904–1915. doi: 10.1109/TKDE.2012.119.
- [8] Abdollahpouri, H., Burke, R. and Mobasher, B. (2019) 'Managing Popularity Bias in Recommender Systems with Personalized Re-ranking', CoRR, abs/1901.07555.
- [9] Yin, H., Cui, B., Li, J., Yao, J. and Chen, C., 2012. Challenging the long tail recommendation. arXiv preprint arXiv:1205.6700.
- [10] Gogna, A. and Majumdar, A. (2017) 'Balancing Accuracy and Diversity in Recommendations Using Matrix Completion Framework', Know.-Based Syst., 125(C), pp. 83–95.
- [11] Breese, J. S., Heckerman, D. and Kadie, C. (1998) 'Empirical Analysis of Predictive Algorithms for Collaborative Filtering', in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (UAI'98), pp. 43–52.
- [12] Mustafa, N. et al. (2017) 'Collaborative filtering: Techniques and applications', in 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), pp. 1–6.
- [13] Singh, P., Dutta Pramanik, P. and Choudhury, P. (2020) 'Collaborative Filtering in Recommender Systems: Technicalities, Challenges, Applications, and Research Trends', in. doi: 10.1201/9781003007210-8.
- [14] Su, X. and Khoshgoftaar, T. (2009) 'A Survey of Collaborative Filtering Techniques', Adv. Artificial Intelligence, 2009. doi: 10.1155/2009/421425.
- [15] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22, 1 (January 2004), 5–53. <https://doi.org/10.1145/963770.963772>