Analisa Algoritma *Cosine Similarity* dengan *Pearson Correlation* pada Metode *Item-based Collaborative Filtering* dengan Menggunakan Dataset *Movielens*

Mufti Robbani[1], Rima Dias Ramadhani[2], Andika Elok Amalia[3]
[1,2] Program Studi Informatika
[3] Program Studi Rekayasa Perangkat Lunak
[1,2,3] Fakultas Teknik Industri dan Informatika, Institut Teknologi Telkom Purwokerto
Jl. DI Panjaitan 128, Purwokerto 53147, Indonesia
Email: [1]14102072@ittelkom-pwt.ac.id, [2]rima@ittelkom-pwt.ac.id, [3]andika@ittelkom-pwt.ac.id

Abstrak – Sistem rekomendasi merupakan sistem yang bertujuan untuk memberikan prediksi sejumlah informasi yang menarik bagi penggunanya. Pada sistem rekomendasi terdapat dua metode rekomendasi yang sudah dikenal secara luas yaitu *Content-based Filtering* dan *Collaborative Filtering*. *Item-based Collaborative Filtering* adalah metode rekomendasi klasik yang menggunakan *rating* pengguna untuk menghasilkan *item* rekomendasi berdasarkan kemiripan antar *item.* Penelitian ini membahas tentang analisa algoritma *Cosine Similarity* dengan *Pearson Correlation* pada metode *Item-based Collaborative Filtering*. Metode rekomendasi *Item-based Collaborative Filtering* mampu memberikan rekomendasi yang lebih baik dari *Content-based Filtering* karena menggunakan *rating* sebagai sumber informasi sehingga *item* rekomendasi yang dihasilkan mempunyai variasi konten dan kemungkinan disukai oleh pengguna. Dengan sample data dari *movie*lens 100k *rating* sebanyak 7 user dan 6 *movie* yang berbeda didapatkan 1 rekomendasi *movie* yang sama pada kategori *movie* yang paling mirip sedangkan pada *movie* yang paling tidak mirip kedua algoritma memberikan hasil yang berbeda dengan perbandingan nilai *similarity* dan komposisi rating yang cukup signifikan.

Kata kunci – *Cosine Similarity*, I*tem-based Collaborative Filtering*, *Movielens*, *Pearson Correlation*, Sistem Rekomendasi.

Abstract— Recommendation system is a system that aims to give a prediction of a number of information of interest to its users. In the system there are two methods of recommendations are already widely known that is Content-based Filtering and Collaborative Filtering. Item-based Collaborative Filtering recommendation method is a classic use of rating users to generate item recommendations based on the similarity between items. This study discusses the Cosine Similarity algorithm analysis with Pearson Correlation method on the item-based Collaborative Filtering. Item-based recommendation method in Collaborative Filtering was able to give a better recommendation of Content-based Filtering because it uses ratings as information sources so that the resulting recommendations items have variations of the content and the possibility favored by users. With sample data from the movielens 100k, 7 user rating and 6 different movies are chosen to obtain the movie recommendation and the result is 1 recommendations of the same movie on the movie most similar categories while in the movie, which at least is similar to both the algorithm gives different results with a comparison of the value of the similarity and the composition of a rating are significant enough.

Keywords – Cosine Similarity, Item-based Collaborative Filtering, Movielens, Pearson Correlation, Recommender System.

I. PENDAHULUAN

Perkembangan dan persebaran data informasi pada jaringan internet dari tahun ke tahun mengalami peningkatan yang *significant*. Peningkatan ini menyebabkan pengguna kesulitan untuk menentukan informasi yang tepat dan relevan sesuai kebutuhannya. Untuk itu diperlukan adanya sistem yang mampu mengolah dan memilah informasi yang dibutuhkan pengguna[1].

Sistem rekomendasi merupakan sebuah sistem yang mampu mengolah dan memilah informasi sesuai dengan kebutuhan pengguna [2]. Prediksi yang diberikan dapat membantu pengguna dalam banyak *scenario* dalam pembentukan keputusan dalam memilih sejumlah informasi yang dibutuhkan pengguna [3]. Sistem rekomendasi menjadi komponen penting yang digunakan dalam berbagai bidang kehidupan terutama dalam mendapatkan informasi

yang tepat dan relevan sesuai kebutuhan [1]. Banyak teknik yang dapat digunakan untuk sistem rekomendasi dan secara umum dikategorikan menjadi *Content-based Filtering* dan *Collaborative Filtering* [2][4]. *Content-based Filtering* memberikan rekomendasi berdasarkan nilai kemiripan dari feature dalam *item* dan pengguna[4][5][6], sedangkan *Collaborative Filtering* memberikan rekomendasi dengan mengolah informasi yang disukai pengguna dengan melihat *rating* yang diberikan oleh pengguna[7][8].

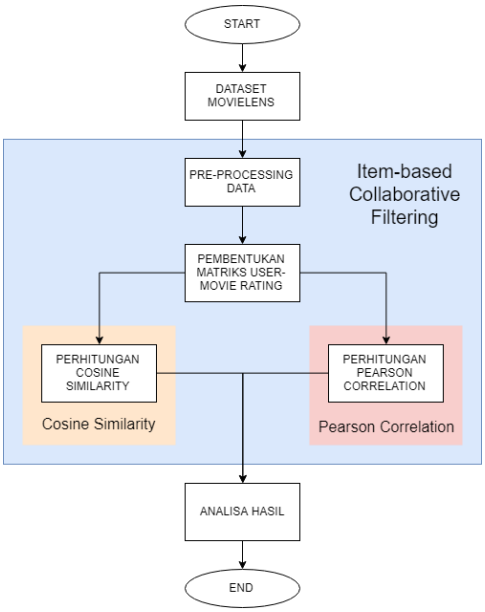
Teknik rekomendasi yang paling populer yaitu *Collaborative Filtering*[9][10][11]. Metode *Collaborative Filtering* memiliki dua pendekatan umum, yaitu *user-based* dan *model-based*. *User-based Collaborative Filtering* menggunakan pola kemiripan antar pengguna sedangkan *model-based* menggunakan pola kemiripan antar *item*[2][11].

Secara umum, pencarian pola kemiripan (*similarity*) tersebut dapat menggunakan *Cosine Similarity*[2] dan dapat dioptimalkan tingkat akurasi rekomendasi dengan menggunakan normalisasi pada matriks *user-movie rating* dengan menggunakan *Mean Centering Normalization*. Pengoptimalan ini dikenal juga sebagai *Centered Cosine Similarity* atau *Pearson Correlation* [12][13].

Penelitian ini menggunakan algoritma Cosine Similarity dan Pearson Correlation untuk mencari nilai similarity setiap movie (item-based). Kemudian dilakukan Analisa hasil dari nilai similarity kedua algoritma tersebut.

II. METODE PENELITIAN

Metode yang digunakan dapat dilihat pada diagram alir di Gambar 1. Pada diagram alir tersebut terdapat dua proses perlakuan data yang sama dengan input data yang berbeda yaitu *Cosine Similarity* dan *Pearson Correlation*.



Gambar 1. Flowchart *Item-based Collaborative Filtering*

A. *Dataset Movielens*

Dataset yang digunakan dalam penelitian ini yaitu dataset *movielens* yang diperoleh dari grup peneliti yang bernama Grouplens di Universitas Minnesota. Dataset *movielens* yang digunakan berisi 6 *file* berbentuk csv diantaranya *file genres*.csv, *genres_movies*.csv, *movies*.csv, occupation.csv, *ratings*.csv, dan *user*s.csv. *File* yang menjadi bahan pada penelitian ini yaitu *file ratings*.csv yang berisi *record* 100.000 *rating* yang diberikan *user* terhadap *movie*[14].

B. *Preprocessing Data*

Preprocessing data ini dilakukan untuk menghilangkan *invalid*, *ambiguous*, *out-of-range* dan *missing values* yang dapat menurunkan kualitas data. Sehingga *preprocessing* dapat mengoptimalkan data untuk penelitian[15]. Proses ini dilakukan dengan menggunakan Python. Setelah *preprocessing* data selesai kemudian data dibagi menjadi dataset training 80% dan testing 20% dengan menggunakan teknik *k-fold cross validation*. Proses pembagian data ini juga menggunakan Python.

C. *Pembentukan Matriks User-Movie Rating*

Pembentukan matriks *user-movie rating* dilakukan dengan menggunakan Python dengan menggunakan *file input users*.csv, *movies*.csv dan *ratings*.csv. Pada *movie j* yang tidak di*rating* oleh *user i* maka nilai *rating* dianggap sebagai 0 yang berarti *user i* belum memberikan *rating* pada *movie j*.

D. *Perhitungan Nilai Kemiripan (Similarity)*

Perhitungan nilai kemiripan pada penelitian ini menggunakan *Cosine Similarity* (persamaan 1)[16] dan *Pearson Correlation* (persamaan 2).

$$s(i,j) = \cos\left(\vec{R}(*,i), \vec{R}(*,j)\right) = \frac{\vec{R}(*,i) \cdot \vec{R}(*,j)}{\|\vec{R}(*,i)\| \cdot \|\vec{R}(*,j)\|} \quad (1)$$

keterangan:

$s(i,j)$ = nilai *similarity item i* dan *item j*

$\vec{R}(*,i) \cdot \vec{R}(*,j)$ = *sumproduct* nilai *rating item i* dan *item j*

$\|\vec{R}(*,i)\| \cdot \|\vec{R}(*,j)\|$ = jumlah perkalian nilai *rating absolute rating item i* dan *item j*

$$Sim(u,v) = P, C(u,v) = \frac{\sum_{i \in I}\left(R_{u,i} - \bar{R}_u\right)\left(R_{v,i} - \bar{R}_v\right)}{\sqrt{\sum_{i \in I}\left(R_{u,i} - \bar{R}_u\right)^2 \left(R_{v,i} - \bar{R}_v\right)^2}} \quad (2)$$

keterangan:

$Sim(u,v)$ = nilai *similarity item u* dan *item v*

$(R_{u,i} - \overline{R_u})$ = normalisasi nilai *rating item u*

$\sum_{i \in I}\left(R_{u,i} - \bar{R}_u\right)\left(R_{v,i} - \bar{R}_v\right)$ = *sumproduct* nilai *rating absolute rating item u* dan *item v*

Dengan menggunakan Persamaan 1 *Cosine Similarity*, perhitungan nilai kemiripan antara *movie* 258 (m258) dengan *movie* 259 (m259) dengan

menggunakan *sample* data *rating* dari *user* 1 sampai *user* 7, dapat dilihat pada Gambar 3.

$$s(m258, m259) = \frac{19}{9.11043 * 3.46410}$$

$$s(m258, m259) = \frac{19}{31.55947}$$

$$s(m258, m259) = 0.60204$$

Gambar 3. Perhitungan *Similarity* Dengan Menggunakan *Cosine Similarity*

Sedangkan perhitungan *similarity* dengan menggunakan *Pearson Correlation* dapat dilihat pada Gambar 4 dengan studi kasus yang sama dengan *Cosine Similarity*.

$$s(m258, m259) = \frac{0.75}{3.08221 * 1.73205}$$

$$s(m258, m259) = \frac{0.75}{5.338539126}$$

$$s(m258, m259) = 0.14049$$

Gambar 1. Perhitungan *Similarity* Dengan Menggunakan *Pearson Correlation*

III. HASIL PENELITIAN

Penelitian ini menggunakan bahasa pemrograman Python untuk membuat matriks *user-movie rating* dan matriks *similarity* dengan algoritma *Cosine Similarity* dan *Pearson Correlation*. Percobaan menggunakan beberapa sample data *rating* pada *movie* dan *user* tertentu yaitu m258, m259, m260, m269, m294, m321, u1, u2, u3, u4, u5, u6 dan u7. Tabel 1 menunjukkan hubungan kemiripan dengan menggunakan *Cosine Similarity* antara *movie* 258 dengan *movie* lainnya.

Tabel 1. *Similarity* dengan *Cosine Similarity* antara *movie* 258 dengan *movie* lainnya.

	u1	**u2**	**u3**	**u4**	**u5**	**u6**	**u7**	***Cosine Similarity* m258**
m258	5	3	2	5		2	4	1.00000
m259	1				1	1	3	0.60204
m260	1	0	4	4			1	0.69650
m269	5	4				4	3	0.77013
m294		4	2	5			4	0.80107
m321			5			3		0.30119

Dari Tabel 1 didapatkan kesimpulan bahwa *movie* yang paling mirip dengan *movie* 258 adalah *movie* 294 dengan nilai *similarity* sebesar 0.80107 diikuti oleh *movie* m269 dengan nilai *similarity* sebesar 0.77013. Sedangkan *movie* yang paling tidak mirip dengan *movie* 258 yaitu *movie* 321 dengan nilai *similarity* 0.30119. Percobaan selanjutnya dengan menggunakan algoritma *Pearson Correlation* didapatkan hasil pada Tabel 3.

Pada Tabel 2 menjelaskan mengenai proses normalisasi pada matriks *user-movie rating* untuk menerapkan algoritma *Pearson Correlation*. Normalisasi ini dilakukan dengan cara mengurangi *rating* dengan nilai rata-rata *rating* pada setiap *movie*, sedangkan pada bagian matriks *user-movie rating* yang tidak memiliki nilai (*user* tidak memberikan *rating* pada *movie*) akan diberi nilai 0. Dengan begitu jumlah dari *rating* yang telah dinormalisasi pada setiap *movie* akan bernilai 0.

Pada Tabel 3 menjelaskan mengenai nilai *similarity* antara *movie* 258 dengan *movie* lainnya dengan menggunakan algoritma *Pearson Correlation*. *Movie* yang memiliki nilai *similarity* tertinggi ada pada *movie* 294 dengan nilai *similarity* sebesar 0.66989. Peringkat kedua yang paling mirip yaitu *movie* 269 dengan nilai *similarity* sebesar 0.22942. Sedangkan *movie* yang paling tidak mirip ditempati oleh *movie* 260 dengan nilai *similarity* terkecil yaitu -0.32444.

Tabel 2. Normalisasi *Rating Movie* 258 dan *Movie* 259

i	**m258** $(R_{u,i})$	**m259** $(R_{v,i})$	$(R_{u,i} - \bar{R}_u)$	$(R_{v,i} - \bar{R}_v)$
u1	5	1	1.5	-0.5
u2	3		-0.5	0
u3	2		-1.5	0
u4	5		1.5	0
u5		1	0	-0.5
u6	2	1	-1.5	-0.5
u7	4	3	0.5	1.5
Mean $(\bar{R})$	3.5	1.5		
Sum (Σ)			0	0

Tabel 3. *Similarity* dengan *Pearson Correlation* antara *movie* 258 dengan *movie* lainnya

	u1	**u2**	**u3**	**u4**	**u5**	**u6**	**u7**	***Pearson Correlation* m258**
m258	5	3	2	5		2	4	1.00000
m259	1				1	1	3	0.14049
m260	1		4	4			1	-0.32444
m269	5	4				4	3	0.22942
m294		4	2	5			4	0.66989
m321			5			3		0.00000

IV. PEMBAHASAN

Pada bab sebelumnya didapatkan hasil *similarity* dari algoritma *Cosine Similarity* dan *Pearson Correlation*. Terdapat perbedaan hasil pada kedua algoritma *similarity* yang dapat dilihat pada Tabel 4. Perbedaan tersebut terletak pada nilai *similarity*.

Tabel 4. Perbandingan *Similarity*

movie	*Cosine Similarity* **m258**	*Pearson Correlation* **m258**
m258	1.00000	1.00000
m259	**0.60204**	0.14049
m260	0.69650	**-0.32444**
m269	**0.77013**	**0.22942**
m294	**0.80107**	**0.66989**
m321	**0.30119**	**0.00000**

Tabel 4 menjelaskan perbedaan nilai *similarity movie* 258 dengan selain *movie* 258. Pada *Cosine Similarity* rentang nilai *similarity* terdapat pada 0 sampai 1 sehingga untuk mengetahui *movie* memiliki kemiripan dengan *movie* lainnya perlu dilakukan perankingan, sedangkan rentang nilai *similarity* pada *Pearson Correlation* adalah -1 (sangat tidak mirip) sampai 1 (sangat mirip) dengan 0 sebagai nilai tengah (netral).

Perbedaan nilai *similarity* yang paling jelas terlihat pada *movie* yang paling tidak mirip dengan *movie* 258. Pada *Pearson Correlation*, *movie* yang paling tidak mirip yaitu *movie* 260 dengan nilai *similarity* sebesar -0.32444 disusul oleh *movie* 321 dengan nilai *similarity* 0. Jika ditelusuri *record rating* yang diberikan *user* pada matriks *user-movie rating* terlihat bahwa pada *movie* 321 memiliki nilai *rating* yang bertentangan sebanyak 1 *rating* yaitu pada *user* 3 dan *user* 6 (Tabel 5), *movie* 260 sebanyak 3 *rating* (Tabel 6) yaitu pada *user* 1, *user* 3, dan *user* 7.

Tabel 5. Perbedaan *Rating Movie* 258 dan *Movie* 321

	u1	**u2**	**u3**	**u4**	**u5**	**u6**	**u7**
m258	5	3	2	5		2	4
m321			5			3	

Tabel 6. Perbedaan *Rating Movie* 258 dan *Movie* 260

	u1	**u2**	**u3**	**u4**	**u5**	**u6**	**u7**
m258	5	3	2	5		2	4
m260	1		4	4			1

Pada perbandingan antara *movie* 258 dengan *movie* 321, perbedaan *rating* cukup signifikan terjadi pada *user* 3 sedangkan pada *user* 6 tidak terjadi perbedaan *rating* yang besar. Sedangkan perbandingan antara *movie* 258 dengan *movie* 260, perbedaan *rating* signifikan terjadi pada *user* 1, *user* 3 dan *user* 7. Pada *user* 4 tidak terjadi perbedaan *rating* yang signifikan karena kedua *rating* yang diberikan termasuk kategori "disukai" dengan *rating* yang tinggi.

Dari perbandingan data *record rating* dapat disimpulkan bahwa *movie* 258 paling tidak mirip dengan *movie* 260 karena terdapat sebanyak 3 *user* yang memberikan perbedaan *rating* yang cukup signifikan (disukai dan tidak disukai). Hal ini sesuai dengan algoritma *Pearson Correlation*.

V. PENUTUP

Kesimpulan yang diperoleh dari penelitian ini yaitu penggunaan algoritma *Pearson Correlation* lebih efektif pada studi kasus metode *Item-based Collaborative Filtering* untuk memberikan rekomendasi *item* berdasarkan nilai kemiripan antar *item* daripada menggunakan *Cosine Similarity* karena pada *Pearson Correlation* data rating tidak langsung dicari nilai similarity nya akan tetapi dinormalisasikan terlebih dahulu menggunakan *Mean Centering Normalization* sehingga hasil yang didapatkan memiliki klasifikasi kriteria *similarity* dengan rentang nilai -1 (sangat tidak mirip) sampai 1 (sangat mirip) dengan 0 sebagai nilai tengah (netral). Hal ini membuat algoritma *Pearson Correlation* lebih efektif memberikan nilai *similarity* daripada algoritma *Cosine Similarity*.

.

DAFTAR PUSTAKA

[1] Z. Qiu, M. Chen, and J. Huang, "Design of Multi-mode E-commerce Recommendation System," *Third Int. Symp. Intell. Inf. Technol. Secur. Informatics*, no. 807018, pp. 530–533, 2010.

[2] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, vol. 532. New York: Springer, 2009.

[3] G. Linden, B. Smith, and J. York, "Amazon.com Tecommendations: Item-to-Item Collaborative Filtering," 2003.

[4] B. M. Kim and B. M. Kim, "A new approach for combining content-based and collaborative filters," no. May, 2014.

[5] R. Oktoria, W. Maharani, and Y. Firdaus, "Content-Based Recommender System Menggunakan Algoritma Apriori," in *Konferensi Nasional Sistem dan Informatika*, 2010, pp. 124–129.

[6] P. S. Adi, "Sistem Rekomendasi Nilai Mata Kuliah menggunakan Metode Content-Based Filtering," in *Seminar Nasional Informatika*, 2010, p. A.90-A.94.

[7] E. A. Laksana, "Collaborative Filtering dan Aplikasinya," vol. 1, no. 1, pp. 36–40, 2014.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. tenth Int. Conf. World Wide Web - WWW '01*, 2001.

[9] R. Katarya and O. P. Verma, "An effective collaborative movie recommender system with cuckoo search," *Egypt. Informatics J.*, vol. 18, no. 2, pp. 105–112, Jul. 2017.

[10] R. A. Djamal, W. Maharani, and P. Kurniati, "Analisis Dan Implementasi Metode Item-Based Clustering Hybrid Pada Recommender System," in *Konferensi Nasional Sistem dan Informatika*, 2010, pp. 216–222.

[11] R. Zhang, Q. Liu, J. Wei, and Huiyi-Ma, "Collaborative Filtering for Recommender Systems," in *Advanced Cloud and Big Data Collaborative*, 2014, pp. 301–308.

[12] T. Arsan, "Comparison Of Collaborative Filtering Algorithms With Various Similarity Measures For Movie Recommendation," vol. 6, no. 3, pp. 1–20, 2016.

[13] I. Journal *et al.*, "International Journal Of Engineering Sciences & Research Technology An Implementation Of

Pearson Correlation Method For Predicting Items To User In E-Commerce," vol. 5, no. 7, pp. 873–882, 2016.

[14] F. M. Harper and J. A. Konstan, "The MovieLens Datasets," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.

[15] C. Hector, *Practical Data Analysis*. 2013.

[16] Wiranto and E. Winarko, "Konsep Multicriteria Collaborative Filtering untuk Perbaikan Rekomendasi," in *Seminar Nasional Informatika*, 2010, pp. D95–D101.