



## Penerapan Metode *Item-Based Collaborative Filtering* Untuk Sistem Rekomendasi Data MovieLens

Yohanes Visser Laja Jaja<sup>1</sup>, Bambang Susanto<sup>1\*</sup>, Leopoldus Ricky Sasongko<sup>1</sup>

<sup>1</sup>Program Studi Matematika–Fakultas Sains dan Matematika–UKSW, Jln. Diponegoro No. 52-60, Salatiga 50711, Indonesia

\*Corresponding author: [bambang.susanto@uksw.edu](mailto:bambang.susanto@uksw.edu)

### ABSTRAK

Pada masa sekarang ini film telah menjadi salah satu hiburan favorit utama masyarakat. Jumlah film pertahun terhitung mencapai ribuan. Hal ini membuat penggemar film kesulitan dalam memilih film mana yang tepat untuk ditonton sesuai keinginan. Sehingga dibutuhkan sistem rekomendasi yang bertujuan untuk memberikan saran film mana yang akan dipilih. Sistem rekomendasi adalah sistem yang membantu pengguna dalam mengatasi informasi yang meluap dengan memberikan rekomendasi spesifik bagi pengguna dan diharapkan rekomendasi tersebut bisa memenuhi keinginan dan kebutuhan pengguna. Terdapat tiga jenis sistem rekomendasi berdasarkan metode yang digunakannya yakni, collaborative filtering, content-based filtering, dan hybrid. Metode yang digunakan adalah collaborative filtering merupakan salah satu yang sering digunakan dalam sistem rekomendasi. Collaborative filtering dibagi menjadi dua bagian yaitu item-based collaborative filtering dan user-based collaborative filtering. Dalam tugas akhir ini penulis menggunakan metode item-based collaborative filtering. Data set yang digunakan adalah data set dari MovieLens.org berupa 100.000 rating yang diberikan oleh pengguna terhadap film. Data MovieLens akan diproses menggunakan program R dan memakai paket R yaitu recommenderlab.

### ABSTRACT

Nowadays movies have become one of the people's favorite pastimes. The number of films per year is counted in the thousands. This makes it difficult for movie fans to choose the right movie to watch as desired. So we need a recommendation system that aims to provide advice on which films to choose. Recommendation system is a system that helps users in overcoming information overflow by providing specific recommendations for users and it is expected that these recommendations can meet the wants and needs of users. There are three types of recommendation systems based on the method used, namely, collaborative filtering, content-based filtering, and hybrid. The method used is collaborative filtering which is often used in the recommendation system. Collaborative filtering is divided into two parts, namely item-based collaborative filtering and user-based collaborative filtering. In this thesis the author uses item-based collaborative filtering methods. The data set used is a data set from MovieLens.org in the form of 100,000 ratings given by users of the film. MovieLens data will be processed using the R program and using the R package that is recommenderlab.

### INFO ARTIKEL

Diterima : 10 April 2020

Diterima setelah revisi : 28 Agustus 2020

Tersedia online : 5 September 2020

### Kata Kunci:

*Sistem Rekomendasi,  
Collaborative Filtering,  
Item-Based Collaborative Filtering,  
MovieLens,  
Recommenderlab.*

### ARTICLE INFO

Received : 10 April 2020

Received in revised form : 28 August 2020

Available online : 5 September 2020

### Keywords:

*Recommendation System,  
Collaborative Filtering,  
Item-Based Collaborative Filtering,  
MovieLens,  
Recommenderlab.*

### 1. PENDAHULUAN dan DASAR TEORI

Pada masa sekarang ini film telah menjadi salah satu hiburan favorit utama masyarakat. Jumlah film pertahun terhitung mencapai ribuan. Hal ini membuat suatu keadaan dimana kita kesulitan dalam mencari informasi yang sesuai dengan kriteria penggemar film. Kemungkinan film yang samasekali tidak terpikirkan olehnya namun ternyata menarik untuk dilihat dan sesuai dengan selera. Salah satu solusi dari permasalahan ini adalah sistem rekomendasi yang memanfaatkan opini atau rating orang lain terhadap suatu film.

Sistem rekomendasi adalah sistem yang membantu pengguna dalam mengatasi informasi yang meluap dengan memberikan rekomendasi spesifik bagi pengguna dan diharapkan rekomendasi tersebut bisa

memenuhi keinginan dan kebutuhan pengguna. Namun pada perkembangannya, diperlukan suatu model yang dapat memberikan nilai lebih kepada pelanggan yaitu berupa rekomendasi yang dapat memberikan informasi mengenai produk yang dianggap sesuai dengan keinginan pelanggan. Karena itu diperlukan model rekomendasi yang tepat agar rekomendasi yang diberikan sistem sesuai dengan keinginan pelanggan, serta mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan dibelinya [6].

*Recommendation system* atau disebut pula *recommender system* mulai diperhatikan sejak kemunculan penelitian tentang *collaborative filtering* pada pertengahan 90'an. Perbandingan terhadap kedua pendekatan dalam *collaborative filtering*. Pada penelian

tersebut, disimpulkan bahwa algoritma *item-based* memberikan kinerja yang lebih baik daripada algoritma *user-based*, sementara pada saat yang sama algoritma *item-based* juga memberikan kualitas prediksi yang lebih baik dari pada algoritma *user-based* [5].

Maka dari itu, penelitian ini mencoba untuk menerapkan metode *item-based collaborative filtering* untuk menghasilkan rekomendasi yang tepat pada film. Data film diambil dari *MovieLens* yaitu situs rekomendasi film personal yang disusun berdasarkan rating yang diberikan pengguna terhadap sebuah film [1]. *MovieLens* dijalankan oleh *GroupLens*, sebuah laboratorium penelitian di University of Minnesota. Data *MovieLens* diproses menggunakan program R dengan paket *Recommenderlab*. *Recommenderlab* ini diciptakan untuk membuat dan melakukan testing pada algoritma *item-based collaborative filtering*.

### 1.1. Data Penelitian

Data yang dianalisis adalah data *MovieLens*. Data tersebut dikumpulkan melalui situs web *MovieLens* [movielens.umn.edu](http://movielens.umn.edu) selama tujuh bulan dari 19 September 1997 hingga 22 April 1998. Kumpulan data berisi sekitar 100.000 peringkat (1-5) dari 943 pengguna pada 1664 film.

### 1.2. Sistem Rekomendasi

Sistem rekomendasi adalah sistem yang membantu pengguna dalam mengatasi informasi yang meluap dengan memberikan rekomendasi spesifik bagi pengguna dan diharapkan rekomendasi tersebut bisa memenuhi keinginan dan kebutuhan pengguna [4]. Sistem rekomendasi banyak digunakan untuk membuat prediksi seperti buku, musik, film, tempat wisata dan lain-lain. Sebagai contoh sistem rekomendasi digunakan untuk pemilihan film yang nantinya dapat membantu pengguna memilih film yang akan ditonton. Diperlukan model rekomendasi yang tepat agar rekomendasi yang diberikan sesuai dengan keinginan pengguna, serta mempermudah pengguna mengambil keputusan yang tepat dalam menentukan film yang akan dipilih.

Terdapat tiga jenis sistem rekomendasi berdasarkan metode yang digunakannya yakni, *collaborative filtering*, *content-based filtering*, dan *hybrid*. *Collaborative Filtering* dibagi menjadi dua bagian yaitu *item-based* dan *user-based*[6].

### 1.3. Collaborative Filtering

*Collaborative filtering* merupakan proses penyaringan atau pengevaluasian item menggunakan opini orang lain [5]. Metode ini menghasilkan rekomendasi bagi pengguna yang dituju terhadap satu item bahkan lebih. Konsepnya adalah berupa asumsi bahwa seseorang yang menyukai suatu barang tertentu, maka barang tersebut tersebut juga akan disukai oleh orang lainnya Item dapat terdiri atas apa saja yang dapat disediakan seperti misalnya buku, film, musik, atau

tujuan wisata. Rating dalam filtering dapat berbentuk: (a) Model rating dengan menilai suatu item seperti rating 1 sampai 5, (b) Model rating dengan memilih antara setuju atau tidak setuju, atau dapat juga baik atau buruk, (c) Rating *unary* dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli item atau merating item dengan baik.

### 1.4. Item-Based Collaborative Filtering

Metode *Item-based collaborative filtering* menggunakan item-item yang telah dirating oleh pengguna menjadi dasar perhitungan. Kata "Collaborative" mengacu pada fakta bahwa pengguna berkolaborasi satu sama lain untuk merekomendasikan item. Metode ini melakukan perhitungan bagaimana kemiripan item-item yang telah dirating dengan item-item lain dan selanjutnya dipilih sekelompok item yang mempunyai nilai kemiripan dengan item yang telah dirating. Nilai kemiripan tersebut dicatat selanjutnya dijadikan nilai bobot untuk memprediksi nilai rating pada item yang ditunjukkan.

Adapun langkah-langkah di dalam metode *item-based collaborative filtering* antara lain:

#### a) Menentukan Matrik Ratin

Data pengguna dan item yang diratingnya dideskripsikan sebagai matriks rating ( $R_{ui}$ ), dimana baris merepresentasikan  $u$  pengguna dan kolom merepresentasikan  $i$  item. Dengan demikian, element matrik  $r_{ui}$  berarti nilai rating yang diberikan oleh pengguna ke  $u$  terhadap item ke  $i$ .

$$(R_{ui}) = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1i} \\ r_{21} & r_{22} & \dots & r_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ r_{u1} & r_{u2} & \dots & r_{ui} \end{bmatrix} \quad (1)$$

#### b) Menghitung Matrik Kemiripan

Tahap penting di dalam algoritma *item-based collaborative filtering* adalah menghitung kemiripan (*similarity*) antara item-item dan memilih item mana yang paling mirip [5]. Berikut ini adalah persamaan yang digunakan untuk menghitung nilai kemiripan item  $i$  dan pengguna  $u$ .

*Consine similarity* digunakan untuk melakukan perhitungan tingkat kemiripan dari data. Rumus yang digunakan oleh *consine similarity* [2].

$$sim(i, u) = \cos(\vec{i}, \vec{u}) = \frac{\vec{i} \cdot \vec{u}}{\|\vec{i}\|_2 \|\vec{u}\|_2} \quad (2)$$

Keterangan:

- $\vec{i}$  = Vektor  $i$ , yang akan dibandingkan kemiripannya
- $\vec{u}$  = Vektor  $u$ , yang akan dibandingkan kemiripannya
- $\|\vec{i}\|$  = panjang vektor item  $i$
- $\|\vec{u}\|$  = panjang vector pengguna  $u$

**c) Menghitung Matrik Kemiripan**

Setelah mendapatkan sekumpulan item yang sangat mirip dengan menggunakan rumus *cosine similarity*, dilakukan proses menghitung prediksi yang memperkirakan nilai rating dari pengguna. Prediksi yang diperkirakan adalah item yang belum pernah diberi rating. Jika nilai prediksinya sangat tinggi, berarti target pengguna mungkin tertarik dengan item tersebut. Teknik yang digunakan untuk mendapatkan nilai prediksi adalah dengan persamaan *weighted sum*.

$$P_{u,i} = \frac{\sum_{i \in I}(S_{i,u} * R_{u,i})}{\sum_{i \in I}(|S_{i,u}|)} \quad (3)$$

Keterangan:

- $P_{u,i}$  = Nilai prediksi rating untuk pengguna  $u$  pada item  $i$
- $\sum_{i \in I}$  = Semua himpunan item yang mirip dengan item  $I$
- $R_{u,i}$  = Rating pengguna  $u$  pada item  $i$
- $S_{i,u}$  = Nilai kemiripan antara item  $i$  dan pengguna  $u$  yang mirip

**1.5. MovieLens**

MovieLens adalah situs rekomendasi film personal yang disusun berdasarkan rating yang diberikan pengguna terhadap sebuah film [1]. Dataset MovieLense berisi susunan rating/peringkat yang diberikan pengguna terhadap film yang diminati. MovieLens berisi dari 943 pengguna dan 1664 film yang telah dirating sebanyak 99392 kali yang dikumpulkan selama tujuh bulan dari 19 September 1997 hingga 22 April 1998. MovieLens dijalankan oleh GroupLens, yaitu sebuah laboratorium penelitian di University of Minnesota (<https://movielens.org/>).

**1.6. Recommenderlab**

Recommenderlab menyediakan tiga metode untuk menghitung kemiripan adalah *cosinus*, *pearson*, dan *jaccard*. Tujuan aplikasi ini adalah sebagai penelitian umum infrastruktur di dalam colaboratif filtering. Algoritma yang telah tersedia diantaranya, Item-Based Collaborative Filtering (IBCF), User-Based Collaborative Filtering (UBCF), lengkap dengan mekanisme evaluasinya [3].

Recommenderlab dibentuk diatas platform R, sehingga dapat menggunakan fasilitas-fasilitas dalam platform tersebut. Paket ini masih dimungkinkan untuk dikembangkan menggunakan algoritma baru, sesuai dengan tujuannya [3].

**2. METODE PENELITIAN**

Disini dilakukan pengolahan data yang dimiliki dengan menerapkan metode *item-based collaborative filtering* dengan bantuan program R melalui langkah-langkah sebagai berikut

1. Menginstal paket recommenderlab diatas platform R, jika menggunakan Rstudio dapat langsung menginstal paket tersebut melalui *Repository*(CRAN) lewat menu *Packages ->*

*install*. Kemudian, untuk melihat dokumentasinya gunakan perintah library (“recommenderlab”),

2. Data yang dianalisis adalah data MovieLens. Data film tersebut termasuk klas *realRatingMatrix*. Klas dari matriks peringkat ini didefinisikan khusus dalam paket recommenderlab untuk mengantisipasi data *sparse*, karena matriks rating biasanya berbentuk matriks *sparse*,
3. Data film dianalisis lalu ditentukan jumlah pengguna per film dan sebaliknya dengan film yang telah diberi rating. Selanjutnya data tersebut diberi nama *ratings\_movies*,
4. Model rekomendasi film dibangun menggunakan sebagian dari data *MovieLens* (data *training*) dan mengaplikasikannya ke sisa data lainnya (data uji).
5. Data *training* ditetapkan dengan probabilitas suatu data terpilih sebagai data *training* sebesar 80 persen
6. Menghitung matrik kemiripan antar film dengan cara mengitung kemiripan antara empat item film pertama. Dengan perintah R *similarity\_items*,
7. Membangun model rekomendasi film dengan fungsi rekomen yang inputnya adalah data, metode dan parameter,
8. Langkah selanjutnya adalah menerapkan model rekomendasi film pada data uji untuk menunjukan banyaknya item yang direkomendasikan kepada setiap pengguna. Kemudian mengidentifikasi film yang paling direkomendasikan.

**3. HASIL DAN PEMBAHASAN**

**3. 1. Penyiapan Data MovieLens**

Didalam program R, recommenderlab berisi beberapa data set dapat dilihat melalui perintah berikut:

**Tabel 1.** Penyiapan Data film

[1] “Jester5k”
[2] “JesterJokes (Jester5k)”
[3] “MSWeb”
[4] “MovieLense”
[5] “MovieLenseMeta (MovieLense)”

```
data_package<-data(package="recommenderlab")
data_package$results[, "Item"]
data(MovieLense)
MovieLense
## 943 x 1664 rating matrix of class
'realRatingMatrix' with 99392 ratings.
```

Tabel 1. merupakan macam-macam data set dalam recommenderlab. Data set yang dipakai yaitu MovieLens/MovieLense yang berupa kumpulan data berisi sekitar 100.000 rating dengan skala 1-5 dari 943 pengguna pada 1664 film

### 3. 2. Klas realRatingMatrix

Bagian ini data film pada *MovieLens* akan dipilih kriterianya dengan membuat Klas dari matriks peringkat dan didefinisikan khusus dalam paket recommenderlab.

```
class(MovieLense)
## [1] "realRatingMatrix"
## attr("package")
## [1] "recommenderlab"
Methods (class = class(MovieLense))
```

**Tabel 2.** Metode yang tersedia untuk klas realRatingMatrix

[1] [	<-	binarize
[4] calcPredictionAccuracy	coerce	colCounts
[7] colMeans	colSds	colSums
[10] denormalize	dim	dimnames
[13] dimnames<- evaluationScheme	dissimilarit	
[16] getData.frame	getList	getNormalize
[19] getRatingMatrix getTopNLists	getRati	
[22] image	normalize	nratings
[25] Recommende rowCounts	removeKnownRatings	
[28] rowMeans	rowSds	rowSums
[31] sample	show	similarity

### 3. 3. Memilih Data Rating Fim yang Relevan

Menghindari terjadinya bias dalam analisis data perlu ditentukan jumlah minimum pengguna per film dan sebaliknya. Dalam penelitian ini data yang akan olah dipilih dengan kriteria:

- Pengguna yang telah memberi peringkat setidaknya 50 film.
- Film yang telah ditonton setidaknya 100 kali.

Untuk selanjutnya data tersebut diberi nama ratings\_movies sebagaimana terlihat dalam perintah berikut:

```
ratings_movies <-  
MovieLense[rowCounts(MovieLense) >  
50,colCounts(MovieLense) > 100]  
ratings_movies  
## 560 x 332 rating matrix of class 'realRatingMatrix'  
with 55298 ratings
```

Jadi data yang akan dianalisis adalah data ratings\_movies berisi sekitar setengah dari pengguna dan seperlima film dari data MovieLens.

### 3. 4. Model Rekomendasi Film

Sebagaimana telah dipaparkan sebelumnya bahwa data rating\_movies berisi pengguna MovieLens yang telah memberi peringkat setidaknya 50 film dan film yang telah dinilai setidaknya 100 kali. Data tranning ditetapkan probabilitasnya sebesar 80 pesen.

```
which_train <- sample(x = c(TRUE, FALSE), size =  
nrow(ratings_movies), replace = TRUE, prob =  
c(0.8,0.2))  
recc_data_train <- ratings_movies [which_train, ]  
recc_data_test <- ratings_movies[!which_train, ]
```

### 3. 5. Menghitung Matrik Kemiripan

Recommenderlab menyediakan tiga metode untuk menghitung kemiripan adalah *cosinus*, *pearson*, dan *jaccard*. Dipenelitian ini penulis hanya menggunakan metode *cosinus*.

Berikut disajikan perintah R untuk menghitung kemiripan antara empat item pertama

```
similarity_items <- similarity(MovieLense[, 1:4],  
method = "cosine", which = "items")  
as.matrix(similarity_items)
```

**Tabel 3.** Nilai kemiripan antara item

	Toy Story (1995)	GoldenEye (1995)	Four Rooms (1995)	Get Shorty (1995)
Toy Story (1995)	0.0000	0.9487374	0.9132997	0.9429069
GoldenEye (1995)	0.9487374	0.0000	0.9088797	0.9394926
Four Rooms (1995)	0.9132997	0.9088797	0.0000	0.8991940
Get Shorty (1995)	0.9429069	0.9394926	0.8991940	0.0000

### 3. 6. Membangun Model Rekomendasi

**Tabel 4.** Parameter

Parameters	Default
K	30
Method	Cosine
Normalize	Center
normalize_sim_matrix	FALSE
Alpha	0.5
Na_as_zero	FALSE
minRating	NA

Berikut menjamin model rekomendasi telah terbangun dengan parameter pada tabel 4:

```
recc_model <- Recommender (data = recc_data_train,  
method = "IBCF", parameter = list (k = 30))  
recc_model  
## Recommender of type 'IBCF' for 'realRatingMatrix'  
## learned using 437 users.  
class(recc_model)  
## [1] "Recommender"  
## attr("package")
```

```
## [1] "recommenderlab"
```

### 3.7. Menerapkan Model Rekomendasi pada Data Uji

```
n_recommended <- 10
recc_predicted <- predict (object = recc_model,
newdata = recc_data_test, n = n_recommended)
recc_predicted
## rekomendasi 10 film untuk 123 pengguna
```

Perintah `n_recommended = n` menunjukkan banyaknya item yang direkomendasikan kepada setiap pengguna. Perintah berikut memberikan rekomendasi `n` film dengan rating teratas. Perhatikan struktur dari hasil prediksi diatas:

```
class(recc_predicted)
## [1] "topNList"
## attr(,"package")
## [1] "recommenderlab"
slotNames(recc_predicted)
## [1] "items" "itemLabels" "n"
```

Keterangan:

- i. item berisi daftar dengan indeks item yang direkomendasikan untuk setiap pengguna.
- ii. itemLabels berisi nama item
- iii. n berisi jumlah rekomendasi

Jika diinginkan rekomendasi untuk pengguna pertama:

```
recc_predicted@items[[1]]
## 64 79 85 121 154 166 181 14 88 16
```

Selanjutnya dapat mengekstrak film yang direkomendasikan dari:

```
recc_user_1 <- recc_predicted@items[[1]]
movies_user_1 <-
recc_predicted@itemLabels[recc_user_1]
movies_user_1
```

**Tabel 5.** Rekomendasi film untk pengguna pertama

[1] "Independence Day (ID4) (1996)"
[2] "Long Kiss Goodnight, The (1996)"
[3] "Reservoir Dogs (1992)"
[4] "Shining, The (1980)"
[5] "Last of the Mohicans, The (1992)"
[6] "George of the Jungle (1997)"
[7] "Tin Cup (1996)"
[8] "Rumble in the Bronx (1995)"
[9] "Top Gun (1986)"
[10] "Apollo 13 (1995)"

**Tabel 6.** Jumlah film yang telah dirating paling banyak

Indeks	Film
64	"Independence Day (ID4) (1996)"
79	"Long Kiss Goodnight, The (1996)"
85	"Reservoir Dogs (1992)"
121	"Shining, The (1980)"
154	"Last of the Mohicans, The (1992)"
166	"George of the Jungle (1997)"
181	"Tin Cup (1996)"
14	"Rumble in the Bronx (1995)"
88	"Top Gun (1986)"
16	"Apollo 13 (1995)"

Berikut perintah untuk dapat mendefinisikan matriks dengan rekomendasi untuk setiap pengguna:

```
recc_matrix <- sapply(recc_predicted@items,
function(x)
{colnames(ratings_movies)[x]
})
dim(recc_matrix)
## [1] 10 123
```

Selanjutnya rekomendasi untuk empat pengguna pertama:

```
recc_matrix[, 1:4]
```

Untuk dapat mengidentifikasi film yang paling direkomendasikan, didefinisikan suatu vektor yang berisi semua rekomendasi dan kemudian membuat distribusi frekuensi banyaknya item tersebut:

```
number_of_items <- factor(table(recc_matrix))
chart_title <- "Distribution of the number of items for IBCF"
```

Sebagian besar film yang direkomendasikan hanya beberapa kali dilihat dan beberapa film direkomendasikan berkali-kali. Berikut daftar rekomendasi untuk film yang terpopuler:

```
number_of_items_sorted <- sort(number_of_items,
decreasing = TRUE)
number_of_items_top <- head
(number_of_items_sorted, n = 8)
table_top <-
data.frame(names(number_of_items_top),
number_of_items_top)
table_top
```

names	number_of_items_top
Crash (1996)	30
Lost Highway (1997)	30
Ace Ventura: Pet Detective (1994)	29
G.I. Jane (1997)	28
Cable Guy, The (1996)	27
GoldenEye (1995)	26
Benny & Joon (1993)	23
Firm, The (1993)	23

**Gambar 1.** Film populer dengan rekomendasi rating yang tinggi

#### 4. KESIMPULAN

Melalui penelitian ini dapat ditarik kesimpulan bahwa paket recommenderlab telah digunakan untuk menunjukkan bagaimana menerapkan metode *item-based collaborative filtering* pada data *MovieLens*. Sistem rekomendasi *item-based collaborative filtering* merekomendasikan item berdasarkan matriks kemiripan.

#### REFERENSI

- [1] Dwicahya, I. (2018). Perbandingan Sistem Rekomendasi Film Metode User-Based dan Item-Based Collaborative Filtering.
- [2] Fathoni, Putra, P., & Sucipta, R. A. (2016). Penerapan Metode Item-Based Collaborative Filtering Pada Sistem Electronic Commerce Berbasis Website. *Prosiding Annual Reserach Seminar*, 2(1), 227–230. <https://www.neliti.com/id/publications/169651/penerapan-metode-item-based-collaborative-filtering-pada-sistem-electronic-comm>
- [3] Hahsler, M. (2009). Recommenderlab : A Framework for Developing and Testing Recommendation Algorithms.
- [4] Jepriana, I. W. (2018). Algoritme Genetika untuk Mengurangi Galat Prediksi Metode Item-Based Collaborative Filtering. 2001, 1–7
- [5] Sarwar, B. (2001). Item-Based Collaborative Filtering Recommendation Algorithms.
- [6] Uyun, S., Fahrurrozi, I., & Mulyanto, A. (2011). Item Collaborative Filtering untuk Rekomendasi Pembelian Buku secara Online. 1(1), 63–70.



**Yohanes Visher Laja Jaja** ([jajavijav27@gmail.com](mailto:jajavijav27@gmail.com)) lahir Baucau, Timor-timor, pada tanggal 27 Mei 1998. Dia masih menempuh pendidikan tinggi di Program Studi Matematika, Fakultas Sains dan Matematika, Universitas Kristen Satya Wacana (UKSW) Salatiga. Tahun 2020 adalah tahun terakhir ia menempuh studi. Makalah ini merupakan hasil penelitian skripsinya yang dipublikasikan.



**BambangSusanto** ([bambang.susanto@staff.uksw.edu](mailto:bambang.susanto@staff.uksw.edu)) lahir di Ambarawa, 12 Juli 1963. Lulus sarjana pada tahun 1988 dari Universitas Diponegoro Semarang. Gelar Magister Sain diperolehnya pada tahun 1992 dari Institut Tekonologi Bandung dan Doktor Matematika pada tahun 2005 dari Institut Teknologi Bandung. Sejak tahun 1988, ia bekerja di Universitas Kristen Satya Wacana, Salatiga. Beberapa matakuliah yang diampunya sampai saat ini adalah : Matematika Diskret, Aljabar Linear, Fungsi Kompleks, Geometri Euclid, Statistika Matematika, Komputasi Finansial, Teknik Peramalan dan Analisis Data Multivariat. Bidang penelitian yang diminati adalah time series modeling dan cryptography serta landasan teoritis yang melatarbelakanginya seperti aljabar linear dan matematika diskrit. Berikut dua makalah hasil penelitiannya bersama rekan dosen dan mahasiswa bimbingan: Desain S-Box Fleksibel : Regenerasi Konstanta dan Koefisien Fungsi Linier Berbasis CSPRING Chaos yang dipublikasikan di *Jurnal Nasional Teknik Elektro dan Teknik Informatika* Vol. 8 No. 1 (2019) dan *Modeling of Return Volatility using GARCH(1,1) Model under Tuckey Transformations* *Jurnal Akutansi dan Keuangan* 21(1), Mei 2019, Universitas Petra.



**Leopoldus Ricky Sasongko** ([leopoldus.sasongko@staff.uksw.edu](mailto:leopoldus.sasongko@staff.uksw.edu)) lahir di Ketapang, Kalimantan Barat, pada tanggal 14 November 1989. Pada tahun 2011, gelar Sarjana Sains (S.Si) diperoleh dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Gelar Magister Sains (M.Si) didapat dari Program Pascasarjana Magister Aktuaria, Institut Teknologi Bandung (ITB), pada tahun 2014. Ia bekerja di UKSW sejak tahun 2011 sebagai Calon Pengajar Akademik (Dosen) di Program Studi Matematika, Fakultas Sains dan Matematika, UKSW. Saat ini, ia menjadi Pengajar Akademik Tetap di UKSW. Sasongko, M.Si, merupakan salah satu anggota Asosiasi Matematikawan Indonesia, IndoMS. Bidang penelitian yang digeluti adalah Matematika Aktuaria dan Garansi (*Warranty*). Salah satu makalah hasil penelitian adalah *The Estimation of Renewal Functions Using the Mean Value Theorem for Integrals (MeVTI) Method* yang terpublikasi di *Jurnal Matematika dan Aplikasi deCartesiaN*, Universitas Sam Ratulangi (UNSRAT).