

TESIS

**IMPLEMENTASI METODE HYBRID FILTERING PADA SISTEM
REKOMENDASI UNTUK MENINGKATKAN KUALITAS
REKOMENDASI**



Disusun oleh:

Nama : Annas Al Amin
NIM : 19.51.1276
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

TESIS

**IMPLEMENTASI METODE HYBRID FILTERING PADA SISTEM
REKOMENDASI UNTUK MENINGKATKAN KUALITAS
REKOMENDASI**

**IMPLEMENTATION OF HYBRID FILTERING METHOD IN
RECOMMENDATION SYSTEM TO IMPROVE
RECOMMENDATION QUALITY**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Annas Al Amln
NIM : 19.51.1276
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

HALAMAN PENGESAHAN

**IMPLEMENTASI METODE HYBRID FILTERING PADA SISTEM
REKOMENDASI UNTUK MENINGKATKAN KUALITAS
REKOMENDASI**

**IMPLEMENTATION OF HYBRID FILTERING METHOD IN
RECOMMENDATION SYSTEM TO IMPROVE
RECOMMENDATION QUALITY**

Dipersiapkan dan Disusun oleh

Annas Al Amin

19.51.1276

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 08 Juni 2021

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 08 Juni 2021

Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

**IMPLEMENTASI METODE HYBRID FILTERING PADA SISTEM
REKOMENDASI UNTUK MENINGKATKAN KUALITAS
REKOMENDASI**

**IMPLEMENTATION OF HYBRID FILTERING METHOD IN
RECOMMENDATION SYSTEM TO IMPROVE
RECOMMENDATION QUALITY**

Dipersiapkan dan Disusun oleh

Annas Al Amin

19.51.1276

Telah Drujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 08 Juni 2021

Pembimbing Utama

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Anggota Tim Penguji

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Pembimbing Pendamping

Hanif Al Fatta, S.Kom., M.Kom
NIK. 190302096

Dr. Wing Wahyu W, MAFIS, Ak
NIK. 555195

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 08 Juni 2021

Direktur Program Pascasarjana

Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang berandatangan di bawah ini,

Nama mahasiswa : Annas Al Amin
NIM : 19.51.1276
Konentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi

Dosen Pembimbing Utama : Dr. Andi Sunyati, M.Eng.
Dosen Pembimbing Pendamping : Hanif Al Fatah, S.Kom., M. Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan diabaikan nama pengarang dan dicantumkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menyetujui SANKSI AKADEMIK dengan peralihan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 08 Juni 2021
Yang Menyatakan,



Annas Al Amin

HALAMAN PERSEMBAHAN

Alhamdulillahirrabil'alamin, puji syukur kehadiran Allah Subhanahu wa ta'ala yang telah memberikan rahmat, hidayah, kesehatan, kemudahan, serta kemampuan kepada saya, sehingga saya dapat menyelesaikan Penelitian Tesis ini selama 3 Semester. Pada halaman persembahan ini, saya ingin berterimakasih yang sebesar-besarnya kepada:

1. Sang pencipta Allah Subhanahu wa ta'ala, Insha Allah selama proses menuntut ilmu sampai jenjang S2 ini saya niatkan semuanya hanya untuk beribadah kepadamu ya Allah, semoga setiap penelitian ini tidak hanya bermanfaat untuk saya tetapi juga bisa bermanfaat untuk orang lain yang membacanya.
2. Kedua orang tua saya Bapak Amin Budiyono dan Ibu Yuniarti yang telah melahirkan, merawat saya dengan penuh kasih sayang serta telah memberikan banyak ilmu, dan pengetahuan kepada saya. Terimakasih atas doa dan dukungan yang diberikan kepada saya selama ini.
3. Kakak saya, Rizki Al-Amin, yang telah membantu membiayai biaya kuliah saya pada saat kuliah S1, karena setelah lulus atas izin dari Allah SWT saya mendapatkan Beasiswa untuk melanjutkan kuliah pada jejang S2.
4. Dosen pembimbing saya Bapak Dr. Andi Sunyoto, M.Kom, dan Bapak Hanif Al Fatta, M.Kom yang telah memberikan banyak dukungan, ilmu, pengetahuan, dan wawasan baru kepada saya selama menyelesaikan Penelitian Tesis ini.
5. Teman-teman sekelas 19 MTI A Reguler yang selalu memberikan doa dan dukungan kepada saya sampai saat ini. Terimakasih atas doa dan dukungannya yang telah diberikan selama ini, semoga kalian semua bisa cepat menyusul.
6. Magister Teknik Informatika Universitas AMIKOM Yogyakarta beserta seluruh jajaran Dosen yang telah mengajarkan banyak ilmu kepada saya dan juga seluruh Karyawannya.

HALAMAN MOTTO

“Pencapaian saya sampai saat ini tidak akan pernah terjadi kecuali atas **izin** dari **Allah Subhanahu wa ta'ala** yang telah memberikan segala keberkahan, kelancaran dan kemudahan kepada saya sampai titik ini”.

“Esensi dari sebuah pencapaian adalah ketika kita bisa mendapatkan **Berkah** dan **Ridho** dari **Allah Subhanahu wa ta'ala** untuk **bekal** di **Akhirat** kelak”.

“Sebuah kesuksesan akan lebih cepat tercapai jika dalam prosesnya melibatkan **Allah Subhanahu wa ta'ala**”.

“Tidak ada artinya sebuah kesuksesan di dunia ini jika tidak menghasilkan nilai **kebalkan** dan **keberkahan** di hadapan **Allah Subhanahu wa ta'ala**”.

“Menuntut **ilmu** itu wajib atas setiap Muslim” (HR. Ibnu Majah no. 224, dari sahabat Anas bin Malik radhiyallahu'anhu, dishahihkan Al Albani dalam Shahiih al-Jaami'ish Shaghiir no. 3913).

“Jika seorang manusia meninggal, terputuslah amalnya, kecuali dari tiga hal: sedekah jariyah, **ilmu** yang bermanfaat atau anak shalih yang berdoa untuknya.” (HR. Muslim).

“Para ulama adalah pewaris para nabi. Sesungguhnya para nabi tidak mewariskan dinar ataupun dirham, tetapi mewariskan **ilmu**. Maka dari itu, barang siapa mengambilnya, ia telah mengambil bagian yang cukup.” (HR. Abu Dawud, at-Tirmidzi dan Ibnu Majah).

KATA PENGANTAR

Puji dan syukur kehadiran Allah Subhanahu wa ta'ala yang telah memberikan rahmat, hidayah, kesehatan, kemudahan, serta kemampuan sehingga penulis dapat menyelesaikan Penelitian Tesis ini sesuai dengan waktu yang ditargetkan. Tidak lupa sholawat dan salam selalu disampaikan kepada manusia paling sempurna yang pernah hidup di dunia yaitu Nabi Muhammad Shallallahu 'alaihi wasallam, semoga kami semua termasuk kedalam golongan orang-orang yang mendapatkan Syafa'at dari beliau pada saat di Akhirat kelak.

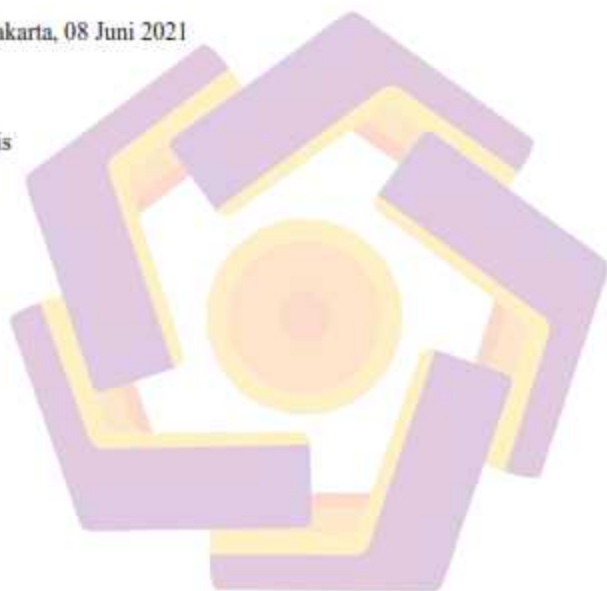
Proses penyusunan Laporan Tesis ini tidak lepas dari doa, dan dukungan dari berbagai pihak, oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua saya Bapak Amin Budiyono dan Ibu Yuniarti yang telah melahirkan, merawat saya dengan penuh kasih sayang serta telah memberikan banyak ilmu, dan pengetahuan kepada saya.
2. Kakak saya, Rizki Al-Amin, yang telah membantu membiayai biaya kuliah saya pada saat kuliah S1.
3. Bapak Dr. Andi Sunyoto, M.Kom selaku dosen pembimbing utama dan Bapak Hanif Al Fatta, M.Kom selaku dosen pembimbing kedua yang telah memberikan banyak dukungan, ilmu, pengetahuan, dan wawasan baru kepada saya selama menyelesaikan Penelitian Tesis ini.
4. Bapak Dr. Hanafi, S.Kom., M.Eng., selaku dosen saya yang telah memberikan saya banyak pencerahan, ilmu, dan pengetahuan baru kepada saya dalam menyelesaikan penelitian ini.
5. Teman sekelas 19 MTI A Reguler yang selalu memberikan doa dan dukungan kepada saya dalam menyelesaikan Penelitian Tesis ini.
6. Seluruh jajaran Dosen yang telah mengajarkan banyak ilmu kepada saya dan juga seluruh Karyawan Magister Teknik Informatika Universitas AMIKOM Yogyakarta.

Penulis meyakini bahwa tidak ada karya penelitian yang sempurna di dunia ini, karena kesempurnaan yang sesungguhnya hanya milik Allah Subhanahu wa ta'ala. Oleh karena itu, penulis sangat menerima setiap saran dan kritik yang diberikan pembaca terhadap penelitian ini dengan tujuan untuk memperbaiki penelitian ini menjadi lebih baik lagi untuk kedepannya. Penulis berharap dari penelitian bisa memberikan banyak manfaat dan berkah tidak hanya untuk Penulis tetapi juga untuk orang lain yang membacanya.

Yogyakarta, 08 Juni 2021

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
INTISARI.....	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	6
1.6. Hipotesis.....	7
BAB II TINJAUAN PUSTAKA.....	9
2.1. Tinjauan Pustaka.....	9

2.2. Keaslian Penelitian	21
2.3. Landasan Teori.....	30
2.3.1 Sistem Rekomendasi	30
2.3.2 Pendekatan Sistem Rekomendasi.....	31
2.3.2.1 Demographic Filtering	31
2.3.2.2 Collaborative Filtering	32
2.3.2.3 Content Based Filtering.....	35
2.3.2.4 Hybrid Filtering.....	36
2.3.3 Algoritma Siste Rekomendasi.....	37
2.3.3.1 Count Vectorizer	37
2.3.3.2 Cosine Similarity.....	38
2.3.3.3 Cosine Distance.....	39
2.3.3.4 Matrix Factorization.....	39
2.3.4 Algoritma Evaluasi Error Prediksi.....	40
2.3.4.1 Root Mean Squared Error (RMSE).....	40
2.3.4.2 Mean Absolute Percentage Error (MAPE)	41
BAB III METODE PENELITIAN.....	42
3.1. Jenis, Sifat, dan Pendekatan Penelitian	42
3.2. Metode Pengumpulan Data.....	42
3.3. Metode Analisis Data	43
3.4. Alur Penelitian	44
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	50
4.1. <i>Data and Text Preprocessing</i>	50

4.1.1 <i>Collect MovieLens 1M Dataset</i>	50
4.1.2 <i>Data Selection</i>	51
4.1.3 <i>Data Filter</i>	52
4.1.4 <i>Text Preprocessing</i>	54
4.2. <i>Training Data and Results</i>	55
4.2.1 <i>Text Feature Extraction</i>	55
4.2.2 <i>Document Search by Similarity Content Item & First Item Recommendation Based on Similarity Content</i>	55
4.2.3 <i>Predict Item Rating & Second Item Recommendation Based on Highest Rating Predictions</i>	56
4.3. <i>Evaluation and Comparison of Results</i>	58
4.3.1 <i>Pra Evaluasi Error Prediksi</i>	58
4.3.2 <i>Evaluasi dan Perbandingan Error Prediksi</i>	63
4.3.3 <i>Evaluasi dan Perbandingan Hasil Rekomendasi Item</i>	67
BAB V PENUTUP	72
5.1. <i>Kesimpulan</i>	72
5.2. <i>Saran</i>	73
DAFTAR PUSTAKA	75

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	21
Tabel 2.2. Sampel Data Sebelum Ekstraksi	38
Tabel 2.3. Sampel Kata Kunci	38
Tabel 2.4. Sampel Data Sesudah Ekstraksi	38
Tabel 3.1. Sampel Data Sebelum Ekstraksi	43
Tabel 4.1. Selisih Error Prediksi Nilai RMSE dan MAPE Kondisi Pertama.....	67
Tabel 4.2. Selisih Error Prediksi Nilai RMSE dan MAPE Kondisi Kedua	67
Tabel 4.3. Hasil Rekomendasi Film Metode CBF	68
Tabel 4.4. Hasil Rekomendasi Film Metode HF.....	69

DAFTAR GAMBAR

Gambar 2.1. Ilustrasi Metode DF.....	32
Gambar 2.2. Ilustrasi Metode CF.....	35
Gambar 2.3. Ilustrasi Metode CBF.....	36
Gambar 2.4. Ilustrasi Metode HF.....	37
Gambar 3.1. Tahapan Implementasi Metode Hybrid Filtering.....	44
Gambar 4.1. <i>Source Code</i> Sampel Dataset MovieLens 1.....	50
Gambar 4.2. <i>Source Code</i> Sampel Dataset MovieLens 2.....	51
Gambar 4.3. <i>Source Code</i> Data Selection.....	52
Gambar 4.4. <i>Source Code</i> Filter Atribut User ID Kondisi Pertama.....	53
Gambar 4.5. <i>Source Code</i> Filter Atribut User ID Kondisi Kedua.....	53
Gambar 4.6. <i>Source Code</i> Pembuatan Data Frame untuk <i>Text Preprocessing</i>	54
Gambar 4.7. <i>Source Code</i> Feature Text Extraction.....	55
Gambar 4.8. <i>Source Code</i> Document Search by Similarity Content Item.....	56
Gambar 4.9. <i>Source Code</i> Second Training Process using CF based on the MF Model (SVD).....	57
Gambar 4.10. <i>Source Code</i> Top 10 Personalized Item Recommendations.....	57
Gambar 4.11. <i>Source Code</i> Proses Evaluasi Kondisi Pertama Metode HF dengan RMSE.....	59
Gambar 4.12. <i>Source Code</i> Proses Evaluasi Kondisi Pertama Metode CF Berbasis Memory dengan RMSE.....	59

Gambar 4.13. <i>Source Code</i> Proses Evaluasi Kondisi Pertama Metode HF dengan MAPE.....	60
Gambar 4.14. <i>Source Code</i> Proses Evaluasi Kondisi Pertama Metode CF Berbasis Memory dengan MAPE.....	60
Gambar 4.15. <i>Source Code</i> Proses Evaluasi Kondisi Kedua Metode HF dengan RMSE	61
Gambar 4.16. <i>Source Code</i> Proses Evaluasi Kondisi Kedua Metode CF Berbasis Memory dengan RMSE	61
Gambar 4.17. <i>Source Code</i> Proses Evaluasi Kondisi Kedua Metode HF dengan MAPE.....	62
Gambar 4.18. <i>Source Code</i> Proses Evaluasi Kondisi Kedua Metode CF Berbasis Memory dengan MAPE.....	62
Gambar 4.19. Hasil Evaluasi dan Perbandingan Nilai RMSE Kondisi Pertama ..	64
Gambar 4.20. Hasil Evaluasi dan Perbandingan Nilai MAPE Kondisi Pertama ..	64
Gambar 4.21. Hasil Evaluasi dan Perbandingan Nilai RMSE Kondisi Kedua.....	65
Gambar 4.22. Hasil Evaluasi dan Perbandingan Nilai MAPE Kondisi Kedua.....	65

INTISARI

Sistem rekomendasi banyak digunakan pada platform website seperti *e-commerce*, *marketplace*, dan streaming film untuk menghasilkan rekomendasi *item* yang sesuai kepada setiap pengguna. Metode tradisional Collaborative Filtering (CF) berbasis memory dan Content Based Filtering (CBF) saat ini masih banyak digunakan dalam sistem rekomendasi. Metode CF bertumpu pada *rating item* pengguna terhadap *item* sebagai pendekatan dasar untuk menghitung kesamaan respon pengguna terhadap produk untuk memprediksi *item* yang akan direkomendasikan, sedangkan metode CBF memanfaatkan kesamaan data konten *item* sebagai dasar untuk menghasilkan rekomendasi *item*. Terdapat kelemahan pada metode CF berbasis memory yaitu error prediksi tinggi, sedangkan pada metode CBF hasil rekomendasi *itemnya* tidak *personalized*. Penelitian ini bertujuan untuk mereduksi error prediksi dari pendekatan CF berbasis memory dan mampu menghasilkan rekomendasi *item* yang *personalized* menggunakan metode Hybrid Filtering (HF) untuk meningkatkan kualitas sistem rekomendasi. Metode HF merupakan kombinasi dari pendekatan CF berbasis model Matrix Factorization dan CBF yang mampu mengurangi error prediksi untuk menghasilkan rekomendasi *item* yang akurat dan *personalized*. Metode yang diusulkan dievaluasi berdasarkan 2 kondisi yaitu data *non sparsity* dan *sparsity* menggunakan RMSE dan MAPE untuk mengukur error prediksi. Setelah evaluasi kedua kondisi selesai dilakukan, metode Hybrid CF mampu mereduksi error prediksi lebih kecil dibandingkan dengan CF berbasis memory dengan penurunan nilai RMSE 0,10, MAPE 5,57% pada kondisi pertama, penurunan nilai RMSE 0,11, MAPE 4,81% pada kondisi kedua, dan mampu memberikan rekomendasi *item* yang *personalized*. Berdasarkan hasil, pendekatan HF lebih baik daripada CF berbasis memory dan CBF.

Kata kunci: Recommender System, Hybrid Filtering, Matrix Factorization, Collaborative Filtering, Content Based Filtering.

ABSTRACT

Recommendation systems are widely used on website platforms such as e-commerce, marketplaces, streaming movies to produce appropriate item recommendations for each user. The traditional Collaborative Filtering (CF) methods based on memory and Content-Based Filtering (CBF) are still widely used today. The CF method relies on rating user items as a fundamental approach to calculating user responses to products to predict things to be recommended. In contrast, the CBF method uses item content data as the basis for generating item recommendations. There is a weakness in the memory-based CF method, namely the high prediction error, whereas in the CBF method, the item recommendation is not personalized. This study aims to reduce predictive errors from the memory-based CF approach and recommend personalized items using the Hybrid Filtering (HF) method to improve the quality of recommendation system. The HF method combines the CF approach based on the Matrix Factorization and CBF model, which can reduce prediction errors to produce accurate and personalized item recommendations. The proposed method is evaluated based on two conditions: non-sparsity and sparsity data, using RMSE and MAPE to measure the prediction error. After evaluation of the two conditions was completed, Hybrid CF method was able to reduce the prediction error smaller than memory-based CF with a decrease in RMSE value 0.10, MAPE 5.57% in the first condition, decrease in RMSE value 0.11, MAPE 4.81% in the second condition, and able to provide personalized item recommendations. Based on the results, the HF approach is better than memory-based CF and CBF.

Keyword: Recommender System, Hybrid Filtering, Matrix Factorization, Collaborative Filtering, Content Based Filtering.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sistem rekomendasi merupakan salah satu sistem pendukung yang digunakan pada sebuah *platform* berbasis *website* seperti *ecommerce*, *marketplace*, streaming film untuk memberikan rekomendasi *item* yang disukai oleh penggunanya. Saat ini metode yang banyak digunakan pada sistem rekomendasi adalah Collaborative Filtering (CF) berbasis *memory* yang merupakan model tradisional pada sistem rekomendasi. Metode tradisional CF adalah salah satu metode umum yang sering digunakan pada sistem rekomendasi menggunakan data *rating* yang diberikan oleh pengguna. Ide utamanya untuk memprediksi *item* yang mungkin disukai oleh pengguna berdasarkan preferensi pengguna lain dalam memberikan *rating* pada suatu *item* (Nassar, Jafar, and Rahhal 2020; Cui et al. 2019). Metode ini merupakan salah satu yang paling sukses dan sering diimplementasikan pada sistem rekomendasi di berbagai macam *platform* karena model yang dihasilkan bisa memberikan rekomendasi *item* yang *personalized* (Jiang et al. 2019; Lv et al. 2020; Li et al. 2019; K. Wang et al. 2019).

Selain metode CF terdapat metode lain yang juga banyak digunakan pada sistem rekomendasi yaitu Content Based Filtering (CBF), metode ini memanfaatkan kesamaan data konten suatu *item* seperti kategori, deskripsi, atau atribut lain sebagai dasar untuk menghasilkan rekomendasi *item*. Metode ini mampu memberikan

rekomendasi *item* kepada pengguna tanpa menggunakan data *rating* yang telah diberikan pengguna pada suatu *item*.

Metode CF berbasis memory dan CBF ini memiliki beberapa kelemahan, kekurangan pertama pada metode CF, jika terdapat *item* yang tidak memiliki *rating*, maka sistem tidak bisa memberikan rekomendasi (*Cold Start Problem*) (Rahmawati, Nurjanah, and Rismala 2018; Wang et al. 2019). Kedua, metode ini mengabaikan data konten *item* yang sebenarnya bisa digunakan sebagai dasar untuk rekomendasi *item*. Ketiga, metode CF berbasis memory memerlukan komputasi yang tinggi karena setiap ada data *rating* baru, model harus *ditraining* ulang untuk menghasilkan rekomendasi *item* yang tepat tetapi membuat hasil error prediksinya tinggi (Prasetyo et al. 2019; Wijaya and Alfian 2018; Mongia et al. 2020; Guo et al. 2019). Lalu pada metode CBF, kelemahan pertama hasil rekomendasi *item* belum *personalized*, artinya rekomendasi *item* kepada pengguna semuanya sama. Kedua, sistem tidak bisa memberikan rekomendasi *item* yang memiliki konten berbeda karena hanya menggunakan informasi *item* yang telah dilihat dan tidak memanfaatkan preferensi *rating* dari pengguna lain (Tewari 2020).

Tujuan penelitian ini untuk mereduksi error prediksi dari metode CF berbasis memory dan menghasilkan rekomendasi *item personalized* dengan konten yang terarah untuk memperbaiki kelemahan metode CBF yang rekomendasi *itemnya* belum *personalized* pada penelitian sebelumnya menggunakan kombinasi metode CF berbasis model Matrix Factorization (MF) dan CBF menjadi metode Hybrid Filtering (HF) sehingga mampu meningkatkan kualitas sistem rekomendasi (Pujahari and Sisodia 2020).

Alasan menggunakan metode HF pada penelitian ini karena terdapat beberapa kelebihan, pertama sistem tetap bisa memberikan rekomendasi *item* kepada pengguna baru yang belum memberikan *rating item* sama sekali (Natarajan et al. 2020). Kedua, model mampu memberikan rekomendasi *item* dengan konten yang terarah secara *personalized*, sehingga setiap pengguna bisa mendapatkan rekomendasi *item* yang berbeda (Hu et al. 2019). Ketiga, metode HF mampu mengurangi beban komputasi sehingga bisa memberikan rekomendasi *item* yang cepat dengan error prediksi yang kecil.

Model yang dihasilkan dari metode HF ini nantinya mampu merekomendasikan *item* berdasarkan kemiripan konten karena kemampuan dari metode CBF. Setelah itu, hasil rekomendasi *item* berdasarkan kemiripan konten akan diolah lagi menggunakan metode CF berbasis model MF untuk mencoba menghasilkan rekomendasi *item* yang *personalized*. Dengan demikian, metode HF nantinya mampu menghasilkan rekomendasi *item* berdasarkan kesamaan konten *item* secara *personalized*, dan kemungkinan model MF pada metode CF yang digunakan ini mampu mereduksi error prediksi lebih kecil dibandingkan dengan metode tradisional CF berbasis *memory*, sehingga harapannya metode HF bisa meningkatkan kualitas pada sistem rekomendasi.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat diketahui bahwa pokok bahasan penelitian yaitu:

- a. Apa metode Hybrid CF dapat mereduksi error prediksi lebih kecil dari metode Collaborative Filtering berbasis *memory*?

- b. Berapa penurunan error prediksi yang dihasilkan metode Hybrid CF berdasarkan evaluasi RMSE dan MAPE?
- c. Apa metode Hybrid Filtering dapat memberikan rekomendasi *item* yang *personalized* dan terarah kontennya?

1.3. Batasan Masalah

Agar penelitian ini terarah dan pembahasan relevan maka diberi batasan-batasan dan ruang lingkup sebagai berikut:

- a. Penelitian ini hanya membahas performa metode dalam sistem rekomendasi yaitu CBF, CF, dan HF.
- b. Penelitian ini menggunakan MovieLens 1M Dataset dari GroupLens yang telah diterbitkan pada tahun 2003.
- c. Konten film yang digunakan dari Dataset MovieLens saat tahap *Text Preprocessing* hanya menggunakan kata-kata bahasa Inggris.
- d. Pada tahap *Text Preprocessing* menggunakan fitur *word_tokenize* dari *library Natural Language Toolkit (nltk)* versi 3.5, lalu proses *Feature Text Extraction* menggunakan *library* dari Scikit-Learn versi 0.24.1.
- e. Proses *training* metode CF berbasis-memory dan metode HF menggunakan *library* dari Scikit-Learn versi 0.24.1 dan *library Surprise* versi 1.1.0.
- f. Penelitian ini menggunakan bahasa pemrograman Python untuk membantu proses *preprocessing data*, *training data*, *testing data*, visualisasi data, dan evaluasi performa metode.
- g. Penelitian ini akan ditampilkan *output* perbedaan hasil rekomendasi *item* yang *personalized* dan tidak *personalized*.

- h. Penelitian ini menggunakan metode RMSE dan MAPE untuk mengukur nilai error prediksi dari metode Hybrid CF dan CF berbasis memory.
- i. RMSE tidak ada *range* atau batas nilainya, sedangkan MAPE ditetapkan batas nilai dari 1 – 100 %, jadi parameter baik atau tidaknya performa metode yang akan dibandingkan pada penelitian ini dari metode yang mampu menghasilkan error prediksi terkecil berdasarkan nilai RMSE dan MAPE.
- j. Penelitian ini tidak sampai membuat *Graphical User Interface* (GUI) sistem rekomendasi sampai jadi, tetapi hanya implementasi logika dan evaluasi error prediksi metode HF pada Sistem Rekomendasi.
- k. Penelitian ini ditujukan kepada platform penyedia layanan streaming film berbasis *website* yang ingin membangun sebuah sistem rekomendasi.
- l. Tahap evaluasi metode dilakukan sebanyak 10 pada kondisi data *non sparsity* dan 7 kali iterasi pada kondisi data *sparsity*, jumlah iterasi ditentukan berdasarkan hasil filter jumlah data *rating* pada kedua kondisi.
- m. Pada tahap evaluasi kondisi data *sparsity*, *non sparsity* akan ditampilkan perbandingan hasil error prediksi dari metode Hybrid CF dan CF berbasis memory.
- n. Proses *training* metode CBF masih bisa ditambahkan dengan data konten lain seperti Durasi Film, Usia, dan *Gender* penonton, namun karena keterbatasan atribut dari dataset yang didapatkan maka hanya menggunakan Metadata yang terdiri dari kombinasi atribut Genres, Cast, Keywords, dan Director untuk menghasilkan rekomendasi film berdasarkan kemiripan konten.

- o. Konteks rekomendasi film yang terarah ketika selesai proses *training* metode CBF untuk menghasilkan rekomendasi *item* pertama yang memiliki kemiripan data konten yang diambil berdasarkan atribut Metadata.

1.4. Tujuan Penelitian

Tujuan dilakukannya penelitian ini yaitu:

- a. Memperbaiki metode tradisional CF berbasis memory pada penelitian sebelumnya dengan mereduksi error prediksi menggunakan metode Hybrid CF sehingga mampu meningkatkan kualitas sistem rekomendasi.
- b. Menghasilkan rekomendasi *item* yang *personalized* untuk mengatasi kelemahan pada metode CBF yang belum *personalized*.
- c. Untuk mengetahui penurunan error prediksi yang dihasilkan oleh metode Hybrid Filtering berdasarkan evaluasi RMSE dan MAPE.
- d. Sebagai syarat untuk kelulusan masa *study* jenjang S2 di Universitas Amikom Yogyakarta.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi:

- a. Pembaca, menjadi salah satu bacaan untuk menambah ilmu dan wawasan tentang penelitian mengenai Sistem Rekomendasi.
- b. Peneliti lain, menjadi salah satu referensi untuk menambah ilmu dan wawasan untuk mencoba penelitian tentang sistem rekomendasi, atau sebagai referensi untuk mengembangkan penelitian ini menjadi lebih baik.
- c. Platform penyedia layanan streaming film berbasis *website*, penelitian ini dapat menjadi salah satu metode Sistem Rekomendasi yang dapat diimplementasikan

menjadi fitur pada sebuah *platform* yang baru atau sudah dibangun untuk memudahkan penggunaanya dalam menemukan film yang mungkin disukai berdasarkan rekomendasi *item* yang dihasilkan oleh sistem.

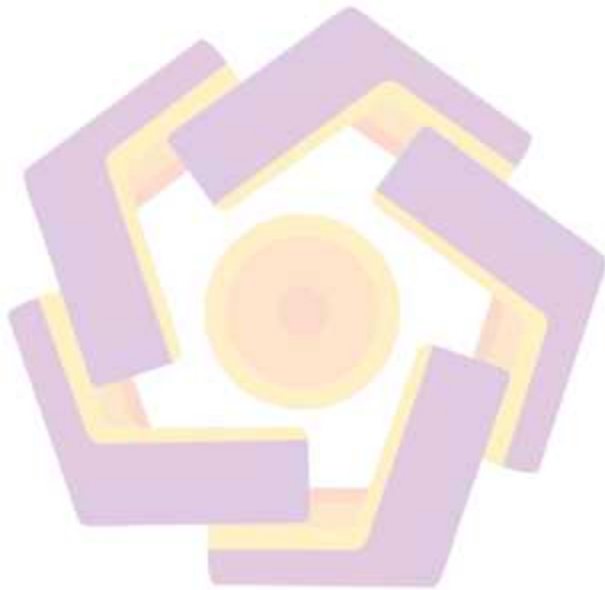
- d. Pada platform penyedia layanan streaming film berbasis *website* yang baru dan belum memiliki data *rating* film sama sekali, metode HF ini tetap bisa digunakan untuk mengatasi hal tersebut, karena metode ini mengkombinasikan metode CBF yang mampu memberikan rekomendasi *item* berdasarkan data konten dari *item* yang pernah dilihat oleh pengguna tanpa membutuhkan data *rating item*, walaupun hasil rekomendasi ini belum *personalized* setidaknya bisa digunakan sementara sambil menunggu *website* berkembang hingga memiliki banyak *rating* untuk menghasilkan rekomendasi *item* yang *personalized*.
- e. Penulis, mendapatkan banyak ilmu, wawasan, pengalaman baru, dan dapat mengimplementasikan ilmu yang sudah dipelajari selama masa kuliah menjadi sebuah penelitian yang InsyaAllah bermanfaat bagi orang lain.

1.6. Hipotesis

Berdasarkan penelitian ini didapatkan hipotesa sebagai berikut:

- a. Mampu meningkatkan kualitas sistem rekomendasi berdasarkan penurunan nilai error prediksi yang nantinya dihasilkan oleh metode Hybrid CF lalu dibandingkan performanya dengan metode CF berbasis memory.
- b. Mampu memberikan *output* berupa rekomendasi *item* yang *personalized* untuk mengatasi kelemahan pada metode CBF yang masih belum *personalized*.
- c. Memberikan hasil evaluasi perbandingan metode yang dilakukan sebanyak 10 sampai 7 kali iterasi menggunakan metode RMSE, MAPE pada kondisi data

sparsity dan *non sparsity*, setiap evaluasi akan ditampilkan perbandingan nilai RMSE, MAPE, lalu ditampilkan rata-rata nilai RMSE, MAPE kedua metode setelah itu ditampilkan hasil penurunan error prediksi dari metode Hybrid CF.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian terdahulu yang dijadikan acuan dan tinjauan pustaka pada penelitian ini diantaranya:

Wijaya & Alfian (2018) yang melakukan penelitian tentang sistem rekomendasi laptop menggunakan metode CF, CBF, dan HF. Pada metode CF peneliti menggunakan algoritma Adjusted Cosine Similarity untuk menghitung kemiripan antar pengguna dan algoritma Weighted Sum untuk prediksi *rating item*, sedangkan pada metode CBF peneliti menggunakan algoritma TF-IDF untuk mencari kemiripan konten dari *item*.

Tujuan penelitian Wijaya & Alfian (2018) adalah untuk membuat sistem rekomendasi yang dapat memberikan rekomendasi laptop sesuai dengan ketertarikan dan kebutuhan dari preferensi pengguna. Pada penelitian ini hanya dilakukan pembuktian mengenai waktu eksekusi tercepat dari ketiga metode yang digunakan, hasilnya metode CBF memiliki waktu eksekusi lebih cepat dibandingkan dengan metode CF dan HF.

Terdapat beberapa kekurangan dari penelitian Wijaya & Alfian (2018). Pertama, tidak dicantumkan perbedaan waktu eksekusi dari masing-masing metode. Kedua, tidak dilakukan pengujian error prediksi. Ketiga, dataset yang digunakan hanya sedikit sehingga wajar jika waktu eksekusinya cepat. Terakhir, algoritma

Weighted Sum tidak cocok untuk digunakan pada pengguna yang hanya memiliki *rating* yang sedikit karena dapat menghasilkan error prediksi yang tinggi.

Selanjutnya Prasetyo et al (2019) juga melakukan penelitian tentang sistem rekomendasi aksesoris *Smartphone* menggunakan metode CF berbasis *item*. Peneliti menggunakan algoritma Pearson Correlation Based Similarity pada metode CF untuk menghitung kemiripan *rating* antar *item*, setelah itu dilakukan perhitungan prediksi *rating item* menggunakan algoritma Weighted Average of Deviation, hasil prediksi akan dievaluasi menggunakan metode Mean Absolute Error (MAE) untuk mengetahui error prediksi yang dihasilkan.

Tujuan Prasetyo et al (2019) melakukan penelitian ini untuk membantu Toko Flazzstore memberikan rekomendasi *casing Smartphone* kepada calon pembelinya. Setelah dilakukan evaluasi error prediksi, metode peneliti menghasilkan nilai MAE sebesar 0,572039 dengan waktu eksekusi selama 6,4 detik.

Beberapa kekurangan yang ditemukan dari penelitian Prasetyo et al (2019). Pertama, tidak dicantumkan berapa jumlah data *training* dan data *testing* pada proses evaluasi error prediksi. Kedua, metode CF hanya memberikan rekomendasi *item* berdasarkan data *rating* dari pengguna dan mengabaikan data dari aktivitas pengguna saat melihat *item* tertentu, sehingga jika terdapat kondisi pengguna belum pernah memberikan *rating item* maka sistem tidak bisa memberikan rekomendasi *item* kepada pengguna.

Terdapat peneliti Rahmawati et al (2018) yang menggabungkan metode Knowledge Based (KB) dan CF untuk dilakukan analisis dan implementasinya pada

Sistem Rekomendasi. Pada metode CF berbasis pengguna peneliti menggunakan algoritma Pearson Correlation Based Similarity untuk menghitung kemiripan antar pengguna, kemudian hasil perhitungan digunakan untuk prediksi tiap *item* menggunakan algoritma Weighted Average of Deviation, sedangkan pada metode KB digunakan untuk mencari pengaruh fitur dari pekerjaan terhadap kebutuhan dan preferensi pribadi pengguna. Setelah mendapatkan rekomendasi pekerjaan akan dilakukan evaluasi error prediksi yang dihasilkan menggunakan metode Root Mean Square Error (RMSE).

Rahmawati et al (2018) melakukan penelitian ini dengan tujuan untuk membantu memberikan rekomendasi pekerjaan kepada orang yang sedang mencari pekerjaan sesuai dengan jenis pekerjaan yang diinginkan dan memberikan rekomendasi karyawan yang cocok pada sebuah perusahaan yang sedang mencari karyawan baru.

Pada metode CF peneliti Rahmawati et al (2018) menggunakan data interaksi berdasarkan *social aperture* yang diberikan, *social aperture* dibagi menjadi 2 yaitu *moderate* dan *liberal*, *moderate* artinya pengguna lebih cenderung memilih rekomendasi sebanyak 75% berdasarkan preferensi pribadinya, 25% berdasarkan hasil rekomendasi dari pengguna lain, lalu *liberal* artinya menggabungkan hasil rekomendasi dengan nilai seimbang yaitu 50% dari preferensi pribadi dan 50% berdasarkan rekomendasi dari pengguna lain.

Proses evaluasi hasil penelitian yang dilakukan oleh Rahmawati et al (2018) menggunakan metode K-Fold Cross Validation, dengan nilai $k=10$, lalu jumlah data interaksi yang di *training* sebanyak 41381 dan data *testing* 4034. Setelah dilakukan

10 kali iterasi, metode KB menghasilkan rata-rata error prediksi sebesar 0,357, CF 0,526, Hybrid Liberal 0,449, dan Hybrid Moderate 0,347. Nilai RMSE yang terkecil didapatkan pada metode Hybrid Moderate sebesar 0,347. Sedangkan pada evaluasi dengan F1 Measure untuk rekomendasi Top N, F1 *score* terbaik didapatkan oleh metode Hybrid Liberal sebesar 0,347 dengan nilai $n=50$.

Terdapat kekurangan yang ditemukan dari penelitian Rahmawati et al (2018). Pertama, dataset yang digunakan mengandung data pengguna buatan dan data interaksi tidak berasal dari pengguna sebenarnya. Kedua, hasil F1 score pada rekomendasi belum maksimal karena pada data *testing* banyak pekerjaan yang seharusnya direkomendasikan tetapi memiliki peringkat lebih dari n , hal ini disebabkan karena banyak informasi pengguna yang kosong sehingga mempengaruhi proses inferensi pengguna.

Dalam memperbaiki error prediksi yang dihasilkan metode tradisional CF berbasis memory pada Sistem Rekomendasi, Hu et al (2019) membuat algoritma baru dengan nama RMIF yang diterapkan pada metode CF, algoritma ini memanfaatkan banyak data masukan secara implisit. Pada metode CF peneliti menggunakan algoritma Pearson Correlation Based Similarity (PCC) dan Vector Space Similarity (VSS) untuk menghitung kemiripan *rating* antar *item*, kemudian hasilnya digunakan untuk prediksi *rating* menggunakan algoritma yang dirancang oleh peneliti untuk menghasilkan rekomendasi *item*. Hasil prediksi akan dievaluasi menggunakan metode Mean Absolute Error (MAE) dan Root Mean Square Error (RMSE) untuk mengetahui error prediksi yang dihasilkan dari algoritma peneliti.

Tujuan Hu et al (2019) membuat algoritma baru untuk mereduksi error prediksi dari metode CF berbasis memory dengan memanfaatkan umpan balik atau masukan secara implisit dari pengguna berupa kesamaan antar pengguna, rating *item*, dan perilaku positif pengguna yang didapat dari konversi data *rating item* yang diberikan, jika pengguna memberikan nilai *rating* tidak kurang dari 3 dari nilai maksimal 5 yang tersedia maka perilaku pengguna akan dianggap positif, dengan memanfaatkan umpan balik secara implisit dari pengguna dapat mengatasi error prediksi pada kondisi data *sparsity*.

Hu et al (2019) melakukan proses evaluasi menggunakan 2 dataset Movielens berjumlah 100.000 *rating item* dan 1.000.000 *rating item*, dataset tersebut dibagi menjadi 80% untuk data *training* dan 20% untuk data *testing*. Evaluasi dilakukan sebanyak 5 kali iterasi dan error prediksi yang dihasilkan dari algoritma RMIF akan dibandingkan dengan 7 algoritma lainnya. Setelah dilakukan evaluasi menggunakan data pengguna yang memberikan kurang dari 10 *rating item*, hasilnya metode RMIF mendapatkan nilai MAE dan RMSE terkecil dibandingkan dengan algoritma lainnya, pada proses evaluasi menggunakan dataset Movielens berjumlah 100.000 mendapatkan nilai MSE sebesar 0,703, dan RMSE sebesar 0,898, lalu untuk dataset Movielens berjumlah 1.000.000 mendapatkan nilai MSE sebesar 0,670, dan RMSE sebesar 0,851.

Terdapat kekurangan pada penelitian yang dilakukan oleh Hu et al (2019), yaitu data umpan balik dari perilaku pengguna didapat dari konversi nilai *rating item* yang diberikan, jika hanya menggunakan data umpan balik yang berasal dari *rating*, pada saat kondisi sistem rekomendasi baru yang hanya memiliki data

pengguna tetapi tidak memiliki data *rating* sama sekali maka algoritma RMIF ini tidak bisa memberikan rekomendasi *item* yang akurat kepada pengguna karena kekurangan 2 sumber data untuk sistem yaitu *rating item*, dan perilaku pengguna.

Selain membuat algoritma baru untuk mengatasi error prediksi pada metode tradisional CF berbasis memory, terdapat peneliti Hwangbo et al (2018) yang membuat algoritma baru pada metode CF dengan nama K-Recsys untuk memperbaiki hasil rekomendasi *item* dari metode lama yang digunakan pada *e-commerce retail fashion*. Peneliti memanfaatkan algoritma subproses perhitungan nilai asosiasi waktu diskon yang memanfaatkan data *rating item* atau klik *item* dari pengguna, algoritma K-Means Clustering juga digunakan untuk mengelompokkan data bulanan penjualan *item*, lalu untuk mencari kandidat *item* yang akan direkomendasikan menggunakan algoritma Association Score berdasarkan perhitungan nilai *similarity item* dari data klik atau *rating item* pengguna, data klik Association Score dihitung dari dataset yang diambil dari *online shopping* mall dan data pembelian Association Score dihitung dari dataset yang diambil dari *offline shopping* mall, lalu keduanya digabung menjadi algoritma Recommendation Score untuk memberikan rekomendasi *item*.

Hwangbo et al (2018) membagi hasil rekomendasi *item* menjadi 2 yaitu rekomendasi *item* pengganti yang berisi *item* dari kategori yang sama, dan rekomendasi *item* pelengkap yang berisi *item* dari kategori yang berbeda. Kemudian dilakukan evaluasi menggunakan metode Mean Absolute Error (MAE) dan Root Mean Square Error (RMSE) untuk mengetahui error prediksi yang dihasilkan.

Hwangbo et al (2018) melakukan penelitian ini dengan tujuan untuk memperluas penggunaan metode CF berbasis *item* dengan membuat algoritma baru dengan nama K-RecSys yang memanfaatkan data klik pengguna terhadap *item* yang dilihat saat mengunjungi toko *online* dan data penjualan *item* yang didapatkan dari toko *offline* untuk memberikan rekomendasi 2 jenis rekomendasi *item*.

Peneliti Hwangbo et al (2018) melakukan proses evaluasi dengan mengimplementasikan metode K-RecSys pada perusahaan K's *shopping mall* yang sudah mempunyai sistem rekomendasi. Proses *testing* dilakukan selama 3 minggu untuk membandingkan performa dari sistem rekomendasi milik perusahaan dengan K-RecSys milik peneliti. Pengguna yang diberikan rekomendasi oleh sistem perusahaan dinamakan *Control Group* sedangkan oleh K-RecSys dinamakan *Experimental Group*. Pengguna diberikan rekomendasi 40 *item* oleh sistem perusahaan, sedangkan dari K-RecSys 20 *item* pengganti, dan 20 *item* pelengkap. Jika pengguna klik atau membeli *item* dari rekomendasi *item* yang diberikan maka inilah yang akan dievaluasi, yaitu dampak rekomendasi *item* pengganti dan pelengkap dari metode K-RecSys dalam membantu penjualan *item* dan klik *item*.

Pada proses *testing* pertama, Hwangbo et al (2018) mengamati jumlah data klik *item* pada sistem rekomendasi selama 3 minggu, dari hasil pengamatan terdapat data klik sebanyak 532,598 pada *Control Group*, dan 543,796 pada *Experimental Group*. Pengguna yang klik *item* berdasarkan rekomendasi *item* dari *Control Group* sebesar 5,8%, sedangkan pada *Experimental Group* 9,9%, artinya performa metode K-RecSys dalam memberikan rekomendasi *item* lebih baik dibanding dengan sistem rekomendasi perusahaan.

Pada proses *testing* kedua, Hwangbo et al (2018) menginvestigasi dampak sistem rekomendasi dalam membantu penjualan *item*. Selama 3 minggu terkumpul 14 data penjualan *online* sebanyak 7476 *item*, data penjualan yang berasal dari rekomendasi *item* *Control Group* sebesar 8,9%, sedangkan dari *Experimental Group* sebesar 12,3 %, walaupun hasil penjualan ini tidak konsisten selama proses evaluasi, tetapi metode K-RecSys tetap memberikan dampak yang besar dalam penjualan *item*.

Terdapat kekurangan dari penelitian yang dilakukan Hwangbo et al (2018). Pertama, tidak dilakukan evaluasi error prediksi dari hasil algoritma K-RecSys jadi keakuratannya masih diragukan. Kedua, tidak diketahui metode sebelumnya yang digunakan oleh sistem rekomendasi pada perusahaan K's *shopping* mall untuk perbandingan, jika sistem tidak menggunakan metode sistem rekomendasi maka perbandingan performa dengan metode K-RecSys tidak adil.

Terdapat peneliti Feng et al (2020) yang membuat algoritma baru dengan nama FPMF pada metode CF berbasis model MF untuk mereduksi error prediksi dari metode tradisional CF berbasis memory pada Sistem Rekomendasi. Peneliti menggunakan model Fusion yang mempertimbangkan kemiripan *multi* faktor dan informasi *rating* global yang diusulkan dalam algoritma Matrix Factorization. Peneliti juga menggunakan algoritma baru yang dinamakan Extreme Behavior Similarity Measure dan Linier Similarity untuk mengukur *multi factor similarity* yang menangkap korelasi linier dan non linier antara pengguna dari perilaku yang *extreme* dalam memberikan *rating item*, setelah didapatkan *multi factor similarity (neighbor matrix)* lalu digabungkan dengan *original rating matrix* yang dinamakan

proses Fusing, kemudian jadilah *matrix* prediksi yang digunakan untuk rekomendasi *item* berdasarkan representasi faktor laten dari pengguna dan *item* kepada pengguna secara *online*. Hasil prediksi dari metode yang diusulkan peneliti dievaluasi menggunakan metode Mean Absolute Error (MAE) dan Root Mean Square Error (RMSE) untuk mengetahui error prediksi menggunakan data *sparse* dan *extreme sparse*.

Tujuan Feng et al (2020) melakukan penelitian ini untuk mengatasi permasalahan error prediksi pada kondisi data *sparse* dengan membuat algoritma baru FPMF pada metode CF berbasis model MF, yaitu kondisi dimana pada Sistem Rekomendasi terdapat pengguna yang memiliki data *rating item* yang tidak seimbang dengan jumlah keseluruhan *item*.

Proses evaluasi error prediksi yang dilakukan Feng et al (2020) menggunakan dataset film MovieLens (MI-100K) sebanyak 100.000 data *rating*, MovieLens *Latest Small* (MILatest-Small), FilmTrust, dan CiaoDVD. Dataset MovieLens mempunyai skala *rating item* dari 1 – 5, Dataset FilmTrust 0,5 – 4, dan Dataset CiaoDVD 1 – 5. Tiga dataset pertama dianggap sebagai data *sparse*, lalu dataset keempat dianggap sebagai data *extreme sparse*. Dataset dibagi menjadi data *training* sebanyak 80 % dan *testing* 20 %, lalu proses evaluasi dilakukan sebanyak 5 kali iterasi. Sebelum dievaluasi, karena tujuan peneliti untuk memecahkan masalah *sparse* data, maka semua dataset akan dihapus data *rating item* secara acak sebanyak 20%, 40%, 60%, dan 80%.

Setelah proses evaluasi selesai dilakukan oleh Feng et al (2020), hasil error prediksi dibandingkan dengan 7 algoritma lainnya, dari hasil evaluasi error prediksi

menunjukkan bahwa algoritma peneliti mendapatkan nilai RMSE terkecil dibanding metode lain sebesar 0,9307 (MI-100K), 0,8057 (FilmTrust), dan 1,0084 (CiaoDVD), tetapi pada dataset MI-Latest-Small mendapatkan nilai RMSE sebesar 0,9018, dimana nilai ini lebih besar dibandingkan dengan metode BPMF dengan nilai RMSE lebih kecil yaitu 0,8905. Lalu untuk hasil evaluasi MAE metode FPMF mengungguli metode lain yaitu 0,7318 (MI-100K), 0,6238 (FilmTrust), dan 0,7835 (CiaoDVD), tetapi pada dataset MI-Latest-Small sebesar 0,6960, dimana nilai ini lebih besar dibanding metode BPMF dengan nilai MAE lebih kecil 0,6821, jadi metode yang diusulkan peneliti dapat mengatasi permasalahan data *sparsity* pada Sistem Rekomendasi.

Terdapat kekurangan dari algoritma FPMF yang dibuat oleh Feng et al (2020), jika hanya menggunakan data *rating* sebagai *input* untuk menghasilkan rekomendasi *item*, kelemahannya pada saat mengalami kondisi sistem baru yang sudah memiliki banyak *item* tetapi semua *item* belum memiliki *rating* dari penggunaannya, otomatis sistem tidak bisa memberikan rekomendasi *item* yang akurat (*Cold Start Problem*).

Kombinasi 2 metode pada penelitian Sistem Rekomendasi juga dilakukan oleh Geetha et al (2018) tentang sistem rekomendasi film menggunakan metode HF. Peneliti mengkombinasikan metode CF berbasis memory dan CBF. Pada metode CF peneliti menggunakan algoritma Pearson Correlation Coefficient (PCC), kemudian hasil dari perhitungan kemiripan antar *item* akan dilakukan prediksi untuk merekomendasi film menggunakan algoritma Weighted Averages of Deviations, lalu pada metode CBF peneliti menggunakan algoritma K-Means

Clustering untuk mengelompokkan kemiripan film berdasarkan *rating* film bukan dari konten film dengan algoritma The Euclidean Distance. Tujuan dari penelitian ini untuk melengkapi kekurangan dari jika hanya menggunakan metode CF atau CBF sehingga dapat meningkatkan akurasi dari sistem rekomendasi.

Pada proses implementasi metode, Geetha et al (2018) hanya menggunakan 2 data pengguna beserta *rating* dari dataset, data *rating* dari kedua pengguna akan *dicluster* berdasarkan *rating* film menggunakan algoritma K-Means, lalu dicari kemiripan *rating* antar pengguna yang diberikan menggunakan algoritma PC berdasarkan nilai korelasi yang tinggi antar pengguna terhadap suatu film. Selanjutnya prediksi *rating* menggunakan algoritma Weighted Averages of Deviations untuk menghasilkan rekomendasi film. Hasil dari implementasi ini mendapatkan *feedback* positif dari pengguna.

Terdapat beberapa kekurangan dari penelitian yang dilakukan oleh Geetha et al (2018). Pertama, metode CBF yang *dicluster* bukan berdasarkan konten film tetapi dari *rating* film, jika terdapat kasus film belum mempunyai *rating* maka data tidak bisa *dicluster*. Kedua, pada metode CF berbasis memory jika terdapat data *rating* film baru dalam *database* maka model tersebut harus *ditraining* ulang untuk dapat memprediksi *rating* dari data baru sehingga film bisa direkomendasikan. Ketiga, tujuan penelitian yang tidak sesuai dengan hasil penelitian yang dilakukan, penelitian ini bertujuan untuk mereduksi error prediksi dari sistem rekomendasi tetapi kenyataannya saat pembuktian peneliti tidak melakukan pengujian error prediksi dari metode HF yang digunakan tetapi hanya memberikan contoh implementasi metode dengan hanya menggunakan data kecil dan melakukan

evaluasi yang tidak *formal* kepada pengguna untuk mendapatkan *feedback*, walaupun *feedbacknya* positif tetapi belum menjawab tujuan dari penelitian ini.

Dari penelitian yang sudah diuraikan diatas, kebanyakan peneliti menggunakan metode yang hanya memanfaatkan data *rating item* pada modelnya. Perbedaan pada penelitian ini adalah data yang digunakan tidak hanya *rating item* tetapi juga informasi konten *item* untuk menghasilkan rekomendasi *personalized* yang terarah kontennya, dan mampu mereduksi error prediksi dari metode tradisional CF berbasis *memory* pada penelitian sebelumnya menggunakan model Matrix Factorization, sehingga penelitian ini bisa memberikan kontribusi untuk meningkatkan kualitas rekomendasi pada Sistem Rekomendasi menggunakan metode HF.



2.2. Keaslian Penelitian

Implementasi metode HF pada Sistem Rekomendasi untuk meningkatkan kualitas rekomendasi pada penelitian ini dilakukan dengan sebenar-benarnya tanpa adanya unsur plagiarisme, oleh karena itu pada Tabel 2.1 dibuat *review* perbandingan posisi penelitian yang sedang dilakukan ini dengan penelitian lain yang terkait tentang Sistem Rekomendasi.

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering	Wijaya & Alfian (2018)	Membuat sistem rekomendasi yang dapat memberikan rekomendasi laptop sesuai dengan ketertarikan dan kebutuhan dari referensi pengguna.	Metode CBF memiliki waktu eksekusi lebih cepat dibandingkan dengan metode CF dan HF.	<ul style="list-style-type: none"> • Tidak dicantumkan berapa perbedaan waktu eksekusi dari masing-masing metode. • Tidak dilakukan pengujian error prediksi dari masing-masing metode. • Dataset yang digunakan hanya sedikit sehingga wajar jika waktu eksekusinya cepat. • Algoritma Weighted Sum tidak cocok untuk digunakan pada pengguna yang hanya memiliki <i>rating</i> yang sedikit, karena membuat hasil 	Peneliti hanya melakukan uji coba metode CF, CBF, dan HF berbasis memory yang hasilnya untuk membandingkan performa waktu eksekusi dalam memberikan rekomendasi tidak untuk mengukur performa error prediksi atau perbandingan hasil rekomendasi, sedangkan penelitian kami menggunakan metode HF berbasis model MF pastinya dengan algoritma yang berbeda yang hasilnya nanti untuk mereduksi error prediksi metode CF berbasis memory dari penelitian ini pada kondisi data <i>sparsity</i> dan non <i>sparsity</i> . Selain itu penelitian kami tidak

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
					rekomendasinya tidak akurat.	hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi <i>item</i> yang dihasilkan jika hanya menggunakan metode CBF.
2	Implementasi Metode Item Based Collaborative Filtering dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone	Prasetyo et al (2019)	Membantu Flazzstore yang merupakan toko penjualan casing smartphone untuk memberikan rekomendasi <i>item</i> kepada calon pembelinya	Hasil nilai MAE mendapatkan nilai sebesar 0,572039 dengan waktu eksekusi selama 6,4 detik.	<ul style="list-style-type: none"> • Tidak dicantumkan berapa jumlah data <i>training</i> dan data <i>testing</i> pada proses evaluasi prediksi sehingga menghasilkan nilai MAE sebesar 0,572039. • Metode CF mengabaikan data-data dari aktivitas pengguna saat melihat <i>item</i> tertentu, sehingga jika terdapat kondisi pengguna belum pernah memberikan <i>rating item</i> maka sistem tidak bisa memberikan rekomendasi <i>item</i> kepada pengguna. 	Peneliti hanya menggunakan metode CF berbasis memory untuk membuat sistem rekomendasi pada toko Flazzstore yang hasilnya diuji dengan MAE, sedangkan penelitian kami menggunakan metode HF berbasis model MF dengan algoritma yang berbeda yang hasilnya nanti untuk mereduksi error prediksi dari metode CF berbasis memory yang digunakan oleh peneliti ini pada kondisi data <i>sparsity</i> dan non <i>sparsity</i> , pengujiannya tidak menggunakan MAE tetapi dengan RMSE dan MAPE. Selain itu penelitian kami tidak hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi <i>item</i>

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						yang dihasilkan jika hanya menggunakan metode CBF.
3	Analisis dan Implementasi Pendekatan Hybrid untuk Sistem Rekomendasi dengan Metode Knowledge Based Recommender System dan Collaborative Filtering	Rahmawati et al (2018)	<ul style="list-style-type: none"> Membantu memberikan rekomendasi pekerjaan kepada pelamar kerja sesuai jenis pekerjaan yang diinginkan. Memberikan rekomendasi karyawan pada perusahaan yang sedang mencari karyawan baru yang sesuai. 	<ul style="list-style-type: none"> Nilai RMSE Metode Knowledge Based 0,357, CF 0,526, Hybrid Liberal 0,449, dan Hybrid Moderate 0,347. Rata-rata nilai RMSE terkecil pada metode Hybrid Moderate sebesar 0,347. F1 score terbaik didapatkan oleh metode Hybrid Liberal sebesar 0,347 dengan nilai $n=50$. 	<ul style="list-style-type: none"> Dataset yang digunakan mengandung data pengguna buatan dan data interaksi tidak berasal dari pengguna sebenarnya. Hasil F1 score pada rekomendasi belum maksimal karena pada data <i>testing</i> banyak pekerjaan yang seharusnya direkomendasikan tetapi memiliki ranking lebih dari n, hal ini karena banyak informasi pengguna yang kosong sehingga mempengaruhi proses inferensi pengguna. 	Peneliti menggunakan metode HF dari kombinasi metode KB dan CF berbasis memory untuk menghasilkan sistem rekomendasi pekerjaan dan rekomendasi karyawan, yang hasilnya diuji dengan RMSE, sedangkan penelitian kami menggunakan metode HF dengan kombinasi metode yang berbeda yaitu dengan metode CBF dan CF berbasis model MF pastinya dengan algoritma yang berbeda, hasilnya nanti untuk mereduksi error prediksi dari metode CF berbasis memory yang digunakan oleh peneliti ini pada kondisi data <i>sparsity</i> dan non <i>sparsity</i> , pengujiannya tidak hanya menggunakan RMSE tetapi juga MAPE. Selain itu penelitian kami tidak hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi <i>item</i> yang

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						dihasilkan jika hanya menggunakan metode CBF.
4	Movie collaborative filtering with multiplex implicit feedbacks	Hu et al (2019)	<ul style="list-style-type: none"> Mengurangi error prediksi dari metode CF dengan memanfaatkan umpan balik atau masukan secara implisit dari pengguna. Mengatasi data <i>sparsity</i> dari <i>rating item</i>. 	<ul style="list-style-type: none"> Metode RMIF mendapatkan nilai MAE dan RMSE terkecil dibandingkan dengan 7 algoritma lain. Nilai MSE 0,703 dan nilai RMSE 0,898 pada dataset MovieLens 100.000 data rating. Nilai MSE 0,670 dan nilai RMSE 0,851 pada dataset MovieLens 1.000.000 data rating. 	Data umpan balik dari perilaku pengguna didapat dari konversi nilai <i>rating item</i> yang diberikan, jika sistem baru yang tidak memiliki data <i>rating</i> sama sekali maka algoritma RMIF tidak dapat memberikan rekomendasi yang akurat kepada pengguna karena tidak terdapat data <i>rating item</i> , dan perilaku pengguna.	Peneliti mengusulkan metode baru RMIF yang merupakan perbaikan dari metode CF berbasis memory untuk mereduksi error prediksi dari metode CF berbasis memory pada kondisi data <i>sparsity</i> , sedangkan pada penelitian ini menggunakan kombinasi metode CBF dan CF berbasis model MF dengan algoritma yang berbeda yang hasilnya nanti juga untuk mereduksi error prediksi dari metode CF berbasis memory pada peneliti sebelumnya tidak hanya pada kondisi data <i>sparsity</i> tetapi juga pada kondisi non <i>sparsity</i> , pengujiannya tidak hanya menggunakan RMSE tetapi juga MAPE. Selain itu penelitian kami tidak hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi <i>item</i> yang

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						dihasilkan jika hanya menggunakan metode CBF.
5	Recommendation system development for fashion retail ecommerce	Hwangbo et al (2018)	Membuat algoritma baru dengan nama KRecSys untuk memperbaiki sistem rekomendasi larva yang terdapat pada perusahaan <i>fashion retail e-commerce</i>	<ul style="list-style-type: none"> Menggunakan data klik berjumlah 532.598 pada <i>Control Group</i>, dan 543.796 pada <i>Experimental Group</i>. Pengguna yang klik <i>item</i> berdasarkan rekomendasi <i>item</i> dari <i>Control Group</i> sebesar 5,8%, sedangkan pada <i>Experimental Group</i> 9,9%, performa metode K-RecSys dalam memberikan rekomendasi lebih baik dibanding dengan sistem rekomendasi perusahaan. Menggunakan data penjualan <i>online</i> sebanyak 7476 <i>item</i>, 	<ul style="list-style-type: none"> Tidak dilakukan evaluasi error prediksi dari rekomendasi <i>item</i> menggunakan metode K-RecSys jadi keakuratannya masih diragukan Tidak diketahui metode sistem rekomendasi pada perusahaan K's <i>shopping mall</i>, jika sistem tidak menggunakan metode sistem rekomendasi maka perbandingan performa dengan metode K-RecSys tidak adil. 	Peneliti mengusulkan metode baru KrecSys pada sistem rekomendasi untuk memperbaiki hasil rekomendasi <i>item</i> dari sistem lama pada perusahaan <i>fashion retail e-commerce</i> dengan tujuan untuk meningkatkan penjualan, sedangkan penelitian kami menggunakan metode HF berbasis model MF dengan algoritma yang berbeda yang hasilnya nanti juga untuk memperbaiki hasil rekomendasi <i>item</i> yang belum <i>personalized</i> dari metode CBF pada peneliti sebelumnya sekaligus untuk memperbaiki penelitian ini yang hasil rekomendasi <i>item</i> nya tidak efektif. Selain itu penelitian kami tidak hanya untuk memperbaiki hasil rekomendasi <i>item</i> tetapi juga untuk mereduksi error prediksi metode CF berbasis memory dari penelitian

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				data penjualan yang berasal dari rekomendasi <i>item Control Group</i> sebesar 8,9%, sedangkan dari <i>Experimental Group</i> sebesar 12,3 %, metode KrecSys tetap memberikan dampak yang besar dalam penjualan <i>item</i> .		sebelumnya pada kondisi data <i>sparsity</i> dan non <i>sparsity</i> .
6	A fusion collaborative filtering method for sparse data in recommender systems	Feng et al (2020)	Mengatasi permasalahan <i>sparsity</i> data pada pengguna yang memiliki jumlah <i>rating item</i> yang tidak seimbang dengan jumlah keseluruhan <i>item</i> pada sistem rekomendasi.	<ul style="list-style-type: none"> Metode peneliti menghasilkan nilai RMSE terkecil dibanding metode lain sebesar 0,9307 (MI-100K), 0,8057 (FilmTrust), dan 1,0084(CiaoDVD). Pada MI -Latest – Small nilai RMSE sebesar 0,9018, nilai ini lebih besar dibandingkan dengan metode 	Metode FPMF yang diusulkan peneliti hanya memanfaatkan data <i>rating</i> sebagai <i>input</i> untuk menghasilkan rekomendasi <i>item</i> , tanpa data <i>rating item</i> metode ini tidak bisa menghasilkan rekomendasi <i>item (Cold Start Problem)</i>	Peneliti mengusulkan metode baru FPMF pada sistem rekomendasi untuk mereduksi error prediksi dari metode CF berbasis memory pada kondisi data <i>sparsity</i> saja, sedangkan pada penelitian ini menggunakan kombinasi metode CBF dan CF berbasis model MF dengan algoritma yang berbeda yang hasilnya nanti juga untuk mereduksi error prediksi dari metode CF berbasis memory pada peneliti sebelumnya tidak

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				<p>BPMF dengan nilai RMSE lebih kecil 0,8905.</p> <ul style="list-style-type: none"> Evaluasi nilai MAE metode FPMF unggul dari metode lain yaitu 0,7318 (MI-100K), 0,6238 (FilmTrust), dan 0,7835 (CiaoDVD). Pada MI-Latest – Small memiliki nilai MAE sebesar 0,6960, nilai ini lebih besar dibanding metode BPMF dengan nilai MAE lebih kecil 0,6821. Metode yang diusulkan peneliti dapat mengatasi permasalahan data sparsity pada sistem rekomendasi. 		hanya pada kondisi data <i>sparsity</i> tetapi juga pada kondisi non <i>sparsity</i> , pengujiannya tidak hanya menggunakan RMSE tetapi juga MAPE. Selain itu penelitian kami tidak hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi item yang dihasilkan jika hanya menggunakan metode CBF
7	A Hybrid Approach using Collaborative	Geetha et al (2018)	Melengkapi kekurangan dari metode CF dan	Hasil evaluasi menggunakan 2 data <i>user</i> beserta <i>ratingnya</i>	<ul style="list-style-type: none"> Metode CBF yang <i>dicluster</i> bukan berdasarkan konten film 	Peneliti menggunakan algoritma yang berbeda pada metode HF kombinasi dari metode CBF dan

Tabel 2.1. Matriks literatur *review* dan posisi penelitian

Implementasi Metode Hybrid Filtering pada Sistem Rekomendasi untuk Meningkatkan Kualitas Rekomendasi (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	filtering and Content based Filtering for Recommender System		CBF sehingga dapat mereduksi error prediksi dari sistem rekomendasi.	dari dataset mendapatkan <i>feedback</i> positif dari pengguna.	<p>tetapi dari <i>rating</i> film, jika film belum terdapat <i>rating</i> pada film, data tidak bisa <i>dicluster</i>.</p> <ul style="list-style-type: none"> Model berbasis memory yang digunakan pada metode CF harus melakukan training ulang jika terdapat data <i>rating</i> film baru dalam database Tujuan penelitian tidak sinkron dengan pembuktian yang dilakukan, tujuannya mereduksi error prediksi namun peneliti hanya hanya memberikan contoh implementasi metode HF saja, sehingga belum menjawab tujuan dari penelitian ini. 	CF berbasis memory untuk mereduksi prediksi pada sistem rekomendasi pada kondisi data non <i>sparsity</i> saja, sedangkan pada penelitian ini menggunakan kombinasi metode CBF dan CF berbasis model MF pastinya dengan algoritma yang berbeda yang hasilnya nanti juga untuk mereduksi error prediksi dari metode CF berbasis memory pada peneliti ini tidak hanya pada kondisi data non <i>sparsity</i> , pengujiannya menggunakan RMSE dan MAPE yang tidak dilakukan pada penelitian ini. Selain itu penelitian kami tidak hanya untuk mereduksi error prediksi tetapi juga untuk memperbaiki rekomendasi item yang dihasilkan jika hanya menggunakan metode CBF.

Berdasarkan *matrix literature review* yang sudah dipaparkan pada Tabel 2.1, dapat disimpulkan bahwa penelitian ini fokus untuk meningkatkan kualitas sistem rekomendasi menggunakan kombinasi metode CBF dan CF berbasis model MF menjadi metode HF, kualitas yang ditingkatkan adalah hasil rekomendasi *item* yang *personalized* dengan konten yang terarah untuk memperbaiki kelemahan rekomendasi *item* yang di hasilkan jika hanya menggunakan metode CBF, dan mampu mereduksi error prediksi pada metode CF berbasis memory (Wijaya & Alfian, 2018; Prasetyo et al. 2019; Rahmawati, Nurjanah, and Rismala 2018; Geetha et al. 2018).

Perbedaan penelitian ini dengan penelitian yang telah *direview*. Pertama, memanfaatkan kombinasi metadata konten *item* untuk menghasilkan rekomendasi yang diambil dari dataset Harper & Konstan (2015). Kedua, peneliti Geetha et al (2018) menggunakan algoritma K-Means *Clustering* pada metode CBF, dan algoritma WAOD pada metode CF. Sedangkan pada penelitian ini menggunakan algoritma Count Vectorizer, Cosine Similarity, dan Distance pada metode CBF, lalu pada metode CF menggunakan algoritma SVD. Perbedaan ketiga, kebanyakan peneliti sebelumnya hanya berkontribusi untuk mereduksi error prediksi metode tradisional CF pada salah satu kondisi dan tidak pada hasil rekomendasi *item* yang dihasilkan, sedangkan pada penelitian kami juga untuk mereduksi error prediksi pada metode tradisional CF tetapi pada 2 kondisi sekaligus yaitu data *sparsity* dan *non sparsity*, kontribusi lain dari penelitian untuk menghasilkan rekomendasi *item* yang *personalized* dan terarah kontennya dalam rangka untuk memperbaiki hasil rekomendasi *item* jika hanya menggunakan metode CBF saja .

2.3. Landasan Teori

2.3.1 Sistem Rekomendasi

Sistem rekomendasi adalah sebuah sistem pendukung yang digunakan pada suatu *platform* seperti *website marketplace*, *e-commerce*, dan streaming film untuk menyaring atau *memfilter* banyaknya informasi menjadi informasi yang lebih spesifik sesuai dengan kebutuhan dan keinginan penggunanya. Sistem rekomendasi memiliki kemampuan untuk memprediksi item yang mungkin disukai oleh suatu pengguna berdasarkan profil pengguna tersebut dan juga preferensi dari pengguna lain yang mirip dengan pengguna tersebut.

Sistem rekomendasi bisa bermanfaat untuk pihak pemberi layanan karena bisa meningkatkan penjualan atau pendapatan dengan bantuan sistem yang mampu memberikan rekomendasi *item* yang mungkin disukai oleh pengguna dengan cepat, menarik lebih banyak pengguna untuk mengunjungi *website/aplikasi*, sedangkan untuk pihak pengguna bisa lebih mudah dan cepat untuk menemukan *item* yang disukai sehingga tidak memakan banyak waktu, membantu pengguna dalam menangani masalah kelebihan informasi dengan memberikan rekomendasi konten *item* yang sudah dipersonalisasi. Kebutuhan untuk menggunakan teknik rekomendasi yang efisien dan akurat dalam sistem dapat memberikan rekomendasi *item* yang relevan dan dapat diandalkan bagi pengguna (Isinkaye, Folajimi, and Ojokoh 2015).

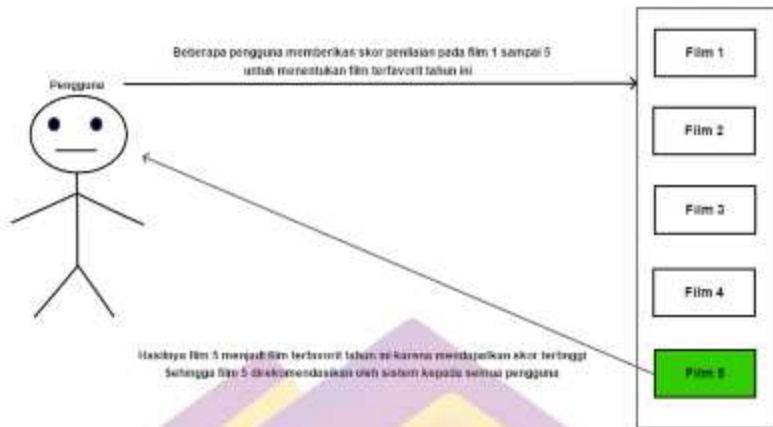
2.3.2 Pendekatan Sistem Rekomendasi

Sistem rekomendasi memiliki beberapa metode atau pendekatan yang bisa digunakan, berikut beberapa pendekatan yang bisa digunakan salah satunya atau juga bisa digabungkan beberapa metodenya menjadi satu:

2.3.2.1 Demographic Filtering

Demographic Filtering (DF) adalah salah satu metode sistem rekomendasi yang memanfaatkan informasi umum dari beberapa profil *user* mengenai suatu item yang paling sering dilihat oleh pengguna. Metode ini dapat digunakan untuk mengidentifikasi minat pengguna yang termasuk dalam komunitas tertentu. Oleh karena itu, dalam metode ini diperlukan beberapa informasi dari semua pengguna untuk mengkategorikannya ke dalam sebuah kelompok. Kemudian, jika beberapa pengguna dalam kelompok tertentu menyukai suatu *item*, mungkin pengguna lain dari kelompok ini cenderung melakukan hal yang sama (Seyednezhad et al. 2018).

Contoh rekomendasi *item* yang dihasilkan menggunakan metode ini seperti *Top 10* rekomendasi film terbaik, *Top 10* produk paling laku, *Top 10* produk yang disukai, dll. Dasar rekomendasi *item* berdasarkan data umum dari aktivitas semua pengguna yang telah digabung menjadi menjadi satu kesatuan untuk menghasilkan sebuah rekomendasi *item*. Ilustrasi dari metode DF dapat dilihat pada Gambar 2.1.



Gambar 2.1. Ilustrasi Metode DF

2.3.2.2 Collaborative Filtering

Collaborative filtering adalah suatu konsep dimana opini dari pengguna lain yang ada digunakan untuk memprediksi *item* yang mungkin disukai/diminati oleh seorang pengguna. Kualitas rekomendasi yang diberikan dengan menggunakan metode ini sangat bergantung dari opini pengguna lain (*neighbor*) terhadap suatu *item*. Belakangan diketahui bahwa melakukan reduksi *neighbor* (yaitu dengan memotong *neighbor* sehingga hanya beberapa pengguna yang memiliki kesamaan / *similarity* tertinggi saja yang akan digunakan dalam perhitungan) mampu meningkatkan kualitas rekomendasi yang diberikan.

CF memberikan rekomendasi berdasarkan kumpulan dari pendapat, minat dan ketertarikan beberapa *user* lain yang biasanya diberikan dalam bentuk *rating* yang diberikan *user* terhadap suatu *item* (Wijaya & Alfian, 2018). Untuk

memperoleh data *rating* dari *user* yang digunakan dalam sistem rekomendasi, dibedakan menjadi dua cara yaitu:

- A. Secara Eksplisit, yaitu proses pengumpulan data dimana *user* memberikan data secara sadar/ sengaja.
- B. Secara Implisit, yaitu proses pengumpulan data dimana *user* tidak menyadari bahwa ia telah memberikan masukan terhadap sistem.

Metode *item-based CF* memberikan rekomendasi berdasarkan kemiripan antar *item* (Wijaya & Alfian, 2018). Metode ini merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu *item* dengan *item* yang pernah dirating *user* lain. *Item* yang telah di *rating* oleh *user* akan menjadi patokan untuk mencari sejumlah *item* lainnya yang berkorelasi dengan *item* yang telah dirating *user*. Kunci dibalik metode ini adalah *user* akan cenderung menyukai *item* yang sejenis atau mempunyai korelasi dengan *item* yang telah disukainya. Teknik dalam metode CF dibagi menjadi 2 (Isinkaye, Folajimi, and Ojokoh 2015) yaitu:

A. Teknik berbasis memory

Teknik berbasis memory ini merupakan teknik lama yang masih banyak digunakan pada sistem rekomendasi yang menggunakan metode CF karena kemampuannya dalam menghasilkan rekomendasi *item* yang *personalized*. CF berbasis memory ini juga terbagi lagi menjadi 2 teknik yaitu berbasis pengguna dan berbasis *item*.

Teknik berbasis pengguna menghitung kemiripan *rating item* yang telah diberikan antar pengguna kemudian menghitung prediksi *rating item* pada *item*

yang belum pernah dilihat oleh pengguna satu berdasarkan bantuan preferensi *rating* dari pengguna lain yang memiliki nilai kemiripan *rating* yang tinggi dengan pengguna satu, sehingga pengguna satu bisa mendapatkan rekomendasi *item* baru yang *personalized*.

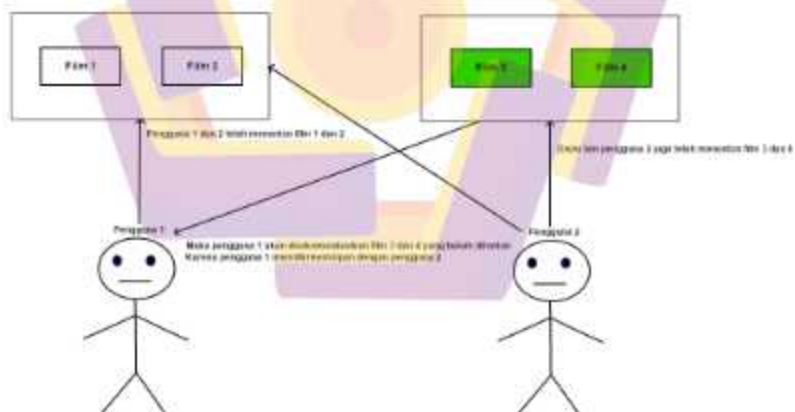
Teknik berbasis *item* menghitung kesamaan *rating* antar *item* dan bukan kesamaan *rating* antar pengguna, nilai kesamaan *item* ini diambil dari *rating* yang telah diberikan pengguna pada *item* tersebut, hal ini untuk menentukan seberapa mirip *item* yang telah dilihat dengan beberapa *item* lain, kemudian memilih *k* *item* yang paling mirip dengan *item* yang telah dilihat oleh pengguna. Setelah itu menghitung nilai prediksi *rating item* pada *item* yang memiliki nilai kemiripan yang tinggi dengan *item* yang telah dilihat oleh pengguna sebelumnya, sehingga pengguna tersebut bisa mendapatkan rekomendasi *item* baru yang *personalized* berdasarkan bantuan preferensi *rating* dari pengguna lain. Beberapa jenis perhitungan nilai kemiripan digunakan untuk menghitung kesamaan antara *item* / pengguna. Dua perhitungan nilai kesamaan yang paling populer adalah berbasis korelasi dan berbasis kosinus.

B. Teknik berbasis model

Teknik ini menggunakan *rating item* sebelumnya untuk mempelajari model dalam meningkatkan kinerja metode CF. Proses pembuatan model dapat dilakukan dengan menggunakan *Machine Learning* atau teknik *Data Mining*. Teknik ini dapat dengan cepat merekomendasikan *item* karena menggunakan model yang telah dihitung atau *training* sebelumnya dan mampu menghasilkan rekomendasi *item* yang *personalized* dengan komputasi yang rendah. Teknik ini menggunakan salah

satu metode Matrix Factorization dalam *Machine Learning* Dimension Reduction seperti Singular Value Decomposition (SVD), Matrix Completion Technique, Latent Semantic method, dan Regression and Clustering.

Teknik berbasis model dilakukan dengan menganalisis matriks *item pengguna* untuk mengidentifikasi hubungan antar *item*, hubungan ini untuk membandingkan daftar rekomendasi N teratas. Teknik berbasis model menyelesaikan masalah data *sparsity* yang terkait dengan sistem rekomendasi. Penggunaan teknik *Machine Learning* juga telah mengubah cara kerjanya rekomendasi item kepada pengguna. Oleh karena itu, sangat penting untuk memeriksa metode *Machine Learning* lain yang cocok digunakan dalam sistem rekomendasi berbasis model. Ilustrasi metode CF dapat dilihat pada Gambar 2.2.



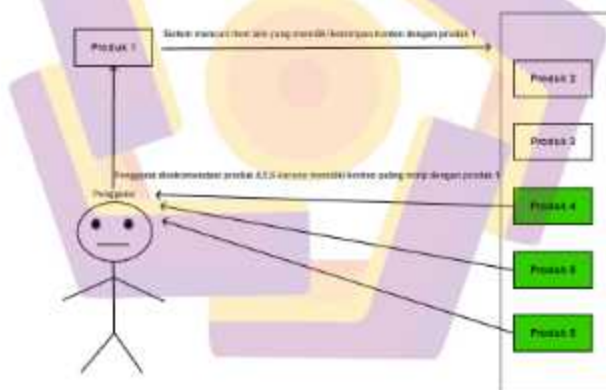
Gambar 2.2. Ilustrasi Metode CF

2.3.2.3 Content Based Filtering

Sistem rekomendasi berbasis konten (*Content-based Recommendation System*) menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut

atau karakteristik) sebuah *item* sebagai basis dalam pemberian rekomendasi. Sebagai contoh, sebuah film mempunyai attribut seperti *genres*, *author*, tahun rilis, dan lain-lain, atau sebuah file dokumen memiliki konten berupa informasi yang ada di dalamnya (Wijaya & Alfian, 2018).

Metode CBF biasa digunakan untuk merekomendasikan berita, artikel, produk, maupun situs web. Metode ini akan mengekstrak informasi yang terdapat pada *item* kemudian membandingkannya dengan informasi *item* yang pernah dilihat atau disukai oleh *user*. Algoritma yang digunakan seperti TF-IDF, Bayesian Classifiers, Cluster Analysis, Decision Trees dan Artificial Neural Networks (Wijaya & Alfian, 2018). Ilustrasi dari metode CBF dapat dilihat pada Gambar 2.3.

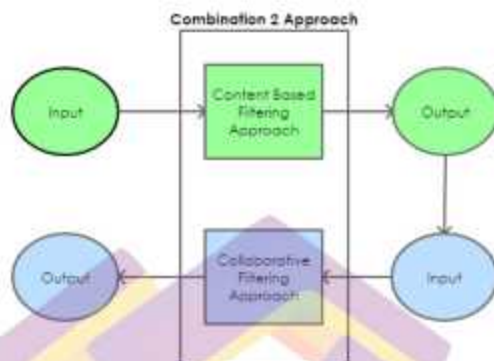


Gambar 2.3. Ilustrasi Metode CBF

2.3.2.4 Hybrid Filtering

Secara umum metode HF merupakan kombinasi dari dua atau lebih metode pada sistem rekomendasi untuk mengatasi kelemahan pada masing-masing metode. Contohnya seperti metode CBF yang hasil dari proses trainingnya akan menjadi

input untuk proses *training* kedua pada metode lain seperti CF, DF, KB, dll (Rahmawati et al., 2018). Ilustrasi dari metode HF dapat dilihat pada Gambar 2.4.



Gambar 2.4. Ilustrasi Metode HF

2.3.3 Algoritma Sistem Rekomendasi

Beberapa pendekatan pada Sistem Rekomendasi mempunyai algoritma yang bisa digunakan untuk proses *training* datanya, contohnya pada metode CBF sebelum masuk proses *training* data dilakukan *Text Feature Extraction* menggunakan algoritma *Count Vectorizer*, lalu untuk proses *training* dengan *Cosine Similarity*, dan *Cosine Distance*. Lalu pada metode CF proses *training* menggunakan salah satu algoritma *Matrix Factorization* yaitu *Singular Value Decomposition*, berikut definisi dan rumus dari masing-masing algoritma:

2.3.3.1 Count Vectorizer

Count Vectorizer merupakan salah satu algoritma untuk ekstraksi atau transformasi data diskret (*categorical*) kedalam bentuk *binary* sehingga dari data *text* menjadi data *numeric* agar dapat diolah pada menggunakan metode ini.

Beberapa algoritma tidak dapat langsung menggunakan variabel diskret sebagai masukannya pada proses komputasi, sehingga diperlukan proses ekstraksi agar data bisa diolah (Prayoginingsih & Kusumawardani, 2018). Contoh proses ekstraksi dapat dilihat pada Tabel 2.2 sampai Tabel 2.4.

Tabel 2.2. Sampel Data Sebelum Ekstraksi

1	Merah	Kuning	Hijau	Merah	Biru
2	Hijau	Biru	Biru	Merah	Biru

Tabel 2.3. Sampel Kata Kunci

Merah	Kuning	Hijau	Biru
-------	--------	-------	------

Tabel 2.4. Sampel Data Sesudah Ekstraksi

Encoding 1	2	1	1	1
Encoding 2	1	0	1	3

2.3.3.2 Cosine Similarity

Algoritma Cosine-based Similarity digunakan untuk menemukan kemiripan pada dua *item* yang dianggap sebagai 2 vektor. Kesamaan antara 2 *item* ini diukur dengan menghitung kosinus dari sudut antara 2 vektor *item*. *Item* dibandingkan misalnya u dan v , dianggap sebagai sebuah vektor baris dengan anggotanya adalah *rating* yang diberikan pada kedua *item* tersebut. Dua vektor dikatakan sama jika membentuk sudut atau nilai kosinusnya 1. Dengan kata lain dua *item* dikatakan mirip jika nilai kosinus dari perhitungan mendekati angka 1 (Wijaya & Alfian, 2018). Berikut ini adalah rumus dari Cosine Similarity (1):

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Dimana variabel A_i dan B_i merupakan komponen dari vektor A dan B yang merepresentasikan kesamaan antar *item*.

2.3.3.3 Cosine Distance

Cosine Distance digunakan untuk menemukan kemiripan *item* berdasarkan jarak terdekat, *input* algoritma ini hanya khusus untuk nilai positif. Jika *menyinputkan* atau terdapat nilai negatif dalam perhitungan maka jarak cosinus tidak akan dihitung. Cosine Distance bisa digunakan dalam sistem rekomendasi untuk merekomendasikan *item* kepada pengguna yang mungkin disukai berdasarkan kemiripan konten dari hasil perhitungan jarak terdekat (Guo et al., 2020). Berikut ini merupakan rumus dari Cosine Distance (2):

$$distance - D_c(A,B) = 1 - S_c(A, B) \quad (2)$$

Dimana $D_c(A,B)$ merupakan kemiripan *item* berdasarkan jarak nilai dari vektor *item* A dan B , lalu $S_c(A,B)$ merupakan hasil perhitungan nilai similarity dari vektor *item* A dan B .

2.3.3.4 Matrix Factorization

Matrix Factorization (MF) merupakan salah satu teknik yang populer untuk memprediksi *rating*. Pada dasarnya, dalam MF diberikan sebuah matriks berisi *rating* (contoh: antara nilai 1 dan 5) dari *user* untuk *item* r_{ui} $m \times n$ (m untuk *user* dan n untuk *item*) diuraikan menjadi matriks *two lower dimensional*, sehingga prediksi *rating* dari semua entitas yang tidak dikenal pada matriks original r_{ui} dapat dihitung.

Dalam MF, *user* dan *item* dimodelkan dengan vektor dari *abstract factor* yang dipelajari oleh *mining rating* yang tersedia. Karena alasan itu algoritma MF juga dapat disebut dengan faktor model. Dalam faktor model, sangat mungkin untuk membangun kesamaan dari *user* dan *item* karena mereka berbagi representasi yang

umum. Persamaan dan hubungan antara *user* dan *item* dimanfaatkan untuk memprediksi *rating* yang hilang dan menghasilkan rekomendasi (Krishnamurthy et al., 2017).

Algoritma yang akan digunakan untuk memfaktorkan suatu matriks pada penelitian ini adalah Singular Value Decomposition (SVD), berikut ini merupakan rumus dari SVD (3):

$$R = M(\Sigma)U^T \quad (3)$$

Dimana variabel R merupakan matrix yang merepresentasikan *rating user* terhadap setiap *item*, variabel M merupakan eigenvector dari matrix RR^T , variabel U merupakan eigenvector dari matrix R^TR , dan (Σ) merupakan asosiasi eigenvalues menjadi matriks diagonal.

2.3.4 Algoritma Evaluasi Error Prediksi

Pada Supervised Learning teknik Regresi dalam kelompok Machine Learning bisa dilakukan proses evaluasi performanya dengan mengukur berapa besar tingkat error prediksi yang dihasilkan ketika melakukan proses Regresi. Terdapat beberapa algoritma yang bisa digunakan untuk mengevaluasi error prediksi yaitu RMSE dan MAPE, berikut definisi dan rumus dari masing-masing algoritma;

2.3.4.1 Root Mean Squared Error (RMSE)

RMSE merupakan salah satu metode yang digunakan untuk mengevaluasi performa hasil prediksi dari suatu metode atau algoritma. Metode ini diterapkan dengan melakukan penjumlahan nilai *rating actual* dikurangi dengan hasil nilai prediksi *rating* lalu dipangkatkan, setelah itu dibagi dengan jumlah data, lalu

diakarkan. RMSE merupakan cara untuk mengevaluasi model dengan mengukur tingkat nilai error prediksi yang dihasilkan oleh model (Hu et al., 2019). Berikut ini merupakan implementasi dari rumus RMSE (4).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (A_i - P_i)^2}{n}} \quad (4)$$

Dimana variabel A_i merupakan nilai *rating item* sebenarnya, variabel P_i merupakan nilai prediksi *rating item*, dan variabel n merupakan jumlah data *rating* sebenarnya.

2.3.4.2 Mean Absolute Percentage Error (MAPE)

MAPE merupakan salah satu metode yang digunakan untuk mengevaluasi performa error prediksi dari hasil prediksi nilai *continue* atau regresi. Metode ini diterapkan dengan menjumlahkan nilai *actual* dikurangi dengan nilai hasil prediksi lalu dibagi dengan nilai *actual* pada keseluruhan data. Kemudian, merata-rata kesalahan persentase *absolute* tersebut setelah itu di kali 100 untuk mendapatkan presentase error prediksinya. MAPE merupakan pengukuran kesalahan prediksi yang menghitung ukuran presentase error prediksi antara data *actual* dengan data prediksi (Al Dhaheri, Woon, and Aung 2017). Berikut ini merupakan implementasi dari rumus MAPE (5).

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{x_i} \right| \quad (5)$$

Dimana variabel x_i merupakan nilai *actual rating item*, variabel \hat{x}_i merupakan nilai prediksi *rating item*, dan variabel n merupakan jumlah keseluruhan data *rating item*.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah Eksperimen, karena melakukan investigasi hubungan sebab akibat jika menggunakan metode CF berbasis memory pada sistem rekomendasi, lalu melakukan pengujian error prediksi yang dihasilkan dari metode HF pada penelitian ini dan membandingkannya dengan CF berbasis memory menggunakan referensi teori atau metode pengujian dari peneliti lain untuk menguatkan hasil pengujian yang didapatkan.

Sifat penelitian ini adalah Kausal, karena *output* atau hasil dari penelitian ini adalah mendapatkan informasi dan pengetahuan yang berhubungan dengan hubungan sebab akibat dari penggunaan metode HF dan CF berbasis memory.

Pendekatan penelitian ini adalah Kuantitatif, karena membangun / memperbaiki performa error prediksi pada metode CF berbasis memory dari penelitian sebelumnya supaya menjadi lebih baik. Jenis data yang digunakan adalah kuantitatif berupa angka *numeric* untuk mengukur tingkat error prediksi dari hasil pengujian dan perbandingan kedua metode untuk mendapatkan kesimpulan performa metode yang terbaik.

3.2. Metode Pengumpulan Data

Pengumpulan data untuk kebutuhan penelitian ini dengan melakukan pencarian dataset *open source* yang cocok dan sesuai untuk kebutuhan penelitian ini pada *website* Kaggle secara *online*, setelah dilakukan pencarian didapatkan

dataset bersumber dari Harper & Konstan (2015) dalam ekstensi atau format csv. Dataset ini dinamakan dengan MovieLens yang berisi kumpulan film internasional yang memiliki atribut atau variabel yang dibutuhkan untuk penelitian ini yang meliputi metadata, konten *item*, dan *rating item* pengguna. Dataset ini bisa didapatkan secara gratis oleh semua orang karena bersifat *open source* dan bisa *download* melalui *website* Kaggle atau langsung dari sumber pembuatnya yaitu GroupLens.

3.3. Metode Analisis Data

Setelah proses pengumpulan data selesai dilakukan, selanjutnya akan dilakukan proses analisis data. Analisa data pada penelitian ini dilakukan dengan menganalisis dataset yang didapatkan dari sumber Harper and Konstan (2015) untuk memastikan bahwa dataset yang didapat sesuai dengan kebutuhan untuk proses penelitian ini. Sistem analisa data disini dengan melakukan *Data and Text Preprocessing*. Langkah pertama dataset akan melalui proses *Data Selection* yaitu memilih beberapa variabel atau atribut pada dataset yang sesuai dengan kebutuhan penelitian. Daftar variabel dan jumlah data masing-masing variabel yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1. Sampel Data Sebelum Ekstraksi

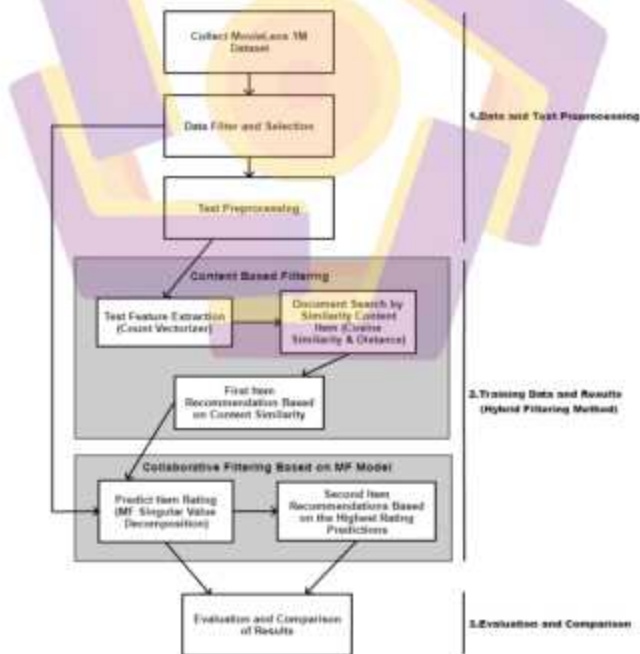
No	Nama Atribut	Jumlah Data
1.	Metadata (<i>Genres, Cast, Keywords, Director</i>)	3706
2.	<i>Movie Title</i>	3706
3.	<i>User ID</i>	6040
4.	<i>Rating Movie</i>	1.000.209

Setelah melalui proses *Data Selection*, selanjutnya pada atribut *User ID* akan *filter* untuk mendapatkan data *rating* dari *user* dengan jumlah yang seimbang. Langkah selanjutnya *Text Preprocessing*, langkah ini dilakukan

menggunakan atribut metadata tujuannya agar isi data metadata yang mengandung kata-kata bahasa inggris dan dianggap tidak informatif seperti kata "and", "the", "him" akan dihapus dari isi metadata sehingga atribut metadata dapat merepresentasikan informasi konten *item* yang informatif. Setelah melalui *Data and Text Preprocessing*, atribut metadata akan digunakan untuk proses *feature text extraction* atau *encoding* untuk mengkonversi data *text* menjadi data *numeric* atau angka sehingga nantinya bisa dijadikan *input* untuk proses *training*.

3.4. Alur Penelitian

Tahapan keseluruhan implementasi metode HF dalam penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Implementasi Metode Hybrid Filtering

Tahap Pertama pada Gambar 3.1 adalah *Data and Text Preprocessing*, penelitian ini menggunakan Dataset MovieLens yang diambil dari sumber Harper & Konstan (2015) yang memiliki atribut *Genres*, *Cast*, *Keywords*, *Director*, *Overview*, *Movie Title*, *User ID*, dan *Rating Movie*. Dari keseluruhan atribut dilakukan proses *Data Selection* yaitu memilih beberapa atribut yang sesuai dengan kebutuhan penelitian, hasil pemilihan atribut dan jumlah data masing-masing atribut dapat dilihat pada Tabel 3.1. Setelah melalui proses data *selection*. Selanjutnya data akan *difilter* pada atribut *User ID* untuk proses evaluasi kondisi pertama data *non sparsity* yaitu memilih pengguna yang telah merating banyak *item* sekitar 150 – 200 *rating* dan kondisi evaluasi kedua data *sparsity* yaitu memilih pengguna yang memiliki sedikit *rating item* sekitar 25 – 50 *rating*, langkah ini perlu dilakukan supaya pada saat proses evaluasi dan perbandingan metode nanti data pengguna yang digunakan memiliki jumlah *rating* yang seimbang.

Setelah melalui proses *filter* didapatkan 518 pengguna dengan total 89290 *rating* untuk kondisi pertama data *non sparsity*, lalu untuk kondisi kedua data *sparsity* didapatkan 1377 pengguna dengan total 50106 *rating*. Langkah selanjutnya adalah *Text Preprocessing*, langkah ini dilakukan agar isi data pada atribut metadata yang mengandung kata-kata bahasa inggris dan dianggap tidak informatif seperti kata “*and*”, “*the*”, “*him*” akan dihapus dari isi metadata sehingga atribut metadata dapat merepresentasikan informasi konten *item* yang informatif.

Tahap Kedua pada Gambar 3.1 adalah *Training Data and Results*, sebelum melakukan proses *training* data dilakukan proses *Feature Text Extraction* atau Ekstraksi terlebih dahulu menggunakan algoritma Count Vectorizer untuk

mengkonversi data *text* menjadi data *numeric* atau angka agar nantinya data pada atribut metadata dapat *diinput* dan diolah atau *ditraining* menggunakan metode CBF, contoh proses dan hasil konversi dapat dilihat pada Tabel 2.2 sampai Tabel 2.4.

Setelah proses *Feature Text Extraction* selesai, selanjutnya akan dilakukan proses *training* data pertama menggunakan metode CBF yang memanfaatkan atribut dari metadata yang berisi kombinasi atribut *Genres*, *Cast*, *Keywords*, dan *Director* film yang sudah dikonversi datanya, lalu dilakukan proses pencarian *item* yang memiliki konten paling mirip berdasarkan *item* yang sudah dilihat oleh pengguna menggunakan algoritma Cosine Similarity (1) dan Cosine Distance (2). Hasil dari proses training metode CBF adalah daftar rekomendasi *item* berdasarkan konten yang paling mirip dengan *item* yang sudah pernah dilihat oleh pengguna sebelumnya.

Setelah proses *training* pertama selesai, lalu dilakukan proses *training* kedua menggunakan metode CF berbasis model MF yang memanfaatkan atribut *User ID*, dan *Rating Movie* dari hasil rekomendasi *item* pertama berdasarkan kemiripan konten, dari rekomendasi tersebut data *ratingnya* akan dijadikan *input* untuk dilakukan proses regresi / prediksi *rating item* menggunakan algoritma SVD (3) sehingga didapatkan hasil rekomendasi *item* kedua berdasarkan kemiripan konten yang sudah *personalized* dan sudah terarah kontennya.

Tahap Ketiga pada Gambar 3.1 *Evaluation and Comparison*, setelah proses *training* data selesai, selanjutnya pada bagian metode Hybrid CF akan dievaluasi

untuk mengukur berapa besar error prediksi yang dihasilkan metode HF untuk menghasilkan rekomendasi *item* kepada pengguna.

Tahap ini dibagi menjadi 2 yaitu evaluasi error prediksi metode Hybrid CF dan proses implementasi metode HF untuk melihat perbedaan rekomendasi *item personalized* dan yang tidak *personalized*. Tahap pertama akan dilakukan evaluasi error prediksi menggunakan algoritma RMSE (4) dan MAPE (5), tahap ini dilakukan hanya pada bagian metode CF berbasis Model MF, karena itu dari proses *Data Filter and Selection* datanya langsung di *training* pada metode CF tanpa melalui proses *training* metode CBF terlebih dahulu. Proses evaluasi kondisi pertama data *non sparsity* menggunakan data *rating* yang sudah difilter sebanyak 89290 lalu dibagi menjadi 80361 data *training* dan 8929 data *testing*, sedangkan untuk evaluasi kondisi kedua data *sparsity* menggunakan data *rating* sebanyak 50106 yang dibagi menjadi 42948 data *training* dan 7158 data *testing*.

Evaluasi kondisi pertama akan dilakukan sebanyak 10 kali iterasi, sedangkan kondisi kedua 7 kali iterasi. Penentuan banyaknya jumlah iterasi ini berdasarkan pertimbangan dari beberapa pernyataan para peneliti sebelumnya yang berdiskusi melalui *website* researchgate.net, para peneliti banyak merekomendasikan 5 sampai 10 iterasi karena jika semakin sedikit jumlah iterasi yang dilakukan akan menyebabkan hasil error prediksinya tinggi karena model kekurangan data *training* untuk proses belajar dan terlalu banyak data *testing* untuk divalidasi.

Penentuan jumlah data *training* harus lebih banyak dibandingkan dengan data *testing* ini bisa diibaratkan sama seperti seorang mahasiswa yang akan

menempuh ujian semester. Jika mahasiswa ingin mendapatkan nilai yang maksimal otomatis harus mempelajari banyak materi sebelumnya supaya bisa menjawab semua soal ujian dengan baik dan benar sehingga bisa mendapatkan nilai yang optimal. Sebaliknya jika mahasiswa tersebut hanya mempelajari sedikit materi sebelumnya otomatis ketika ujian berlangsung mahasiswa tersebut akan kesulitan untuk bisa menjawab semua soal ujian dengan benar sehingga mendapatkan nilai yang tidak maksimal. Dari contoh tersebut bisa diambil kesimpulan bahwa pada proses evaluasi jumlah data *training* harus lebih banyak dibandingkan dengan data *testing* yang akan diuji sehingga setiap iterasinya model bisa menghasilkan performa yang maksimal yaitu menurunkan bias terhadap estimasi error prediksi.

Berdasarkan dataset proses evaluasi kondisi pertama data *non sparsity* dilakukan dengan jumlah data *rating* 89290 yang pengujiannya bisa dilakukan maksimal 10 iterasi, sedangkan pada kondisi kedua data *sparsity* dengan jumlah data *rating* 50106 bisa dilakukan maksimal 7 iterasi. Untuk lebih menguatkan lagi terdapat peneliti (Hu et al 2019; Feng et al 2020; Li et al 2019) yang juga menggunakan jumlah iterasi dari 5 sampai 10 lebih untuk memaksimalkan performa model ketika proses pengujian dan evaluasi error prediksi. Hasil evaluasi kedua kondisi akan ditampilkan nilai RMSE dan MAPE, setelah itu juga ditampilkan rata-rata nilai RMSE dan MAPE dari evaluasi kedua kondisi. Hasil evaluasi 2 kondisi dari metode Hybrid CF akan dibandingkan error prediksinya dengan metode CF berbasis memory yang menggunakan algoritma umum Cosine Similarity dan Weighted Average of Deviation dengan dataset yang sama.

Setelah evaluasi error prediksi selesai dilakukan, selanjutnya akan dilakukan implementasi metode HF dari awal sampai akhir yang prosesnya lengkap dari *Collect MovieLens 1M Dataset* sampai *Evaluation and Comparison of Results*. Tahap ini dilakukan untuk mengetahui perbedaan hasil rekomendasi *item personalized* yang sudah terarah kontennya dari metode HF dan yang belum *personalized* dari metode CBF, sehingga nantinya akan didapatkan pengetahuan dan gambaran mengenai perbedaan jenis rekomendasi *item personalized* dan tidak *personalized* yang dihasilkan dari kedua metode tersebut.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Data and Text Preprocessing

4.1.1 Collect MovieLens 1M Dataset

Langkah pertama pada tahap *Data and Text Preprocessing* adalah mengumpulkan atau menyiapkan *dataset* MovieLens yang didapatkan dari sumber (Harper & Konstan, 2015) yang memiliki atribut *Genres*, *Cast*, *Keywords*, *Director*, *Overview*, *Movie Title*, *User ID*, dan *Rating Movie*. Dataset ini berekstensi csv dan untuk memudahkan dalam membaca dan menganalisis datanya maka dilakukan proses *load dataset* dan import beberapa *library* yang dibutuhkan menggunakan bahasa pemrograman python. *Source code* untuk menampilkan sampel *dataset* yang sudah *diload* bisa dilihat pada Gambar 4.1 dan 4.2.

```
In [10]: import numpy as np
import pandas as pd
import math
import os
import re
from surprise import Reader, Dataset, SVD, SVDpp
from surprise.knn_selection import knn_svd, knn_als, knn_als_data, GridSearchCV
from surprise.knn_selection import knn
from surprise.knn_selection import knn_nearest_neighbors
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
```

Load Dataset

```
In [11]: df = pd.read_csv('dataset/movie_titles_1m.csv')
df_2df = pd.read_csv('dataset/content_1m_multiple.csv')
df
```

```
Out[11]:
```

movie_id	movie	rating
0	The Five-Year Engagement (2015)	5
1	Apes and the Gold Rush (1980)	5
2	My Fair Lady (1964)	5
3	Chinatown (1974)	4
4	Apocalypse Now (1979)	4
...
1000000	Witness at Everest (1980)	1
1000001	Crash Course, The (1960)	3
1000002	Welcome to the Database (1993)	2
1000003	Boomer's Choice (1982)	4
1000004	E.T. the Extra-Terrestrial (1982)	4

1000000 rows x 3 columns

Gambar 4.1. Source Code Sampel Dataset MovieLens 1

```
In [10]: import numpy as np
import pandas as pd
import math
import re
import os
from surprise import Reader, Dataset, SVD, SVDpp
from surprise.model_selection import cross_validation, GridSearchCV
from surprise.model_selection import kfold
from sklearn.metrics.pairwise import cosine_distances, cosine_similarity
from sklearn.feature_extraction.text import CountVecorizer
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
```

Load Dataset

```
In [11]: df = pd.read_csv("dataset/movie_lens/movie_ratings.csv")
df_inf = pd.read_csv("dataset/content_to_multiple.csv")
df_inf
```

```
Out[11]:
```

	title	genres	cast	keywords	director	metadata
0	The Shory	animation, comedy, family	bob_osteen, bob_osteen	patrick_hy_bry	john_wood	animation, comedy, family, bob_osteen, bob_osteen
1	Jarant	adventure, fantasy, family	robin_williams, jessica_hahn, bryan_lynch	looney_tunes, disney, cartoon, rabbit, in, children's, book	jean_cochran	adventure, fantasy, family, robin_williams, jessica_hahn
2	Cherise, Old Man	comedy, drama	willie_nelson, ted_danson, jim_geraghty	looney_tunes, disney, cartoon, rabbit, in, children's, book	harold_blaust	animation, comedy, family, willie_nelson, ted_danson, jim_geraghty
3	Waiting to Exhale	comedy, drama, romance	angela_bassett, angela_bassett, laurie_rivero	black, in, movie, interracial, relationship, single	randy_chaker	comedy, drama, romance, angela_bassett, angela_bassett, laurie_rivero
4	Factor of the Five Part 2	comedy	mark_martin, mark_martin, mark_martin, john	comic, movie, crime, detective	stuart_ayke	comedy, drama, thriller, crime, action, mark_martin, john
...
42272	Expatriate	action, drama	john_wood, john_wood, jim_cummings	hall	bob_ross	action, drama, john_wood, john_wood, jim_cummings
42273	Suburb	drama, family	john_wood, john_wood, john_wood	hall	george_yule	drama, family, john_wood, john_wood, john_wood
42274	Century of Soling	drama	john_wood, john_wood, john_wood	hall	robert_gordon	drama, john_wood, john_wood, john_wood
42275	Factor of the Five Part 1	comedy	mark_martin, mark_martin, mark_martin, john	comic, movie, crime, detective	stuart_ayke	comedy, drama, thriller, crime, action, mark_martin, john
42276	Quarantine	hall	john_wood, john_wood, john_wood, john_wood	hall	john_wood	action, thriller, john_wood, john_wood, john_wood, john_wood
42277	Factor of the Five Part 3	comedy	mark_martin, mark_martin, mark_martin, john	comic, movie, crime, detective	stuart_ayke	comedy, drama, thriller, crime, action, mark_martin, john

Gambar 4.2. Source Code Sampel Dataset MovieLens 2

4.1.2 Data Selection

Setelah dataset berhasil *diload* langkah selanjutnya adalah *Data Selection* yaitu memilih beberapa atribut yang sesuai dengan kebutuhan penelitian, lalu hasilnya disimpan dalam *Data Frame* baru untuk menampung atribut yang sudah melalui proses *Data Selection*. Hasil pemilihan atribut dan jumlah data masing-masing atribut dapat dilihat pada Gambar 4.3.

```

In [3]: movie = {}
        userID = []
        rating = []
        metadata = []
        for i in range(len(df_cf.movie)):
            regex = re.sub("[^a-zA-Z]", "", df_cf.movie.to_numpy()[i])
            if not df_cf[df_cf.title == regex.rstrip()].metadata.empty:
                userID.append(df_cf.userID.to_numpy()[i])
                movie.append(df_cf[df_cf.title == regex.rstrip()].title.to_numpy()[0])
                rating.append(df_cf.rating.to_numpy()[i])
                metadata.append(df_cf[df_cf.title == regex.rstrip()].metadata.to_numpy()[0])
df_new = ["userID":userID,"movie":movie,"rating":rating,"metadata":metadata]
df_new = pd.DataFrame(df_new)
df_new.to_csv("1:\Kampus\in\delejar_python\id_cf\dataweb\hybrid_filtering.csv", index = False, header=True)
df_new

```

```

Out[3]:

```

	userID	movie	rating	metadata
0	1	One Flew Over the Cuckoo's Nest	5	drama_joc_1stebellon_buiss_felcher_danny_dier
1	1	James and the Giant Peach	3	adventure_animation_family_miran_margalita_jo
2	1	My Fair Lady	3	drama_filmis_music_audrey_hepburn_rol_harrison
3	1	Don Brackovich	4	drama_suka_indarto_ahart_bray_aron_actor
4	1	Sanjaya	5	action_olahraga_drama_startan_hesker_pakr_ha
...				
719002	5040	Prison	3	drama_war_action_film_burkhan_dharma_green_ol
719001	5040	Insidious at Berkary	1	comedy_olahraga_incorrectly_olahraga_olahraga_cah
719004	5040	Welcome to the Dharma	5	comedy_drama_health_substance_wedding_olahraga
719005	5040	Beauty Choice	4	drama_romance_musik_film_olahraga_suka_pakr_h
719006	5040	8 1/2 The Extra Terrestrial	4	science_fiction_adventure_filmis_filmis_olahraga

715037 rows x 4 columns

Gambar 4.3. Source Code Data Selection

4.1.3 Data Filter

Selanjutnya dilakukan *filter* data pada atribut User ID untuk kebutuhan evaluasi kondisi pertama data *non sparsity* dengan memilih pengguna yang telah *merating* sebanyak 150 – 200 *item*, dan kebutuhan evaluasi kondisi kedua data *sparsity* dengan *memfilter* pengguna yang telah *merating* 25-50 *item*. Langkah ini perlu dilakukan supaya pada saat proses evaluasi dan perbandingan error prediksi kedua metode nanti, jumlah data *rating* penggunanya seimbang.

Hasil proses *filter* didapatkan 518 pengguna untuk kondisi pertama dan 1377 pengguna untuk kondisi kedua, setelah itu membuat *Data Frame* baru untuk menampung seluruh *rating* film dari 518 pengguna pada kondisi pertama didapatkan total 89290 *rating*, dan *Data Frame* untuk kondisi kedua dari 1377 pengguna didapatkan total 50106 *rating* film. *Data Frame* ini digunakan untuk

proses evaluasi dan perbandingan error prediksi yang dihasilkan dari metode HF dengan CF berbasis memory. *Source code* proses filter dan pembuatan data *frame* baru untuk proses evaluasi dan perbandingan error prediksi kedua kondisi bisa dilihat pada Gambar 4.4 dan Gambar 4.5.

Proses filter user yang memiliki jumlah rating sekitar 150-200 dan pembuatan data frame baru

```
In [1]: user_id = []
movie = []
rating = []
for i in range(len(filter_userid)):
    if len(df[df.userid == filter_userid[i],rating] >= 150 and len(df[df.userid == filter_userid[i],rating] <= 200):
        for a in range(len(df[df.userid == filter_userid[i]])):
            user_id.append(df[df.userid == filter_userid[i],movie][a][4])
            movie.append(df[df.userid == filter_userid[i],rating][a][1])
            rating.append(df[df.userid == filter_userid[i],rating][a][2])
            new_df = pd.DataFrame({'userid':user_id, 'movie':movie, 'rating':rating})
            new_df
```

```
Out[1]:
```

userid	movie	rating	
0	The Princess Diaries (2001)	4	
0	State and Mainways (1990)	4	
2	Delicatessen (1991)	4	
3	Quentin (1999)	3	
6	Harold Ramo Films (1964)	2	
60260	020	Madison, The (1961)	3
60261	020	Swing Parade Ryo (1960)	3
60262	020	Diana, The (1961)	3
60263	020	B.T. the Duke Terrestrial (1962)	3
60264	020	Chercher David, A (1959)	3

Out[1]: rows = 2 columns

Gambar 4.4. *Source Code Filter* Atribut *User ID* Kondisi Pertama

Proses filter user yang memiliki jumlah rating sekitar 25-50 dan pembuatan data frame baru

```
In [2]: user_id = []
movie = []
rating = []
for i in range(len(filter_userid)):
    if len(df[df.userid == filter_userid[i],rating] >= 25 and len(df[df.userid == filter_userid[i],rating] <= 50):
        for a in range(len(df[df.userid == filter_userid[i]])):
            user_id.append(df[df.userid == filter_userid[i],movie][a][4])
            movie.append(df[df.userid == filter_userid[i],rating][a][1])
            rating.append(df[df.userid == filter_userid[i],rating][a][2])
            new_df = pd.DataFrame({'userid':user_id, 'movie':movie, 'rating':rating})
            new_df
```

```
Out[2]:
```

userid	movie	rating	
0	7	Madam Impresario (1936)	4
1	7	Superior (1930)	4
2	7	Delicatessen (1930)	4
3	7	Delicat (1930)	3
4	7	Killer (1930)	3
...
60191	020	Paper (1936)	4
60192	020	Our First Episode - The Program Director (1936)	2
60193	020	Rever (1936)	4
60194	020	Orphan in the Park (1932)	3
60195	020	Black Legion, The (1935)	3

Out[2]: rows = 5 columns

Gambar 4.5. *Source Code Filter* Atribut *User ID* Kondisi Kedua

4.2. Training Data and Results

4.2.1 Text Feature Extraction

Setelah *filter text* atribut metadata dilakukan selanjutnya sebelum dilakukan *training data* yang pertama akan dilakukan proses *Feature Text Extraction* menggunakan metode Count Vectorizer, langkah ini dilakukan untuk mengkonversi data *text* yang ada pada atribut metadata menjadi data *numeric* / angka supaya dapat menjadi *input* untuk proses *training* pertama menggunakan metode CBF. *Source code* untuk melakukan *Text Feature Extraction* bisa dilihat pada Gambar 4.7.

Text Feature Extraction

```
In [4]: df_movie_metadata = pd.read_csv("dataset/hybrid_filtering_metadata.csv")
bank = bow.fit_transform(df_movie_metadata.metadata)
```

Gambar 4.7. *Source Code Feature Text Extraction*

4.2.2 Document Search by Similarity Content Item & First Item Recommendation Based on Similarity Content

Setelah data *text* selesai di ekstraksi langkah selanjutnya dilakukan proses *Document Search by Similar Content Item*, langkah ini untuk mencari *item* yang memiliki konten paling mirip berdasarkan *item* yang sudah dilihat oleh pengguna menggunakan algoritma Cosine Similarity dan Cosine Distance yang rumusnya bisa dilihat pada (1) dan (2). Langkah ini dilakukan untuk memberikan contoh rekomendasi *item* yang belum *personalized*, untuk implementasinya menggunakan data User ID 1 dan 6 yang telah menonton film dengan judul *My Fair Lady*, setelah

itu hasil dari proses *training* pertama menggunakan metode CBF ini adalah *Top 30* rekomendasi *item* yang belum *personalized* berdasarkan konten yang paling mirip dengan *item* yang sudah pernah dilihat oleh User ID 1. *Source code* dan hasil *output* dari rekomendasi *itemnya* bisa dilihat pada Gambar 4.8.

```

Top 30 Recommend movie by similar content

24 [12]: movie_by_content = df_movie_metadata[df_movie_metadata.movie_id == '1412'].metadata.to_numpy()[0]
code = bow.transform([movie_by_content])
code.toarray()
(dfdf = df_movie_metadata[dfdf.movie_id == code])
movie_id = dfdf.columns[0]
rer_rank

# for i in range(len(df_movie_metadata)):

94[12]: array([[1412, 1628, 1638, 2122, 1675, 1677, 1688, 1689, 2276, 1567, 1130,
1625, 1584, 1713, 1498, 4384, 518, 1363, 532, 2452, 1601, 2394,
1465, 2838, 2338, 1542, 134, 1889, 1367, 1837], dtype=int64)]

24 [114]: df_movie_metadata.ix[rer_rank, movie_id]

34[114]: 1403    Oliver!
1404    Death
1405    Carmen
2122    Mike Figg Music
1428    Tomo
1477    Light of Day
1480    King King
1481    Grand 2
2276    Beauty
1567    Colette (Le)
1584    Holiday Inn
1622    Vertigo
1638    Musical
1713    Daddy Long Legs
1675    Lady Mary
1677    Charlotte's Web
1688    Mr. Miller's Organ
1689    King of Kings
2276    Peter
1465    Cheaper
1498    Sun Sets
2338    Doublet: A Double - Teil
1542    Desert Bloom
1465    Time Masters
2114    Honey
2334    Honey
1489    Tander Harlek
1689    Homes and the Holy Wallroad
1501    Lady Tea
2527    Saffron
name: movie, dtype: object

```

Gambar 4.8. *Source Code Document Search by Similar Content Item*

4.2.3 Predict Item Rating & Second Item Recommendation Based on Highest Rating Predictions

Setelah proses *training* pertama selesai, selanjutnya data rekomendasi *item* dari hasil *training* pertama akan dijadikan *input* untuk langkah kedua yaitu *Predict Item Rating* menggunakan metode CF berbasis model MF yang memanfaatkan data *rating* dari User ID 1 Dan 6. Algoritma yang digunakan adalah SVD yang rumus

perhitungannya bisa dilihat pada (3). Hasil dari proses *training* kedua ini adalah *Top 10* rekomendasi *item* yang sudah *personalized* diurutkan berdasarkan nilai prediksi *rating* tertinggi ke terendah yang ditujukan untuk User ID 1 dan 6 setelah melihat film dengan judul *My Fair Lady*. Proses *training* kedua metode ini dilakukan untuk mengetahui hasil rekomendasi *item* dari metode CBF dan HF. Source code proses *training* kedua bisa dilihat pada Gambar 4.9, lalu hasil rekomendasi *item* yang *personalized* untuk kedua pengguna dengan User ID 1 dan 6 bisa dilihat pada Gambar 4.10.

Second Training Process using CF based on the MF Model (SVD)

```
In [1]: df_movies = df_movies[['movie_id', 'movie_title']]
data = Dataset.load_from_df(df_movies, Reader())
evaluator = MeanSquaredErrorWrapper()
model_svd = SVD()
model_svd.fit(data)

Out[1]: Completed recommendation algorithm matrix factorization SVD at 04:34:27.9404
```

Gambar 4.9. Source Code Second Training Process using CF based on the MF Model (SVD)

Top 10 Movie Recommendations Based on the Highest Rating

```
In [224]: df_movies[['movie_id', 'movie_title', 'rating']] = df_movies[['movie_id', 'movie_title', 'rating']]
score = []
for i in range(len(df_movies_metadata.loc[rec_id].movie_to_recommend)):
    score.append((model_svd.predict(df_movies_metadata.loc[rec_id].movie_to_recommend[i]),
    model_svd.predict(df_movies_metadata.loc[rec_id].movie_to_recommend[i]),
    score))
score.sort(key = lambda x: -x[1])

Out[224]: [('Oliver!', 4.4979234440188),
('Carmen', 4.405301474464204),
('Milk and Honey', 4.31883708989884),
('The Godfather', 4.31786248438919),
('Mr. Holland's Opus', 4.06624881011708),
('Bandits', 4.06423327806004),
('Charlie's Web', 3.95831191648194),
('Napoli', 3.901358743222097),
('Detroit', 3.94354025675474),
('Baby Long Legs', 3.888884175484902)]

In [225]: df_movies[['movie_id', 'movie_title', 'rating']] = df_movies[['movie_id', 'movie_title', 'rating']]
score2 = []
for i in range(len(df_movies_metadata.loc[rec_id].movie_to_recommend)):
    score2.append((model_svd.predict(df_movies_metadata.loc[rec_id].movie_to_recommend[i]),
    model_svd.predict(df_movies_metadata.loc[rec_id].movie_to_recommend[i]),
    score2))
score2.sort(key = lambda x: -x[1])
score2[1:10]

Out[225]: [('Milk and Honey', 4.31883708989884),
('Mr. Holland's Opus', 4.06624881011708),
('The Godfather', 4.31786248438919),
('Carmen', 4.405301474464204),
('Laramie', 4.248524587416145),
('The Godfather', 4.31786248438919),
('Bandits', 4.06423327806004),
('Swing Kids', 3.88829820272482),
('Newline', 3.80881577948041),
('Baby Long Legs', 3.888884175484902),
('The Godfather', 4.31786248438919)]
```

Gambar 4.10. *Source Code Top 10 Personalized Item Recommendations*

4.3. Evaluation and Comparison of Results

4.3.1 Pra Evaluasi Error Prediksi

Sebelum melakukan Evaluasi dan Perbandingan Error Prediksi metode akan dilakukan proses *training* data ulang tetapi hanya pada bagian metode CF berbasis model MF untuk mengevaluasi error prediksi yang dihasilkan pada kondisi data non *sparsity* dan data *sparsity* yang prosesnya bisa dilihat pada Gambar 3.1 dari *Data Filter and Selection* langsung menuju ke *Predict Item Rating*, proses *training* ulang hanya hanya dilakukan pada metode CF berbasis model MF karena yang terdapat proses regresi atau prediksi *rating* hanya pada metode ini sedangkan pada metode CBF digunakan hanya untuk membantu menghasilkan rekomendasi *item* yang terarah kontennya dan tidak berpengaruh terhadap error prediksi saat dikombinasikan dengan CF.

Proses *training* ulang metode CF berbasis model MF menggunakan fitur *GridSearchCV* dan *Kfold* yang disediakan dari *library* *Surprise* versi 1.1.0 untuk membantu melakukan *preprocessing*, *data splitting*, *training*, dan evaluasi metode menjadi 1 paket. Lalu untuk proses *training* metode CF berbasis memory tidak menggunakan bantuan *library* tetapi hanya berdasarkan algoritma yang kami buat sesuai dengan referensi dari peneliti sebelumnya. *Source code* untuk proses evaluasi kondisi pertama dari kedua metode dapat dilihat pada Gambar 4.11 sampai Gambar 4.14, sedangkan untuk proses evaluasi kondisi kedua bisa dilihat pada Gambar 4.15 sampai Gambar 4.18.

Proses training dan evaluasi metode Hybrid Filtering Algoritma SVD menggunakan RMSE 10 Iterasi

```

In [4]: reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(new_df[['userid', 'movie', 'rating']], reader)
param_grid = {'n_epochs': [1, 10], 'lr_all': [0.002, 0.001],
              'reg_all': [0.4, 0.4]}
gs = GridSearchCV(SVD, param_grid, measures=['rmse'], cv=10)
gs.fit(data)

```

Gambar 4.11. Source Code Proses Evaluasi Kondisi Pertama Metode HF dengan RMSE

Proses Training dan Evaluasi Metode CF Berbasis Memory menggunakan RMSE 10 Iterasi

```

In [ ]: def standardize(row):
    row_user = (row - min_user()) / (max_user() - min_user())
    return row_user

all_movies = new_df
all_movies = all_movies.pivot_table(index='movie', columns='movie', values='rating')
all_movies = all_movies.apply(standardize)
all_movies.fillna(0, inplace=True)
item_similarity = cosine_similarity(all_movies.T)
item_similarity_df = pd.DataFrame(item_similarity, index=all_movies.columns, columns=all_movies.columns)
mean_rating_movies = new_df
mean_rating_movies = mean_rating_movies.pivot_table(index='movie', columns='movie', values='rating')
mean_rating_movies.fillna(0, inplace=True)
rating_all_user = new_df
rating_all_user = rating_all_user.pivot_table(index='movie', columns='movie', values='rating')
rating_all_user.fillna(0, inplace=True)
predict_uuid = 0
sum_similarities = 0
WGD = 0
RMSE_uuid = 0
fold_RMSE = []
mean_fold_RMSE = 0
tampung_RMSE = []
mean_rating_file = 0
actual = []
predicted = []
for i in range(10): #Membuat looping 10 evaluasi
    for k in range(10): #Membuat looping 10 kali
        #Membuat looping untuk user
        user = new_df[new_df.userid == (i+1)*1000].movie.to_numpy()[k]
        #Membuat looping rating user ke
        predict_uuid = (rating_all_user[new_df.userid == (i+1)*1000].movie.to_numpy()[k]) * item_similarity[new_df.userid == (i+1)*1000].movie.to_numpy()[k] + mean_rating_movies[new_df.userid == (i+1)*1000].movie.to_numpy()[k]
        WGD_uuid = new_df[new_df.userid == (i+1)*1000].rating.to_numpy()[k]
        actual.append(new_df[new_df.userid == (i+1)*1000].rating.to_numpy()[k])
        predicted.append(predict_uuid)
        sum_similarities = 0
        WGD = 0
        mean_rating_file = 0
    #Membuat looping untuk movie
    fold_RMSE.append(math.sqrt(tampung_RMSE))
    mean_fold_RMSE += (math.sqrt(tampung_RMSE))
    tampung_RMSE = 0
    RMSE_uuid = 0
dummy_data = pd.DataFrame(dummy_data)
fold_RMSE

```

Gambar 4.12. Source Code Proses Evaluasi Kondisi Pertama Metode CF Berbasis Memory dengan RMSE

Proses training metode HF dengan model SVD

```
In [77]: reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(rms_df[['userId', 'movie', 'rating']], reader)
rate = []
# define a cross-validation iterator
kf = kfold(n, splits=10)
algo = SVD()
for (trainset, testset) in kf.split(data):
    # train and test algorithm
    algo.fit(trainset)
    predictions = algo.test(testset)
    rate.append(algo.test(testset))
```

Proses Evaluasi metode HF model SVD menggunakan MAPE 10 Iterasi

```
In [ ]: iterasi = []
score = 0
Mean_score = 0
for i in range(10):
    for a in range(len(MAPE1[1])):
        score = abs((MAPE1[1][a][2] - MAPE1[1][a][1])/MAPE1[1][a][2])
    iterasi.append((score/10)*100)
Mean_score = (score/10)*100
score = 0
iterasi
```

Gambar 4.13. *Source Code* Proses Evaluasi Kondisi Pertama Metode HF dengan MAPE

Proses Training dan Evaluasi Metode CF Berbasis Memory menggunakan MAPE 10 Iterasi

```
In [ ]: def standarize(user):
    rms_user = rms[rms.userId==user].as_matrix()
    return rms_user
all_movies = rms_df
all_movies = all_movies.pivot_table(index='movie', columns='movie', values='rating')
all_movies = all_movies.apply(standardize)
all_movies.fillna(0, inplace=True)
item_similarity = cosine_similarity(all_movies.T)
item_similarity_df = pd.DataFrame(item_similarity, index=all_movies.columns, columns=all_movies.columns)
Mean_Rating_Movies = rms_df
Mean_Rating_Movies = Mean_Rating_Movies.pivot_table(index='movie', columns='movie', values='rating')
Mean_Rating_Movies.fillna(0, inplace=True)
rating_all_user = rms_df
rating_all_user = rating_all_user.pivot_table(index='movie', columns='movie', values='rating')
rating_all_user.fillna(0, inplace=True)
predict_u000 = 0
sum_similarity = 0
u000 = 0
rms_u000 = 0
fold_MAPE = []
Mean_fold_MAPE = 0
Mean_Rating_File = 0
aktual = []
prediksi = []
for i in range(10): #untuk looping 10 kali
    for a in range(len(rms_df.loc[rms_df['movieId']==i].movie)): #untuk looping nilai user per fold
        Mean_Rating_File = Mean_Rating_Movies[rms_df.loc[rms_df['movieId']==i].movie.to_numpy()[a]].mean()
        for b in range(len(rating_all_user[rms_df.loc[rms_df['movieId']==i].movieId.to_numpy()[a]])): #untuk looping rating user ke
            predict_u000 = (rating_all_user[rms_df.loc[rms_df['movieId']==i].movieId.to_numpy()[a]][b] * Mean_Rating_Movies[rating_a
            sum_similarity = item_similarity_df[rms_df.loc[rms_df['movieId']==i].movieId.to_numpy()[a]][b] * rating_all_user_index[b])
        u000 = Mean_Rating_File + (predict_u000 * sum_similarity)
        aktual.append(rms_df.loc[rms_df['movieId']==i].rating.to_numpy()[a])
        prediksi.append(Mean_Rating_File + (predict_u000 * sum_similarity))
    Mean_u000 = abs((rms_df.loc[rms_df['movieId']==i].rating.to_numpy()[a] - u000)/rms_df.loc[rms_df['movieId']==i].rating.to_numpy
    sum_similarity = 0
    u000 = 0
    Mean_Rating_File = 0
    fold_MAPE.append((MAPE_u000/100)*100)
    Mean_fold_MAPE += ((MAPE_u000/100)*100)
    MAPE_u000 = 0
dummy_data = ('movie', aktual, 'prediksi', prediksi)
dummy_data = pd.DataFrame(dummy_data)
fold_MAPE
```

Gambar 4.14. *Source Code* Proses Evaluasi Kondisi Pertama Metode CF Berbasis Memory dengan MAPE

Proses training dan evaluasi metode Hybrid Filtering Algoritma SVD menggunakan RMSE 7 Iterasi

```

In [13]: reader = Reader(rating_scale=[1, 5])
data = Dataset.load_from_df([userId, 'movie', 'rating'], reader)
param_grid = {'n_epochs': [1, 10], 'lr_all': [0.001, 0.005],
              'reg_all': [0.4, 0.6]}
gs = GridSearchCV(SVD, param_grid, measures=['rmse'], cv=7)
gs.fit(data)

```

Gambar 4.15. *Source Code* Proses Evaluasi Kondisi Kedua Metode HF dengan RMSE

Proses Evaluasi Metode CF Berbasis Memory menggunakan RMSE 7 iterasi

```

20 [ ] def standardize(rms): #
    rms_rms = (rms - rms.mean())/(rms.std())
    return rms_rms
all_movies = rms_df
all_movies = all_movies.pivot(index='movie', columns='movie', values='rating')
all_movies = all_movies.apply(lambda row: row, axis=1)
all_movies.fillna(0.0, inplace=True)
item_similarity = cosine_similarity(all_movies.T)
user_similarity_df = pd.DataFrame(item_similarity, index=all_movies.columns, columns=all_movies.columns)
user_rating_movies = rms_df
user_rating_movies.fillna(0, inplace=True)
rating_all_user = rms_df
rating_all_user = rating_all_user.pivot(index='movie', columns='movie', values='rating')
rating_all_user.fillna(0, inplace=True)
predict_u000 = 0
u000 = 0
RMSE_u000 = 0
for i in range(1, 7):
    user_similarity = 0
    u000 = 0
    RMSE_u000 = 0
    temping_u000 = 0
    user_rating_file = 0
    actual = []
    predict = []
    for i in range(1, jumlah_bayang_7_movies):
        for k in range(1, jumlah_bayang_7_movies):
            # Untuk looping movie dan user
            user_rating_file = user_rating_movies[rms_df.loc[('7150'+i+1)*7150:movie_to_numpy()[k]]].values
            for j in range(1, jumlah_bayang_7_movies):
                rating_all_user = rating_all_user[rms_df.loc[('7150'+i+1)*7150:movie_to_numpy()[k]]]
                user_id_to_numpy()[k] = j
                user_similarity = cosine_similarity(rms_df.loc[('7150'+i+1)*7150:movie_to_numpy()[k]]).values
                u000 = user_rating_file[predict_u000, user_similarity]
            RMSE_u000 = rms_df.loc[('7150'+i+1)*7150:rating_to_numpy()[k]] - u000
            actual.append(rms_df.loc[('7150'+i+1)*7150:rating_to_numpy()[k]])
            predict.append(user_rating_file[predict_u000, user_similarity])
            predict_u000 = 0
            user_similarity = 0
            u000 = 0
            user_rating_file = 0
    temping_u000 = RMSE_u000/7150
    fold_RMSE.append(math.sqrt(temping_u000))
    user_rating_file = 0
    temping_u000 = 0
    RMSE_u000 = 0
dummy_data = ['actual', 'predict', 'predict']
dummy_data = pd.DataFrame(dummy_data)
fold_RMSE

```

Gambar 4.16. *Source Code* Proses Evaluasi Kondisi Kedua Metode CF Berbasis Memory dengan RMSE

Proses training metode HF dengan model SVD

```
In [7]: reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df([user_id, 'movie', 'rating'], reader)
MAPC = 1
# Define a cross-validation iterahar
kf = kFold(n_splits=3)
kfold = 500()
for (trainset, testset) in kf.split(data):
    # train and test algorithm
    algo.fit(trainset)
    predictScore = algo.test(testset)
    MAPC.append(algo.test(testset))
```

Proses Evaluasi metode HF model SVD menggunakan MAPE 7 Iterasi

```
In [00]: iterasi = []
score = 0;
Mean_score = 0
for i in range(7):
    for a in range(len(MAPE[1])):
        score = abs((MAPE[1][a][2] - WPT[1][a][2]) / WPT[1][a][2])
    iterasi.append((score/7200)*100)
    Mean_score += (score/7200)*100
score = 0
iterasi
```

Gambar 4.17. Source Code Proses Evaluasi Kondisi Kedua Metode HF dengan MAPE

Proses Evaluasi Metode CF Berbasis Memory menggunakan MAPE 7 Iterasi

```
In [ ]: def trainCFMemory():
    mem_data = mem_data[[(trainset)-mem_data]]
    return mem_data
    all_movies = mem_data
    all_movies = all_movies.pivot_table(index=[user_id], columns=[movie], values=rating)
    all_movies = all_movies.apply(lambda x: x)
    all_movies.fillna(0, inplace=True)
    item_similarity = cosine_similarity(all_movies.T)
    item_similarity_df = pd.DataFrame(item_similarity, index=all_movies.columns, columns=all_movies.columns)
    mem_rating_movies = mem_data
    mem_rating_movies = mem_rating_movies.pivot_table(index=[user_id], columns=[movie], values=rating)
    mem_rating_movies.fillna(0, inplace=True)
    rating_all_user = mem_data
    rating_all_user = rating_all_user.pivot_table(index=[movie], columns=[user_id], values=rating)
    rating_all_user.fillna(0, inplace=True)
    predict_u400 = 0
    mem_similarity = 0
    u400 = 0
    mem_u400 = 0
    fold_MAPE = []
    Mean_fold_MAPE = 0
    mem_rating_file = 0
    actual = []
    predicted = []
    for i in range(7): #untuk looping 7 kali
        for a in range(len(mem_data.loc[177100:(1+177100)])): #untuk looping nilai user per fold
            mem_rating_file += mem_rating_movies[mem_data.loc[177100:(1+177100)].movie.to_numpy()[a]].mean()
            for k in range(len(rating_all_user[mem_data.loc[177100:(1+177100)].user_id.to_numpy()[a]])): #untuk looping rating user ke
                predict_u400 += (rating_all_user[mem_data.loc[177100:(1+177100)].user_id.to_numpy()[a]][k] * mem_rating_movies[rating_all
                    user_similarity=item_similarity_df[mem_data.loc[177100:(1+177100)].movie.to_numpy()[a]][rating_all_user.index[k]])
            actual.append(mem_data.loc[177100:(1+177100)].rating.to_numpy()[a])
            predicted.append(mem_rating_file*(predict_u400/item_similarity))
            MAPC.append(abs((mem_data.loc[177100:(1+177100)].rating.to_numpy()[a] - u400)/mem_data.loc[177100:(1+177100)].rating.to_numpy
                predict_u400 = 0
                mem_similarity = 0
                u400 = 0
                mem_u400 = 0
            mem_rating_file = 0
            fold_MAPE.append((MAPC_u400/(mem_u400))
            Mean_fold_MAPE += (MAPC_u400/(mem_u400))
            MAPC = 0
    dummy_data = {'actual':actual, 'predicted':predicted}
    dummy_data = pd.DataFrame(dummy_data)
    fold_MAPE
```

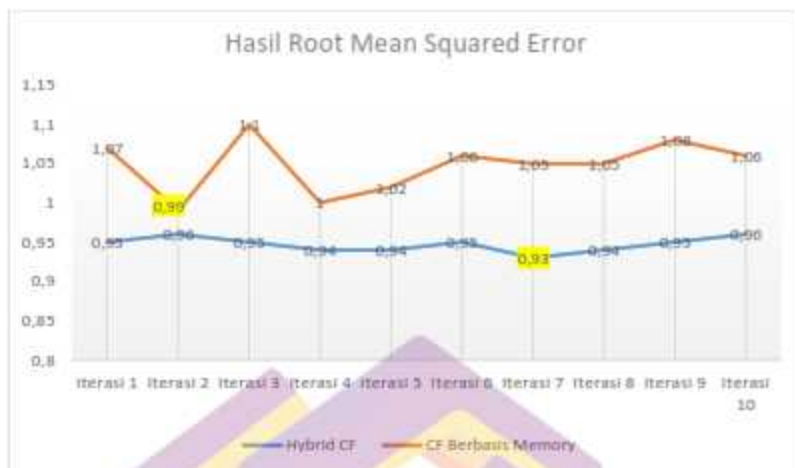
Gambar 4.18. Source Code Proses Evaluasi Kondisi Kedua Metode CF Berbasis Memory dengan MAPE

4.3.2 Evaluasi dan Perbandingan Error Prediksi

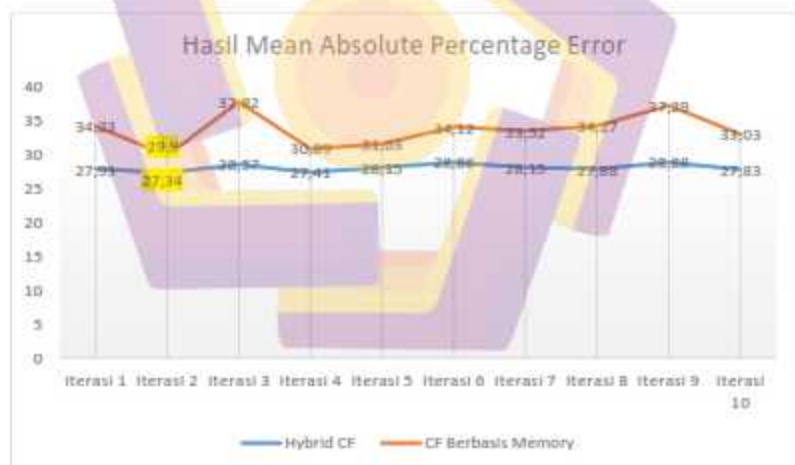
Tahap akhir adalah *Evaluation and Comparison*, pertama akan dilakukan evaluasi error prediksi untuk mengevaluasi performa dan membandingkan error prediksi yang dihasilkan dari metode Hybrid CF dan CF berbasis memory dari penelitian sebelumnya. Metode yang digunakan untuk mengukur error prediksi kedua model adalah RMSE (4) dan MAPE (5), proses evaluasi kondisi pertama dilakukan sebanyak 10 kali iterasi atau perulangan menggunakan 89290 data rating yang dibagi menjadi 80361 data *training* dan 8929 data *testing*, lalu untuk kondisi kedua dilakukan evaluasi sebanyak 7 kali iterasi menggunakan 50106 data rating yang dibagi menjadi 42948 data *training* dan 7158 data *testing*.

Setiap iterasi akan ditampilkan hasil nilai RMSE dan MAPE, setelah itu ditampilkan rata-rata nilai RMSE, MAPE dari seluruh iterasi masing-masing kondisi yang sudah dilakukan untuk memutuskan metode yang terbaik berdasarkan nilai RMSE dan MAPE yang terkecil.

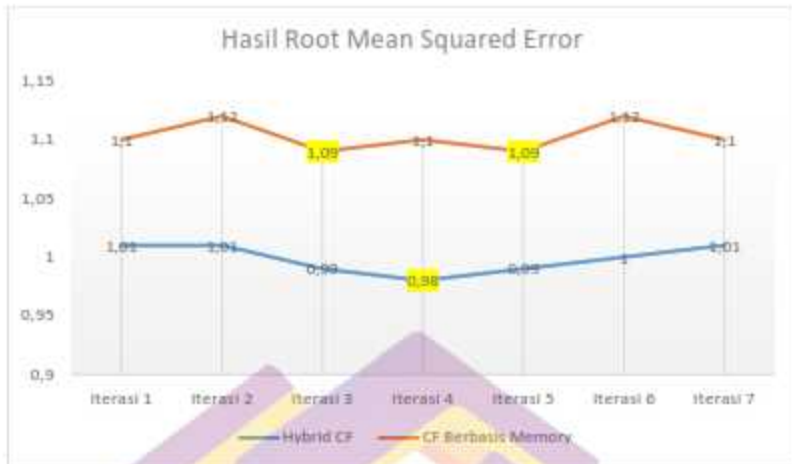
Setelah proses evaluasi kedua metode sebanyak selesai dilakukan, hasil evaluasi dan perbandingan nilai error prediksi kondisi pertama menggunakan metode RMSE, MAPE dapat dilihat pada Gambar 4.19 dan Gambar 4.20. Sedangkan untuk hasil evaluasi kondisi kedua bisa dilihat pada Gambar 4.21 dan Gambar 4.22.



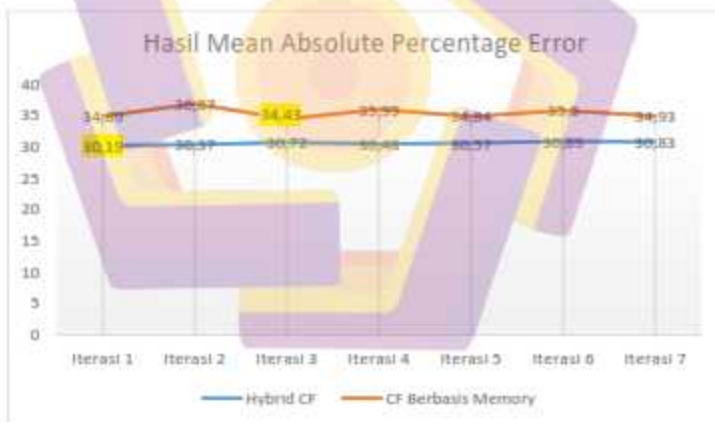
Gambar 4.19. Hasil Evaluasi dan Perbandingan Nilai RMSE Kondisi Pertama



Gambar 4.20. Hasil Evaluasi dan Perbandingan Nilai MAPE Kondisi Pertama



Gambar 4.21. Hasil Evaluasi dan Perbandingan Nilai RMSE Kondisi Kedua



Gambar 4.22. Hasil Evaluasi dan Perbandingan Nilai MAPE Kondisi Kedua

Berdasarkan hasil evaluasi dan perbandingan error prediksi dari kedua metode kondisi pertama data non *sparsity* pada Gambar 4.19 dan Gambar 4.20, dapat dilihat bahwa setelah dilakukan evaluasi sebanyak 10 kali iterasi menggunakan RMSE dan MAPE, metode Hybrid CF mampu menghasilkan error

prediksi lebih kecil pada iterasi ke 7 dengan nilai RMSE 0,93 dan nilai MAPE terkecilnya pada iterasi ke 2 sebesar 27,34%, sedangkan pada metode CF berbasis memory nilai RMSE terkecilnya pada iterasi ke 2 sebesar 0,99 dan nilai MAPE terkecilnya pada iterasi ke 2 sebesar 29,90%. Dari 10 iterasi yang telah dilakukan, metode Hybrid CF menghasilkan rata-rata error prediksi terkecil dengan nilai RMSE 0,95 dan MAPE 28,10%, sedangkan metode CF berbasis memory mendapatkan rata-rata error prediksi lebih besar dengan nilai RMSE 1,05 dan MAPE 33,67%.

Setelah itu hasil evaluasi dan perbandingan error prediksi dari kedua metode untuk kondisi kedua data *sparsity* pada Gambar 4.21 dan Gambar 4.22, dapat dilihat bahwa setelah dilakukan evaluasi sebanyak 7 kali iterasi menggunakan RMSE dan MAPE, metode Hybrid CF mampu menghasilkan error prediksi lebih kecil pada iterasi ke 4 dengan nilai RMSE 0,98 dan iterasi ke 1 dengan nilai MAPE 30,19%, sedangkan pada metode CF berbasis memory nilai RMSE terkecilnya pada iterasi ke 3, 5 sebesar 1,09 dan nilai MAPE terkecilnya pada iterasi ke 3 sebesar 34,43%. Dari 7 iterasi yang telah dilakukan, metode Hybrid CF menghasilkan rata-rata error prediksi terkecil dengan nilai RMSE 0,99 dan MAPE 30,58%, sedangkan metode CF berbasis memory mendapatkan rata-rata error prediksi lebih besar dengan nilai RMSE 1,10 dan MAPE 35,39%.

Detail selisih error prediksi berdasarkan hasil rata-rata nilai RMSE dan MAPE kedua metode pada kondisi pertama data non *sparsity* dapat dilihat pada Tabel 4.1 dan kondisi kedua data *sparsity* pada Tabel 4.2.

Tabel 4.1. Selisih Error Prediksi Nilai RMSE dan MAPE Kondisi Pertama

Metode Evaluasi	Hybrid CF	CF Berbasis Memory	Selisih Error Prediksi
RMSE	0,95	1,05	↓0,10
MAPE	28,10%	33,67%	↓5,57%

Tabel 4.2. Selisih Error Prediksi Nilai RMSE dan MAPE Kondisi Kedua

Metode Evaluasi	Hybrid CF	CF Berbasis Memory	Selisih Error Prediksi
RMSE	0,99	1,10	↓0,11
MAPE	30,58 %	35,39 %	↓4,81%

Berdasarkan hasil selisih error prediksi dari nilai RMSE dan MAPE kedua metode pada Tabel 4.1 dan Tabel 4.2, dapat dilihat bahwa metode Hybrid CF mempunyai performa lebih baik dibandingkan dengan metode tradisional CF berbasis memory pada penelitian sebelumnya karena mampu mereduksi error prediksi lebih kecil dengan penurunan nilai RMSE 0,10, MAPE 5,57% pada kondisi pertama dan penurunan nilai RMSE 0,11, MAPE 4,81% pada kondisi kedua.

4.3.3 Evaluasi dan Perbandingan Hasil Rekomendasi Item

Setelah evaluasi error prediksi dilakukan, selanjutnya pada tahap *Evaluation and Comparison* akan dilakukan evaluasi hasil rekomendasi *item* dari metode HF dan metode CBF. Evaluasi ini dilakukan untuk mengetahui perbedaan hasil rekomendasi *item personalized* yang sudah terarah kontennya dari metode HF dan yang belum *personalized* dari metode CBF.

Pengujian dilakukan menggunakan 2 sampel data pengguna pada atribut User ID dengan ID 1 dan 6 yang telah menonton film dengan judul My Fair Lady dengan isi metadata "drama family music audrey_hepburn rex_harrison stanley_holloway musical transformation flower_girl george_cukor", setelah itu akan diimplementasikan metode CBF dan HF untuk menghasilkan rekomendasi

film kepada kedua pengguna tersebut, hasil rekomendasi CBF bisa dilihat pada Gambar Tabel 4.3 dan rekomendasi HF pada Tabel 4.4.

Tabel 4.3. Hasil Rekomendasi Film Metode CBF

No	Judul Film	Metadata
1	Oliver!	Drama family music ron_moody shani_wallis oliver_reed pickpocket musical victorian_england carol_reed
2	Ducts	comedy drama music gwyneth_paltrow scott_speedman maria_bello musical bruce_paltrow
3	Carmen	music drama julia_migenes ruggero_ramondi faith_esham opera musical francesco_rosi
4	Make Mine Music	animation family music nelson_eddy dinah_shore benny_goodman musical robert_cornack
5	Tommy	drama music oliver_reed ann-margret roger_daltrey christmas_party musical christmas_tree ken_russell
6	Light of Day	music drama michael_j_fox gena_rowlands joan_jett brother_sister_relationship cleveland_musical paul_schrader
7	Swing Kids	music drama robert_sean_leonard christian_bale frank_whaley suicide_world_war_ii musical thomas_carter
8	Grease 2	comedy drama family maxwell_caulfield michelle_pfeiffer lorna_luft musical talent_show motorcycle patricia_birch
9	Bandits	action drama foreign katja_ricmann jasmine_tabatabai nicollette_kreibitz prison musical music katja_von_garnier
10	Coyote Ugly	comedy drama music piper_perabo adam_garcia john_goodman bar musical beautiful_woman david_mcnally
11	Holiday Inn	comedy drama music bing_crosby fred_astaire marjorie_reynolds holiday_new_year's_eve musical mark_sandrich
12	Footloose	drama family music kevin_bacon john_lithgow dianne_wiest dancing_dancer_dance herbert_ross
13	Newsies	drama music family christian_bale bill_pullman ann-margret juvenile_crime child_empowerment brawl kenny_ortega
14	Daddy Long Legs	drama music romance fred_astaire leslie_caron terry_moore paris_college musical jean_negulesco
15	Topsy-Turvy	comedy drama music allan_corduner jim_broadbent timothy_spall prostitute musical biography mike_leigh
16	Charlotte's Web	animation drama family debbie_reynolds henry_gibson danny_bonaduce spider musical talking_animal charles_a_nichols
17	Mr. Holland's Opus	music drama family richard_dreyfuss glenne_headly jay_thomas composer mentor deaf-mute stephen_herek

Tabel 4.3. Hasil Rekomendasi Film Metode CBF (Lanjutan)

No	Judul Film	Metadata
18	King Creole	drama action music elvis_presley carolyn_jones walter_matthau nightclub musical crime_boss michael_curtiz
19	Help!	Comedy music george_harrison john_lennon paul_mccartney musical music cult richard_lester
20	Chectah	drama family keith_coogan lucy_deakins timothy_landfield jeff_blyth
21	Two Bits	drama family jerry_barone al_pacino mary_elizabeth_mastrantonio james_foley
22	Squanto: A Warrior's Tale	drama family adam_beach sheldon_peters_wolfchild irene_bedard_xavier_koller
23	Desert Bloom	drama family annabeth_gish jon_voight jobeth_williams eugene_corr
24	Time Masters	animation drama family jean_valmont michel_elias frédéric_legros rené_laloux
25	Gordy	drama family fantasy ted_manson afemo_omilami country_music mark_lewis
26	Andre	drama family adventure tina_majorino keith_carradine chelsea_field george_t_miller
27	Tender Mercies	drama music romance robert_duvall tess_harper betty_buckley bruce_beresford
28	Thomas and the Magic Railroad	animation drama family alee_baldwin peter_fonda mara_wilson britt_allcroft
29	Melody Time	music family animation roy_rogers dennis_day freddy_martin clyde_geronimi
30	Sarafina!	Drama family foreign leleti_khumalo whoopi_goldberg john_kani darrell_james_roodt

Tabel 4.4. Hasil Rekomendasi Film Metode HF

No	Judul Film	Prediksi Rating Film	User ID
1	Oliver!	4,5	ID - 1
2	Carmen	4,49	
3	Holiday Inn	4,32	
4	Tender Mercies	4,18	
5	Mr. Holland's Opus	4,06	
6	Bandits	4,04	
7	Charlotte's Web	3,97	
8	Help!	3,96	
9	Desert Bloom	3,94	

Tabel 4.4. Hasil Rekomendasi Film Metode HF (Lanjutan)

No	Judul Film	Prediksi Rating Film	User ID
10	Daddy Long Legs	3,89	ID - 6
1	Holiday Inn	4,31	
2	Mr. Holland's Opus	4,29	
3	Charlotte's Web	4,27	
4	Carmen	4,24	
5	Tender Mercies	4,06	
6	Bandits	4,04	
7	Swing Kids	3,99	
8	Newsies	3,96	
9	Daddy Long Legs	3,89	
10	Footloose	3,87	

Berdasarkan hasil implementasi kedua metode bisa dilihat pada Tabel 4.3 merupakan hasil rekomendasi *item* belum *personalized* jika hanya menggunakan metode CBF, artinya User ID 1 dan 6 setelah menonton film My Fair Lady rekomendasi *item* masing-masing pengguna sama tidak ada yang beda, sedangkan pada Tabel 4.4 hasil rekomendasi *item* yang sudah *personalized* dan terarah kontennya jika menggunakan metode HF. Antara User ID 1 dan 6 masing-masing bisa mendapatkan rekomendasi film yang berbeda karena pada kombinasi metode HF ini terdapat 2 metode yang memiliki kelebihan masing-masing yang dijadikan satu, kelebihan metode pertama CBF untuk merekomendasikan *item* berdasarkan kemiripan konten yang terarah, setelah itu metode kedua CF berbasis model MF mampu merekomendasikan *item* secara *personalized*.

Jadi kombinasi metode CBF dan CF menjadi metode HF ini membuat sistem rekomendasi mampu menghasilkan rekomendasi *item* yang sudah *personalized* dan sudah terarah kontennya, sehingga kombinasi metode ini mampu

memperbaiki kelemahan rekomendasi *item* yang dihasilkan jika hanya menggunakan metode CBF yang belum *personalized* atau jika hanya menggunakan metode CF yang sudah *personalized* namun belum terarah kontennya.

Pada metode HF kondisi yang membuat Sistem Rekomendasi gagal memberikan rekomendasi *item* yang *personalized* kepada pengguna adalah ketika melakukan proses *regressi* atau prediksi *rating* film. Pada Tabel 4.4 User ID 1 memiliki *history rating* film sebanyak 41, sedangkan User ID 6 sebanyak 58 *rating*, data *history rating* inilah yang membuat 30 film dari hasil rekomendasi pertama berdasarkan kemiripan konten menjadi *personalized* ketika dilakukan proses *training* dengan metode CF karena data *history rating* dari pengguna juga menjadi *input* untuk proses prediksi *rating* dari film yang belum pernah dilihat oleh pengguna.

Jika User ID 1 dan 6 tidak memiliki tidak memiliki data *rating* sama sekali, maka yang akan terjadi ketika sudah mendapatkan rekomendasi *item* pertama berdasarkan kemiripan konten dari hasil *training* pertama metode CBF, saat masuk pada proses *training* kedua menggunakan metode CF, hasil rekomendasi film yang kedua menjadi tidak *personalized* artinya kedua pengguna hanya mendapatkan rekomendasi film berdasarkan kemiripan konten saja tetapi belum *personalized* karena kedua pengguna tidak memiliki data *history rating* yang bisa diolah pada metode CF untuk menghasilkan rekomendasi *item* yang *personalized*. Untuk mengukur seberapa *personalized* hasil rekomendasi *item* dengan melakukan pengujian error prediksi saat proses *training* khusus metode CF yang hasilnya bisa dilihat pada Gambar 4.19 sampai Gambar 4.22.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil evaluasi dan perbandingan error prediksi yang telah dilakukan pada penelitian ini, dapat disimpulkan bahwa:

1. Pendekatan Hybrid CF mampu mereduksi error prediksi lebih kecil dibandingkan dengan pendekatan tradisional CF berbasis memory, dari hasil evaluasi kondisi pertama dengan data *non sparsity* yang telah dilakukan, metode Hybrid CF menghasilkan rata-rata error prediksi terkecil dengan nilai RMSE 0,95 dan MAPE 28,10%, sedangkan metode CF berbasis memory menghasilkan rata-rata error prediksi lebih besar dengan nilai RMSE 1,05 dan MAPE 33,67%. Lalu dari hasil evaluasi kondisi kedua dengan data *sparsity* yang telah dilakukan, metode Hybrid CF dapat menghasilkan rata-rata error prediksi terkecil dengan nilai RMSE 0,99 dan MAPE 30,58%, sedangkan metode CF berbasis memory menghasilkan rata-rata error prediksi lebih besar dengan nilai RMSE 1,10 dan MAPE 35,39%.
2. Metode Hybrid CF mampu menurunkan error prediksi pada metode CF berbasis memory dari penelitian sebelumnya dengan penurunan nilai RMSE sebesar 0,10, MAPE 5,57% pada kondisi pertama data *non sparsity*, lalu pada kondisi kedua data *sparsity* penurunan nilai RMSE sebesar 0,11 dan MAPE 4,81. Jadi, metode Hybrid CF yang diusulkan dapat memperbaiki metode CF berbasis memory.
3. Setelah itu berdasarkan evaluasi dan perbandingan hasil rekomendasi *item* metode CBF pada Tabel 4.3 dan metode HF pada Tabel 4.4 membuktikan bahwa

metode HF mampu menghasilkan rekomendasi *item personalized* yang sudah terarah kontennya kepada setiap pengguna dan mampu memperbaiki metode CBF yang belum bisa menghasilkan rekomendasi *item yang personalized*.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat saran untuk penelitian selanjutnya dari beberapa temuan baru yang belum diteliti diantaranya :

1. Menggabungkan lagi metode Demographic Filtering (DF) pada penelitian ini, menjadi 3 kombinasi metode, hal ini mampu mengatasi permasalahan *Cold Start Problem* yang artinya jika terdapat pengguna baru yang belum memiliki aktivitas atau *rating item* apapun, sistem tetap dapat merekomendasikan *item* kepada pengguna dengan bantuan metode DF.
2. Mencoba mengimplementasikan metode pada sistem rekomendasi dengan teknik Deep Learning supaya mampu mengatasi lebih baik permasalahan data *sparsity* maupun non *sparsity*, dan menurunkan error prediksi menjadi lebih kecil lagi.
3. Terdapat temuan kelemahan pada penelitian yang telah dilakukan ini, yaitu dalam menghasilkan rekomendasi *item yang personalized* dan terarah kontennya dibutuhkan 2 sumber data yaitu data konten *item* dan *history rating* dari *user*, jika kekurangan atau tidak terdapat salah satu dari 2 sumber data tersebut otomatis hasil rekomendasi *item* tidak sesuai dan tidak optimal. Untuk penelitian selanjutnya yang bisa dikerjakan dalam memperbaiki kelemahan pada penelitian ini adalah mencoba teknik Human In The Loop, teknik ini hanya memanfaatkan 1 sumber data saja yaitu data konten tanpa menggunakan *history rating* pengguna. Implementasi teknik ini dengan menyimpan setiap aktivitas yang

dilakukan pengguna saat melihat atau menonton *item* tertentu, setelah tersimpan akan menjadi data *history* konten yang pernah dilihat oleh tiap *user* dan bisa digunakan untuk menghasilkan rekomendasi *item* yang *personalized* dan terarah kontennya tanpa menggunakan data *history rating* sama sekali. Kelebihan dari teknik ini adalah *user* lama maupun baru tidak perlu repot memberikan *rating* pada setiap *item* untuk menghasilkan rekomendasi yang *personalized*, karena cukup dari riwayat aktivitas *user* ketika melihat atau menonton beberapa konten *item* saja sudah bisa digunakan untuk menghasilkan rekomendasi *item* yang *personalized* dan terarah kontennya untuk mengatasi kelemahan dari metode HF yang harus membutuhkan 2 sumber data.



DAFTAR PUSTAKA

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Al Dhaheri, Khawla, Wei Lee Woon, and Zeyar Aung. 2017. "Wind Speed Forecasting Using Statistical and Machine Learning Methods: A Case Study in the UAE." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10691 LNAI(November 2017): 107–20.
- Feng, Chenjiao, Jiye Liang, Peng Song, and Zhiqiang Wang. 2020. "A Fusion Collaborative Filtering Method for Sparse Data in Recommender Systems." *Information Sciences* 521: 365–79. <https://doi.org/10.1016/j.ins.2020.02.052>.
- Guo, Taolin, Junzhou Luo, Kai Dong, and Ming Yang. 2019. "Locally Differentially Private Item-Based Collaborative Filtering." *Information Sciences* 502: 229–46.
- Harper, F. Maxwell, and Joseph A. Konstan. 2015. "The Movielens Datasets: History and Context." *ACM Transactions on Interactive Intelligent Systems* 5(4): 1–19.
- Hu, Yutian et al. 2019. "Movie Collaborative Filtering with Multiplex Implicit Feedbacks." *Neurocomputing*(xxxx).<https://doi.org/10.1016/j.neucom.2019.03.098>.
- Isinkaye, F. O., Y. O. Folajimi, and B. A. Ojokoh. 2015. "Recommendation Systems: Principles, Methods and Evaluation." *Egyptian Informatics Journal* 16(3): 261–73.
- Jiang, Shan, Shu Cherg Fang, Qi An, and John E. Lavery. 2019. "A Sub-One Quasi-Norm-Based Similarity Measure for Collaborative Filtering in Recommender Systems." *Information Sciences* 487: 142–55.
- Li, Kangkang et al. 2019. "Sparse Online Collaborative Filtering with Dynamic Regularization." *Information Sciences* 505: 535–48.
- Mongia, Aanchal, Neha Jhamb, Emilie Chouzenoux, and Angshul Majumdar. 2020. "Deep Latent Factor Model for Collaborative Filtering." *Signal Processing* 169.
- Nassar, Nour, Assef Jafar, and Yasser Rahhal. 2020. "A Novel Deep Multi-Criteria Collaborative Filtering Model for Recommendation System." *Knowledge-Based Systems* 187: 104811. <https://doi.org/10.1016/j.knosys.2019.06.019>.

- Natarajan, Senthilselvan, Subramaniaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H. Gandomi. 2020. "Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data." *Expert Systems with Applications* 149: 113248. <https://doi.org/10.1016/j.eswa.2020.113248>.
- Prasetyo, Bondan et al. 2019. "Implementasi Metode Item-Based Collaborative Filtering Dalam Pemberian Rekomendasi Calon Pembeli Aksesoris Smartphone." *Eksplora Informatika* 9(1): 17–27.
- Pujahari, Abinash, and Dilip Singh Sisodia. 2020. "Pair-Wise Preference Relation Based Probabilistic Matrix Factorization for Collaborative Filtering in Recommender System." *Knowledge-Based Systems* (xxxx): 105798. <https://doi.org/10.1016/j.knosys.2020.105798>.
- Rahmawati, Sari, Dade Nurjanah, and Rita Rismala. 2018. "Analisis Dan Implementasi Pendekatan Hybrid Untuk Sistem Rekomendasi Pekerjaan Dengan Metode Knowledge Based Dan Collaborative Filtering." *Indonesian Journal on Computing (Indo-JC)* 3(2): 11.
- Seyednezhad, S. M. Mahdi, Kailey Nobuko Cozart, John Anthony Bowllan, and Anthony O. Smith. 2018. "A Review on Recommendation Systems: Context-Aware to Social-Based." *arXiv*.
- Tewari, Anand Shanker. 2020. "Generating Items Recommendations by Fusing Content and User-Item Based Collaborative Filtering." *Procedia Computer Science* 167(2019): 1934–40. <https://doi.org/10.1016/j.procs.2020.03.215>.
- Wang, Kai et al. 2019. "E-Commerce Personalized Recommendation Analysis by Deeply-Learned Clustering." *Journal of Visual Communication and Image Representation*: 102735.
- Wijaya, Anderias Eko, and Deni Alfian. 2018. "Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering." *Jurnal Computech & Bisnis* 12(1): 11–27.