Sistem Rekomendasi Buku dengan Metode Berbasis *Clustering*

Hilmi Eko Arianto[1], Dade Nurjanah, S. T. , M. T. , Ph. D.[2], Rita Rismala, S. T. , M. T[3]

[123]Fakultas Informatika, Universitas Telkom, Bandung
[1]hilmiea@students. telkomuniversity. ac. id, [2]dadenurjanah@telkomuniversity. ac. id,
[2]ritaris@telkomuniversity. ac. id

Abstrak

Metode *collaborative filtering* adalah metode populer yang digunakan untuk sistem rekomendasi dengan berbagai macam domain. Pada domain buku, metode tersebut menggunakan *rating* yang diberikan user terhadap buku. Tetapi ada kekurangan terhadap metode tersebut dikarenakan harus mempertimbangkan semua buku yang ada untuk proses rekomendasi. Karena harus mempertimbangkan keseluruhan buku, maka membutuhkan waktu yang lebih lama untuk melakukan rekomendasi. *Clustering* adalah salah satu cara untuk mengatasi kekurangan metode *collaborative filtering*. Metode ini akan mengelompokkan buku berdasarkan kemiripan *user*, sehingga proses rekomendasi tidak perlu mempertimbangkan keseluruhan buku. Kebanyakan metode berbasis *clustering* harus mengetahui berapa jumlah kelompok buku yang akan digunakan. Karena tidak memiliki jumlah kelompok buku sebelumnya, *self-constructing clustering* dapat digunakan jika data yang digunakan tidak memiliki jumlah kelompok. Pada tugas akhir ini, dilakukan studi tentang implementasi metode berbasis *clustering* dengan algoritma *self-constructing clustering*. Algoritma ini akan mengelompokkan buku berdasarkan kemiripan *user* tanpa mengetahui jumlah kelompok buku yang ada. Hasil pengujian menunjukkan bahwa metode dengan algoritma tersebut dapat digunakan hingga merekomendasikan buku kepada *user* pada data yang hanya berupa data *user*, buku, dan *rating*. Pengujian dilakukan dengan menggunakan 2 data. Hasil pengujian menghasilkan DOA dan MAE sebesar 50% dan 1.10283, serta pada data kedua didapatkan 56% dan 1.137.

Abstract

Collaborative filtering method is a popular method used for recommendation systems with various domains. In the book domain, the method uses the rating that the user gives to the book. But there are disadvantages to the method because they have to consider all the books available for the recommendation process. Having to consider the whole book, it will take longer to make a recommendation. Clustering based is one way to overcome the lack of collaborative filtering methods. This method will group books according to user resemblance, so the recommendation process does not need to consider the entire book. Most clustering based methods must know how many groups of books will be used. Because it does not have the number of previous book groups, self-constructing clustering can be used if the data used has no number of groups. In this final project, a study about clustering based method implementation with self-constructing clustering algorithm. This algorithm will group the book based on the user's similarity without knowing the number of existing book groups. The test results show that the method with the algorithm can be used up to recommend the book to the user on the data only in the form of user data, books, and rating. Testing is using 2 data. The test results produced DOA and MAE of 50% and 1.10283, and the second data obtained 56% and 1,137.

1. Pendahuluan

Latar Belakang

Pada zaman sekarang sistem rekomendasi sudah diterapkan pada berbagai macam domain seperti musik, film, buku dan produk lainnya berdasarkan kesukaan atau ketidaksukaan pengguna terhadap suatu produk. Toko buku *online* seperti *Amazon* telah memiliki layanan rekomendasi buku yang populer untuk merekomendasikan

buku sesuai dengan selera pembaca [9]. Umumnya rekomendasi buku disesuaikan dengan selera pembaca dapat menggunakan infomasi antara pembaca dan buku, yaitu berupa *rating* yang diberikan pembaca terhadap buku atau menggunakan kategori yang ada pada buku, seperti Agama, Sosial, Umum dan lain sebagainya [3].

Selama beberapa tahun terakhir, sistem rekomendasi dengan berbagai macam domain telah banyak dikembangkan termasuk dalam domain buku. Salah satu metode yang sering digunakan pada sistem rekomendasi adalah metode *Collaborative Filtering* [3]. Metode *Collaborative Filtering* menggunakan data *rating* yang diberikan oleh *user* terhadap buku. Tetapi ada beberapa kekurangan yang ada pada metode *Collaborative Filtering*. Pertama adalah *New User Problem*, karena *user* baru belum pernah memberikan *rating* terhadap buku sehingga tidak dapat diberikan rekomendasi. Kedua adalah *New Item Problem*, karena buku yang tidak pernah diberikan *rating* oleh *user* tidak dapat direkomendasikan. Ketiga adalah waktu dalam melakukan rekomendasi tidak efisien, karena metode ini harus mempertimbangkan keseluruhan data buku [3].

Namun, salah satu kekurangan pada metode *Collaborative Filtering* yaitu tidak efisiennya waktu untuk melakukan rekomendasi sudah dapat diatasi. Para peneliti mengatasi permasalahan tersebut dengan menggunakan teknik reduksi. Teknik reduksi ini memiliki manfaat untuk memperkecil dimensi data pada data yang besar, sehingga setelah data diperkecil, akan membuat sistem rekomendasi lebih efisien waktu dalam proses rekomendasi karena tidak harus mempertimbangkan keseluruhan data buku dan tanpa mengurangi kualitas rekomendasi [3].

Metode lainnya adalah dengan penggunaan metode *clustering* seperti algoritma *K-means* [11]. Penelitian oleh (Ba, 2013) mengelompokkan *user* ke dalam kelompok sesuai dengan atribut, misalnya jenis kelamin, pekerjaan, dan usia. Kemudian peringkat produk pengguna dibentuk ke dalam matriks dan digabungkan menjadi matriks peringkat baru untuk menghitung kesamaan dua *user*. Penelitian lainnya [2], menerapkan algoritma *fuzzy* untuk membentuk kelompok *user*. Walupun memiliki kelebihan, metode-metode tersebut memiliki kekurangan yaitu jumlah kelompok yang akan digunakan sudah diketahui jumlah kelompoknya, sehingga tidak dapat diimplementasikan pada data yang tidak memiliki jumlah kelompok [3].

Topik dan Batasannya

Metode yang digunakan pada tugas akhir ini adalah metode berbasis *clustering* dengan algoritma *self-constructing clustering*. Pada tugas akhir ini dilakukan studi tentang implementasi metode tersebut pada domain buku hingga menghasilkan rekomendasi. Lalu berdasarkan data latih dan uji, akan dihitung DOA dan MAE untuk mengukur kesalahan yang ada.

Terdapat beberapa batasan yang ada pada tugas akhir ini yaitu tidak seluruh data pada domain buku yang digunakan. Untuk dataset *Book Crossing* hanya menggunakan data buku yang diberikan *rating* minimal oleh 20 user, dan user yang memberikan *rating* minimal 20 buku. Pada dataset *Good Books* hanya menggunakan data buku yang diberikan *rating* minimal oleh 80 *user*, dan *user* yang memberikan *rating* minimal 100 buku. Untuk hasil rekomendasi, hanya merekomendasikan 10 buku dengan nilai rekomendasi terbesar.

Tujuan

Tujuan dari tugas akhir ini adalah untuk mengimplementasikan metode berbasis *clustering* dan algoritma *self-constructing clustering* pada sistem rekomendasi buku dan menguji performansi algoritma tersebut dengan mengukur DOA beserta MAE pada data yang digunakan. Uji coba dilakukan 2 kali, yaitu untuk dataset *Book Crossing* dan *Good Books*.

Organisasi Tulisan

Urutan penulisan laporan ini adalah sebagai berikut : bagian 2 menunjukkan penelitian terkait penelitian ini. Sistem yang diajukan untuk Sistem Rekomendasi Buku menggunakan Metode berbasis *Clustering* akan dijelaskan pada bagian 3. Pada bagian 4 akan ditunjukkan hasil pengujian dan evaluasi sistem. Akhirnya, kesimpulan akan dipaparkan pada bagian 5.

2. Studi Terkait

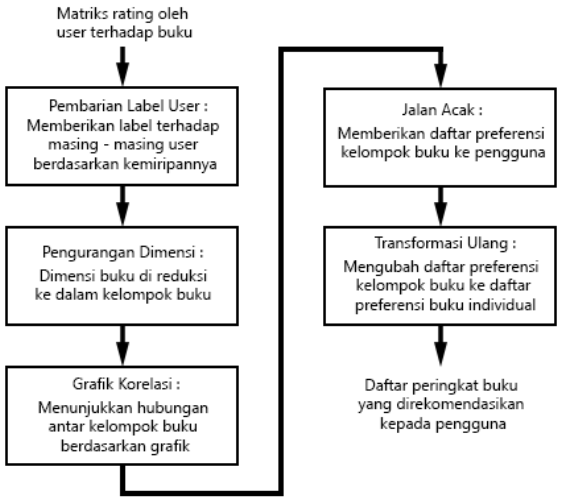
Banyak metode yang pernah digunakan dalam sistem rekomendasi pada domain buku. Metode *ItemRank* adalah salah satu metode *Collaborative Filtering* yang dapat digunakan dalam sistem rekomendasi pada domain buku. Akurasi yang didapatkan dengan menggunakan metode tersebut diukur dengan menghitung *Degree Of Aggrement* (DOA) dan *Mean Absolute Error* (MAE). DOA dan MAE yang didapatkan adalah sebesar 0.715 dan 1.210.

Telah dijelaskan di latar belakang bahwa metode *Collaborative Filtering* memiliki kekurangan yaitu kurang efisien waktu dalam proses rekomendasi. Hal tersebut dapat diatasi dengan salah satu teknik yaitu teknik reduksi. *Clustering* adalah salah satu teknik reduksi yang dapat digunakan dalam sistem rekomendasi pada domain buku. Salah satu algoritma *clustering* yang dapat digunakan dalam sistem rekomendasi buku adalah algoritma *self-constructing clustering*. Algoritma ini memiliki kelebihan dibandingkan dengan algoritma *clustering* lain yaitu tidak memerlukan berapa jumlah *cluster* awal yang harus digunakan pada proses rekomendasi, sehingga sangat

cocok diterapkan pada *dataset* buku yang hanya memiliki data *user*, buku dan *rating*. Algoritma tersebut sudah berhasil diterapkan pada sistem rekomendasi domain buku, dan hasil dari pengukuran akurasinya jauh lebih baik dibandingkan dengan metode *Collaborative Filtering* [3].

Metode yang digunakan pada tugas akhir ini adalah metode berbasis *clustering* dengan algoritma *self-constructing clustering*. Terdapat 5 tahapan pada metode yang akan digunakan, pemberian label *user*, pengurangan dimensi, membuat grafik korelasi, jalan acak, dan transformasi ulang [3]. Metode tersebut pernah diimplementasikan pada sistem rekomendasi dengan data *MovieLens, Yahoo Movie, dan Amazon Video*. Dan pada hasil akhir menunjukkan bahwa metode *clustering* dengan algoritma *self-constructing clustering* lebih cepat dan efisien dalam waktu eksekusi dibanding dengan metode *Collaborative Filtering* [3]. Tingkat keakuratan rekomendasi untuk data *MovieLens* dan *Amazon Video* juga mendapatkan nilai tertinggi.

Pada metode berbasis *clustering* dengan algoritma *self-constructing clustering*, langkah pertamanya yaitu pemberian label *user*, digunakan algoritma *self-constructing clustering* untuk memberikan label terhadap masing - masing *user* berdasarkan kemiripannya. Pada langkah kedua, pengurangan dimensi, *self-constructing clustering* digunakan kembali dimana buku yang serupa dikelompokkan dalam kelompok yang sama dan buku yang beda akan dikelompokkan ke dalam kelompok yang berbeda [3]. Karena jumlah kelompok buku jauh lebih kecil dari pada jumlah buku, dimensi yang terlibat jauh lebih berkurang. Kemudian grafik korelasi akan menunjukkan hubungan antar kelompok buku yang dihasilkan pada langkah ketiga. Berdasarkan grafik korelasi, jalan acak di eksekusi dan daftar preferensi kelompok buku diturunkan untuk masing - masing *user* dilangkah keempat. Akhirnya, dalam transformasi ulang, preferensi daftar kelompok buku diubah ke dalam daftar preferensi buku individual, dan daftar peringkat buku akan direkomendasikan kepada setiap *user* [3].



Gambar 1. Tahapan dalam metode berbasis *clustering* dengan algoritma *self-constructing clustering*

2.1. Self-Constructing Clustering

Diberikan himpunan X dari n pola $x_1, x_2, \dots, x_n$, dengan $x_i = x_{i1}, x_{i2}, \dots, x_{ip}$ untuk $1 \leq i \leq n$. Tujuan dari algoritma *self-constructing clustering* adalah untuk mengelompokkan pola tersebut ke dalam kumpulan kelompok, dimana pola yang serupa dikelompokkan dalam kelompok yang sama, dan kelompok yang beda dikelompokkan dengan yang berbeda juga [3]. Anggap K adalah kelompok yang ada sekarang, dinamakan sebagai $G_1, G_2, \dots, dan\ G_K$. Setiap kelompok G_j memiliki nilai rata – rata $m_j = m_{j1}, m_{j2}, \dots, m_{jp}$ dan deviasi $\sigma_j = \sigma_{j1}, \sigma_{j2}, \dots, \sigma_{jp}$ yang mewakili nilai rata – rata dan standar deviasi. Awal mula $K = 0$ dimiliki yang menandakan bahwa belum memiliki kelompok sama sekali. Untuk setiap $x_i, 1 \leq i \leq M$ akan dihitung derajat keanggotaan dari x_i menggunakan :

$$\mu_{G_j}(x_i) = \prod_{q=1}^{p} exp\left[-\left(\frac{x_{iq}-m_{jq}}{\sigma_{jq}}\right)^2\right] \quad (1)$$

untuk $1 \leq j \leq K$. Proses perhitungan kesamaan pada kelompok G_j bisa dilewati jika $\mu_{G_j}(x_i) \geq \rho$ dimana $\rho, 0 \leq \rho \leq 1$. Asumsikan bahwa x_i tidak memiliki kemiripan satupun dengan kelompok yang ada, maka x_i tersebut membuat kelompok baru dengan $m = x_i$ dan $\sigma = \sigma_0$ dan hal tersebut berlaku juga jika ada kelompok baru yang

terbentuk. Lalu beranggapan bahwa x_i memiliki kemiripan dengan G_t, maka $m_t$dan σ_t pada kelompok G_t harus diperbaharui berdasarkan x_i yang masuk ke kelompok tersebut dengan :

$$\sigma_{tj} = \sqrt{A - B} + \sigma_0, \tag{2}$$

$$A = \frac{(S_t-1)(\sigma_{tj}-\sigma_0)^2+S_t\, x\, m_{tj}^2+x_{ij}^2}{S_t}, \tag{3}$$

$$B = \frac{S_t+1}{S_t}\left(\frac{S_t\, x\, m_{tj}+x_{ij}}{S_t+1}\right)^2, \tag{4}$$

$$m_{tj} = \frac{S_t\, x\, m_{tj}+x_{ij}}{S_t+1} \tag{5}$$

untuk $1 \leq j \leq p$, dan

$$S_t = S_t + 1 \tag{6}$$

dimana S_t adalah banyak anggota kelompok pada kelompok t.

2.2. Pemberian Label *User*

Untuk melakukan pengurangan dimensi, sebelumnya perlu menetapkan label kelas untuk *user*. Caranya adalah dengan mengelompokkan *user* kedalam kelompok [Sarwat 2014]. *User* serupa dikelompokkan ke dalam kelompok yang sama, dan *user* yang berbeda dikelompokkan ke dalam kelompok yang berbeda. Lalu semua *user* diberi label kelas yang unik sesuai dengan kelompoknya. Algoritma *self-constructing clustering* digunakan untuk tujuan tersebut. Anggap rating yang diberikan oleh user terhadap produk adalah $x_i = x_{i1}, x_{i2}, \dots, x_{iM}$, $1 \leq i \leq N$, dan $X = \{x_i | 1 \leq i \leq N\}$.
Diterapkan algoritma *self-constructing clustering* pada X. Misalkan z cluster $G_1, G_2, \dots, G_z$ diperoleh. Setiap cluster dianggap sebagai kelas, dan dimiliki z kelas, lalu diberi label sebagai $c_1, c_2, \dots, c_z$ masing – masing untuk semua user yang ada. Misalkan contoh data yang digunakan seperti :

Tabel 1. Data *user*, buku, dan *rating* yang diberikan oleh *user* terhadap buku

	Buku 1	Buku 2	Buku 3	Buku 4	Buku 5
User 1	3	5	8	7	5
User 2	4	2	5	9	0
User 3	0	0	0	7	6
User 4	0	0	0	8	2

Lalu menghitung kesamaan antar *user* dan pengelompokkan user dengan menggunakan algoritma self-contructing clustering. Setelah algoritma tersebuh diterapkan, data diatas akan menjadi :

Tabel 2. Data *user*, *buku*, dan *rating* yang diberikan oleh *user* terhadap buku setelah diberi label

	Buku 1	Buku 2	Buku 3	Buku 4	Buku 5	Kelas
User 1	3	5	8	7	5	c_1
User 2	4	2	5	9	0	c_1
User 3	0	0	0	7	6	c_2
User 4	0	0	0	8	2	c_2

2.3. Pengurangan Dimensi

Tahap ini bertujuan untuk memperkecil dimensi buku dengan cara pengeelompokkan buku. Sehingga pada tahapan berikutnya, kelompok buku akan digunakan untuk proses rekomendasi. Dari data yang digunakan, matriks yang dimiliki adalah sebesar NxM. Dimensi M dari atribut buku akan diperkecil [7]. Untuk setiap buku $p_j, 1 \leq j \leq M$. akan dibangun *feature pattern* $x_j = x_{j1}, x_{j2}, \dots, x_{jz}$ dengan

$$x_{jk} = P(c_k|p_j) = \frac{\sum_{d=1}^{N} r_{dj} x \delta_{dk}}{\sum_{d=1}^{N} r_{dj}}, 1 \leq k \leq z \tag{7}$$

Untuk $1 \leq j \leq M$, dimana δ_{dk} didefinisikan sebagai

$$\delta_{dk} = \begin{cases} 1, \ jika \ u_d = c_k \\ 0, \ jika \ u_d \neq c_k \end{cases} \quad (8)$$

Lalu, akan didapatkan M *feature patterns* $x_1, x_2, \dots, x_M$ yang memiliki z komponen, Misalkan $Y = \{x_i | 1 \leq i \leq M\}$. Selanjutnya diterapkan algoritma *self-constructing clustering* pada Y. Anggap memiliki q kelompok, $G_1, G_2, \dots, G_q$. Perlu diketahui bahwa buku yang terkandung didalam kelompok mirip satu sama lainnya. Oleh sebab itu memungkinkan untuk menggunakan kelompok untuk mewakili semua buku yang terkandung didalam kelompok tersebut. Karena memiliki q kelompok buku, data *user* dengan M komponen dapat diganti dengan data baru dengan q komponen [3]. Dengan cara ini, dapat mengurangi dimensi buku M ke dimensi q yang lebih rendah. Biarkan T menjadi matriks yang telah diperkecil :

$$T = \begin{bmatrix} t_{11} \ t_{12} \dots t_{1q} \\ t_{21} \ t_{22} \dots t_{2q} \\ \vdots \ \ \vdots \ \ \vdots \ \ \vdots \\ t_{M1} t_{M2} \cdots t_{Mq} \end{bmatrix} \quad (9)$$

dimana

$$t_{ij} = \mu_{G_j}(x_i) \quad (10)$$

$t_{ij} = \mu_{G_j}(x_i)$ adalah tingkat keanggotaan x_i dikelompok G_j. Untuk $1 \leq i \leq M$ dan $1 \leq j \leq q$. Lalu dimensi tinggi R yang mana NxM matriks akan diubah menjadi matriks dimensi rendah B dari :

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = RT = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} T \quad (11)$$

yang merupakan $N \ x \ q$ matriks. Perhatikan bahwa

$$B_i = [b_{i1}, b_{i2}, \dots, b_{iq}] \quad (12)$$

Untuk $1 \leq i \leq N$. Lalu B didapatkan, dan dapat memanggil setiap kolom dalam B adalah suatu kelompok buku. Jadi q kelompok buku didapatkan dan dinamakan sebagai $G_1, G_2, \dots, G_q$. Dengan kelompok buku tersebut, data user dengan komponen M diubah, masing – masing komponen terkait dengan satu buku menjadi data *user* baru dengan q komponen dan masing – masing komponen sesuai dengan satu kelompok buku [3].

2.4. Grafik Korelasi

Pada langkah ini, akan dibuat grafik korelasi berdasarkan B yang akan menunjukkan hubungan antar kelompok buku q [6]. Setiap kelompok buku dianggap sebagai *node* dalam grafik, dengan demikian memiliki q *node* dalam grafik. Anggap W adalah grafik korelasi yang akan dibuat dan berbentuk matriks, dimana W memiliki bobot w_{ij}. Bobot w_{ij} antara *node* g_i dan *node* g_j, $1 \leq i, j \leq q$, dihitung dari :

$$w_{ij} = \begin{cases} 0, \ jika \ i = j \\ \sum_{k=1}^{N} limit \left(\frac{b_{ki}}{b_{kj}}\right), \ jika \ i \neq j \end{cases} \quad (13)$$

dimana

$$limit \left(\frac{a_1}{a_2}\right) = \begin{cases} 0, jika \ a_1 = 0 \ atau \ a_2 = 0 \\ \frac{a_1}{a_2}, jika \ a_1 < a_2 \\ 1, jika \ dalam \ keadaan \ yang \ lain \end{cases} \quad (14)$$

Ketika grafik korelasi selesai, matriks korelasi yang didapatkan adalah seperti berikut ini :

$$W = \begin{bmatrix} W_{11} W_{12} \dots W_{1q} \\ W_{21} W_{22} \dots W_{2q} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ W_{q1} W_{q2} \cdots W_{qq} \end{bmatrix} \tag{15}$$

yang merupakan matriks $q\ x\ q$. Lalu setiap kolom W akan dinormalisasi, yaitu dengan :

$$Q_j = \sum_{k=1}^{q} w_{kj}, \tag{16}$$

$$w_{ij} = \frac{w_{ij}}{Q_i}, 1 \le i \le q \tag{17}$$

$$untuk\ 1 \le j \le q \tag{18}$$

2.5. Jalan Acak

Pada tahap ini, jalan acak [4][5][8][12] digunakan untuk menurunkan daftar preferensi kelompok buku kepada *user*. Lalu akan didapatkan V_i. V_i merupakan vektor yang didapatkan setelah konvergen tercapai pada tahap jalan acak, dimana vektor tersebut daftar preferensi kelompok buku yang diutamakan untuk *user* u_i. Mempertimbangkan setiap *user* u_i, $1 \le i \le N$. Biar $V_i(0)$ menjadi

$$V_i(0) = [\tfrac{1}{q}\ \ \tfrac{1}{q}\ \dots\ \tfrac{1}{q}]^T \tag{19}$$

yang merupakan vektor dengan ukuran q. Operasi berikutnya

$$V_i(t+1) = \alpha W V_i(t) + (1-\alpha) B_i^T \tag{20}$$

Dilakukan berulang kali untuk $t = 0,1,2,\dots$ hingga konvergen tercapai. Matriks W adalah matriks korelasi grafik, B_i adalah R yang telah diperkecil, dan $\alpha \in [0,1]$ konstanta yang ditentukan oleh *user*. Berdasarkan [8] α adalah sebesar 0.85. Untuk operasi diatas, dilakukan sebanyak 20 kali agar konvergen tercapai untuk setiap *user*. Setelah itu didapatkan V_i sebagai vektor yang telah tercapai konvergennya, yang memiliki q komponen. Lalu V_i adalah preferensi kelompok buku yang diturunkan untuk *user* $u_i, 1 \le i \le N$.

2.6. Transformasi Ulang

Pada tahapan sebelumnya, didapatkan V_i yaitu preferensi kelompok buku. Setiap vektor $V_i, 1 \le i \le N$ diperoleh untuk *user* u_i pada tahapan jalan acak yang berisi nilai sebesar q. Karena harus merekomendasikan buku secara individu bukan secara kelompok, maka perlu mengubah V_i menjadi S_i yang berisi preferensi buku individu [3]. Ingat pada tahapan pengurangan dimensi, derajat keanggotaan dari $x_j, 1 \le j \le M$ di $G_1, G_2, \dots, dan\ G_q$ adalah $t_{j1} = \mu_{G1}(x_j)$, $t_{j2} = \mu_{G2}(x_j)$,…, dan $t_{jq} = \mu_{Gq}(x_j)$. Pertama dilakukan normalisasi setiap kolom T yang didapatkan pada tahapan pengurangan dimensi. Normalisasikan dengan cara

$$Q_k = \sum_{j=1}^{M} t_{jk}, \tag{21}$$

$$t_{jk} = \frac{t_{jk}}{Q_k}, 1 \le j \le M \tag{22}$$

untuk $1 \le k \le q$. Untuk setiap baris, nilai proporsi buku p_j untuk setiap kelompok buku dihitung. Lalu didapatkan

$$S_i[j] = t_{j1} V_i[1] + t_{j2} V_i[2] + \cdots + t_{jq} V_i[q] \tag{23}$$

dimana $S_i[j]$ adalah komponen ke-j S_i dan $V_i[k], 1 \le k \le q$ adalah komponen dari V_i. Perhatikan bahwa t_{jk} adalah proporsi buku p_j yang sudah dihitung sebelumnya untuk *user* u_i. Oleh karena itu, hasil perhitungan S_i merupakan daftar preferensi yang diprediksi S_i untuk *user* u_i.

2.7. Degree Of Agreement (DOA)

Untuk mengukur akurasi dari sistem yang dibuat dalam memprediksi buku yang direkomendasikan, akan digunakan *degree of agreement* (DOA). Pengujian dilakukan dengan cara menghilangkan beberapa rating pada setiap user. Biarkan P menjadi himpunan semua buku, L_i adalah himpunan yang berisi buku yang telah diberi rating oleh user u_i dalam data uji, dan T_i adalah himpunan yang berisi buku yang telah user u_i berikan *rating* dalam data tes.

Lalu data tersebut akan dijadikan data uji untuk mengukur akurasi pada sistem. DOA didefinisikan dengan :

$$DOA = \frac{\sum_{u_i \in U} DOA_i}{|U|} \tag{24}$$

dimana U adalah himpunan semua user, $|U|$ adalah jumlah U, dan DOA_i adalah persentase pasangan produk dalam urutan yang benar. Untuk pengecekan pasangan produk dalam urutan yang benar yaitu dengan cara :

$$check\_order(p_j, p_k) = \begin{cases} 1, jika\ PP_i^j \ \geq \ PP_i^k \\ 0, lainnya \end{cases} \tag{25}$$

dimana PP_i^j dan PP_i^k menunjukkan preferensi yang diprediksi kepada user u_i pada buku p_j dan p_k. DOA_i didefinisikan oleh :

$$DOA = \frac{\sum_{P_J \in T_I \cap P_k \in NW_i} check\_order(p_j, p_k)}{|T_I| \ x \ |NW_i|} \tag{26}$$

dimana

$$NW_i = P - (L_i \cup T_I) \tag{27}$$

2.8. Mean Absolute Error (MAE)

MAE ($Mean\ Absolute\ Error$) dihitung untuk mengukur tingkat keakuratan pada data hasil prediksi dengan data aktual yang digunakan pada tugas akhir ini. Untuk menghitung MAE, harus mengubah preferensi yang diprediksi menjadi skor prediksi. Anggap $\hat{r}_{ij}$adalah skor prediksi buku p_j untuk *user* u_i [3]. Lalu $\hat{r}_{ij}$ dihitung dengan

$$\hat{r}_{ij} = r_{a,i} + \frac{\sum_{k=1 \wedge k \neq 1}^{N} [Sim(u_i, u_k) x (r_{kj} - r_{a,k})]}{\sum_{k=1 \wedge k \neq 1}^{N} Sim(u_i, u_k)} \tag{28}$$

dimana $r_{a,i}$ adalah rata – rata *rating* di L_i, $r_{a,k}$ adalah rata – rata *rating* di L_k, r_{kj} adalah *rating user* u_i terhadap buku p_j di L_k. L_i merupakan himpunan yang berisi produk yang *user* u_i berikan *rating* pada data latih dan L_k merupakan himpunan berisi buku yang *user* u_k berikan *rating* pada data latih. $Sim(u_i, u_k)$ adalah kemiripan antara *user* u_i dan *user* u_k, dan kemiripan tersebut dihitung menggunakan *cosine similarity* dari daftar preferensi yang diprediksi S_i dan S_k :

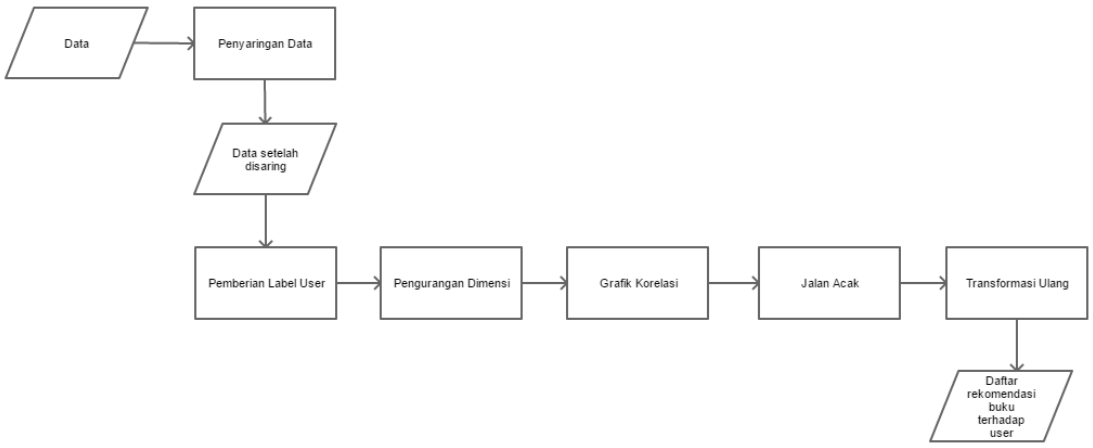
$$Sim(u_i, u_k) = \frac{S_i \cdot S_k}{\sqrt{S_i \cdot S_i} \sqrt{S_k \cdot S_k}} \tag{29}$$

lalu MAE dihitung dengan cara :

$$MAE = \frac{\sum_{i=1}^{N} \sum_{j=1 \wedge r_{ij} \in T_i}^{M} |r_{ij} - \hat{r}_{ij}|}{\sum_{i=1}^{N} |T_i|} \tag{30}$$

3. Sistem yang Dibangun

Gambar 2 menunjukkan gambar umum dari sistem, dimulai dari data awal sebelum disaring, hingga menghasilkan daftar rekomendasi buku kepada *user*.



Gambar 2. Sistem yang dibangun

3.1 Data

Ada 2 data yang digunakan pada tugas akhir ini. Pertama adalah data *Book Crossing* yang memiliki jumlah data sebanyak 1.048.576 rating. *Range rating* pada data tersebut adalah 1 – 10. Data kedua adalah data *Good Books*, data tersebut memiliki 5.976.479 data rating dan range ratingnya adalah 1 – 5. Kedua data tersebut memiliki bentuk data *user*, buku, dan *rating*. Data tersebut belum bisa digunakan untuk pengujian dengan metode yang digunakan pada tugas akhir ini. Data tersebut harus disaring terlebih dahulu, karena jika tidak disaring, akan membutuhkan waktu lama untuk melakukan rekomendasi karena data tersebut dalam jumlah besar. Format dari dataset adalah sebagai berikut :

Tabel 3. *Format Dataset*

Format Dataset (User;Buku;Rating)
276744;"038550120X";"7"

3.2 Penyaringan Data

Karena data yang akan digunakan adalah dalam jumlah besar, maka dilakukan penyaringan terhadap data tersebut. Untuk data *Book Crossing*, data disaring berdasarkan buku yang setidaknya diberikan *rating* oleh minimal 20 *user* dan *user* yang memberikan *rating* minimal terhadap 20 buku. Pada data *Good Books*, hanya digunakan 500.000 data rating awal, lalu di saring berdasarkan minimal *user* memberikan *rating* terhadap 80 buku dan buku yang diberikan *rating* minimal oleh 100 *user*.

Setelah data disaring, data yang awalnya sebesar 1.048.576 data menjadi data sebesar 25.767 data pada *Book Crossing*. Pada data *Good Books*, yang awalnya sebesar 500.000 data, menjadi sebesar 122.346 data. Format data setelah disaring tidak berubah, melainkan hanya ada penyusutan jumlah data, sehingga bentuk data yang digunakan sama seperti pada Tabel 3. Beberapa contoh data setelah dilakukan penyaringan adalah sebagai berikut :

Tabel 4. Contoh data setelah dilakukan penyaringan

User/Buku	002542730X	006016848X	006099486X	006101351X	⋯	1878424319
100004	0	0	0	0	⋯	0
100009	0	0	0	0	⋯	0
100067	0	0	0	0	⋯	0
100088	0	9	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋯	⋮
99996	0	0	0	0	⋯	0

Tabel 4 adalah contoh hasil dari data yang telah disaring, dengan syarat produk yang ada harus diberikan *rating* minimal oleh 20 *user*, dan *user* yang telah memberikan *rating* minimal 20 *rating* terhadap buku pada data *Book Crossing*. Data pada tabel diatas akan digunakan pada tahap selanjutnya, yaitu pelabelan *user*.

3.3 Pemberian Label User

Setelah data di saring, lalu data masuk pada tahapan pertama yaitu pemberian label berdasarkan kemiripan antar *user*. *User* dapat dikatakan mirip jika ada kesamaan dalam pemberian *rating* terhadap buku. Beberapa contoh hasil pada tahapan ini adalah sebagai berikut :

Tabel 5. Hasil data yang telah diberi label

User/Buku	002542730X	006016848X	006099486X	006101351X	⋯	1878424319	Label
100004	0	0	0	0	⋯	0	1
100009	0	0	0	0	⋯	0	2
100067	0	0	0	0	⋯	0	3
100088	0	9	0	0	⋯	0	4
⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮
99996	0	0	0	0	⋯	0	7

Tabel 5 adalah hasil data yang telah diberi label berdasarkan tingkat kemiripan antar *user* dan posisi clusternya. Tabel diatas nantinya akan digunakan untuk tahap selanjutnya, yaitu pengurangan dimensi. Tabel 5 didapatkan dengan menggunakan rumus (1) hingga (6).

3.4 Pengurangan Dimensi

Pada tahap kedua, setelah pemberian label pada data yang dilakukan pada tahap pertama. Akan dilakukan reduksi pada dimensi data NxM. Pertama akan dibangun *feature pattern* berdasarkan data yang ada dengan menggunakan rumus (7). Tabel 6 menunjukkan pola fitur yang didapatkan dari tahap pengurangan dimensi :

Tabel 6. Feature pattern yang didapatkan

x_i	$x_{i1}, x_{i2}, x_{i3}, x_{i4}, \dots, x_{iz}$
x_1	$0, 0, 0, 0, \dots, x_{1z}$
x_2	$0, 0, 0, 0.515152, \dots, x_{2z}$
x_3	$0, 0, 0, 0, \dots, x_{3z}$
⋮	⋮
x_{72}	$0, 0, 0, 0, \dots, x_{72z}$

Tabel 6 menunjukkan pola buku terhadap masing - masing kelompok *user*. *Feature pattern* yang didapatkan sebesar $x_i = x_{i1}, x_{i2}, \dots, x_{ij}$, $1 \leq i \leq N$ dan $1 \leq j \leq M$, dimana N = jumlah *user* dan M = jumlah buku. Lalu dengan menggunakan *feature pattern* yang didapatkan, akan memperkecil dimensi M. Anggap T sebagai matriks M yang telah diperkecil :

$$T = \begin{bmatrix} 1 & 0.059264 & \cdots & 0.030861 \\ 0.059264 & 1 & \cdots & 0.056379 \\ 0.042711 & 0.077836 & \cdots & 0.040479 \\ \vdots & \vdots & \vdots & \vdots \\ 0.030861 & 0.056379 & \cdots & 1 \end{bmatrix} \quad (31)$$

Setelah T didapatkan dengan menggunakan rumus (10) yang menunjukkan derajat keanggotaan *feature pattern* x_i terhadap kelompok G_j , barulah matriks $N \; x \; M$ dapat diperkecil. Anggap B adalah matriks yang telah diperkecil, maka :

Tabel 7. Hasil matriks N x M yang telah diperkecil

	G_1	G_2	⋯	G_{71}
100004	0.35379	0.646498	⋯	0.335403
100009	0.217467	0.395889	⋯	0.206018
⋮	⋮	⋮	⋯	⋮
99996	0.225428	0.410295	⋯	0.214

Tabel 7 adalah tabel hasil dari pengurangan dimensi buku yang menunjukkan *rating* yang diberikan *user* terhadap kelompok buku. Tabel diatas didapatkan dengan menggunakan rumus (11). Pada awalnya dimensi buku adalah sebesar M. Lalu setelah diperkecil, dimensi buku akan menjadi sebesar G_q yaitu sebesar banyak *cluster* yang diperoleh. Untuk tahapan selanjutnya, nilai – nilai dari tabel diatas digunakan karena tabel diatas mewakili *rating* yang diberikan terhadap buku oleh *user*.

3.5 Grafik Korelasi

Pada tahap ketiga, setelah mendapatkan B, akan dibuat korelasi grafik berdasarkan data tersebut. Anggap W adalah korelasi grafik, maka :

$$W = \begin{bmatrix} 0 & 0.011165 & \cdots & 0.030861 \\ 0.014949 & 0 & \cdots & 0.056379 \\ 0.016992 & 0.012357 & \cdots & 0.040479 \\ \vdots & \vdots & \vdots & \vdots \\ 0.030861 & 0.056379 & \cdots & 0 \end{bmatrix} \quad (32)$$

Grafik diatas didapatkan dengan menggunakan rumus (13) dan (14) yang menunjukkan hubungan antar kelompok buku.

3.6 Jalan Acak

Pada tahap keempat, akan diterapkan jalan acak pada data yang telah didapatkan pada tahapan – tahapan sebelumnya. Anggap V_i adalah vektor yang didapatkan setelah konvergen tercapai. Tabel 8 menunjukkan hasil dari V_i yang telah konvergen :

Tabel 8. Hasil vektor yang didapatkan setelah konvergen

V_{100004}	$[0.578963, 0.3766009, \dots, 0.575634]^T$
V_{100009}	$[0.376009, 0.401123, \dots, 0.374772]^T$
⋮	⋮
V_{99996}	$[0.396754, 0.422486, \dots, 0.395719]^T$

Hasil dari tabel 8 didapatkan dengan menggunakan rumus (20) yang menunjukkan daftar preferensi kelompok buku terhadap masing-masing *user*.

3.7 Transformasi Ulang

Pada tahap kelima, hasil preferensi kelompok buku V_i akan diubah menjadi preferensi buku individual dengan menggunakan rumus (23). Anggap S_i adalah hasil preferensi buku individual. Tabel 9 menunjukkan hasil preferensi buku individual :

Tabel 9. Hasil preferensi buku individual

	002542730X	006016848X	⋯	1878424319
S_{100004}	0.641503	0.603665	⋯	0.6463477
S_{100009}	0.414775	0.389517	⋯	0.41822046
⋮	⋮	⋮	⋯	⋮
S_{99996}	0.437643	0.410903	⋯	0.4413817

Lalu hasil dari transformasi ulang diurutkan berdasarkan nilai tersebesar hingga terkecil. Perlu diketahui, hanya 10 buku dengan nilai tertinggi yang direkomendasikan kepada user. Tabel 10 menunjukkan nilai rekomendasi tertinggi buku pada masing – masing user :

Tabel 10. Rekomendasi buku terhadap user berdasarkan nilai rekomendasi 10 tertinggi

User/Rekomendasi	1	2	3	4	⋯	10
100459	446601977	575400951	553560735	000649840X	⋯	8806142100
110973	552999954	446601977	575400951	000649840X	⋯	8408043641
149908	446601977	553560735	8807813025	8806142100	⋯	3492045170
177432	552999954	575400951	000649840X	048627263X	⋯	552996009
⋮	⋮	⋮	⋮	⋮	⋯	⋮
180651	552999954	446601977	553560735	000649840X	⋯	8408043641

4. Evaluasi

4.1. Proses Pengujian

Proses pengujian dilakukan dalam beberapa tahapan. Pertama data awal yang akan digunakan harus disaring terlebih dahulu dikarenakan data awal merupakan data yang besar. Setelah disaring, data tersebut akan dibagi menjadi data latih dan data uji. Untuk pembagian datanya adalah sebesar 80% data latih dan 20% data uji. Cara pembagian datanya adalah dengan menghilangkan *rating* sebanyak 20% dari total *rating* dengan rentang *rating* yang tertinggi dan terendah dari masing – masing *user*.

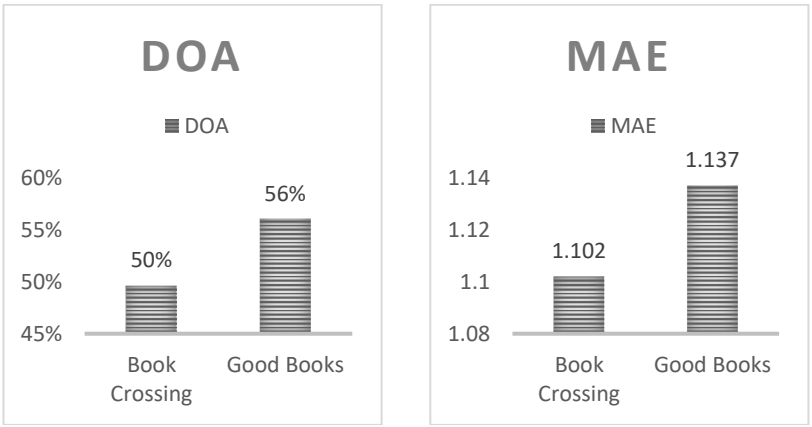
Setelah pemabagian data dan mendapatkan data uji, data tersebut akan melalui 5 tahapan metode yang diterapkan pada tugas akhir ini. 5 tahapan tersebut yaitu pemberian label *user*, pengurangan dimensi, grafik korelasi, jalan acak, dan transformasi ulang. Setelah itu, akan didapatkan daftar rekomendasi buku kepada masing – masing *user*. DOA dihitung untuk melihat persentase prediksi yang benar dan MAE dihitung untuk mengukur kesalahan yang ada pada data uji.

4.2. Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk menerapkan metode berbasis *clustering* dengan algoritma *self-constructing clustering* pada domain buku beserta mengukur performansinya dengan menghitung nilai DOA dan MAE. Akan dipaparkan hasil dari tahapan metode tersebut, mulai dari data awal yang belum di proses, hingga sistem menghasilkan rekomendasi. Setelah mendapatkan hasil rekomendasi, akan dihitung DOA dan MAE untuk melihat berapa nilai kebenaran prediksi dan kesalahan yang ada pada data yang digunakan.

4.3. Hasil Pengujian

Hasil pengujian dari tugas akhir ini adalah data yang didapatkan pada sub bab 3.7. Lalu DOA dan MAE akan dihitung untuk melihat besar persentase kesalahan pada data uji dan juga melihat persentase kebenaran prediksi data pada sistem. Berdasarkan penjelasan bagaimana cara menghitung DOA dan MAE pada sub bab 2.7 dan 2.8, hasil dari DOA dan MAE adalah sebagai berikut :



Gambar 3. DOA dan MAE yang didapatkan pada data uji

4.4. Analisis Hasil Pengujian

Pada tahapan pemberian label perlu diketahui bahwa setiap *user* akan diberikan label yang sama ketika dalam suatu cluster yang sama, dan label yang berbeda jika dalam cluster yang berbeda. Pemberian label *user* pada pengujian di tugas akhir ini, memiliki 110 label yang berbeda pada data *Book Crossing* dan pada data *Good Books* ada 170 label berbeda. Jumlah label tersebut bisa berbeda jika *threshold* dan deviasi untuk pelabelan *user* diperkecil ataupun diperbesar.

Pengurangan dimensi ini bertujuan untuk memperkecil dimensi atribut buku dengan menggunakan algoritma *self-constructing clustering*. Awalnya dimensi buku yang digunakan pada data *Book Crossing* adalah sebesar 2121 jenis buku. Setelah dilakukan pengurangan dimensi, didapatkan kelompok buku sebesar 5 kelompok buku. Untuk *Good Books* awalnya memiliki 1020 buku, menjadi 171 kelompok buku. Jumlah kelompok buku tersebut bisa berbeda jika *threshold* dan deviasinya diperkecil ataupun diperbesar. Semakin besar *threshold* maka semakin banyak *cluster* yang dihasilkan dan semakin kecil *threshold* maka semakin sedikit *cluster* yang dihasilkan. Hal tersebut juga berpengaruh terhadap tahapan pelabelan user. Jika kelompok buku yang dihasilkan dalam jumlah kecil, maka waktu untuk melakukan rekomendasi semakin efisien dan jika kelompok buku yang dihasilkan dalam jumlah besar, maka waktu yang dibutuhkan untuk melakukan rekomendasi akan tidak efisien.

Grafik korelasi yang didapatkan pada masing – masing dataset adalah sebesar $k \; x \; k$, dengan k = kelompok buku. Nilai – nilai dari grafik korelasi menunjukkan hubungan antar kelompok buku. Perlu diketahui bahwa kelompok buku yang sama tidak akan memiliki hubungan, sehingga menyebabkan nilai hubungan kelompoknya sebesar 0.

Hasil dari pengurangan dimensi dan grafik korelasi digunakan untuk menerapkan jalan acak. Hasil dari tahapan ini adalah masing – masing *user* mendapatkan preferensi daftar kelompok buku. Besar dari preferensi daftar kelompok buku yang diberikan kepada user adalah sebesar jumlah *cluster* yang ada.

Transformasi ulang dilakukan untuk mengubah preferensi daftar kelompok buku menjadi preferensi daftar buku. Hasilnya adalah setiap *user* akan memiliki nilai untuk setiap buku. Buku akan diurutkan sesuai dengan nilai terbesar hingga terkecil. Buku dengan nilai yang paling besar adalah buku pertama yang akan direkomendasikan kepada *user*, begitu juga seterusnya.

Berdasarkan nilai MAE dan DOA yang didapatkan, diketahui bahwa beberapa hal yang mempengaruhi nilai MAE dan DOA. Banyaknya jumlah rating pada masing-masing user mempengaruhi nilai MAE. Jika jumlah rating yang diberikan oleh user terhadap buku semakin besar, maka MAE yang didapatkan semakin besar, hal itu berlaku juga untuk sebaliknya. Untuk nilai DOA dipengaruhi oleh banyak jarak antar rating buku dengan buku lainnya yang diberikan oleh user. Prediksi DOA yang salah lebih dominan pada banyak jarak yang sedikit, begitu juga sebaliknya.

5. Kesimpulan

Metode berbasis *clustering* dengan menggunakan algoritma *self-constructing clustering* berhasil diterapkan pada sistem rekomendasi dengan domain buku. Dari kedua data yang digunakan, DOA dan MAE yang dihasilkan masing – masing adalah 50% dan 1.102 pada data uji *Book Crossing*, 56% dan 1.137 pada data uji *Good Books*. MAE yang dihasilkan pada kedua data tersebut cukup bagus karena nilai MAE mendekati nilai 0. Untuk DOA yang dihasilkan juga cukup bagus, karena kebenaran prediksi >= 50%.

Daftar Pustaka

[1] Ba, Q. L. (2013). Clustering collaborative filtering recommendation system based on SVD algorithm. *IEEE Int. Conf. Software Eng. Service Sci.*, 963–967.

[2] Cai, Y. f. (2014). Typicality-based collaborative filtering recommendation. *IEEE Trans. Knowl. Data Eng. 26 (3)*, 766–779.

[3] Chih-Lun Liao, S.-J. L. (2016). A clustering based approach to improving the efficiency of collaborative. *Electronic Commerce Research and Applications*, 1-9.

[4] Gyöngyi, Z. G.-M. (2004). Combating web spam with trustrank. *International Conference on Very Large Data Bases (VLDB) Morgan Kaufmann*, 576–587.

[5] Harel, D. K. (2001). On clustering using random walks. *Lecture Notes in Computer Science-Foundations of Software Technology and Theoretical Computer Science 2245*, 18–41.

[6] Huang, Z. C. (2004). A graph model for e-commerce recommender systems. *J. Am. Soc. Inf. Sci. Technol. 55 (3)*, 259–274.

[7] Jiang, J.-Y. L.-J.-J. (2011). A fuzzy self-constructing feature clustering algorithm for text classification. *IEEE Trans. Knowl. Data Eng. 23 (3)*, 335–349.

[8] Pucci, A. G. (2007). A random-walk based scoring algorithm applied to recommender engines. *Lecture Notes in Computer Science – Advances in Web Mining and Web Usage Analysis 4811*, 127-146.

[9] Raymond J. Mooney, L. R. (2000). Content-Based Book Recommending Using Learning for Text Categorization. *5th ACM conference on Digital libraries*, 195-204.

[10] Sarwar, B. K. (2002). Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering. *5th International Conference on Computer and Information Technology*.

[11] Sarwat, M. L. (2014). LARS: an efficient and scalable location-aware recommender system. *IEEE Trans. Knowl. Data Eng. 26 (6)*, 1384–1399.

[12] Yildirim, H. K. (2008). A random walk method for alleviating the sparsity problem in collaborative filtering. *ACM Conference on Recommender Systems*, 131–138.