**LAPORAN AKHIR PROYEK PENAMBANGAN DATA**

**Sistem Rekomendasi Buku Menggunakan Algoritma Apriori**



**Disusun oleh:**

|  |  |
| --- | --- |
| **12S17005** | **Kiky Purnamasari Napitupulu** |
| **12S17006** | **Tripheni Simanjuntak** |
| **12S17023** | **Jessycha Royanti Tampubolon** |

**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**

**INSTITUT TEKNOLOGI DEL**

**JANUARI 2020**

# Daftar Isi

[Daftar Isi 1](#_Toc60504223)

[Daftar Gambar 3](#_Toc60504224)

[Daftar Tabel 4](#_Toc60504225)

[1. Business Understanding 5](#_Toc60504226)

[1.1 Determine Business Objectives 5](#_Toc60504227)

[1.2 Situation Assessment 6](#_Toc60504228)

[1.3 Determine Data Mining Goal 6](#_Toc60504229)

[1.4 Produce Project Plan 6](#_Toc60504230)

[2. Data Understanding 9](#_Toc60504231)

[2.1 Collect Initial Data 9](#_Toc60504232)

[2.2 Describe Data 9](#_Toc60504233)

[2.3 Explore Data 9](#_Toc60504234)

[2.4 Verify Data Quality 10](#_Toc60504235)

[3. Data Preparation 11](#_Toc60504236)

[3.1 Data Set Description 11](#_Toc60504237)

[3.2 Select Data 13](#_Toc60504238)

[3.3 Clean Data 13](#_Toc60504239)

[3.4 Construct Data 14](#_Toc60504240)

[3.5 Investigate Data 15](#_Toc60504241)

[4. Modeling 17](#_Toc60504242)

[4.1 Select Modeling Technique 17](#_Toc60504243)

[4.2 Build Model 19](#_Toc60504244)

[5. Evaluation 27](#_Toc60504245)

[6. Deployment 30](#_Toc60504246)

[Referensi 32](#_Toc60504247)

# Daftar Gambar

[Gambar 1 Data pada Books 12](#_Toc60504017)

[Gambar 2 Data pada Book-Ratings 12](#_Toc60504018)

[Gambar 3 Data pada Users 12](#_Toc60504019)

[Gambar 4 Data gabungan 15](#_Toc60504020)

[Gambar 5 Rating Buku 15](#_Toc60504021)

[Gambar 6 *User* 16](#_Toc60504022)

[Gambar 7 Pseudocode Algoritma Apriori 18](#_Toc60504023)

[Gambar 8 Metode *Waterfall* 30](#_Toc60504024)

[Gambar 9 *Rule* pada model 31](#_Toc60504025)

# Daftar Tabel

[Tabel 1 *Produce Project Plan* 7](#_Toc60504031)

[Tabel 2 Deskripsi Dataset 11](#_Toc60504032)

# Business Understanding

Pada tahap *Business Understanding*, dilakukan proses menentukan tujuan pengerjaan proyek*,* menilai situasi sistem yang ada saat ini, menentukan tujuan penerapan *data mining* dan merencanakan tahapan pengerjaan proyek.

## 1.1 Determine Business Objectives

Informasi merupakan sekumpulan data atau fakta yang telah diproses dan dikelola sedemikian rupa sehingga menjadi sesuatu yang mudah dimengerti dan bermanfaat kepada pembaca. Seiring berjalannya waktu, informasi akan semakin banyak. Informasi yang dibutuhkan dapat berupa jurnal, artikel, buku dan saling berkaitan satu dengan yang lainnya. Untuk membantu mencari informasi mengenai buku yang akan dicari oleh pembaca, dibutuhkan sebuah sistem rekomendasi buku. Sistem rekomendasi buku ini akan sangat membantu dan dapat dijadikan solusi bagi pengguna dalam merekomendasikan sebuah informasi yang relevan mengenai buku yang akan dicari oleh pembaca.

Banyak nya jumlah buku yang dapat diperoleh, terkadang membuat pembaca kesulitan dalam menentukan buku yang hendak mereka cari. Dalam sebuah kasus, terkadang dijumpai pembaca hanya ingin membaca buku-buku yang dengan memiliki nilai penjualan yang terbaik. Namun, ada pula pembaca yang hanya ingin membaca buku yang mirip dengan buku-buku yang sudah pernah dibaca sebelumnya. Bahkan tidak jarang ditemui pembaca yang hanya akan membaca yang menentukan buku-buku yang akan dibaca selanjutnya berdasarkan *rating* dari buku-buku yang telah dilihat oleh pembaca. Semakin tinggi *rating* dari buku yang akan dicari, maka semakin tertarik pula pembaca untuk membacanya. Untuk menentukan tinggi rendahnya *rating* terhadap buku yang akan dicari oleh pembaca, dapat mempengaruhi buku-buku yang akan direkomendasikan oleh sistem yang akan dibangun. Nilai kemiripan antara buku dan *rating* buku dapat dijadikan sebagai landasan untuk memberikan rekomendasi buku kepada pembaca.

Sistem rekomendasi buku yang akan dibangun, akan memberikan solusi terhadap permasalahan dalam menentukan buku yang belum pernah dibaca oleh pembaca. Sistem rekomendasi buku menggunakan algoritma *apriori* diharapkan dapat membantu pembaca buku untuk menentukan buku yang layak dibaca dan buku yang tidak layak untuk dibaca berdasarkan rating dari buku tersebut. Penentuan rekomendasi buku dengan metode ini ditentukan berdasarkan fitur yang dimiliki oleh sistem yang memungkinkan pembaca nya untuk memberikan *rating* atau nilai terhadap buku-buku yang telah dibaca sebelumnya.

## 1.2 Situation Assessment

Orang - orang dengan minat membaca yang tinggi, jumlahnya cukup banyak, mulai dari kalangan pelajar maupun umum, sehingga mereka membutuhkan referensi atau rekomendasi buku untuk dibaca. Dengan adanya banyak sekali jumlah buku, sistem rekomendasi buku dapat menjadi alternatif untuk orang - orang yang bingung dalam menentukan buku yang ingin dibaca. Dengan memanfaatkan algoritma apriori dalam menentukan rekomendasi buku, maka akan memudahkan dalam menentukan buku yang ingin direkomendasikan pada pembaca. Batasan dari proyek ini adalah dataset ini dikumpulkan dari komunitas BookCrossing pada Agustus-September 2004 [1]. Penentuan keputusan dalam pembuatan strategi sistem rekomendasi buku adalah berfokus pada data rating buku.

## 1.3 Determine Data Mining Goal

Tujuan penerapan Data Mining ini adalah sebagai berikut :

1. Menghasilkan model sistem rekomendasi buku berdasarkan data yang sudah tersedia.
2. Menerapkan algoritma Apriori dalam pemodelan sistem rekomendasi buku yang akan dibangun.

## 1.4 Produce Project Plan

Rencana pengerjaan proyek yang akan dilakukan untuk mencapai tujuan *data mining* dan pengerjaan sistem rekomendasi buku menggunakan algoritma apriori dapat dilihat pada Tabel 1.

Tabel 1 *Produce Project Plan*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tahapan | Kegiatan | Keterangan |
| 1 | Business Understanding | 1. Determine Business Objectives 2. Situation Assessment 3. Determine Data Mining Goal 4. Produce Project Plan | Dilakukan proses menentukan tujuan pengerjaan proyek*,* menilai situasi sistem yang ada saat ini, menentukan tujuan penerapan *data mining* dan merencanakan tahapan pengerjaan proyek. |
| 2 | Data Understanding | 1. Collect Initial Data 2. Describe Data 3. Explore Data 4. Verify Data Quality | Dilakukan proses mengumpulkan data awal, mendeskripsikan data, *explore* data, dan memverifikasi kualitas data yang akan digunakan. |
| 3 | Data Preparation | 1. Data Set Description 2. Select Data 3. Clean Data 4. Construct Data 5. Investigate Data | Proses ini menggunakan jupyter notebook untuk menyiapkan dan meninjau data awal, memilih data yang relevan (menghilangkan data duplikat, mengurangi data dan, memisahkan data) dan melakukan perubahan perubahan data |
| 4 | Modeling | 1. Select Modeling Technique 2. Build Model | Proses ini menggunakan jupyter notebook untuk memodelkan sistem dan membangun mode menggunakan algoritma yang ditentukan |
| 5 | Evaluation | Evaluasi Model | Proses ini menggunakan jupyter notebook untuk mengevaluasi performa model sistem rekomendasi. |
| 6 | Deployment | 1. Deployment Plan 2. Monitoring and maintenance | Proses ini dilakukan untuk merencanakan kegiatan deploy terhadap performa dari sistem rekomendasi yang sudah dibangun serta melakukan monitoring dan melakukan pemeliharaan terhadap sistem rekomendasi yang sudah dibangun, agar apabila terdapat kelemahan pada sistem yang sudah dibangun, dapat segera ditangani oleh *developer*. |

# Data Understanding

Pada tahap pemahaman data *(data understanding)*, dilakukan proses mengumpulkan data awal, mendeskripsikan data, *explore* data, dan memverifikasi kualitas data yang akan digunakan.

## 2.1 Collect Initial Data

Data tersedia dalam format Csv dan dapat diunduh untuk penggunaan non komersial karena disediakan secara *open source*. Dataset dapat di download di link berikut: <http://www2.informatik.uni-freiburg.de/~cziegler/BX/>.

## 2.2 Describe Data

Data merupakan sebuah bahan mentah yang dapat digunakan sebagai bahan utama yang mendorong pertumbuhan bisnis dan penambangan data yang disimpan secara elektronik serta pencarian secara otomatis dengan komputer [2]. Saat ini, kebutuhan akan data semakin tinggi baik dalam bidang sains maupun teknologi, hal ini dikarenakan informasi yang diperoleh setiap waktu adalah hasil pengolahan dari kumpulan data yang telah diolah. Data diciri-cirikan sebagai fakta yang kemudian akan diolah menjadi informasi yang berisi kumpulan pola dari data atau ekspektasi yang mendasari data. Himpunan dari data disebut dengan *dataset*.

Dalam proyek ini, *dataset* yang digunakan dalam membangun sistem rekomendasi buku menggunakan algoritma Apriori adalah Book-Crossing Dataset. Book-Crossing Dataset merupakan kumpulan data buku yang dikumpulkan oleh Cai-Nicolas Ziegler dari komunitas Book Crossing dan terdiri dari 278.858 ribu data *user* yang disamarkan serta 1.149.780 data *rating* baik eksplisit maupun implisit dari 271.379 data buku [3].

## 2.3 Explore Data

*Dataset* memiliki nilai fitur atau atribut yang juga terdiri kumpulan atau koleksi dari semua bahan mentah data yang dikumpulkan dengan melalui metode penelitian data. Kumpulan data ini akan didistribusikan kepada pihak lain yang ingin menggunakannya sebagai masukan dalam penelitian dan hampir keseluruhan dapat diakses secara publik. Pada *dataset* BookCrossing terdapat tiga tabel yaitu tabel BX-Users, BX-Books, dan BX-Book-Ratings. Tabel BX-Users merupakan tabel yang terdiri dari data pengguna seperti *user ID* yang dipetakan ke dalam tipe data *integer* dan data demografi pengguna seperti *location* dan *age* yang berisi nilai Null-Values. Pada tabeL BX-Books terdapat data buku yang diidentifikasikan dengan kode ISBN yang valid, *Book-Title, Book-Author, Year-Of-Publication, Publisher,* dan beberapa alamat URL Amazon Website yang merujuk pada gambar halaman buku dalam tiga jenis ukuran (*small* (S)*, medium* (M) *,*dan *large* (L)). Sedangkan pada tabel BX-Book-Ratings terdiri dari informasi *rating* buku (Book-Rating) yang ditampilkan dalam skala 1-10 secara eksplisit dan 0 secara implisit.

## 2.4 Verify Data Quality

Pada dasarnya *dataset* Book-Crossing merupakan *dataset* yang memiliki tingkat *sparsity* yang tinggi sehingga ketika ingin mengolah *dataset* Book-Crossing dilakukan proses *condensation* [3]*.* Proses ini akan lebih memadatkan data dengan menghapus semua data buku yang memiliki taksonomi yang buruk sehingga mendapatkan hasil penambangan data yang lebih baik. Berdasarkan hasil proses *condensation* yang telah dilakukan pada penelitian sebelumnya maka dihasilkan dimensi *dataset* yang lebih layak dimana data yang ditampilkan adalah 10.339 data pengguna, 6.708 buku, 361.349 *rating*.

# 3. Data Preparation

Pada tahap pengolahan data *(data preparation)*, dilakukan proses menyiapkan data awal, memilih variabel yang akan dianalisis, melakukan perubahan pada variabel jika diperlukan sehingga data siap digunakan untuk pemodelan.

## 3.1 Data Set Description

Pada fase ini, *dataset* akan dideskripsikan dalam bahasa pemrograman python menggunakan tool jupyter *notebook.* Untuk mendeskripsikan dataset, ada beberapa fungsi yang dijalankan pada python, yaitu :

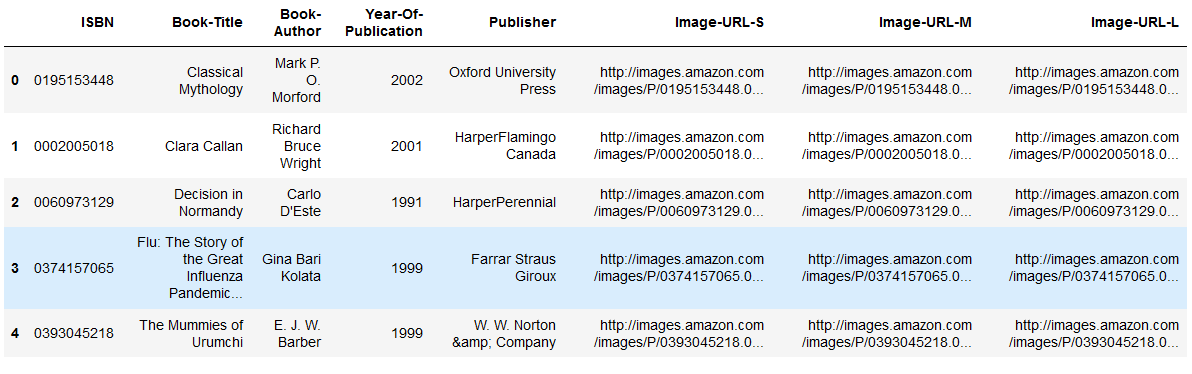
1. **.shape,** fungsi untuk memperoleh ukuran data (jumlah baris dan kolom) pada *dataset.*
2. **.columns,** fungsi untuk menampilkan nama - nama kolom yang ada pada *dataset.*

Deskripsi data yang diperoleh dari menjalankan kedua fungsi tersebut dapat dilihat pada Tabel 2.

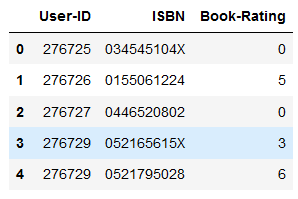
Tabel 2 Deskripsi Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **File** | **Baris** | **Jumlah Kolom** | **Nama Kolom** |
| Books | 271360 | 8 | 'ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher', 'Image-URL-S', 'Image-URL-M', 'Image-URL-L' |
| Book-Ratings | 1149780 | 3 | 'User-ID', 'ISBN', 'Book-Rating' |
| Users | 278858 | 3 | 'User-ID', 'Location', 'Age' |

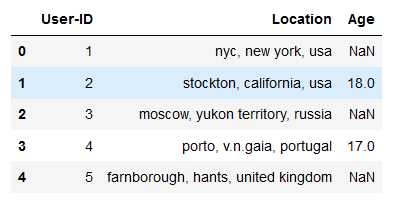
1. **head()**, fungsi untuk menampilkan n data pertama pada setiap file dataset. Berikut adalah tampilan masing - masing data dalam dataset.



Gambar 1 Data pada Books



Gambar 2 Data pada Book-Ratings

****

Gambar 3 Data pada Users

## 3.2 Select Data

Fase ini diperlukan untuk menghasilkan dataset yang lebih sedikit dari data asli untuk meningkatkan efisiensi *data mining.* Pada fase ini, akan dilakukan pemilihan data yang relevan seperti menghilangkan data duplikat dan mengurangi data, lalu memisahkan data yang akan dieksekusi dan data yang tidak dibutuhkan untuk proses pemodelan, yaitu :

1. Sebagian besar buku memiliki rating nol (0). Skala rating yang sebenarnya dimulai dari 1 hingga 10 yang disebut sebagai rating 'eksplisit'), sementara rating 0 menunjukkan rating 'implisit'. Untuk langkah pemodelan, yang diperlukan adalah rating eksplisit, sehingga rating dengan nilai 0 akan dihapus.



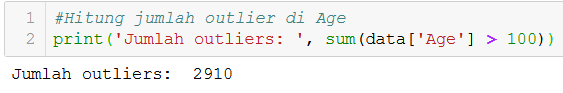
1. Pada kolom ‘Age’ di data Users, rentang usia dimulai dari 0 hingga 244 tahun, nilai ini tidak mungkin benar. Maka untuk langkah pemodelan, ditetapkan usia 5 sampai dengan 100 untuk mencoba membuat data *user* tampak realistis.



## 3.3 Clean Data

Data asli bisa mengandung data yang tidak lengkap, *noise* (mengandung *outliers*), atau tidak konsisten. Pada fase ini, dilakukan proses menghapus atau memodifikasi data salah, tidak relevan, duplikat, dan tidak terformat, yaitu :

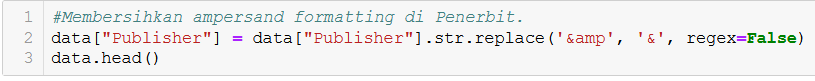
1. Dilakukan penanganan terhadap outliers, dimana jumlah *outliers* pada ‘Age’ dimana ['Age'] > 100 adalah 2910.



1. Terdapat 4619 yang di entri sebagai '0', dan 3 yang di entri sebagai NaN di kolom ‘Year-Of-Publication’ sehingga semua ‘Year-Of-Publication’ yang bernilai nol diganti dengan NaN.

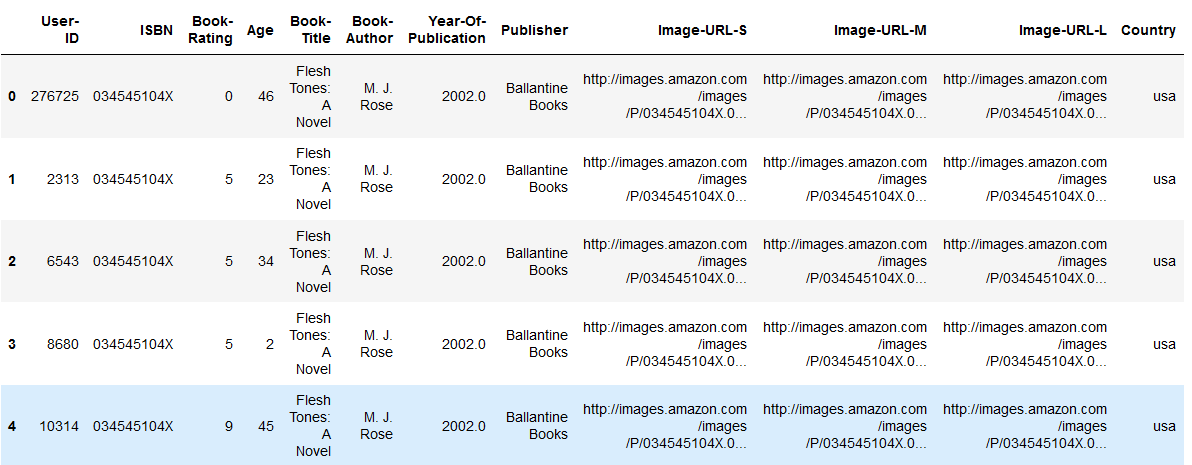


1. Data *Books* muncul banyak karakter ampersand, yang diubah menjadi hanya '& amp' sehingga dilakukan pembersihan *ampersand formatting* di kolom ‘Publisher’.



## 3.4 Construct Data

Pada fase ini, dilakukan proses pembuatan data dengan menggabungkan file ‘Books’, ‘book-Ratings’, dan ‘Users sehingga terdiri dari 12 kolom yaitu 'User-ID', 'ISBN', 'Book-Rating', 'Location', 'Age', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher', 'Image-URL-S', 'Image-URL-M', 'Image-URL-L' dan 1031136 baris dengan ukuran 102.270263671875 MB yang terlihat seperti pada gambar berikut.

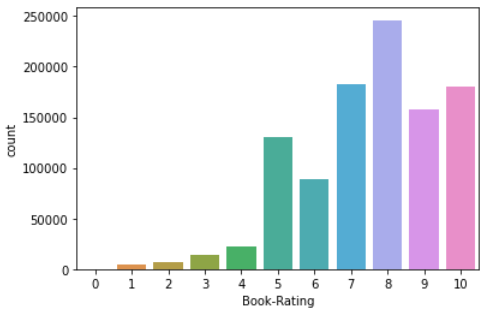


Gambar 4 Data gabungan

## 3.5 Investigate Data

Pada fase ini, dilakukan proses pencarian dan analisis informasi yang dapat diperoleh dari data yaitu :

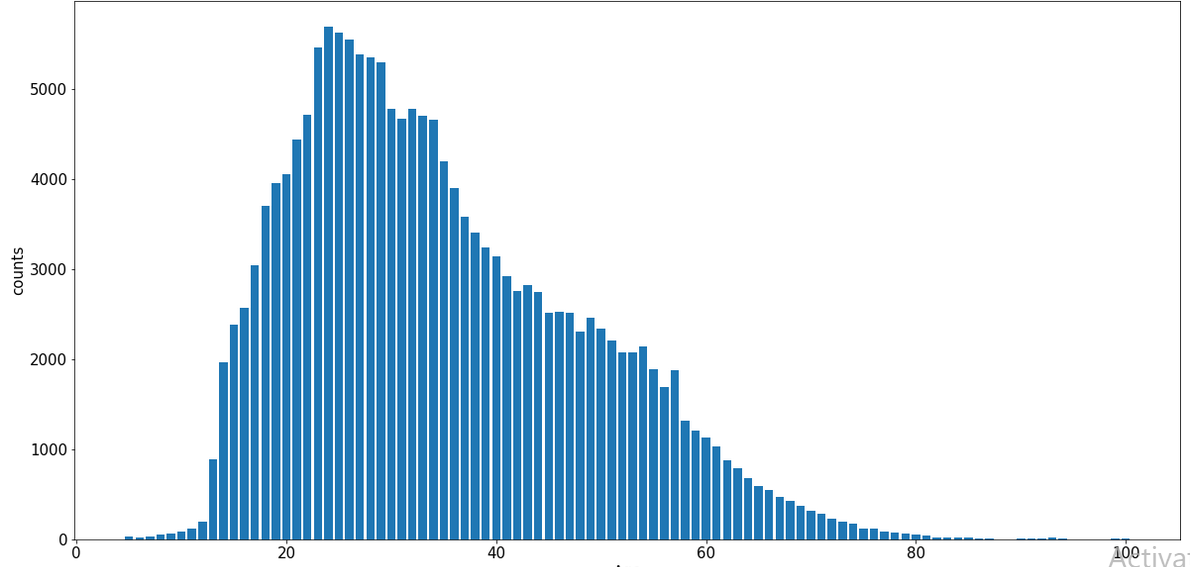
1. Rating Buku



Gambar 5 Rating Buku

Pada histogram, dapat disimpulkan bahwa skala *rating* buku adalah dari 1 hingga 10. 8 adalah rating yang paling sering diberikan *user*. *User* yang memberikan rating > 5 cenderung sedikit. Setelah dilakukan perhitungan, rata - rata rating buku adalah 7.55. Rating terbanyak diberikan oleh para *user* yang berlokasi di USA.

1. User



Gambar 6 *User*

Jumlah *user* adalah sebanyak 92106 orang. Dari histogram yang ditampilkan, terlihat bahwa sebagian besar *user* berusia akhir 20-an hingga awal 30-an.

1. Buku

Ada sebanyak 270151 yang diidentifikasi unik, sedangkan untuk total keseluruhan terdapat 1031136 buku. Buku dengan judul buku yang sama namun versinya berbeda akan memiliki ISBN berbeda - misalnya *paperback , ebook*, dll. Koleksi buku yang ada paling banyak adalah terbitan Ballantine Book. Penulis dengan jumlah buku terbanyak dalam koleksi buku adalah Stephen King. Buku yang paling banyak jumlahnya adalah buku dengan judul Wild Animus.

# Modeling

Pada tahap pemodelan *(modeling)*, dilakukan proses pemilihan algoritma yang akan digunakan untuk memodelkan sistem dan membangun model.

## 4.1 Select Modeling Technique

Metode yang akan mendukung pembangunan sistem rekomendasi buku adalah *association rule*, yaitu metode yang digunakan untuk mencari pola yang sering muncul pada beberapa item. *Association rule* akan menghasilkan *rules* untuk menentukan seberapa besar hubungan antara item A dan B. Terdapat 2 ukuran pada *rules* yang dihasilkan, yaitu *support* dan *confidence.* *Support* adalah kemungkinan A dan B muncul bersamaan. *Confidence* adalah kemungkinan munculnya B ketika A juga muncul. Untuk menghitung *frequent* *itemsets* dan membandingkan setiap *candidate itemsets*, dapat dilakukan pendekatan dengan menerapkan algoritma apriori Algoritma apriori akan digunakan untuk menemukan *frequent itemsets* yang memenuhi *minimum support* lalu mendapatkan *rule* yang memenuhi *minimum confidence* [4].

Prosedur dasar algoritma apriori adalah membangun *candidate itemsets* baru dari *frequent itemsets* sebelumnya. Kandidat ini akan diuji untuk melihat apakah *frequent* atau tidak. Iterasi algoritma apriori adalah sebagai berikut [5].

1. Buat *frequent itemsets* awal dengan menempatkan setiap item di itemset-nya sendiri. Hanya item yang sesuai dengan minimum *support* yang akan digunakan.

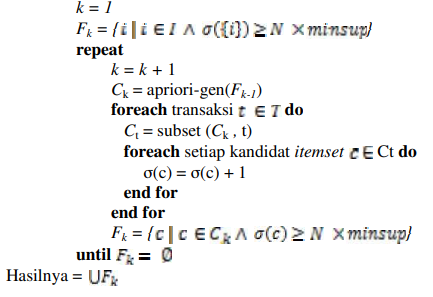
2. *Candidate itemsets* baru dibuat dari *frequent itemsets* yang paling baru ditemukan dengan mencari *superset* dari *frequent itemsets* yang ada.

3. Semua *candidate itemsets* diuji untuk dilihat apakah *frequent.* Jika *itemset* digolongkan dalam *frequent itemsets* dan *support*-nya lebih dari yang sudah ditetapkan, maka semua subsetnya termasuk ke dalam golongan *frequent itemset*, dan sebaliknya. Jika *candidate itemsets* tidak *frequent* (nilai s*upport-*nya di bawah *minimum support*)*,* maka akan dibuang. Jika tidak ada *frequent itemsets* baru dari langkah ini, lanjutkan ke langkah terakhir.

4. Simpan *frequent itemsets* yang baru ditemukan dan lanjutkan ke langkah kedua.

5. Kembalikan semua *frequent itemsets* yang ditemukan.

Apabila *frequent itemset* sudah didapatkan, maka dicari *rule* yang memenuhi nilai *confidence*. Jika *rule* yang didapat memenuhi batasan yang ditentukan, maka digolongkan sebagai *strong rules. Pseudocode* dari algoritma apriori dapat dilihat pada gambar berikut [4].



Gambar 7 Pseudocode Algoritma Apriori

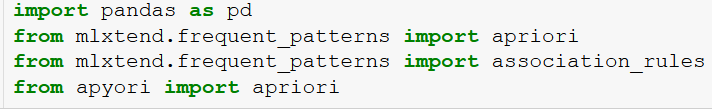
Ide utama pada algoritma ini adalah mencari *frequent itemset*, yaitu himpunan item yang memenuhi *minimum support*, menghilangkan itemset dengan frekuensi yang rendah berdasarkan level *minimum support* yang sudah ditentukan dan membangun *association rule* dari itemset yang memenuhi nilai minimum *confidence.* Untuk membentuk *candidate itemsets,* proses yang dilakukan pada algoritma apriori, yaitu J*oin Step* (Penggabungan) dimana setiap item dikombinasikan dengan item lainnya sampai tidak terbentuk kombinasi lagi dan *Prune Step* (Pemangkasan) dimana hasil item yang dikombinasikan dipangkas sesuai aturan yang ditetapkan sebelumnya [6].

## 4.2 Build Model

Pada sub bab ini akan dijelaskan proses dalam membangun model sistem rekomendasi buku dengan algoritma apriori.

1. **Implementasi Algoritma Apriori**

Langkah pertama adalah mengimpor pustaka yang diperlukan dengan menjalankan skrip berikut.



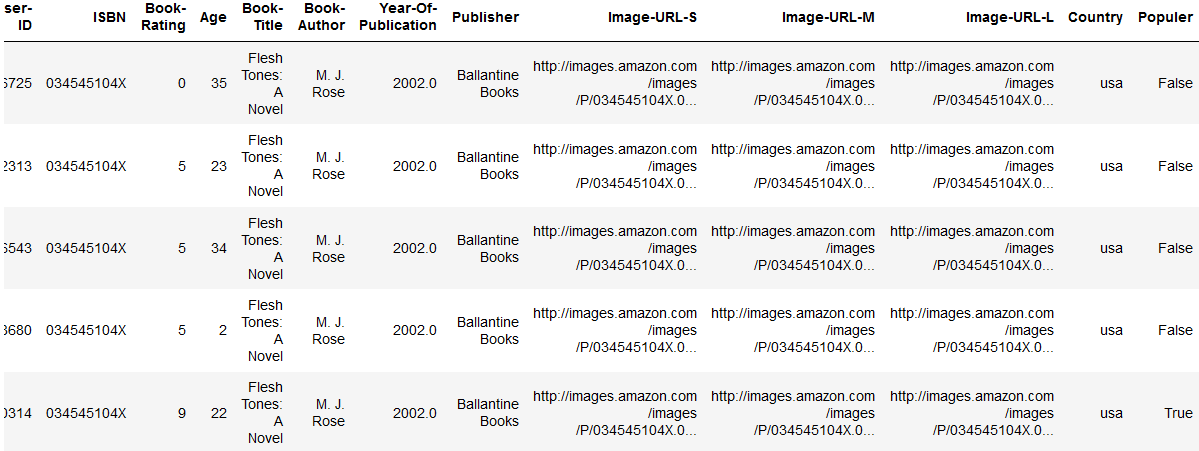
Pada skrip di atas yang diimpor adalah pandas dan apriori. Untuk menggunakan pandas, import pandas sebagai pd. *Line* 2 adalah fungsi apriori untuk mengekstrak *frequent itemsets* untuk penambangan *association rule* dan *line* 3 berfungsi untuk menghasilkan *association rule* dari *frequent itemsets.*

Tujuan dari implementasi algoritma apriori adalah menghasilkan aturan dalam bentuk berikut: “jika *user* merekomendasikan buku ini, mereka juga akan merekomendasikan buku ini”. Maka perlu ditentukan apakah *user* merekomendasikan sebuah buku dengan membuat kolom baru yaitu “Populer”, yang akan bernilai “True” jika *user* memberikan rating yang lebih besar dari 5 untuk sebuah buku.



Berikut adalah hasil dari kolom “Populer” yang dibuat pada dataset.





Dari sampel *dataset* akan dibentuk *training set.* Hal ini dilakukan untuk membantu mengurangi ukuran *dataset* yang akan dicari, membuat algoritma Apriori berjalan lebih cepat dengan memperoleh semua ulasan rating dari 3000 *user* pertama.



Metode Pandas isin () digunakan untuk memfilter *data frames*. Metode isin () membantu dalam memilih baris yang memiliki nilai tertentu dalam kolom tertentu.

Selanjutnya, dapat dibuat kumpulan data hanya dari ulasan yang ada dalam sampel.



Untuk menghitung buku - buku yang telah di *rating* oleh *user* dilakukan dengan mengelompokkan kumpulan data berdasarkan “User-ID” dan melakukan iterasi pada buku.

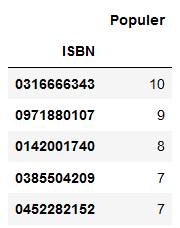


Nilai disimpan sebagai *frozenset,* memungkinkan untuk dengan cepat memeriksa apakah sebuah buku telah diberi *rating* oleh *user.* Terakhir, dapat dibuat DataFrame yang memberi tahu seberapa sering setiap buku mendapatkan ulasan yang bagus.



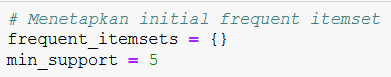
Lima buku teratas bisa dapat dilihat dengan menjalankan kode berikut.



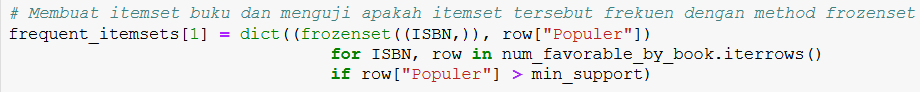


Pada iterasi pertama algoritma apriori, itemset yang baru ditemukan akan memiliki panjang 2, karena akan menjadi superset dari itemsets awal yang dibuat pada langkah pertama. Pada iterasi kedua (setelah menerapkan langkah keempat), itemsets yang baru ditemukan akan memiliki panjang 3. Hal ini memungkinkan untuk dengan cepat mengidentifikasi itemsets yang baru ditemukan, sebagaimana diperlukan pada langkah kedua.

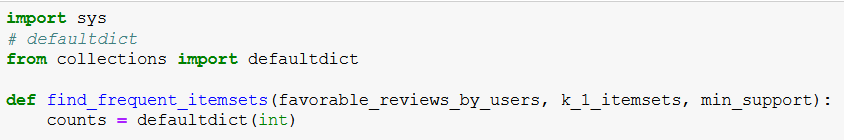
*Frequent itemsets* yang ditemukan dapat disimpan di *dictionary*, di mana kuncinya adalah panjang kumpulan item. Hal ini memungkinkan untuk dengan cepat mengakses kumpulan item dengan panjang tertentu, dan oleh karena itu merupakan *frequent itemsets* yang paling baru ditemukan, lalu tentukan *minimum support*  yang diperlukan agar *itemset* dianggap frequent. Nilai ini dipilih berdasarkan kumpulan data.



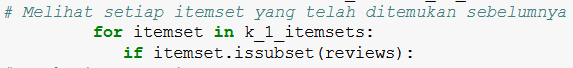
Untuk mengimplementasikan langkah pertama dari algoritma Apriori, itemset dibuat dengan setiap buku secara individual dan menguji apakah itemset tersebut *frequent* menggunakan *frozenset.*

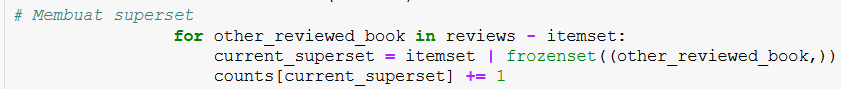


Langkah kedua dan ketiga diimplementasikan bersama-sama agar lebih efisien dengan membuat fungsi yang mengambil *frequent itemsets* yang ditemukan, membuat superset, dan kemudian melakukan pengujian apakah *frequent.*

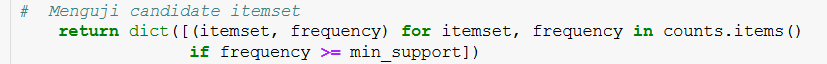


Selanjutnya, periksa setiap itemset yang ditemukan sebelumnya dan melihat apakah itu adalah subset dari set review saat ini. Jika ya, ini berarti *user* telah meninjau setiap buku di itemset.

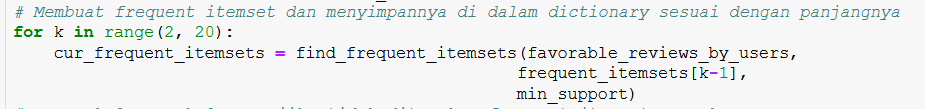


Telusuri setiap buku yang telah ditinjau *user* yang tidak ada dalam kumpulan item, membuat superset darinya, dan mencatat dalam *dictionary.*

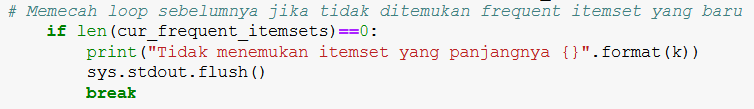
Akhiri fungsi dengan menguji kumpulan candidate itemsets mana yang memiliki *support* yang cukup dan dianggap *frequent.*

**

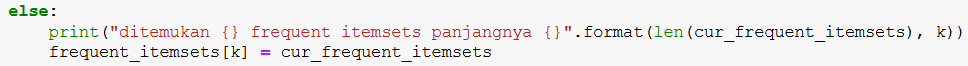
Lalu buat *loop* yang mengulang langkah-langkah algoritma apriori dan menyimpan kumpulan item baru Dalam loop ini, k mewakili panjang dari *frequent itemsets,* memungkinkan untuk mengakses itemsets sebelumnya yang paling banyak ditemukan dengan mencarinya di *frequent itemsets dictionary* menggunakan key k - 1. frequent itemsets dibuat dan disimpan dalam *dictionary* sesuai panjangnya.



Jika tidak menemukan *frequent itemsets* baru, keluar dari *loop* sebelumnya.

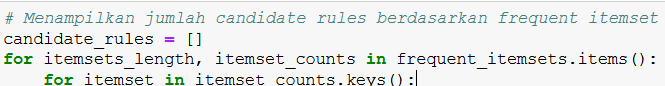


Jika ditemukan *frequent itemsets,* *loop* akan berjalan kembali.



1. **Extracting association rules**

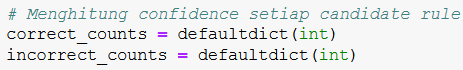
Setelah algoritma Apriori selesai, maka diperoleh daftar *frequent itemsets.* Association rule bisa dibuat dari *frequent itemset* dengan mengambil salah satu buku di itemset dan menandainya sebagai kesimpulan. Buku lain di itemset akan menjadi premis. Maka aturannya adalah jika *user* merekomendasikan semua buku dalam premis, mereka juga akan merekomendasikan kesimpulan. Pertama, membuat daftar semua aturan dari *frequent itemsets*.



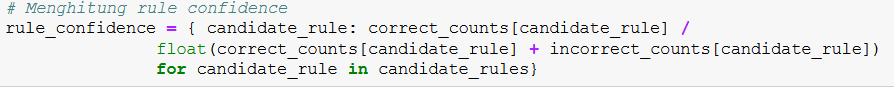
Keluaran yang dihasilkan menunjukkan aturan yang diperoleh



Dalam *rule* ini, bagian pertama (frozenset) adalah daftar buku yang ada di premis, sedangkan nomor setelahnya adalah kesimpulan. Dalam kasus pertama, jika *user* merekomendasikan buku 0316776963, mereka juga cenderung merekomendasikan buku 0316666343. Selanjutnya, menghitung *confidence* masing-masing *rule.* Prosesnya dimulai dengan membuat *dictionary* untuk menyimpan premis yang mengarah ke kesimpulan.



Kemudian hitung *confidence* untuk setiap *rule* dengan membagi jumlah *rule* yang benar dengan total berapa kali *rule* tersebut dilihat



*Print* lima *rule* teratas dengan mengurutkan *confidence dictionary* dan hasilnya adalah sebagai berikut.

*Rule 1*

*Rule:* Jika *user* merekomendasikan frozenset({'0316776963'}) mereka juga akan merekomendasikan buku dengan ISBN 0316666343

- *Confidence*: 0.500

*Rule* 2

*Rule:* Jika *user* merekomendasikan frozenset({'0316776963'}) mereka juga akan merekomendasikan buku dengan ISBN 0452282152

- *Confidence*: 0.500

*Rule* 3

*Rule*: Jika *user* merekomendasikan frozenset({'0060392452'}) mereka juga akan merekomendasikan buku dengan ISBN 0142001740

- *Confidence*: 0.500

*Rule* 4

*Rule:* Jika *user* merekomendasikan frozen set({'0452282152'}) mereka juga akan merekomendasikan buku dengan ISBN 0316776963

- *Confidence*: 0.429

*Rule* 5

*Rule:* Jika *user* merekomendasikan frozen set({'0142001740'}) mereka juga akan merekomendasikan buku dengan ISBN 0060392452

- *Confidence*: 0.375

Hasilnya hanya menampilkan ISBN, yang tidak terlalu membantu pembaca karena tidak terdapat judul buku maka dibuatfungsi yang akan mengembalikan judul buku dari ISBN-nya.



Sesuaikan *code* sebelumnya untuk menghasilkan 5 *rule* teratas yang menyertakan judul buku sehingga hasilnya sebagai berikut.

*Rule* 1

*Rule:* Jika *user* merekomendasikan *Me Talk Pretty One Day,* mereka juga akan merekomendasikan buku dengan ISBN :0316666343

- *Confidence*: 0.500

*Rule* 2

*Rule*: Jika *user* merekomendasikan *Me Talk Pretty One Day*, mereka juga akan merekomendasikan buku dengan ISBN :0452282152

*- Confidence*: 0.500

*Rule* 3

*Rule:* Jika *user* merekomendasikan *Stupid White Men ...and Other Sorry Excuses for the State of the Nation!*, mereka juga akan merekomendasikan buku dengan ISBN :0142001740

*- Confidence*: 0.500

*Rule* 4

*Rule:* Jika *user* merekomendasikan *Girl with a Pearl Earring*, mereka juga akan merekomendasikan buku dengan ISBN :0316776963

- *Confidence*: 0.429

*Rule* 5

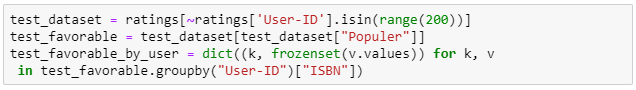
*Rule:* Jika *user* merekomendasikan *The Secret Life of Bees*, mereka juga akan merekomendasikan buku dengan ISBN :0060392452

- *Confidence*: 0.375

# 5. Evaluation

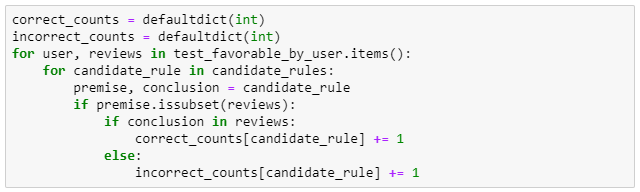
Pada tahap evaluasi (*Evaluation*), dilakukan proses evaluasi model sistem rekomendasi yang telah dibangun. Tahap ini dilakukan untuk mengetahui performa *association rule* dengan menggunakan beberapa data uji dan menghitung *test set confidence* atau nilai *confidence* dari setiap *rule* dalam *testing set*.

Langkah pertama adalah mengekstrak dataset yang akan digunakan sebagai *test dataset* dalam proses evaluasi.

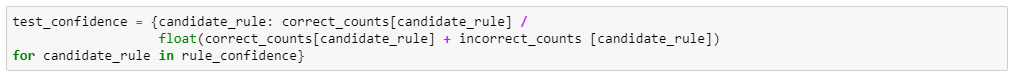


Pada potongan kode diatas,dapat dilihat bahwa tahap evaluasi menggunakan *training set* namun tidak seluruh data dan hanya menggunakan 200 data *users* berdasarkan “User-ID”. Pada *test set* ini juga disertakan “Populer” di setiap *user* pada dataset.

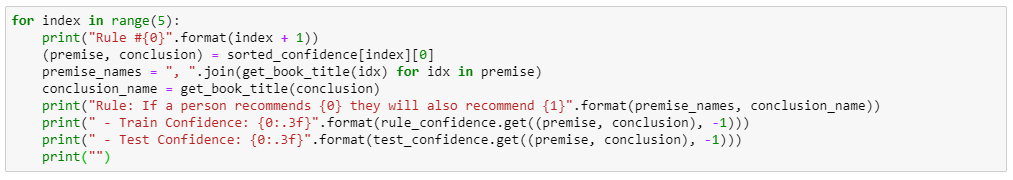
Langkah selanjutnya adalah menghitung *instances* yang benar dalam menghasilkan kesimpulan dari setiap premis dan berdasarkan *testing set* yang digunakan.



Setelah menghitung *instances* dari premis, lalu akan dihitung nilai *confidence* dari setiap *rule.*



Langkah terakhir adalah mencetak nilai *train and test confidence* tertinggi di setiap *association rules* yang diperoleh.



Potongan kode diatas akan menghasilkan 5 *association rules* dengan nilai *confidence* tertinggi yaitu sebagai berikut.

Rule #1

Rule: If a person recommends Me Talk Pretty One Day they will also recommend The Lovely Bones: A Novel

- Train Confidence: 0.500

- Test Confidence: 0.500

Rule #2

Rule: If a person recommends Me Talk Pretty One Day they will also recommend Girl with a Pearl Earring

- Train Confidence: 0.500

- Test Confidence: 0.500

Rule #3

Rule: If a person recommends Stupid White Men ...and Other Sorry Excuses for the State of the Nation! they will also recommend The Secret Life of Bees

- Train Confidence: 0.500

- Test Confidence: 0.500

Rule #4

Rule: If a person recommends Girl with a Pearl Earring they will also recommend Me Talk Pretty One Day

- Train Confidence: 0.429

- Test Confidence: 0.429

Rule #5

Rule: If a person recommends The Secret Life of Bees they will also recommend Stupid White Men ...and Other Sorry Excuses for the State of the Nation!

- Train Confidence: 0.375

- Test Confidence: 0.375

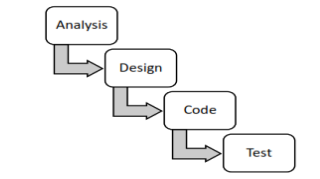
Berdasarkan hasil evaluasi diatas, *rule* dengan nilai *confidence* paling tinggi adalah *rule* pertama, kedua, dan ketiga yaitu 0.500. Sehingga dapat disimpulkan bahwa performa model sistem rekomendasi yang dibangun tergolong cukup baik.

# 6. Deployment

Pada bab ini akan dijelaskan  mengenai *deployment plan* serta proses monitor dan *maintain* pada saat membangun sebuah sistem agar dapat dilihat apakah sistem yang akan dibangun sudah sesuai dengan yang diharapkan.

1. **Deployment Plan**

Berdasarkan buku algoritma dan data mining, tahap deployment merupakan tahap pembuatan laporan akhir atau yang disebut juga dengan *final report,* selain itu, tahap deployment dapat juga didefinisikan sebagai penerapan proses data mining secara paralel. Pada saat membangun sebuah sistem, setelah tahap evaluasi dilakukan dan sudah mendapatkan hasil sesuai dengan yang diharapkan, maka pada saat proses pengembangan sistem rekomendasi buku yang akan dibangun. Namun pada saat membangun sistem rekomendasi buku, tidak dilakukan tahap *deployment* lebih lanjut hingga terbangun sebuah sistem seutuhnya. metode yang dapat digunakan adalah waterfall seperti pada Gambar 8.

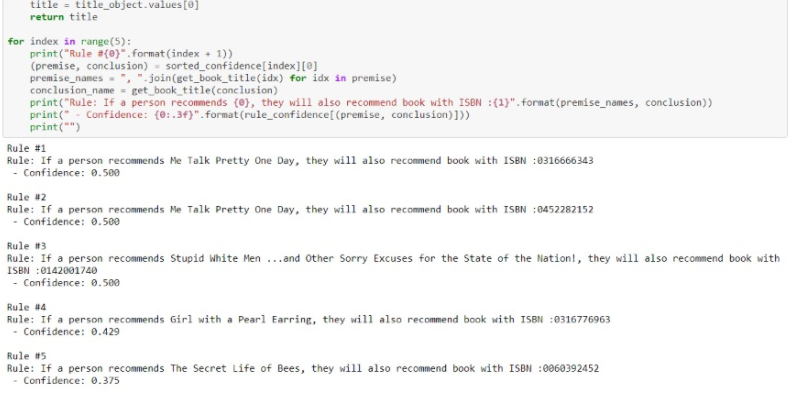


Gambar 8 Metode *Waterfall*

Pendekatan pada proses pembangunan sistem rekomendasi ini dimulai dari tahap *analysis,* dimana pada tahap ini yang dilakukan adalah menentukan kebutuhan sistem yang akan dibangun seperti data *rating* yang diperlukan pada saat pembangunan sistem rekomendasi buku. Selanjutnya adalah tahap *design,* dimana pada tahap dilakukan *design* sebuah sistem untuk mempermudah membangun sistem. Pada tahap *code* dilakukan sebuah implementasi berupa penulisan kode dengan editor untuk membangun sebuah sistem rekomendasi buku serta dilakukan tahap *testing* untuk mengetahui apakah sistem rekomendasi buku yang sudah dibangun sudah sesuai dengan kebutuhan user. Dan pada proses pembangunan sistem ini, *user*yang melakukan pada tahap *testing* ini sendiri adalah pihak *developer.*

1. **Monitoring and Maintenance**

Pada tahap *deployment* dalam proses pembangunan sistem rekomendasi buku dengan menggunakan algoritma apriori, terdapat kelemahan di dalam sistem yang sudah dibangun. Kelemahan sistem yang dimaksud dapat dilihat seperti pada Gambar 9.



Gambar 9 *Rule* pada model

Pada potongan *code* proses pembangunan sistem dapat dilihat pada bagian *association rule* yang ditampilkan pada gambar diatas, dimana buku direkomendasikan berdasarkan nomor ISBN pada buku, sementara *user* kurang tertarik apabila pada saat akan mencari buku yang diinginkan dengan mencari buku berdasarkan ISBN buku karena pada dasarnya *user* tidak menghafal nomor ISBN buku yang akan dicari, maka dari itu tim *developer* akan tetap melakukan *maintenance* dan tetap melakukan *develop* terhadap sistem rekomendasi buku yang sudah dibangun agar sistem dapat menyajikan rekomendasi berdasarkan judul buku sesuai dengan buku yang ingin dicari oleh *user*.

# Referensi

[1] E. Cano and M. Morisio, “Characterization of public datasets for Recommender Systems,” in *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, Sep. 2015, pp. 249–257, doi: 10.1109/RTSI.2015.7325106.

[2] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining Practical Machine Learning Tools and Techniques*. San Francesco: Elsevier, 2005.

[3] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in *Proceedings of the 14th international conference on World Wide Web - WWW ’05*, 2005, p. 22, doi: 10.1145/1060745.1060754.

[4] N. Wandi and R. A. H. A. Mukhlason, “Pengembangan Sistem Rekomendasi Penelusuran Buku dengan Penggalian Association Rule Menggunakan Algoritma Apriori (Studi Kasus Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur),” *J. Tek. ITS*, vol. 1, pp. 445–449, 2012.

[5] R. Layton, *Learning Data Mining with Python - Second Edition*. Birmingham: Packt Publishing, 2017.

[6] S. Saefudin and D. Fernando, “Penerapan Data Mining Rekomendasi Buku Menggunakan Algoritma Apriori,” *JSiI (Jurnal Sist. Informasi)*, vol. 7, p. 50, 2020.