Sistem Rekomendasi Film Menggunakan Integrated Kohonen K-Means Clustering

Joshua Maximillian [1], Henry Novianus Palit [2], Alvin Nathaniel Tjondrowiguno [3]
Program Studi Informatika, Fakultas Teknologi Industri. Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp (031) – 2983455, Fax. (031) - 8417658
j05hu4030597@gmail.com [1], hnpalit@petra.ac.id [2], alvin.nathaniel@petra.ac.id [3]

ABSTRAK

Dengan seiring berkembangnya industri film semakin banyak pula film yang bisa untuk ditonton. Tetapi karena terlalu banyaknya film yang bisa untuk ditonton itu menyebabkan *user* bingung dalam mencari film yang sesuai dengan apa yang mereka suka. Sehingga ada sistem rekomendasi film yang dibuat untuk membantu *user*. Sistem rekomendasi film sendiri punya berbagai cara agar dapat menghasilkan rekomendasi film yang *user* mungkin suka.

Sistem rekomendasi film menggunakan *Integrated Kohonen K-Means Clustering* merupakan salah satu metode *Data Mining* yang dapat digunakan dalam merekomendasikan film. *Integrated Kohonen K-Means Clustering* dibandingkan dengan *Kohonen Self Organizing Maps*, dan juga *K-Means Clustering* dalam merekomendasikan film.

Menurut hasil pengujian terhadap metode *Integrated Kohonen K-Means Clustering*, jumlah K cluster yang optimal untuk *K-Mean Clustering* didapatkan dengan menggunakan *Elbow Method.* Untuk mengetahui seberapa bagus cluster yang dihasilkan menggunakan metode *Silhouette Coefficient* sebesar -0.389. Akurasi yang dihasilkan berdasarkan *Mean Reciprocal Rank* dengan nilai sebesar 0.362 menggunakan *Integrated Kohonen K-Means Clustering* lebih baik dibandingkan *K-Means Clustering* dengan nilai sebesar 0.003 dan *Kohonen Self Organizing Maps* dengan nilai sebesar 0.002 terhadap user 1.

Kata Kunci: *KSOM, K-Means Clustering, Data Mining, Movie Recommendation, Kohonen K-Means, Elbow Method, Silhouette Coefficient, MRR*

ABSTRACT

With the development of the film industry, more and more films can be watched. But because there are too many films that can be watched that cause users to be confused in finding films that match what they like. So there is a movie recommendation system to help user. The movie recommendation system itself has various ways to produce movie recommendations that users might like.

The movie recommendation system using Integrated Kohonen K-Means Clustering is one of the Data Mining methods that can be used in recommending films. Intergrated Kohonen K-Means Clustering compared to Kohonen Self Organizing Maps, and also K-Means Clustering in recommending films.

According to the result of Integrated Kohonen K-Means Clustering to know how many K cluster that is optimal for K-Means Clustering use the Elbow Method. To know how good the cluster you produce use Silhouette Coefficient and the score -0.389 for the Integrated Kohonen K-Means Clustering. The Mean Reciprocal Rank produced by Integrated Kohonen K-Means Clustering which score is 0.362 is better than K-Means Clustering which score is 0.003 and Kohonen Self Organizing Maps which score is 0.002.

Keywords: *KSOM, K-Means Clustering, Data Mining, Movie Recommendation, Kohonen K-Means, Elbow Method, Silhouette Coefficient, MRR*

1. PENDAHULUAN

Dengan seiring berkembangnya industri film semakin banyak pula film yang bisa untuk ditonton. Tetapi karena terlalu banyaknya film yang bisa untuk ditonton itu menyebabkan *user* bingung dalam mencari film yang sesuai dengan apa yang mereka suka. Pada suatu film sendiri pasti mempunyai *genre-genre* yang mungkin akan disukai oleh *user* itu. Dari *genre-genre* itu sendiri bisa diolah untuk mengelompokkan film-film yang mungkin akan disukai oleh *user* [7].

Berdasarkan penelitian berjudul *Movie Recommendation System based on Self-Organizing Maps* oleh Kaivan Wadia dengan Pulkit Gulpa menjelaskan bahwa metode *Kohonen Self Organizing Maps* mempunyai kelebihan yaitu kemudahan yang dapat mengatur data secara visual dan mudah untuk dipahami, dan dapat digunakan untuk menjelajahi bagian informasi yang tidak diketahui. Kelemahannya yaitu membutuhkan waktu yang lama untuk mengolah data itu [11].

Berdasarkan penelitian berjudul *Clustering Algorithms in Hybrid Recommender System on MovieLens Data* oleh Urzula Kużelewska menjelaskan bahwa metode *K-Means Clustering* mempunyai kelebihan mudah untuk diimplementasikan dan dijalankan,waktu yang dibutuhkan untuk menjalankan pembelajaran ini relatif cepat, mudah untuk diadaptasi. Kelemahannya yaitu kurang optimal pada saat inisialisasi pertama karena *weight* awal *random* [5].

Berdasarkan penelitian yang telah dikutip kesimpulannya adalah kedua metode tersebut dapat diimplementasikan pada *movie recommendation*, dan juga kelebihan dan kekurangan kedua metode ini dapat dihubungkan.

Berdasarkan penelitian *Kohonen Self Organizing Maps with Modified K-means clustering For High Dimensional Data Set* oleh Madhusmita Mishra dengan H.S. Behera menjelaskan bahwa *Integrated Kohonen K-Means Clustering* sendiri merupakan metode yang berasal dari kombinasi metode *Kohonen Self Organizing Maps* dengan *K-Means Clustering* yang dioptimasi menggunakan *Genetic Algorithm*, mempunyai performa yang cukup baik dan juga bisa menyelesaikan banyak masalah yang

dihadapi oleh *K-Means Clustering* seperti banyaknya *cluster* yang tidak diketahui dan tingkat *sensitivitas* dalam menginisialisasi *centroid* [6].

Pada penelitian ini *Integrated Kohonen K-Means Clustering* akan digunakan untuk merekomendasikan film dalam bentuk *web*. Yang dimana penelitian ini sendiri belum pernah dilakukan. Dan juga proses ini membutuhkan banyak waktu dimana kita harus menyaring *data* melalui *Kohonen Self Organizing Maps* terlebih dahulu lalu dioptimalkan menggunakan *Genetic Algorithm* untuk memperkecil *data set* yang ingin diambil, kemudian *cluster* yang diinginkan dan *centroid* awal dimasukan ke dalam *K-Means* untuk menemukan akurasi dan banyaknya iterasi yang dibutuhkan.

2. LANDASAN TEORI

2.1 Tinjauan Studi

Pada penelitian sebelumnya mengatakan bahwa *Kohonen Self Organizing Maps* lebih baik dalam memprediksi rating pada *user* baru untuk suatu *movie* secara akurat dibandingkan dengan *K-Means Clustering*. Meskipun *Kohonen Self Organizing Maps* membutuhkan waktu yang lebih lama dibandingkan dengan *K-Means Clustering* dalam memprediksi *rating.* Dalam penelitian ini juga penulis menggunakan *Root Mean Square Error* untuk mengukur prediksi *rating* dari beberapa *test sets* dan juga untuk membandingkan performa dari *Kohonen Self Organizing Maps* dan *K-Means Clustering.* Pada penelitian ini penulis mengatakan bahwa *Root Mean Square Error* yang didapatkan pada beberapa kali percobaan pada metode *Kohonen Self Organizing Maps* sekitar 0 hingga 1,1 sedangkan untuk *K-Means Clustering* sekitar 2,7 hingga 3,1. Penulis mengatakan bahwa semakin besar nilai *Root Mean Square Error* yang didapatkan maka semakin jelek pula akurasi yang didapatkan. Tetapi waktu yang diperlukan oleh *Kohonen Self Organizing Maps* sekitar 3600 *second* dibandingkan dengan waktu *K-Means Clustering* yang membutuhkan waktu sekitar 150 *seconds.*[10]

2.2 Kohonen Self Organizing Maps

Kohonen Self Organising Maps (KSOM) merupakan metode *unsupervised competitive learning.* Pada *Kohonen Self Organizing Maps* berusaha untuk memetakan *weight-weight* pada suatu *map* untuk memastikan input data mereka. Dan tujuan dari KSOM itu sendiri merupakan untuk menyederhanakan *multidimensional data* supaya dapat mudah untuk dimengerti. Pada *training* KSOM sendiri tidak memerlukan target *vector*.

Tiap *computational node* sendiri terhubung dengan *input node* untuk membentuk suatu *lattice*. *Weight* suatu *vector* dipengaruhi oleh banyaknya *input vector* [11].

2.3 K-Means Clustering

K-means clustering merupakan metode *unsupervised learning*. *K-means clustering* suatu algoritma yang digunakan untuk mengelompokkan berapa banyak objek berdasarkan atribut ke dalam beberapa k-partisi yang dimana jumlah k < banyak objek. Di dalam k-means berasumsi bahwa bentuk atribut suatu objek ke dalam *vector space*.[8]

2.4 Kohonen K-Means Clustering

Penelitian terhadap *integrated kohonen k-means clustering* sendiri sudah pernah dilakukan. Mishra & Behera [6] menerapkan *Kohonen Self Organizing Maps With Modified K-Means Clustering* pada *High Dimensional Data Set.* Mishra & Behera [6] mengatakan bahwa metode ini sendiri bisa menyelesaikan berbagai masalah yang dihadapi oleh *K-Means Clustering* seperti tidak diketahui banyaknya *cluster* yang diketahui dan memberikan *sensitivity* pada *centroid* awal.

2.5 Elbow Function

Elbow Method merupakan suatu metode yang melihat prosentase *variance* yang dijelaskan dalam bentuk *function* dari jumlah cluster. Metode ini atas adanya gagasan bahwa salah seorang harus memilih sejumlah *cluster* sehingga *cluster* lain tidak memberikan pemodelan data yang jauh lebih baik. Jumlah prosentase *variance* dapat dijelaskan dengan jumlah *cluster* yang sudah direncanakan dibandingkan dengan jumlah *cluster* yang ditentukan. Pada *cluster* pertama akan menambahkan banyak informasi tetapi ketika pada *point* tertentu *marginal gain* akan turun secara drastis dan akan memberikan sudut dalam bentuk grafik. Jumlah k *cluster* yang benar dipilih pada *point* ini, karena adanya *"elbow criterion".* [2]

2.6 Silhouette Coefficient

Silhouette Coefficient merupakan ukuran validitas suatu *cluster*. [1] Jika nilai *Silhouette coefficent* mendekati angka 1 maka objek-objek yang ditentukan berada di *cluster* yang tepat. Apabila nilai *Silhouette Coefficient negative* maka objek-objek yang ditentukan berada di cluster yang salah. Dan bila nilai *Silhouette Coefficient* berada di angka 0 maka objek-objek yang ditentukan berada diantara 2 *cluster* yang berdekatan.

2.7 Mean Reciprocal Rank

Mean Reciprocal Rank (MRR) dikaitkan dengan *user model* di mana *user* hanya ingin melihat satu dokumen yang relevan. Dengan asumsi bahwa pengguna akan melihat ke bawah peringkat sampai dokumen yang relevan ditemukan, dan itu dokumen berada di peringkat n, maka ketepatan *set* mereka melihat 1 / n, yang juga merupakan ukuran *reciprocal rank*. MRR adalah ukuran yang tepat untuk *known item search*, tempat dimana pengguna berusaha menemukan dokumen yang dia pernah lihat sebelumnya atau sebelumnya tahu apabila ada. MRR ini bisa disebut pencarian navigasi dalam kasus pencarian web. [3]

2.8 Dataset Movielens

Dataset movielens yang dipakai dalam penelitian ini *movies.csv*, *ratings.csv*, *genome-scores.csv* dan *genome-tags.csv*. Pada penelitian ini *dataset movielens* yang digunakan adalah *movielens latest datasets* yang mempunyai sekitar 27.000.000 *rating*, 58098 *movie*, 283228 user yang *unknown* dan 1128 *tag*. Isi dari *movies.csv* merupakan *data movieId, title, genre*. Isi *ratings.csv* merupakan *data userId, movieId, rating, timestamp*. Isi *genome-scores.csv* merupakan *data movieId, tagId, relevance.* Isi *genome-tags.csv* merupakan *data tagId, tag.*[4]. *Data* yang digunakan untuk melakukan *clustering* ada sebanyak 13176 *movie* yang dimana mempunyai 1128 *tag* tiap-tiap *movie*. *User dataset movielens* sendiri dipilih secara *random* untuk dicantumkan saja. Dan seluruh *user* yang berada di *dataset movielens* paling sedikit pernah melakukan *rating* pada 1 film. *MovieId* yang ada disesuaikan dengan *data movie* yang diperoleh dari *Movielens*. Seluruh *data rating* berada di *ratings.csv* yang diurutkan berdasarkan *userId,*dan *movieId. Data rating* yang ada juga mempunyai skala dari angka 0.5-5.0., *data Timestamp* yang ada di dalam *ratings.csv* dalam bentuk *second. Movie data* yang ada merupakan hasil *input* manual yang berasal dari *themoviedb.org*

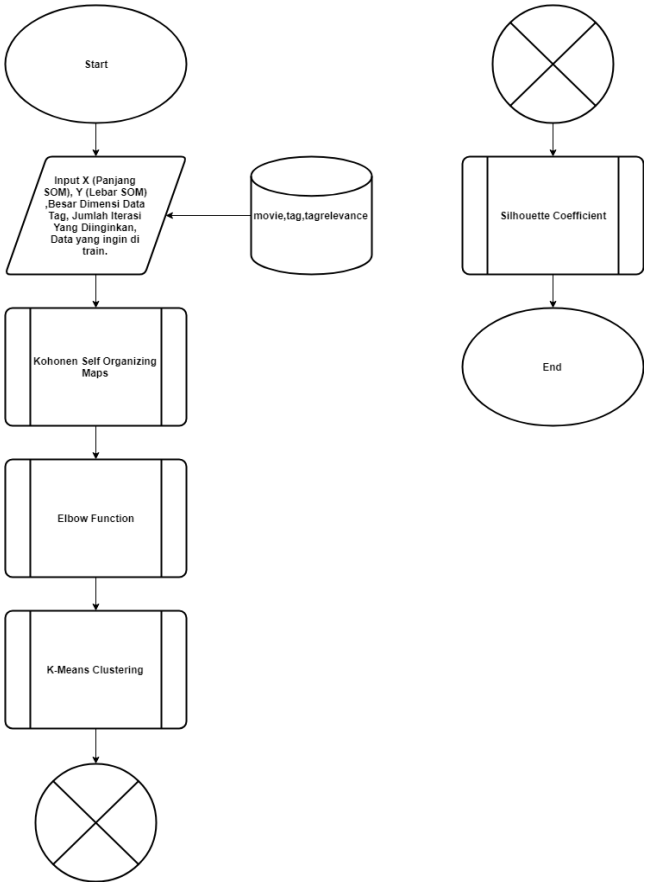
dan juga tahun terbit *movie* dicantumkan. *Genome-tags.csv* merupakan suatu *data structure* yang berisi *tag relevance score* untuk tiap film. Struktur data nya berisikan *dense matrix.* Setiap *movie* mempunyai setiap *tag* yang berada di *genome.*

3. ANALISIS DAN DESAIN SISTEM

3.1 Analisis Sistem

Analisis sistem membahas permasalahan bagaimana data yang di-input diproses sehingga dapat menghasilkan output yang sesuai. Sistem mencakup pengambilan data, proses data mining untuk training, dan juga sistem untuk testing. Analisis system sendiri terbagi menjadi 6 metode dan akan dijelaskan dalam berupa poin-poin sebagai berikut:

3.1.1 Integrated Kohonen K-Means Clustering



Gambar 1. ***Flowchart Integrated Kohonen Self Organizing Maps***

Pada Gambar 1. Dijelaskan prose Intergrated Kohonen K-Means Clustering yang dimulai dengan input panjang dan lebar *SOM grid* pada *Kohonen Self Organizing Maps*, besar dimensi yang diperoleh dari *data tag*, jumlah iterasi yang diinginkan, *data relevance* tiap *movie.* Kemudian masuk ke dalam proses *Kohonen Self Organizing Maps* yang mengahasilkan *cluster-cluster* dalam bentuk titik *centroid grid*. Setelah itu titik-titik *centroid grid* dimasukkan kedalam *elbow function* untuk mencari jumlah *k cluster* yang optimal. Tahap berikutnya setelah mendapatkan jumlah *k cluster* yang optimal yang sebelumnya didapatkan dari *elbow function* jumlah *k cluster* yang dipilih dan digunakan sebagai *input* ke dalam proses *K-Means Clustering* dan juga titik *centroid grid* yang dihasilkan oleh *Kohonen Self Organizing Maps* digunakan sebagai input data ke dalam *K-Means Clustering*. Lalu proses *K-Means Clustering* akan Menghasilkan *cluster-cluster* baru yang kemudian akan dicoba untuk apakah *cluster* yang dihasilkan itu *valid* atau tidak dengan menggunakan *Silhouette Coeffecient*. Hasil dari *Silhouette Coefficient* sendiri berupa angka yang menunjukkan seberapa bagus *cluster* yang kita pilih. *Silhouette Coefficient* akan menghasilkan angka yang mempunyai *range* -1 sampai 1. Apabila angka yang didapatkan dari *Silhouette Coefficient* mendekati 1 maka *cluster* yang kita pilih itu sudah tepat pada tempat-tempatnya. Apabila angka yang didapatkan dari *Silhouette Coefficient* mendekati -1 maka *cluster* yang kita gunakan itu kurang cocok.

3.1.2 Kohonen Self Organizing Maps

Dimulai dengan X & Y untuk menentukan ukuran SOM , jumlah iterasi, besar dimensi *input data* yang ingin di-*train* dalam kasus ini menggunakan besarnya *data tag* ,dan *input data* yang ingin di *train* dari *user* berupa *vector* yang berisikan *relevance* pada suatu *movie* yang mempunyai *tag* yang ditentukan.*Step* kedua menentukan besarnya SOM yang diinginkan terlebih dahulu lalu *weight* dipilih secara *random* untuk diinisialisasi kedalam *input data vector* untuk di-*train*. *Step* ketiga, Menentukan BMU (*Best Matching Unit)* dengan menggunakan *Euclidean Distance*. BMU sendiri digunakan untuk mendapatkan *weight* yang mendekati dengan *input vector*. *Weight* yang mendekati *input vector* dijadikan sebagai *sample vector*. *Step* keempat menentukan *neighborhood* berdasarkan BMU yang dijadikan sebagai *sample vector* . *Step* kelima *neighborhood* yang merupakan *sample vector* dari BMU diupdate sehingga mendapatkan *weight* yang baru, pada proses ini jumlah *neighbor* akan terus berkurang tiap iterasinya. Kemudian lakukan kembali *step* ketiga dan selanjutnya sampai sesuai dengan jumlah iterasi yang kita inginkan. Apabila iterasi yang kita inginkan sudah tercapai maka selesai dan menghasilkan *output* berupa lokasi *vector* untuk tiap movienya.

3.1.3 K-Means Clustering

Dimulai dengan *input data* yang berupa *vector input* yang berasal dari *relevance* pada suatu *movie* yang mempunyai *tag-tag* yang sudah ditentukan, dan juga jumlah *cluster* yang diinginkan. Lalu masuk pada *Step* pertama yaitu mencari *centroid* yang ditentukan secara *random* sejumlah *cluster* yang diinginkan. *Step* kedua setelah mendapatkan *centroid, vector-vector input* dimasukkan kedalam *Euclidean Distance* untuk mendaptakan *distance* antara *centroid* dengan *vector-vector* yang telah *diinputkan. Step ketiga* setelah mendapatkan *distance, vector input* dimasukkan kedalam *cluster-cluster* yang sudah ditentukan berdasarkan *distance* yang paling kecil. Lalu lakukan kembali *step* kedua sampai iterasi sesuai yang kita inginkan.

3.1.4 Elbow Method

Dimulai dengan *input data Relevance* yang berasal dari *database tagrelevance.* Kemudian *input range cluster* yang diinginkan. Setelah itu masuk *proses K-Means Clustering.* Lalu hitung *Intra Cluster Distance.* Kemudian dilihat apakah *range k* sudah sesuai dengan yang *diinputkan*. Apabila ya maka akan mengeluarkan grafik yang berisikan nilai *error* berdasarkan rumus *intra cluster distance* untuk tiap *k cluster*. Apabila tidak maka akan mengulangi proses *k-means clustering* lagi.

3.1.5 Silhouette Coefficient

Dimulai dengan *input data Relevance* yang berasal dari *database tagrelevance.* Kemudian *input cluster* tujuan yang diinginkan. Lalu hitung *Intra Cluster Distance* dan juga *Inter Cluster Distance.* Kemudian mencari nilai *max* dari *Intra* dan *Inter Cluster Distance*. Setelah itu *Inter Cluster Distance - Intra Cluster Distance* / *max* dari *intra* dan *inter cluster distance.*

3.1.6 Mean Reciprocal Rank

Dimulai dengan *input* dari *data testing* yang dibagi menjadi 40% untuk tiap *user*-nya kemudian *input data* Rekomendasi film yang berasal dari *cluster* yang dihasilkan yang berada di *database movie* yang sebelumnya sudah melalui proses *Integrated Kohonen K-Means Clustering , Kohonen Self Organizing Maps, K-Means Clustering.* Kemudian mencari *rank* yang berada di *data* rekomendasi yang dibandingkan dengan *data testing*. Data rekomendasi didaptakan dari melihat interest user yang ditentukan dengan cara melihat jumlah film dengan genre apa yang ia senangi terlebih dahulu, lalu melihat *movie* dengan *rating* tertinggi yang pernah *user* berikan dalam genre yang dia sukai, kemudian melihat berada di *cluster* mana, kemudian di urutkan berdasarkan *distance movie* yang berada di *training* dengan *movie* yang berada dalam 1 *cluster* tetapi belum ada di dalam *training*. Apabila ada film dalam *testing* dan juga ada dalam *data* rekomendasi, maka masuk kedalam rumus 1/*rank* yang dimana *rank* didapatkan dengan melihat urutan *data testing* berada di urutan ke berapa di dalam *data* rekomendasi. Setelah itu dilihat apakah jumlah *testing* sudah sesuai apabila belum maka kembali ke proses *data testing* kembali apabila ya maka akan masuk ke dalam proses semua *rank* sebelumnya dijumlah lalu dibagi dengan jumlah data testing atau *mean* jumlah *rank* yang didapat dengan jumlah *data testing* yang pernah dilakukan.

3.2 Desain Sistem

3.2.1 Alur Rekomendasi

Alur rekomendasi yang digunakan oleh penulis dimulai *input* berupa *movie* yang dipilih oleh user. Kemudian *movie* yang dipilih *user* dilihat ada pada *cluster* yang sebelumnya telah dilakukan proses *Integrated Kohonen K-Means Clustering, Kohonen Self Organizing Maps*, dan *K-Means Clustering.* Selanjutnnya melihat *movie* apa saja yang berada pada *cluster* dari movie yang dipilih. Apabila tidak ada maka tidak akan dijadikan rekomendasi dan kembali lagi mencari *movie* dengan *cluster* yang sama. Apabila ada maka akan dimasukkan ke dalam rekomendasi film. Kemudian setelah dimaukkan kedalam rekomendasi dihitung *Distance* antara *movie* yang direkomendasikan degna movie yang dipilih oleh user menggunakan *Euclidean Distance* untuk mendapatkan urutan *movie* yang direkomendasikan. Kemudian dilihat apakah ada *movie* yang cocok dengan keinginan *user* apabila iya maka *movie* yang cocok tadi dilihat ada pada urutan berapa lalu dimasukkan ke dalam proses *Mean Reciprocal Rank.*

4. PENGUJIAN SISTEM

4.1 Pengujian Sistem

Pada Implementasi ini akan menghasilkan Akurasi yang berupa nilai *Mean Reciprocla Rank, Silhouette Coefficient* dan juga prediksi waktu dari metode *Integrated Kohonen K-Means Clustering, Kohonen Self Organizing Maps,*dan *K-Means Clustering.* Berikut merupakan Tabel hasil dari Implementasi:

Tabel 1. Percobaan *K-Means Clustering*

Percobaan	Cluster	Jumlah *Movie*	Time (S)
1	7	13176	17534.3
2	8	13176	15316.7
3	9	13176	15619.8

Tabel 2. Percobaan *Kohonen Self Organizing Maps*

Perc-obaan	X	Y	Lear-ning Rate	Jumlah Movie	Time(S)
1	50	50	0.7	13176	15982.7
2	30	30	0.7	13176	14563.8
3	20	20	0.7	13176	5891.0
4	20	20	0.3	13176	6006.6
5	4	2	0.7	13176	459.8
6	3	3	0.7	13176	480.3
7	2	4	0.7	13176	480.1

Tabel 3. Percobaan *Integrated Kohonen K-Means Clustering*

Perco-baan	X	Y	Cluster	Jumlah Movie	Time(S)
1	50	50	8	13176	60307.8
2	3	3	4	13176	16414.4
3	3	3	3	13176	16496.4

Pada Tabel 1., Tabel 2., Tabel 3. merupakan hasil percobaan *time* yang didapatkan dengan menggunakan metode-metode yang digunakan. Semakin banyak *time* yang didapatkan semakin lama juga proses yang dilakukan oleh metode yang digunakan.

Pada percobaan yang telah dilakukan dapat dilihat bahwa dalam beberapa percobaan bahwa metode *Integrated Kohonen K-Means Clustering* membutuhkan waktu yang lebih lama dibandingkan *Kohonen Self Organizing Maps* dan *K-Means Clustering*. Pada percobaan *K-Means Clustering* banyanya *cluster* yang diuji didapatkan dari *elbow function* sebelumnya telah dilakukukan pada Gambar 3. yang dimana *K* yang terbaik berada di posisi 8.

Pada percobaan KSOM dapat dilihat bahwa hasil yang terbaik ada pada percobaan ke 5 dan ke 7, tetapi mengapa percobaan ke 6 yang digunakan untuk *Integrated Kohonen K-Means Clustering*? Karena pada KSOM lebih baik apabila *SOM Grid* mempunyai dimensi yang sama antara X & Y.

Tabel 4. *Silhouette Coefficient* pada *K-Means Clustering*

Percobaan	Cluster	Jumlah Movie	*Silhouette Coefficient*
1	7	13176	-0.553
2	8	13176	-0.466
3	9	13176	-0.489

Tabel 5. *Silhouette Coefficient* pada *Kohonen Self Organizing Maps*

Per-cob-aan	X	Y	Le-arn-ing Ra-te	Jumlah Movie	*Silhouette Coefficient*
1	50	50	0.7	13176	-0.984
2	30	30	0.7	13176	-0.893
3	20	20	0.7	13176	-0.810
4	20	20	0.3	13176	-0.822
5	4	2	0.7	13176	-0.528
6	3	3	0.7	13176	-0.530
7	2	4	0.7	13176	-0.528

Tabel 6. *Silhouette Coefficient* pada *Integrated Kohonen K-Means Clustering*

Per-cob-aan	X	Y	Clus-ter	Jumlah Movie	*Silhouet-te Coeffici-ent*
1	50	50	8	13176	-0.531
2	3	3	4	13176	-0.632
3	3	3	3	13176	-0.389

Pada Tabel 4., Tabel 5., & Tabel 6. merupakan Grafik *Silhouette Coefficient* dari beberapa percobaan dengan menggunakan 3 metode. Semakin tinggi nilai *silhouette coefficient* maka semakin bagus juga *cluster* yang dihasilkan.

Dalam percobaan yang telah dilakukan dapat dilihat bahwa *best cluster* terdapat pada *Integrated Kohonen K-Means Clustering* pada percobaan ke-3 yang telah dihasilkan oleh proses *Silhouette Coefficient*. Parameter X & Y merupakan parameter untuk menentukan dimensi *SOM Grid*. Lalu *cluster* merupakan jumlah *cluster* yang diinginkan yang dilakukan dalam *K-Means Clustering*. *Learning rate* merupakan parameter yang dibutuhkan dalam proses KSOM. Pada beberapa percobaan dapat kita lihat bahwa hasil *Silhouette Coefficient* yang terbaik ada pada percobaan *Integrated Kohonen K-Means Clustering*.

Pada beberapa kali percobaan *Intergrated Kohonen K-Means Clustering* dapat kita lihat nilai *cluster* yang ada didapatkan dari proses KSOM yang sebelumnya dijalankan yang dimana KSOM pada percobaan 1 lalu dimasukkan hasil KSOM yang berupa *SOM Grid* ke dalam *elbow function* dan dapat dilihat pada Gambar 4. bahwa *K cluster* yang bagus berada di angka 8. Sedangkan *cluster* yang bagus dari percobaan 6 di KSOM menunjukkan bahwa *K cluster* yang bagus ada di angka 3 yang dapat dilihat dari Gambar 5.

Tabel 7. MRR tiap *user* menggunakan metode *Integrated Kohonen K-Means Clustering*

User	Mean Reciprocal Rank	Jumlah *Movie*	Data Pengujian
1	0.36276127692932886	10	6

Lanjutan Tabel 7. MRR tiap *user* menggunakan metode *Integrated Kohonen K-Means Clustering*

User	Mean Reciprocal Rank	Jumlah *Movie*	Data Pengujian
2	0.33783783794691163	9	6
3	0.375	7	4
7	0.16666666666666666	9	6
11	0.358233369687306	12	7
12	0.298575680578194	11	7
13	0.007452068879501894	12	8
21	0.6724137930820385	9	6
23	0.013642402897988046	10	7
25	0.4027777798473835	10	6
27	0.48888889079292613	10	6
30	0.4323308270956789	10	7
32	0.625	9	6
40	0.375	10	4
50	0.625	9	4
61	0.3751249375345651	7	6
63	0.2651515156030655	9	6
64	0.46666666865348816	12	5
66	0.19047619154055914	11	6
74	0.15476190511669433	12	7

Tabel 8. MRR tiap *user* menggunakan metode *Kohonen Self Organizing Maps*

User	Mean Reciprocal Rank	Jumlah *Movie*	Data Pengujian
1	0.0019379844889044762	10	6
2	0.16666666666666666	9	6
3	0.2916666679084301	7	4
7	0.3333333333333333	9	6
11	0.0011160714285714285	12	7
12	0.0017421602138451167	11	7
13	0	12	8
21	0.34375	9	6
23	0.3591836734807917	10	7
25	0.25	10	6
27	0.2833333338300387	10	6
30	0.14732142857142858	10	7
32	0.25	9	6
40	0	10	4
50	0.375	9	4
61	0.00009448223863728344	7	6

Lanjutan Tabel 8. MRR tiap *user* menggunakan metode *Kohonen Self Organizing Maps*

User	Mean Reciprocal Rank	Jumlah *Movie*	Data Pengujian
63	0.2222222238779068	9	6
64	0	12	5
66	0.16666666666666666	11	6
74	0.14285714285714285	12	7

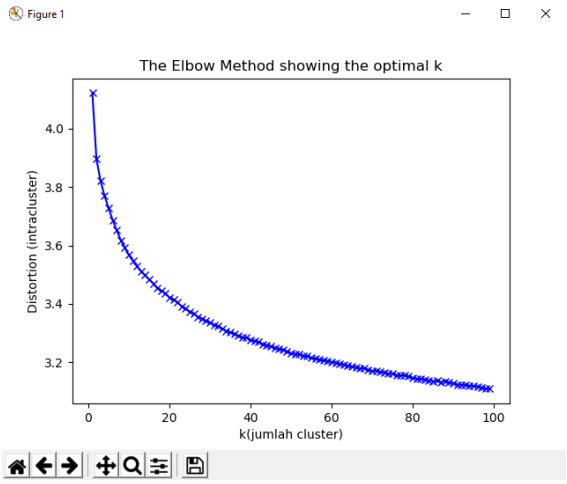
Tabel 9. MRR tiap *user* menggunakan metode *K-Means Clustering*

User	Mean Reciprocal Rank	Jumlah *Movie*	Data Pengujian
1	0.0026041666666666665	10	6
2	0.16666666666666666	9	6
3	0.3125	7	4
7	0.3333333333333333	9	6
11	0.0021321961124028477	12	7
12	0.1438492063565978	11	7
13	0	12	8
21	0.341666666790843	9	6
23	0.14634146328483308	10	7
25	0.25	10	6
27	0.3333333333333333	10	6
30	0.14795918390154839	10	7
32	0.3333333333333333	9	6
40	0	10	4
50	0.25	9	4
61	0.00011956001981161535	7	6
63	0.041666666666666664	9	6
64	0	12	5
66	0.16666666666666666	11	6
74	0	12	7

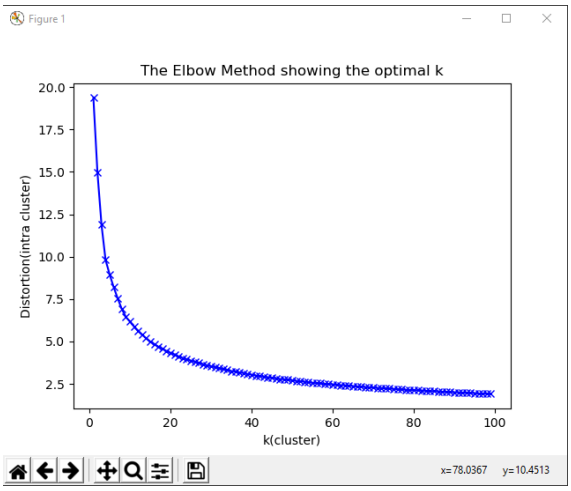
Pada Tabel 7., Tabel 8., Tabel 9. merupakan hasil *Mean Reciprocal Rank* (MRR) yang didapatkan dari metode-metode yang digunakan. Semakin tinggi hasil *Mean Reciprocal Rank* maka semakin bagus pula hasil rekomendasi film yang didapatkan. MRR yang didapatkan dapat dilihat bahwa metode *Integrated Kohonen K-Means Clustering* merupakan metode yang paling baik diantara metode-metode lainnya. Paramater jumlah *movie* merupakan jumlah *movie* yang *user* pernah *rating*.

Data pengujian adalah beberapa *movie* yang pernah di-*rating* oleh *user* tetapi belum dianggap pernah di-*rating*. Cara untuk mendapatkan MRR sendiri harus melalui beberapa kali percobaan yang dimana untuk mendapatkan *Reciprocal Rank* sendiri diuji dengan melihat apa film yang direkomendasikan apakah *user* akan *rating* dan berada di urutan ke berapa. Dan cara ini sendiri dilakukan sebanyak dari berapa *data* pengujian yang dibutuhkan. Dan apabila sudah menemukan *Reciprocal Rank* maka movie itu dimasukkan ka dalam *data training* yang dimana berasal dari jumlah movie.

Rank movie yang direkomendasikan bisa sewaktu-waktu berubah apabila *Distance movie* yang direkomendasikan dengan *movie* yang pernah *user* tonton itu bertambah. Cara ini dilakukan terus sampai *data* pengujian yang dibutuhkan sudah habis. Apabila sudah habis cara ini dicoba lagi untuk *user* yang lain.

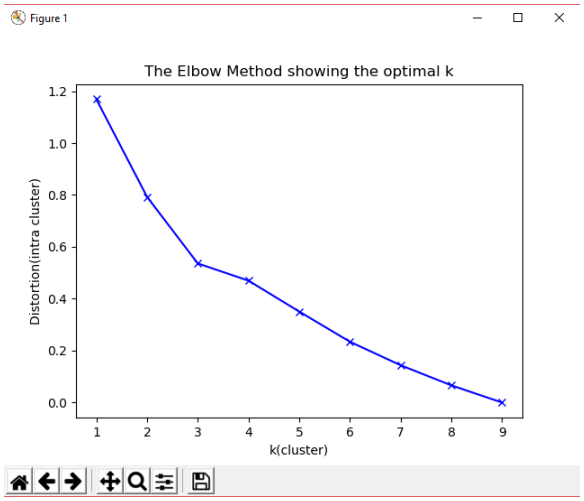


Gambar 2. *Elbow Function* untuk percobaan dari *data tagrelevance*



Gambar 3. *Elbow Function* untuk percobaan KSOM 50x50

Pada Gambar 2. merupakan *elbow function* yang digunakan untuk mendapatkan nilai *K-Means* agar mendapatkan nilai *K* yang optimal yang *input*-nya berasal dari *data relevance* tiap film. Pada Gambar 3. yang berada di atas merupakan *elbow function* yang berasal dari *input data tagrelevance* yang sudah melalui proses *Kohonen Self Organizing Maps* sebesar 50 x 50 *som grid*. Pada Gambar 4. merupakan hasil *elbow function* yang didapatkan dari *input data tagrelevance* yang sudah melalui proses *Kohonen Self Organizing Maps* sebesar 3x3 *som grid*.



Gambar 4. *Elbow Function* untuk percobaan KSOM 3x3

5. KESIMPULAN DAN SARAN

Dari hasil perancangan dan pembuatan sistem dan aplikasi, dapat diambil kesimpulan antara lain:

- Nilai MRR yang didapatkan dalam merekomendasikan film pada *Integrated Kohonen K-Means Clustering* yang bernilai 0.36276127692932886 lebih baik dibandingkan *K-Means Clustering* yang bernilai 0.0026041666666666665 dan *Kohonen Self Organizing Maps* yang bernilai 0.0019379844889044762 .
- Nilai *Silhouette Coefficient* yang dihasilkan *Integrated Kohonen K-Means Clustering* -0.389 yang dimana lebih baik dibandingkan *K-Means Clustering* yang bernilai -0.466 dan juga *Kohonen Self Organizing Maps* yang bernilai -0.530

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan Implementasi ini adalah:

- Parameter data yang di-*training* tidak hanya *relevance* saja
- Tiap tiap *movie* mempunyai nilai *tag*
- Untuk penelitian yang berikutnya bisa menggabungkan dengan *collaborative filtering* dan juga dibandingkan hasilnya dalam merekomendasikan
- *User* yang digunakan lebih diperbanyak agar mendapatkkan akurasi yang lebih akurat
- Pada saat melakukan rekomendasi *rating* diperhitungkan juga

6. DAFTAR REFERENSI

[1] Aranganayagi & Thangavel, 2007. *Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure.* URI= https://ieeexplore.ieee.org/abstract/document/4426662

[2] Bholowalia & Kumar, 2014. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. DOI=10.1.1.735.7337

[3] Caragea, C., Honavar, V., Boncz, P., Boncz, P., Larson, P.-Å., Dietrich, S. W., Wolfson, O., 2009. *Mean Reciprocal Rank.* URI= https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_488

[4] Grouplens, 2019. URI= https://grouplens.org/datasets/movielens/

[5] Kużelewska ,2014. Clustering Algorithms in Hybrid Recommender System on MovieLens Data. URI=http://logika.uwb.edu.pl/studies/download.php?volid=50&artid=50-07&format=PDF

[6] Mishra & Behera ,2012. Kohonen Self Organizing Map with Modified K-means clustering For High Dimensional Data Set. URI= https://pdfs.semanticscholar.org/9ef4/d5d6503c706ba5f09ed18fe5bef2c4ef62f8.pdf

[7] Praba et al.,2018. Movie Recommendation System. URI= https://www.ijresm.com/Vol_1_2018/Vol1_Iss10_October18/IJRESM_V1_I10_217.pdf

[8] Prabhu, 2015. K-mean Clustering Algortihm. URI= https://www.slideshare.net/parryprabhu/k-meanclustering-algorithm

[9] Raphael, 2015. Kohonen self organizing maps. URI=https://www.slideshare.net/raphaelkiminya/kohonen-self-organizing-maps

[10] Seo E. & Choi H., 2010. Movie Recommendation with K-Means Clustering and Self Organizing Methods. URI=https://scitepress.org/papers/2010/27376/27376.pdf

[11] Wadia & Gupta ,n.d.. Movie Recommendation System based on Self-Organizing Maps. URI= http://www.kaivanwadia.com/Projects/MRS-NN.pdf