

Rancang Bangun *Movie Recommender System* Dengan Metode *Cluster-Based Smoothing Collaborative Filtering*

Teguh Budianto¹, Luh Kesuma Wardhani²

Jurusan Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. H.R Soebrantas KM. 15 No.155, Panam, Pekanbaru
e-mail: mr.teguhbudianto@yahoo.com¹, luhkesuma@uin-suska.ac.id²

Abstrak

Recommender system pada dasarnya dikembangkan untuk membantu pengguna internet dalam menemukan informasi berupa film, buku, berita atau informasi lainnya yang mungkin akan diminati dan bersifat personal. Pada penelitian ini dibangun sebuah movie recommender system dengan menggunakan metode cluster-based smoothing collaborative filtering yang dapat memberikan prediksi film kepada pengguna dengan akurasi yang baik. Penerapan clustering dan smoothing pada metode ini diharapkan dapat mengatasi masalah kekosongan data rating (sparsity) dan peningkatan jumlah pengguna dalam jumlah besar (scalability) yang kerap kali terjadi pada collaborative filtering. Pengujian metode ini pada sistem yang dibangun menggunakan 3 parameter, yaitu jumlah cluster, jumlah k-neighbors, dan tingkat sparsity. Pada pengujian setiap parameter digunakan dataset rating sebanyak 1063 data contoh yang telah dirating oleh 40 orang pengguna, kemudian data tersebut dibagi menjadi 2 bagian dengan 80% sebagai data latih dan 20% sisanya sebagai data uji. Hasil pengujian menunjukkan bahwa jumlah cluster = 3 dan jumlah k = 50% menghasilkan nilai akurasi paling rendah yaitu 0,6713 dan sistem mampu menangani sparsity data hingga 70% dengan nilai rata-rata MAE = 0,8361.

Kata kunci: *cluster-based smoothing, collaborative filtering, film, recommender system, scalability, sparsity*

Abstract

Recommender systems are basically developed to help internet users to find information such as movies, books, news or other informations that probably preferred personally. In this study we constructed a movie recommender system using cluster-based smoothing method of collaborative filtering that can provide movie predictions for a user with good accuracy. The implementation of clustering and smoothing of this method is expected to address the issue of data gap rating (sparsity) and an increasing number of users in large numbers (scalability) that often occurs in collaborative filtering. The test of this method is using three parameters, namely the number of clusters, the number of k-neighbors, and the level of sparsity. In the test from each parameter is used dataset rating as much as 1063 instance that has been rated by 40 users, then the data is divided into 2 parts, with 80% as data train and 20% as data test. Test results showed that the number of clusters = 3 and number of k = 50% produced the lowest accuracy value is 0.6713 and the system is capable to handling level of data sparsity from up to 70% with MAE = 0.8361.

Keywords: *Cluster-based Smoothing, Collaborative Filtering, movie, recommender system, Scalability, Sparsity*

1. Pendahuluan

Recommender system adalah sebuah sistem *information filtering* untuk mencari dan memberikan rekomendasi *item* berupa informasi, produk, atau layanan kepada *pengguna* berdasarkan prediksi yang bersifat personal [5]. Metode *recommender system* yang paling sukses dikembangkan selama dekade terakhir adalah *Collaborative Filtering* (CF) karena tidak membutuhkan pengetahuan awal tentang sifat-sifat *item* yang ingin dicari [2]. Namun metode ini memiliki beberapa kelemahan, beberapa diantaranya adalah *scalability* yaitu keadaan dimana tingginya jumlah peningkatan *user-item* di dalam *basis data* yang mempengaruhi menurunnya komputasi algoritma CF, sedangkan *sparsity*, yaitu terjadinya kekosongan data matriks *user-item*, yang disebabkan karena pengguna merating dalam jumlah kecil dari jumlah *item* yang tersedia di dalam *basis data*.

Latar Belakang Masalah

Pertumbuhan jumlah situs internet memicu terjadinya ledakan informasi di dunia maya yang menyulitkan pengguna dalam mencari suatu informasi yang diperlukan secara cepat dan relevan. Khusus di bidang film, industri ini bisa memproduksi sekitar 5000 judul film internasional [1] setiap tahunnya.

Kondisi tersebut menyebabkan penikmat film mengalami kesulitan dalam mencari film yang akan ditonton dan sesuai dengan selera masing-masing individu.

Berkaitan dengan itu, sebuah aplikasi dirancang untuk membantu pengguna *internet* menemukan informasi produk dalam hal ini yaitu film, serta layanan dengan membuat rekomendasi yang bersifat personal [5] berdasarkan nilai *rating*, atau ulasan dari pengguna serupa lainnya. Aplikasi ini dikenal sebagai *recommender system* (sistem rekomendasi).

Dalam tulisan ini *recommender system* dianalisis dan dibangun dengan menggunakan metode *cluster-based smoothing collaborative filtering* karena dengan algoritma ini sistem akan memiliki akurasi yang lebih baik untuk merekomendasikan *item* ke pengguna daripada *collaborative filtering* model konvensional [6]. Penerapan *clustering* dan *smoothing* bertujuan untuk menangani dua masalah utama pada *collaborative filtering*, yaitu *scalability* dan *sparsity*.

Tulisan ini membahas rancang bangun *movie recommender system* dan mengukur tingkat akurasi prediksi dengan menggunakan *dataset rating* yang diperoleh dari pengguna yang mendaftar ke dalam perangkat lunak yang dibangun. Akurasi prediksi *rating recommender system* dievaluasi dengan *Mean Absolute Error* (MAE).

Tujuan

Tujuan dari penelitian ini adalah merancang bangun *movie recommender system* dengan metode *cluster-based smoothing collaborative filtering*, serta untuk mengetahui sejauh mana tingkat akurasi dari penerapan metode *cluster-based smoothing collaborative filtering*.

Perumusan Masalah

Beberapa permasalahan yang diselesaikan dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun sebuah *movie recommender system* yang mampu memberikan rekomendasi terbaik kepada pengguna?
2. Bagaimana akurasi dari penerapan teknik *Cluster-based Smoothing* pada hasil rekomendasi berdasarkan nilai MAE (*Mean Absolute Error*).

2. Dasar Teori

Recommender System

Recommender system merupakan sebuah sistem *information filtering* untuk mencari dan memberikan rekomendasi *item* berupa informasi, produk, atau layanan kepada pengguna berdasarkan prediksi yang bersifat personal [5].

Metode Collaborative Filtering

Collaborative filtering (CF) adalah teknik *recommender system* yang paling sukses dan populer saat ini serta penggunaan CF sangat sukses untuk berbagai *recommender system* yang ada di internet. Teknik ini menggunakan teknik statistik untuk menemukan satu set pengguna yang dikenal sebagai *neighbors*, dimana setiap pengguna memiliki kesamaan minat dan pendapat dengan target pengguna (yaitu, mereka memiliki beberapa *rating item* yang sama atau kecenderungan pengguna menyukai *item* yang sama). Setelah lingkungan *neighbors* terbentuk, sistem ini akan menggunakan beberapa algoritma untuk menghasilkan rekomendasi [5].

Cluster-based Smoothing Collaborative Filtering

Cluster-based Smoothing CF merupakan salah satu metode yang dikembangkan untuk menangani masalah *sparsity* dan *scalability* pada *collaborative filtering*. *Cluster-based* CF melakukan proses *clustering* dengan memilih pengguna yang memiliki nilai kemiripan yang sangat tinggi untuk dijadikan ke dalam beberapa kelompok. Proses *clustering* semacam ini memberikan dampak pada proses komputasi yang lebih cepat, karena pemilihan tetangga terdekat oleh pengguna aktif (U_a) tidak dilakukan dengan membandingkan nilai kesamaan ke seluruh data. Kemampuan seperti ini merupakan cara untuk mengatasi masalah *scalability* pada algoritma CF model konvensional. Proses data *smoothing* pada algoritma ini mampu mengurangi tingkat *sparsity* pada sistem CF [6].

Algoritma Clustering

Algoritma yang digunakan pada proses pembentukan *cluster* pengguna adalah algoritma *K-means*. Jumlah k merupakan input yang digunakan algoritma untuk menentukan seberapa banyak jumlah *cluster* yang ingin dibentuk. Langkah-langkah pada *K-means clustering* adalah sebagai berikut [6],[3]:

1. Menentukan jumlah k pengguna yang dijadikan sebagai *centroid*.
2. Bandingkan setiap pengguna yang tidak menjadi *centroid* ke *centroid* terdekat berdasarkan nilai kesamaan.
3. Lakukan perhitungan ulang nilai *centroid* berdasarkan rata-rata nilai kumpulan pengguna di setiap *cluster* yang ada.
4. Proses pembentukan ulang *cluster* dengan nilai *centroid* yang baru dilakukan hingga nilai *centroid* stabil atau mendekati nilai *centroid* sebelumnya.

Nilai kesamaan antara *centroid* dan pengguna lainnya didefinisikan menggunakan *Pearson correlation-coefficient*, adalah sebagai berikut

$$Sim_{u,u'} = \frac{\sum_{t \in T(u) \cap T(u')} (R_u(t) - \bar{R}_u) \cdot (R_{u'}(t) - \bar{R}_{u'})}{\sqrt{\sum_{t \in T(u) \cap T(u')} (R_u(t) - \bar{R}_u)^2} \sqrt{\sum_{t \in T(u) \cap T(u')} (R_{u'}(t) - \bar{R}_{u'})^2}} \quad (1)$$

$Sim_{u,u'}$ merupakan nilai kesamaan antara *centroid* dan pengguna lainnya.

$R_u(t)$ merupakan nilai *rating centroid* untuk *item t*.

$\bar{R}_u(t)$ merupakan rata-rata *rating centroid*.

$R_u(t)$ merupakan nilai *rating pengguna-u* untuk *item t*.

$\bar{R}_u(t)$ merupakan rata-rata *rating pengguna -u*.

$t \in T(u) \cap T(u')$ adalah irisan *item-item* yang dirating *centroid* dan pengguna *u*.

Data Smoothing

Smoothing dilakukan dengan mengisi nilai yang kosong pada dataset dengan *rating* bayangan. Berdasarkan hasil *clustering*, teknik *smoothing* untuk *rating* bayangan didefinisikan sebagai penilaian khusus berikut[6]:

$$R_u(t) = \begin{cases} R_u(t), & \text{jika user } u \text{ merating item } t \\ \hat{R}_u(t), & \text{lainnya} \end{cases} \quad (2)$$

$\hat{R}_u(t)$ merupakan nilai *smoothing* untuk pengguna terhadap *item t*, dimana $\hat{R}_u(t)$ didefinisikan seperti berikut ini:

$$\hat{R}_u(t) = \bar{R}_u + \Delta R_{C_u}(t) \quad (3)$$

$\Delta R_{C_u}(t)$ merupakan rata-rata deviasi untuk semua pengguna di dalam *cluster* C_u untuk *item t*, yang mana didefinisikan sebagai:

$$\Delta R_{C_u}(t) = \frac{\sum_{u' \in C_u(t)} (R_{u'}(t) - \bar{R}_{u'})}{|C_u(t)|} \quad (4)$$

$C_u(t) \in C_u$ adalah kumpulan pengguna yang berada di *cluster* C_u yang telah merating *item t*.

$|C_u(t)|$ merupakan jumlah pengguna di *cluster* C_u yang telah merating *item t*.

Menentukan Cluster untuk Pengguna Aktif

Dalam menentukan beberapa *cluster* yang memiliki nilai kesamaan dengan pengguna aktif dilakukan dengan mengkalkulasi nilai kesamaan antara pengguna aktif dengan kumpulan pengguna di suatu *cluster* menggunakan persamaan berikut [6]:

$$Sim_{u_a,c} = \frac{\sum_{t \in T(u_a) \cap T(c)} \Delta R_C(t) (R_{u_a}(t) - \bar{R}_{u_a})}{\sqrt{\sum_{t \in T(u_a) \cap T(c)} (\Delta R_C(t))^2} \sqrt{\sum_{t \in T(u_a) \cap T(c)} (R_{u_a}(t) - \bar{R}_{u_a})^2}} \quad (5)$$

$Sim_{u_a,c}$ merupakan nilai kesamaan antara pengguna aktif dengan pengguna di suatu *cluster* atau nilai *centroid*.

$\Delta R_C(t)$ merupakan selisih antara *rating centroid* dengan rata-rata *rating centroid* suatu *cluster*.

$R_{u_a}(t)$ merupakan *rating* pengguna aktif untuk *item t*.

\bar{R}_{u_a} merupakan rata-rata *rating item* pengguna aktif.

$t \in T(u_a) \cap T(c)$ merupakan irisan *item-item* yang dirating oleh pengguna aktif dan kumpulan pengguna di dalam *cluster*.

Pencarian Tetangga Terdekat

Proses *smoothing* menyebabkan *rating* pengguna terdiri dari 2 bagian, yaitu *rating* orisinal dan *rating* kelompok. Hal ini berdampak pada proses perhitungan untuk mencari tetangga terdekat bagi pengguna aktif sehingga terdapat nilai untuk variabel w_{ut} yang merupakan *confidential weight* untuk pengguna *u* terhadap *item t*, didefinisikan sebagai berikut [6]:

$$w_{ut} = \begin{cases} 1 - \lambda, & \text{jika pengguna } u \text{ merating item } t \\ \lambda, & \text{lainnya} \end{cases} \quad (6)$$

λ merupakan parameter yang digunakan untuk penyesuaian antara *rating orisinal* dan *rating* kelompok. Nilai λ berkisar antara 0 hingga 1. Dalam pencarian tetangga terdekat digunakan persamaan berikut ini untuk menentukan nilai kesamaan:

$$Sim_{u_a,u} = \frac{\sum_{t \in T(u_a)} w_{ut} \cdot (R_u(t) - \bar{R}_u) \cdot (R_{u_a}(t) - \bar{R}_{u_a})}{\sqrt{\sum_{t \in T(u_a)} w_{ut}^2 \cdot (R_u(t) - \bar{R}_u)^2} \sqrt{\sum_{t \in T(u_a)} (R_{u_a}(t) - \bar{R}_{u_a})^2}} \quad (7)$$

$Sim_{u_a,u}$ merupakan nilai kesamaan antara pengguna aktif dengan pengguna dalam suatu cluster.

Perhitungan Nilai Prediksi

Perhitungan nilai prediksi penelitian ini menggunakan persamaan sebagai berikut [6],[3]:

$$R_{u_a}(t) = \overline{R_{u_a}} + \frac{\sum_{i=1}^K w_{ut} \cdot sim_{u_a,u} \cdot (R_u(t) - \overline{R_u})}{\sum_{i=1}^K w_{ut} \cdot sim_{u_a,u}} \quad (8)$$

K merupakan jumlah pengguna yang menjadi tetangga terdekat pengguna aktif

3. Metode Penelitian

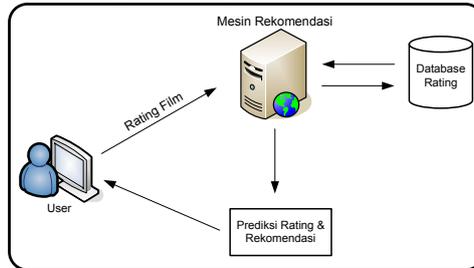
Metode penelitian yang dilakukan adalah sebagai berikut:

1. Studi literatur, dengan mencari informasi dan data yang berkaitan dengan *recommender system* menggunakan metode *cluster-based smoothing collaborative filtering*.
2. Analisa dan perancangan *recommender system*.
3. Pengujian dan analisis berdasarkan parameter-parameter yang dapat mempengaruhi kualitas prediksi dari *recommender system*.
4. Kesimpulan yang didapat dari penelitian ini.

4. Perancangan

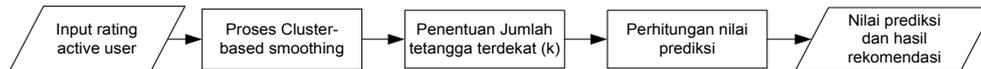
Deskripsi Umum Sistem

Movie Recommender System yang dibangun merupakan sebuah aplikasi berbasis web yang bertujuan untuk memberikan rekomendasi film berdasarkan nilai *rating* yang mungkin akan disukai oleh pengguna. Sistem ini juga memfasilitasi pengguna untuk memberikan *rating* tertentu kepada film yang ada di dalamnya. Gambaran umum sistem dapat dilihat pada Gambar 1.



Gambar 1. Gambaran Umum Sistem

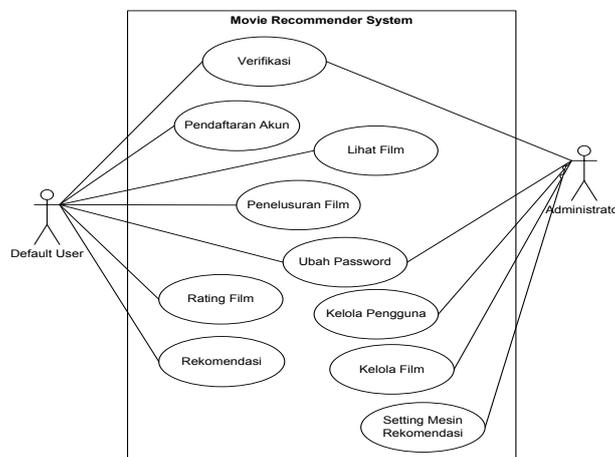
Diagram penerapan metode *Cluster-based Collaborative Filtering* dapat dilihat pada Gambar 2.



Gambar 2. Penerapan *Cluster-based Smoothing Collaborative Filtering*

Perancangan Sistem

Perancangan perangkat lunak ini dibangun dengan konsep pendekatan berorientasi objek, yaitu dengan *Unified Modeling Language (UML)*. Salah satu diagram yang dihasilkan adalah diagram *use case* pada Gambar 3.



Gambar 3. *Use Case Diagram* dari Sistem Rekomendasi

Pada penelitian ini *recommender system* dibangun dengan 2 bentuk hak akses pengguna, yaitu administrator dan pengguna biasa (*default user*).

5. Analisis dan Pengujian

Dataset

Pengujian tingkat akurasi dilakukan dengan mengambil contoh data sebanyak 1.063 *rating* dari total *rating* yang sudah dikumpulkan yaitu berjumlah 1.403 data *rating*. Seluruh data *rating* dikumpulkan dari pengguna yang mendaftar secara langsung ke dalam perangkat lunak yang dibangun. Skala penilaian yang digunakan berdasarkan *rating* yang sama dengan *movielens.umn.edu*, dimana *rating* 1 untuk film sangat buruk, *rating* 2 untuk film cukup buruk, *rating* 3 untuk film yang biasa saja, *rating* 4 berarti film baik, dan terakhir *rating* 5 berarti film sangat baik. Data yang dijadikan contoh dalam penelitian ini harus memenuhi kriteria yaitu data *rating* berasal dari pengguna yang telah merating minimal sebanyak 20 film dengan batas maksimal *rating* setiap pengguna adalah 30 film.

Pengujian

Tahap pertama dari pengujian tingkat akurasi ialah dengan membagi dua *dataset* yang ada menjadi 80% untuk data latih dan 20% sisanya sebagai data uji. Pengujian tingkat akurasi dihitung berdasarkan 3 parameter uji yaitu berdasarkan jumlah *cluster*, jumlah *k neighbors*, dan tingkat *sparsity*. Pada setiap parameter uji tersebut akan dilakukan percobaan sebanyak 5 kali.

Setelah dilakukan pengujian dengan beberapa parameter, maka hasil pengujian untuk setiap parameter tingkat akurasi diperoleh nilai MAE sebagai berikut:

1. Berdasarkan Jumlah *Cluster*.

Tabel 1. MAE Berdasarkan Jumlah *Cluster*

Jumlah <i>Cluster</i>	Percobaan ke-					Rata-rata
	I	II	III	IV	V	
2	0,6969	0,7088	0,7192	0,6634	0,6716	0,6919
3	0,6519	0,6894	0,6994	0,6429	0,6732	0,6713
4	0,7247	0,7601	0,7237	0,9584	0,7436	0,7821

Tabel 1 menunjukkan hasil pengujian pada *dataset* untuk jumlah *cluster* yang telah ditentukan sebanyak 2, 3 dan 4 *cluster*. Pada pengujian ini diketahui bahwa iterasi ke-4 dengan jumlah *cluster* = 3 merupakan pengujian dengan nilai MAE terendah yaitu 0,6429, sedangkan pada iterasi ke-4 dengan jumlah *cluster* = 4 tercatat sebagai pengujian dengan nilai MAE tertinggi.

Dari rata-rata perhitungan MAE ketiga jumlah *cluster* ini ditemukan jumlah *cluster* = 3 merupakan penghasil nilai MAE yang paling kecil yaitu 0,6713 dan nilai MAE yang dihasilkannya dari 5 kali pengujian cenderung stabil. Pada pengujian jumlah *cluster* = 4 yang membagi *dataset* menjadi 4 *cluster*, dimana masing-masing *cluster* berisi data yang lebih sedikit dari *cluster* lainnya menyebabkan tingkat akurasi tidak stabil dikarenakan pilihan untuk tetangga terdekat pengguna aktif terlalu sedikit. Pada pengujian ini, jumlah seluruh pengguna yang akan dicarikan *cluster*nya adalah sebanyak 40 pengguna. Jumlah anggota dari tiap-tiap *cluster* cenderung selalu berbeda, yang menyebabkan ada kalanya seorang pengguna aktif masuk ke dalam sebuah *cluster* dengan jumlah anggota yang sedikit. Misalnya pengguna aktif masuk menjadi anggota dari *cluster*-x dengan total anggota hanya sebanyak 5 pengguna. Logikanya adalah, jika anggota suatu *cluster* hanya terdapat 5 pengguna maka pada proses pemilihan jumlah tetangga terdekat pengguna aktif hanya ada kandidat di antara 5 pengguna tersebut.

Kesimpulan dari pengujian jumlah *cluster* adalah pemilihan jumlah *cluster* pada penerapan *collaborative filtering* berbasis *cluster-based smoothing* perlu memperhatikan antara jumlah yang akan diclusterisasi dengan jumlah maksimum *cluster* yang akan dibuat. Sehingga proses setelahnya yaitu mulai dari pengisian *rating* bayangan, pemilihan tetangga terdekat pengguna aktif dan proses prediksi film mendapat keakuratan yang baik dengan nilai MAE mendekati 0.

2. Berdasarkan Jumlah *K-Neighbors*.

Tabel 2. MAE Berdasarkan Jumlah *K-Neighbors*

Jumlah K	Percobaan ke-					Rata-rata
	I	II	III	IV	V	
30%	0,6506	0,7496	0,7424	0,7520	0,7247	0,7239
50%	0,6519	0,6894	0,6994	0,6429	0,6732	0,6713
70%	0,7270	0,6790	0,7546	0,7034	0,7482	0,7224

Tabel 2. menunjukkan hasil pengujian tingkat akurasi dengan parameter jumlah *k* di set sebanyak 30%, 50%, dan 70%. Terlihat bahwa rata-rata dari pengujian tersebut bernilai stabil sejak nilai nilai *k* >

30%. Hal ini dapat terjadi karena terdapat banyak kandidat untuk tetangga terdekat pengguna aktif. Penyebab dari banyaknya kandidat tersebut dikarenakan pengujian ini mengasumsikan jika nilai kesamaan antara pengguna aktif dengan kandidat tetangga terdekat adalah 0 maka kandidat tetangga terdekat ini akan diambil sebagai tetangga terdekat dengan syarat jumlah k maksimum belum terpenuhi. Pada dasarnya nilai kesamaan < 0 mengindikasikan bahwa pengguna tersebut memiliki nilai kesamaan yang kecil terhadap pengguna aktif.

Perubahan yang tidak terlalu drastis antara ketiga pengujian jumlah k juga dikarenakan proses menghasilkan prediksi dari metode ini menggunakan nilai *dataset rating* yang telah diisi oleh *rating* bayangan. Sehingga pada dasarnya, sejak pelatihan data dilakukan, tidak ada nilai *rating* film = 0 yang dimasukkan kedalam perhitungan prediksi.

3. Berdasarkan Tingkat *Sparsity*

Tabel 3. MAE Berdasarkan Tingkat *Sparsity*

Tingkat <i>Sparsity</i>	Percobaan ke-					Rata-rata
	I	II	III	IV	V	
30%	0,7540	0,7687	0,7802	0,7900	0,7696	0,7725
50%	0,7991	0,7722	0,7678	0,7523	0,7730	0,7728
70%	0,8552	0,8956	0,7864	0,8191	0,8244	0,8361

Dari Tabel 3 pengujian tingkat *sparsity* menunjukkan adanya peningkatan nilai MAE ketika tingkat kekosongan data dinaikan. Peningkatan nilai ini terjadi karena ketika kondisi data dalam keadaan *sparse* yang tinggi, menyebabkan berkurangnya data *rating* yang akan dijadikan pertimbangan untuk pemberian prediksi. Penerapan teknik *smoothing* pada keadaan ini juga menyebabkan data yang digunakan untuk melakukan proses prediksi lebih banyak diisi oleh mesin rekomendasi daripada jumlah *rating* sebenarnya yang dirating langsung oleh pengguna. Meskipun demikian, masalah kekosongan data pada *collaborative filtering* dapat teratasi dengan menerapkan teknik *smoothing* pada sistem yang dibangun.

Ada kalanya nilai kesamaan antara pengguna aktif dan pengguna lain di dalam suatu *cluster* menemui angka 0, yang berpengaruh kepada peningkatan nilai MAE. Meskipun demikian, pengujian tingkat akurasi dengan *sparsity* data sebanyak 70% dapat menghasilkan nilai MAE = 0,8361. Dapat disimpulkan bahwa sistem yang dibangun mampu menangani kekosongan data hingga 70% total data *rating* yang ada.

Merujuk pada hasil penelitian lain yang menggunakan metode serupa [6] pada *dataset* MovieLens dan EachMovie masing-masing menghasilkan nilai MAE terendah yaitu 0,778 dan 0,956. Dengan demikian tingkat akurasi dari penelitian ini tidaklah terlalu buruk meskipun menggunakan *dataset* yang berbeda.

6. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Dapat dibangun sebuah perangkat lunak *Movie Recommender System* dengan menerapkan metode *Cluster-based Smoothing Collaborative Filtering* dengan pengaturan jumlah *cluster* sebanyak 3, dan jumlah *k-nearest neighbors* sebanyak 50%.
2. Hasil akurasi yang dicapai pada pengujian dengan metode ini tergolong akurat karena dengan menerapkan jumlah *cluster* = 3 dan jumlah $k=50\%$ dapat menghasilkan rata-rata MAE = 0,6713.
3. *Movie recommender system* berbasis *cluster-based smoothing collaborative filtering* yang dibangun mampu mengatasi masalah kekosongan data dengan tingkat *sparsity* sebanyak 70% yang menghasilkan nilai MAE = 0,8361.
4. Semakin tinggi tingkat kekosongan data, maka semakin tinggi pula nilai MAE yang dihasilkan. Hal ini dikarenakan berkurangnya data *rating* pengguna yang dijadikan pertimbangan dalam membuat prediksi.
5. Pengisian *rating* bayangan pada proses *smoothing* menyebabkan tidak adanya nilai *rating* = 0 yang dimasukkan ke dalam proses perhitungan prediksi.

7. Saran

Pada penerapan *collaborative filtering* berbasis *cluster-based smoothing* perlu memperhatikan antara jumlah data yang akan di-*cluster* dengan jumlah maksimum *cluster* yang akan dibuat agar menghasilkan prediksi film mendapat keakuratan yang baik dengan nilai MAE mendekati 0.

Daftar Pustaka

- [1] Hakim, Irfan Aris Nur. *Sistem Rekomendasi Film Berbasis Web Menggunakan Metode Item-based Collaborative Filtering*. Skripsi. Universitas Pendidikan Indonesia. Bandung. 2010.
- [2] Herlocker, Jonathon L., Konstan, Joseph A., Terveen, Loren G., Riedl, John T. *Evaluating Collaborative Filtering Recommender Systems*, ACM Transactions on Information Systems, Vol. 22, No. 1, Hal 5-53. 2004.
- [3] Sanjung, Ariyani. *Perbandingan Semantic Classification dan Cluster-based Smoothed pada Recommender System berbasis Collaborative Filtering*. Skripsi. Institut Teknologi Telkom. Bandung. 2010.

-
- [4] Sarwar, Badrul M., Karypis, George., Konstan, Joseph., Riedl, John. *Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering*. In Proceedings of the Fifth International Conference on Computer and Information Technology. 2001.
 - [5] Sarwar, Badrul., Karypis, George., Konstan, Joseph., Riedl, John. *Item-based Collaborative Filtering Recommendation Algorithms*. Proceedings of the 10th international conference on World Wide Web, ISBN:1-58113-348-0. 2001.
 - [6] Xue, Gui-Rong., Lin, Chenxi., Yang, Qiang., Xi, WenSi., Zeng, Hua-Jun., Yong, Yu., Chen, Zheng. *Scalable Collaborative Filtering Using Cluster-based Smoothing*. In: Proceedings of the 2005 ACM SIGIR Conference, Pages 114 – 121. Salvador, Brazil. 2005.