

# Analisis dan Implementasi Efficient Sequencing pada Hybrid Clustering (K-Means dan Fuzzy C-Means) pada Recommender System

Amru Rizaldy Sandiagro<sup>1</sup>, Ade Romadhony<sup>2</sup>, Alfian Akbar Gozali<sup>3</sup>

Fakultas Informatika Institut Teknologi Telkom, Bandung

[amru\\_rs@y7mail.com](mailto:amru_rs@y7mail.com)<sup>1</sup>, [ade.romadhony@gmail.com](mailto:ade.romadhony@gmail.com)<sup>2</sup>, [panggil.aku.ian@gmail.com](mailto:panggil.aku.ian@gmail.com)<sup>3</sup>

---

## Abstrak

Sistem rekomendasi merupakan sebuah aplikasi/program yang mencoba untuk melakukan prediksi ataupun melakukan pemberian rekomendasi atas sebuah *item* terhadap pengguna berdasarkan informasi yang didapatkan dari pengguna. Pendekatan yang paling umum digunakan dalam membangun sistem rekomendasi adalah *collaborative filtering* dan *content-based recommendation*. Masing-masing pendekatan tersebut memiliki kelemahannya masing-masing.

Oleh karena itu dilakukanlah *hybrid recommendation system* yang melakukan kombinasi dari *collaborative filtering* dan *content-based recommendation* yang diharapkan dapat meminimalisir kekurangan yang terdapat pada satu pendekatan saja. Terdapat dua buah metode untuk melakukan *hybrid recommendation system*, yaitu *linear combination* dan *sequential combination*. Pada metode *sequential combination* dilakukan penggabungan antara *collaborative filtering* yang kemudian dilanjutkan dengan *content-based recommendation*.

Pada Tugas Akhir ini akan diimplementasikan *hybrid recommender system* yang menggunakan metode *efficient sequencing* yang menggabungkan *content-based recommendation* dan *collaborative filtering* dengan menggunakan algoritma *Fuzzy C-Means* dan *K-Means clustering*[10] untuk kasus dataset *MovieLens*. Dengan menggabungkan kedua pendekatan tersebut yang berbasis algoritma *clustering*, maka penggabungan algoritma *clustering* tersebut dikenal dengan *hybrid clustering*.

Dengan menggunakan metode *efficient sequencing* yang menerapkan algoritma *Fuzzy C-Means clustering* dan *K-Means clustering* pada *hybrid recommender system* diharapkan dapat menghasilkan prediksi yang lebih akurat dibandingkan dengan jika menggunakan satu pendekatan saja. Nantinya hasil prediksi akan dievaluasi dengan menggunakan *Mean Absolute Error* (MAE).

**Kata Kunci :** *Hybrid Recommender System, Collaborative Filtering, Content-based Recommendation, Efficient Sequencing*

---

## Abstract

*Recommender system is an application/program that trying to predict or give a recommendation from an item to a user, based on information that retrieved from the user. The common approach that often used in developing recommender system is collaborative filtering and content-based recommendation. Every approach have its own weakness.*

*Therefore, hybrid recommendation system exist, and it combine collaborative filtering and content-based recommendation, can be expected to minimize the drawbacks in every approach. There is two method to do hybrid recommendation system, that is linear combination and sequential combination. In sequential combination method, the combine is doing one after another. First, content-based recommendation, and then continued with collaborative filtering.*

*In this final project, I will implement hybrid recommender system that using efficient sequencing that combine content-based recommendation and collaborative filtering with using Fuzzy C-Means clustering algorithm and K-Means clustering algorithm for MovieLens Dataset. With combine the two approach based on clustering algorithm, so we can called the combine for the two algorithm as hybrid clustering.*

*With using efficient sequencing method that applied Fuzzy C-Means clustering and K-Means clustering algorithm in hybrid recommender system, hope that the result from the recommendation system is more accurate than if using one approach. Next, the prediction will be evaluated using Mean Absolute Error (MAE).*

**Key Word :** *Hybrid Recommender System, Collaborative Filtering, Content-based Recommendation, Efficient Sequencing*

---

## 1. Pendahuluan

Sistem rekomendasi merupakan sebuah aplikasi/program yang mencoba untuk melakukan prediksi ataupun melakukan pemberian rekomendasi

atas sebuah *item* terhadap pengguna berdasarkan informasi yang didapatkan dari pengguna[4]. Sistem rekomendasi melakukan rekomendasi dengan

mencari kemiripan yang didapatkan pada suatu *item* dengan *item* lainnya.

Empat metode dasar yang dapat digunakan untuk melakukan rekomendasi pada sistem rekomendasi adalah *demographic filtering*, *collaborative filtering*, *content-based recommendation*, dan *simplified statistical approach*[5]. Dua pendekatan yang sering digunakan untuk membuat sistem rekomendasi adalah *collaborative filtering* dan *content-based recommendation*[13].

*Hybrid recommender system* adalah suatu metode penggabungan lebih dari satu pendekatan recommender system, misalnya *collaborative filtering* dan *content-based recommendation*. *Efficient sequencing* adalah metode penggabungan secara *sequential* terhadap pendekatan *collaborative filtering* dan *content-based recommendation*. *Hybrid clustering* adalah metode penggabungan algoritma *clustering* yang akan diimplementasikan pada *hybrid recommender system*. Pada tugas akhir ini, akan menggunakan *efficient sequencing* yang menggunakan algoritma *Fuzzy C-Means Clustering* dan *K-Means Clustering* yang mengacu pada *paper*[10]. Metode ini awalnya akan melakukan *collaborative filtering* yang akan diterapkan dengan menggunakan *K-Means Clustering*, yang kemudian akan dilanjutkan dengan *content-based recommendation* yang akan diterapkan dengan menggunakan *Fuzzy C-Means Clustering*. *Item* yang menjadi objek rekomendasi adalah film yang telah *dirating* oleh pengguna yang diperoleh dari dataset *MovieLens*.

Pada *Efficient Sequencing*, akan dilakukan *collaborative filtering* yang kemudian hasilnya akan dilanjutkan pada *content-based recommendation*. Sehingga diharapkan hasil prediksi yang dihasilkan akan lebih akurat dan dapat menangani permasalahan yang terjadi jika hanya menggunakan satu pendekatan saja.

## 2. Tinjauan Pustaka

### 2.1 Recommender System

*Recommender system* adalah sebuah program yang mencoba untuk memprediksi sebuah *item* berdasarkan informasi yang diperoleh dari pengguna [4]. Cara pencarian *item* yang akan direkomendasikan nantinya dapat dilakukan dengan mencocokkan kemiripan pengguna yang satu dengan yang lain berdasarkan *rating* yang diberikan pada sebuah *item*.

Dalam proses pengumpulan data yang akan digunakan dalam sistem rekomendasi, dibedakan menjadi dua cara (Alex Lin, 2009), yaitu :

#### a. Eksplisit

1. *Rating* yang diberikan pengguna terhadap suatu *item*
2. Komentar yang diberikan pengguna terhadap suatu *item*

#### b. Implisit

1. List *item* yang diklik oleh pengguna
2. List *item* yang dilihat oleh pengguna

### 2.2 Collaborative Filtering

Ide dasar dari *collaborative filtering recommendation* adalah mengeksplorasi informasi yang telah dikumpulkan dari masa lalu tentang kebiasaan yang sering dilakukan oleh untuk memprediksi sesuatu yang biasanya paling disukai oleh pengguna [5]. *Collaborative filtering* bekerja dengan membangun sebuah *database* dari berbagai jenis *item* yang dipilih oleh pengguna.

Contoh dari *collaborative filtering* pada kehidupan sehari-hari adalah adalah *top ten* lagu yang kita lihat di web ataupun televisi, ataupun *bestseller list*.

Beberapa algoritma yang dapat digunakan untuk melakukan *collaborative filtering* adalah *ranking based collaborative filtering*, *memory based*, atau *probabilistic model-based collaborative filtering*.

*Collaborative filtering* mencoba untuk memprediksikan pendapat dari pengguna yang memiliki *item* berbeda dan yang direkomendasikan paling baik berdasarkan hal yang disukai sebelumnya dan pendapat dari pengguna lain yang serupa.

Beberapa tantangan dalam *collaborative filtering* adalah [8]:

- *Sparsity*. Dimana kesulitan untuk mengevaluasi sejumlah *item set* yang sangat besar, dan memiliki tingkat kecenderungan kurang dari 1%
- *Cold-Start* : *Item* yang tidak dapat direkomendasikan meskipun telah di-rate oleh beberapa *user*, dikarenakan *item* ini termasuk baru.
- *Scalability*. Nearest neighbor membutuhkan komputasi yang semakin lama semakin berkembang seiring dengan perkembangan jumlah pengguna dan jumlah *item*

### 2.3 Content-Based Recommendation

Secara umum, *recommender system* mungkin saja menyajikan dua tujuan yang berbeda. Di satu sisi, sistem dapat digunakan untuk membantu pengguna untuk melakukan sesuatu, anggap saja membeli sebuah buku. Di sisi yang lain, *recommender system* dapat juga dilihat sebagai *tools* yang berhadapan dengan *information overload*, misalnya adalah menentukan untuk menampilkan barang yang paling menarik dari data yang besar [5]. Pada konteks *content-based recommendation*, beberapa hal yang harus dapat dijawab adalah : Bagaimana sistem dapat secara rutin dan tepat mengetahui kebutuhan pengguna? Bagaimana menentukan *item* yang tepat, atau setidaknya mirip dengan yang diinginkan oleh pengguna?

Umumnya, jika kita membuat sebuah web yang menggunakan basis data, maka kita akan menyimpan isi konten sebuah postingan pada atribut basis data tersebut. Namun, biasanya kita juga mendapatkan sebuah teks yang mendefinisikan informasi yang lengkap seperti deskripsi objek barang atau review barang. Pada penyimpanan basis data, informasi tersebut dapat disimpan pada field tambahan di database. Namun informasi yang biasanya bersifat teks bebas tersebut memberikan sejumlah tantangan baru ketika mempelajari profil pengguna. Mudah-mudahan saja misalkan sebuah profil mungkin menunjukkan ada 70% kecocokan di dalamnya, maka hal ini akan menjadikan *item* tersebut akan direkomendasikan oleh sistem karena menunjukkan adanya positive review. Hal ini dapat terjadi karena pada text bebas yang tidak memiliki batasan, tidak memiliki nama atribut yang mendefinisikannya secara tegas.

Meskipun terdapat banyak pendekatan yang berbeda untuk mempelajari tingkat kecenderungan yang *user* sukai dengan menggunakan *content-based recommendation*, namun tidak ada *content-based recommendation* system yang dapat memberikan rekomendasi yang baik ketika konten yang terdapat tidak memiliki informasi yang cukup untuk membedakan *item* yang disukai dan yang tidak disukai oleh pengguna.

Kekurangan pada *content-based recommendation* system :

- Memberikan rekomendasi yang kurang tepat jika informasi yang didapatkan dari dokumen yang merepresentasikan suatu *item* terlalu singkat, sehingga membuat sulit untuk mengklasifikasikannya
- *New User Problem*. Maksudnya adalah untuk pengguna baru, jika ingin mendapatkan rekomendasi yang lebih akurat, pengguna tersebut harus terlebih dahulu memberikan sejumlah *rating*
- *Over-specialization*. Pengguna mungkin saja direkomendasikan *item* yang mirip dengan *item* yang telah di *rated* sebelumnya

## 2.4 Hybrid Recommender System

Kesulitan dalam *content-based recommendation* adalah dalam menghasilkan rekomendasi yang belum ada sebelumnya karena semua informasi dan rekomendasi pada *content-based recommendation* didapatkan dari konten *item* nya. Selanjutnya akan sulit bagi pengguna baru untuk mendapatkan rekomendasi yang efektif karena pengguna baru tersebut belum melakukan *rating* terhadap *item* dan akan sulit bagi sistem untuk mengidentifikasi minat pengguna. Sedangkan pada pendekatan *collaborative filtering* akan mengalami kesulitan untuk

merekomendasikan *item* baru kepada pengguna karena *item* baru yang belum di-*rate*[8].

Maka dari itu ditemukanlah teknik *hybrid recommender system*, yang ide dasarnya adalah dengan menggabungkan teknik *collaborative filtering* dengan *content-based recommendation* [8]. Sistem akan melakukan rekomendasi kepada pengguna berdasarkan *rating item*, yang akan digabungkan dengan konten dari *item* tersebut, maka diharapkan kualitas dari rekomendasi yang diberikan akan meningkat.

### 2.4.1 Sequential Combination

*Sequential combination* melakukan penggabungan antara *content-based* dan *collaborative filtering*. Jadi pada *sequential combination* akan dilakukan dua pendekatan secara berurutan. Pada Tugas Akhir ini akan dilakukan *hybrid recommender system* yang menggunakan metode *Sequential Combination* dengan *efficient sequencing*. Pada sistem *efficient sequencing* yang akan diaplikasikan ini akan mengaplikasikan pendekatan *collaborative filtering* terlebih dahulu untuk menemukan pengguna yang memiliki tingkat *interest* yang mirip. Kemudian setelah itu akan melakukan *content-based recommendation*.

Pada Tugas Akhir ini akan dilakukan kombinasi dari tingkat *interest* pengguna dengan konten data untuk meningkatkan kualitas prediksi dan menangani masalah *cold start problem*. filtering.

Untuk mengaplikasikan *collaborative filtering* akan menggunakan algoritma K-Means clustering. Hal ini dilakukan untuk menemukan kemiripan pengguna yang kemudian akan ditemukan item yang memiliki rate tinggi pada user yang memiliki karakteristik sama. Penggunaan K-Means clustering dilakukan pada data user karena pada data yang akan diolah yaitu data user bersifat diskrit. Sehingga nantinya setiap user hanya termasuk ke dalam 1 cluster saja.

Sedangkan untuk mengaplikasikan *content-based recommendation* akan menggunakan algoritma Fuzzy C-Means clustering. Algoritma ini berfungsi untuk mengelompokkan item ke dalam beberapa cluster untuk kemudian dicari kesamaan dengan item yang lainnya. Penggunaan Fuzzy C-Means clustering dilakukan pada data item yang memiliki zona abu-abu sehingga nantinya setiap item memiliki kemungkinan termasuk ke dalam beberapa cluster dengan derajat keanggotaan yang berbeda.

Prosedur yang dilakukan dengan pendekatan *efficient sequencing* adalah sebagai berikut :

- Menemukan kemiripan dari pengguna menggunakan algoritma *K-means clustering*
- Menemukan *item* pada *user* yang terdapat pada *cluster* yang sama, yang memiliki *rate* tinggi

- c. Mengaplikasikan algoritma *fuzzy C-means* untuk menemukan konten pada *cluster* yang sama dengan yang di-*request*
- d. Menemukan *common set* antara 2 koleksi set pada tahap b dan c

## 2.5 K-Means Clustering

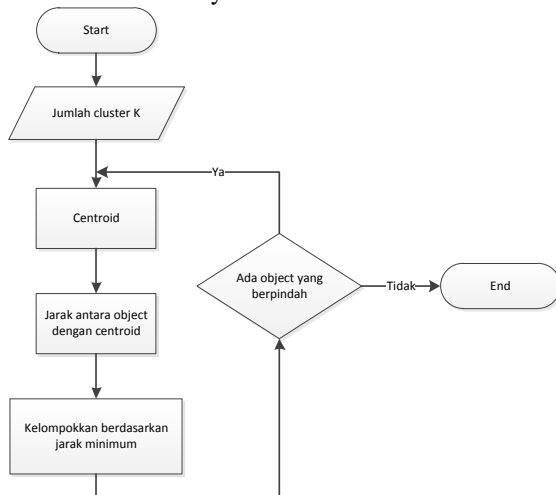
Algoritma K-means adalah sebuah algoritma untuk mengidentifikasi K *cluster* dari sejumlah data [2]. Asumsi kita memiliki sejumlah N data  $\{x_1, x_2, \dots, x_N\}$  dimana  $x_i \in \mathbb{R}^D$ , dengan tujuannya adalah memperoleh K group dari data point yang memiliki intra-*cluster* minimal dengan *Euclidean* untuk setiap pusat *cluster*,  $\mu_i$ :

$$J = \sum_{n=1}^N \sum_{i=1}^K a_{n,i} \|x_n - \mu_i\|^2 \quad (1)$$

Dimana  $a_{n,i}$  adalah variable biner yang mengindikasikan penetapan *cluster*. Jika  $x_n$  ditetapkan untuk *cluster* j, maka  $a_{n,i} = 1$  dan  $a_{n,m} = 0$ , untuk  $m \neq j$

Langkah-langkah dasar dalam melakukan k-means *clustering* adalah sebagai berikut :

- a. Kita harus menentukan beberapa faktor yang salah satunya adalah jumlah *cluster* (K)
- b. Tentukan centroid dari *cluster* tersebut
- c. Kemudian selanjutnya adalah melakukan iterasi sampai tidak ada lagi object yang berpindah grup. Dalam iterasi tersebut dilakukan beberapa algoritma :
  1. Tentukan centroid dari koordinat
  2. Tentukan jarak antara masing-masing objek dengan centroid
  3. Kelompokkan object berdasarkan jarak minimal nya



Gambar 2.1 Algoritma K-Means Clustering

## 2.6 Fuzzy C-Means Clustering

Fuzzy C-means adalah sebuah metode *clustering* yang mengijinkan sebuah data memiliki *cluster* lebih dari satu. FCM menggunakan model pengelompokkan *fuzzy* sehingga data dapat menjadi anggota dari semua kelas atau *cluster* terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1. Tingkat keberadaan data dalam suatu kelas atau *cluster* ditentukan oleh derajat keanggotaannya. Metode FCM ini biasanya digunakan pada pengenalan pola [12]. Algoritma ini dapat dirumuskan dengan fungsi :

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad (2)$$

Dengan  $m$  adalah bilangan yang memiliki nilai lebih dari 1, Bezdek mengesetnya dengan 2.00.  $u_{ij}$  adalah derajat keanggotaan dari  $x_i$  pada *cluster* j, dimana  $x_i$  adalah  $i^{th}$  dari data yang terukur, dan  $\|*\|$  merupakan ekspresi yang menyatakan kemiripan antara data. Algoritma diatas digunakan untuk menghitung fungsi objektif.

Langkah-langkah dalam algoritma Fuzzy C-Means adalah sebagai berikut :

- Tentukan parameter-parameter dalam Fuzzy C-Means.
- Bangkitkan bilangan random sebagai matriks partisi awal  $u_{ik}(i=1..n; k=1..c)$
- Hitunglah pusat cluster
- Hitung fungsi objektif
- Perbaharui matrix partisi
- Cek kondisi berhenti, jika  $P_0 - P_1 < e$  atau iter > maxIter

## 2.7 Mean Absolute Error

*Mean Absolute Error* adalah suatu jenis pengukuran yang umumnya digunakan untuk mengevaluasi kualitas dari sistem rekomendasi. *Mean absolute error* termasuk dalam jenis *statistical accuracy metrics*, yang mengukur akurasi sistem dengan cara membandingkan nilai yang dikeluarkan oleh sistem rekomendasi yang dibangun dengan nilai yang sebenarnya. Pada *mean absolute error* ini menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai yang sebearnya (George Karypis, 2001).

Berikut adalah persamaan *mean absolute error*:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (3)$$

MAE = Nilai rata-rata kesalahan hitung

N = Jumlah *item* yang dihitung

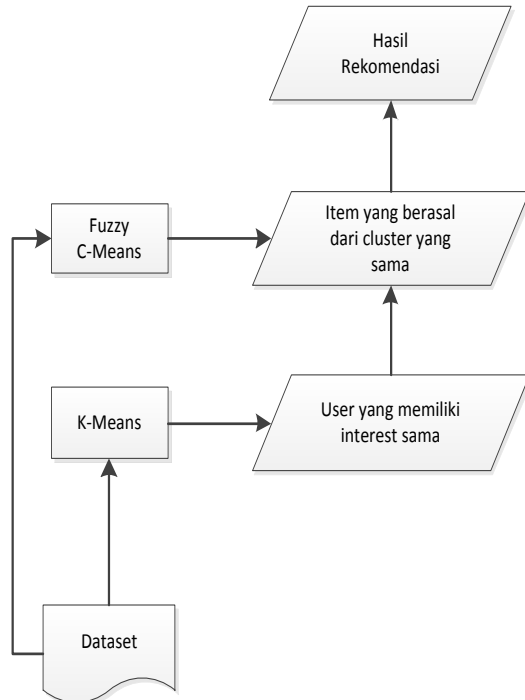
$p_i$  = Nilai prediksi *item* ke-i

$q_i$  = Nilai *rating* sebenarnya *item* ke-i

Maka berdasarkan rumus tersebut dapat terlihat bahwa semakin kecil nilai MAE, berarti semakin akurat sistem dalam memprediksi *rating*

### 3. Perancangan

#### 3.1 Gambaran Umum



**Gambar 3.1 Rancangan Umum Sistem**

Sistem rekomendasi yang akan dibangun, pada awalnya akan melakukan proses *K-Means clustering* serta *Fuzzy C-Means Clustering*. Pada masing-masing proses *clustering* tersebut akan mengolah dataset untuk mendapatkan tingkat kemiripan *user* yang menggunakan *K-Means Clustering*, serta mendapatkan kemiripan *item* dengan menggunakan *Fuzzy C-Means Clustering*. Masing-masing hasilnya kemudian akan dimasukkan ke dalam database untuk digunakan kemudian. Data tersebut akan digunakan untuk menghitung tingkat kemiripan *user* dengan menggunakan database *K-Means* yang telah dibangun, dan selanjutnya akan dilihat juga *movie* yang di-*rate* oleh *user* terdapat pada *cluster* mana. Setelah dihitung, akan digabungkan hasilnya untuk kemudian diberikan kepada *user* sebagai hasil dari rekomendasi sistem.

#### 4. Hasil Pengujian

Berikut adalah tabel hasil MAE yang didapatkan dari hasil testing terhadap *user* yang dikategorikan dalam 3 kategori :

**Tabel 4.1 Hasil Perbedaan MAE dengan Kategori Jumlah Item**

Kategori User	MAE
---------------	-----

Sedikit (<80 item)	0.836024
Rata-rata (80 - 150 item)	0.817012
Banyak (>150 item)	0.837586

Kemudian pada pengujian selanjutnya ini akan memperlihatkan perbandingan hasil MAE dari *recommender system* yang mengimplementasikan *hybrid clustering* dengan hasil MAE dari *recommender system* yang menggunakan *collaborative filtering*.

Setelah dilakukan percobaan, maka berikut hasil dari MAE kedua *recommender system* :

**Tabel 4.2 Perbandingan MAE Hybrid Recommender System dengan Collaborative Filtering**

Kategori User	MAE	
	Hybrid Recommender	Collaborative Filtering
Sedikit (<80 item)	0.83602368	0.82406404
Rata-rata (80 - 150 item)	0.81701155	0.81121056
Banyak (>150 item)	0.83758637	0.81511890

### 5. Kesimpulan dan Saran

#### 5.1 Kesimpulan

1. *Hybrid recommender system* yang menggabungkan *K-Means Clustering* dengan *Fuzzy C-Means Clustering* belum tentu memiliki hasil MAE yang lebih baik jika dibandingkan dengan *recommender system* yang menggunakan *collaborative filtering*
2. Semakin baik setiap proses *clustering* yang dilakukan pada *hybrid recommender system*, maka akan semakin baik pula hasil dari *recommender system* yang didapatkan
3. Jumlah *rating* yang diberikan oleh *user* tidak menjamin akan menghasilkan nilai MAE yang lebih baik. Oleh karena itu jumlah *rating* yang diberikan oleh *user* tidak berpengaruh terhadap nilai MAE yang dihasilkan

#### 5.2 Saran

1. Untuk mendapatkan hasil rekomendasi dengan *hybrid recommender system* yang lebih baik dapat dengan memperbaiki proses *clustering* yang dilakukan, oleh karena itu dapat mencari algoritma *clustering* lain yang lebih cocok menangani tipe data yang sesuai dengan yang didapatkan

2. Untuk mempercepat proses komputasi, dapat menggunakan bahasa yang dapat menangani data besar dalam waktu yang lebih cepat seperti Matlab

#### Daftar Pustaka

- [1] Burke, Robin. (2002). *Hybrid Recommender System : Survey and Experiments*. Kluwer Academic Publishers.
- [2] Derpanis, Konstantinos G. March 5, 2006. *K-Means Clustering*.
- [3] Gong, SongJie. (2010). *Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering*.
- [4] [http://digilib.ittelkom.ac.id/index.php?option=com\\_content&view=article&id=707:recsystem&catid=20:informatika&itemid=14](http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=707:recsystem&catid=20:informatika&itemid=14)
- [5] Jannach, Dietmar and Zanker, Markus. (2011). *Recommender System An Introduction*.  
<<http://www.cambridge.org/9780521493369>>
- [6] Kazienko, Przemyslaw and Kolodziejwski, Pawel. (2006). *Personalized Integration of Recommendation Methods for E-commerce*. International Journal of Computer Science & Applications Vol. 3 Issue 3, pp 12-26
- [7] Li, Qing and Kim, Byeong Man. *An Approach for Combining Content-based and Collaborative Filters*.
- [8] Li, Qing and Kim, Byeong Man. *Clustering Approach for Hybrid Recommender System*.
- [9] [Movielens.umn.edu](http://movielens.umn.edu). (24 April 2012).  
<<http://movielens.umn.edu/html/tour/index.html>>
- [10] Nayak, Richi and Mital Namita. (21 Maret 2008). *A Hybrid Clustering Based Filtering Approach with Efficient Sequencing*.
- [11] Park, Seung-Taek and Pennock David. *Naïve Filterbots for Robust Cold-Start Recommendations*.
- [12] Ramini, Abdullah and Alata, Mohanad. (2008). *Optimizing of Fuzzy C-Means Clustering Algorithm Using GA*.
- [13] Spiegel, Stephan and Li, Fang. *Hydra : A Hybrid Recommender System*.
- [14] Uchyigit, Gulden and Y.Ma, Matthew. (2008). *Personalization Techniques and Recommender Systems*.