

Association Rule Mining on Library Books Lending Data Using Combination of Apriori And Jaccard Similarity

Muhammad Hezby Al Haq
Informatics Department
Sebelas Maret University
36 A Ir. Sutami Street, Surakarta
mail@hezby.com

Ristu Saptono
Informatics Department
Sebelas Maret University
36 A Ir. Sutami Street, Surakarta
ristu.saptono@staff.uns.ac.id

Sarngadi Palgunadi
Informatics Department
Sebelas Maret University
36 A Ir. Sutami Street, Surakarta
palgunadi@gmail.com

ABSTRACT

UPT Perpustakaan UNS has 37,271 collections and on average 75,316 annual circulations of the book that is managed by UNSLA (UNS Library Automation). An analysis is needed to discover valuable information that can be used for various purposes. Association rule mining is one of data mining techniques to look for relationship pattern in the market basket data. Apriori algorithm is commonly used in association rule mining. However, Apriori has limitations in conducting association rule mining on sparse data. Jaccard Similarity algorithm is used to find the similarities between the two sets. Application of Jaccard Similarity to the association rule mining can find association rule on sparse data. This research was conducted to determine the consistency of association rule generated by the combination of both Apriori and Jaccard Similarity compared to regular Apriori and Jaccard Similarity on the book lending data of UPT Library UNS. Data are grouped into ten different categories of books and split by month and year. Association rule mining is done by using all three methods. Association rules produced by each method compared for consistency in the known month and year. As a result, it is known that the association rule mining using a combination of Apriori and Jaccard Similarity is more consistent than the original Apriori and Jaccard Similarity. However, association rule mining using Jaccard Similarity generate more variation than Apriori and combination.

Keywords

Library, Data Mining, Association Rule, Apriori, Jaccard Similarity

1. PENDAHULUAN

Universitas Sebelas Maret (UNS) melalui Unit Pelaksana Teknis (UPT) Perpustakaan Pusat memiliki wewenang dalam kaitannya dengan fasilitas penyediaan informasi [1]. Sebagai salah satu pusat informasi di UNS, UPT Perpustakaan Pusat mempunyai berbagai layanan yang dapat dimanfaatkan oleh entitas akademis UNS. Salah satu layanan yang dimiliki adalah layanan peminjaman pustaka dimana dalam layanan ini UPT Perpustakaan Pusat memiliki kewajiban untuk melayani peminjaman dan pengembalian bahan pustaka. Dalam menunjang layanan ini UPT Perpustakaan Pusat secara teratur melakukan pengadaan terhadap koleksi pustaka.

Saat ini UPT Perpustakaan Pusat memiliki jumlah koleksi sebanyak 37.271 buah pustaka yang dikelola dalam sistem UNSLA (UNS Library Automation), sedangkan rata-rata

peminjaman koleksi pustaka ini dari tahun 2006 hingga tahun 2014 adalah 75.316 pertahun. UNS memiliki rata-rata jumlah entitas akademis UNS 20.669 dari tahun 2006 sampai tahun 2014. Berdasarkan data tersebut dapat diketahui bahwa jumlah permintaan pustaka setiap tahunnya tiga kali lipat dari jumlah koleksi yang ada. Sedangkan jika dilihat dari jumlah entitas akademis UNS, jumlah peminjaman yang ada adalah empat kali lipat dari jumlah entitas akademis UNS yang ada. Ada sebanyak 698.725 baris yang masing-masing barisnya menampung detail catatan peminjaman koleksi pustaka [2].

Dalam upaya meningkatkan kualitas koleksi pustaka diperlukan strategi berdasar analisis peminjaman pustaka [3]. Analisis diperlukan untuk menggali informasi berharga yang dapat digunakan untuk berbagai kepentingan. Berdasarkan hal ini maka diperlukan metode *data mining* pada data peminjaman yang ada. *Data Mining* adalah proses menemukan informasi berharga dari suatu kumpulan besar data secara otomatis [4].

Ada berbagai macam teknik dalam *data mining* salah satunya adalah *association analysis*. *Association analysis* adalah teknik yang dikembangkan untuk mencari pola hubungan pada data *market basket*. Data *market basket* diidentifikasi sebagai baris dan kolom dimana baris merepresentasikan transaksi yang dilakukan oleh pengguna dan kolom merepresentasikan item yang diambil oleh pengguna. Hasil dari *association analysis* adalah ekspresi implikasi $X \rightarrow Y$ dimana X dan Y adalah bagian dari itemset. Ekspresi ini disebut dengan *association rule*. Kekuatan *association rule* diukur menggunakan dua cara, yaitu *support* dan *confidence*. *Association analysis* dibagi menjadi 2 tahap, yaitu *frequent itemset generation* dan *rule generation* [4].

Ada dua permasalahan pada penerapan *association analysis*. Pertama, bagaimana mendapatkan *association rule* dalam waktu yang singkat. Kedua, bagaimana mendapatkan *association rule* yang akurat menggambarkan pola *market basket* [4].

Agrawal (1993) mengusulkan algoritma *apriori* sebagai upaya untuk memecahkan permasalahan *association analysis*. *Apriori* adalah algoritma yang digunakan untuk mempercepat proses komputasi pada tahap *frequent itemset generation* dengan mengeliminasi kandidat rule berdasarkan nilai *support*. Prinsip utama *apriori* adalah jika satu item memiliki nilai *support* tinggi maka subset dari item tersebut juga memiliki *support* tinggi dan sebaliknya. Algoritma *apriori* menjadi algoritma paling populer dalam *association analysis* dikarenakan pendekatannya yang sederhana dan kemampuannya yang mumpuni [5].

Selain *apriori*, beberapa usaha lain dibuat untuk memecahkan permasalahan *association analysis*. Cohen et al (2001) mengembangkan pencarian *association rule* menggunakan

modifikasi algoritma *jaccard similarity* sehingga dimungkinkan untuk mencari *association rule* tanpa *support* dan *confidence*. *Jaccard similarity* adalah algoritma yang digunakan untuk mencari kesamaan antar dua set. Penerapan *jaccard similarity* pada *association analysis* diketahui dapat menemukan *association rule* pada dataset dengan tingkat *support* rendah [6].

Penelitian untuk mengkombinasikan *apriori* dan *jaccard similarity* masih belum banyak. Padahal berdasarkan Cohen et al (2001) algoritma *jaccard similarity* diketahui dapat digunakan sebagai faktor *interestingness* selain *support* dan *confidence* pada pencarian *association rule*. Perlu ada usaha untuk mengkombinasikan ketiga faktor *interestingness* tersebut supaya dapat diketahui bagaimana efek jika ketiga faktor *interestingness* tersebut digunakan secara bersamaan [6].

Penelitian ini dilakukan untuk mencari tahu bagaimana konsistensi *association rule* yang dihasilkan algoritma *apriori*, *jaccard similarity* dan kombinasi keduanya pada data peminjaman buku UPT perpustakaan UNS. *Jaccard similarity* dan *apriori* diketahui dapat digunakan untuk menemukan *association rule*, namun penelitian untuk mencari konsistensi *association rule* yang dihasilkan oleh kombinasi *apriori* dan *jaccard similarity* masih jarang dilakukan. Pada penelitian ini dibandingkan konsistensi *association rule* yang dihasilkan dari penggunaan algoritma *apriori*, *jaccard similarity*, dan kombinasi *jaccard similarity* dengan *apriori*.

2. DASAR TEORI

2.1 UNS Library Automation (UNSLA)

UNSLA adalah sistem informasi perpustakaan yang terintegrasi. UNSLA meliputi katalog buku secara online, administrasi, keuangan, dan inventori perpustakaan (UNSLA, 2016). UNSLA menggunakan MySQL sebagai *Database Management System* (DBMS) sistem informasi. Terdapat 82 komponen tabel dalam database UNSLA.[2]

2.2 Database Management System (DBMS)

DBMS adalah koleksi program yang membuat pengguna dapat membuat dan mengatur data secara lebih mudah [7]. DBMS adalah *general-purpose* software yang memfasilitasi pengguna untuk melakukan proses *defining*, *constructing*, *manipulating*, dan *sharing*. Proses *defining* melibatkan spesifikasi tipe data, struktur dan *constraint* untuk data supaya dapat disimpan di DBMS. *Constructing* database adalah proses menyimpan data ke medium yang diatur oleh database misalnya harddisk. *Manipulating* database melibatkan proses untuk mendapatkan data, mengupdate data, dan membuat report dari data. Proses *sharing* memungkinkan beberapa pengguna untuk dapat menggunakan DBMS secara bersamaan.

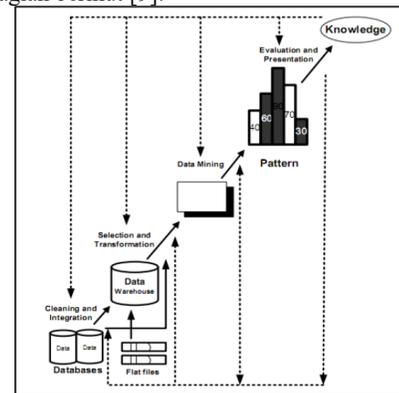
2.3 MySQL

MySQL adalah salah satu dari banyak DBMS. MySQL adalah open source DBMS dengan lisensi GPL versi 2, pengembangan MySQL dilakukan oleh komunitas dari seluruh dunia. MySQL dapat digunakan secara gratis maupun komersil [8].

2.4 Knowledge Discovery

Knowledge discovery adalah keseluruhan proses mengambil informasi berharga dari data. Berdasarkan prosesnya,

Jiawei Han (2006) membagi *knowledge discovery* menjadi beberapa bagian berikut [9].



Gambar 1. Knowledge Discovery

2.5 Preprocessing

Belum tentu data dapat langsung digunakan dalam proses *mining*. Data tersebut bisa saja perlu diolah terlebih dahulu untuk mendapatkan data yang siap untuk digunakan. Proses pengolahan data mentah menjadi data yang siap untuk digunakan disebut dengan *preprocessing* [9].

2.5.1 Data Cleaning

Data cleaning adalah proses untuk menghilangkan distorsi nilai (*noise*) dan data yang inkonsisten.

2.5.2 Data Integration

Data integration adalah proses menggabungkan sumber data dari berbagai sumber.

2.5.3 Data Selection

Data selection adalah proses pengambilan data yang relevan untuk dilakukan proses analisis pada tahap selanjutnya.

2.5.4 Data Transformation

Data transformation adalah proses dimana data dilakukan transformasi supaya cocok digunakan pada proses *data mining*.

2.5.5 Data Mining

Data mining adalah proses utama diterapkannya metode-metode *data mining* pada data untuk mengambil pola data.

2.5.6 Evaluasi Pola

Proses *data mining* memiliki potensi untuk menghasilkan pola palsu yang terjadi hanya karena kebetulan. Evaluasi pola dilakukan untuk mencegah munculnya pola palsu tersebut.

2.5.7 Knowledge Presentation

Knowledge presentation adalah teknik membuat visualisasi dan representasi pengetahuan yang didapat.

2.6 Data Mining

Data mining adalah proses menemukan informasi berharga dari suatu kumpulan besar data secara otomatis. Data mining adalah bagian dari *knowledge discovery in databases* (KDD) yang mana adalah seluruh proses mengubah data mentah menjadi informasi [4].

Metode-metode data mining dibagi menjadi dua kategori yaitu predictive dan descriptive. Predictive digunakan untuk memprediksi nilai suatu variabel berdasarkan variabel yang lain. Descriptive digunakan untuk mencari pola yang dapat menyimpulkan hubungan antar data [4].

2.7 Association Analysis

Ada beberapa metode yang digunakan dalam *data mining*, *Association analysis* adalah salah satunya. Tan dan Steinbach (2005) menyatakan bahwa *association analysis* adalah metodologi yang digunakan untuk mencari hubungan tersembunyi antar data. Hubungan antar data yang ditemukan disebut sebagai *association rule*. *Association analysis* mulanya dikembangkan untuk menyelesaikan permasalahan pada *market basket data*, namun saat ini *association analysis* juga dapat diterapkan ke berbagai macam data lain seperti *geo-spatial*. *Association analysis* termasuk dalam *predictive data mining* yang mana fungsi utamanya adalah untuk memprediksi nilai berdasarkan hubungan antar data yang ditemukan [4].

2.8 Market Basket Data

Market basket data adalah jenis data spesial dimana setiap record berisi urutan item. Contoh dari *market basket data* adalah data pembelian barang di supermarket dimana masing-masing record berisi daftar barang yang dibeli [9].

2.9 Binary Representation

Market basket data dapat direpresentasikan dalam format biner seperti yang terlihat pada Tabel 2 dimana masing-masing baris merepresentasikan transaksi dan kolom merepresentasikan barang. Nilai kolom dapat berupa satu jika barang muncul dalam transaksi atau nol jika sebaliknya. Karena kemunculan barang dalam transaksi lebih penting untuk dipertimbangkan dibanding absennya, maka proses komputasi variabel barang dapat direpresentasikan menjadi *binary market basket data* [4].

2.10 Itemset dan Support Count

Ambil $I = \{i_1, i_2, i_3, \dots, i_n\}$ sebagai set semua barang dalam *market basket data*. Dan $T = \{t_1, t_2, t_3, \dots, t_d\}$ sebagai set semua transaksi. Setiap transaksi t_n terdiri dari subset barang yang terdapat pada I . Dalam *association analysis*, koleksi dari kosong atau lebih barang disebut sebagai *itemset* [4].

Support count adalah salah satu karakteristik dari *itemset*, *support count* menentukan jumlah kemunculan *itemset* dalam transaksi [4]. Secara matematis, *support count* $\sigma(X)$ untuk *itemset* X dapat dideklarasikan sebagai berikut.

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}|$$

dimana simbol $|\cdot|$ mengindikasikan jumlah elemen dalam sebuah set.

2.11 Association Rule

Association rule adalah ekspresi implikasi yang berbentuk $X \rightarrow Y$, dimana X dan Y memiliki hubungan disjoint dalam satu *itemset*. Kekuatan *association rule* diukur dengan menggunakan istilah *support* dan *confidence*. *Support* menentukan seberapa sering sebuah *rule* muncul dalam keseluruhan transaksi. Sedangkan *confidence* menentukan seberapa sering item dalam Y muncul dalam transaksi yang memiliki X [4].

$$\text{Support}, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$\text{Confidence}, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

2.12 Association Rule Discovery

Pencarian *association rule* dapat dilakukan dengan mencari semua *rule* yang memiliki *support* $\geq \text{minsup}$ dan *confidence* $\geq \text{minconf}$ dimana *minsup* dan *minconf* ditentukan oleh pengguna. Pencarian *association rule* dengan menggunakan pendekatan tersebut dikenal sebagai pendekatan *brute force*. Waktu komputasi yang dibutuhkan oleh pendekatan tersebut bernilai eksponensial [4].

Untuk dapat memecahkan permasalahan eksponensial tersebut, dibuat strategi memecah komputasi pencarian *association rule* menjadi dua tahap [4].

2.12.1 Frequent Itemset Generation

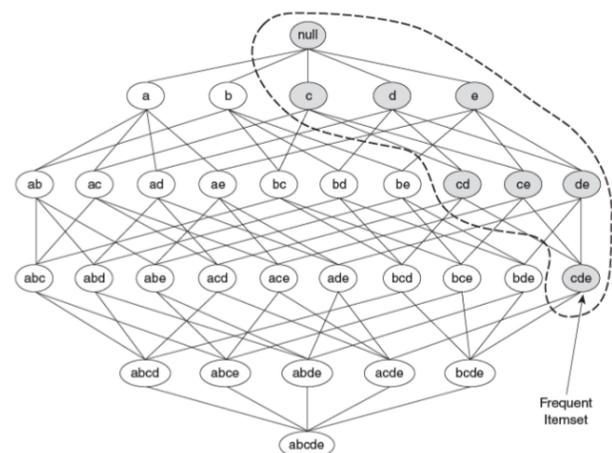
Tujuan tahap ini adalah mencari semua *itemset* yang memenuhi nilai *minsup*. *Itemset* hasil disebut sebagai *frequent itemset*.

2.12.2 Rule Generation

Tujuan utama tahap ini adalah mengambil semua *rule* dari *frequent itemset* yang memiliki nilai *confidence* tinggi. *Rule* hasil disebut sebagai *strong rule*.

2.13 Apriori

Apriori adalah algoritma yang digunakan untuk mempercepat proses *frequent itemset generation*. Algoritma *apriori* bekerja dengan prinsip bahwa jika suatu *itemset* memiliki frekuensi kemunculan tinggi maka subsetnya juga memiliki frekuensi kemunculan tinggi [4]. Katakan $\{c, d, e\}$ adalah *itemset* yang memiliki frekuensi kemunculan tinggi. Untuk membentuk *itemset* $\{c, d, e\}$ jelas dibutuhkan transaksi yang melibatkan $\{c\}, \{d\}, \{e\}, \{c, d\}, \{d, e\}, \{c, e\}$. Hasilnya, jika $\{c, d, e\}$ memiliki frekuensi tinggi maka semua subsetnya juga memiliki frekuensi tinggi.



Gambar 2. Ilustrasi Apriori (Tan, 2005)

2.14 Jaccard Similarity

Jaccard similarity pada dasarnya digunakan untuk mengukur kesamaan dalam dua atau lebih set [5]. Ambil S_1 dan S_2 sebagai dua buah set. *Jaccard coefficient* dapat dirumuskan sebagai berikut.

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

Jaccard similarity dapat juga diterapkan pada kasus *multiple set* sebagai berikut

$$J(S_1, S_2, \dots, S_k) = \frac{|\cap S_k|}{|\cup S_k|}$$

2.15 Association Rule Mining Menggunakan Jaccard Similarity

Penelitian untuk menerapkan *jaccard similarity* pada pencarian *association rule* telah dilakukan oleh Cohen et all (2001). Dalam penelitian tersebut Cohen et all mengajukan 3 langkah untuk mendapatkan *rule*, yaitu *compute signature*, *generate candidate*, dan *prune candidate*. Dua langkah awal digunakan sebagai modifikasi tahap *frequent itemset generation* dan langkah akhir digunakan sebagai *generate candidate*. Penghitungan *jaccard similarity* menggunakan konsep kesamaan kemunculan masing-masing itemset terhadap semua transaksi yang ada. Jika kesamaan itemset melebihi nilai koefisien *jaccard similarity* (*minsim*) maka itemset tersebut saling berkaitan [6].

2.15.1 Compute Signature

Pada tahap ini *market basket data* perlu ditransformasikan sehingga dapat dilakukan komputasi *jaccard similarity*. Cohen (2001) merepresentasikan *market basket data* dalam bentuk *binary matrix*. Proses transformasi *market basket data* menjadi *binary matrix* dilakukan pada tahap ini. Selain transformasi menjadi *binary matrix*, pada tahap ini dapat dilakukan optimasi *binary matrix* sehingga proses komputasi *jaccard similarity* dapat lebih cepat. Cohen (2001) melakukan optimasi berbasis algoritma *minhash*.

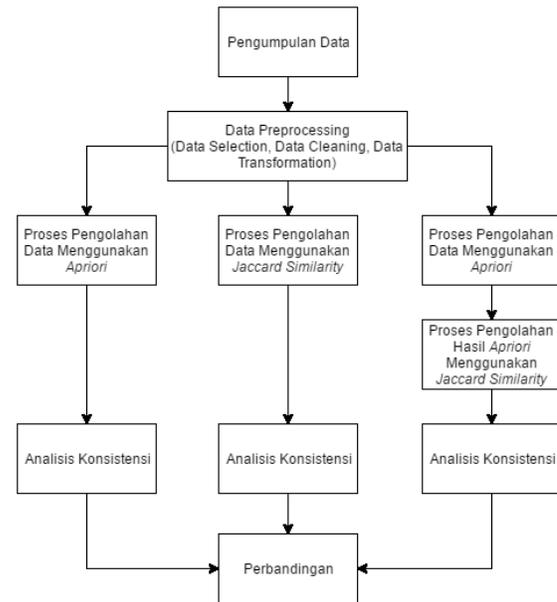
2.15.2 Generate Candidate

Pada tahap ini semua itemset dihitung tingkat kemiripannya kemunculan dengan menerapkan *jaccard similarity* terhadap itemset. Kombinasi itemset dibuat terlebih dahulu dari seluruh item yang ada. Kemudian dilakukan komputasi *jaccard similarity* terhadap seluruh itemset. *Jaccard coefficient* kemudian diurutkan untuk mempermudah proses selanjutnya.

2.15.3 Prune Candidate

Tahap ini adalah tahap pengambilan kandidat aturan kuat dengan menerapkan *minsim*. Semua itemset yang memiliki nilai *jaccard coefficient* diatas *minsim* diambil sebagai *association rule*. Bentuk *association rule* yang ditemukan oleh *jaccard similarity* berupa itemset yang saling berkaitan.

3. METODOLOGI



Gambar 3. Metodologi Penelitian

3.1 Pengumpulan Data

Data yang digunakan berasal dari database UNSLA. Proses pengambilan data adalah dengan mengajukan pengambilan data *dump* ke UPT Perpustakaan Pusat UNS. Perijinan untuk pengambilan data diperoleh dari ketua UPT Perpustakaan Pusat UNS. Data diambil langsung dari server database UNSLA.

3.2 Preprocessing

Pada tahap *preprocessing* dilakukan 3 metode, yaitu *data selection*, *data cleaning* dan *data transformation*.

Pada data *selection* dipilih hanya tabel yang berhubungan dengan transaksi peminjaman koleksi pustaka saja. Pada tahap ini diambil tabel *tbl_anggota*, *tbl_golongan*, *master_buku*, dan *sirkulasi*. Data pada 3 tabel tersebut digabungkan menjadi satu tabel dengan tabel utama adalah tabel *sirkulasi*.

Pada data *cleaning* dilakukan filter terhadap redundansi kolom *id_user* dan *golongan buku*. Semua baris yang memuat informasi ganda dihilangkan supaya tidak ada informasi ganda yang mengganggu proses komputasi pada tahap selanjutnya.

Pada data *transformation* dilakukan transformasi data peminjaman menjadi *binary market basket data*. Transformasi data dilakukan supaya dapat dilakukan perbandingan antara *apriori* dan *jaccard similarity*. *Jaccard similarity* membutuhkan data supaya berbentuk matrix. *Apriori* sebenarnya dapat langsung dicari menggunakan *market basket data*, namun untuk menghasilkan perbandingan yang akurat data terlebih dahulu disamakan bentuknya menjadi *binary market basket data*.

Tabel 1. Binary Market Basket Data

Itemset	000	100	200	300	400	500	600	700	800	900
1	0	1	1	1	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0	0

3	0	0	0	0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

3.3 Pengolahan Data Menggunakan Apriori

Binary market basket data yang didapat pada tahap preprocessing dikomputasi melalui 2 tahap, yaitu *frequent itemset generation* dan *rule generation*.

Pada *frequent itemset generation* dilakukan pencarian itemset yang memiliki nilai *support* melebihi *minsup*. Komputasi pencarian itemset ini menggunakan kaedah *apriori*. Semua itemset yang memiliki nilai *support* diatas *minsup* adalah *frequent itemset*. Misal *minsup* ditentukan bernilai 0.33, dari data pada tabel 1 diperoleh *frequent itemset* {100}, {200}, {100,200} seperti yang ditunjukkan pada Tabel 2 dan 3.

Tabel 2. Contoh Penerapan Apriori pada 1-itemset

Itemset	Support
{100}	0.33
{200}	0.33

Tabel 3. Contoh Penerapan Apriori pada 2-itemset

Itemset	Support
{100, 200}	0.33

Pada *rule generation* semua *frequent itemset* dihitung nilai *confidence*. Setelah didapat nilai *confidence* semua itemset, semua itemset yang memiliki nilai *confidence* diatas nilai *minconf* diambil sebagai *association rule*. Nilai *minconf* diobservasi untuk mencari nilai optimal yang dapat digunakan pada tahap analisis konsistensi. Misal *minconf* ditentukan bernilai 0.9, dari *frequent itemset* yang diperoleh pada Tabel 2 dan 3, diperoleh *association rule* {100} → {200} dan {200} → {100} seperti yang ditunjukkan pada tabel 4.

Itemset	Support
{100, 200}	0.33

Pada tahap ini dilakukan optimasi pada *frequent itemset generation* dan *rule generation*. Optimasi pada *frequent itemset generation* dilakukan untuk mendapatkan itemset yang cukup banyak dan nilai *support* yang cukup tinggi. Itemset dibutuhkan jumlah yang cukup banyak untuk dapat menghasilkan *association rule* yang lebih bervariasi, sedangkan nilai *support* dibutuhkan cukup tinggi supaya nilai *confidence association rule* yang dihasilkan juga tinggi. Optimasi pada *rule generation* dilakukan untuk mendapatkan jumlah *association rule* yang cukup banyak dan nilai *confidence* yang cukup tinggi. *Association rule* dibutuhkan cukup banyak supaya analisis konsistensi dapat menemukan semakin banyak pola konsistensi, sedangkan nilai *confidence* dibutuhkan cukup tinggi untuk mengeliminasi *association rule* yang terjadi hanya karena kebetulan.

3.4 Pengolahan Data Menggunakan Jaccard Similarity

Binary market basket data yang didapat dari preprocessing dikomputasi melalui 2 tahap, yaitu *generate candidate* dan *prune candidate*. Tahap *compute signature* tidak perlu dilakukan karena data pada tahap ini telah berbentuk *binary market basket data*.

Sehingga data dapat langsung diolah menggunakan *jaccard similarity*.

Tabel 4. Generate Candidate pada Penerapan Jaccard Similarity

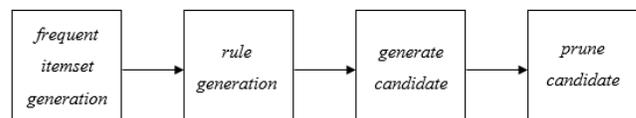
Itemset	Jaccard similarity
{100, 200}	100%
{100, 300}	50%
{200, 300}	50%

Pada tahap *generate candidate* nilai *jaccard similarity* semua itemset dihitung kemudian diurutkan. Misal dari data pada Tabel 6 diperoleh tabel hasil proses *generate candidate* pada Tabel 10. Setelah diurutkan, semua itemset yang memiliki nilai *jaccard similarity* melebihi nilai *minimum jaccard similarity* yang ditetapkan diambil sebagai *association rule* pada tahap *prune candidate*. Dari Tabel 10, dengan ditentukan nilai *minimum jaccard similarity* 95% diperoleh *association rule* {100} → {200} dan {200} → {100}.

Pada tahap pengolahan dilakukan optimasi pada proses *prune candidate*. Optimasi pada tahap *prune candidate* dilakukan untuk mendapatkan *association rule* yang cukup banyak dan memiliki nilai *jaccard coefficient* cukup tinggi. *Association rule* dibutuhkan cukup banyak supaya analisis konsistensi dapat menemukan semakin banyak pola konsistensi, sedangkan nilai *jaccard coefficient* dibutuhkan cukup tinggi untuk mengeliminasi *association rule* yang terjadi hanya karena kebetulan.

3.5 Pengolahan Hasil Apriori Menggunakan Jaccard Similarity

Pada tahap ini hasil *association rule apriori* diolah menggunakan *jaccard similarity*. Pengolahan menggunakan *jaccard similarity* dilakukan setelah tahap *frequent itemset generation* dan *rule generation*.



Gambar 4. Kombinasi Apriori dan Jaccard Similarity

Pengolahan data dilakukan dengan 4 tahap, yaitu *frequent itemset generation*, *rule generation*, *generate candidate* dan *prune candidate*. Dua tahap awal merupakan tahapan algoritma *apriori* sedangkan dua tahap terakhir merupakan implementasi *jaccard similarity*.

Tahap *frequent itemset generation* dan *rule generation* memiliki implementasi sama seperti tahap yang terdapat pada algoritma *apriori*. Setelah didapat *rule* dari tahap *rule generation*, semua *rule* yang didapat dicari nilai *jaccard similarity* pada tahap *generate candidate*. Kemudian semua *rule* yang memiliki nilai *jaccard similarity* lebih besar dari nilai *minimum jaccard similarity* diambil sebagai hasil *association rule* pada tahap *prune candidate*. Misal pada Tabel 9 ditemukan *association rule* dari hasil apriori {100} → {200} dan {200} → {100}, pada tahap ini dihitung *jaccard similarity* antara {100} dan {200} yang hasilnya adalah 100%. Pada *prune candidate* ditetapkan nilai *minimum jaccard similarity* 95%, maka {100, 200} adalah hasil *association rule* dari tahap ini.

Pada tahap pengolahan dilakukan optimasi pada *frequent itemset generation rule generation*, dan *prune candidate*. Optimasi pada *frequent itemset generation* dilakukan untuk mendapatkan itemset yang cukup banyak dan nilai *support* yang cukup tinggi. Itemset dibutuhkan jumlah yang cukup banyak untuk dapat menghasilkan *association rule* yang lebih bervariasi, sedangkan nilai *support* dibutuhkan cukup tinggi supaya nilai *confidence association rule* yang dihasilkan juga tinggi. Optimasi pada *rule generation* dilakukan untuk mendapatkan jumlah *association rule* yang cukup banyak dan nilai *confidence* yang cukup tinggi. *Association rule* dibutuhkan cukup banyak supaya analisis konsistensi dapat menemukan semakin banyak pola konsistensi, sedangkan nilai *confidence* dibutuhkan cukup tinggi untuk mengeliminasi *association rule* yang terjadi hanya karena kebetulan. Optimasi pada tahap *prune candidate* dilakukan untuk mendapatkan *association rule* yang cukup banyak dan memiliki nilai *jaccard coefficient* cukup tinggi. *Association rule* dibutuhkan cukup banyak supaya analisis konsistensi dapat menemukan semakin banyak pola konsistensi, sedangkan nilai *jaccard coefficient* dibutuhkan cukup tinggi untuk mengeliminasi *association rule* hasil *rule generation* yang terjadi hanya karena kebetulan.

3.6 Analisis Konsistensi

Pada tahap analisis konsistensi, dilakukan pencarian kesamaan *association rule* yang dihasilkan setiap bulan. Metode yang digunakan untuk mencari kesamaan adalah *jaccard similarity*. Pencarian kesamaan dilakukan untuk menarritahu seberapa konsisten kemunculan *association rule*. Seberapa konsisten *association rule* dapat terlihat dari nilai *jaccard similarity*. Jika nilai *jaccard similarity* rendah dapat dikatakan bahwa *association rule* tidak konsisten muncul. Namun, bila nilai *jaccard similarity* tinggi maka *association rule* konsisten muncul. Misal pada Bulan Januari ditemukan *association rule* $\{100\} \rightarrow \{200\}$, $\{100, 200\} \rightarrow \{300\}$. Pada Bulan Februari hanya ditemukan *association rule* $\{100\} \rightarrow \{200\}$. Maka nilai *jaccard similarity* dari *association* yang ditemukan adalah 0.33. Berdasarkan nilai *jaccard similarity* tersebut maka dapat dikatakan *association rule* yang muncul di bulan Januari tidak konsisten muncul pada bulan Februari. Analisis konsistensi dilakukan pada:

3.6.1 Bulan pada Tahun Berbeda

Setiap bulan yang sama pada tahun yang berbeda. Misal *association rule* yang muncul pada bulan Januari 2012 dibandingkan dengan Januari 2013, Januari 2014 dan Januari 2015.

3.6.2 Bulan dan Tahun Berbeda

Setiap bulan pada semua data yang muncul setelah bulan tersebut. Misal *association rule* yang muncul pada bulan Januari 2012 dibandingkan dengan Februari 2012, Maret 2012, April 2012 hingga Juli 2015.

3.7 Perbandingan

Pada tahap perbandingan, analisis konsistensi masing-masing algoritma dibandingkan untuk menarritahu kelebihan dan kekurangan masing-masing algoritma. Perbandingan dilakukan terhadap seberapa banyak dan seberapa konsisten *association rule* hasil analisis konsistensi yang didapatkan pada tahap sebelumnya.

4. HASIL

4.1 Analisis Konsistensi

Data diambil dari aplikasi UNSLA dalam bentuk data transaksional. Data transaksional ini belum dapat diolah pada tahap analisis secara optimal dikarenakan bentuknya yang masih sangat kompleks. Jika dipaksakan untuk diolah diolah pada proses analisis, proses analisis akan memakan waktu yang sangat banyak karena data transaksional dibuat optimal untuk proses *create read update delete* (CRUD) pada DBMS. Data UNSLA sendiri tersebar dalam 82 tabel. Rincian jumlah data transaksional yang digunakan pada proses ini adalah sebagai berikut.

Tabel 5. Rincian Jumlah Data Transaksional

Bulan	Tahun			
	2012	2013	2014	2015
Januari	1.653	2.348	2.438	2.766
Februari	3.201	2.837	5.838	4.387
Maret	3.923	5.498	7.748	7.837
April	3.257	5.786	7.113	7.056
Mei	3.042	5.159	5.913	5.124
Juni	2.527	4.410	4.614	1.467
Juli	1.085	1.709	1.058	
Agustus	974	1.738	1.030	
September	4.234	6.660	6.887	
Oktober	4.995	6.091	6.934	
November	4.475	5.376	5.239	
Desember	3.805	5.504	5.516	
Total	37.171	53.116	60.328	28.637
	179.252			

4.2 Preprocessing

Diperoleh data dengan jumlah pada tabel 6. Sedangkan kategori buku yang didapat ada pada tabel 7.

Tabel 6. Rincian Jumlah Data Hasil Preprocessing

Bulan	Tahun			
	2012	2013	2014	2015
Januari	259	406	426	479
Februari	534	522	1.056	822
Maret	661	985	1.396	1.356
April	581	969	1.291	1.246
Mei	510	877	1.040	966
Juni	442	777	901	320
Juli	199	313	208	
Agustus	196	363	191	
September	732	1.200	1.216	
Oktober	838	1.108	1.246	
November	803	999	1.013	
Desember	717	1.043	1.083	
Total	6.472	9.562	11.067	5.189
	32.290			

Tabel 7 Rincian Kategori Buku Hasil Preprocessing

Kode	Kategori
000	Umum
100	Filsafat
200	Agama
300	Sosial

400	Bahasa
-----	--------

Tabel 7 (Lanjutan)

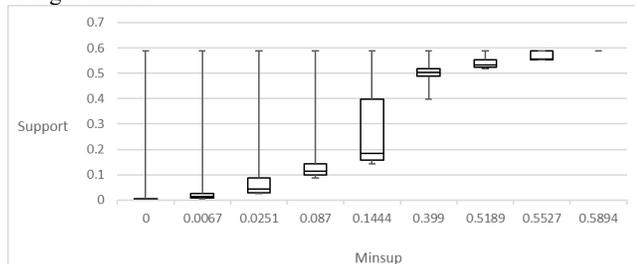
Kode	Kategori
500	Ilmu Murni
600	Ilmu Terapan
700	Seni dan Olahraga
800	Kesusastraan
900	Sejarah

4.3 Pengolahan Data Menggunakan Apriori

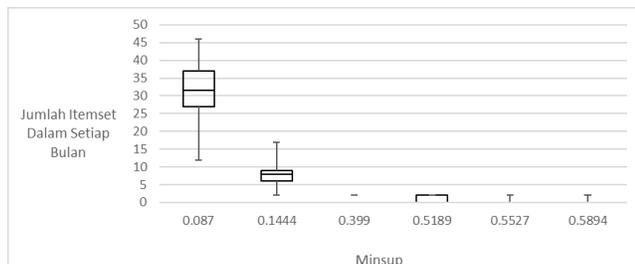
Pada tahap ini dilakukan pencarian nilai optimal *minsup* dan *minconf* dengan membandingkan efek perubahan nilai *minsup* terhadap nilai *support* dan jumlah *frequent itemset* yang dihasilkan pada *frequent itemset generation*, *minconf* terhadap nilai *confidence* dan jumlah *association rule* yang dihasilkan pada *candidate generation*. *Minsup* dan *minconf* optimal digunakan untuk menghasilkan *association rule* yang akan dicari konsistensinya pada tahapan selanjutnya.

4.3.1 Optimasi Frequent Itemset Generation

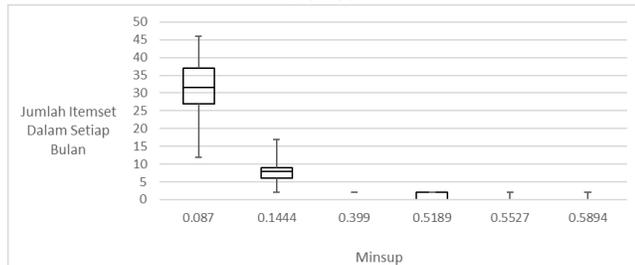
Pada tahap ini observasi dilakukan terhadap perubahan nilai *minsup* terhadap nilai *support* dan jumlah *frequent itemset* yang dihasilkan. Perubahan dilakukan dengan meningkatkan nilai *minsup* berdasarkan nilai Q_3 *support* yang dihasilkan. Nilai awal *minsup* ditentukan 0. Observasi telah dilakukan dengan hasil sebagai berikut.



Gambar 5. Efek Perubahan Nilai Minsup Terhadap Support



Gambar 6. Efek Perubahan Nilai Minsup Terhadap Jumlah Itemset



Gambar 6 (Lanjutan)

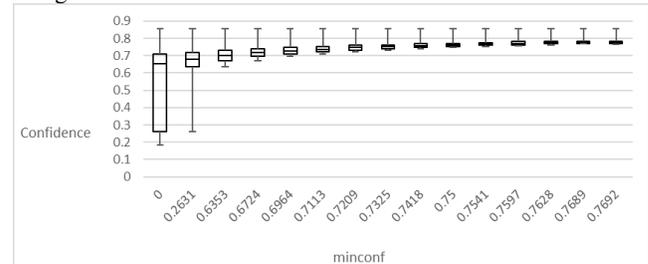
Berdasarkan Gambar 5 nilai *minsup* yang menghasilkan perubahan paling banyak ada pada 0.1444 dan 0.3990. Peningkatan nilai *minsup* dibawah 0.1444 dan diatas 0.3990

menghasilkan perubahan yang tidak berbeda jauh. *Minsup* 0.1444 menghasilkan nilai *support* yang lebih rendah dibanding 0.3990 pada kebanyakan *association rule* yang dihasilkan. Nilai Q_1 , median, dan Q_3 0.1444 lebih rendah dibanding 0.3990. Berdasarkan hal tersebut dapat dikatakan bahwa pusat data 0.1444 lebih rendah dibanding 0.3990.

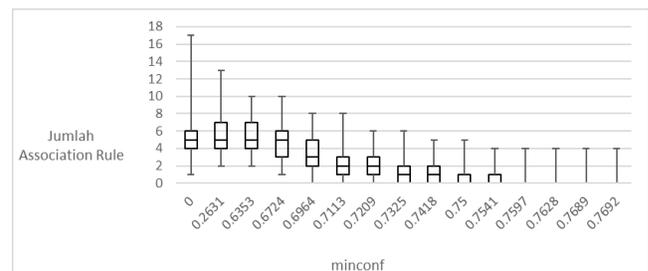
Walaupun pusat data dan mayoritas *association rule* yang dihasilkan *minsup* 0.1444 lebih rendah dibanding 0.3990, jika dilihat dari Gambar 6 jumlah *association rule* yang dapat ditemukan oleh *minsup* 0.1444 jauh lebih banyak. *Minsup* 0.3990 hanya menghasilkan 2 *association rule* pada semua observasi, sedangkan *minsup* 0.1444 dapat menghasilkan 2 sampai 16 *association rule*. Walaupun *minsup* 0.1444 menghasilkan *association rule* lebih rendah, namun nilai maksimal yang dihasilkan masih sama dengan *minsup* 0.3990. Namun jumlah *association rule* yang dihasilkan berbeda jauh antara 0.1444 dan 0.3990. Berdasarkan hal ini maka dipilih *minsup* 0.1444 sebagai nilai *minsup* optimal.

4.3.2 Optimasi Rule Generation

Pada proses sebelumnya telah didapatkan *minsup* optimal pada nilai 0.1444. Pada proses ini, dilakukan optimasi dengan menggunakan *minsup* 0.1444. Optimasi dilakukan dengan melakukan observasi efek perubahan *minconf* terhadap nilai *confidence* dan jumlah *association rule* yang dihasilkan. Perubahan dilakukan dengan meningkatkan nilai *minconf* berdasarkan nilai Q_1 *confidence* yang dihasilkan. Nilai awal *minconf* ditentukan 0. Observasi telah dilakukan dengan hasil sebagai berikut.



Gambar 7. Efek Perubahan Minconf Terhadap Nilai Confidence



Gambar 8. Efek Perubahan Nilai Minconf Terhadap Jumlah Association Rule

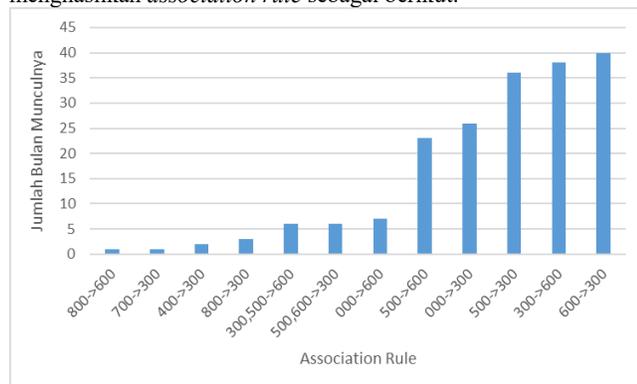
Berdasar Gambar 7, *minconf* 0.2631 dan 0.6353 menghasilkan perubahan nilai *confidence* paling banyak. Perubahan nilai *minconf* dibawah 0.2631 dan diatas 0.6353 menghasilkan nilai *confidence* yang tidak berbeda jauh. *Minconf* 0.2631 menghasilkan nilai *confidence* yang sangat menyebar, dapat dilihat dari jauhnya jarak antara Q_1 dan nilai minimum. Sedangkan *minconf* 0.6353 menghasilkan nilai *confidence* yang relatif cukup merata.

Berdasarkan Gambar 8, jumlah *association rule* mencapai puncak pada nilai *minconf* 0.2631 dan 0.6353. Kedua *minconf* tersebut menghasilkan jumlah yang mirip. Perbedaan keduanya hanya terdapat pada jumlah maksimal dengan 0.2631 memiliki nilai yang lebih tinggi. Sebelumnya telah diketahui bahwa *minconf* 0.2631 menghasilkan nilai *confidence* yang lebih tersebar, sedangkan 0.6353 lebih merata.

Berdasarkan pemaparan tersebut maka dipilih *minconf* 0.6353 sebagai nilai optimal. Walaupun 0.6353 menghasilkan jumlah yang sedikit lebih sedikit dibandingkan 0.2631, sebaran nilai *confidence* yang dihasilkan jauh lebih rata dibanding 0.2631.

4.3.3 Hasil Pengolahan Data Menggunakan *Apriori*

Nilai optimal *minsup* dan *minconf* telah diperoleh. Nilai *minsup* optimal adalah 0.1444 dan *minconf* optimal 0.6353. *Minsup* dan *minconf* tersebut diterapkan pada *apriori* sehingga menghasilkan *association rule* sebagai berikut.



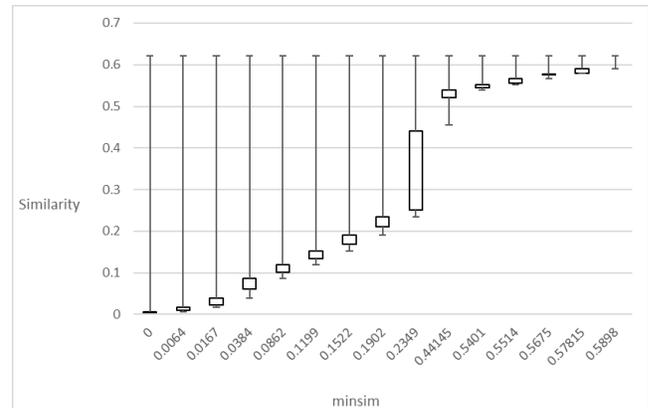
Gambar 9. *Association Rule* Hasil Pengolahan Menggunakan *Apriori*

4.4 Pengolahan Data Menggunakan *Jaccard Similarity*

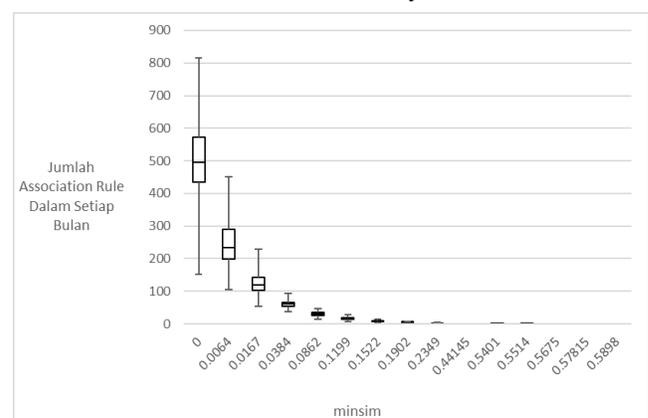
Pada tahap ini dilakukan pencarian nilai optimal *minsim* dengan membandingkan efek perubahan nilai *minsim* terhadap nilai *jaccard similarity coefficient* dan jumlah *association rule* yang dihasilkan. *Minsim* optimal digunakan untuk menghasilkan *association rule* yang akan dicari konsistensinya pada tahapan selanjutnya.

4.4.1 Optimasi *Prune Candidate*

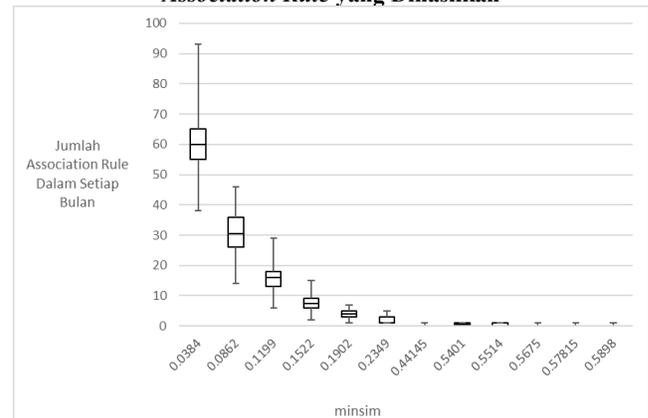
Pada tahap ini observasi dilakukan terhadap perubahan nilai *minsim* terhadap nilai *jaccard similarity* dan jumlah *association rule* yang dihasilkan. Perubahan dilakukan dengan meningkatkan nilai *minsup* berdasarkan nilai Q_1 *minsim* yang dihasilkan. Nilai awal *minsim* ditentukan 0. Observasi telah dilakukan dengan hasil sebagai berikut.



Gambar 10. Efek Perubahan Nilai *Minsim* Terhadap Nilai *Jaccard Similarity*



Gambar 11. Efek Perubahan Nilai *Minsim* Terhadap Jumlah *Association Rule* yang Dihasilkan



Gambar 11. (Lanjutan)

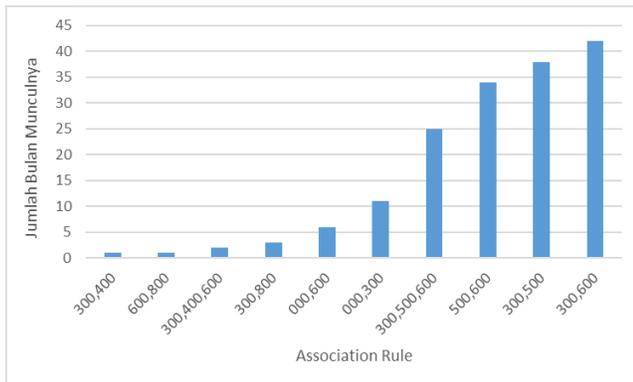
Berdasarkan Gambar 10 dapat diketahui bahwa penggunaan *minsim* selain 0.2349 dan 0.4414 menghasilkan grafik *box-plot* yang sangat cenderung kebawah. Kebanyakan hasil *association rule* berada disekitar nilai *minsim*. Penggunaan *minsim* 0.2349 menghasilkan grafik *box-plot* yang relatif lebih rendah kecenderungannya. Sedangkan penggunaan *minsim* 0.4414 menghasilkan grafik *box-plot* yang seimbang. Selain itu juga dapat diketahui bahwa nilai maksimal *jaccard similarity coefficient* yang dihasilkan oleh penggunaan *minsim* sama.

Berdasarkan Gambar 11 dapat diketahui bahwa penggunaan *minsim* dari 0 sampai 0.1902 menghasilkan *box-plot*

yang seimbang. Penggunaan *minsim* dari 0.2349 hingga 0.5898 menghasilkan *box-plot* yang cenderung kebawah. Dapat dikatakan juga penggunaan *minsim* 0.2349 hingga 0.5898 menghasilkan jumlah *association rule* sedikit. Berdasarkan gambar 14 telah diketahui bahwa nilai *jaccard similarity coefficient* penggunaan *minsim* 0 sampai 0.1902 berada sedikit diatas nilai *minsim*. Maka nilai optimal yang diambil adalah 0.1902 karena memiliki nilai *jaccard coefficient* lebih tinggi dari yang lain dan jumlah rule yang cukup merata.

4.4.2 Hasil Pengolahan Data Menggunakan *Jaccard Similarity*

Nilai optimal *minsim* telah diperoleh. Nilai *minsim* optimal adalah 0.1902. *Minsim* tersebut diterapkan pada *jaccard similarity* sehingga menghasilkan *association rule* sebagai berikut.



Gambar 12. *Association Rule* Hasil Pengolahan Menggunakan *Jaccard Similarity*

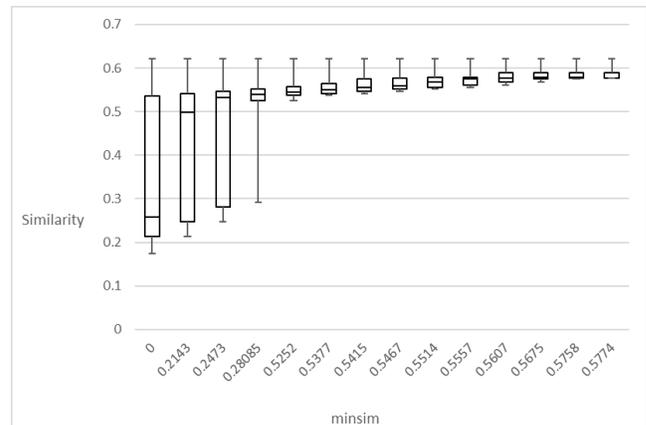
4.5 Pengolahan Hasil *Apriori* Menggunakan *Jaccard Similarity*

Pada tahap ini dilakukan kombinasi *apriori* dan *jaccard*. Metode kombinasi yang dilakukan adalah dengan memproses *association rule* hasil *apriori* menggunakan *jaccard* untuk mendapatkan *association rule* yang memiliki *interestingness* lebih dari sekedar *support* dan *confidence*.

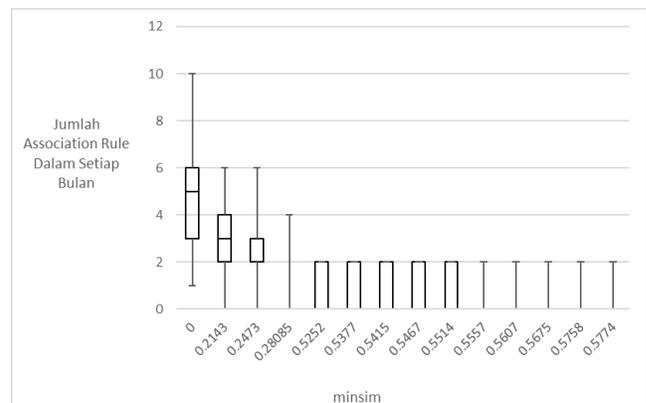
Pada tahap sebelumnya telah didapatkan *apriori* optimal pada nilai *minsup* 0.1444 dan *minconf* 0.6353. Pada tahap ini hasil dari *apriori* tersebut diolah kembali menggunakan *jaccard similarity* untuk mendapatkan *association rule*. Maka perlu dicari berapa nilai *minsim* yang diperlukan untuk menghasilkan *association rule* paling optimal. *Association rule* optimal yang dihasilkan dari proses ini akan dicari konsistensinya pada tahap selanjutnya.

4.5.1 Optimasi *Prune Candidate*

Pada tahap ini observasi dilakukan terhadap perubahan nilai *minsim* terhadap nilai *jaccard similarity* dan jumlah *association rule* yang dihasilkan. Perubahan dilakukan dengan meningkatkan nilai *minsim* berdasarkan nilai Q_1 *jaccard similarity* yang dihasilkan. Nilai awal *minsim* ditentukan 0. Observasi telah dilakukan dengan hasil sebagai berikut.



Gambar 13. Efek Perubahan Nilai *Minsim* Terhadap Nilai *Jaccard Similarity Coefficient* Pada Metode Kombinasi

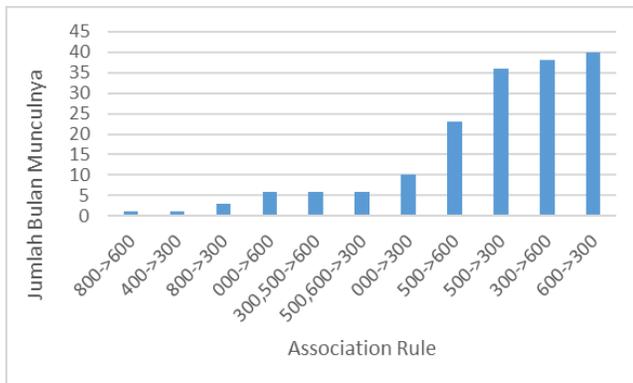


Gambar 14. Efek Perubahan Nilai *Minsim* Terhadap Jumlah *Association Rule* yang Dihasilkan Pada Metode Kombinasi

Berdasarkan Gambar 13 diketahui bahwa nilai peningkatan *minsim* mengakibatkan nilai median *jaccard similarity* meningkat mendekati 0.6. Sementara itu, berdasarkan Gambar 14 diketahui peningkatan nilai *minsim* mengakibatkan jumlah *association rule* yang dihasilkan semakin sedikit dengan nilai tengah mendekati 0. Jumlah *association rule* yang dihasilkan oleh *minsim* 0, 0.2143, dan 0.2473 termasuk yang paling tinggi. Pada penelitian ini digunakan *minsim* 0.2143 sebagai nilai optimal karena berdasarkan jumlah *association rule* lebih baik dibanding 0.2473 dan berdasarkan nilai *jaccard similarity* yang dihasilkan lebih tinggi dibanding *minsim* 0.

4.5.2 Hasil Kombinasi

Nilai optimal *minsim* telah diperoleh. Nilai *minsim* optimal adalah 0.2132. *Minsim* tersebut diterapkan pada *jaccard similarity* sehingga menghasilkan *association rule* sebagai berikut.



Gambar 15. Association Rule Hasil Pengolahan Menggunakan Kombinasi Apriori dan Jaccard Similarity

4.6 Analisis Konsistensi

Pada tahap ini dilakukan pengamatan terhadap konsistensi *association rule* yang ditemukan menggunakan *apriori*, *jaccard similarity* dan kombinasi keduanya. Pengamatan dilakukan dengan menggunakan dua pendekatan, yaitu: 1.) per bulan pada tahun yang berbeda, 2.) per bulan dan tahun.

4.6.1 Analisis Konsistensi Per Bulan pada Tahun Berbeda

Pada tahap ini dilakukan penghitungan konsistensi *association rule* yang dihasilkan setiap tahun pada masing-masing bulan. Digunakan batas bawah 0.7 sebagai nilai minimal pada bulan tertentu metode tertentu menghasilkan *association rule* konsisten. Penghitungan telah dilakukan, didapatkan nilai konsistensi sebagai berikut.

Tabel 7. Konsistensi Association Rule Per Bulan Dalam Tahun Berbeda

Bulan	Nilai Konsistensi Pada Algoritma		
	Apriori	Jaccard Similarity	Kombinasi
Maret	0.9333	0.8667	0.8
April	0.75	0.7233	0.8
Juli	-	0.75	-
September	0.9	-	0.8
Oktober	-	0.8	-
November	0.7	0.75	0.875
Desember	0.75	0.775	0.8
Rata-Rata	0.8066	0.7775	0.815

4.6.2 Analisis Konsistensi Per Bulan dan Tahun Berbeda

Pada tahap ini dilakukan penghitungan konsistensi *association rule* yang dihasilkan setiap bulan dan tahun. Penghitungan telah dilakukan, didapatkan nilai konsistensi sebagaimana terdapat pada Lampiran 1.

4.7 Perbandingan

4.7.1 Perbandingan Konsistensi Association Rule Per Bulan Dalam Tahun Berbeda

Berdasarkan Tabel 7 didapati bahwa *apriori* menghasilkan nilai konsistensi rata-rata 0.8066, *jaccard similarity* menghasilkan 0.7775, dan kombinasi keduanya menghasilkan 0.815. Berdasarkan ini maka dapat dikatakan bahwa secara keseluruhan kombinasi *apriori* dan *jaccard similarity* mampu menghasilkan

association rule yang lebih konsisten dibandingkan *apriori* dan *jaccard similarity*.

Berdasarkan Tabel 7 didapati bahwa jumlah bulan yang menghasilkan *association rule* konsisten menggunakan *apriori* sama dengan jumlah bulan konsisten kombinasi. Maka dapat dikatakan bahwa penambahan *jaccard similarity* pada *apriori* tidak mereduksi jumlah bulan konsisten pada *apriori*.

Berdasarkan Tabel 7 didapati bahwa jumlah bulan yang menghasilkan *association rule* konsisten menggunakan *jaccard similarity* lebih banyak dibandingkan jumlah bulan konsisten kombinasi. Maka dapat dikatakan bahwa penambahan *apriori* pada *jaccard similarity* mereduksi jumlah bulan konsisten pada *jaccard similarity*.

4.7.2 Perbandingan Konsistensi Association Rule Per Bulan dan Tahun Berbeda

Berdasarkan Lampiran 1 dapat diketahui bahwa *jaccard similarity* lebih bervariasi menghasilkan *association rule* dibanding *apriori* dan kombinasinya. *Jaccard similarity* mampu menghasilkan 9 variasi *association rule*, sedangkan *apriori* dan kombinasinya hanya menghasilkan 4 variasi dan 3 variasi.

Berdasarkan Lampiran 1 dapat diketahui bahwa dalam setiap bulan *apriori* selalu menghasilkan *association rule* konsisten lebih banyak atau sama dengan kombinasi *apriori* dan *jaccard*. Maka dapat dikatakan bahwa penggunaan *jaccard similarity* pada *apriori* menyebabkan *association rule* yang dihasilkan *apriori* menjadi tereduksi.

Berdasarkan Lampiran 1 dapat diketahui bahwa *jaccard similarity* selalu menghasilkan *association rule* lebih banyak atau sama dengan kombinasi *apriori* dan *jaccard*. Maka dapat dikatakan bahwa penggunaan *apriori* pada *jaccard similarity* menyebabkan *association rule* yang dihasilkan *jaccard similarity* menjadi tereduksi.

Berdasarkan Lampiran 1 dapat diketahui bahwa secara keseluruhan, baik menggunakan *apriori*, *jaccard similarity*, maupun kombinasi keduanya, pada Bulan Maret dan April pola peminjaman yang dapat ditemukan relatif lebih banyak dibandingkan dengan bulan-bulan lain. Secara keseluruhan, bulan tersebut merupakan bulan paling konsisten.

Berdasarkan Lampiran 1 dapat diketahui bahwa pada Bulan Agustus, jumlah *association rule* konsisten jauh lebih sedikit jika dibandingkan dengan bulan lain. Maka dapat dikatakan bahwa Bulan Agustus adalah bulan paling tidak konsisten.

Berdasarkan Lampiran 1 dapat diketahui bahwa *association rule* paling konsisten adalah $\{300\} \rightarrow \{600\}$. *Association rule* tersebut selalu ditemukan pada semua bulan dan menggunakan semua algoritma kecuali pada Bulan Juli dan Agustus.

5. PENUTUP

5.1 Kesimpulan

Percobaan penerapan *jaccard similarity* sebagai kombinasi *factor of interest* algoritma *apriori* telah dilakukan. Pada bab ini dipresentasikan rangkuman permasalahan dan hasil yang telah diperoleh. Diberikan *association rule* $A \rightarrow B$ dengan A dan B merupakan kombinasi *itemset* data transaksi $I = \{000, 100, 200, \dots, 900\}$ yang direpresentasikan dengan $m \times n$ matrix biner. Tujuan dari percobaan ini adalah menemukan *association rule* $A \rightarrow B$ yang memenuhi *minsup*, *minconf*, dan *minsim* pada setiap data transaksi perbulan yang dimiliki.

Permasalahan dalam penelitian ini dapat terselesaikan dengan mengamati konsistensi *association rule* yang dihasilkan. Dalam penelitian ini kami telah memberikan hasil analisis konsistensi *association rule* menggunakan tiga metode yang berbeda. Semua metode yang telah dilakukan menghasilkan set *association rule* masing-masing. Kami melakukan perbandingan *association rule* yang dihasilkan pada setiap percobaan untuk mendapatkan konsistensi.

Association rule mining menggunakan kombinasi *apriori* dan *jaccard similarity* menghasilkan *association rule* yang lebih konsisten jika dibandingkan dengan *apriori* dan *jaccard similarity*. Penerapan *jaccard similarity* pada *apriori* terbukti mereduksi beberapa *association rule* yang dihasilkan *apriori* dan *jaccard similarity* sehingga hasil konsistensi yang diperoleh lebih baik.

Namun ada beberapa bulan dimana penggunaan kombinasi tidak dapat menemukan *association rule* lebih baik dibanding *jaccard similarity* yaitu pada Bulan Juli dan Agustus. Pada kedua bulan tersebut, dari ketiga metode yang diamati hanya *jaccard similarity* saja yang dapat menemukan *association rule*.

5.2 Saran

Penelitian ini hanya bertujuan untuk mencari metode *association rule* yang lebih baik dari aspek konsistensi *association rule* yang dihasilkan. Sedangkan dalam aplikasinya, salah satu faktor yang seharusnya perlu diperhatikan adalah performa suatu metode dalam menemukan *association rule*. Berdasarkan ini maka perlu ada penelitian lebih lanjut untuk membandingkan performa antar metode.

6. REFERENCES

- [1] Universitas Sebelas Maret. Unit Pelayanan Teknis (UPT) Perpustakaan, Profil UPT Perpustakaan. <http://pustaka.uns.ac.id/?opt=1001&menu=profile>. Diakses pada tanggal 10 April 2015.
- [2] Universitas Sebelas Maret (2015). Perpustakaan Pusat, UNS Library Automation (UNSLA): Backup Database, 2006-2014 [computer file]. Surakarta, Jawa Tengah: *ICT Perpustakaan Pusat*, Maret 2015.
- [3] Reference and User Services Association (RUSA) Access to Information Committee, 2000. Information Services for Information Consumers: Guidelines for Providers. Prepared by the Standards and Guidelines Committee, Reference and User Services Association. Approved by the RUSA Board of Directors: <http://www.ala.org/rusa/resources/guidelines/guidelinesinformation>.
- [4] Tan, P.N., Steinbach, M. and Kumar, V., 2006. Introduction to data mining (Vol. 1). Boston: Pearson Addison Wesley.
- [5] Agrawal, R., Imieliński, T. and Swami, A., 1993. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2), pp.207-216.
- [6] Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J.D. and Yang, C., 2001. Finding interesting associations without support pruning. *Knowledge and Data Engineering, IEEE Transactions on*, 13(1), pp.64-78.
- [7] Elmasri, R., 2008. *Fundamentals of database systems*. Pearson Education India.
- [8] MySQL, A.B., *Mysql 5.7 reference manual*, 2016. <http://dev.mysql.com/doc>. Diakses pada tanggal 15 Juni 2016.
- [9] Han, J., Kamber, M. and Pei, J., 2011. *Data mining: concepts and techniques*. Elsevier.

