

## ANALISIS DAN IMPLEMENTASI SISTEM REKOMENDASI MENGGUNAKAN *MOST-FREQUENT ITEM* DAN *ASSOCIATION RULE TECHNIQUE*

### ANALYSIS AND IMPLEMENTATION RECOMMENDER SYSTEM USING *MOST-FREQUENT ITEM* AND *ASSOCIATION RULE TECHNIQUE*

Yulisna Gita Hapsari<sup>1</sup>, Agung Toto Wibowo<sup>2</sup>, ZK Abdurahman Baizal<sup>3</sup>

<sup>1,2</sup>Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

<sup>3</sup>Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

[yulisnagita@students.telkomuniversity.ac.id](mailto:yulisnagita@students.telkomuniversity.ac.id), [agungtoto@telkomuniversity.co.id](mailto:agungtoto@telkomuniversity.co.id),<sup>2</sup>

[baizal@telkomuniversity.co.id](mailto:baizal@telkomuniversity.co.id),<sup>3</sup>

---

#### Abstrak

Informasi yang sangat banyak jumlahnya mendorong kebutuhan akan adanya sistem rekomendasi. Sistem rekomendasi merupakan sebuah perangkat lunak yang dapat digunakan untuk menghasilkan rekomendasi item yang dapat digunakan oleh user. Permasalahan yang ada dalam pembangunan sistem ini diantaranya terletak pada ketepatan rekomendasi yang dihasilkan. Teknik pendekatan paling sukses untuk membangun sistem rekomendasi ini salah satunya adalah dengan menggunakan *collaborative filtering*. Dari hasil penelitian sebelumnya, *association rule* memiliki nilai performansi tertinggi dalam menentukan rekomendasi bila dibandingkan dengan algoritma *matrix factorization* dan algoritma *item-based recommendation*. Sedangkan teknik *most-frequent item* secara sederhana dapat digunakan untuk menghasilkan rekomendasi berdasarkan item yang paling sering digunakan oleh user lain (*neighbor*). Pada penelitian ini dibuat sebuah sistem rekomendasi yang membandingkan antara 2 metode yaitu *association rule* dan *most-frequent item*. Sistem rekomendasi ini diterapkan pada data film dan menghasilkan rekomendasi berupa data item yang dapat ditonton oleh *user*. Hasil dari penelitian yang telah dilakukan adalah bahwa nilai *precision* yang dihasilkan dari metode *association rule* lebih tinggi dari nilai *precision* pada *most frequent item* pada semua skenario pengujian yang dilakukan. Nilai *precision* dari *association rule* adalah 0,38 sedangkan nilai *precision most-frequent item* adalah 0,23.

**Kata kunci :** sistem rekomendasi, *collaborative filtering*, *data mining*, *association rule*, *most-frequent item*

---

#### Abstract

The huge amount of information encouraging the need of *recommender systems*. A *recommender system* is a software that can be used to produce item recommendations for user. The problems that exist in the construction of this system lies in the accuracy of recommendations. The most successful approaches to build a recommendation system is *collaborative filtering*. In this research, we compares two methods to construct *recommender system*. That methods is *association rule* and *most-frequent items*. From the previous research, *association rule* has higher performance than *matrix factorization* algorithms and *item-based recommendation* algorithm in determining recommendation. While *most-frequent items* methods can be used to generate recommendations based on most preferred items by another user. Later *recommender system* are analyzed and implemented by two methods that is *association rule method* and *most-frequent item*. The *recommender system* is applied to the movie data and produce film recommendation for user. The results of this research is the *precision* score of *Association rule* method is higher than the *precision* score of *most frequent item* method. The *precision* score of *association rule* is 0,38 and the *most-frequent item precision* score is 0,23.

**Keywords :** *recommender system*, *collaborative filtering*, *data mining*, *association rule*, *most-frequent item*

---

### 1. Pendahuluan

*Recommender system* hadir sebagai solusi bagi manusia dalam menyaring informasi [1]. *Recommender system* merupakan perangkat lunak yang digunakan untuk menghasilkan rekomendasi item yang dapat digunakan oleh user

dan disesuaikan dengan ketertarikan atau profil user [2]. Item ini dapat berupa benda, artikel, film, lagu, dan lain sebagainya. Ada beberapa pendekatan untuk membangun sebuah *recommender system*, salah satunya yang paling populer dan tersukses yaitu *collaborative filtering* [3] [4].

Penelitian mengenai *recommender system* dengan pendekatan *collaborative filtering* pernah dilakukan sebelumnya yaitu dengan membandingkan algoritma *item-based collaborative filtering*, *matrix factorization method*, dan *association rule*. Pada penelitian tersebut, *association rule* memiliki nilai performansi tertinggi dalam menentukan rekomendasi bila dibandingkan dengan dua algoritma yang lain [2]. *Association rule* menghasilkan rekomendasi dengan cara mencari pola keterkaitan antar *user* [4]. Sedangkan teknik lain, yaitu *most-frequent item* juga dapat digunakan untuk menghasilkan rekomendasi berdasarkan ketertarikan *user* lain terhadap suatu item [5].

Pada *recommender system* kualitas dari rekomendasi yang dihasilkan menjadi sebuah hal yang penting untuk diperhatikan agar *user* puas dengan hasil rekomendasi yang diberikan. Oleh karena itu perlu dicari solusi untuk permasalahan ini.

Dalam tugas akhir ini proses *collaborative filtering* dilakukan dengan membuat *cluster user* agar dapat menemukan kelompok *user* yang memiliki ketertarikan yang sama berdasarkan data rating yang ada. Sedangkan untuk proses pembentukan rekomendasi dilakukan dengan 2 metode yaitu *most-frequent item* dan *association rule*. Untuk mengetahui performansi yang paling baik diantara kedua metode tersebut maka perlu dilakukan proses analisa terhadap hasil rekomendasi untuk mengetahui nilai performansi dari setiap metode.

## 2. Dasar Teori dan Metodologi Penelitian

### 2.1 Recommender System

Sistem rekomendasi (*Recommender System*) adalah sebuah perangkat lunak untuk memberikan rekomendasi kepada pengguna mengenai produk yang dapat digunakannya. Produk ini dapat berupa barang elektronik, buku, musik, film, dan sebagainya. Rekomendasi ini dibuat berdasarkan adanya personalisasi sehingga rekomendasi yang dihasilkan mungkin berbeda-beda bagi tiap *user*. Personalisasi ini dapat dihasilkan dari informasi *user*, yang berupa *rating* atau data transaksi *user*. [1]

Tujuan dari *recommender system* adalah untuk melakukan scanning secara otomatis dari informasi yang bermacam-macam dan sangat banyak jumlahnya untuk mendapatkan rekomendasi produk yang berpotensi disukai oleh seorang *user*. Beberapa *recommender system* online yang ada diantaranya bekerja untuk memberikan rekomendasi produk berupa buku, musik, film, restoran, *web page*, dan lain sebagainya. Contohnya adalah Amazon ([www.amazon.com](http://www.amazon.com)), FireFly ([www.firefly.com](http://www.firefly.com)), dan GroupLens ([www.cs.umn.edu/Research/GroupLens](http://www.cs.umn.edu/Research/GroupLens)). [4]

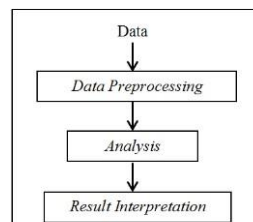
Beberapa jenis *recommender system* yang ada adalah sebagai berikut :

- *Content-based filtering*, menghasilkan rekomendasi berdasarkan item-item yang pernah digunakan oleh *user* tersebut. [1]
- *Collaborative filtering*, menghasilkan rekomendasi untuk *user target* berdasarkan *user-user* lain yang sebelumnya memiliki minat yang sama dengan *user* tersebut.
- *Demographic*, menghasilkan rekomendasi bagi *user* berdasarkan profil demografi (kependudukan) *user* [1].
- *Hybrid recommender systems*, menghasilkan rekomendasi dengan cara menggabungkan berbagai teknik pencarian rekomendasi. [1]

### 2.2 Metode Data Mining untuk Recommender System

*Data mining* bertujuan untuk melakukan proses ekstraksi informasi/pengetahuan yang masih tersembunyi dari data yang sangat banyak jumlahnya. Proses ini dilakukan untuk menemukan pola-pola tertentu agar dapat mempermudah proses penarikan informasi.

Teknik *data mining* seringkali digunakan pada *recommender system* seperti *Human Computer Interaction* (HCI) dan *Information Retrieval* (IR).



Gambar 2.1 Tahapan Utama pada Data Mining [1]

Pada gambar 2.1 terdapat 3 tahapan utama pada permasalahan *data mining*, yaitu *data preprocessing*, *data analysis*, dan *result interpretation*. [1]

### 2.3 Collaborative filtering

*Collaborative filtering* (CF) merupakan salah satu pendekatan terbaik untuk membangun sistem rekomendasi [3]. CF bisa digunakan untuk melakukan *filtering* pada data yang akan diproses pada *recommender system*. CF bekerja dengan membandingkan karakteristik antar user. Sehingga pembuatan rekomendasi didasarkan pada opini user, item-item yang sering digunakan/dibeli oleh user lain, ataupun berdasarkan rating yang diberikan dari user.

CF terbagi lagi ke dalam 3 metode, yaitu :

- *Memory-based Collaborative Filtering*

*Memory-based* memanfaatkan database *user-item* untuk membuat prediksi. Pada awalnya teknik ini akan membuat kumpulan user yang memiliki tingkat kesamaan yang tinggi dengan user target. Kumpulan user ini disebut dengan (*neighbor*), kemudian dari data user tersebut digunakan sebagai acuan pemberian rekomendasi untuk user target. Beberapa teknik yang memanfaatkan *memory-based CF* adalah *user-based Top-N recommendation algorithm* dan *item-based Top-N recommendation algorithm*. [3]

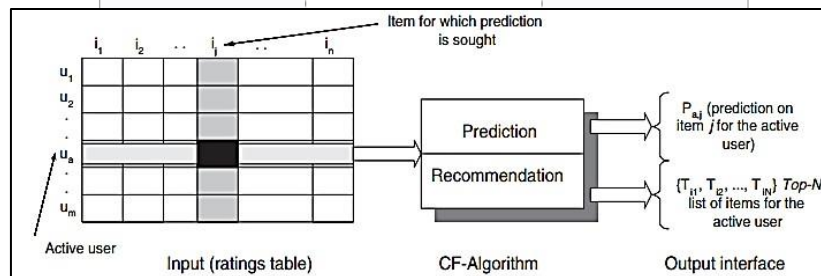
- *Model-based Collaborative Filtering*

*Model-based* menghasilkan rekomendasi dengan membuat model atau dengan mempelajari suatu algoritma tertentu pada proses training. Proses pembuatan model ini banyak jenisnya, seperti *Bayesian network*, *clustering*, dan *rule-based*. [3]

- *Hybrid Collaborative Filtering*

*Hybrid CF* dibuat dengan menggabungkan metode CF dengan metode lain, misalnya dengan *content-based* atau dengan metode *recommender system* yang lain. Tujuannya adalah untuk saling melengkapi kelebihan ataupun kekurangan yang ada pada tiap-tiap metode dan untuk meningkatkan performansi *recommender system*. [3]

Secara sederhana dapat dijelaskan, jika *user A* memiliki kemiripan dengan *user B*, maka item yang disukai oleh *user B* akan direkomendasikan kepada *user A*. *Collaborative filtering* melakukan hal tersebut dengan proses seperti pada gambar berikut. [6]



Gambar 2.2 Proses Collaborative Filtering menggunakan Matriks Rating [6]

### 2.4 Clustering

*Clustering* dilakukan dengan tujuan untuk meningkatkan efisiensi komputasi sehingga penghitungan jarak yang diperlukan tidak terlalu banyak. Pada *clustering* data akan dikelompokkan namun label-label klasifikasinya belum diketahui. *Clustering* termasuk dalam klasifikasi *unsupervised learning* yang bekerja dengan cara mengelompokkan objek-objek yang ada (contohnya *item* atau *user*). Tingkat kemiripan antar objek dapat diukur dengan *similarity measures* ataupun dengan *distance measures*. Algoritma *clustering* bertujuan untuk membuat jarak *intra-cluster* sekecil-kecilnya dan membuat jarak *inter-cluster* yang sebesar-besarnya. Salah satu contoh algoritma *clustering* adalah *k-Means*.

Dengan *k-Means* maka jumlah *cluster* yang akan dibuat dapat ditentukan, yaitu berdasarkan nilai *k*. Jika *k* = 5 maka akan dibuat sejumlah 5 *cluster*. Setiap *cluster* ditentukan kedekatannya berdasar sebuah titik tengah (*centroid*). Algoritma *k-Means* didefinisikan sebagai proses iterative dari persamaan 2.1 berikut.

$$\sum \sum \quad (2.1)$$

= jumlah *cluster*

= objek (bisa berupa item, user, dan lain-lain)

= satuan *cluster* j

= jarak antara objek dengan *centroid*

Persamaan 2.1 dihitung berulang kali hingga didapat cluster yang tetap atau tidak berubah secara signifikan. Berikut ini adalah alur dari algoritma *k-Means*.

### 2.5 Association Rule

*Association rule* adalah suatu teknik data mining untuk menemukan aturan/*rule* yang dapat memprediksi kemunculan objek pada suatu transaksi [1]. Contoh *association rule* pada proses pembelian adalah ketika terjadi keterkaitan antar satu barang dengan barang lain yang dibeli secara bersamaan pada satu transaksi. Dengan mengetahui teknik *association rule*, maka akan lebih mudah menganalisis kebutuhan user. Analisis asosiasi ini juga sering disebut dengan istilah *market basket analysis*.

*Itemset* merupakan kumpulan *item* dengan jumlah satu atau lebih. Contohnya (Susu, Roti, Diaper). Sedangkan *k-itemset* merupakan *itemset* yang mengandung sejumlah *k item*. Persentase kemunculan *itemset* dalam sebuah database dinyatakan dengan nilai *support* (nilai penunjang). Contohnya (Susu, Roti, Diaper) = 60%. *Frequent itemset* dinyatakan sebagai *itemset* yang memiliki nilai *support* lebih besar dari atau sama dengan nilai *minsup* threshold. [1]

Contoh dari *association rule* :

**{roti, mentega} → {susu} (support = 40%, confidence = 50%)**

Yang artinya : "50% dari transaksi di database yang memuat *item* roti dan mentega juga memuat *item* susu. Sedangkan 40% dari seluruh transaksi yang ada di database memuat ketiga *item* itu."

Dapat juga diartikan : "Seorang konsumen yang membeli roti dan mentega punya kemungkinan 50% untuk juga membeli susu. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini."

Analisis asosiasi didefinisikan sebagai suatu proses untuk menemukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support* (minimum *support*) dan syarat minimum untuk *confidence* (minimum *confidence*).

Metodologi dasar analisis asosiasi terbagi menjadi dua tahapan :

- a. Analisa pola frekuensi tinggi (*frequent itemset generation*)

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam database. *Support* berguna untuk menentukan seberapa sering sebuah *rule* yang memiliki kemunculan *item* ( atau ) dapat dipenuhi pada database transaksi ( ).

Bentuk matematis dari *support* dapat dijabarkan sebagai berikut :

$$\text{_____} \quad (2.2)$$

- b. Pembentukan aturan asosiatif (*Rule Generation*)

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum *confidence*. *Confidence* digunakan untuk menentukan seberapa sering *item* muncul pada transaksi yang mengandung *item* .

Bentuk matematis dari *support* dapat dijabarkan sebagai berikut :

$$\text{_____} \quad (2.3)$$

### 2.6 Most-Frequent Item

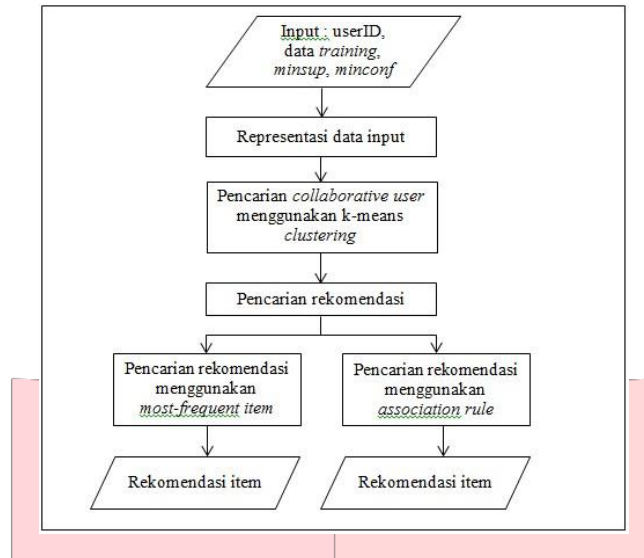
MFI (*Most Frequent Item*) *recommendation* merupakan salah satu teknik untuk mendapatkan rekomendasi bagi target user. MFI mengacu pada ketetanggaan (*neighborhood-based*) yang telah terbentuk, baik ketetanggaan antar *user* (*user-based*) ataupun antar *item* (*item-based*). MFI memilih sejumlah rekomendasi yang paling sering muncul atau yang paling disukai. Teknik ini dapat dilakukan dengan menggunakan *Top-N recommendation*.

Pada teknik *user-based Top-N recommendation* pembentukan rekomendasi dihasilkan dengan melakukan tahapan-tahapan sebagai berikut [3]:

1. Identifikasi sejumlah *k* user yang paling mirip dengan target user (*nearest neighbor*) dengan menggunakan *Pearson correlation* ataupun *Cosine similarity*.
2. Setelah *k* user ditemukan, buat matriks *R* yang menampung *user-item* untuk mendapatkan kumpulan *item C* yang dibeli pada kelompok tersebut serta frekuensi pembelian *item* tersebut.
3. Dari kumpulan *item C* dapat dilihat *item-item* yang paling sering dibeli oleh *user* dalam kelompok tersebut.

Pilih *N* *item* yang paling sering dibeli oleh *user* lain dan belum pernah dibeli oleh *user* target (*N* adalah jumlah yang sudah ditentukan sebelumnya)

## 2.7 Metodologi Penelitian



Gambar 2.4 Gambaran Umum Perancangan Sistem

Inputan awal yang diperlukan dalam sistem ini adalah userID dari *user target*, *minsup*, *minconf*, dan data *training*. Pada awal proses, data *training* yang telah diinputkan perlu direpresentasikan dahulu ke dalam bentuk matriks *rating*. Kemudian pencarian *collaborative user* dilakukan dengan teknik *clustering* menggunakan algoritma *k-means*. *Collaborative user* merupakan kelompok *user* yang memiliki minat yang sama dengan *user target*. Pada proses pencarian rekomendasi, ada 2 metode yang digunakan dalam pencarian rekomendasi ini, yaitu *most-frequent item* dan *association rule*. Output dari sistem rekomendasi ini adalah rekomendasi *item* untuk *user target* beserta nilai performansi dari hasil rekomendasi tersebut.

## 3. Pembahasan

Pada tugas akhir ini proses pembuatan sistem rekomendasi diawali dengan perancangan sistem. Pada proses implementasi, metode-metode yang telah disebutkan sebelumnya diterapkan pada sistem rekomendasi ini. Setelah sistem rekomendasi mampu menghasilkan rekomendasi item, selanjutnya dilakukan beberapa proses pengujian untuk menganalisa performansi yang dimiliki oleh sistem rekomendasi.

Data yang digunakan dalam penelitian ini adalah data *rating* film MovieLens 100k yang didownload dari <http://www.grouplens.org/>. Data ini terdiri dari 100.000 *rating* dengan rentang 1-5 dari 943 *user* dengan 1682 judul film.

Pengujian dilakukan terhadap kualitas rekomendasi yang dihasilkan berdasarkan nilai performansi *precision*, karena yang akan diukur adalah ketepatan dari rekomendasi yang diberikan oleh *recommender system*. *Precision* merupakan perbandingan dari item yang direkomendasikan dengan tepat terhadap total item yang direkomendasikan. Beberapa scenario pengujian yang akan dilakukan adalah sebagai berikut.

### 3.1 Pengujian perbandingan data *training* dan data *testing* terhadap nilai *precision*

Pada skenario pengujian pertama dilakukan pengujian mengenai pengaruh perbandingan ukuran data *training* dan data *testing* terhadap nilai performansi. Proses pengujian ini dilakukan terhadap seluruh *user target* yang terdapat pada data *testing*. Ukuran data *training* yang diujikan adalah 60%, 70 %, 80%, dan 90%. Pada pembentukan *collaborative user*, digunakan *Pearson correlation* untuk menghitung jarak antar *user*. Pada metode *association rule*, nilai minimum *confidence* pada pengujian ini di-set 0.1 dan nilai minimum *support* di-set 0.3 agar mendapatkan jumlah *rule* yang optimum. Sedangkan pada MFI digunakan top-5 *recommendation* atau rekomendasi sebanyak 5 item yang paling diminati oleh *user-user* pada tiap *cluster*.

Hasil dari pengujian ini adalah sebagai berikut.

Tabel 3. 1 Nilai *precision* pada pengujian perbandingan ukuran data

Data training	Data testing	Precision Most-frequent item	Precision Association Rule
60 %	40%	0.23	0.24
70 %	30%	0.19	0.38
80 %	20%	0.11	0.26
90 %	10%	0.06	0.12

Dari pengujian tersebut terlihat bahwa *association rule* memiliki nilai performansi yang lebih baik dibanding metode *most-frequent item*.

**3.2 Pengujian normalisasi data *rating* terhadap nilai *precision***

Normalisasi data *rating* akan diuji dengan 2 proses yang berbeda yaitu normalisasi *by-value* (nilai) dan normalisasi *by-mean* (rata-rata). Proses pengujian ini dilakukan terhadap seluruh *user target* yang terdapat pada data *testing*. Pada pengujian pertama, seluruh *rating* dengan nilai lebih dari atau sama dengan 3 ( $\geq 3$ ) akan dikonversi menjadi 1 yang menandakan *user* suka terhadap film tertentu, sedangkan nilai lainnya dikonversi menjadi 0 yang menandakan *user* tidak menyukai film tersebut. Pada pengujian kedua, proses normalisasi *rating* dilakukan per *user* dengan melihat rata-rata dari tiap *user* dalam memberikan *rating*. Hal ini perlu diujikan karena *rating* yang diberikan oleh tiap *user* adalah bersifat relatif yaitu standar nilai “bagus” untuk tiap film bagi tiap *user* mungkin berbeda. Jadi *rating* dengan nilai lebih dari atau sama dengan rata-rata *rating* yang diberikan oleh seorang *user* akan dikonversi menjadi 1 yang menandakan *user* menyukai film tertentu, berlaku sebaliknya nilai lainnya dikonversi menjadi 0.

Tabel 3. 2 Nilai *precision* pada pengujian normalisasi

Metode	Precision Most-frequent item	Precision Association Rule
Konversi <i>by-value</i>	0.058	0.125
Konversi <i>by-mean</i>	0.059	0.181

Dari pengujian tersebut terlihat bahwa *association rule* memiliki nilai performansi yang lebih baik dibanding metode *most-frequent item* dan metode konversi *by-mean* menghasilkan nilai performansi yang lebih baik dibanding metode konversi *by-value*.

**3.3 Pengujian penghitungan *collaborative user* terhadap nilai *precision***

Penghitungan kedekatan ataupun kesamaan antar *user* dapat diukur dengan berbagai rumus jarak atau rumus kesamaan yang ada. Pada penelitian ini akan diuji 2 jenis pengukuran kedekatan *user* yaitu dengan menggunakan *Pearson correlation* dan *Cosine similarity*. Pada pengujian pertama, kedekatan antar *user* diukur dengan *Pearson correlation*, sedangkan pada pengujian kedua menggunakan *Cosine similarity* untuk mengukur kesamaan *user* hingga dihasilkan *collaborative user*. Ukuran data yang digunakan adalah data *training* sebesar 90% dan data *testing* sebesar 10 %. Pada metode *association rule* diterapkan minimum *support* sebesar 0.4 dan minimum *confidence* 0.1 agar dihasilkan jumlah *rule* yang optimum. Sedangkan pada metode *most-frequent item* digunakan top-5 *recommendation*. Setelah menghasilkan rekomendasi, kemudian dari tiap metode dihitung nilai *precision*-nya.

Dari keseluruhan pengujian yang telah dilakukan pada metode *association rule* dan *most-frequent item* dapat disimpulkan bahwa penghitungan jarak menggunakan *Pearson correlation* akan menghasilkan nilai *precision* yang lebih baik. Berarti pada data yang diujikan ini lebih cocok menggunakan *Pearson correlation*. Hal ini juga dikarenakan *Pearson correlation* merupakan metode yang cocok untuk menghitung jarak kedekatan antar *user* berdasarkan *rating* yang memiliki faktor relatifitas. Nilai *precision* yang dihasilkan pada metode *association rule* adalah 0,17 sedangkan pada metode *most-frequent item* menghasilkan nilai 0,06.

**3.4 Pengujian minimum *support* pada metode *Association rule* terhadap nilai *precision***

Pada skenario pengujian ini dilakukan dengan nilai minimum *support* yang berubah-ubah dan nilai minimum *confidence* yang tetap. Nilai minimum *support* yang diuji adalah antara 0.20 hingga 0.35 dengan interval 0,01. Sedangkan nilai minimum *confidence* pada pengujian ini di-*set* tetap sebesar 0.1 agar mendapatkan jumlah *rule* yang optimum. Tujuan yang ingin dicapai pada pengujian ini adalah untuk menganalisa bagaimana pengaruh perubahan minimum *support* pada metode *association rule* terhadap nilai *precision*.

Pada pengujian ini nilai *precision* terbaik didapat pada nilai minimum *support* 0,29. Sehingga dapat disimpulkan bahwa 0,29 merupakan minimum *support* yang paling optimum dalam menghasilkan nilai *precision* terbaik.

### 3.5 Pengujian minimum *confidence* pada metode *Association rule* terhadap nilai *precision*

Pada skenario pengujian ini dilakukan dengan nilai minimum *confidence* yang berubah-ubah dan nilai minimum *support* yang tetap. Nilai minimum *confidence* yang diuji adalah antara 0.10 hingga 0.60 dengan interval 0,01. Sedangkan nilai minimum *support* pada pengujian ini di-*set* tetap sebesar 0.3 agar mendapatkan jumlah *rule* yang optimum. Minimum *support* di-*set* tetap sebesar 0.3 karena nilai ini merupakan nilai terkecil yang bisa menghasilkan *rule* optimum. Jika dilakukan pengujian dengan minimum *support* kurang dari 0.3 maka ditemukan kondisi sistem yang tereksekusi dengan kurang baik yaitu proses komputasi yang memakan waktu cukup lama (contohnya untuk minimum *support* 0.2 menghabiskan waktu selama 30 menit untuk menghasilkan *rule*). Sedangkan untuk nilai minimum *support* lebih dari 0.3 maka akan ada *cluster* yang tidak dapat menghasilkan *rule* karena batas minimum *support* yang cukup besar. Tujuan yang ingin dicapai pada pengujian ini adalah untuk menganalisa bagaimana pengaruh perubahan minimum *confidence* pada metode *association rule* terhadap nilai *precision*.

Pada pengujian ini nilai *precision* terbaik didapat pada nilai minimum *confidence* 0,52. Sehingga dapat disimpulkan bahwa 0,52 merupakan minimum *confidence* yang paling optimum dalam menghasilkan nilai *precision* terbaik.

### 3.6 Pengujian *Top-N* pada metode *most-frequent item* terhadap nilai *precision*

Pada skenario pengujian ini dilakukan pengujian terhadap pengaruh nilai *Top-N* rekomendasi terhadap nilai performansi. *Top-N* rekomendasi merupakan sejumlah *N* rekomendasi yang paling diminati oleh user. Masing-masing kelompok *collaborative user* (*cluster*) memiliki *Top-N* rekomendasi masing-masing. Nilai *N* yang diuji adalah antara 3 hingga 15 dengan interval 1. Tujuan dari pengujian ini adalah untuk menganalisa bagaimana pengaruh perubahan *N* pada metode *most-frequent item* terhadap nilai *precision*.

Pada pengujian ini nilai *precision* terbaik didapat pada  $N=4$ . Sehingga dapat disimpulkan bahwa  $N=4$  merupakan *top-N* yang paling optimum dalam menghasilkan nilai *precision* terbaik.

## 4. Kesimpulan

1. Metode *association rule* dan *most-frequent item* dapat diterapkan pada sistem rekomendasi *collaborative filtering* berbasis *user* karena metode ini mampu menghasilkan rekomendasi item bagi *user target*.
2. Performansi sistem rekomendasi menggunakan *association rule* dipengaruhi oleh nilai minimum *support* dan nilai minimum *confidence*. Minimum *support* paling optimum untuk data *training* 90% adalah 0,29. Minimum *confidence* paling optimum untuk data *training* 90% adalah 0,52. Sedangkan ukuran data paling optimum adalah data *training* sebesar 70% dan data *testing* 30% dengan nilai *precision* sebesar 0,38.
3. Performansi sistem rekomendasi menggunakan *most-frequent item* dipengaruhi oleh rekomendasi *Top-N* serta ukuran data *training* dan data *testing*. Ukuran *Top-N* paling optimum pada ukuran data *training* 90% adalah  $N=4$ . Sedangkan ukuran data paling optimum adalah data *training* sebesar 60% dan data *testing* 40% dengan nilai *precision* sebesar 0,23.
4. Untuk penghitungan jarak menggunakan *Pearson correlation* menghasilkan nilai *precision* yang lebih baik dibandingkan dengan *Cosine Similarity*.

## Daftar Pustaka

- [1] Lior Rokach Francesco Ricci, *Recommender System Handbook*. London: Springer, 2011.
- [2] Savaneary Sean, Julien Delporte Bruno Pradel, "A Case Study in a Recommender System Based on Purchase Data," vol. 1, no. Recommender System, 2011.
- [3] Taghi M. Khoshgoftaar Xiaoyuan Su, "A Survey of Collaborative Filtering Techniques," vol. I, 2009.
- [4] Weiyang Lin, "Association Rule Mining for Collaborative Recommender System," 2000.
- [5] Badrul Sarwar, "Analysis of recommendation Algorithms for E-Commerce," vol. 1, no. Recommendation Algorithm, 2000.
- [6] Ron Zacharski. (2015, Mei) Guide to Data Mining. [Online]. <http://www.guidetodatamining.com/>

- [7] George Karypis Badrul Sarwar, "Item-based Collaborative Filtering Recommendation Algorithms," vol. I, 2001.
- [8] Cambridge University Press. (2015, September) K-means. [Online]. <http://nlp.stanford.edu/IR-book/html/htmledition/k-means-1.html>
- [9] Rakesh Agrawal, "Fast Algorithm for Mining Association Rules," 1994.
- [10] Juntae Kim Choonho Kim, "A Recommendation Algorithm Using Multi-Level Association Rule," , vol. I, Seoul, 2003.
- [11] Dr. Novrina. (2009) Universitas Gunadarma Staffsite. [Online]. <http://novrina.staff.gunadarma.ac.id/>
- [12] Sheng-Long Lv Zhi-Hong Deng, "Fast mining frequent itemsets using Nodesets," 2014.
- [13] Lilis Puspitawati. (2015, Agustus) [elib.unikom.ac.id](http://elib.unikom.ac.id). [Online]. <http://elib.unikom.ac.id/files/disk1/374/jbptunikompp-gdl-lilispuspi-18691-10-pertemua-7.pdf>

