

# Penerapan Kombinasi Algoritma *Minhash* dan *Binary Hamming Distance* pada *Hybrid* Rekomendasi Lagu

Lutvi Satriyo Putro  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Kentingan  
Surakarta 57126  
lutvi\_sp@student.uns.ac.id

Ristu Saptono, S.Si., M.T.  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Kentingan  
Surakarta 57126  
r\_saptono@uns.ac.id

Rini Anggrainingsih, S.T.,M.T.  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Kentingan  
Surakarta 57126  
rinianggra@uns.ac.id

## ABSTRAK

Teknik yang biasanya digunakan untuk menghasilkan rekomendasi adalah *content-based*, *social-based* dan *hybrid*. *Content-based* dan *social-based* adalah teknik rekomendasi yang berdiri sendiri, sedangkan *hybrid* adalah kombinasi dari beberapa teknik rekomendasi, seperti kombinasi *knowledge-based* dan *social-based* pada aplikasi *EntreeC* dan kombinasi *social-based* dan *content-based* pada *DailyLearner*. Teknik *hybrid* menghasilkan rekomendasi yang lebih baik daripada teknik rekomendasi yang berdiri sendiri, seperti meningkatkan akurasi rekomendasi dan mempercepat produksi rekomendasi. Akan tetapi, berdasarkan penelitian sebelumnya, *hybrid social-based* dan *content-based* yang diterapkan pada data lagu, tidak menghasilkan rekomendasi yang lebih baik. Oleh karena itu, pada penelitian ini kombinasi *social-based* dan *content-based* yang diterapkan pada data lagu akan dikombinasikan dengan strategi *switching*, *mixed*, dan *cascade* yang diharapkan mampu menghasilkan keluaran yang lebih baik.

Hasil dari penelitian ini, *hybrid* yang menggunakan strategi *switching*, *cascade*, dan *mixed* mampu membuat rekomendasi yang nilai akurasinya melebihi rekomendasi yang dihasilkan *social-based* dan *content-based* murni pada data yang memiliki nilai *sparsity* 0,94. Hal ini dibuktikan dengan nilai rata-rata akurasi *hybrid* yang mencapai 0,48, nilai *minhash* 0,17 dan nilai *hamming* 0,42. Sedangkan untuk data yang memiliki nilai *sparsity* 0,96, algoritma ini nilai akurasinya dibawah algoritma *binary hamming distance*. Hal ini dibuktikan dengan nilai rata-rata akurasi *hybrid* 0,4, nilai *minhash* 0,06 dan nilai *binary hamming distance* 0,41.

## Kata Kunci :

*Binary Hamming Distance*, *Content-based*, *Minhash*, *Social-based*

## 1. PENDAHULUAN

Perkembangan teknologi digital, internet dan entertainment yang pesat turut mempengaruhi perkembangan lagu di dunia. Hal ini memunculkan sebuah gagasan untuk membuat sebuah sistem rekomendasi lagu yang akan memberikan rekomendasi bagi user yang bingung dalam memilih sebuah lagu untuk didengarkan.

Secara teori, ada 5 teknik rekomendasi yang bisa digunakan, yaitu *social-based*, *content-based*, *demographic*, *utility-based* dan *knowledge-based* [1].

Selain 5 teknik diatas, terdapat teknik yang mengkombinasikan 2 atau lebih teknik rekomendasi, yang disebut dengan *hybrid*. Kombinasi yang paling sering dilakukan adalah *content-based* dan *social-based*, akan tetapi penelitian yang dilakukan oleh Cai dan kawan-kawan menyatakan bahwa teknik *hybrid* ini tidak menghasilkan keluaran yang lebih baik daripada teknik *content-based* dan *social-based* murni [2]. Metode *social-based* memiliki kekurangan yaitu *cold-start problem*. *Cold-start problem* merupakan kondisi dimana jika terdapat *item* baru, metode *social-based* tidak dapat menghasilkan rekomendasi berupa *item* baru tersebut, karena *item* tersebut tidak memiliki *rating* [3]. Kelemahan tersebut dapat diatasi dengan *first-rater* yang dimiliki oleh metode *content-based*. Disisi lain, metode *content-based* memiliki kelemahan yaitu *overspecialization* yang dapat diatasi oleh metode *social-based*.

Pada penelitian kali ini, dilakukan teknik *hybrid* dengan menggunakan metode *switching*, *cascade*, dan *mixed* yang diharapkan mampu menghasilkan keluaran yang lebih baik. Teknik *hybrid* yang dilakukan menggabungkan algoritma *Minhash* (*social-based*) dan *Binary Hamming Distance* (*content-based*). Kedua algoritma diatas memiliki persamaan yaitu digunakan untuk menganalisa hubungan antar *item* [4][5]. Algoritma *Minhash* akan menemukan kombinasi *item* yang akan dipakai oleh *user* dan algoritma *Binary Hamming Distance* juga akan meningkatkan rekomendasi. Penggabungan kedua algoritma tersebut dalam penelitian ini diharapkan agar menghasilkan rekomendasi yang lebih akurat.

## 2. DASAR TEORI

### 2.1 *Social-based Recommendation*

*Social-based* lebih terkenal dengan nama *collaborative filtering*. Metode ini merekomendasikan suatu *item* berdasarkan kedekatan *user* yang memberikan *rating*. Fitur-fitur dalam *item* tidak dipedulikan sehingga rekomendasi yang diberikan bersifat subyektif. *Social-based* paling banyak digunakan dalam rekomendasi karena metode ini yang paling mudah untuk diimplementasikan.

Metode *social-based* memiliki beberapa kelemahan diantaranya *cold-start problem* dan *sparsity*. *Cold-start problem* merupakan kondisi dimana sistem rekomendasi tidak mampu memberikan rekomendasi *item* baru kepada *user* karena *item* baru tidak memiliki *rating*. *Rating* merupakan kunci pada metode ini, jika sebuah *item* tidak memiliki *rating* maka tidak dapat direkomendasikan. Kelemahan lain dari *social-based* yaitu *sparsity* atau tingkat kerenggangan data. Semakin tinggi nilai *sparsity* atau

disebut *sparse* (renggang), maka rekomendasi yang diberikan oleh *social-based* cenderung baru [6].

**2.2 Content-based Recommendation**

*Content-based* merupakan kebalikan dari *social-based*, yaitu rekomendasi berdasarkan fitur-fitur yang ada di dalam *item*. *Item* hasil rekomendasi adalah *item* yang memiliki fitur yang sama atau mendekati fitur *item* yang sudah pernah dirating *user* dan memiliki *rating* tertinggi [7]. Pemberian rating pada item baru agar item baru tersebut dapat direkomendasikan disebut *first-rater*. Metode ini cukup baik untuk *item* baru yang belum memiliki *rating*.

Metode *content-based* memiliki beberapa kelemahan diantaranya yaitu *overspecialization* dan *new user problem*. *Overspecialization* merupakan kondisi dimana rekomendasi yang diberikan cenderung bisa ditebak. Hal ini bisa terjadi karena faktor rekomendasi yang dipakai oleh *content-based* adalah kemiripan konten. Maka, jika *user* memilih sebuah *item*, maka *item* yang akan direkomendasikan selalu memiliki konten yang tingkat kemiripannya tinggi. *New user problem* adalah masalah yang muncul untuk *user* baru yang belum memberikan rekomendasi. *User* ini tidak akan diberikan rekomendasi sebelum ia memilih *item* [3].

Teknik *content-based* dan *social-based* sama-sama memiliki kelemahan dan kelebihan, berikut tabel perbandingan kelemahan dan kelebihan kedua metode tersebut [1][8][9] :

Teknik	Kelebihan	Kelemahan
<i>Social-Based</i>	1. Dapat mengidentifikasi cross-genre dengan baik 2. Pengetahuan tentang item tidak dibutuhkan 3. Adaptif (dapat berubah seiring waktu) 4. Memakai masukan implisit	1. Cold-start problem 2. Kumpulan informasi dapat hilang. 3. Hanya cocok untuk data dengan sparsity rendah
<i>Content-Based</i>	1. Pengetahuan tentang item tidak dibutuhkan 2. Adaptif (dapat berubah seiring waktu) 3. Memakai masukan implisit	1. Overspecialization 2. New user problem 3. Kualitas rekomendasi tergantung pada konten

**2.3 Hybrid**

*Hybrid* merupakan gabungan dari dua atau lebih teknik rekomendasi. Ada beberapa metode penggabungan, yaitu sebagai berikut [1] :

Metode	Deskripsi
Weighted	Masing-masing teknik diberikan nilai rekomendasi dan digabung menghasilkan satu rekomendasi.
Switching	Sistem memilih teknik rekomendasi mana yang sesuai dengan kondisi yang

	ada.
Mixed	Menggunakan beberapa teknik rekomendasi bersama-sama.
Feature Recommendation	Fitur-fitur milik teknik rekomendasi yang dipakai dijadikan satu kemudian melakukan rekomendasi
Cascade	Rekomendasi yang menyempurnakan rekomendasi teknik sebelumnya.
Feature Augmentation	Rekomendasi dari teknik rekomendasi yang lain, digunakan sebagai bahan untuk melakukan rekomendasi.
Meta-Level	Model pembelajaran dari suatu teknik rekomendasi yang digunakan sebagai inputan rekomendasi yang lain.

**2.4 Algoritma Minhash**

Algoritma *minhash* merupakan salah satu algoritma rekomendasi *social-based*. Algoritma ini adalah bagian dari *association rules*. Algoritma ini biasanya digunakan untuk data yang memiliki tingkat *sparse* cukup tinggi [4]. Algoritma *minhash* memiliki 3 tahapan :

1. Memetakan data
2. Mencari kandidat *rule*
3. Membuang kandidat *rule* yang tidak sesuai harapan.

Contoh *rule* :  $A \rightarrow B$

A disebut antecedent

B disebut consequent

Pada tahapan pemotongan kandidat *rule* yang tidak sesuai harapan membutuhkan nilai *confidence*. Nilai *confidence* dapat dihitung dari A dan B, dimana dari *rule* diatas, nilai *confidence* dapat dihitung dengan rumus sebagai berikut :

$$confidence(A \rightarrow B) = \frac{\text{jumlah Transaksi mengandung A dan B}}{\text{total Transaksi mengandung A}}$$

Kandidat *rule* yang dibuang adalah kandidat *rule* yang nilai *confidence*-nya lebih dari 90% atau 0,9 [7].

**2.5 Binary Hamming Distance**

*Hamming Distance* merupakan salah satu algoritma rekomendasi *content-based*. Metode ini merupakan salah satu algoritma mengukur kedekatan item. Jika nilai kedekatannya makin kecil, maka kedua item tersebut semakin dekat dan berlaku sebaliknya. Ada 2 macam *hamming distance*, yaitu berbentuk *string* dan *biner*. Untuk tipe *string*, cara menghitung kedekatan item dapat diukur dari nama (*string*) karena setiap item pasti memiliki nama. Hal sama dapat dilakukan untuk mengukur kedekatan dengan angka *biner*. Pengukuran *Hamming Distance* pada angka *biner* disebut metode *Binary Hamming Distance* [5].

**3. METODE PENELITIAN**

**3.1 Pengumpulan dan pemodelan Data**

Data yang dipakai dalam penelitian ini adalah *dataset* yang disediakan oleh website Last Fm. *Dataset* yang dipakai adalah *dataset* 1K, yang berisi daftar *user* dan lagu yang sudah diputar *user*. *Dataset* dari website Last Fm berupa data 1000 *user* dan daftar lagu yang sudah dimainkan dalam jangka waktu *user* pertama kali memainkan lagu di Last Fm sampai Januari 2012. *Dataset* yang diperoleh memiliki ukuran 2,3 GB. Data yang digunakan pada penelitian ini adalah data hasil penyaringan *dataset* dari website Last Fm.

**3.2 Penerapan dan perancangan Algoritma**

Teknik kombinasi yang digunakan pada penelitian ini yaitu *switching, cascade dan mixed*. Sedangkan algoritma yang dikombinasikan adalah *Minhash* dan *Binary Hamming Distance*.

Teknik *switching* digunakan ketika sistem mencari lagu baru. Jika ditemukan lagu baru, algoritma *Binary Hamming Distance* dijalankan untuk mencari lagu baru yang similar dengan *user*. Di saat bersamaan, dijalankan algoritma *Minhash*. Teknik ini disebut juga dengan teknik *mixed* karena menggabungkan 2 algoritma yaitu *Minhash* dan *Binary Hamming Distance*. Jika tidak ada lagu baru yang ditemukan, sistem akan menjalankan algoritma *minhash* tanpa menjalankan *Binary Hamming Distance*. Ketika *Minhash* dijalankan, akan tetapi rekomendasi yang dihasilkan belum sesuai dengan ketentuan, maka algoritma *Binary Hamming Distance* kembali dijalankan untuk menyempurnakan rekomendasi *minhash*. Teknik ini yang disebut dengan *cascade*. Perancangan sistem dilakukan dengan membuat skema tahapan algoritma *Minhash* dan *Binary Hamming Distance*

**3.3 Metode pengujian**

Pengujian dilakukan dengan menghilangkan beberapa transaksi secara acak. Setelah transaksi dihilangkan, maka akan dibangun *rule* menurut transaksi yang tersisa. Kemudian dihitung akurasinya dengan menghitung jumlah lagu yang dipilih dibagi dengan jumlah lagu yang hilang.

$$Akurasi = \frac{\text{jumlah lagu yang terpilih}}{\text{total jumlah lagu yang hilang}}$$

**4. PEMBAHASAN**

Data yang digunakan dalam penelitian ini diambil dari *website* resmi Last Fm. Data berupa daftar 1000 *user* dengan seluruh lagu yang mereka mainkan dengan rentang waktu mereka mendaftar sampai tahun 2012. Data yang diambil adalah data yang tersedia dalam metadata. Data yang diambil berupa *dataset* yang berisi judul lagu dan penyanyi, sedangkan data yang diambil dari *website* adalah data album, genre dan waktu pemutaran. Data yang didapat ini kemudian dimodelkan agar bisa dipakai oleh sistem. Teknik pengambilan data dari Last Fm menggunakan PHP/CURL.

**4.1 Pemodelan Data**

Data yang sudah dikumpulkan disaring berdasarkan lagu-lagu yang diputar lebih dari 1 *user*. Pada hasil penyaringan

didapatkan 46 *user* dan 255 lagu. Data ini disebut dengan data *user*. Contoh data *user* yang disajikan pada tabel 1 berikut :

**Tabel 1. Contoh data user dan lagu yang dipilih**

<b>Id_user</b>	<b>Id_lagu_pilihan</b>
1	77, 172, 173
2	12, 23, 29, 43, 68, 77, 79, 81, 82, 95, 108, 109, 110, 124, 129, 155, 165, 173, 178, 185, 188, 196, 207, 213, 242

Tahap selanjutnya yaitu merubah data *user* (tabel 1) menjadi data transaksi. Perubahan dilakukan dengan memisahkan *id\_lagu* untuk setiap *user* kemudian *id\_lagu* itu dimasukkan ke dalam *database*, sehingga data *user* tersebut menjadi data transaksi seperti contoh dibawah ini :

**Tabel 2. Contoh data transaksi**

<b>Id_user</b>	<b>Id_lagu</b>
1	77
1	172
1	173
2	12
2	23
2	29
2	77
2	173

*Rule* dibuat dari data transaksi dimana pembuatan *rule* berdasarkan pilihan lagu tiap *user*. *Rule* yang dibuat dinamakan calon *rule*, karena belum difilter nilai *confidence*-nya. Berikut calon-calon *rule* yang bisa dihasilkan dari tabel 2 :

**Tabel 3. Calon Rule**

<b>Antecedent</b>	<b>Consequent</b>	<b>Jumlah</b>	<b>Confidence</b>
77	172	1	0,72
77	173	2	1
172	173	1	0,54
12	23	1	0,44
12	29	1	0,23
12	77	1	0,76
12	173	1	0,85

Proses selanjutnya adalah memilih calon-calon *rule* yang nilai *confidence*-nya memenuhi *threshold*. Nilai *threshold* didapat dari pembagian antara jumlah transaksi dengan jumlah *user* dan jumlah lagu dengan rumus sebagai berikut :

$$\text{Minimum Threshold} = \frac{\text{jumlah transaksi}}{\text{jumlah user} * \text{jumlah item}}$$

Threshold yang didapatkan dalam penelitian ini adalah sebagai berikut :

$$\text{Minimum Threshold} = \frac{820}{46 * 255} = 0,069$$

Penelitian ini menggunakan algoritma *Minhash*, sehingga nilai *threshold* dikurangi 1. Maka, nilai mutlak *threshold confidence* yang dipakai adalah **0,931**. Nilai ini cukup tinggi, sehingga memenuhi aturan *minhash* yang mewajibkan nilai minimal *confidence* adalah 0,9 [6].

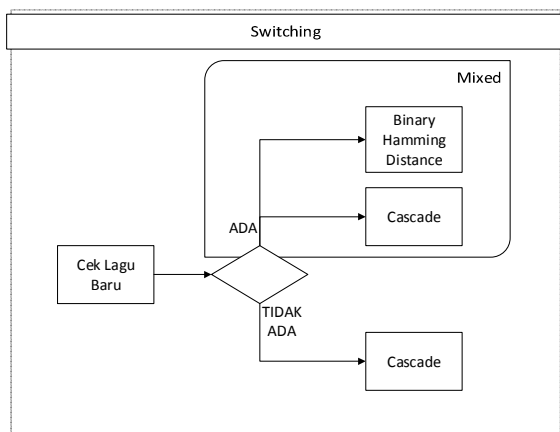
Pada tabel 3, data yang memenuhi *threshold* hanyalah data dengan antecedent 77 dan consequent 173. Data tersebut memiliki *confidence* lebih dari *threshold* yang telah ditentukan sedangkan data selain itu memiliki *confidence* dibawah *threshold*. Data dengan nilai *confidence* dibawah nilai *threshold* dibuang sehingga menghasilkan data *rule* seperti pada tabel berikut :

Tabel 4. Contoh rule

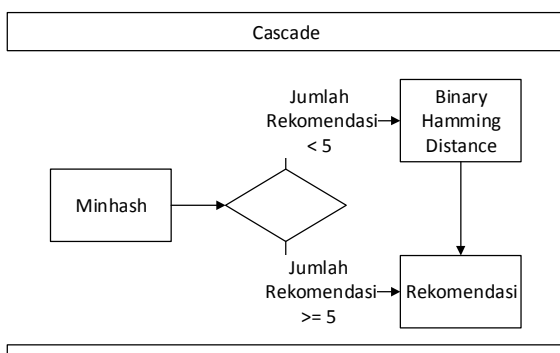
Antecedent	Consequent	Jumlah	Confidence
77	173	2	1

### 4.2 Penerapan Algoritma

Ada 3 metode *hybrid* yang dipakai, yaitu *switching*, *mixed*, dan *cascade*, sedangkan algoritma yang dipakai adalah *minhash* dan *binary hamming distance*. Hubungan 2 algoritma dan 3 metode tersebut dapat dilihat pada gambar berikut :



Gambar 1. Proses *switching* dan *mixed*



Gambar 2. proses *cascade*

### 4.2.1 Switching

Proses *switching* dilakukan ketika *user* memilih lagu. Sistem akan melakukan pencarian lagu baru di *database*. Jika ada lagu baru yang ditemukan, sistem akan memakai algoritma *Binary Hamming Distance* untuk mencari kemiripan dengan lagu yang dipilih *user*. Contoh rekomendasi lagu baru sebagai berikut.

Tabel 5. Contoh tabel lagu

No	Judul	Penyanyi	Genre
1	Kelly Watch The Stars	Air	Electronic
2	Back to Black	Amy Winehouse	Soul
3	Alone in Kyoto	Air	Ambient
4	La Femme d'argent	Air	Electronic

Misalkan *user* memilih lagu Kelly Watch The Stars yang dinyanyikan oleh Air dan mempunyai *genre Electronic*. Dari lagu ini, ada 2 informasi yang bisa diambil, yaitu Air (Penyanyi) dan *Electronic (Genre)*. Sistem akan mencari lagu yang baru yang mirip dengan lagu yang dipilih oleh *user* itu. Jika ada lagu yang mirip, lagu ini akan direkomendasikan kepada *user* bersama rekomendasi yang dihasilkan oleh *Minhash*. Jika tidak ada, maka rekomendasi yang diberikan hanya rekomendasi dari *Minhash*.

Untuk melakukan pencarian lagu yang mirip, algoritma ini mengubah informasi yang ada, penyanyi dan *genre*, ke dalam bentuk biner. Bentuk biner yang dihasilkan adalah 11. Ketika sistem mencari lagu, lagu yang dicari juga dibentuk menjadi biner. Tujuan perubahan ini, agar mempercepat proses pencarian [10].

### 4.2.2 Mixed

Proses *mixed* dilakukan ketika ada lagu baru yang ditemukan oleh sistem, lalu sistem mulai mencari kemiripan lagu baru dengan lagu yang dipilih dengan menggunakan algoritma *Binary Hamming Distance*, sistem juga mulai mencari kemiripan lagu yang lain dengan lagu yang dipilih oleh *user* dengan algoritma *Minhash*.

### 4.2.3 Cascade

Pada penelitian ini, algoritma *minhash* dijalankan terlebih dahulu karena menurut penelitian sebelumnya, penggabungan *content-based* dan *social-based*, dimana *content-based* didahulukan penggunaannya, akurasi rekomendasinya malah lebih rendah daripada *content-based* murni atau *social-based* murni.

Proses *cascade* dilakukan ketika algoritma *minhash* dijalankan. Setelah algoritma *minhash* menghasilkan rekomendasi, sistem akan menilai apakah rekomendasi yang dihasilkan sudah sesuai ketentuan atau belum. Jika belum, maka algoritma *Binary Hamming Distance* dijalankan untuk menyempurnakan rekomendasi ini. Walau demikian, sistem mengusahakan agar rekomendasi sudah memenuhi ketentuan dengan algoritma *Minhash* saja.

Untuk menghasilkan rekomendasi, *minhash* membutuhkan model *rule* yang memnuhi syarat *confidence*. Berikut adalah contoh model *rule* yang memnuhi syarat *confidence*.

Tabel 6. Contoh tabel rule

Antecedent	Consequent
82	79
95	185
89	13
29	213
82	207
80	82
80	85

Angka-angka pada tabel 6 adalah *id* lagu yang ada di *database*. Misalkan *user* memilih lagu yang memiliki *id* 80. Sistem akan mencari *id* 80 di kolom *antecedent*. Ketika sudah ditemukan, sistem akan merekomendasikan, lagu-lagu yang memiliki *id* di kolom *consequent* yang kolom *antecedent*-nya berisi *id* lagu 80. Dalam tabel di atas, sistem akan merekomendasikan *id* 82 dan 85.

Ketika *user* memilih lagu yang memiliki *id* 82 dan *id* itu adalah hasil rekomendasi, sistem akan kembali mencari *consequent* yang di kolom *antecedent* berisi *id* 82. Pada tabel di atas, *id* 79 dan 207 yang ditemukan. Sistem akan merekomendasikan 85,79 dan 207. *id* 85 bisa muncul kembali, karena *id* ini sudah pernah direkomendasikan pada *id* sebelumnya, sehingga sebenarnya memiliki kemungkinan yang cukup tinggi untuk dipilih. Sehingga, jika ada lagu yang belum dipilih oleh *user*, lagu ini akan direkomendasikan kembali oleh sistem pada pemilihan lagu berikutnya.

Pada penelitian ini, rekomendasi yang dihasilkan haruslah berjumlah 5 buah. Sehingga, jika rekomendasi yang dihasilkan oleh *minhash* urang dari 5, sistem akan memakai *binary hamming distance* untuk menyempurnakan hasil rekomendasi *minhash*.

**4.3 Pengujian dan analisa hasil**

Pengujian dilakukan dengan tujuan untuk mengukur akurasi rekomendasi lagu. Parameter yang digunakan ada 2 yaitu lagu yang terpilih dan lagu yang dihilangkan. Berikut rumus yang digunakan :

$$akurasi = \frac{\text{lagu terpilih}}{\text{lagu yang dihilangkan}}$$

Langkah pertama yang dilakukan untuk pengujian yaitu menghilangkan transaksi secara acak. Pada penelitian ini, jumlah yang dihilangkan adalah 25%, 50% dan 75%. Ketiga penghilangan tersebut didapatkan dari nilai tengah himpunan jumlah *rule*. Sebagai contoh, berikut data berupa *id\_lagu* dan jumlah pemutarannya :

**Tabel 7. Contoh jumlah pemutaran lagu**

Id lagu	Jumlah Pemutaran
1,3,8	2
5,6	3
2,4,9	4
7,11	5
10	6

Tabel diatas sudah diurutkan berdasarkan jumlah pemutaran lagu. Berdasarkan jumlah pemutaran, diambil nilai tengah dari nilai jumlah pemutaran lagu, yaitu 4. Selanjutnya dilakukan 2 macam penghilangan transaksi yaitu sebesar 25% dan 50%.

Dari setiap jumlah *rule* dilakukan pengujian sebanyak 30 kali. Ada 4 metode yang diuji : rekomendasi lagu yang menggunakan *binary hamming distance*, *minhash*, *minhash-binary hamming distance* dan kombinasi *minhash* dan *binary hamming distance*

Hasilnya dari pengujian dapat dilihat pada grafik di gambar 1 dan 2 :



**Gambar 3. Grafik Pengujian Penghilangan 25% (1 - 5)**



**Gambar 4. Grafik Pengujian Penghilangan 25% (6 - 10)**



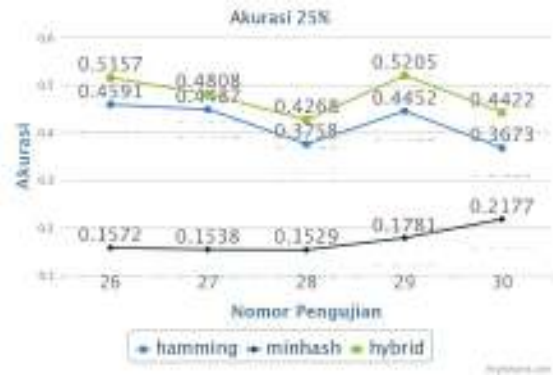
Gambar 5. Grafik Pengujian Penghilangan 25% (11-15)



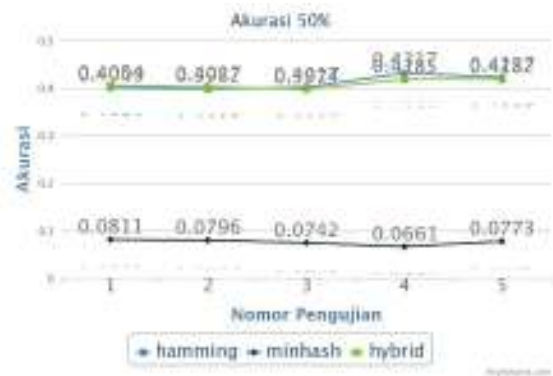
Gambar 6. Grafik Pengujian Penghilangan 25% (16-20)



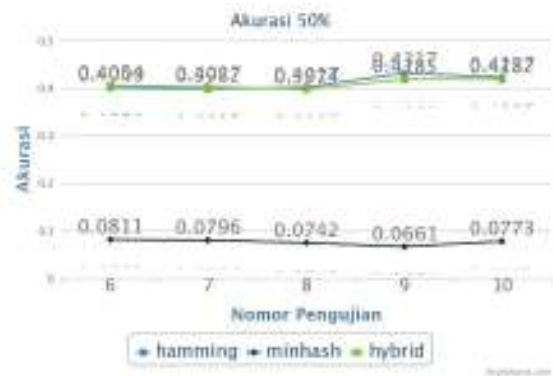
Gambar 7. Grafik Pengujian Penghilangan 25% (21-25)



Gambar 8. Grafik Pengujian Penghilangan 25% (26-30)



Gambar 9. Grafik Pengujian Penghilangan 50% (1-5)



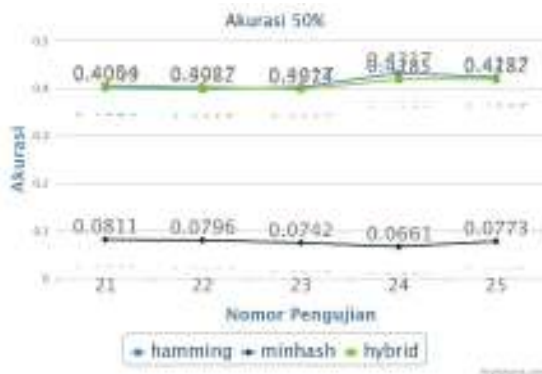
Gambar 10. Grafik Pengujian Penghilangan 50% (6-10)



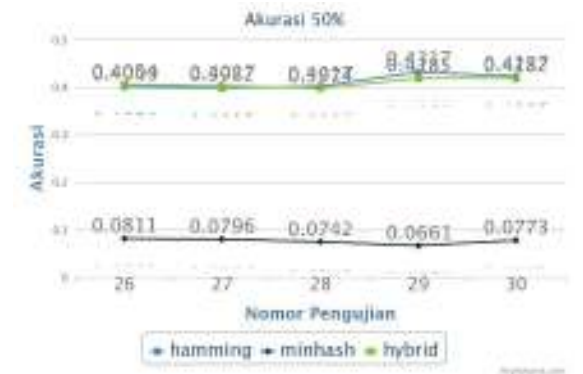
Gambar 11. Grafik Pengujian Penghilangan 50%(11-15)



Gambar 8. Grafik Pengujian Penghilangan 50% (16-20)



Gambar 8. Grafik Pengujian Penghilangan 50% (21-25)



Gambar 8. Grafik Pengujian Penghilangan 50% (26-30)

5. PENUTUP

Dari hasil penelitian dapat disimpulkan bahwa kombinasi *Minhash* dan *Binary Hamming Distance* mampu meningkatkan akurasi rekomendasi dibandingkan dengan *social-based* dan *content-based* murni, meskipun nilai akurasi rekomendasinya dibawah 0,5. Akan tetapi untuk data yang memiliki tingkat *sparse* tinggi, metode *Minhash*, *Binary Hamming Distance* dan kombinasi keduanya tidak bisa dipakai, karena akurasi maksimal yang dihasilkan masih dibawah 0,5.

Untuk penelitian berikutnya, algoritma ini cocok digunakan dengan data yang memiliki nilai *sparse* yang cukup rendah, sehingga algoritma dapat diuji dengan lebih baik lagi. Selain itu, perlu dilakukan penelitian dengan memakai algoritma yang lain, misalnya PCA atau SVD, untuk data yang memiliki nilai *sparse* cukup tinggi. Kedua algoritma ini digunakan untuk mempersiapkan data terlebih dahulu. Selain itu, bisa memakai kombinasi teknik rekomendasi yang lain seperti *knowledge-based* atau *utility-based* agar akurasi rekomendasi *hybrid* meningkat.

6. DAFTAR PUSTAKA

- [1] Robin Burke. Hybrid Recommender Systems : Survey and Experiments. *User Modelling and User-Adapted Interaction*.
- [3] Cai, J., Francis, J., & Gheysens, S. (2009). Creating a Hybrid Music Recommendation System from Content and Social-Based Algorithms.
- [3] Djamal, R. A., Mahrani, W., & Kurniati, A. P. (2010). Analisis dan Implementasi Metode Item-Based Clustering Hybrid pada Recommender System. *Proceeding Konferensi Nasional Sistem dan Informatika*, 216-222.
- [4] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2011). Recommender System : An Introduction. New York: Cambridge University Press.
- [5] Oktoria, R., Maharani, W., & Firdaus, Y. (2010). Content Based Recommender System Menggunakan Algoritma Apriori. *Proceeding Konferensi Nasional Sistem dan Informatika 2010*, 124-129.

- [6] Cohen E., Datar M., Fujiwara S., Gionis A., Indyk P., Motwani R., Ullman J.D., Yang C.(2001) Finding interesting associations without support pruning. *Knowledge and Data Engineering, IEEE Transactions on, Vol. 13 Issue 1.* 64–78.
- [7] Lai, K. Cerpa, N.(2001). Support vs Confidence in Association Rule Algorithms. *OPTIMA*.
- [8] Adomavicius, Gediminas and Tuzhilin, Alexander (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, June 2005.*
- [9] Breese, Jhon S., Heckerman, David, and Kadie, Carl (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Microsoft Research, Microsoft Corporation.*
- [10] Zhou, K., & Zha, H. (2012). Learning Binary Codes for Collaborative Filtering. *KDD*.