

## IMPLEMENTASI METODE *HYBRID COLLABORATIVE FILTERING* DAN *CONTENT-BASED FILTERING* UNTUK REKOMENDASI LAGU

Ariel Jusuf Indrastata<sup>1</sup>, Yudi Wibisono<sup>2</sup>, Muhammad Nursalman<sup>3</sup>  
Universitas Pendidikan Indonesia<sup>123</sup>

Jl. Dr. Setiabudhi No. 229 Bandung 40154 Jawa Barat - Indonesia  
E-mail : ariel@upi.edu<sup>1</sup>, yudi@upi.edu<sup>2</sup>, mnursalman@upi.edu<sup>3</sup>

### ABSTRAK

Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi lagu menggunakan metode gabungan *collaborative filtering* dan *content-based filtering*. *Collaborative filtering* merekomendasikan lagu berdasarkan keterkaitan selera antar pengguna. *Content-based filtering* memberikan rekomendasi lagu yang memiliki kemiripan dalam karakteristik audionya, seperti *genre*, tempo, dan nada. *Hybrid filtering* adalah metode rekomendasi yang menggabungkan dua atau lebih metode. Data yang digunakan yaitu riwayat pemutaran lagu oleh pengguna dan data fitur audio dari lagu. Tahapan penelitian mencakup pengumpulan dan praproses data, proses rekomendasi dan, evaluasi. Hasil pengujian menunjukkan metode *collaborative filtering* dengan nilai *F1-Score* tertinggi, yaitu 0,0713, sedangkan metode *content-based filtering* memiliki nilai *F1-Score* terendah, yaitu 0,0060. Metode *weighted hybrid filtering* memiliki nilai *F1-Score* sebesar 0,0550, dan nilai *hit-rate* tertinggi dibanding dengan metode lainnya, yaitu 0,7137.

Kata kunci : Sistem Rekomendasi Lagu, *Collaborative Filtering*, *Content-Based Filtering*, *Hybrid Filtering*

### ABSTRACTS

The goal of this study is to develop a song recommendation system using hybrid filtering method by combining collaborative filtering and content-based filtering method. Collaborative filtering recommends songs based on the similarity of users' tastes. Content-based filtering provide song recommendations based on the similarities in their audio characteristics, such as genre, tempo, and pitch. Hybrid filtering is a recommendation method that combines two or more methods. The data that are used to build the recommendation system are users' song playback history and audio feature data from the songs. Test results show that the collaborative filtering method has the highest F1-Score value, 0.0713, while the content-based filtering method has the lowest F1-Score value, 0.0060. The weighted hybrid filtering method has an F1-Score value of 0.0550 and the highest hit-rate compared to other methods, 0.7137.

Keywords: Song Recommendation System, *Collaborative Filtering*, *Content-Based Filtering*, *Hybrid Filtering*

### 1. PENDAHULUAN

Musik menjadi sarana hiburan untuk berbagai kalangan masyarakat. Berkat kemajuan teknologi, musik digital kini telah menggantikan musik berbentuk fisik sebagai media utama untuk mengkonsumsi musik. Kemudahan dan fleksibilitas yang disediakan musik digital mempengaruhi tingkat konsumsi musik oleh masyarakat.

Spotify, sebagai salah satu layanan streaming musik, mencatat angka pengguna aktif di akhir tahun 2020 sebanyak 345 juta, meningkat 27% dari tahun lalu [1]. Musik digital juga memudahkan musisi untuk membagikan karya

mereka ke masyarakat. Terdapat lebih dari 70 juta lagu dalam katalog Spotify dan angka ini akan terus bertambah seiring dengan munculnya musisi-musisi baru.

Banyaknya lagu dengan berbagai subgenre belum tentu dapat dinikmati oleh setiap pendengar musik. Pengguna akan kewalahan untuk mendengarkan satu per satu hingga mendapat lagu yang cocok dengan selera mereka. Salah satu upaya untuk menangani hal tersebut adalah dengan membuat sebuah sistem rekomendasi yang akan membantu pendengar musik untuk menemukan lagu yang mereka sukai.

Sistem rekomendasi adalah sebuah sistem

penyaring yang bertujuan untuk memprediksi penilaian pengguna terhadap sebuah produk [2]. Dalam konteks musik, sistem rekomendasi menyaring kumpulan lagu hingga mendapat lagu yang relevan berdasarkan preferensi seseorang. Rekomendasi musik dapat dilakukan secara implisit, yaitu dengan melihat kebiasaan mendengar pengguna, sehingga pengguna tidak perlu diminta untuk menilai sebuah lagu agar dapat diberikan rekomendasi [3]. Beberapa metode yang digunakan untuk mendapatkan rekomendasi adalah melalui *collaborative filtering* dan *content-based filtering*.

*Collaborative filtering* membuat rekomendasi berdasarkan preferensi dari pengguna atau produk lain yang memiliki kesamaan. Dalam *collaborative filtering*, pengguna dicocokkan dengan pengguna lain berdasarkan keterkaitannya, dilihat dari riwayat pemutaran lagu oleh pengguna. Dengan menggunakan metode *collaborative filtering*, tidak diperlukan pengetahuan mengenai domain produk yang akan direkomendasikan. Selama tersedia data riwayat pengguna yang memadai, maka metode ini dapat digunakan. Namun, apabila tidak ada data riwayat pengguna terhadap suatu produk, maka metode *collaborative filtering* tidak dapat digunakan [4].

Berbeda dengan *collaborative filtering*, *content-based filtering* menghasilkan rekomendasi tanpa harus ada data riwayat dari pengguna. Dengan perkembangan di bidang *music information retrieval* (MIR), karakteristik pada sebuah lagu dapat diekstrak. *Content-based filtering* bisa merekomendasikan musik yang tidak populer selama ada kemiripan suara dengan musik yang disukai oleh pengguna. Kelemahan dari sistem rekomendasi musik yang menggunakan metode ini adalah hasil rekomendasi yang terbatas, karena hanya melihat dari minat seorang pengguna, sehingga akan kurang optimal bagi mereka yang ingin memperluas selera musik mereka.

Penelitian ini akan mengembangkan sistem rekomendasi musik dengan menggabungkan metode *collaborative filtering* dan *content-based filtering*. Dengan perancangan sistem *hybrid* ini, diharapkan dapat memberikan hasil rekomendasi yang baik, sehingga pendengar musik terbantu akan adanya sistem rekomendasi ini.

## 2. METODE PENELITIAN

Penelitian ini diawali dengan pengumpulan dan praproses *dataset*. Jenis data yang digunakan untuk *collaborative filtering* dan *content-based filtering* adalah berbeda. Data yang sudah dikumpulkan dan dipraproses, akan digunakan untuk membangun sistem rekomendasi lagu menggunakan metode *collaborative filtering*,

*content-based filtering*, dan gabungan kedua metode tersebut (*hybrid*). Sistem akan dievaluasi untuk melihat tingkat akurasi hasil rekomendasi.

### 2.1 Pengumpulan Data

Untuk penelitian ini, metode *collaborative filtering* menggunakan *dataset* LFM-1B, yang berisi riwayat pemutaran lagu oleh pengguna situs *Last.fm* [5]. *Dataset* berisi lebih dari 1 miliar aktivitas pemutaran lagu dengan *timestamp*-nya. Metode *content-based filtering* menggunakan data fitur audio yang bersumber dari *dataset FreeMusicArchive* [6]. Fitur audio yang ada di *dataset FreeMusicArchive* adalah hasil ekstraksi audio lagu menggunakan *library Python librosa*. Jumlah lagu yang terdapat di *dataset* adalah 106.574 dengan 518 fitur audionya, seperti fitur *chroma*, fitur *spectral*, RMSE, *zero-crossing rate* dan *Mel-Frequency Cepstral Coefficient* (MFCC).

### 2.2 Praproses Data

Kedua *dataset* yang sudah dikumpulkan akan melalui tahapan praproses agar dapat digunakan pada metode rekomendasi. Data riwayat pemutaran lagu diintegrasikan berdasarkan pengguna, sehingga menampilkan total jumlah pemutaran setiap lagu untuk setiap pengguna dalam *dataset*. Data fitur audio dilakukan proses normalisasi dan *feature selection* menggunakan *Variance Threshold* untuk membuang fitur audio yang varians-nya berada di bawah ambang batas. Kedua *dataset* kemudian digabung dengan mencari lagu yang *overlap*, sehingga hanya data dengan lagu yang ada di kedua *dataset* yang akan digunakan untuk membuat sistem rekomendasi. Data kemudian di-*split* menjadi data *training* dan data *testing* untuk evaluasi.

### 2.3 Collaborative Filtering

*Collaborative filtering* menghasilkan rekomendasi dengan melihat keterkaitan minat pengguna dengan pengguna lain berdasarkan *item* yang disukainya [7]. Penelitian ini menggunakan data riwayat pemutaran lagu oleh pengguna. Data yang sudah melalui praproses diubah ke bentuk *user-item matrix*. Matriks menunjukkan interaksi antara pengguna dengan lagu, di mana jumlah pemutaran lagu adalah *value*-nya dan merepresentasikan *rating* pengguna terhadap lagu.

Penelitian ini menggunakan metode *user-based collaborative filtering*, mencocokkan pengguna dengan pengguna lain berdasarkan lagu yang disukainya. Masing-masing pengguna akan dihitung keterkaitan minatnya (*similarity*) antar pengguna lain menggunakan *cosine similarity*. Hasil rekomendasi seorang pengguna akan dipengaruhi besar oleh nilai *similarity* pengguna

lain yang tinggi, yang artinya pengguna tersebut memiliki selera yang mirip dengan target pengguna yang akan diberikan rekomendasi lagu. Lagu yang akan direkomendasikan ke pengguna adalah lagu dengan hasil perkalian nilai *similarity* pengguna lain dengan *rating* lagunya.

**2.4 Content-Based Filtering**

*Content-based filtering* memanfaatkan karakteristik konten audio untuk melakukan pencocokan berdasarkan kesamaan konten [8]. Penelitian ini menggunakan algoritma *K-nearest neighbor* (KNN) untuk mencari lagu yang mirip berdasarkan fitur audio-nya. Untuk menghasilkan rekomendasi, metode *content-based filtering* akan mengambil fitur audio dari lagu yang diputarkan paling banyak oleh pengguna, kemudian menghitung jarak dengan lagu lainnya menggunakan metrik *euclidean distance*. Lagu yang akan direkomendasikan adalah lagu dengan nilai jarak terkecil.

**2.5 Hybrid Filtering**

Metode hybrid dalam sistem rekomendasi adalah metode yang menggabungkan dua atau lebih metode lain. Untuk menggabungkan metode *collaborative filtering* dan *content-based filtering*, adalah dengan menampilkan hasil rekomendasi dari masing-masing metode, atau disebut *mixed* [9]. Cara lain adalah dengan mempertimbangkan *weight* pada masing-masing metode. Untuk rasio *weight* 50:50, maka masing-masing metode memiliki kepentingan yang sama. Sedangkan untuk rasio *weight* 70:30, di mana metode *collaborative filtering* adalah 70 dan metode *content-based* adalah 30, maka hasil rekomendasi dari metode *collaborative filtering* lebih diutamakan. Untuk menghitung *weighted average*, hasil rekomendasi lagu dari masing-masing metode dikalikan dengan rasio *weight*-nya. Pada *collaborative filtering*, nilai *similarity* dikalikan dengan rasio-nya, sedangkan untuk *content-based filtering*, nilai *distance* dikalikan dengan rasio-nya, kemudian hasilnya dijumlahkan dan dibagi 2. Lagu dengan nilai *weighted average* tertinggi akan ditampilkan sebagai rekomendasi teratas.

**3. HASIL DAN PEMBAHASAN**

**3.1 Pengumpulan dan Praproses Dataset**

Data aktivitas pendengaran lagu yang bersumber dari *dataset* LFM-1B di-agregasi berdasarkan pengguna dan lagu, sehingga menampilkan jumlah berapa kali seorang pengguna memutar sebuah lagu. Contoh data dapat dilihat pada Tabel 1.

Tabel 1. Contoh data interaksi pemutaran lagu oleh pengguna

ID Pengguna	ID Artis	ID Lagu	Jumlah Pemutaran
31435741	2	4	13
31435741	13	17	23
31435741	13	19	22
31435741	21	53	54
31435741	32	86	84

Dari Tabel 1, ditampilkan pengguna dengan ID “31435741”, yang memutar 5 lagu berbeda dari 5 artis yang berbeda. Dari kelima lagu yang ditampilkan pengguna tersebut paling banyak memutar lagu dengan ID “86”, yang bisa berarti pengguna paling suka lagu tersebut dibandingkan dengan lagu lainnya.

Data fitur audio bersumber dari *dataset* *FreeMusicArchive*. Proses normalisasi dilakukan menggunakan *standard scaler* dan seleksi fitur menggunakan *variance threshold* dengan nilai *threshold* sebesar 0,25, sehingga jumlah fitur yang awalnya sebanyak 518 menjadi 512 fitur. Tabel 2 menunjukkan contoh data fitur audio lagu.

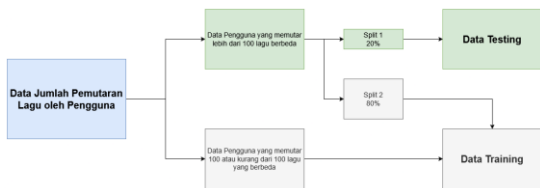
Tabel 2. Contoh data fitur audio lagu

ID Lagu	zcr_min	zcr_skew	zcr_std
10	-0,506919	0,110545	0,059874
144	-0,506919	-1,021056	-0,194701
145	-0,506919	-0,734929	-0,372075

Tabel 2 menampilkan contoh data 3 dari 512 fitur audio pada lagu yang telah dinormalisasi. Fitur audio yang ditampilkan pada contoh Tabel 2 adalah *zero-crossing rate* dengan ukuran statistik berupa nilai minimumnya, *skewness*, dan standar deviasi-nya.

Kedua *dataset* memiliki jumlah dan daftar lagu yang berbeda, sehingga diperlukan pencarian *overlap* lagu agar metode bisa digunakan bersamaan. Dari hasil *overlap* ditemukan beberapa

lagu duplikat dengan ID berbeda dari kedua *dataset*, sehingga lagu tersebut tidak akan digunakan untuk membuat sistem rekomendasi. Dari proses tersebut, ditemukan 21.580 lagu yang *overlap*. Kemudian dilakukan *split* menjadi data *training* dan data *testing*. Data jumlah pemutaran lagu oleh pengguna dijadikan sebagai data *testing*. *Split* data *testing* dapat dilihat di Gambar 3.1.



Gambar 3.1 Penentuan *split* data *training* dan data *testing*

Berdasarkan gambar di atas, data *testing* dipilih dari data pengguna yang memutar lebih dari 100 lagu berbeda. Kemudian data tersebut diambil 20% sebagai data *testing*, dan sisanya sebagai data *training* bersama data pengguna yang memutar kurang dari 100 lagu yang berbeda. Jumlah data setelah proses *split* dapat dilihat di Tabel 3.

Tabel 3. Jumlah data *training* dan data *testing*

Jumlah Data	Data Training	Data Testing
Interaksi Pengguna dengan Lagu	387.673	8.300
Pengguna	43.069	248
Lagu	20.978	4.433

Dari gambar di atas, data *testing* terdiri dari 248 pengguna. Rekomendasi akan diberikan kepada pengguna tersebut dan kemudian diuji akurasi.

### 3.2 Implementasi Collaborative Filtering

Metode *collaborative filtering* akan memberikan rekomendasi dengan menghitung *similarity* antar-pengguna. Data interaksi pemutaran lagu diubah ke bentuk *user-item matrix* agar dapat dihitung *cosine similarity*-nya menggunakan *library Python Scikit-Learn*. Setelah diubah ke bentuk matriks, *value* dari matriks dinormalisasi ke nilai dengan jangkauan antara 0 sampai dengan 5 sebagai representasi *rating* pengguna. Bentuk *user-item matrix* dapat dilihat di Gambar 3.2.

track_id	18610	20410	20420	20429	20611	20613	20614	20618	20626
user_id									
1206	0.0	0.0	0.0	0.0	0.714286	0.714286	0.714286	0.0	2.857143
3653	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	1.250000
4813	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000
4830	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000
5069	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000

Gambar 3.2 *User-item matrix*

Berdasarkan gambar di atas, dapat dilihat matriks di mana kolomnya berupa ID lagu dan barisnya berupa ID pengguna. *Value* dari setiap baris adalah jumlah pemutaran lagu oleh pengguna yang sudah dinormalisasi. Angka 0 pada matriks menandakan bahwa pengguna tidak pernah memutar lagu tersebut. Dengan *function "cosine\_similarity"* dari *Scikit-Learn*, dihitung nilai *similarity* sehingga menghasilkan matriks *user-user* yang dapat dilihat di Gambar 3.3.

user_id	1206	3653	4813	4830	5069
user_id					
1206	1.000000	0.058465	0.0	0.000000	0.007079
3653	0.058465	1.000000	0.0	0.40482	0.000000
4813	0.000000	0.000000	1.0	0.000000	0.000000
4830	0.000000	0.404820	0.0	1.000000	0.000000
5069	0.007079	0.000000	0.0	0.000000	1.000000

Gambar 3.3 Hasil perhitungan *cosine similarity*

Gambar di atas menunjukkan nilai *similarity* antar pengguna. Semakin tinggi nilai *similarity*, maka minat antara kedua pengguna semakin mirip. Untuk menghasilkan rekomendasi, akan dihitung *weighted average* dengan cara menjadikan nilai *similarity* masing-masing pengguna sebagai *weight* dan kemudian dikalikan dengan *rating* lagunya masing-masing, lalu dibagi dengan total *similarity* seluruh pengguna terhadap pengguna target. Lagu dengan nilai hasil perhitungan terbesar adalah lagu yang akan direkomendasikan. Dengan mengambil data judul lagu dan nama artis dari *dataset LFM-1B*, maka dapat ditampilkan hasil rekomendasi seperti pada Gambar 3.4.

track_id	weighted_avg	title	artist_name
0	199891	0.659591	One Big Holiday My Morning Jacket
1	1327520	0.480456	Butterfly Delicate Steve
2	226162	0.444544	Green Typewriters The Olivia Tremor Control
3	130367	0.398450	9 Ghosts I Nine Inch Nails
4	451328	0.353950	Grande Comore Southern Shores
5	179967	0.299291	7 Ghosts I Nine Inch Nails
6	1252500	0.183197	Listen Close Constrobuz
7	942056	0.158391	C.L.U.B. MNDR
8	93295	0.156363	Echoplex Nine Inch Nails
9	186771	0.154129	999,999 Nine Inch Nails

Gambar 3.4 Hasil rekomendasi metode *collaborative filtering*

Gambar di atas adalah contoh hasil rekomendasi dari metode *collaborative filtering* untuk seorang pengguna yang terdiri atas 10 lagu. Lagu dengan *weighted average* tertinggi akan ditampilkan paling atas.

### 3.3 Implementasi Content-Based Filtering

Pada metode *content-based filtering* yang menggunakan algoritma *K-nearest neighbor*, rekomendasi lagu dilakukan dengan cara melihat lagu dengan pemutaran terbanyak oleh seorang pengguna, kemudian menghitung *euclidean distance* dengan lagu lain berdasarkan fitur audionya. Lagu dengan nilai *distance* terkecil atau terdekat adalah lagu yang akan direkomendasikan ke pengguna. Contoh hasil rekomendasi dengan metode *content-based filtering* dapat dilihat di Gambar 3.5.

fma_track_id	distance	rank	title	artist_name
0	75267	19.738416	1	Álomcsapda Obsidian Shell
1	75272	21.640652	2	Emlékmás Obsidian Shell
2	75246	32.776687	3	Óú Molecul
3	75291	33.359386	4	D.T.I.O.G. ID:VISION
4	75288	33.601210	5	Disphenoid's Equilibrium ID:VISION
5	75286	33.838017	6	The Axe Of Wrath ID:VISION
6	75293	34.709054	7	Decagon Deathcamp ID:VISION
7	75287	37.638108	8	Doden Force Division ID:VISION
8	75292	38.403429	9	Deathcamp Prelude ID:VISION
9	75285	38.404752	10	Mel(un)holy ID:VISION

Gambar 3.5 Hasil rekomendasi metode *collaborative filtering*

Gambar di atas adalah contoh hasil rekomendasi dari metode *content-based filtering* untuk seorang pengguna yang terdiri atas 10 lagu. Lagu dengan nilai *distance* terkecil akan ditampilkan paling atas.

### 3.4 Implementasi Hybrid Filtering

*Hybrid filtering* adalah menggabungkan dua metode *filtering* untuk menghasilkan rekomendasi. Penelitian ini menggunakan dua jenis *hybrid filtering*, yaitu *mixed* dan *weighted*. Pada *mixed hybrid filtering*, hasil rekomendasi metode *collaborative filtering* dan *content-based filtering* ditampilkan. Jika akan merekomendasikan 10 lagu kepada seorang pengguna, maka *mixed hybrid filtering* akan menampilkan 5 lagu rekomendasi teratas dari metode *collaborative filtering* dan *content-based filtering*. Pada *weighted hybrid filtering*, nilai *similarity* dan *distance* pada lagu

hasil rekomendasi masing-masing metode dikalikan dengan rasio *weight*, lalu dijumlahkan dan dibagi 2. Berbeda dengan nilai *similarity*, semakin besar nilai *distance*, maka semakin tidak direkomendasikan. Oleh karena itu nilai *distance* diubah dengan cara invers agar selaras dengan nilai *similarity*, sehingga dapat dihitung *weighted average*. Contoh hasil rekomendasi oleh metode *weighted hybrid filtering* dapat dilihat di Gambar 3.6.

fma_track_id	track_id	weighted_avg	distance	weight_final	rank	
0	29625	381252	0.390805	1.000000	0.695402	1
1	29609	607132	1.000000	0.052856	0.526428	2
2	75239	753306	0.871700	0.031661	0.451680	3
3	75238	1348242	0.787179	0.029085	0.408132	4
4	29610	222408	0.396036	0.054254	0.225145	5
5	75244	1337737	0.412197	0.030367	0.221282	6
6	24014	756686	0.392724	0.035200	0.213962	7
7	29611	607131	0.369652	0.052363	0.211007	8
8	50288	606908	0.357627	0.031912	0.194769	9
9	39527	132105	0.324948	0.039475	0.182211	10

Gambar 3.6 Hasil rekomendasi metode *weighted hybrid filtering*

Pada gambar di atas, nilai "*weighted\_avg*" adalah nilai *weight* yang didapat dari perhitungan *similarity* dengan *rating* pada metode *collaborative filtering*. Nilai "*distance*" adalah hasil perhitungan *euclidean distance* antar lagu berdasarkan fitur audionya. Nilai "*weight\_final*" adalah hasil perhitungan *weight* akhir dan menentukan lagu yang akan direkomendasikan paling atas. Rasio yang digunakan pada contoh di atas adalah 50:50.

### 3.5 Hasil Pengujian

Metode rekomendasi dievaluasi dengan cara mengecek apakah lagu hasil rekomendasi kepada pengguna ada di data *testing* atau tidak. Evaluasi dilakukan kepada 248 pengguna yang data pemutaran lagunya ditemukan di data *testing*. Pemutaran lagu minimal oleh pengguna pada data *testing* yang ditetapkan adalah lebih dari 1. Apabila di dalam data *testing* pengguna hanya memutar sebuah lagu sekali saja, maka lagu tersebut tidak akan dihitung sebagai *true positive*.

Metrik pengujian yang digunakan adalah *precision*, *recall*, *F1-score* dan *hit-rate*. Dengan *hit-rate*, dapat diketahui berapa banyak pengguna dari total 248 yang minimal mendapatkan 1 rekomendasi lagu yang benar. Rasio yang ditetapkan sebagai *weight* pada metode *weighted hybrid filtering* adalah 50:50. *K* atau jumlah rekomendasi yang ditetapkan adalah 10. Perbandingan hasil pengujian dapat dilihat pada Tabel 4.

0,7137, namun masih kalah unggul dalam lainnya dibandingkan dengan metode *collaborative filtering*. Kinerja *hybrid filtering* bisa dipengaruhi oleh kinerja masing-masing metode yang digabungkannya.

#### 4.2 Saran

Saran yang dapat digunakan untuk penelitian mengenai sistem rekomendasi musik berbasis *hybrid* selanjutnya, yaitu:

- a. Menambahkan data konteks pendengaran lagu, seperti waktu, lokasi, jenis kelamin, dan usia pengguna, agar hasil rekomendasi lebih personal dan adaptif.
- b. Menerapkan model machine learning yang berbeda untuk meningkatkan kinerja akurasi rekomendasi.
- c. Melibatkan pengguna secara langsung untuk uji coba sistem rekomendasi agar dapat diketahui kualitas rekomendasi secara subjektif.

#### DAFTAR PUSTAKA

- [1]. Business Wire, "Spotify Technology S.A. Announces Financial Results for Fourth Quarter 2020", 2021, [Online], Available: <https://www.businesswire.com/news/home/20210203005304/en/>
- [2]. Ricci F, Rokach L, Shapira B, "Introduction To Recommender Systems Handbook", Recommender systems handbook, pp 1-35, Springer, 2011.
- [3]. Celma Ò, "Music Recommendation", Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space, pp 43-85, 2010.
- [4]. Burke R, "Integrating Knowledge-based and Collaborative-filtering Recommender Systems", 2000.
- [5]. Schedl M, "The LFM-1b Dataset for Music Retrieval and Recommendation", Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, pp 103-110, 2016, doi: 10.1145/2911996.2912004
- [6]. Defferrard M, Benzi K, Vandergheynst P, Bresson X, "FMA: A Dataset For Music Analysis", CoRR, 2016.
- [7]. Herlocker J, Konstan J, Terveen L, Riedl J, "Evaluating collaborative filtering recommender systems", ACM Trans. Inf. Syst. 2004; 22(1):5-53.

Tabel 4. Hasil pengujian untuk  $K = 10$

Metrik	CF	CB	Mixed HF	Weighted HF
Precision	<b>0,1339</b>	0,0133	0,0859	0,1310
Recall	<b>0,0486</b>	0,0039	0,0310	0,0475
F1-Score	<b>0,0713</b>	0,0060	0,0443	0,0680
Hit-Rate	0,7097	0,1089	0,5847	<b>0,7137</b>

Berdasarkan Tabel 4, metode *collaborative filtering* (CF) memiliki nilai *precision*, *recall*, dan *F1-Score* tertinggi dibanding metode lainnya, yaitu nilai *F1-Score* 0,0713. Metode dengan kinerja terendah adalah *content-based filtering* (CB) dengan nilai *F1-Score* 0,006. Untuk *hit-rate*, metode *weighted hybrid filtering* (*Weighted HF*) adalah yang terbaik dan mengungguli CF dengan nilai sebesar 0,7137. Artinya sebanyak 71% dari 248 pengguna yang dievaluasi mendapatkan setidaknya satu lagu rekomendasi yang cocok. Kalahnya kinerja metode *hybrid filtering* bisa disebabkan karena kinerja *content-based filtering* yang buruk, bisa terjadi karena kurangnya interaksi antar pengguna dengan lagu dalam data.

#### 4. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Penelitian ini mengembangkan sistem rekomendasi lagu menggunakan metode *hybrid*, yaitu metode *collaborative filtering* dan metode *content-based filtering*. Data yang digunakan berupa riwayat pemutaran lagu yang bersumber dari dataset LFM-1B dan data fitur audio yang bersumber dari dataset *FreeMusicArchive*. Data dilakukan praproses agar dapat diolah oleh algoritma untuk menghasilkan rekomendasi lagu. Pengujian dilakukan ke masing-masing metode untuk mengukur kinerja sistem rekomendasi. Metode *collaborative filtering* memiliki nilai *F1-Score* tertinggi, yaitu 0,0713, sedangkan metode *content-based filtering* memiliki nilai *F1-Score* terendah, yaitu 0,006. Metode *weighted hybrid filtering* memiliki nilai *hit-rate* tertinggi, yaitu

- [8]. Srinivasa Murthy Y, Koolagudi S, "Content-Based Music Information Retrieval (CB-MIR) and Its Applications toward the Music Industry: A Review", *ACM Computing Surveys* 2018; 51:1-46.
  
- [9]. Burke R, "Hybrid Recommender Systems: Survey and Experiments", *User Modeling and User-Adapted Interaction* 2002; 12.