

## Sistem Rekomendasi pada Buku dengan Menggunakan Metode *Trust-Aware Recommendation*

### Recommendation System for book by using Trust-Aware Recommendation Method

Mohammad Iqbal Fathurrahman<sup>#1</sup>, Dade Nurjanah<sup>#2</sup>, Rita Rismala<sup>#3</sup>

*#School of Computing, Telkom University*

*Jl. Telekomunikasi No.01, Terusan Buah Batu, Bandung, Jawa Barat, Indonesia*

<sup>1</sup>iqbalfathur2@gmail.com

<sup>2</sup>dadenurjanah@gmail.com

<sup>3</sup>rismala.rita@gmail.com

#### Abstrak

Sistem rekomendasi merupakan fitur yang banyak digunakan pada perangkat lunak zaman sekarang. Sistem rekomendasi sangat berguna untuk pengguna yang menggunakan sebuah perangkat lunak terutama sistem rekomendasi pada buku, karena fitur ini dapat memanjakan pengguna dengan memberikan rekomendasi buku yang mungkin sesuai dengan preferensi buku yang diinginkan. Sistem rekomendasi pada Tugas Akhir ini menggunakan metode Trust-Aware, dimana metode ini merupakan hasil penggabungan metode Collaborative Filtering dan PageRank. Dimana Collaborative Filtering menggunakan similarity metric untuk melakukan penghitungan rating, dan PageRank menggunakan trust metric untuk melakukan penghitungan terhadap setiap buku yang dikunjungi dengan melakukan show synopsis. Kemudian akan dilakukan pengukuran hasil pengujian terhadap sistem rekomendasi ini menggunakan MAE. Pengujian dilakukan dengan 3 skenario yang menggunakan 3 jenis jumlah data yang berbeda. Hasil pengujian memberikan angka 1,267 , 1,294 dan 1,181, yang artinya ketiga nilai tersebut tidak mempunyai selisih yang tidak terlalu jauh. Sehingga dapat ditarik kesimpulan bahwa metode Trust-Aware dapat digunakan pada sistem rekomendasi buku dan tidak terpengaruh oleh jumlah buku yang digunakan.

**Kata kunci:** sistem rekomendasi, *collaborative filtering*, *pagerank*, *trust-aware*

#### 1. Pendahuluan

Sistem rekomendasi adalah salah satu fitur pada sebuah perangkat lunak yang sangat bermanfaat untuk memudahkan pengguna. Sistem rekomendasi sendiri sangat diperlukan dikarenakan terlalu banyaknya jenis dan jumlah data yang ada. Dengan adanya sistem rekomendasi, pengguna akan dimanjakan dengan rekomendasi – rekomendasi buku yang sesuai dengan preferensi masing – masing pengguna, sehingga pengguna tidak perlu repot melakukan pencarian buku yang diinginkan. Sistem rekomendasi sendiri harus dapat menganalisis sekian banyak data tentang pengguna dan buku yang tersedia, dapat juga didukung dengan data *rating* agar hasil yang diberikan lebih akurat.

Teknik untuk sebuah sistem rekomendasi pada buku yang telah ada adalah *Collaborative Filtering* (CF). Secara khusus, sistem rekomendasi yang berbasis CF mengandalkan *rating* yang diberikan oleh pengguna lain. Pada dasarnya CF mencoba untuk secara otomatis menemukan pengguna yang mirip dengan yang aktif dan merekomendasikan kepadanya *item* yang disukai oleh pengguna serupa ini. Intuisi sederhana ini efektif dalam menghasilkan rekomendasi dan banyak digunakan. Namun sistem rekomendasi yang berbasis pada CF menderita beberapa kelemahan inheren yang bersifat intrinsik di proses menemukan pengguna serupa [1].

Untuk mengatasi kekurangan tersebut maka perlu dibuat metode hasil pengembangan metode CF sebelumnya, yaitu *Trust-Aware*. Metode ini dapat menganalisis pengguna baru tersebut dengan menggunakan data dari lingkaran pertemanannya sehingga sistem dapat memberikan rekomendasi yang lebih akurat dibanding metode pendahulunya. Metode ini juga dapat memilih diantara sekian banyak teman yang dimiliki, teman yang mana yang akan dijadikan acuan untuk memberikan rekomendasi, hal itu didapat dari data *trust* antar sesama pengguna. Teman yang mempunyai tingkat *trust* paling tinggi mempunyai kemungkinan lebih besar akan digunakan sebagai acuan daripada teman yang lainnya [1]. Metode ini pernah digunakan sebelumnya pada sosial media [2].

## 2. Tinjauan Pustaka

### 2.1 Sistem Rekomendasi

Sistem rekomendasi dapat didefinisikan sebagai program yang mencoba untuk merekomendasikan *item* yang paling cocok (produk atau jasa) untuk pengguna tertentu (individu atau bisnis) dengan memprediksi minat pengguna di *item* berdasarkan informasi terkait tentang *item*, pengguna dan interaksi antara *item* dan pengguna. Tujuan dari pengembangan sistem rekomendasi adalah untuk mengurangi informasi yang berlebihan dengan mengambil informasi dan layanan yang paling relevan dari sejumlah besar data, sehingga memberikan layanan pribadi. Fitur yang paling penting dari sebuah sistem rekomendasi adalah kemampuannya untuk “menebak” preferensi dan kepentingan pengguna dengan menganalisis perilaku pengguna dan / atau perilaku pengguna lain untuk menghasilkan rekomendasi pribadi [3]. Saran terhadap buku pada Amazon adalah contoh dunia nyata dari operasi sebuah industri yang mementingkan sebuah sistem rekomendasi [4].

Ekspektasi dari sebuah sistem rekomendasi ada memberikan rekomendasi dan estimasi yang bagus (*accuracy*), memprediksi *rating* buku (*coverage*), merekomendasikan buku baru (*novelty*), merekomendasikan buku yang berbeda (*diversity*), tidak terjadi perubahan rekomendasi yang banyak dalam waktu yang singkat (*stability*), dan tidak terpengaruh oleh serangan (*resistance to attacks*) [5].

Tantangan utama sebuah sistem rekomendasi adalah:

- *Data Sparsity*: Hal ini sangat jarang bahwa dua pengguna menilai *item* yang sama berkali-kali. Hal ini membuat lebih sulit untuk menghitung kesamaan.
- *Cold start for user*: Pengguna yang belum melakukan *rating* terhadap beberapa barang dapat menyebabkan ia mendapatkan hasil yang tidak akurat. Hal ini mungkin terjadi karena pengguna lain yang memiliki kemiripan dengan pengguna ini tidak dapat ditemukan
- *Cold start for item*: *Item* yang baru diperkenalkan memungkinkan tidak mendapatkan *rating* yang cukup, yang menyebabkan tidak direkomendasikan kepada pengguna.

- *Attacks*: Sistem rekomendasi memiliki ancaman terhadap serangan, seperti menyalin keseluruhan profil pengguna dan membuat sistem berpikir bahwa si penyerang dan pengguna tersebut sangat mirip. Hal ini membuat penyerang dapat membodohi sistem dan membuatnya menyarankan *item* apapun yang di *rating* oleh penyerang kepada pengguna [5].

### 2.1.1 Collaborative Filtering

*Collaborative Filtering* (CF) adalah salah satu teknik rekomendasi yang populer yang prediksi dan rekomendasinya berbasis pada nilai *rating* atau tingkah laku dari pengguna lain dalam sistem tersebut. Anggapan mendasar pada metode ini adalah opini pengguna lain dapat dipilih dan diagregasikan untuk memberikan prediksi dari preferensi pengguna aktif. Intinya, diasumsikan bahwa apabila beberapa pengguna mempunyai minat yang sama terhadap suatu buku, maka besar kemungkinan mereka mempunyai minat yang sama juga untuk buku yang lain [6].

Teknik ini telah banyak digunakan pada berbagai aplikasi, contohnya adalah MovieLens [7], Netnews [8], dan pada Amazon.com [9].

Kelebihan teknik CF memungkinkan pengguna aktif untuk mendapatkan rekomendasi berdasarkan produk yang pengguna dengan minat yang sama telah membeli dan diberi nilai positif, dan dengan menggunakan peringkat sebelumnya pengguna aktif dan riwayat transaksi untuk membangun model yang menyediakan satu set baru produk serupa [10]. Namun kekurangan pada teknik ini adalah penggunaannya tidak bisa memilih pengguna mana yang ingin dijadikan acuan agar mendapatkan rekomendasi produk yang diinginkan.

### 2.1.2 Content-based Recommendation

Teknik *Content-based Recommendation* (CB) merekomendasikan artikel atau komoditas yang mirip dengan *item* yang sebelumnya disukai oleh pengguna tertentu. Prinsip-prinsip dasar dari sistem rekomendasi *Content-based* adalah: 1) Untuk menganalisis deskripsi item disukai oleh pengguna tertentu untuk menentukan atribut umum pokok (preferensi) yang dapat digunakan untuk membedakan barang-barang tersebut. preferensi ini disimpan dalam profil pengguna. 2) Untuk membandingkan atribut masing-masing item dengan profil pengguna sehingga hanya item yang memiliki tingkat kesamaan yang tinggi dengan profil pengguna akan direkomendasikan [3].

Penerapan teknik *Content-based* telah banyak digunakan, beberapa contohnya adalah untuk melakukan *spam filtering* pada email [11], fitur *filtering* pada berbagai *On-Line Social Networks* [12], dan bahkan untuk *spam filtering* pada fitur SMS [13].

Penerapan teknik *Content-based* ini pernah digunakan pada buku sebelumnya [14]. Tahapan yang terjadi adalah: 1) Mengekstrak informasi dan membangun *database*. 2) Mempelajari profile. 3) Memproduksi, menjelaskan, dan me-revisi rekomendasi. Dari ke-tiga tahapan diatas, didapatkan hasil eksperimen berupa: 1) Metodologi yang berupa koleksi data, evaluasi performansi, dan diskusi metodologis. 2) Hasil dasar. 3) Hasil dari peran *Collaborative Content* [14].

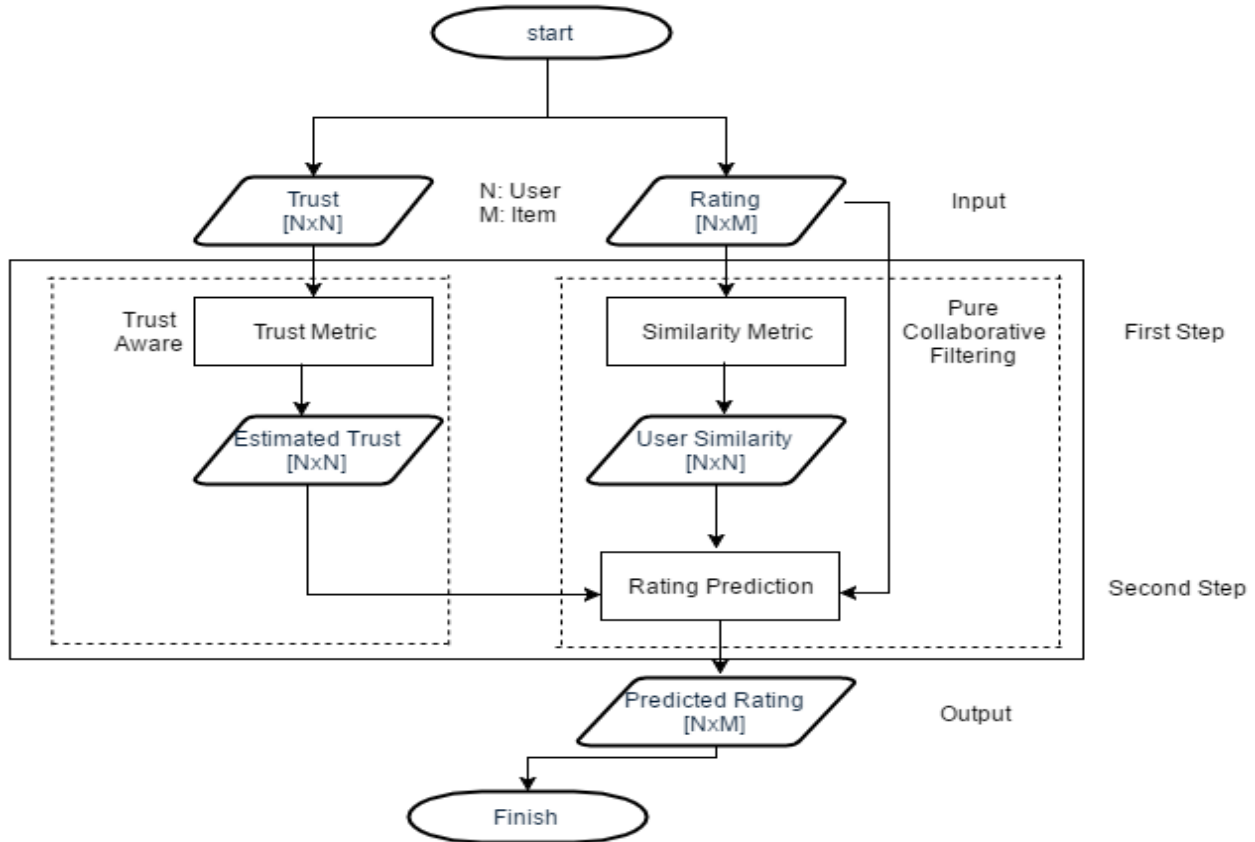
Teknik *Content-based* ini mempunyai kelebihan yaitu dapat menganalisis produk dan menemukan kemiripan dengan pengguna aktif untuk merekomendasikan produk. Tidak seperti CF, teknik ini tidak memerlukan *database* aktif dari riwayat pembelian [10]. Namun kekurangan pada teknik ini adalah ketidakmampuan menangani pengguna baru yang sistem tidak memiliki informasi apa – apa terhadap pengguna baru tersebut, atau yang biasa dikenal dengan *cold start*.

## 2.2 Trust-Aware Recommendation

Beberapa teknik diatas adalah pendekatan yang paling tua dan banyak digunakan pada sistem rekomendasi. Namun seiring berjalannya waktu, teknik – teknik tersebut mulai “usang” dan berbagai kekurangannya membuat kurang dapat diaplikasikan pada sistem rekomendasi yang ada pada hari ini. Sehingga perlu dibuat teknik – teknik baru yang berlandaskan teknik pendahulunya yang lebih efektif dan dapat mengatasi kekurangan – kekurangan yang ada. Salah satu teknik baru tersebut adalah *Trust-Aware Recommendation*.

Di dalam lingkungan desentralisasi dimana semua orang bebas untuk membuat konten dan tidak ada entitas *quality control* yang terpusat, mengevaluasi kualitas dari konten – konten ini menjadi hal yang penting. Situasi ini dapat dilihat pada komunitas *online* (contohnya, pada Slashdot.org dimana berjuta – juta pengguna dapat menerbitkan berita dan komen setiap hari), dalam jaringan *peer-to-peer* (dimana anggotanya dapat memasukkan *corrupted item*), atau pada website berbelanja *online* (seperti eBay.com, dimana para pengguna dapat membuat lelang palsu). Pada lingkungan – lingkungan ini, adalah sebuah strategis yang bagus untuk melimpahkan tugas penilaian kualitas kepada pengguna – pengguna nya sendiri. Sistem dapat menanyakan pengguna – penggunanya untuk *me-rating* pengguna lain: dengan cara ini, seorang penggua dapat mengekspresikan tingkat kepercayaannya kepada pengguna lain yang ia sedang berinteraksi. Contohnya pada sebuah kalimat kepercayaan adalah “Saya, Alice, mempercayai Bob dengan skala 0.8”. sistem kemudian dapat mengagregasikan semua kalimat kepercayaan ke sebuah jaringan kepercayaan tunggal yang merepresentasikan hubungan / relasi antara sesama pengguna [2].

Metode ini dapat mengatasi kelemahan pada metode - sebelumnya yaitu *cold start*, karena metode ini dapat melihat sebuah kesamaan preferensi dari *user* baru dengan *user* lama dengan melihat lingkaran pertemanan *user* baru tersebut, sehingga sistem dapat memberikan rekomendasi yang lebih akurat. Metode ini mempunyai arsitektur sebagai berikut [1]:



Gambar 1 Arsitektur Trust Aware

Berdasarkan gambar di atas, dapat dijelaskan tahapan proses pada metode ini:

- Trust Metric:
 

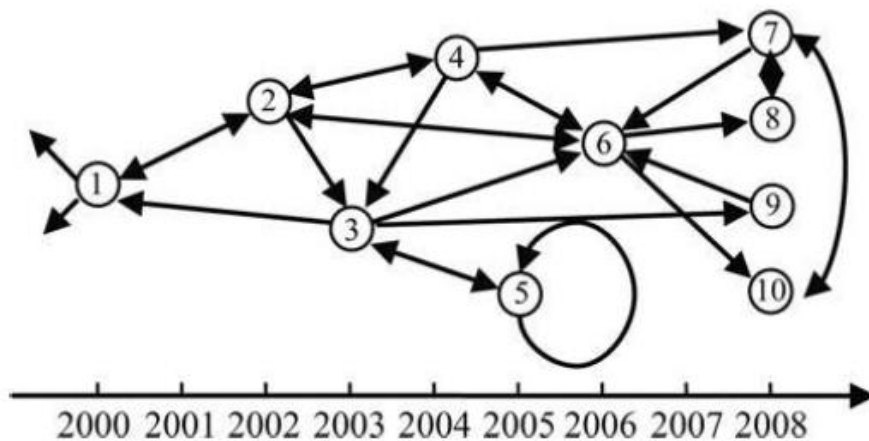
Modul *Trust Metric* berperan sebagai inputan dari jaringan *trust* dan mengeksploitasi propagasi untuk memprediksi, untuk setiap pengguna, seberapa banyak seorang pengguna bisa percaya dengan pengguna lain. Modul ini nantinya akan menghasilkan *Estimated Trust Matrix* [15].
- Similarity Metric:
 

Mengkomputasi tingkat kemiripan dari seorang pengguna dengan setiap pengguna lain adalah salah satu langkah dasar dalam teknik CF. Tugasnya adalah mengkomputasi korelasi antara 2 pengguna, dan menghasilkan outputan berupa  $n \times n$  matriks *User Similarity* yang pada kolom ke- $i$  mengandung nilai *similarity* dari pengguna ke- $i$  dengan setiap pengguna lain. Nilai korelasi digunakan pada tahap berikutnya sebagai pertimbangan untuk *rating* pengguna, berdasarkan hal itu, jika seorang pengguna A *me-rating* barang mirip dengan pengguna B, maka *rating* pengguna A berguna untuk memprediksi *rating* dari pengguna B [15].
- Rating Prediction:

Tahap ini adalah langkah terakhir yang klasik dari CF. Prediksi *rating* dari pengguna saat ini adalah jumlah dari *rating* yang diberikan ke *item* i oleh tetangga a. Tetangga dapat diambil dari matriks *User Similarity* atau dari matriks *Estimated Trust* dan *weights* adalah sel pada matriks yang dituju.

**2.3 PageRank**

*PageRank* adalah salah satu metode *global trust metric* yang paling banyak digunakan [1]. *PageRank* adalah metode yang dapat menghubungkan 1 *item* ke *item* lain. *PageRank* dapat digambarkan seperti sebuah *node* yang terhubung dengan *node* lain melalui *arrow*, dimana arah *arrow* dapat menunjukkan keterkaitan sebuah *node* terhadap *node* lain, dimana dalam contoh kasus buku bisa saja sebuah *node* tersebut mengutip dari *node* lain [16].



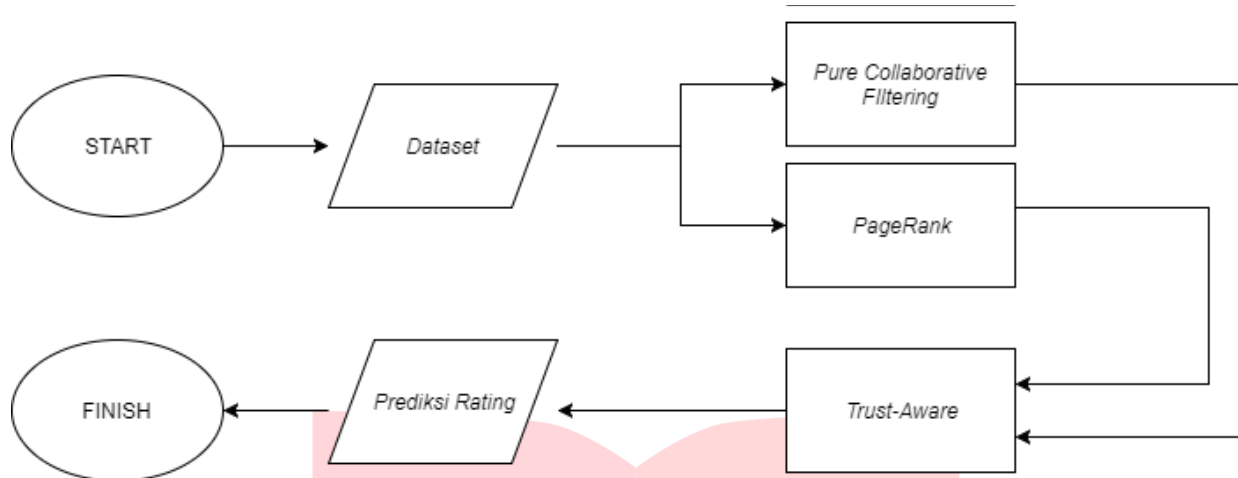
Gambar 2 Contoh PageRank dalam sebuah grafik [16]

Namun terdapat batasan dalam pengerjaan Tugas Akhir ini. Dikarenakan Tugas Akhir ini berbasis *desktop* jadi untuk penerapan *PageRank* menggunakan pendekatan yang sedikit berbeda. Ketika pengguna ditampilkan daftar buku terdapat *tombol* “*Show Synopsis*”, dimana nilai *trust* akan bertambah setiap pengguna melakukan klik pada *tombol Show Synopsis* tersebut.

**3. Desain Sistem**

**3.1 Gambaran Umum Sistem**

Sistem ini dibangun dengan mengikuti alur sebagai berikut:



Gambar 3 Flowchart Gambaran Umum Sistem

Berikut penjelasan alur sistem berdasarkan Gambar 3:

- *Dataset*: Sistem menerima inputan *dataset*.
- *Pure Collaborative Filtering*: sistem mengolah *dataset* menggunakan *collaborative filtering*
- *PageRank*: sistem mengolah data menggunakan data yang didapatkan dari tiap pengguna yang melakukan *show synopsis* pada setiap buku
- *Trust-Aware*: Sistem menjalankan algoritma *Rating Predictor*.
- *Prediksi rating*: Sistem menampilkan hasil prediksi *rating* pada setiap buku berdasarkan hasil perhitungan *Trust-Aware*

### 3.2 Dataset

*Dataset* yang digunakan pada Tugas Akhir ini adalah data buku, *rating*, pengguna yang telah terdaftar dan asal pengguna. *Dataset* buku didapat dari Book-Crossing (<http://www2.informatik.uni-freiburg.de/~ciegler/BX/>). Jumlah buku yang digunakan sebanyak 100 dari 271,379 buku yang disediakan. Hal ini dikarenakan waktu untuk memproses buku lebih dari 100 akan memakan waktu yang sangat lama, termasuk untuk membaca data maupun untuk memproses data buku beserta *rating* nya.

#### 3.2.1 Book-Crossing

*Dataset* Book-Crossing memberikan 3 jenis data, yaitu *dataset* buku, pengguna dan *rating*, namun yang digunakan hanya *dataset* buku saja karena *dataset user* yang diberikan berisi *user* dengan alamat yang berada di luar negeri. *Dataset* buku tersebut mempunyai format *csv (comma-separated values)*. Data buku tersebut mempunyai 8 atribut, yaitu ISBN, Book-Title, Book-Author, Year-of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L.

Table 1 contoh format dan isi data buku

ISBN;"Book-Title";"Book-Author";"Year-Of-Publication";"Publisher";"Image-URL-S";"Image-URL-M";"Image-URL-L"
0399135782;"The Kitchen God's Wife";"Amy Tan";"1991";"Putnam Pub Group";"http://images.amazon.com/images/P/0399135782.01.THUMBZZZ.jpg";"http://imag

es.amazon.com/images/P/0399135782.01.MZZZZZZZ.jpg";"http://images.amazon.com/image s/P/0399135782.01.LZZZZZZZ.jpg"
<b>0609804618;"Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source";"The Onion";"1999";"Three Rivers Press";"http://images.amazon.com/images/P/0609804618.01.THUMBZZZ.jpg";"http://image s.amazon.com/images/P/0609804618.01.MZZZZZZZ.jpg";"http://images.amazon.com/images /P/0609804618.01.LZZZZZZZ.jpg"</b>

Untuk melengkapi data yang tidak disediakan oleh Book-Crossing maka perlu dibuat *dataset* lain seperti *dataset user*, *region*, *survey page* dan *survey rating*.

Table 2 contoh format dan isi data user

<b>id_user;"nama_user";id_region</b>
<b>1;"ADMIN";1107</b>
<b>11;"IQBAL";1813</b>

Table 3 contoh format dan isi data region

<b>id_region;province_id;"nama_region"</b>
<b>1107;11;"KABUPATEN ACEH BARAT"</b>
<b>1871;18;"KOTA BANDAR LAMPUNG"</b>

Table 4 contoh format dan isi data survey page

<b>id;id_book;id_user</b>
<b>16;83;7</b>
<b>36;51;12</b>

Table 5 contoh format dan isi data survey rating

<b>id;id_book;stars;id_user</b>
<b>21;3;5;10</b>
<b>72;64;4;13</b>

### 3.3.1 Pure Collaborative Filtering

Untuk penerapan CF menggunakan teknik yang banyak digunakan yaitu Pearson Correlation Coefficient [15].

$$W_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2 \sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2}} \tag{3-1}$$

Keterangan:

$W_{a,u}$  : Nilai *user similarity*.

$\bar{r}_{a,i}$  : *Rating user a terhadap item i*.

$\bar{r}_a$  : Rata – rata rating oleh *user a*.

$\bar{r}_{u,i}$  : *Rating tetangga user a terhadap item i*.

$\bar{r}_u$  : Rata – rata rating tetangga *user a terhadap item i*.



$m$  : Jumlah *user* yang memberi *rating* terhadap *item i*.

### 3.3.2 PageRank

Algoritma *PageRank* oleh Google didesain untuk memproses jaringan link seperti yang ditunjukkan pada Gambar 2. Kunci utama algoritma nya ditunjukkan pada rumus berikut [16]:

$$PageRank(A) = 1 - d + d \times \sum_{i=1}^n \frac{PageRank(Pi)}{O(Pi)} \quad (3-2)$$

Keterangan:

PageRank(A) : menunjukkann nilai dari Page(A)

$d$  : *Damping Factor* yang bernilai antara 0 sampai 1, biasanya 0.85

$Pi$  : 1 dari  $n$  *page* yang memberikan *link* menuju Page(A)

$O(Pi)$  : *outdegree* dari Page(Pi) yang berada dalam jaringan *link* tersebut

### 3.4 Trust-Aware

Tahap berikutnya adalah melakukan perhitungan menggunakan *Rating Predictor*. Tahap ini adalah langkah klasik terakhir dari CF [1].

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^k w_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^k w_{a,u}} \quad (3-3)$$

Keterangan:

$p_{a,i}$  : Prediksi *rating* terhadap *item i*.

$\bar{r}_a$  : Rata – rata *rating* oleh *user a*

$\bar{r}_u$  : Rata – rata dari *rating* yang diberikan oleh *user u*.

$r_{u,i}$  : *Rating* oleh *user u* terhadap *item i*.

$w_{a,u}$  : Bobot dari *user similarity* dari  $a$  dan  $u$  yang telah dihitung pada tahap sebelumnya.

$k$  : Jumlah *user* yang melakukan *rating* terhadap *item i*.

## 4. Pengujian dan Analisis

### 4.1 Strategi Pengujian

Tujuan dari sistem yang dibangun adalah memberikan daftar buku dari yang paling direkomendasikan kepada *user* sampai yang paling tidak direkomendasikan. Sebuah buku dianggap menarik apabila hasil prediksi *rating* bernilai tinggi berdasarkan penggabungan *PageRank* dan CF. Pengujian sistem dilakukan dengan menggunakan *evaluation metrics* yaitu Mean Absolute Error (MAE). Pengujian ini dipengaruhi oleh jumlah buku, *rating* dan *user*, dengan variabel jumlah buku yang bisa diubah.

### 4.2 Skenario Pengujian

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \tag{4-1}$$

Berdasarkan rumus MAE, skenario pengujian dilakukan sebagai berikut:

Table 6 skenario pengujian

No	Pengujian	Keterangan
1	MAE dengan data rating TA dan data CF	Data rating TA: $f_i$ Data rating CF: $y_i$ Jumlah buku yg digunakan: 101
2	MAE dengan data rating TA dan data CF	Data rating TA: $f_i$ Data rating CF: $y_i$ Jumlah buku yg digunakan: 51
3	MAE dengan data rating TA dan data CF	Data rating TA: $f_i$ Data rating CF: $y_i$ Jumlah buku yg digunakan: 11

### 4.3 Analisis Hasil Pengujian

Berikut ditampilkan statistik dari data *rating* dan klik pada *show synopsis* dan yang didapatkan:

Table 7 statistik rating

Rating	Jumlah rating
1	62
2	109
3	143
4	185
5	92

Table 8 statistik klik show synopsis

ID Buku	Jumlah Klik Show Synopsis
1 – 20	228
21 – 40	250
41 – 60	261
61 – 80	224
81 – 101	205

Hasil dari setiap pengujian ditampilkan sebagai berikut:

Table 9 tabel pengujian

Nomor Pengujian	Total Rating (TA)	Total Rating (CF)	Jumlah buku	Jumlah User	MAE (skala 5)
1	180	308	101	20	1.267
2	92	158	51	20	1.294
3	19	32	11	20	1.181

Pada Table 9, terlihat bahwa pada semua hasil pengujian mendapatkan nilai dengan perbedaan yang tidak terlalu jauh. Hal ini menunjukkan bahwa jumlah buku tidak berpengaruh terlalu besar pada hasil akhir rekomendasi. Maka dapat disimpulkan metode *Trust-Aware* ini mempunyai nilai akurasi yang cukup stabil terhadap berapapun jumlah buku yang digunakan.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan, yaitu Sistem rekomendasi yang menggunakan metode *Trust-Aware* dapat memberikan rekomendasi yang cukup akurat dengan memberikan nilai MAE 1,267 pada 101 buku, 1,294 pada 51 buku dan 1,181 pada 11 buku dan tidak terpengaruh terhadap berapapun jumlah buku yang digunakan.

### 5.2 Saran

Terdapat beberapa saran yang dapat dijadikan pertimbangan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian pada Tugas Akhir ini hanya dilakukan secara *offline* sehingga belum bisa sepenuhnya merepresentasikan kemampuan maksimal dari setiap unsur pada sistem rekomendasi yang dibangun, maka untuk penelitian selanjutnya dapat dilakukan secara *online* untuk dapat memberikan hasil evaluasi yang lebih optimal.
2. Penerapan metode ini sebaiknya dilakukan ketika riset terhadap metode ini sendiri telah sempurna sehingga dapat memberikan hasil rekomendasi yang jauh lebih akurat.

## Daftar Pustaka

- [1] P. A. Paolo Massa, "Trust-aware Recommender Systems," *Trust-aware Recommender Systems*, pp. 17-24, 2007.
- [2] M. N. R. D. G. Na Li, "Trust-aware Privacy Control for Social Media," *Human Factors in Computing System*, pp. 1597-1602, 2011.
- [3] D. W. M. M. W. W. G. Z. Jie Lu, "Recommender System Application Developments: A Survey," pp. 1-30, 2015.
- [4] V. S. Prem Melville, "Recommender System," *Encyclopedia of Machine Learning*, pp. 829-837, 2010.
- [5] F. P. Makbule Gulcin Ozsoy, "Trust Based Recommendation Systems," *Advances in Sosial Networks Analysis and Mining*, pp. 1-8, 2013.
- [6] J. T. R. J. A. K. Michael D. Ekstrand, "Collaborative Filtering Recommender Systems," *Foundations and Trends® in Human-Computer Interaction*, pp. 175-243, 2011.

- [7] J. A. K. J. R. Jonathan L. Herlocker, "Explaining Collaborative Filtering Recommendations," *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 241-250, 2000.
- [8] N. I. M. S. P. B. J. R. Paul Resnick, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175-186, 1994.
- [9] B. S. J. Y. Greg Linden, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, pp. 76-80, 2003.
- [10] A. S. H. R. A. M. M. Sanjeevan Sivalapan, "Recommender Systems in E-Commerce," pp. 1-6, 2014.
- [11] A. Y. Tiago A. Almeida, "Content-Based Spam Filtering," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1-7, 2010.
- [12] E. B. B. C. M. C. E. F. Marco Vanetti, "Content-Based Filtering in On-Line Social Networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 127-140, 2011.
- [13] G. C. B. E. P. S. Jose Maria Gomez Hidalgo, "Content Based SMS Spam Filtering," *Proc. of the 2006 DOCENG*, pp. 107-114, 2016.
- [14] L. R. Raymond J. Mooney, "Content-Based Book Recommending Using Learning for Text Categorization," *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 195-204, 2000.
- [15] P. A. Paolo Massa, "Trust-aware Collaborative Filtering for Recommender Systems," *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pp. 492-508, 2004.
- [16] P. W. Jiang Li, "ArticleRank: a PageRank-based alternative to numbers of citations for analysing citation networks," *Emerald Insight*, 2009.