Analisis dan Implementasi Imputation-Boosted Neighborhood-Based Collaborative Filtering Menggunakan Genre Film

Analysis and Implementation of Imputation-Boosted Neighborhood-Based Collaborative Filtering Using Film Genre

Muhammad Hanif Oktavianto[1], Agung Toto Wibowo, ST., MT.[2], Rita Rismala, ST., MT.[3]

[1,2,3]Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom

[1]**hanifoktav@gmail.com**, [2] **atwbox@gmail.com**, [3]**ritaris@telkomuniversity.ac.id**

Abstrak

Sistem rekomendasi adalah sebuah sistem yang mampu memberikan rekomendasi sejumlah *item* kepada *user* dengan memprediksi *rating* terhadap *item* berdasarkan minat *user*. *Neighborhood-based collaborative filtering* adalah salah satu metode pada Sistem Rekomendasi untuk melakukan perhitungan prediksi *rating*. Akan tetapi, *neighborhood-based collaborative filtering* tidak mampu memberikan prediksi *rating* yang akurat ketika data *rating* yang ada bersifat *sparse* atau memiliki banyak kekosongan. Kekosongan data mengakibatkan perhitungan *similarity* antar *user* atau *item* menjadi kurang tepat, yang berakibat pada pemilihan *neighbor* dan perhitungan prediksi yang tidak tepat pula. Salah satu solusi adalah melakukan imputasi yaitu proses pengisian awal terhadap data dengan metode tertentu. Dengan memanfaatkan *feature item* berupa genre, dilakukan imputasi terhadap data untuk selanjutnya digunakan oleh *neighborhood-based collaborative filtering*. Penelitian ini berfokus pada penerapan proses imputasi terhadap *neighborhood-based collaborative filtering* dan menganalisis pengaruhnya terhadap performansi. Hasil yang diperoleh adalah proses imputasi meningkatkan performansi akurasi prediksi *rating* pada dataset dengan *sparsity* 85%, dan peningkatan performansi yang terukur menjadi semakin besar seiring semakin *sparse* dataset yang ada.

Kata kunci : ***collaborative filtering, data imputation, neighborhood-based collaborative filtering***

Abstract

Recommender system is a system which can give recommendation of a number of items to the user by predicting the ratings of the items according to the user's interest. *Neighborhood-based collaborative filtering* is one the methods used by recommender systems to calculate the rating prediction. But, *neighborhood-based collaborative filtering* is unable to produce accurate rating predictions when the available rating data is sparse, or has many empty values. The sparsity of the data causes the computation for similarity of user or item to be inaccurate, which causes the rating predictions scores also to be inaccurate. One of the solution is to do imputation, which is a process of filling the data using certain methods. By using item feature in the form of genre, the data is getting imputed, which is then used by *neighborhood-based collaborative filtering*. This research focuses on the application of imputation process on *neighborhood-based collaborative filtering* and the analysis of the effect on the performance. The result is that the imputation process increases accuration performance of rating prediction using the dataset with sparsity level of 85%, and the calculated increment of performance becomes larger as the dataset becomes more sparse.

Keywords: ***collaborative filtering, data imputation, neighborhood-based collaborative filtering***

1. Pendahuluan

Seiring perkembangan teknologi, kemudahan akses terhadap informasi tidak hanya mendatangkan manfaat, tetapi juga menyulitkan pengguna dalam menyaring informasi yang dia butuhkan dari yang tidak dibutuhkan. Permasalahan tersebut mendorong terciptanya *recommender system* atau sistem rekomendasi, yaitu sistem yang berfungsi untuk memberikan rekomendasi akan suatu hal yang sesuai dengan kebutuhan pengguna. Salah satu

metode yang digunakan oleh sistem rekomendasi dalam membuat rekomendasi adalah *collaborative filtering*, yang memberikan prediksi berdasarkan pola rating dari user [2].

Dalam memberikan prediksi rating yang akurat, *collaborative filtering* memiliki beberapa tantangan. Salah satu masalah utama adalah *data sparsity*, yaitu ketika data yang dimiliki memiliki banyak kekosongan sehingga mengakibatkan *collaborative filtering* tidak mampu memberi prediksi yang akurat [10]. Penanganan masalah sparsity salah satunya adalah dengan menggunakan informasi *item* [13]. Pengisian data yang kosong yang disebut imputasi juga membantu menghadapi masalah *sparsity*, seperti yang ditunjukkan pada [13].

Fokus pada penelitian ini adalah mengimplementasikan metode Imputation-boosted Neighborhood-based Collaborative filtering, dan menganalisis performansi prediksi yang dihasilkan.

2. Landasan Teori

2.1 Neighborhood-based Collaborative Filtering

Neighborhood-based collaborative filtering (NBCF) adalah metode collaborative filtering dimana sejumlah user dipilih untuk menjadi acuan dalam menghitung prediksi rating [3]. Dengan menghitung *similarity* antar user, akan dipilih beberapa *user* yang paling mirip pola *rating*-nya sehingga perhitungan prediksi menjadi lebih akurat., *Metode* ini memiliki tingkat akurasi yang baik, namun rentan terhadap data *sparsity* yang dapat mengakibatkan perhitungan *similarity* menjadi tidak tepat.

Dalam menghitung prediksi nilai rating, NBCF menggunakan *weighted sum of deviation* yang merupakan perhitungan berdasarkan simpangan nilai rating terhadap rata rata, menggunakan persamaan (1) berikut.

$$r_{11} = \frac{\sum_{j \in N} s(i, 1)(r_{j1} - \bar{r}_j)}{\sum_{j \in N} |s(i, 1)|} \quad (1)$$

Dimana,

r_{11} adalah nilai rating yang akan diprediksi,
$s(i, 1)$ adalah bobot *similarity* antar *user* dengan *user* aktif,
r_{j1} adalah rating dari user lain terhadap *item* aktif,
$\bar{r}_j$ adalah rata-rata *rating user* tersebut,
N adalah kumpulan user yang termasuk pada *neighbor*.

Bobot *similarity* dari NBCF adalah ukuran kemiripan pola seorang user dengan user aktif. Bobot tersebut dihitung menggunakan Pearson Correlation Coefficient [4], dengan persamaan (2) berikut.

$$s_{ab} = \frac{\sum_j (r_{aj} - \bar{r}_a)(r_{bj} - \bar{r}_b)}{\sqrt{\sum_j (r_{aj} - \bar{r}_a)^2 \sum_j (r_{bj} - \bar{r}_b)^2}} \quad (2)$$

Dimana,

r_{aj} adalah rating milik user a terhadap item j,
$\bar{r}_a$ adalah rating rata-rata dari user a.

Perhitungan tersebut dilakukan untuk kumpulan item yang telah diberi *rating* oleh kedua user.

2.2 Imputasi

Imputasi adalah proses pengisian nilai kosong terhadap data menggunakan metode tertentu [9]. Salah satu metode dasar dalam melakukan pengisian data adalah menggunakan mean. Nilai mean menunjukkan kecenderungan besarnya suatu nilai dalam kumpulan data, dan dapat digunakan untuk menggantikan kekosongan suatu nilai pada

kumpulan data seperti pada [9]. Penggunaan mean pada Imputation-boosted NBCF adalah dengan membentuk matriks preferensi yang berisi rating user terhadap genre. Nilai pada matriks preferensi merupakan nilai mean dari kumpulan rating user terhadap genre tertentu, menggunakan persamaan (3).

$$\text{[illegible]} = \frac{\sum_{\text{[illegible]}} r_{\text{[illegible]}}}{\|I_{\text{[illegible]}}\|} \quad (3)$$

Dimana,

[illegible] adalah nilai rating dari user u terhadap genre g,

$r_{\text{[illegible]}}$ adalah rating milik user u terhadap item i,

$I_{\text{[illegible]}}$ adalah kumpulan item yang memiliki genre g.

Nilai imputasi mean yang akan digunakan pada pengisian rating sebuah item adalah nilai mean dari rating user tersebut terhadap genre-genre yang dimiliki item tersebut. Perhitungan imputasi mean terdapat pada persamaan (4) berikut.

$$\text{[illegible]} = \frac{\sum_{\text{[illegible]}} r_{\text{[illegible]}}}{\|\text{[illegible]}\|} \quad (4)$$

Dimana,

$i_{\text{[illegible]}}$ adalah nilai imputasi mean dari user u terhadap item i,

$r_{\text{[illegible]}}$ adalah rating milik user u terhadap genre g,

[illegible] adalah kumpulan genre yang dimiliki item i.

Sebuah metode lain untuk imputasi juga digunakan, yaitu modus. Modus menunjukkan kecenderungan dari kumpulan data yang berbentuk diskrit. Pada dasarnya, bentuk rating pada *collaborative filtering* bermacam-macam. Untuk dataset Movielens, data yang ada berbentuk numerik yang diskrit, sehingga metode modus juga dapat digunakan untuk imputasi. Terdapat nilai [illegible] yang menunjukkan nilai imputasi modus untuk rating user u pada item i. Nilai tersebut adalah modus dari kumpulan rating terhadap item-item yang setidaknya memiliki satu buah kesamaan genre dengan item i.

Selanjutnya, nilai imputasi yang digunakan adalah perpaduan kedua imputasi, menggunakan sebuah parameter bobot berupa α dengan nilai 0-1. Parameter tersebut menunjukkan besarnya bobot imputasi mean, dan berbanding terbalik dengan bobot imputasi modus. Matriks rating yang padat (dense) akan terbentuk dengan mengisikan nilai rating yang kosong menggunakan persamaan (5).

$$\text{[illegible]} = \alpha * \text{[illegible]} + (1 - \text{[illegible]}) * \text{[illegible]} \quad (5)$$

Dimana,

$i_{\text{[illegible]}}$ adalah nilai imputasi terhadap rating user u untuk item i,

$i_{\text{[illegible]}}$ adalah nilai imputasi mean dari user u terhadap item i,

$i_{\text{[illegible]}}$ adalah nilai imputasi modus dari user u terhadap item i.

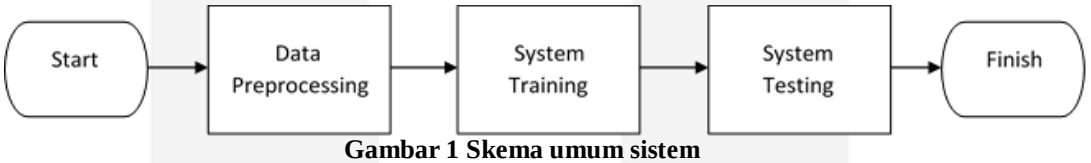
2.3 Perancangan Sistem

Secara umum, sistem terbagi menjadi tiga tahapan utama seperti pada gambar 1, yaitu Preprocessing, Training, dan Testing. Tahap preprocessing mengubah bentuk dataset awal agar dapat diolah lebih lanjut oleh sistem. Tahap training melatih sistem sehingga sistem dapat melakukan penghitungan prediksi. Tahap testing melakukan perhitungan prediksi untuk dapat dianalisis performansi sistem dalam hal akurasi prediksi yang diberikan. Terdapat dua sistem yang dibuat, yaitu yang menggunakan imputasi (*Imputed*) dan yang tidak menggunakan (*Base*), sebagai perbandingan.

Data awal yang digunakan adalah dataset Movielens 100K [5]. Dataset terdiri dari data rating user terhadap item, dan data atribut item, yaitu genre. Jumlah data rating sebanyak 100.000, terdiri dari 943 user dan 1682 item. Dataset awal memiliki kepadatan data sebesar 6%, atau tingkat sparsity sebesar 94%. Untuk mendapatkan dataset yang lebih padat, dilakukan penghilangan terhadap sejumlah item, yaitu dengan mengikutsertakan hanya 209 buah (12,4%) item terpadat saja. Dari langkah tersebut, didapatkan subset 1 berupa 64.619 buah rating dari 943 user terhadap 209 item, dengan kepadatan 25% (75% *sparsity*). Dari subset 1, dibentuk pula subset 2, 3, 4, dan 5 yang masing-masing memiliki kepadatan 20%, 15%, 10%, dan 5% dengan cara menghilangkan sejumlah data rating secara random dari subset 1. Selanjutnya dilakukan pemisahan berupa 80% training dan 20% testing untuk setiap subset.

Setelah tahap preprocessing, dilakukan proses training, yaitu melatih sistem agar dapat melakukan perhitungan prediksi. Proses ini terdiri dari penghitungan nilai imputasi dan penghitungan *similarity*. Pada Neighborhood-based collaborative filtering, output dari tahap training meliputi matriks *similarity* antar user, berukuran 943x943, dan matriks daftar urutan neighbor berukuran 943x942. Matriks *similarity* dihitung dengan Pearson dari persamaan (2) dengan memanfaatkan nilai imputasi. Jika pada penghitungan *similarity* dua user, terdapat item yang hanya dirating oleh salah satu saja, maka nilai dari user yang belum memberi rating akan menggunakan imputasi. Pada item yang tidak diberi rating oleh keduanya, item tersebut diabaikan. Setelah didapatkan matriks *similarity*, matriks daftar neighbor juga bisa didapat dengan mengurutkan berdasarkan besarnya nilai *similarity*. Matriks daftar neighbor akan terdiri dari 943 buah vektor, masing-masing berisi 942 buah id user yang terurut dari yang terbesar secara *similarity*.

Pada tahap testing, dilakukan perhitungan prediksi terhadap rating dari data testing menggunakan beberapa komponen hasil output tahap training. Menggunakan persamaan (1), neighbor N menggunakan matriks daftar neighbor dengan ukuran berupa parameter k. Nilai imputasi tidak digunakan pada tahap ini, sehingga user yang tidak memberi rating pada item aktif akan diabaikan dari perhitungan prediksi. Nilai bobot (1) menggunakan matriks *similarity*. Hasil prediksi rating tersebut kemudian dihitung MAE nya dengan acuan berupa nilai rating asli pada data testing.



Gambar 1 Skema umum sistem

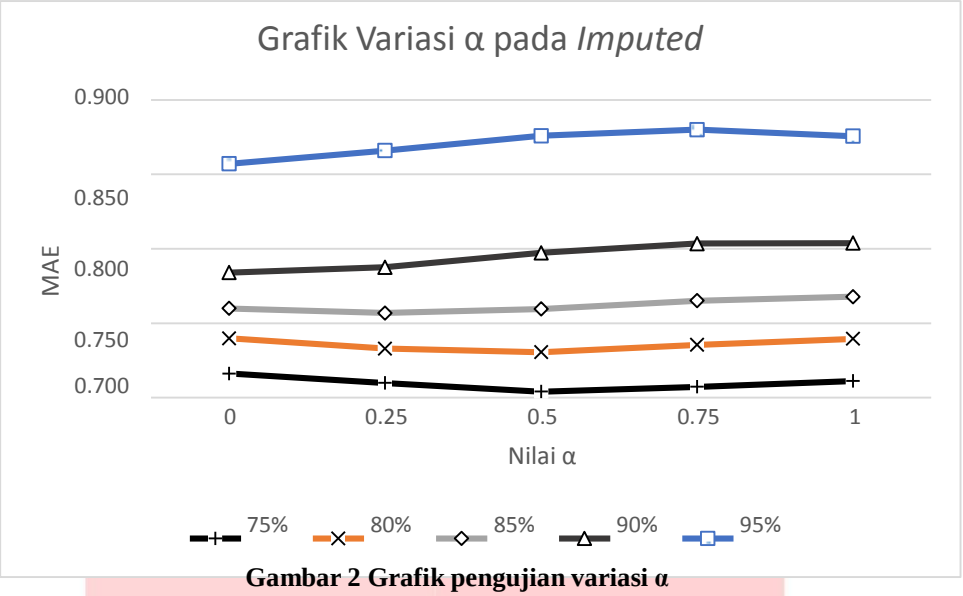
3. Hasil Pengujian dan Kesimpulan

Terdapat dua buah parameter sistem yang diujikan, yaitu α dan *k*. Parameter α adalah parameter pada proses imputasi, mengatur bobot antara penggunaan imputasi mean dan modus pada nilai imputasi. Parameter *k* adalah parameter pada proses penghitungan prediksi, mengatur ukuran neighbor yang diikutkan saat menghitung prediksi rating. Selain itu, sistem juga diujikan terhadap sejumlah subset data dengan tingkat sparsity yang berbeda, untuk mengetahui performansi sistem untuk masing-masing variasi sparsity.

3.1 Parameter α

Untuk parameter α, hanya digunakan sistem *Imputed* karena proses imputasi tidak dilakukan pada sistem *Base*. Dengan nilai α yang divariasikan, pengujian dilakukan menggunakan nilai *k* sebesar 60, seperti pada [14]. Nilai α yang diujikan adalah antara 0 – 1, dengan peningkatan sebesar 0,25. Terlihat pada gambar 2, pola tren performansi tiap variasi *sparsity* berbeda. Pada tabel 1 terlihat bahwa posisi pembobotan optimal ada pada nilai 0,5 untuk data yang 75% *sparse*, dan nilai yang optimal cenderung bergerak ke arah nilai 0 (murni modus), pada data 90% dan 95% *sparse*. Melihat subset data 1 - 3 untuk performansi α = 1 dan α = 0, ada indikasi bahwa semakin padat data maka imputasi mean akan lebih baik dari modus dan bobot optimal dapat bergeser ke arah nilai 1 (murni mean). Nilai α 0,25 memiliki MAE rata-rata yang sama dengan nilai α 0, tetapi juga menjadi titik tengah antara posisi optimal subset 1-2 dan posisi subset 4-5. Oleh karena itu, selanjutnya nilai 0,25 digunakan sebagai nilai optimal di skenario berikutnya.

Fungsi modus menunjukkan hasil yang lebih baik dari fungsi mean, terutama untuk data yang *sparse* (subset 4 dan 5). Nilai mean dari suatu kumpulan data cenderung lebih terpengaruh oleh data pencilan, dibandingkan nilai modus, dan efeknya semakin terlihat ketika jumlah data relatif sedikit. Hal ini menyebabkan pergeseran posisi α yang optimal, yang cenderung mengarah ke nilai 0 (murni modus) seiring meningkatnya *sparsity* data.



Gambar 2 Grafik pengujian variasi α

Tabel 1 Hasil pengujian variasi α

Subset \ α	0	0,25	0,5	0,75	1
1	0.716	0.710	**0.704**	0.707	0.711
2	0.740	0.733	**0.731**	0.735	0.740
3	0.760	**0.757**	0.760	0.765	0.768
4	**0.784**	0.788	0.797	0.804	0.804
5	**0.857**	0.866	0.876	0.880	0.876
Avg	**0.771**	**0.771**	0.773	0.778	0.780

3.2 Parameter *k* pada *Imputed*

Dengan melakukan pengujian terhadap variasi nilai *k*, dapat diketahui ukuran neighbor yang optimal untuk kedua sistem, *Imputed* dan *Base*. Pada pengujian terhadap *Imputed*, nilai α yang digunakan adalah 0,5 yang didapat dari pengujian sebelumnya.

Pada sistem *Imputed*, seiring peningkatan nilai *k*, terjadi penurunan MAE dengan efek yang makin mengecil. Secara umum, nilai *k* 70 adalah posisi terakhir terjadinya penurunan MAE > 0,001. Tabel 2 menunjukkan bahwa penurunan lebih lanjut dari 70, bahkan hingga ukuran *k* lebih dari dua kali lipat, hanya menurunkan sebesar 0,002. Selain itu, untuk beberapa subset, terdapat titik tertentu dimana MAE kembali naik. Nilai 70 digunakan sebagai *k* optimal untuk sistem *Imputed*.

Tabel 2 Hasil pengujian *k* untuk *Imputed*

k	Subset 1	2	3	4	5
10	0.731	0.758	0.791	0.826	0.903
20	0.714	0.742	0.770	0.804	0.878
30	0.709	0.736	0.761	0.796	0.871
40	0.707	0.734	0.757	0.793	0.869
50	**0.706**	0.733	**0.755**	0.788	**0.866**
60	0.706	**0.731**	0.755	0.787	0.866
70	0.706	0.730	0.754	**0.785**	0.865
80	0.706	0.731	0.754	0.784	0.865
90	0.706	0.730	0.753	0.784	0.865
100	0.706	0.730	0.753	0.784	0.865
110	0.707	0.730	0.753	0.783	0.865
120	0.707	0.730	0.752	0.783	0.865
130	0.707	0.729	0.752	0.783	0.865
140	0.707	0.729	0.752	0.783	0.865
150	0.707	0.729	0.752	0.783	0.865

3.3 Parameter *k* pada *Base*

Pada sistem *Base*, terdapat pola yang mirip dengan hasil sebelumnya. Nilai optimal tercapai pada *k* sebesar 50. Peningkatan lebih jauh lagi hanya memberi penurunan MAE 0,001, bahkan mengalami peningkatan pada titik tertentu. Terlihat pada tabel 3, untuk subset 4 dan 5, nilai optimal telah tercapai sebelum 50, tetapi pada titik tersebut belum/tidak mengalami peningkatan kembali, sehingga nilai 50 digunakan sebagai nilai *k* optimal untuk *Base*.

Tabel 3 Hasil pengujian *k* untuk *Base*

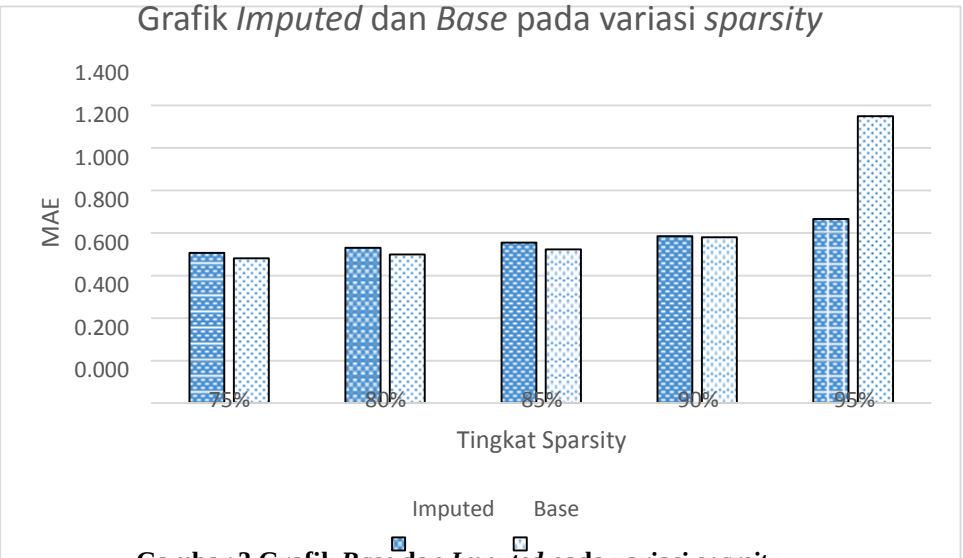
k	Subset 1	2	3	4	5
10	0.729	0.745	0.764	0.808	1.349
20	0.698	0.714	0.735	0.785	**1.348**
30	0.688	0.706	0.726	0.781	1.348
40	0.683	0.701	0.724	**0.780**	1.348
50	**0.681**	**0.699**	**0.723**	0.781	1.348
60	0.680	0.699	0.723	0.781	1.348
70	0.679	0.699	0.723	0.781	1.348
80	0.679	0.699	0.723	0.781	1.348
90	0.679	0.700	0.724	0.781	1.348
100	0.679	0.700	0.724	0.780	1.348
110	0.679	0.701	0.724	0.780	1.348
120	0.679	0.700	0.724	0.780	1.348
130	0.680	0.701	0.724	0.780	1.348
140	0.680	0.701	0.724	0.780	1.348
150	0.681	0.701	0.724	0.780	1.348

3.4 Perbandingan performansi

Menggunakan konfigurasi terbaik pada masing-masing sistem menggunakan hasil pengujian sebelumnya, dapat dibandingkan performansi *Imputed* dan *Base* secara umum dengan mengujikannya terhadap sejumlah subset data dengan variasi *sparsity*. Sistem *Imputed* menggunakan *k* sebesar 60 dan α 0,5, sedangkan *Base* menggunakan *k* sebesar 50.

Secara umum, grafik pada gambar 3 menunjukkan bahwa kedua sistem mengalami peningkatan MAE (penurunan performansi) seiring meningkatnya *sparsity* data. Pada subset 1, 2, dan 3, masing-masing dengan *sparsity* 75%, 80%, dan 85%, *Imputed* memiliki MAE lebih tinggi dari *Base*. Selanjutnya, pada subset 4 dengan *sparsity* 85%, sistem memiliki performansi yang relatif sama (selisih MAE <1%), dan pada subset 5 (95%), *Imputed* memiliki MAE lebih rendah dari *Base*.

Seperti yang ditunjukkan pada tabel 4, tiap peningkatan *sparsity* data, *Imputed* mengalami peningkatan MAE yang lebih stabil dibanding peningkatan yang terjadi pada *Base*, terutama di kedua subset terakhir. Dari subset 1 (75%), peningkatan *sparsity* ke subset 2 (80%) dan selanjutnya ke subset 3 (85%) menunjukkan peningkatan MAE yang relatif sama, baik *Imputed* maupun *Base*, yaitu 3% - 4%. Tetapi, peningkatan *sparsity* ke subset 4 (90%) , dan selanjutnya ke subset 5 (95%), MAE pada sistem *Imputed* meningkat hanya sebesar 4% dan 10%, dibandingkan *Base*, yang meningkat 8% dan 73%.



Gambar 3 Grafik *Base* dan *Imputed* pada variasi *sparsity*

Tabel 4 Performansi *Base* dan *Imputed* pada variasi *sparsity*

	Sparsity				
	75%	80%	85%	90%	95%
Imputed	0.706	0.730	0.754	0.785	0.865
Base	0.681	0.699	0.723	0.781	1.348

4. Kesimpulan

Dengan membandingkan sistem yang menggunakan imputasi dan yang tidak, pengujian menunjukkan bahwa imputasi berpotensi menurunkan MAE dari *collaborative filtering*. Dari dataset yang digunakan dengan *sparsity* 75% - 85%, proses imputasi justru menurunkan performansi, dengan meningkatnya MAE sebesar 3% - 4% akibat bias yang muncul dari proses imputasi. Pada *sparsity* 90%, hal ini sudah berkurang, menjadi <1%, dan pada *sparsity* 95% proses imputasi berhasil menurunkan MAE sebesar 36% dibandingkan sistem tanpa imputasi. Dilihat dari perbedaan tren peningkatan MAE akibat variasi *sparsity*, imputasi membuat *collaborative filtering* lebih baik dalam menghadapi masalah *sparsity*, namun juga mengakibatkan bias akibat karakteristik data asli yang mengalami perubahan. *Collaborative filtering* masih mampu memberi prediksi yang baik untuk data yang padat, dalam hal ini untuk *sparsity* <=85%. Bias yang dihasilkan imputasi tidak sebanding dengan manfaatnya, sehingga proses imputasi berdampak negatif terhadap akurasi pada data yang padat. Namun, pada data yang jarang, proses imputasi menunjukkan efek yang positif, terlihat pada menurunnya MAE jika dibandingkan dengan sistem tanpa imputasi.

Daftar Pustaka:

[1] R. M. Bell and Y. Koren, "Improved Neighborhood-Based Collaborative Filtering," *KDD-Cup and Workshop*, pp. 7-14, 2007.

[2] R. Burke, "Hybrid Web Recommender System," in *The Adaptive Web*.: Springer Berlin Heidelberg, 2007, pp. 377-408.

[3] L. Candillier, F. Meyer, and M. Boulle, "Comparing State-of-the-Art Collaborative Filtering Systems," 2007.

[4] M. D. Ekstrand, J. T. Riedl, and Konstan J. A., "Collaborative Filtering Recommender Systems," *Foundations and Trends® in Human–Computer Interaction: Vol. 4: No. 2*, 2011.

[5] GroupLens. GroupLens. [Online]. http://grouplens.org/datasets/movielens/

[6] R. J. Hyndman and A. B. Koehler, "Another Look at Measures of Forecast Accuracy," *International Journal of Forecasting*, 2006.

[7] J. Kunegis, A. Lommatzsch, M. Mehlitz, and S. Albayrak, "Assessing the Value of Unrated Items in Collaborative Filtering," 2007.

[8] J. Lee, M. Sun, and G. Lebanon, "A Comparative Study of Collaborative Filtering Algorithms," 2012.

[9] R. J. A. Little and Rubin. D. J., *Statistical Analysis With Missing Data*., 2002.

[10] P., Mooney, R. J., Nagarajan, R. Melville, "Content-Boosted Collaborative Filtering," *Proceedings of the SIGIR-2001 Workshop on Recommender Systems*, 2001.

[11] A. M., Lam, S. K., Karypis, G., Riedl, J. Rashid, "ClustKNN: A Highly Scalable Hybrid Model-& Memory-Based CF Algorithm," *In Proc. of WebKDD-06, KDD Workshop on Web Mining and Web Usage Analysis, at 12 th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2006.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of Recommendation Algorithms for E-Commerce," *ACM Conference on Electronic Commerce*, 2000.

[13] X., Khoshgoftaar, T. M., Zhu, X., Greiner, R. Su, "Imputation-Boosted Collaborative Filtering Using Machine Learning Classifiers," *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 949-950, 2008.

[14] X. Su, T. M. Khosgoftaar, and R. Greiner, "Imputed Neighborhood Based Collaborative Filtering," *Web Intelligence and Intelligent Agent Technology*, 2008.