

## *Hybrid Recommendation System Memanfaatkan Penggalian Frequent Itemset dan Perbandingan Keyword*

**Wayan Gede Suka Parwita\*<sup>1</sup>, Edi Winarko<sup>2</sup>**

<sup>1</sup>Prodi S2/S3 Ilmu Komputer, FMIPA UGM, Yogyakarta

<sup>2</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: \*<sup>1</sup>gede.suka@gmail.com, <sup>2</sup>ewinarko@ugm.ac.id

### **Abstrak**

*Recommendation system sering dibangun dengan memanfaatkan data peringkat item dan data identitas pengguna. Data peringkat item merupakan data yang langka pada sistem yang baru dibangun. Sedangkan, pemberian data identitas pada recommendation system dapat menimbulkan kekhawatiran penyalahgunaan data identitas.*

*Hybrid recommendation system memanfaatkan algoritma penggalian frequent itemset dan perbandingan keyword dapat memberikan daftar rekomendasi tanpa menggunakan data identitas pengguna dan data peringkat item. Penggalian frequent itemset dilakukan menggunakan algoritma FP-Growth. Sedangkan perbandingan keyword dilakukan dengan menghitung similaritas antara dokumen dengan pendekatan cosine similarity.*

*Hybrid recommendation system memanfaatkan kombinasi penggalian frequent itemset dan perbandingan keyword dapat menghasilkan rekomendasi tanpa menggunakan identitas pengguna dan data peringkat dengan penggunaan ambang batas berupa minimum similarity, minimum support, dan jumlah rekomendasi. Nilai pengujian yaitu precision, recall, F-measure, dan MAP dipengaruhi oleh besarnya nilai ambang batas yang ditetapkan.*

**Kata kunci**— *Hybrid recommendation system, frequent itemset, cosine similarity.*

### **Abstract**

*Recommendation system was commonly built by manipulating item is ranking data and user is identity data. Item ranking data were rarely available on newly constructed system. Whereas, giving identity data to the recommendation system causes concerns about identity data misuse.*

*Hybrid recommendation system used frequent itemset mining algorithm and keyword comparison, it can provide recommendations without identity data and item ranking data. Frequent itemset mining was done using FP-Growth algorithm and keyword comparison with calculating document similarity value using cosine similarity approach.*

*Hybrid recommendation system with a combination of frequent itemset mining and keywords comparison can give recommendations without using user identity and rating data. Hybrid recommendation system using 3 thresholds ie minimum similarity, minimum support, and number of recommendations. With the testing data used, precision, recall, F-measure, and MAP testing value are influenced by the threshold value.*

**Keywords**— *Hybrid recommendation system, frequent itemset, cosine similarity.*

## 1. PENDAHULUAN

Peningkatan jumlah dokumen ilmiah yang ada menimbulkan kebutuhan akan suatu sistem yang dapat memberikan rekomendasi dokumen ilmiah yang baik. *Recommendation system* merupakan model aplikasi yang dibangun dari hasil pengamatan terhadap keadaan dan keinginan pengguna [1]. Berbagai metode pendekatan telah diterapkan dan dikembangkan dalam implementasi *recommendation system*. Selama dekade terakhir ini, *recommendation system* telah banyak diterapkan dengan berbagai pendekatan baru, baik oleh dunia industri maupun akademis. Pada dunia industri, *recommendation system* sangat diperlukan terutama pada website e-commerce. Ini ditunjukkan dengan penggunaan *recommendation system* pada sebagian besar *e-commerce web sites* yang dimiliki oleh industri. Selain membantu pengguna dalam mencari *item* yang diinginkan, *recommendation system* juga dapat meningkatkan penjualan, ketertarikan maupun loyalitas pengguna terhadap suatu *item* dan juga perusahaan [2].

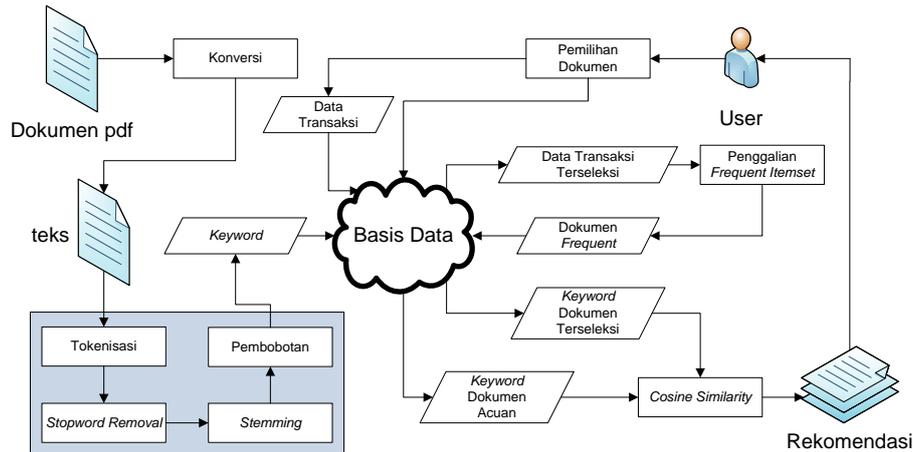
Metode yang banyak digunakan adalah *collaborative filtering* dan *content-based filtering*. Umpan balik yang digunakan pada metode *collaborative filtering* mengakibatkan sistem dapat memprediksi keinginan pengguna. Sedangkan metode *content-based filtering* menggunakan konten dari *item* sehingga dapat memberikan rekomendasi tanpa adanya umpan balik dari pengguna. Akan tetapi, metode *collaborative filtering* sangat tergantung dengan umpan balik yang diberikan oleh pengguna. Umpan balik berupa peringkat, data transaksi, maupun data identitas yang diberikan oleh pengguna. Tanpa umpan balik, metode *collaborative filtering* tidak dapat melakukan rekomendasi. Penggunaan data identitas pada metode *user-based collaborative filtering* juga mengakibatkan pengguna harus terdaftar pada sistem untuk membedakan klasifikasi data yang dikumpulkan dari pengguna. Akan tetapi, pengguna terkadang enggan untuk mendaftar pada sistem karena kekhawatiran terhadap penyalahgunaan data identitas. Di sisi lain, metode *content-based* memiliki ketergantungan terhadap perbandingan konten maupun atribut antara *item*.

Fokus utama dari penelitian ini adalah penerapan kombinasi penggalian *frequent itemset* dan perbandingan *keyword* dalam menghasilkan rekomendasi dokumen ilmiah. Sebelumnya penelitian tentang *hybrid recommendation system* [3, 4, 5, 6, 7, 8, 9] dan *document recommendation system* [9, 10, 11] telah banyak dilakukan. Akan tetapi, pada beberapa bagian penelitian tersebut berbasis *user-based* dan *item-based collaborative filtering*.

## 2. METODE PENELITIAN

Pembangunan *recommendation system* terbagi menjadi beberapa tahapan proses. Tahap-tahap proses tersebut ditunjukkan oleh Gambar 1. Dokumen pdf (*Portable Document Format*) dikonversi terlebih dahulu ke dalam format txt (*text*). Dokumen teks diproses pada *keyword extraction system* untuk memperoleh *keyword* dokumen. *Keyword extraction system* melakukan ekstraksi *keyword* untuk dokumen yang dimasukkan pada sistem sehingga setiap dokumen akan memiliki daftar *keyword* hasil ekstraksi.

Data transaksi dibentuk saat proses pemilihan dokumen yang dilakukan oleh pengguna. Proses pemilihan adalah saat pengguna melakukan pemilihan dokumen yang ada pada daftar dokumen hasil pencarian atau hasil rekomendasi. Setiap sesi merupakan satu data transaksi. Dalam satu data transaksi, terdapat deret *item* yang merupakan id dari dokumen. Deret *item* ini merupakan dokumen-dokumen yang dipilih pada sesi penggunaan sistem. Dengan demikian, walaupun pengguna sama tetapi penggunaan sistem dilakukan pada sesi yang berbeda, maka data pemilihan dokumen tersebut akan dipisahkan. Hal ini dilakukan untuk mengantisipasi proses pemilihan dokumen dengan konten yang berbeda walaupun penggunaannya sama. Penggunaan asumsi tersebut mengakibatkan data identitas pengguna dapat diabaikan. Selain menjadi data transaksi, dokumen pilihan pengguna juga menjadi dokumen acuan untuk menentukan daftar data transaksi terseleksi yang digunakan pada proses penggalian *frequent itemset*.



Gambar 1 Arsitektur *recommendation system*

### 2.1 Keyword Extraction System

Dalam ekstraksi *keyword* dengan pendekatan tata bahasa berbasis struktur sintaksis, ada beberapa tahap yang dilakukan yaitu tokenisasi, *stopword removal*, *stemming*, dan pembobotan kata [12]. Pada tokenisasi dilakukan proses pemecahan deret kata dan penghilangan karakter selain huruf dan angka. Tokenisasi yang digunakan adalah tokenisasi dengan pemisah kata berupa karakter spasi. Daftar kata yang dihasilkan tokenisasi akan diseleksi pada *stopword removal*. *Stopword removal* merupakan proses untuk menghilangkan kata yang dianggap tidak penting. Daftar kata yang tidak mengandung *stopword* lalu diproses pada *stemming*. Proses *stemming* merupakan proses untuk mengubah kata turunan menjadi kata dasar. Algoritma *stemming* yang digunakan adalah algoritma yang dikembangkan di [13]. Daftar kata dasar yang didapatkan pada *stemming* akan diberikan bobot dengan skema *term frequency*. Skema *term frequency* memberikan bobot dengan menghitung jumlah kemunculan kata pada dokumen. Kata dasar beserta bobot yang didapatkan disimpan pada basis data.

### 2.2 Penggalan Frequent Itemset

Penggalan *frequent itemset* menghasilkan daftar *frequent dokumen*. Algoritma penggalan *frequent itemset* yang digunakan adalah algoritma FP-Growth. Dokumen pilihan juga menjadi dokumen acuan yang akan dibandingkan dengan dokumen terseleksi. Dokumen terseleksi untuk perbandingan merupakan *frequent dokument* dan atau dokumen hasil pencarian. FP-Growth berbasis pada *tree* disebut dengan FP (*Frequent Pattern*)-Tree [14]. FP-Tree dapat menghemat penggunaan memori untuk penyimpanan data transaksi. Ide dasar dari FP-Growth dapat digambarkan sebagai skema eliminasi secara rekursif. Dalam langkah *preprocessing*, dilakukan penghapusan semua *item* yang kemunculannya kurang dari *minimum support* yang diberikan. Kemudian dipilih semua transaksi yang mengandung *frequent item* lalu dibentuk FP-Tree berdasarkan data transaksi tersebut. Dalam penggalan *frequent itemset*, dibentuk *conditional* FP-Tree yang berakhir pada salah satu *frequent item*. Pembentukan ini dilakukan secara rekursif dengan mengeliminasi satu persatu *frequent item* akhir yang terdapat pada *tree* tersebut. Penentuan *frequent itemset* dilakukan bersamaan saat pengeliminasian dengan melihat *support* dari *frequent item* tersebut [15].

### 2.3 Perbandingan Keyword

Keyword dokumen acuan dan keyword dokumen terseleksi dibandingkan dengan menggunakan *cosine similarity*. Dokumen hasil pencarian akan digunakan apabila penggalan *frequent itemset* tidak menghasilkan dokumen *frequent* atau dokumen *frequent* yang ditemukan tidak dapat memenuhi kebutuhan jumlah rekomendasi. Proses perbandingan dokumen menggunakan *cosine similarity* menghasilkan nilai similaritas antara dokumen acuan dan dokumen pembanding. Dokumen pembanding akan direkomendasikan kepada pengguna apabila

nilai similaritas memenuhi ambang batas. Hasil rekomendasi berupa daftar dokumen yang memenuhi ambang batas yang ditentukan oleh pengguna.

### 3. HASIL DAN PEMBAHASAN

Pengujian *hybrid recommendation system* memanfaatkan penggalian *frequent itemset* dan perbandingan *keyword* memerlukan 3 set data yang berupa 100 dokumen ilmiah berbahasa Indonesia, *stopword* bahasa Indonesia, dan data transaksi untuk melakukan penggalian *frequent itemset*. Dokumen untuk set data merupakan 95 dokumen ilmiah pada bidang ilmu komputer Prosiding Semantik 2012 dan 5 dokumen ilmiah bidang ilmu komputer Prosiding Senaputro 2012. Untuk rincian dokumen yang digunakan dapat dilihat pada Lampiran 1. Pemilihan dokumen dilakukan agar ekstraksi *keyword* memiliki *output* dengan kriteria dan bentuk yang sama. Pengujian dilakukan menggunakan *minimum similarity* 20% hingga 50% dan *minimum support* 5% hingga 20% dengan peningkatan setiap 5%. Jumlah rekomendasi maksimal akan digunakan untuk pengujian *precision*, *recall*, F-measure, dan *mean average precision*. Perhitungan nilai *precision* juga menggunakan jumlah rekomendasi terbatas. Sistem akan diuji dalam kasus biasa dan juga kasus *coldstart*.

#### 3.1 Precision

*Precision* pada *recommendation system* dihitung dengan persamaan (1) [16].

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ item\ recommended|}{|all\ recommendation|} \quad (1)$$

*True positive* (tp) pada *information retrieval* merupakan *item* relevan yang dihasilkan oleh sistem. Sedangkan *false positive* (fp) merupakan semua *item* yang dihasilkan oleh sistem. Tabel 1 menunjukkan adanya peningkatan nilai *precision* saat nilai *minimum support* dan *minimum similarity* ditingkatkan. Peningkatan ini terjadi karena berkurangnya jumlah rekomendasi yang dihasilkan saat nilai ambang batas tersebut ditingkatkan. Nilai *precision* tertinggi dicapai pada saat *minimum support* 20% untuk *minimum similarity* yang berbeda-beda.

Tabel 1 Nilai *precision* jumlah rekomendasi maksimal

		<i>Minimum support</i> (%)				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity</i> (%)	20	0.148	0.173	0.208	0.282	0.28
	25	0.152	0.177	0.213	0.294	0.293
	30	0.17	0.181	0.218	0.306	0.306
	35	0.247	0.25	0.27	0.333	0.326
	40	0.328	0.33	0.35	0.405	0.39
	45	0.439	0.439	0.444	0.467	0.458
	50	0.716	0.716	0.727	0.729	0.729

Pada kasus *coldstart*, *minimum support* tidak mempengaruhi peningkatan atau penurunan nilai *precision*. Ini disebabkan karena penggalian *frequent itemset* tidak dapat digunakan saat terjadinya kasus *coldstart*. Perbedaan nilai *precision* hanya disebabkan oleh peningkatan *minimum similarity*. Nilai *precision* pada kasus *coldstart* dengan *minimum similarity* 50% sama dengan nilai *precision* kasus biasa dengan *minimum support* 20% dan *minimum similarity* 50%. Untuk *minimum similarity* kurang dari 50%, nilai *precision* pada kasus biasa relatif lebih tinggi dari kasus *coldstart* saat *minimum support* kasus biasa sebesar

20%. Hal ini disebabkan oleh adanya peningkatan jumlah *item hit* yang direkomendasikan karena penggalian *frequent itemset*.

Hasil berbeda terlihat pada pengujian dengan jumlah rekomendasi terbatas. Dengan *minimum support* 5% dan 20%, nilai *precision* kasus biasa relatif lebih tinggi untuk semua *minimum similarity* yang digunakan. Hal ini disebabkan karena adanya perbedaan urutan rekomendasi yang didapatkan antara penggalian *frequent itemset* dan perbandingan dokumen biasa. Selain itu, penggunaan *frequent itemset* juga mampu menemukan *item hit* lebih banyak saat dikenakan *minimum similarity* rendah. Tabel 2 menunjukkan nilai *precision* jumlah rekomendasi terbatas.

Tabel 2 Nilai *precision* jumlah rekomendasi terbatas

		<i>Minimum support (%)</i>				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity (%)</i>	20	0.267	0.178	0.244	0.422	0.178
	25	0.267	0.178	0.244	0.422	0.178
	30	0.267	0.178	0.244	0.422	0.2
	35	0.422	0.222	0.244	0.444	0.222
	40	0.422	0.311	0.333	0.467	0.356
	45	0.444	0.333	0.333	0.467	0.422
	50	0.739	0.694	0.694	0.75	0.706

Seperti yang terjadi pada jumlah rekomendasi maksimal, nilai *precision* tertinggi pada jumlah rekomendasi terbatas dicapai pada saat *minimum support* 20%. Nilai *precision* kasus biasa pada jumlah rekomendasi terbatas kadang kala lebih tinggi dibandingkan pada rekomendasi maksimal. Akan tetapi, nilai *precision* kasus *coldstart* pada rekomendasi terbatas selalu lebih rendah dibandingkan nilai *precision* kasus *coldstart* dengan jumlah rekomendasi maksimal.

### 3.2 Recall

*Recall* digunakan sebagai ukuran dokumen yang relevan yang dihasilkan oleh sistem. *Recall* untuk evaluasi *recommendation system* dihitung dengan persamaan (2) [16].

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ item\ recommended|}{|all\ good\ item|} \quad (2)$$

*False negative* (fn) merupakan semua *item* relevan yang tidak dihasilkan oleh sistem. Nilai *recall* untuk *minimum similarity* yang meningkat cenderung menurun pada kasus biasa dan kasus *coldstart* jumlah rekomendasi maksimal. Hal yang sama terjadi saat *minimum support* ditingkatkan. Penurunan nilai *recall* terjadi karena berkurangnya *item retrieve* yang disertai dengan berkurangnya *item hit* yang dihasilkan oleh sistem. Dari Tabel 3 dapat dilihat bahwa nilai *recall* kasus *coldstart* selalu lebih rendah dari nilai *recall* kasus biasa untuk penggunaan *minimum similarity* yang berbeda-beda. Ini menunjukkan bahwa penggalian *frequent itemset* dapat menemukan *item hit* yang tidak dapat ditemukan pada saat sistem menggunakan perbandingan *keyword* saja.

Tabel 3 Nilai *recall* jumlah rekomendasi maksimal

		<i>Minimum support</i> (%)				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity</i> (%)	20	0.944	0.944	0.875	0.698	0.645
	25	0.944	0.944	0.875	0.698	0.645
	30	0.944	0.944	0.875	0.698	0.645
	35	0.934	0.934	0.866	0.688	0.636
	40	0.809	0.809	0.788	0.668	0.615
	45	0.635	0.635	0.615	0.555	0.535
	50	0.495	0.495	0.495	0.458	0.458

Hal sebaliknya terjadi pada nilai *recall* jumlah rekomendasi terbatas. Peningkatan *minimum similarity* dan *minimum support* cenderung meningkatkan nilai *recall*. Akan tetapi, nilai *recall* pada *minimum support* 5% cenderung lebih tinggi dibandingkan *minimum support* 10% dan 15%. Hal ini disebabkan oleh perbedaan hasil *frequent itemset* yang didapatkan. Di sisi lain, seperti halnya pada jumlah rekomendasi maksimal, nilai *recall* kasus biasa cenderung lebih tinggi dibandingkan dengan kasus *coldstart*. Tabel 4 menunjukkan nilai *recall* jumlah rekomendasi terbatas. Berbeda dengan nilai *recall* pada jumlah rekomendasi maksimal, nilai *recall* tertinggi pada jumlah rekomendasi terbatas didapatkan pada saat nilai *minimum similarity* 50%. Hal tersebut berlaku untuk kasus biasa maupun kasus *coldstart*.

Tabel 4 Nilai *recall* jumlah rekomendasi terbatas

		<i>Minimum support</i> (%)				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity</i> (%)	20	0.137	0.101	0.129	0.209	0.101
	25	0.137	0.101	0.129	0.209	0.101
	30	0.137	0.101	0.129	0.209	0.11
	35	0.208	0.12	0.129	0.218	0.119
	40	0.209	0.16	0.17	0.246	0.183
	45	0.22	0.172	0.171	0.245	0.255
	50	0.429	0.428	0.428	0.42	0.4

### 3.3 F-Measure

F-measure merupakan nilai tunggal hasil kombinasi antara nilai *precision* dan nilai *recall*. F-measure dapat digunakan untuk mengukur kinerja dari *recommendation system* ataupun *information retrieval system*. Karena merupakan rata-rata harmonis dari *precision* dan *recall*, F-measure dapat memberikan penilaian kinerja yang lebih seimbang. Persamaan (3) merupakan persamaan untuk menghitung F-measure [16].

$$F - Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3)$$

Perhitungan F-measure masing-masing dokumen uji pada kasus biasa ditunjukkan oleh Tabel 5. Hasil perhitungan menunjukkan bahwa untuk *minimum similarity* dan *minimum support* yang semakin meningkat, nilai F-measure yang didapatkan juga cenderung meningkat. Peningkatan ini disebabkan oleh meningkatnya nilai *precision* pada saat dilakukan peningkatan *minimum similarity* dan *minimum support* tanpa disertai penurunan nilai *recall* yang dapat menurunkan nilai F-measure.

Tabel 5 Nilai F-measure jumlah rekomendasi maksimal

		<i>Minimum support (%)</i>				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity (%)</i>	20	0.251	0.286	0.328	0.389	0.378
	25	0.257	0.291	0.335	0.4	0.39
	30	0.282	0.298	0.341	0.41	0.401
	35	0.38	0.383	0.399	0.433	0.415
	40	0.447	0.449	0.463	0.482	0.455
	45	0.481	0.481	0.478	0.481	0.466
	50	0.541	0.541	0.548	0.534	0.534

Dari Tabel 5 juga dapat dilihat nilai F-measure pada kasus *coldstart* yang didapatkan oleh sistem yang dibangun. Peningkatan nilai F-measure pada kasus *coldstart* terjadi saat *minimum similarity* yang digunakan semakin besar. Nilai F-measure pada kasus biasa saat penggunaan *minimum support* 20% cenderung lebih tinggi dibandingkan pada kasus *coldstart*. Hal ini disebabkan karena nilai *precision* dan *recall* kasus biasa jumlah rekomendasi terbatas cenderung lebih tinggi dibandingkan pada kasus *coldstart*. Untuk jumlah rekomendasi terbatas ditunjukkan oleh Tabel 6.

Tabel 6 Nilai F-measure jumlah rekomendasi terbatas

		<i>Minimum support (%)</i>				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity (%)</i>	20	0.175	0.124	0.164	0.273	0.124
	25	0.175	0.124	0.164	0.273	0.124
	30	0.175	0.124	0.164	0.273	0.137
	35	0.272	0.151	0.164	0.286	0.15
	40	0.273	0.205	0.219	0.311	0.236
	45	0.288	0.221	0.22	0.31	0.299
	50	0.502	0.486	0.486	0.507	0.48

Pada jumlah rekomendasi terbatas untuk kedua kasus uji, nilai F-measure yang didapatkan lebih rendah dari jumlah rekomendasi maksimal. Seperti halnya pada jumlah rekomendasi maksimal, nilai F-measure untuk kasus *coldstart* relatif lebih rendah dari kasus biasa dengan *minimum support* 20%. Nilai f-measure tertinggi untuk jumlah rekomendasi maksimal dan jumlah rekomendasi terbatas didapatkan saat penggunaan *minimum similarity* 50%.

### 3.4 Mean Average Precision (MAP)

Nilai *mean average precision* (MAP) merupakan nilai rata-rata dari *average precision*. *Average precision* merupakan nilai yang didapatkan dari setiap nilai *precision item* relevan yang dihasilkan dan menggunakan nilai 0 untuk *item* relevan yang tidak dihasilkan oleh sistem. Nilai *precision* untuk *average precision* dihitung dengan memperhatikan urutan *item* yang diberikan oleh sistem, sehingga nilai *precision* diberikan untuk setiap *item* yang dihasilkan oleh sistem. Persamaan (4) merupakan persamaan untuk menghitung nilai *mean average precision* dalam *information retrieval* [17].

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad (4)$$

Dimana:

Q : jumlah *query* uji

R : *item* relevan yang dihasilkan oleh sistem

m : jumlah *item* relevan yang dihasilkan dari *query*

Dalam penelitian ini, *query* (Q) merupakan *item* uji yang digunakan dalam pengujian. Perhitungan *mean average precision* setiap *item* uji ditunjukkan oleh Tabel 7. Pengujian dengan penggunaan *minimum similarity* dan *minimum support* yang berbeda menunjukkan nilai *mean average precision* tidak selalu meningkat. Nilai *mean average precision* maksimal diperoleh saat *minimum similarity* yang digunakan sebesar 50% dan *minimum support* sebesar 5%. Nilai *mean average precision* dengan *minimum support* 5% selalu lebih tinggi saat penggunaan *minimum similarity* lebih besar dari 30%.

Tabel 7 Nilai *mean average precision*

		<i>Minimum support</i> (%)				<i>Coldstart</i>
		5	10	15	20	
<i>Minimum similarity</i> (%)	20	0.284	0.195	0.225	0.329	0.191
	25	0.285	0.195	0.225	0.332	0.198
	30	0.29	0.196	0.226	0.335	0.205
	35	0.359	0.243	0.273	0.35	0.211
	40	0.366	0.298	0.341	0.361	0.263
	45	0.323	0.304	0.306	0.3	0.281
	50	0.417	0.381	0.389	0.35	0.355

Nilai *mean average precision* cenderung selalu meningkat pada saat penggunaan *minimum similarity* 10% dan pada kasus *coldstart*. Selain penggunaan *minimum similarity* 10% dan kasus *coldstart*, nilai *mean average precision* menurun saat *minimum similarity* 45%. Penurunan nilai *mean average precision* saat *minimum similarity* 45% lebih disebabkan oleh berkurangnya *item hit* yang didapatkan oleh sistem.

Pada kasus *coldstart*, nilai *mean average precision* cenderung lebih rendah dari kasus biasa untuk semua *minimum similarity*. Pada penggunaan *minimum similarity* 50%, kasus *coldstart* hanya unggul dari penggunaan *minimum support* 20%. Sedangkan pada saat penggunaan *minimum support* 10%, kasus *coldstart* hanya unggul pada penggunaan *minimum similarity* 25% dan 30%.

#### 4. KESIMPULAN

Setelah melakukan pengujian terhadap *hybrid recommendation system* yang dibangun, maka dapat diperoleh kesimpulan sebagai berikut:

1. *Hybrid recommendation system* dapat dibangun dengan kombinasi penggalian *frequent itemset* dan perbandingan similaritas *keyword* antara dokumen. Penggalian *frequent itemset* dapat menggunakan algoritma FP-Growth dan similaritas antara dokumen dapat dihitung menggunakan *cosine similarity*. Ambang batas yang digunakan adalah *minimum support*, *minimum similarity*, dan jumlah dokumen yang direkomendasikan.
2. Pada pengujian untuk kasus biasa untuk jumlah rekomendasi maksimal, peningkatan nilai *precision* dan F-measure terjadi saat *minimum similarity* dan *minimum support* ditingkatkan. Hasil terbaik didapatkan saat *minimum support* yang digunakan sebesar

- 20% dan *minimum similarity* 50%. Sedangkan nilai *recall* cenderung menurun saat *minimum similarity* dan *minimum support* ditingkatkan. Hasil terbaik didapatkan pada saat *minimum similarity* kurang dari 40% dan *minimum support* kurang dari 15%. Untuk nilai *mean average precision*, nilai terbaik didapatkan pada saat *minimum support* 20% dan *minimum similarity* 50%.
3. Untuk rekomendasi terbatas, nilai *precision* dan F-measure tertinggi didapatkan pada penggunaan *minimum support* 20%. Penetapan *minimum support* yang lebih tinggi, dapat meningkatkan nilai *precision* dan f-measure. Sedangkan pada nilai *recall*, nilai tertinggi didapatkan pada penggunaan *minimum support* 10%. Peningkatan *minimum similarity* mengakibatkan meningkatnya nilai *precision*, dan F-measure. Nilai tertinggi pada *precision* dan F-measure didapatkan pada saat penggunaan *minimum similarity* 50% sedangkan pada *recall*, nilai tertinggi didapatkan pada saat *minimum support* kurang dari 40%.
  4. Nilai pengujian *precision*, *recall*, F-measure, dan *mean average precision* untuk kasus biasa dapat mencapai nilai yang lebih tinggi dibandingkan dengan kasus coldstart baik untuk jumlah rekomendasi terbatas maupun rekomendasi maksimal.

## 5. SARAN

Dari penelitian yang telah dilakukan sistem memiliki kelemahan pada skema *stopword*. Penerapan skema baru untuk mengatasi perubahan *stopword* diperlukan karena penambahan dan pengurangan *stopword* dapat mengakibatkan *keyword* yang diekstraksi dari dokumen menjadi tidak valid. Penggunaan *stopword* yang khusus untuk dokumen ilmiah juga dibutuhkan agar *stopword* yang terdaftar tepat guna sehingga dapat meningkatkan kinerja sistem. Untuk meningkatkan jumlah *item* relevan, diperlukan kombinasi algoritma lain saat penggalian *frequent itemset* tidak dapat memenuhi jumlah rekomendasi yang diinginkan. Penggalian *frequent itemset* membutuhkan waktu yang cukup lama karena adanya pembentukan *tree* yang berulang-ulang. Penerapan algoritma *incremental* untuk penggalian *frequent itemset* yang dapat mengatasi pengulangan proses penggalian *frequent itemset*. Pengujian pada penggunaan dokumen uji yang lebih banyak dan bervariasi juga dibutuhkan.

## DAFTAR PUSTAKA

- [1] McGinty, L. dan Smyth, B., 2006, Adaptive Selection : An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems, *International Journal of Electronic Commerce*, 11, 35–57.
- [2] Godfrey, A. L., 2007, A Product Segmentation Approach and its Relationship to Customer Segmentation Approaches and Recommendation System Approaches, *Thesis*, The University of Texas at Austin.
- [3] Chikhaoui, B., Chiazzaro, M., dan Wang, S., 2011, An Improved Hybrid Recommender System by Combining Predictions, *Workshops of International Conference on Advanced Information Networking and Applications*, 22-25 March 2011, 644-649.
- [4] Djamal, R. A., Maharani, W. dan Kurniati, A. P., 2010, Analisis dan Implementasi Metode Item-Based Clustering Hybrid pada Recommender System, *Konferensi Nasional Sistem dan Informatika 2010*, 13 November 2010, 216-222.
- [5] Hayati, N., 2011, Metode Hybrid (Content dan Collaborative Based) Nearest Neighbour Untuk Sistem Rekomendasi Pariwisata, *Skripsi*, PS. Teknologi Informasi FMIPA USU.

- 
- [6] Hsieh, S. M., Huang, S. J., Hsu, C. C., dan Chang, H. C., 2004, Personal Documents Recommendation System Based on Data Mining Techniques, *International Conference on Web Intelligence*, 20-24 September 2004, 51-57.
- [7] Li, Q. dan Kim, B. M., 2003, Clustering Approach for Hybrid Recommender System, *International Conference on Web Intelligence*, 13-17 Oktober 2003, 33-38.
- [8] Liangxing, Y. dan Aihua, D., 2010, Hybrid Product Recommender System for Apparel Retailing Customers, *WASE International Conference on Information Engineering*, 14-15 Agustus 2010, 356-360.
- [9] Pham, T. V. dan Thach, L. N., 2011, Social-aware Document Similarity Computation for Recommender Systems, *Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing*, 12-14 Desember 2011, 872-878.
- [10] Melamed, D., Shapira, B., dan Elovici, Y., 2007, MarCol: A Market-Based Recommender System, *IEEE Intelligent Systems*, 22, 74-78.
- [11] Popa, H. E., Negru, V., Pop, D., dan Muscalagiu, I., 2008, DL-AgentRecom - A multi-agent based recommendation system for scientific documents, *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 26-29 September 2008, 320-324.
- [12] Oelze, I., 2009, Automatic Keyword Extraction for Database Search, *Ph.D. Thesis*, University of Hannover, Hannover.
- [13] Tala, F. Z., 2003, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia, *Master Thesis*, Universiteit van Amsterdam
- [14] Zaki, M.J. dan Meira, W. Jr., 2014, *Data Mining and Analysis : Fundamental Concepts and Algorithms*, Cambridge University Press, Cambridge.
- [15] Han, J., Kamber, M., dan Pei, J., 2011, *Data Mining : Concepts and Techniques*, Morgan Kaufmann Publisher, San Francisco.
- [16] Jannach D., Zanker M., Felfernig, A., dan Friedrich, G., 2010, *Recommender Systems an Introduction*, Cambridge University Press, Cambridge.
- [17] Manning, C. D., Raghavan, P., dan Schütze, H., 2009, *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England.