

# **User-Based Collaborative Filtering Dengan Memanfaatkan Pearson-Correlation Untuk Mencari Neighbors Terdekat Dalam Sistem Rekomendasi**

Arvid Theodorus<sup>1</sup>, Djoko Budiyanto Setyohadi<sup>2</sup>, Ernawati<sup>3</sup>

**Magister Teknologi Informasi**  
**Universitas Atma Jaya Yogyakarta**  
Email : arvid.thd@gmail.com

## **ABSTRAK**

Perkembangan yang pesat seputar sistem rekomendasi mendorong para peneliti untuk terus mencari teknik-teknik rekomendasi baru. Tujuannya adalah untuk mencari teknik yang terbaik dalam memberi rekomendasi bagi pengguna. Namun, beberapa teknik rekomendasi masih belum memiliki metode yang berfungsi untuk mencari relasi antar pengguna. Untuk itu, penelitian ini akan berfokus pada metode pencarian relasi antar pengguna dalam sebuah sistem rekomendasi. Peneliti memanfaatkan teknik pengukuran *Pearson-Correlation* dalam algoritma *User-Based Collaborative Filtering* untuk menghitung relasi kedekatan antar pengguna. Informasi kedekatan tersebutlah yang nantinya diharapkan akan bermanfaat sebagai dasar sumber rekomendasi yang akurat.

Katakunci : User Based Collaborative Filtering, Pearson Correlation, Sistem Rekomendasi, Evaluasi Sistem Rekomendasi

# User-Based Collaborative Filtering Dengan Memanfaatkan Pearson-Correlation Untuk Mencari Neighbors Terdekat Dalam Sistem Rekomendasi

Arvid Theodorus<sup>1</sup>, Djoko Budiyanto Setyohadi<sup>2</sup>, Ernawati<sup>3</sup>

Magister Teknologi Informasi  
Universitas Atma Jaya Yogyakarta  
Email : arvid.thd@gmail.com

## ABSTRAK

Perkembangan yang pesat seputar sistem rekomendasi mendorong para peneliti untuk terus mencari teknik-teknik rekomendasi baru. Tujuannya adalah untuk mencari teknik yang terbaik dalam memberi rekomendasi bagi pengguna. Namun, beberapa teknik rekomendasi masih belum memiliki metode yang berfungsi untuk mencari relasi antar pengguna. Untuk itu, penelitian ini akan berfokus pada metode pencarian relasi antar pengguna dalam sebuah sistem rekomendasi. Peneliti memanfaatkan teknik pengukuran *Pearson-Correlation* dalam algoritma *User-Based Collaborative Filtering* untuk menghitung relasi kedekatan antar pengguna. Informasi kedekatan tersebutlah yang nantinya diharapkan akan bermanfaat sebagai dasar sumber rekomendasi yang akurat.

Katakunci : User Based Collaborative Filtering, Pearson Correlation, Sistem Rekomendasi, Evaluasi Sistem Rekomendasi

## PENDAHULUAN

Penelitian seputar sistem rekomendasi dalam 1 dekade terakhir ini berkembang dengan pesat. Lebih dari 200 artikel penelitian yang membahas tentang sistem rekomendasi sudah terpublikasi (Beel, 2013). Seiring bertambahnya artikel-artikel tersebut, algoritma rekomendasi yang dikembangkan dan di inovasikan sesuai dengan kebutuhan masing-masing sistem pun semakin banyak. Berdasarkan sudut pandang penyaringan informasinya, algoritma rekomendasi yang sering digunakan dalam sistem rekomendasi terbagi menjadi tiga jenis, yaitu : *Content-Based Filtering* (CB), *User-Based Collaborative Filtering* (UCF) dan *Item-Based Collaborative Filtering* (ICF).

Beberapa peneliti mengembangkan ketiga algoritma penyaringan tersebut secara terpisah, namun ada juga peneliti yang menggabungkan lebih dari satu algoritma dengan tujuan mendapatkan hasil rekomendasi yang terbaik (Hu et al., 2011), (De Campos et al., 2010), (Chen et al. 2010). Untuk mengetahui apakah algoritma yang dikembangkan mendapatkan hasil yang terbaik atau tidak, para peneliti menggunakan beberapa teknik evaluasi seperti *Mean Absolute Error* (MAE), *Precision Value*, *Recall*, dan lainnya. (Beel, 2013), (Jin, 2006), (Claypool, 1999).

Dalam penerapannya pada sistem rekomendasi, UCF seringkali dipasangkan dengan ICF karena memiliki banyak kesamaan dalam teknik penyaringan dan bahkan dalam hal kekurangan dan kelebihan. UCF mengasumsikan bahwa untuk mencari sesuatu yang akan disukai oleh pengguna A, maka kita harus mencari pengguna lain dengan kecenderungan yang sama. Pengguna lain yang memiliki ketertarikan / kecenderungan yang sama dengan pengguna A disebut dengan *neighbors*. Nantinya, segala sesuatu yang disukai oleh *neighbors* akan menjadi sumber rekomendasi untuk pengguna A. Sedikit berbeda dengan UCF, ICF menampung segala *item* yang telah disukai oleh pengguna A terlebih dahulu dan kemudian mulai menelusuri *item* lain yang akan disukai

oleh pengguna A. Perbedaan ICF dan UCF terletak pada titik awal dalam mencari rekomendasi untuk seorang pengguna dan jenis *neighbors*-nya. (Lops et al., 2011), (Sarwar et al., 2001).

Peran *neighbor* sangatlah penting dalam sistem rekomendasi, selain berperan sebagai sumber rekomendasi, ada juga data lain pada *neighbors* yang nantinya akan digunakan untuk menghitung nilai prediksi. Dalam hal ini data tersebut adalah nilai *rating neighbors* terhadap *item* rekomendasi. Nilai prediksi ini nantinya digunakan oleh para peneliti sebagai tolak ukur seberapa “baik” algoritma rekomendasi yang sedang dikembangkan.

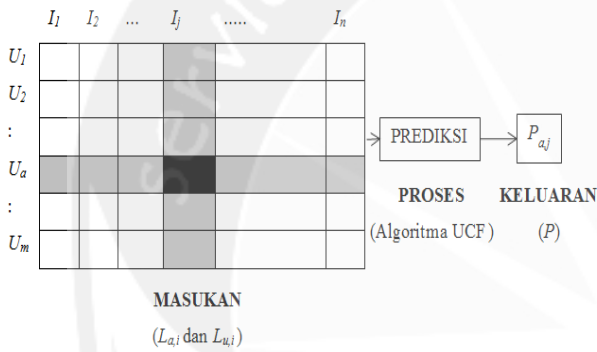
Pada penelitian kali ini, peneliti mencoba memanfaatkan sebuah metode pengukuran dengan harapan bisa mendapatkan *neighbors* terbaik, sebagai dasar dalam pencarian rekomendasi dan nilai prediksi yang akurat. Metode pengukuran yang digunakan nantinya adalah *Pearson-Correlation Coefficient*. Pengukuran tersebut digunakan untuk mencari *neighbors* dengan kemiripan dan kecenderungan yang sama terhadap pengguna aktif.

## USER-BASED COLLABORATIVE FILTERING (UCF)

Pada UCF, ada banyak teknik yang digunakan untuk mencari nilai prediksi untuk pengguna aktif ( $P_{a,i}$ ), dimulai dari cara sederhana dengan menjumlahkan semua *rating* dari pengguna lain yang memiliki selera sama dengan pengguna aktif ( $U_a$ ) dan kemudian dibagi dengan jumlah pengguna tersebut, hingga menggunakan cara pembobotan. Seperti yang sudah dibahas sebelumnya, pengguna lain yang memiliki selera sama dengan  $U_a$  disebut dengan *neighbors*. Untuk mencari *neighbors* tersebut banyak cara yang ditawarkan, ada yang memanfaatkan teknik *K-nearest neighbor* (KNN) ada juga yang mencarinya secara manual dengan mencari irisan dari semua daftar *item* yang sudah di *rating* oleh pengguna aktif ( $L_{a,i}$ ) dengan semua daftar *item* yang sudah di *rating* oleh pengguna lain ( $L_{u,i}$ ) satu-persatu (Shinde, S. K. & Kulkarni, U. V., 2011) dan (Ghauth, K. I. & Abdullah, N.

A., 2011). Hal terpenting yang perlu diperhatikan dalam mencari *neighbors* adalah calon *neighbors* setidaknya harus memiliki beberapa variabel yang tidak kosong (*null*) dari irisan  $L_{u,i}$  dengan  $L_{a,i}$  supaya nantinya dapat dilakukan pengukuran kedekatan.

Secara garis besar, dalam mencari nilai prediksi ( $P$ ) dan daftar rekomendasi ( $L$ ) untuk pengguna aktif, UCF dapat dibagi menjadi tiga tahap yaitu : masukan (*input*), proses (algoritma UCF) dan keluaran ( $P$  dan  $L$ ). *Input* yang dimaksud adalah berupa nilai dari irisan  $L_{a,i}$  dengan  $L_{u,i}$  dan kemudian nilai tersebut akan diproses menggunakan algoritma UCF yang menghasilkan nilai  $P_{a,i}$ . Lihat Rumus (3) untuk skema tahapan UCF, dimana  $I_j$  adalah *item* yang akan dicari nilai  $P$ -nya. Dalam algoritma UCF, sebelum mencari nilai  $P_{a,j}$  perlu dicari terlebih dahulu *neighbors* dari  $U_a$ , setelah mendapatkan *neighbors* yang cukup, maka kemudian diambil  $L_{u,j}$  dengan  $u$  adalah semua *neighbors* dari pengguna aktif ( $u_1$  s/d  $u_m$ ). Dari titik ini,  $L_{u,j}$  bisa diproses dengan banyak jenis algoritma, mulai dari yang paling sederhana, pembobotan ringan dan pembobotan rata-rata *rating*  $U$ , untuk lebih jelasnya dapat dilihat pada Rumus (1).



Gambar 1. Skema Tahapan UCF

$$P_{a,j} = \frac{\sum_{u=1}^m R_{u,j}}{m} \quad (1)$$

Pada Rumus (1), variabel  $m$  mewakili jumlah *neighbors* yang sudah terpilih dengan pendekatan tertentu. Algoritma ini adalah termasuk yang paling sederhana dalam mencari nilai  $P$ , disebut sederhana karena dalam perhitungannya tidak melibatkan variabel yang menunjukkan bahwa seorang *neighbors* berbeda dari *neighbors* lainnya. Algoritma sederhana ini dapat dikembangkan lagi menjadi sedikit lebih spesifik untuk membedakan setiap *neighbors*, caranya dengan memberi bobot pada masing-masing nilai  $R_{u,j}$  *neighbors*.

$$P_{a,j} = \frac{\sum_{u=1}^m R_{u,j} \cdot W_u}{\sum W_u} \quad (2)$$

Pada Rumus (2), variabel  $W$  adalah bobot yang diberikan untuk pengguna  $U_m$ , nilai  $W$  bisa ditentukan secara manual oleh  $U_a$  atau pembuat sistem, bahkan dapat

juga dicari menggunakan algoritma tertentu. Selain dengan memberi bobot untuk setiap *neighbors*, cara lain untuk mengembangkan rumus (2), yaitu dengan menambahkan variabel rata-rata *rating* dari  $U_a$  dan  $U_m$ . Lebih jelasnya lihat Rumus (3).

$$P_{a,j} = \bar{R}_a + \frac{\sum_{u=1}^m (R_{u,j} - \bar{R}_u)}{m} \quad (3)$$

Variabel  $\bar{R}_a$  didapat dari rata-rata  $R_{a,j}$  pada  $L_{a,j}$  sedangkan  $\bar{R}_u$  didapat dari rata-rata  $R_{u,j}$  pada  $L_{u,i}$  dengan asumsi rata-rata *rating* menunjukkan “sifat” dasar seorang pengguna. Kata “sifat” menunjuk pada aktivitas lampau seorang pengguna, apakah dia seorang *hater* yang selalu memberi *rating* rendah, ataukah ia adalah seorang *lover* yang selalu memberi *rating* tinggi pada setiap *item* dalam  $L$  yang dimilikinya. Semakin rendah nilai  $\bar{R}$  pada seorang *neighbor* maka semakin besar kontribusi nilainya terhadap  $P_{a,j}$  ketika suatu saat ia memberi nilai tinggi pada sebuah *item*  $j$ , tetapi jika ia memberi nilai rendah terhadap *item*  $j$  maka kontribusi nilainya kecil terhadap pencarian  $P_{a,j}$ . Sebaliknya, jika seorang *neighbor* memiliki nilai  $\bar{R}$  yang tinggi, maka semakin kecil kontribusinya terhadap pencarian  $P_{a,j}$  ketika ia memberi *rating* tinggi pada *item*  $j$ , tetapi jika ia memberi nilai rendah terhadap *item*  $j$  maka kontribusi nilainya sangat besar.

Algoritma pembobotan dan rata-rata *rating* pengguna juga dapat digabungkan supaya  $P$  yang didapat lebih spesifik. Caranya, pada Rumus (3) akan ditambahkan variabel bobot untuk setiap *neighbors*-nya, sekali lagi bobot dapat dicari dengan menggunakan berbagai cara tergantung dari asumsi pembuat sistem, lihat Rumus (4).

$$P_{a,j} = \bar{R}_a + \frac{\sum_{u=1}^m (R_{u,j} - \bar{R}_u) \cdot W_u}{\sum W_u} \quad (4)$$

Dengan menggabungkan pembobotan dan rata-rata *rating* pengguna, nilai  $P_{a,j}$  yang didapat akan lebih bersifat pribadi karena melibatkan dua variabel yang berasal dari kegiatan lampau pengguna aktif, maupun *neighbors*.

Pada penelitian kali ini, peneliti menggunakan Rumus (4) untuk mencari prediksi, karena variabel yang digunakan sangatlah spesifik dan melibatkan lebih banyak variabel (rata-rata *rating* pengguna aktif, *rating* pengguna aktif terhadap *item* aktif, rata-rata *rating* *neighbors* hingga nilai bobot kedekatan antara *neighbors* dengan pengguna aktif) yang ada dalam sistem rekomendasi. Peneliti juga memanfaatkan metode pengukuran *Pearson-Correlation Coefficient* untuk mengukur seberapa “dekat” pengguna aktif dengan pengguna lainnya. Semakin dekat pengguna tersebut dengan pengguna aktif, maka ia akan menjadi kandidat sumber rekomendasi (*neighbors*) dan nilai *rating*-nya akan mempengaruhi semua nilai prediksi yang akan dihitung nantinya.

#### PEARSON-CORRELATION COEFFICIENT

Korelasi adalah sebuah teknik pengukuran yang menentukan seberapa dekat relasi antar dua himpunan bilangan yang berbeda. Dengan syarat himpunan bilangan

tersebut harus memiliki urutan yang tetap dan berpasangan satu dengan lainnya antar kedua himpunan. Hasil pengukuran dapat berupa relasi positif ataupun relasi negatif. Relasi positif menunjukkan bahwa kedua himpunan memiliki kecenderungan kenaikan atau penambahan nilai yang sejajar. Sedangkan relasi negatif menunjukkan kedua himpunan memiliki kecenderungan penurunan atau pengurangan nilai yang sejajar. Sejajar dalam konteks ini berarti penurunan atau kenaikan nilai yang saling mengikuti antar kedua variabel tersebut. Salah satu teknik pengukuran korelasi adalah *Pearson Product Moment Correlation* atau biasa disingkat menjadi *Pearson Correlation*.

Rumus yang digunakan pada teknik pengukuran ini bisa dilihat dibawah ini:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (5)$$

Variabel  $r$  menunjukkan nilai korelasi yang didapat dari dua himpunan bilangan  $x$  dan  $y$  yang memiliki urutan dan berpasangan. Nilai  $r$  memiliki batas atas yaitu 1 dan batas bawah yaitu -1. Jika  $r$  bernilai 0 berarti kedua himpunan tidak memiliki relasi sama sekali. Jika  $r$  bernilai 1 berarti kedua himpunan memiliki penambahan nilai yang searah (sejajar), sedangkan jika  $r$  bernilai -1 maka kedua himpunan memiliki penurunan yang searah. Untuk  $r$  yang bernilai 1 atau -1 bisa disimpulkan bahwa kedua himpunan memiliki relasi atau kedekatan. Lihat pada Gambar 2 untuk contoh dua himpunan bilangan yang memiliki korelasi negatif dan positif.

Correlation = 0,91			Correlation = -1				
	I1	I2		I1	I2		
U1		5	4	U1	5	1	
U2		4	3	U2		5	1
U3		2	1	U3		5	1
U4		4	3	U4		5	1
U5		1	1	U5		5	1
U6		5	4	U6		1	5
U7		5	5	U7		1	5
U8		1	1	U8		1	5
U9		3	2	U9		1	5
U10		4	2	U10		1	5

Gambar 2. Contoh dua himpunan bilangan yang memiliki korelasi yang positif dan negatif

Variabel I1 dan I2 adalah dua himpunan bilangan yang berurutan dan berpasangan dari U1 – U10.

### MEAN ABSOLUTE ERROR, PRECISION DAN RECALL VALUE

Dalam sistem rekomendasi, terutama yang memanfaatkan perhitungan prediksi, evaluasi keakuratan daftar rekomendasi dan prediksi sangatlah penting. Dengan memanfaatkan teknik evaluasi yang sudah tersedia, peneliti dapat mengetahui seberapa akurat teknik rekomendasi dan prediksi yang dikembangkan.

Pada bagian ini, akan dibahas tiga jenis evaluasi yaitu : *Mean Absolute Error* (MAE), *Precision*, dan *Recall Value*. MAE bertujuan menghitung seberapa besar rata-rata selisih nilai prediksi *rating* yang dihasilkan oleh peneliti dengan nilai *rating* yang diberikan oleh pengguna, sedangkan *Precision* dan *Recall* bertujuan untuk menghitung seberapa banyak persentase *item* yang direkomendasikan dan di beri *rating* oleh pengguna

ataupun *item* yang sudah di *rating* tetapi tidak direkomendasikan oleh sistem.

Evaluasi MAE memanfaatkan teknik perhitungan yang sangat sederhana, yaitu dengan mencari selisih dari semua *item* yang sudah diberi *rating* oleh pengguna dan memiliki nilai *prediksi*. Nantinya selisih tersebut akan di absolute-kan (nilai positif) dan akhirnya dirata-rata. Dari hasil MAE, dapat terlihat jelas seberapa “jauh” selisih nilai prediksi *rating* yang diberikan oleh sistem dengan nilai *rating* yang diberikan oleh pengguna. Semakin besar nilai yang dihasilkan oleh MAE maka dapat diartikan bahwa nilai prediksi yang dihasilkan semakin tidak akurat, sebaliknya jika nilai MAE yang dihasilkan mendekati 0 maka prediksi yang dihasilkan sistem semakin mendekati akurat.

Rumus yang digunakan dalam evaluasi MAE dapat dilihat dibawah ini :

$$MAE = \frac{1}{n} \sum_{i=1}^n |P_{a,i} - R_{a,i}| \quad (6)$$

Untuk  $n$  adalah jumlah semua *item* pada daftar rekomendasi pengguna aktif yang memiliki nilai prediksi.  $P_{a,i}$  adalah nilai prediksi *item* ke- $i$  milik pengguna aktif dan  $R_{a,i}$  adalah nilai *rating* yang diberikan oleh pengguna aktif untuk *item* ke- $i$ .

Selanjutnya, untuk mengevaluasi hasil rekomendasi yang dihasilkan oleh sistem melalui teknik peyaringan yang dibuat oleh peneliti, *Precision* dan *Recall Value* dimanfaatkan untuk melihat seberapa akurat rekomendasi *item* yang dihasilkan untuk tiap pengguna sistem.

Dalam memberikan rekomendasi, ada beberapa kemungkinan yang akan dihasilkan, antara lain : (1) *item* yang sudah diberi *rating* oleh pengguna, juga direkomendasikan oleh sistem (*True-Positive*); (2) *item* yang di rekomendasikan, tetapi tidak di *rating* oleh pengguna; (3) *item* yang sudah di beri *rating* tetapi tidak direkomendasikan. Beberapa kemungkinan tersebut bisa diringkas menjadi tabel dibawah ini.

Tabel 1. Klasifikasi Dari Beberapa Kemungkinan Hasil Rekomendasi

	Rekomendasi	Tidak Direkomendasikan
Sudah Diberi <i>Rating</i>	<i>True-Positive</i> (TP)	<i>False-Negative</i> (FN)
Tidak Diberi <i>Rating</i>	<i>False-Positive</i> (FP)	<i>True-Negative</i> (TN)

Hasil dari klasifikasi diatas dapat dimanfaatkan untuk mencari nilai *Precision* dan *Recall*. *Precision* dapat diasumsikan sebagai sebuah cara untuk melihat seberapa banyak *item* yang “tepat” direkomendasikan untuk pengguna sistem, sedangkan *Recall* dapat diasumsikan sebagai sebuah cara untuk mengetahui seberapa banyak *item* yang sudah di *rating* oleh pengguna dan direkomendasikan untuknya. Untuk itu, rumus untuk mencari nilai *Precision* dan *Recall* adalah :

$$Precision = \frac{\sum TP}{\sum TP + \sum FP} \quad (7)$$

$$Recall = \frac{\sum TP}{\sum TP + \sum FN} \quad (8)$$

Jika nilai *Precision* mendekati 1 maka berarti rekomendasi yang diberikan kepada pengguna banyak yang digunakan (di beri *rating*) oleh pengguna, sebaliknya jika semakin mendekati 0 maka rekomendasi yang diberikan banyak yang tidak tepat untuk pengguna sistem.

## METODOLOGI

Data untuk penelitian ini diambil dari website resmi *MovieLens (GroupLens)* dengan jumlah *rating* lebih dari sepuluh juta baris dari hampir 70 ribu pengguna yang memberi *rating* pada lebih dari 10 ribu jenis film. Skala *rating* yang digunakan adalah 1-5 dengan asumsi semakin besar nilai *rating* berarti pengguna menyukai film tersebut. Tujuan penelitian kali ini, data *rating* akan dibagi menjadi 5 set, dengan porsi masing-masing set 0,2% - 0,4% - 0,6% - 0,8% - 1%. Data diambil secara acak dari sumber aslinya (10 juta data dari website *MovieLens*). Tujuan pemecahan data tersebut adalah untuk mengevaluasi hasil rekomendasi dan prediksi nantinya.

Secara garis besar, proses dalam penelitian ini sebagai berikut :

1. Data yang sudah dibagi menjadi 5 set tersebut masing-masing akan diproses kedalam algoritma rekomendasi, hasil dari proses ini adalah daftar rekomendasi untuk tiap pengguna yang terdaftar.
2. Kemudian, daftar rekomendasi tersebut akan dihitung nilai prediksinya (lihat Gambar 1). Rumus yang digunakan untuk menghitung prediksi adalah Rumus (4).
3. Untuk mengevaluasi apakah rekomendasi sudah mendekati dengan keinginan pengguna atau tidak, peneliti menggunakan teknik evaluasi *Precision* dan *Recall Value*. *Precision* value menunjukkan seberapa banyak rekomendasi yang “tepat” (terpilih oleh pengguna) dan seberapa banyak yang tidak. Sedangkan *Recall* menunjukkan seberapa banyak film yang sudah diberi *rating* tetapi tidak direkomendasikan kepada pengguna.
4. MAE (*Mean Absolute Error*) digunakan untuk mengukur seberapa banyak rata-rata selisih antara nilai prediksi dengan nilai *rating* yang diberi oleh pengguna. Semakin kecil MAE berarti prediksi yang dihasilkan mendekati akurat.

Dalam penelitian ini, peneliti memiliki tujuan untuk menambah kemampuan *User-Based Collaborative Filtering* dalam memberi rekomendasi dan prediksi yang akurat bagi pengguna, dengan cara :

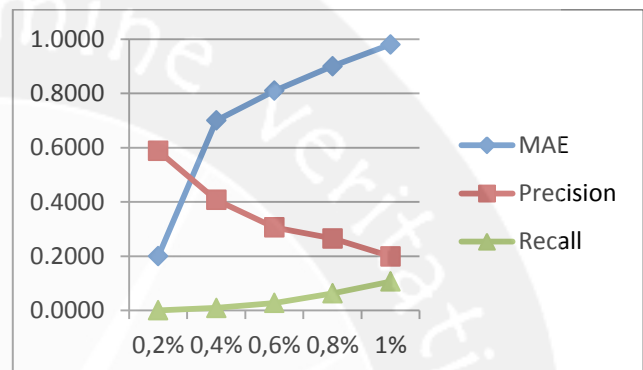
1. Memanfaatkan *Pearson-Correlation* dalam mencari pengguna lain yang terdekat dengan pengguna aktif. Tujuannya supaya pengguna aktif dapat mendapatkan rekomendasi film yang lebih akurat.
2. Menghitung nilai prediksi dengan melibatkan semua pengguna terdekat (*neighbor*) dari pengguna aktif. Tujuannya supaya hasil prediksi dapat mendekati dengan nilai *rating* yang diberi oleh pengguna aktif.
3. Peneliti melibatkan pengguna terdekat (*neighbor*) dalam pencarian rekomendasi dan penghitungan prediksi, supaya mendapatkan hasil rekomendasi dan prediksi yang akurat.

## HASIL

Hasil yang didapat dengan metodologi dan langkah diatas, nilai MAE, *Precision* dan *Recall* dapat dirangkum dalam bentuk tabel sebagai berikut :

**Tabel 2. Hasil Evaluasi UCF Dengan Memanfaatkan *Pearson-Correlation***

Porsi Data	Jumlah Records	MAE	<i>Precision</i>	<i>Recall</i>
0,2%	20.002	0,74	0,5882	0,001
0,4%	40.004	0,7	0,4075	0,0098
0,6%	60.006	0,81	0,3064	0,0279
0,8%	80.008	0,9	0,2656	0,0634
1%	100.010	0,98	0,1993	0,1067



**Gambar 3. Perbandingan Hasil Evaluasi UCF Dengan Memanfaatkan *Pearson-Correlation***

Dari hasil yang disajikan, dapat dilihat bahwa nilai MAE semakin naik seiring dengan bertambahnya jumlah data. Hal ini menunjukkan bahwa kepadatan data mempengaruhi keakuratan prediksi yang diberikan oleh sistem. Dalam kasus ini, prediksi semakin tidak akurat seiring penambahan data.

Sama halnya dengan *Precision* dan *Recall*, nilai *Precision* yang semakin menurun seiring kenaikan jumlah porsi data menunjukkan bahwa rekomendasi yang diberikan oleh sistem banyak yang tidak akurat dengan kata lain banyak *item-item* yang direkomendasikan tidak digunakan oleh para pengguna.

## KESIMPULAN

Dari hasil penelitian yang didapat, peneliti menyimpulkan bahwa dengan memanfaatkan teknik pengukuran *Pearson-Correlation* dalam mencari *neighbors* terdekat pada sistem rekomendasi berbasis *User-Based Collaborative Filtering* tidak menghasilkan prediksi maupun rekomendasi yang akurat.

Untuk tujuan yang sama, yaitu mencari prediksi dan rekomendasi yang akurat, penelitian kali ini dapat dikembangkan lagi dengan cara mengubah teknik pencarian *neighbors* terdekat. Dalam penelitian ini, peran *neighbors* sangat penting, karena mereka adalah variabel dasar dari semua perhitungan prediksi dan rekomendasi yang akan dihasilkan oleh sistem. Untuk itu, dengan mengubah atau memodifikasi teknik pencarian *neighbors* terdekat, peneliti berasumsi hasil rekomendasi dan prediksi yang didapat nantinya dapat lebih baik dari hasil penelitian kali ini.

## DAFTAR PUSTAKA

Beel, J., Langer, S., Genzmehr, M., Gipp, B. and Nürnberger, A. 2013. A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation. *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys)* (2013), 7–14.

Chen, X., Liu, X., Huang, Z., Sun, H., 2010. RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation. *IEEE International Conference on Web Services*, 2010.

Claypool, Mark., Gokhale, A., Murnikov, P., Netes, D., Sartin, M., 1999. Combining Content-Based and Collaborative Filters in an Online Newspaper. *ACM SIGIR Workshop on Recommender Systems – Implementation and Evaluation*.

De Campos, L. M., Fernandez-Luna, J. M., Huete, F. J., Rueda-Morales, M. A., 2010. Combining content-based and collaborative recommendations: *A hybrid approach based on Bayesian networks*. *International Journal of Approximate Reasoning* 51 (2010) 785–799.

Ghauth, K. I., & Abdullah, N. A. 2011. The Effect of Incorporating Good Learners' Ratings in e-Learning Content-based Recommender System. *Educational Technology & Society*, 14 (2), 248–257.

Hu, W., Xie, N., Li, L., Zeng, X., & Maybank, S., 2011. A survey on visual content-based video indexing and retrieval. *IEEE transactions on systems, man, and cybernetics—part c: applications and reviews*, vol. 41, no. 6, november 2011

Jin, R., Si, Luo., Zhai, C., 2006. A Study of Mixture Models Of Collaborative Filtering. *Springer Science + Business Media, LLC; Inf. Retrieval* 9:357-382.

Lops, P., Gemmis, de M., Semeraro, G., 2011. Content-Based Recommender Systems State of the Art and Trends.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. *ACM 1-58113-348-0/01/0005; GroupLens Research Group*.

Shinde, S. K., Kulkarni, U. V., 2011. Hybrid Personalized Recommender System Using Fast K-medoids Clustering Algorithm. *Journal of Advances in Information Technology*, Vol. 2, No. 3, August 2011.