RANCANG BANGUN MUSIC RECOMMENDER SYSTEM DENGAN METODE USER-BASED COLLABORATIVE FILTERING

Teguh Budianto [1], Galih Hermawan [2]

Program Studi Teknik Informatika. UNIKOM.
Jl. Dipatiukur No. 114 – 116, Bandung 40132.
E-mail : teguh.budianto.17@gmail.com [1], galih.hermawan@yahoo.co.id [2]

ABSTRAK

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pengguna, oleh karena itu sistem perekomendasi memerlukan model rekomendasi yang tepat agar yang di rekomendasikan sesuai dengan keinginan pengguna. Industri musik belakangan ini mengalami perubahan yang sangat signifikan. Konsumen kini cenderung mengakses dan membeli konten secara *online* dibandingkan pergi ke sebuah toko, hal ini jelas menimbulkan masalah pertumbuhan data yang sangat cepat di *internet* sehingga menyebabkan terlalu banyaknya informasi yang tersedia.

Dalam penelitian ini sistem rekomendasi dianalisa dan dibangun dengan menggunakan metode *user-based collaborative filtering* karena dengan algoritma ini melibatkan subyektifitas pengguna sehingga rekomendasi yang dihasilkan mempunyai kualitas yang baik. Namun metode *user-based collaborative filtering* ini masih mempunyai kekurangan yaitu adanya *scalability* (keadaan dimana tingginya jumlah peningkatan *user* dan *item* di dalam *database*) dan *sparsity* (terjadinya kekosongan data matriks *user-item*), oleh karena itu perlu digunakan algoritma tambahan yaitu *K-Means clustering* dan proses *smoothing* yang bertujuan untuk menangani dua masalah utama tersebut.

Berdasarkan implementasi dan hasil pengujian maka dapat ditarik kesimpulan sebagai berikut, berdasarkan penggunaan metode *user-based collaborative filtering* ini mampu mengatasi masalah kekosongan data dengan tingkat *sparsity* sebanyak 70% namun untuk hasil akhir dari pengujian metode ini menggunakan MAE *(Mean Absolute Error)* yang mempunyai rentang nilai 0 – 1 didapatkan data yaitu 0,6713 artinya penggunaan metode ini menghasilkan nilai lebih dari 0,5 maka dapat disimpulkan metode ini masih kurang akurat oleh karena itu perlu digunakan algoritma lain yang menghasilkan nilai yang lebih akurat, atau tidak lebih dari 0,5.

Kata kunci : Sistem Rekomendasi, *user-based collaborative filtering*.

1. PENDAHULUAN

Industri musik belakangan ini mengalami perubahan yang sangat signifikan. Konsumen kini cenderung mengakses dan membeli konten secara online dibandingkan pergi ke sebuah toko, hal ini jelas menimbulkan masalah pertumbuhan data yang sangat cepat di internet menyebabkan terlalu banyaknya informasi yang tersedia. Hal ini menyebabkan seseorang mengalami kesulitan dalam mendapatkan informasi mengenai musik yang cepat dan sesuai dengan kebutuhan. Untuk itu diperlukan sistem rekomendasi sehingga dapat membantu seseorang menemukan informasi musik yang sesuai dengan kebutuhannya (Saptariani, Trini. 2014).

Dalam penelitian ini sistem rekomendasi akan dianalisa dan dibangun dengan menggunakan metode *user-based collaborative filtering* karena dengan algoritma ini melibatkan subyektifitas pengguna dalam perhitungan sehingga rekomendasi yang dihasilkan mempunyai kualitas yang baik. Namun metode *user-based collaborative filtering* ini masih mempunyai kekurangan yaitu adanya :

1. *Scalability*, yaitu keadaan dimana tingginya jumlah peningkatan *user* dan *item* di dalam database yang mempengaruhi menurunnya komputasi algoritma *user-based collaborative filtering*.
2. *Sparsity*, yaitu terjadinya kekosongan data matriks *user-item*, yang disebabkan karena *user* merating dalam jumlah kecil dari jumlah item yang tersedia di dalam database.

Oleh karena itu perlu digunakan algoritma tambahan yaiitu *K-Means clustering* dan proses *smoothing* yang bertujuan untuk menangani dua masalah utama tersebut. *Music recommender system* ini dibangun diatas *platform web*, dengan pertimbangan perangkat lunak yang dibangun harus memiliki kemampuan untuk menangani banyak pengguna.

Berdasarkan uraian di atas dengan permasalahan yang terjadi, maka dengan harapan sistem rekomendasi ini nantinya dapat membantu dan memberikan informasi tentang musik yang tepat untuk pengguna.

2. ISI PENELITIAN

2.1 *Information Retrieval* dan *Information Filtering*

Perkembangan internet yang sangat pesat secara tidak langsung memberikan kemampuan kepada pengguna untuk memilih di antara banyak dari informasi yang ada di internet. Informasi ini bisa berkaitan dengan profesi mereka, kejadian yang ada di dunia, atau bahkan informasi yang berkaitan dengan gaya hidup. Informasi yang dibutuhkan oleh pengguna internet terus mengalami peningkatan dan ini bisa datangnya dari berbagai sumber yang berbeda-beda. Misalnya melalui halaman web, email, artikel, berita, jurnal, situs belanja, dan situs multimedia. Perkembangan ini memicu terjadinya *information explosion* di dunia maya yang menyulitkan pengguna dalam mencari suatu informasi dengan cepat dan relevan. Masalah ini menjadi alasan dikembangkannya beberapa teknik untuk *information retrieval* dan *information filtering* (Mortensen, Magnus. 2007).

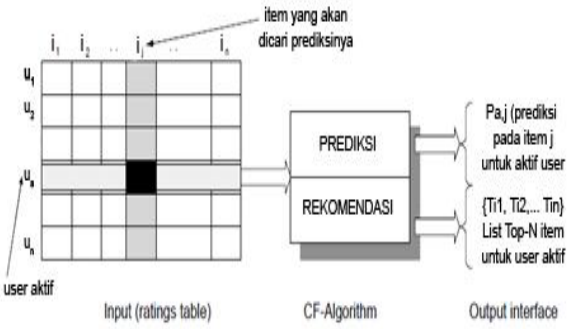
2.2 *Recommender System* (Sistem Rekomendasi)

Recommender System merupakan sebuah aplikasi *information filtering* untuk mencari dan memberikan rekomendasi *item* berupa informasi, produk, atau layanan kepada *user* berdasarkan prediksi yang bersifat personal (Sarwar, Badrul. 2001). Pengembangan *recommender system* oleh berbagai *online vendor* merupakan langkah untuk menarik lebih banyak perhatian pengguna dan meningkatkan kepuasan pengguna terhadap hasil pencarian informasi secara *online*. Pada *e-commerce* misalnya, dimana sistem rekomendasi digunakan secara luas untuk menyarankan produk kepada pelanggan dan untuk menyediakan pelanggan dengan informasi sehingga membantu pelanggan memutuskan pilihan produk yang akan dibeli (Mortensen, Magnus. 2007). Hal ini menjadi sangat penting bagi keberhasilan industri di bidang teknologi informasi dan *e-commerce* saat ini yang secara bertahap mendatangkan keuntungan dari segi popularitas di berbagai aplikasi, misalnya proyek Netflix, Google news, dan Amazon.

Sistem rekomendasi dibangun dengan tujuan membantu *user* untuk memilih *item-item* yang disukainya dari sekian banyak *item* yang tersedia. Teknik pencari *item* yang akan direkomendasikan dapat dilakukan berdasarkan kemiripan, bisa berupa kemiripan suatu *item* dengan *item* lainnya, berdasarkan konten atau kemiripan selera suatu *user* dengan *user* lain berdasarkan *rating* yang diberikan pada *item*.

2.3 Metode *Collaborative Filtering*

Collaborative filtering (CF) adalah teknologi *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 *user* yang dikenal sebagai *neighbors*, dimana setiap *user* memiliki kesamaan minat dan pendapat dengan target *user* (yaitu, mereka memiliki beberapa *rating item* yang sama atau kecenderungan *user* menyukai *item* yang sama). Setelah lingkungan *neighbors* terbentuk, sistem ini akan menggunakan beberapa algoritma untuk menghasilkan rekomendasi.



Gambar 2.1. Proses *Collaborative Filtering* (Sarwar, Badrul. 2001).

Dalam skenario CF terdapat daftar pengguna m *user* U = {u1, u2,…, um} dan daftar *item* I = {p1, p2,…, pn}. Setiap ui *user* mengekspresikan pendapatnya tentang daftar *item* miliknya. Kumpulan set dari pendapat itu disebut dengan *rating* dari *user* ui dan dilambangkan dengan Iui. Setelah sistem ini menentukan ketetanggaan terdekat, maka sistem akan merepresentasikan *item* yang mungkin disukai *user* dalam dua bentuk, yaitu:

1. Prediksi, merupakan nilai numerik dimana Pa,j adalah nilai prediksi *rating item* j yang mungkin disukai oleh *active user* (Ua). Nilai prediksi ini digunakan dengan skala yang sama dengan nilai yang disediakan (misalnya, dari skala 1 sampai 5).
2. Rekomendasi adalah daftar N *item* yang mungkin akan disukai oleh *user* Ua. Daftar yang direkomendasikan biasanya terdiri dari *item* yang belum pernah dibeli atau di*rating* oleh *active user*. Output dari algoritma CF ini juga dikenal sebagai *Top-N Recommendation.*

Gambar 2.1 menunjukkan diagram skema dari proses *collaborative filtering*. Algoritma CF merepresentasikan seluruh m x n *user-item* sebagai matriks *rating* dimana setiap entri merupakan nilai *rating* dari *user* untuk setiap *item*. *Active user* (Ua) pada skema ini merupakan *user* yang akan dicari *item* yang mungkin disukainya dengan menggunakan algoritma CF (Sarwar, Badrul. 2001).

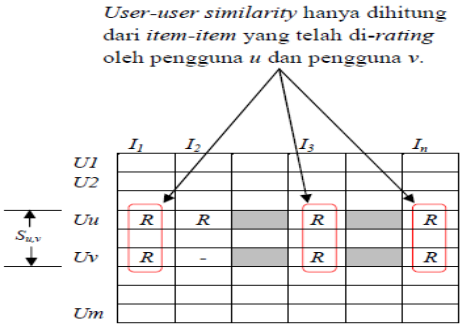
2.4 Metode *User-Based Collacborative Filtering*

User-based Collaborative Filtering menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (neighbour), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga

terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk user aktif. (Sarwar, Badrul. 2001).

Pada pendekatan *user based collaborative filtering* sistem memberikan rekomendasi kepada *user item-item* yang disukai atau *dirating* oleh *user-user* lain yang memiliki banyak kemiripan dengannya. Misalnya, *user* a menyukai atau *merating item* 1, 2 dan 3, kemudian *user* b menyukai *item* 1, 2 dan 4 maka sistem akan merekomedasikan *item* 3 kepada *user* b dan *item* 4 kepada *user* a.

Kelebihan dari pendekantan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Berikut merupakan skema *user based collaborative filtering*.



Gambar 2.2 Pola *user based collaborative filtering* (Sarwar, Badrul. 2001).

2.4.1 Algoritma *K-Means Clustering*

Algoritma yang digunakan pada proses pembentukan *user cluster* adalah algoritma *K-means.* Jumlah k merupakan *input* yang digunakan algoritma untuk menentukan seberapa banyak jumlah *cluster* yang ingin dibentuk. Adapun langkah-langkah pada *K-means clustering* adalah sebagai berikut (Xue, Gui-Rong. 2005) :

1. Tentukan beberapa k *user* yang dijadikan sebagai *centroid*.
2. Setiap *user* yang tidak menjadi *centroid* dibandingkan ke *centroid* terdekat berdasarkan nilai *similarity*.
3. Kalkulasi ulang nilai *centroid* berdasarkan rata-rata nilai kumpulan *user* di setiap *cluster* yang terbentuk.
4. Lakukan proses pembentukan ulang *cluster* dengan nilai *centroid* yang baru hingga nilai *centroid* stabil atau mendekati nilai *centroid* sebelumnya.

Nilai *similarity* antara *centroid user* dan *user* lainnya didefinisikan mengunakan *Pearson correlation-coefficient*, adalah sebagai berikut :

$$Sim_{u,u'} = \frac{\sum_{t \in T(u)\wedge T(u')} (R_u(t) - \overline{R_u}) \cdot (R_{u'}(t) - \overline{R_{u'}})}{\sqrt{\sum_{t \in T(u)\wedge T(u')} (R_u(t) - \overline{R_u})^2} \sqrt{\sum_{t \in T(u)\wedge T(u')} (R_{u'}(t) - \overline{R_{u'}})^2}} \quad (2.1)$$

$Sim_{u,u'}$ merupakan nilai *similarity* antara *user* centroid dan *user* lainnya.

$R_{u'}(t)$ merupakan nilai *rating centroid* untuk *item t.*

$\overline{R_{u'}}(t)$ merupakan rata-rata rating *user centroid.*

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

$\overline{R_u}(t)$ merupakan rata-rata rating *user u.*

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

2.4.2 Data *Smoothing*

Sparsity adalah masalah fundamental yang ada pada *collaborative filtering*, dengan menerapkan data *smoothing* pada sistem CF maka masalah ini dapat diminimalisir. *Smoothing* dilakukan dengan mengisi *missing value* pada *dataset* dengan *rating* bayangan. Berdasarkan hasil *clustering*, teknik *smoothing* data untuk *rating* bayangan didefinisikan sebagai penilaian khusus berikut (Xue, Gui-Rong. 2005) :

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

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

$$\hat{R}_u(t) = \overline{R_u} + \Delta R_{C_u}(t)$$

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

$$\Delta R_{C_u}(t) = \sum_{u' \in C_u(t)} \frac{(R_{u'}(t) - \overline{R_{u'}})}{|C_u(t)|}$$

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

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

2.5 *Mean Absolute Error* (MAE)

Mean Absolute Error merupakan persamaan yang termasuk jenis *statistical accuracy metrics*, dimana MAE merupakan *metrics* yang paling sering digunakan untuk pengukuran akurasi dari *recommender system*. Sesuai dengan namanya, MAE menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai sebenarnya (Xue, Gui-Rong. 2005). Nilai MAE sendiri, berkisar antara 0 hingga 1. Semakin kecil nilai MAE menandakan semakin akurat prediksi nilai *rating* dari sebuah *recommender system* (Xue, Gui-Rong. 2005).

$$MAE = \frac{\sum_{u \in T} |R_u(t_j) - \widetilde{R_u}(t_j)|}{|T|}$$

$R_u(t_j)$ merupakan nilai *rating item* t_j yang diberikan oleh *user u.*

$\widetilde{R_u}(t_j)$ merupakan nilai *rating* prediksi *user u* terhadap *item* t_j.

$|T|$ adalah jumlah data yang dikalkulasikan.

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dangan maksud untuk mengidentifikasikan dan mengevaluasikan permasalah-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan memahami sistem pemikiran yang kompleks dengan memecahnya ke dalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas.

Sebagai analisis pada sistem yang sedang berjalan, akan dibahas bagaimana prosedur-prosedur yang sedang atau yang sudah berjalan pada sistem terdahulu. Analisis sistem fungsional meliputi analisis kebutuhan data, dan pemodelan sistem yang nantinya akan dideskripsikan kedalam bentuk diagram dan, analisis sistem non fungsional yang meliputi analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak yang digunakan dan analisis kebutuhan *user*.

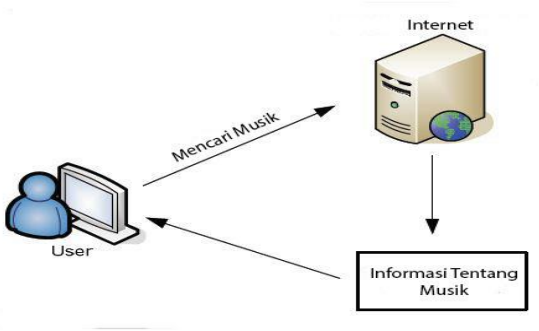
3.1.1 Analisis Masalah

Berdasarkan analisis sistem dari masalah pada penelitian ini maka analisis masalah yang ditemukan adalah sebagai berikut :

1. Seiring berkembangnya industri musik belakangan ini konsumen kini lebih cenderung membeli / mencari konten musik secara online dibandingkan pergi kesebuah toko.
2. Pencarian musik pada saat ini kebanyakan hanya dititik beratkan pada suatu *tagging* atau label saja.
3. Dengan berbedanya tingkat pengetahuan informasi mengenai musik dari setiap *user* maka dibutuhkanya suatu sistem rekomendasi yang nantinya dapat memberikan informasi musik secara personal.

3.1.2 Analisis Peneltian Sejenis.

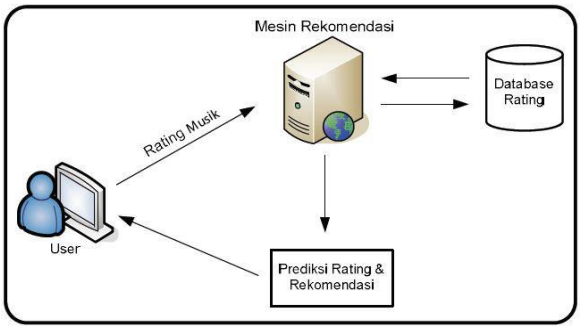
Industri musik belakangan ini mengalami perubahan yang sangat signifikan. Konsumen kini cenderung mengakses dan membeli konten secara online dibandingkan pergi ke sebuah toko. Contoh dari layanan berlangganan musik antara lain iTunes Radio, Grooveshark, Pandora, Spotify, dan Google Play Music All-Access. Untuk layanan berbasis lokal, Guvera menyediakan layanan serupa. Layanan - layanan tersebut memiliki kesamaan yaitu fitur pencarian lagu berdasarkan tagging atau label yang disimpan sebagai informasi tambahan pada setiap lagu dan dititik beratkan untuk fitur pencarian. (Saptariani, Trini. 2014) sehingga nantinya *user* akan mendapatkan informasi secara umum / tidak secara personal.



Gambar 3.1 Analisis penelitian sejenis.
(Saptariani, Trini. 2014).

3.1.3 Analisis Sistem Yang Akan Dibangun

Music Recommender System merupakan sebuah aplikasi berbasis web yang bertujuan untuk memberikan rekomendasi musik berdasarkan nilai rating yang mungkin akan disukai oleh *user*. Sistem ini juga memfasilitasi *user* untuk memberikan *rating* tertentu kepada musik yang ada di dalamnya. Oleh karena pemberian nilai *rating* dilakukan secara personal, maka musik yang direkomendasikan juga akan diberikan secara personal. Secara sederhana, seorang *user* sangat memungkinkan memiliki hasil rekomendasi yang berbeda dengan *user* lainnya.



Gambar 3.2 Alur Sistem Yang Akan Dibangun
(Saptariani, Trini. 2014).

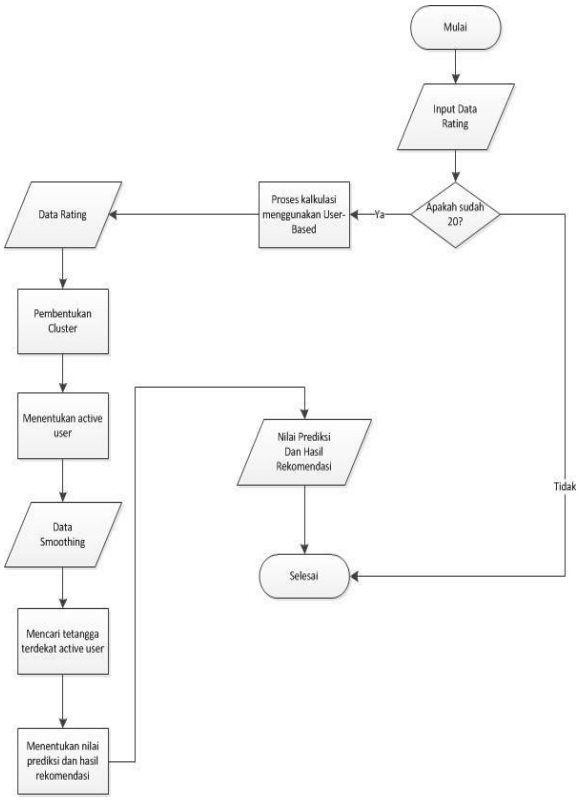
Pada dasarnya sistem ini memiliki data musik untuk di*rating* oleh *user*. Sedangkan data *rating* yang digunakan diperoleh dari *rating user* yang mendaftar ke dalam sistem. Berdasarkan *rating* tersebut, sistem akan melakukan perhitungan dengan memanfaatkan algoritma pada metode *User-Based Collaborative Filtering* untuk memberikan rekomendasinya.

Metode *User-Based Collaborative Filtering* mengelompokkan sekumpulan *user* yang memiliki kemiripan yang sangat tinggi dalam beberapa *cluster* dengan algoritma *K-means Clustering*. Tujuan dari penerapan algoritma *clustering* adalah untuk menangani masalah *scalability* yaitu, kondisi saat tingginya jumlah peningkatan *user* dan *item* di dalam *database rating* yang mempengaruhi menurunnya komputasi algoritma *Collaborative Filtering*, metode

ini memanfaatkan proses *smoothing* untuk mengurangi masalah *sparsity*, yaitu terjadinya kekosongan data matriks *user-item* yang disebabkan karena *user* me*rating* dalam jumlah kecil dari jumlah item yang tersedia didatabase. Untuk menghitung tingkat akurasi dan kualitas dari *music recommender system* ini adalah dengan menggunakan perhitungan MAE (*Mean Absolute Error*). Sesuai dengan namanya MAE menghitung selisih nilai antara nilai prediksi dengan nilai sebenarnya (Xue, Gui-Rong. 2005).

3.1.4 Analisis Metode *User-Based Collaborative Filtering*

Penerapan metode *User-Based Collaborative Filtering* dilakukan dalam beberapa tahap. Penerapan *User-Based Collaborative Filtering* dapat dilihat pada Gambar 3.3 berikut ini :



Gambar 3.3 *Flowchart* Penerapan *User-Based Collaborative Filtering*

3.1.5 Pemodelan Sistem

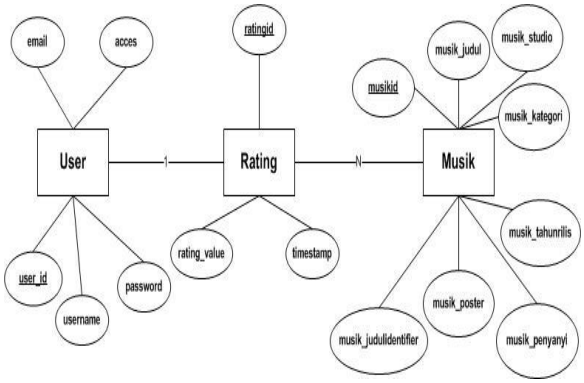
Pada tahap ini dilakukan pemodelan sistem untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

3.1.6 Analisis Basis Data

Basis data atau sering juga disebut *database* merupakan salah satu komponen penting dalam sistem informasi terkomputerisasi. Analisis basis data digunakan untuk memodelkan dan menghubungkan relasi antar data.

3.1.6.1 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan secara sistematis berbagai entitas dan komponen data yang dimiliki sistem dan hubungan antar masing-masing entitas tersebut. Untuk melihat keterhubungan antar entitas yang ada maka digambarkan sebagai berikut :



Gambar 3.4 *Entity Relationship Diagram*

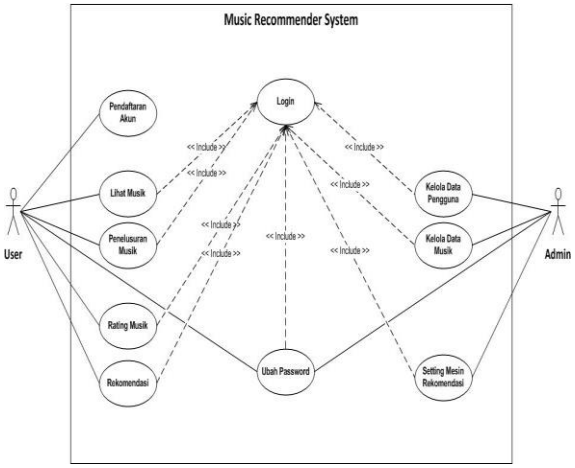
A. *Use Case* Diagram

Diagram *Use Case* merupakan deskripsi fungsi dari sebuah sistem dan perspektif pengguna. Diagram ini juga mendeskripsikan apa yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case*.

1) Identifikasi Aktor

Untuk mengindetifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada sistem. Aktor dalam diagram *use case* aplikasi *music recommender* ini terdiri dari dua aktor yaitu *user* dan admin. *User* merupakan aktor yang menggunakan aplikasi *frontend*, dimana aktor dapat berinteraksi dengan sistem seperti melakukan pendaftaran akun, melihat detail musik, menelusuri daftar musik, ubah *password*, merating musik, dan memperoleh hasil rekomendasi. Sedangkan admin merupakan aktor yang dapat mengelola data pengguna dan data musik, ubah *password*, serta dapat mensetting jumlah *cluster* pada sistem.

2) Identifikasi *Use Case*

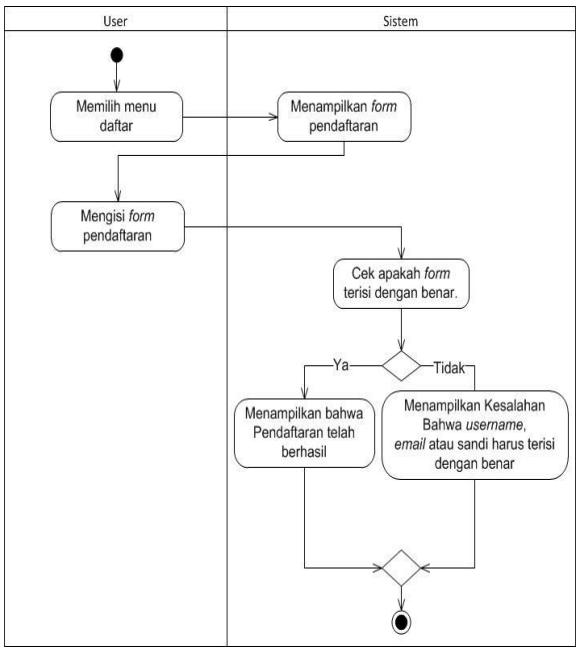


Gambar 3.5 *Use Case Diagram*

B. *Activity Diagram*

Activity Diagram merupakan suatu diagram yang mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* memiliki peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku pararel sedangkan *flowchart* tidak bisa. *Activity Diagram* memodelkan *event-event* yang terjadi didalam suatu *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

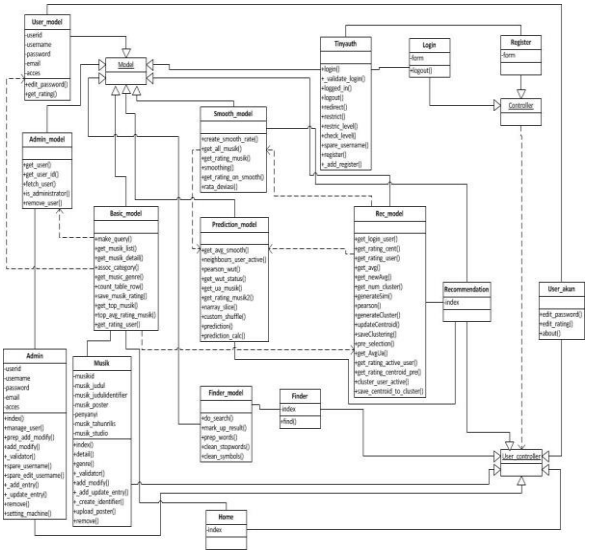
Activity diagram Pendaftaran Akun menjelaskan aktifitas aktor dalam mendaftar / masukan data *user* kedalam sistem. *Activity diagram* Pendafrataran Akun dapat dilihat pada gambar 3.6. Berikut ini :



Gambar 3.6 *Activity Diagram* Pendaftaran Akun

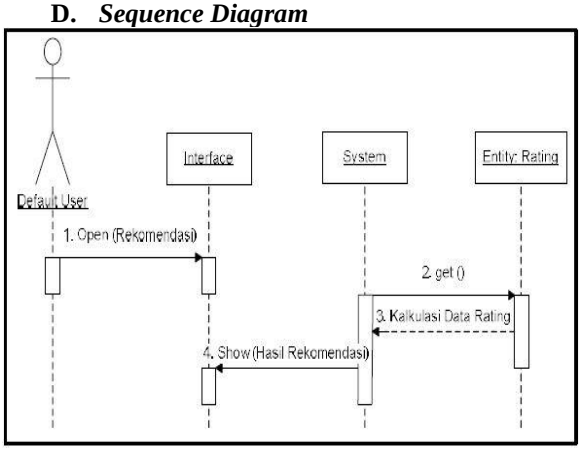
C. *Class Diagram*

Class Diagram menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem tersebut. *Class Diagram* bersifat statis di dalam *class diagram* digambarkan relasi dari masing - masing *class* tetapi tidak menggambarkan apa yang terjadi ketika *class* tersebut berelasi.



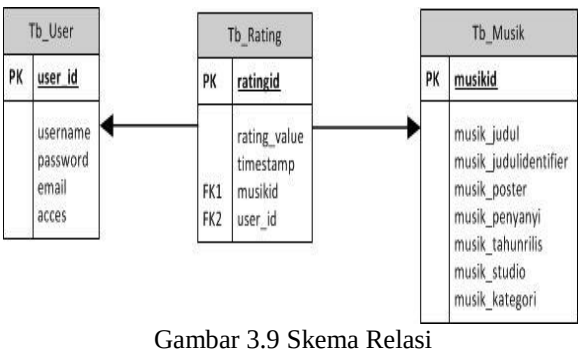
Gambar 3.7 *Class Diagram*

D. *Sequence Diagram*



Gambar 3.8 *Sequence Diagram* Rekomendasi

3.1.7.2 Skema Relasi



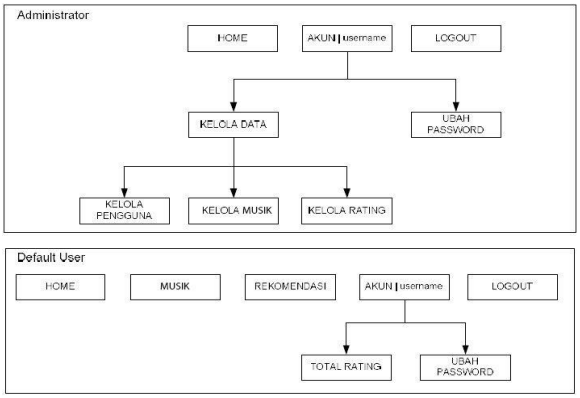
Gambar 3.9 Skema Relasi

3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan *database*, dan perancangan antarmuka.

3.2.1 Struktur Menu

Pada *music recommender system* ini, menu terdapat pada halaman setelah pengguna masuk melalui proses verifikasi, berikut ini merupakan struktur menu pada sistem yang dapat dilihat pada Gambar 3.10 berikut ini :

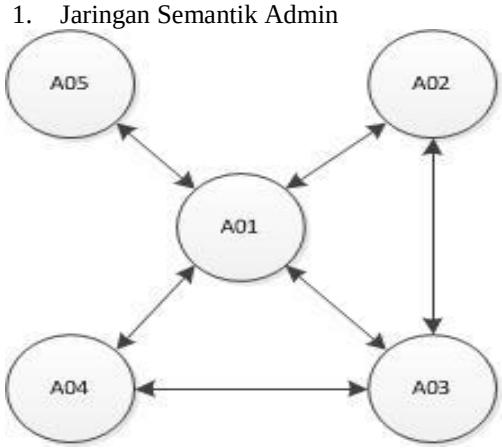


Gambar 3.10 Struktur Menu

3.2.2 Jaringan Semantik

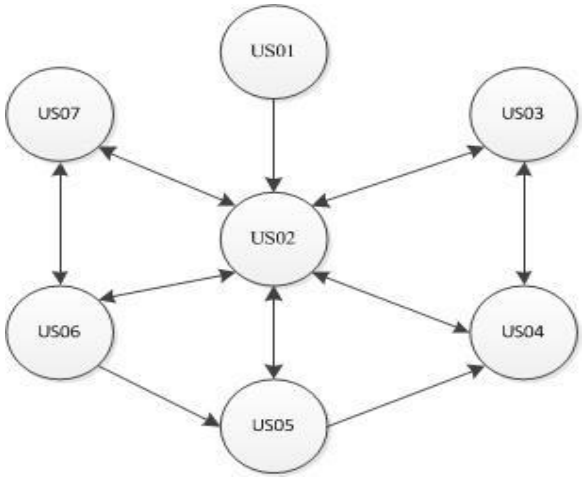
Jaringan semantik merupakan jaringan data dan informasi, yang menunjukan hubungan antar berbagai objek. Berikut adalah jaringan semantik dari *music recommender system* :

1. Jaringan Semantik Admin



Gambar 3.11 Jaringan Semantik Admin

2. Jaringan Semantik User



Gambar 3.12 Jaringan Semantik *User*

4.1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk mengimplementasikan program aplikasi yang dibangun dapat dilihat dalam Tabel 4.1 berikut ini :

Tabel 4.1 Kebutuhan perangkat keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel i3
2	Monitor	Monitor 15 inch
3	Memory	DDR3 1Gb
4	Keyboard	Standar
5	Mouse	Standar
6	Modem	Standar

3.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan aplikasi ini dapat dijelaskan pada Tabel 4.2. berikut ini :

Tabel 4.2 Kebutuhan perangkat lunak

No	Perangkat Lunak	Yang Digunakan
1	Sistem Operasi	Windows 7
2	Bahasa Pemrograman	PHP
3	Database Server	MySQL
4	Web Browser	Google Chrome dan Mozila Firefox
5	Code Editor	Adobe Dreamweaver CS 6 dan Notepad ++

4.3 Pengujian

4.3.1 Pengujian Menggunakan *Blackbox*

Tahap pengujian sistem dilakukan untuk mengetahui apakah hasil implementasi perangkat lunak telah berjalan sesuai dengan yang diharapkan.

4.3.1.1 Kesimpulan Hasil Pengujian Sistem Menggunakan *Blackbox*

Berdasarkan hasil pengujian sistem yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi yang dibangun sudah berjalan sesuai dengan perancangan awal dan berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan, pada saat aplikasi digunakan, baik itu kesalahan pada perangkat yang digunakan, kesalahan pengguna, maupun kesalahan-kesalahan lainnya. Sehingga membutuhkan proses perawatan dan pengecekan *(maintenance)* untuk menjaga agar aplikasi tetap berjalan sesuai dengan yang diharapkan.

4.3.2 Pengujian Tingkat Akurasi Menggunakan MAE

Pengujian tingkat akurasi dilakukan dengan mengambil sampel data sebanyak 1.063 contoh *rating* dari total *rating* yang sudah dikumpulkan yaitu berjumlah 1.403 contoh. Data yang dijadikan *sample* dalam penelitian ini harus memenuhi kriteria yaitu data *rating* berasal dari *user* yang telah merating minimal sebanyak 20 musik dengan batas maksimal *rating* setiap *user* adalah 30 musik.

4.3.2.1 Rencana Pengujian Tingkat Akurasi Menggunakan MAE

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. Persamaan yang gunakan untuk menghitung tingkat akurasi adalah Persamaan (2.9). Berikut ini merupakan prosedur pengujian tiap-tiap parameter yang digunakan :

1. Berdasarkan jumlah *cluster*
   Proses pengujian dilakukan untuk setiap *cluster* yang telah ditentukan, yaitu berjumlah 2 *cluster*, 3 *cluster* dan 4 *cluster*. Jumlah *k-neighbors* yang digunakan untuk setiap pengujian parameter *cluster* adalah sebanyak 50%.
2. Berdasrkan jumlah *K-neighbours*
   Tahap pengujian dilakukan dengan menentukan persentase jumlah tetangga terdekat dari *active user*. Jumlah *k-neighbors user* yang diuji coba diambil dari *user-user* yang memiliki tingkat *similarity* yang besar dengan *active user* yaitu sebanyak 30%, 50%, dan 70%. Jumlah *cluster* yang digunakan adalah sebanyak 3 *cluster*.
3. Berdasarkan tingkat *sparsity*
   Pengujian pada *sample* data akan dipilih data *rating* untuk dikosongkan secara acak dengan tingkat *sparsity* sebanyak 30%, 50% dan 70%. Masalah *sparsity* merupakan masalah yang kerap kali terjadi pada *collaborative filtering*, untuk itu melalui pengujian ini akan dilihat seberapa besar pengaruh pengosongan nilai pada data *rating* atau keadaan data *sparse* terhadap nilai akurasi prediksi sistem. Pengujian diset dengan jumlah *cluster* 3 dan jumlah k sebanyak 50%.

4.3.2.2 Hasil Pengujian Tingkat Akurasi

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

1) Berdasarkan jumlah *cluster*

Tabel 4.4 MAE berdasarkan jumlah *cluster*

Jumlah *Cluster*	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 4.4 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 *active user* terlalu sedikit. Pada pengujian ini, jumlah seluruh *user* yang akan dicarikan *clusternya* adalah sebanyak 40 *user*. Jumlah anggota dari tiap-tiap *cluster* cenderung selalu berbeda, yang menyebabkan ada kalanya seorang *active user* masuk ke dalam sebuah *cluster* dengan jumlah anggota yang sedikit. Misalnya *active user* masuk menjadi anggota dari *cluster-x* dengan total anggota hanya sebanyak 5 *user*. Logikanya adalah, jika anggota suatu *cluster* hanya terdapat 5 *user* maka pada proses pemilihan jumlah tetangga terdekat *active user* hanya ada kandidat di antara 5 *user* tersebut.

Kesimpulan dari pengujian jumlah *cluster* adalah pemilihan jumlah *cluster* pada penerapan *collaborative filtering* berbasis *user-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 *active user* dan proses prediksi musik mendapat keakuratan yang baik dengan nilai MAE mendekati 0.

2) Berdasarkan *K-Neighbours*

Tabel 4.5 MAE berdasarkan *K-Neighbours*

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 4.5 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 *active user*. Penyebab dari banyaknya kandidat tersebut dikarenakan pengujian ini mengasumsikan jika nilai *similarity* antara *active user* 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 *similarity* < 0 mengindikasi bahwa *user* tersebut memiliki nilai kesamaan yang kecil terhadap *active user*.

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* musik = 0 yang dimasukan kedalam perhitungan prediksi.

3) Berdasarkan Tingkat *Sparsity*

Tabel 4.6 MAE berdasarkan tingkat *sparsity*

Tingkat *Sparsity*	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 4.6 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 *user*. Meskipun demikian, masalah kekosongan data pada *collaborative filtering* dapat teratasi dengan menerapkan teknik *smoothing* pada sistem yang dibangun.

Ada kalanya nilai *similarity* antara *active user* dan *user-user* 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.

4.3.2.3 Analisa Hasil Pengujian

Pengujian tingkat akurasi menunjukkan *music recommender system* berbasis *user-based collaborative filtering* yang dibangun mampu mengatasi masalah kekosongan data dengan tingkat *sparsity* sebanyak 70%, dan pengujian tingkat akurasi lainnya dapat disimpulkan bahwa *music recommender system* akan dibangun dengan pengaturan jumlah cluster = 3, dan jumlah *knearest neighbors* sebanyak 50% karena pengujian yang dilakukan pada kedua pengaturan ini menghasilkan rata-rata nilai MAE yang lebih rendah daripada yang lainnya yaitu sebesar 0,6713.

3. PENUTUP

5.1 Kesimpulan

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

1. Untuk mempermudah pengguna dalam menemukan musik yang dicari, maka pengguna dapat menggunakan fitur pencarian yang ada atau dapat melihat dari hasil jumlah rating pengguna itu sendiri serta dapat juga dengan melihat daftar 10 musik terpopuler ataupun hasil rating musik tertinggi.
2. Hasil akurasi yang dicapai pada pengujian dengan metode ini tergolong kurang akurat karena dengan menerapkan jumlah hasil terkecil yaitu jumlah *cluster* = 3 dan jumlah *k* (jumlah *user* yang menjadi tetangga terdekat dari *active user*) = 50% dapat menghasilkan rata-rata MAE = 0,6713 artinya lebih dari 0.5 untuk mendekati nilai 0, karena MAE sendiri memiliki rentang nilai dari 0 – 1.

5.2 Saran

Beberapa hal yang disarankan untuk pengembangan penelitian ini adalah sebagai berikut :

1. Berdasarkan hasil penelitian ini maka disarankan untuk menggunakan metode lain guna menghasilkan prediksi musik yang lebih baik dengan nilai MAE mendekati 0 atau tidak lebih dari 0.5.
2. Prosedur pengujian dapat dilakukan dengan cara yang berbeda, misalnya menambahkan jumlah *dataset*, dan meningkatkan jumlah pengujian dari pengujian pada penelitian ini. Hal ini akan mempengaruhi tingkat akurasi dari *recommender system* yang akan dibangun.

3. Metode ini dapat dicoba pada penelitian dengan item lain, seperti buku, berita pada koran digital, film, dan lainnya dengan pola pengumpulan *rating* yang serupa dengan pengujian ini.

DAFTAR PUSTAKA

[1] Agusta, Indika Satriyana. (2013). *Perbandingan efektifitas metode user-based collaborative filtering dengan metode user-item based collaborative filtering*. (Skripsi). Universitas Sebelas Maret, Surakarta.

[2] Erlangga. (2011). *Modul Kuliah Rekayasa Perangkat Lunak. Jurusan Teknik Informatika*. UNIKOM, Bandung.

[3] J, Durkin. (1994). *Expert System Design And Development. Prentice*. Hall International Edition. New Jersey: Macmilan Publishing Company.

[4] Jogiyanto, HM. (2005). *Analisis & Desain Sistem Informasi : Pendekatan Terstruktur Teory dan Praktek Aplikasi bisnis*. Yogyakatra: ANDI.

[5] Kartadinata, Sunaryo. (2014). *Pedoman Penulisan Karya Ilmiah UPI Tahun 2014*. Bandung: Universitas Pendidikan Indonesia.

[6] Kusumadewi, Sri. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.

[7] Leimstoll, U. & Stormer, H. (2007). *Collaborative Recommender Systems for Online Shops*. Journal: AMCIS 2007, Keystone, CO

[8] McGinty, L. & B. Smyth. (2006). *Adaptive selection: analysis of critiquing and preference based feed back in conversation on recommender system*. International J Electron Commerce.

[9] Mortensen, Magnus. (2007). *Design and Evaluation of a Recommender System*. University of Tromso.

[10] Myer, Thomas. (2008). *Professional CodeIgniter*. Wiley Publishing.

[11] Pazzani, Michael J. & Billsus, Daniel. (2007). *Content-Based Recommendation Systems*. Springer-Verlag Berlin Heidelberg.

[12] Pressman, Roger S. (2001). *Software Engineering : A Practitioner's Approach*. McGraw- Hill Companies, Inc.

[13] Sanjung, Ariyani. (2011). *Perbandingan Semantic Classification dan Cluster-based Smoothed pada Recommender System berbasis Collaborative Filtering*. (Skripsi). Teknik Informatika, Universitas Telkom, Bandung.

[14] Saptariani, Trini. (2014). *Sistem Rekomendasi Musik Menggunakan Latent Semantic Analysis*. (Skripsi). Teknik Informatika, Universitas Gunadarma, Depok.

[15] Sarwar, Badrul. (2001). *Item-Based Collaborative Filtering Algorithms*. Minneapolis : University of Minnesota.

[16] Septian, Gungun. (2011). *Trik Pintar Menguasai CodeIgniter*. Jakarta: Elex Media Komputindo.

[17] Shalahuddin, Muhammad & Rosa Ariani S. (2011). *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.

[18] Twoh & co, Sekilas tentang sistem rekomendasi. [Online] Diakses dari http://www.twoh.co/2013/05/sekilas-tentang-sistem- rekomendasi-recommender-system/ (Diakses tanggal 19 Mei 2015)

[19] Wang, Jun. (2006) *Unfiying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion.* Amsterdam: Delft University Of Technology.

[20] Wiranto & Winarko, Edi. (2010). *Konsep Multicriteria Collaborative Filtering Untuk Perbaikan Rekomendasi*. Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta

[21] Xue, Gui-Rong. (2005). *Journal: Scalable Collaborative Filtering Using Cluster-based Smoothing*, Brazil