

IMPLEMENTASI ALGORITMA APRIORI UNTUK MARKET BASKET ANALYSIS BERBASIS R

Thomas Brian¹⁾ dan Ardhi Sanwidi²⁾

¹⁾ Universitas Trunojoyo Madura

²⁾ Universitas Nahdlatul Ulama Blitar

e-mail: thomasbrian@trunojoyo.ac.id¹⁾, ardhisanwidi@unublitar.ac.id²⁾

ABSTRACT

More sales transactions hence required a system to generate important information. This innovation will solve many problems in the field of sales marketing and inventory, because the products that are not so salable if paired properly will increase the value of sales. However, looking for associations requires a complicated process because of the problem of large product combinations let alone if the retail business has thousands of products. Apriori is a data mining algorithm to find relationships between items on market basket analysis. By finding the pattern of sales transactions is expected to increase business value. In the process undertaken in this research is implemented using R with a priori function to process data. It starts from reading the dataset until it finds a recommendation from the system that has been created using the function in R. Determine the support, confidence and lift values to find the best itemset for the next sale. Trial that has been done with transaction dataset shows best result on value filter support = 0,1 confidence = 0,8 and lift > 1.

Keywords: Apriori, Data Mining, Market Basket Analysis, R.

ABSTRAK

Semakin banyak transaksi penjualan maka diperlukan suatu sistem untuk menghasilkan informasi yang penting. Inovasi ini akan banyak memecahkan masalah di bidang sales marketing dan inventory, karena produk yang tidak begitu laku jika dipasangkan dengan tepat akan naik nilai penjualannya. Namun mencari asosiasi membutuhkan proses yang rumit karena masalah kombinasi produk yang besar apalagi jika bisnis ritel tersebut memiliki ribuan produk. Apriori adalah algoritma data mining untuk mencari hubungan antar item pada market basket analysis. Dengan menemukan pola transaksi penjualan maka diharapkan nilai bisnis meningkat. Pada proses yang dilakukan pada penelitian ini diimplementasikan menggunakan R dengan function apriori untuk mengolah data. Dimulai dari pembacaan dataset sampai dengan menemukan rekomendasi dari sistem yang sudah dibuat menggunakan function di R. Menentukan nilai support, confidence dan lift berpengaruh untuk menemukan itemset terbaik untuk penjualan selanjutnya. Uji coba yang sudah dilakukan dengan dataset transaksi menunjukkan hasil terbaik pada filter nilai support = 0,1 confidence = 0,8 dan lift > 1.

Kata Kunci: Apriori, Data Mining, Market Basket Analysis, R.

I. PENDAHULUAN

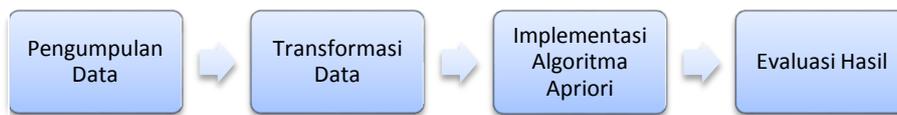
BISNIS yang semakin berkembang bisa dipengaruhi banyak faktor seperti inovasi pada produk penjualan. Salah satu inovasi di bidang ritel adalah mencari asosiasi atau hubungan antar item produk sehingga dapat dipaketkan dan dijual bersamaan, memberikan rekomendasi produk kepada seseorang berdasarkan riwayat historis pembelian produk, menyusun rak *display*, dan menyusun halaman produk *e-commerce*. Inovasi ini akan banyak memecahkan masalah di bidang *sales marketing* dan *inventory*, karena produk yang tidak begitu laku jika dipasangkan dengan tepat akan naik nilai penjualannya. Namun mencari asosiasi membutuhkan proses yang rumit karena masalah kombinasi produk yang besar apalagi jika bisnis ritel tersebut memiliki ribuan produk. Diperlukan algoritma yang mumpuni, dan di dunia *data science* ini dikenali dengan tipe *association rules*, dan salah satu yang sangat populer adalah apriori.

Algoritma apriori termasuk jenis aturan asosiasi pada *data mining*, algoritma apriori yang bertujuan untuk menemukan *frequent item sets* dijalankan pada sekumpulan data. Analisis apriori didefinisikan suatu proses untuk menemukan semua aturan apriori yang memenuhi syarat minimum untuk *support*

dan *confidence* [1]. Algoritma *association rules* adalah algoritma yang fokus mengidentifikasi hubungan yang kuat antar item-item di dalam suatu populasi dataset. Ini sangat penting karena relasi antar item itu biasanya merupakan masalah kombinasi yang bisa berkembang sangat besar, jauh melampaui jumlah item asli. Item adalah produk tunggal yang menjadi bagian dari transaksi, sedangkan Itemset adalah kombinasi produk yang terdapat pada suatu transaksi. Penelitian sebelumnya yang dilakukan oleh Sandi F. R. tentang algoritma apriori untuk analisis keranjang belanja pada data transaksi penjualan menjelaskan pola analisis *association rules* yang ditemukan dapat digunakan sebagai penunjang keputusan manajer dalam mengelola aktifitas perusahaannya [2]. Hasil penerapan *association rules* oleh Winda A. dengan algoritma apriori menggunakan minimum *support* 30% dan minimum *confidence* 80% menghasilkan 4614 *rules* hubungan antar indikator pada dataset kemiskinan Kabupaten Tanah Laut [3]. Penggunaan variabel fuzzy sebelum penerapan algoritma apriori diterapkan pada dataset harga saham untuk mengetahui pola pergerakan harga saham antar perusahaan [4]. Dengan demikian penelitian ini akan mencoba mengimplementasikan algoritma apriori untuk *market basket analysis*. Sedangkan dataset yang digunakan adalah contoh data penjualan yang diperoleh dari http://academy.dqlab.id/dataset/data_transaksi.txt. Tujuan dari penelitian ini difokuskan pada *market basket analysis* untuk menemukan pola unik dari sebuah dataset transaksi.

II. METODE PENELITIAN

Pada penelitian ini dilakukan empat tahapan langkah diantaranya: pengumpulan data, transformasi data, implementasi algoritma apriori dan evaluasi hasilnya sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Langkah awal dari data yang dikumpulkan adalah pengumpulan data transaksi dan barang-barang yang dibeli pada setiap transaksi tersebut. Langkah kedua dilakukan transformasi data untuk perubahan data transaksi yang telah terkumpul menjadi format data standar untuk proses pembacaan data dengan algoritma apriori. Pada Tabel 1, dapat dilihat pola hasil pembacaan transaksi penjualan. “Kode Transaksi” adalah nomor struk dan “Item” adalah item yang dibeli.

TABEL I
TRANSAKSI PENJUALAN

Kode Transaksi	Item
#01	Teh Celup
#01	Sirup
#01	Pet Food
#02	Teh Celup
#02	Gula
#03	Sirup
#03	Pet Food
#04	Teh Celup
#04	Sirup
#05	Teh Celup
#05	Sirup
#05	Gula
#06	Teh Celup
#06	Pet Food
#07	Teh Celup
#07	Sirup
#07	Pet Food
#08	Teh Celup
#09	Teh Celup
#09	Gula
#10	Sirup
#10	Pet Food
#10	Gula

Transaksi penjualan tersebut disimpan dalam format *text file* (.txt) yang selanjutnya diubah menjadi *data.frame* dengan perintah *read.csv* di R. Hasil dari pengubahan ini diberikan pada Gambar 2.

```
> transaksi_tabular<-read.csv("http://academy.dqlab.id/dataset/data_transaksi.txt",
,sep="\t")
> transaksi_tabular
  Kode.Transaksi  Item
1             #01 Teh Celup
2             #01  Sirup
3             #01 Pet Food
4             #02 Teh Celup
5             #02  Gula
6             #03  Sirup
7             #03 Pet Food
8             #04 Teh Celup
9             #04  Sirup
10            #05 Teh Celup
11            #05  Sirup
12            #05  Gula
13            #06 Teh Celup
14            #06 Pet Food
15            #07 Teh Celup
16            #07  Sirup
17            #07 Pet Food
18            #08 Teh Celup
19            #09 Teh Celup
20            #09  Gula
21            #10  Sirup
22            #10 Pet Food
23            #10  Gula
>
```

Gambar 2. Membaca File Sebagai Data Frame

Untuk mempermudah dalam pembacaan data transaksi di *package R* digunakan fungsi *read.transactions* yang diperlihatkan pada Gambar 3.

```
> library(arules)
> transaksi<-read.transactions(file="http://academy.dqlab.id/dataset/data_transaksi.txt",
format="single", sep="\t", cols=c(1,2), skip=1)
> transaksi
transactions in sparse format with
10 transactions (rows) and
4 items (columns)
```

Gambar 3. Fungsi Read Transactions

Item-item yang terdapat pada objek *transactions* dapat dilihat dengan aksesor *@itemInfo* seperti pada Gambar 4 dan daftar transaksi menggunakan *@itemsetInfo* seperti pada Gambar 5.

```
> library(arules)
> transaksi <- read.transactions(file="http://academy.dqlab.id/dataset/data_transaksi.txt",
format="single", sep="\t", cols=c(1,2), skip=1)
> transaksi@itemInfo
  labels
1     Gula
2 Pet Food
3     Sirup
4 Teh Celup
>
```

Gambar 4. Daftar Item dengan *@itemInfo*

```
> library(arules)
> transaksi <- read.transactions(file="http://academy.dqlab.id/dataset/data_transaksi.txt", format="single", sep="\t", cols=c(1,2), skip=1)
> transaksi@itemsetInfo
transactionID
1            #01
2            #02
3            #03
4            #04
5            #05
6            #06
7            #07
8            #08
9            #09
10           #10
```

Gambar 5. Daftar Transaksi dengan @itemSetInfo

Langkah ketiga, dilakukan implementasi algoritma apriori dengan menggunakan bahasa pemrograman R. Beberapa tahapan dalam penerapan *association rules* dengan algoritma apriori ini adalah dengan:

1. Menentukan nilai *minimum support* dan *minimum confidence*, dan
2. Menentukan nilai *lift* terbaik untuk menemukan kombinasi itemset frekuensi tinggi.

Langkah terakhir program yang telah dikembangkan adalah mengolah data yang sudah ada. Pada tahap akhir ini dilakukan evaluasi hasil pengolahan data dengan program yang telah dikembangkan.

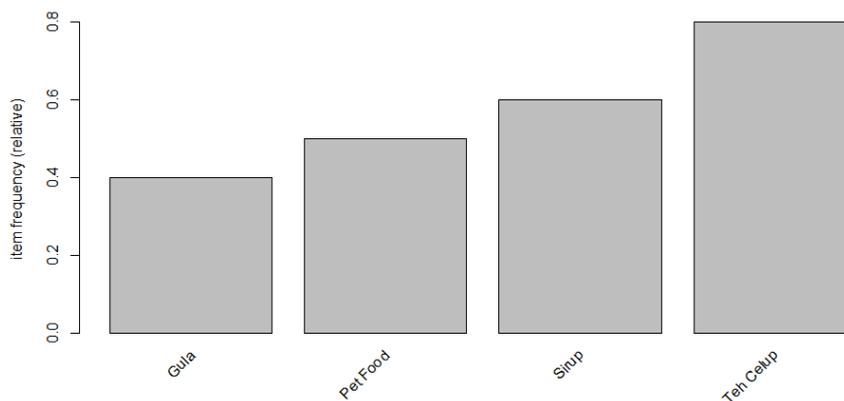
III. HASIL DAN PEMBAHASAN

Dataset yang digunakan berformat *.txt* yang kemudian dibaca menggunakan RStudio. Dari hasil yang diperoleh, terdapat 10 transaksi dengan 4 item (Gula, Pet Food, Sirup dan Teh Celup). Apabila dijalankan dengan perintah *read.transaction* dan dibentuk menjadi sebuah matriks pada R maka diperoleh sebagaimana ditunjukkan pada Gambar 6.

```
> library(arules)
> transaksi <- read.transactions(file="http://academy.dqlab.id/dataset/data_transaksi.txt", format="single", sep="\t", cols=c(1,2), skip=1)
> transaksi@data
4 x 10 sparse Matrix of class "ngCMatrix"

[1,] . | . | . | . | . | . | . | . | . | . |
[2,] | . | . | . | . | . | . | . | . | . |
[3,] | . | | | | . | . | . | . | . |
[4,] | | . | | | | | | | . | . |
>
```

Gambar 6. Pembacaan Dataset di R



Gambar 7. Grafik Item Frequency

Setelah dilakukan pembacaan dataset transaksi, selanjutnya menganalisa pola kombinasi dari itemset beserta detail dari *association rules* menggunakan *function apriori*. Makna “4 x 10” berarti bahwa terdapat 4 item dan 10 transaksi. Selanjutnya simbol [1,] sampai dengan [4,] adalah baris item. Rangkaian simbol tanda titik (“.”) dan garis lurus (“|”) menandakan bahwa item tidak ada (tanda titik) dan item ada (garis lurus) untuk posisi transaksi ke-n. Selain dengan visualisasi matrik sederhana seperti pada Gambar 6, cara lain yang dapat dilakukan untuk melihat distribusi transaksi dari tiap item adalah dalam bentuk grafik, yakni dengan menggunakan *function itemFrequencyPlot* dimana hasilnya dapat ditunjukkan pada Gambar 7.

Hasil lebih detail dapat ditunjukkan seperti Gambar 8 dimana penjelasan masing-masing simbol yang digunakan adalah sebagai berikut: *lhs* adalah itemset di bagian kiri (*left hand side*); *rhs* adalah itemset di bagian kanan (*right hand side*); *support*, *confidence* dan *lift* adalah berbagai metrik yang digunakan untuk menilai kualitas dari *rule* yang umumnya semakin besar semakin baik. Sedangkan *count* adalah jumlah kemunculan *itemset* gabungan *lhs* dan *rhs* pada dataset transaksi.

	lhs	rhs	support	confidence	lift	count
[1]	{}	=> {Teh Celup}	0.8	0.8	1.000000	8
[2]	{Pet Food}	=> {Sirup}	0.4	0.8	1.333333	4
[3]	{Gula,Pet Food}	=> {Sirup}	0.1	1.0	1.666667	1

Gambar 8. Association Rules dengan Function Apriori Tanpa Parameter

Pada Gambar 8, terlihat *rule* kedua dengan {Pet Food} => {Sirup} adalah cukup menarik. Apabila *rule* ini diaplikasikan, maka ada beberapa opsi (pilihan) yang bisa dilakukan, diantaranya yaitu:

- merekomendasikan setiap yang membeli Pet Food dengan Sirup
- rak *display* Pet Food dan Sirup bisa didekatkan
- Pet Food dan Sirup dipaketkan bersama dan dijual dengan harga khusus.

Dari hasil yang ditunjukkan pada gambar 8 dengan menggunakan *function apriori* secara *default* tanpa parameter apapun, dihasilkan tiga *rules* yang merupakan nilai dari minimum *support* 0,1 dan minimum *confidence* 0,8 sebagai *filter* dari *function apriori*. Jika parameter minimum *support* dan *confidence* masing-masing bernilai 0,1 dan 0,5 maka dihasilkan enam belas (16) *rules*. Perintah berikut merupakan perintah yang ditulis pada program R:

```
apriori(transaksi,parameter = list(supp = 0.1, confidence = 0.5))
```

Hasil dari perintah ini dapat dilihat pada Gambar 9.

	lhs	rhs	support	confidence	lift
[1]	{}	=> {Pet Food}	0.5	0.500000	1.000000
[2]	{}	=> {Sirup}	0.6	0.600000	1.000000
[3]	{}	=> {Teh Celup}	0.8	0.800000	1.000000
[4]	{Gula}	=> {Sirup}	0.2	0.500000	0.833333
[5]	{Gula}	=> {Teh Celup}	0.3	0.750000	0.937500
[6]	{Pet Food}	=> {Sirup}	0.4	0.800000	1.333333
[7]	{Sirup}	=> {Pet Food}	0.4	0.666667	1.333333
[8]	{Pet Food}	=> {Teh Celup}	0.3	0.600000	0.750000
[9]	{Sirup}	=> {Teh Celup}	0.4	0.666667	0.833333
[10]	{Teh Celup}	=> {Sirup}	0.4	0.500000	0.833333
[11]	{Gula,Pet Food}	=> {Sirup}	0.1	1.000000	1.666667
[12]	{Gula,Sirup}	=> {Pet Food}	0.1	0.500000	1.000000
[13]	{Gula,Sirup}	=> {Teh Celup}	0.1	0.500000	0.625000
[14]	{Pet Food,Sirup}	=> {Teh Celup}	0.2	0.500000	0.625000
[15]	{Pet Food,Teh Celup}	=> {Sirup}	0.2	0.666667	1.111111
[16]	{Sirup,Teh Celup}	=> {Pet Food}	0.2	0.500000	1.000000

Gambar 9. Association Rules dengan Function Apriori Parameter Support = 0,1 dan Confidence = 0,5

Pada Gambar 9, terdapat 16 jumlah *rules* yang dihasilkan dimana jumlah ini jauh lebih banyak dibandingkan tanpa menggunakan parameter yang hanya menghasilkan 3 *rules* (Gambar 8), sehingga terdapat lebih banyak pilihan untuk melakukan *filter lhs* dan *rhs*. Untuk menampilkan *filter lhs* dan *rhs* yang memiliki item Teh Celup menggunakan perintah berikut:

```
subset(mba, lhs %in% "Teh Celup" | rhs %in% "Teh Celup")
```

Hasil dari perintah ini diberikan pada Gambar 10.

lhs	rhs	support	confidence	lift
[1] {}	=> {Teh Celup}	0.8	0.8000000	1.0000000
[2] {Gula}	=> {Teh Celup}	0.3	0.7500000	0.9375000
[3] {Pet Food}	=> {Teh Celup}	0.3	0.6000000	0.7500000
[4] {Sirup}	=> {Teh Celup}	0.4	0.6666667	0.8333333
[5] {Teh Celup}	=> {Sirup}	0.4	0.5000000	0.8333333
[6] {Gula,Sirup}	=> {Teh Celup}	0.1	0.5000000	0.6250000
[7] {Pet Food,Sirup}	=> {Teh Celup}	0.2	0.5000000	0.6250000
[8] {Pet Food,Teh Celup}	=> {Sirup}	0.2	0.6666667	1.1111111
[9] {Sirup,Teh Celup}	=> {Pet Food}	0.2	0.5000000	1.0000000

Gambar 10. *Filter lhs* dan *rhs* yang Memiliki Item Teh Celup

Pada Gambar 10 terdapat sembilan (9) *rules* sebagai hasil *filter* dan jika digunakan informasi *support*, *confidence* dan *lift* maka kandidat yang bagus terdapat pada *rule* ke-8 dimana nilai *lift* lebih dari 1. Pada *rule* ini terdapat ({Pet Food, Teh Celup} => {Sirup}). Untuk melakukan *filter* terhadap metrik kualitas dari *association rules*: *support*, *confidence* dan *lift* adalah dengan menggunakan *function subset*. *Function* yang sama juga digunakan untuk melakukan *filter* terhadap *lhs* dan *rhs*. Perbedaannya adalah karena nilainya berupa angka, maka untuk ketiga metrik tersebut digunakan operator perbandingan angka. Sebagai contoh, untuk melakukan *filter* dengan kondisi berikut:

- *lhs* atau *rhs* memiliki Teh Celup
- *lift* di atas 1

maka perintahnya adalah sebagai berikut:

```
subset(mba, (lhs %in% "Teh Celup" | rhs %in% "Teh Celup") & lift>1)
```

Hanya terdapat satu (1) *rule* sebagai hasil *filter* dan paket produk ini cukup menjanjikan, dimana Teh Celup menjadi komponen dari *itemset* di *lhs*. Selengkapnya hasil *filter* ini dapat ditunjukkan pada Gambar 11.

```
> library(arules)
> transaksi <- read.transactions(file="http://academy.dqlab.id/dataset/data_transaksi.txt", format="single", sep="\t", cols=c(1,2), skip=1)
> mba <- apriori(transaksi,parameter = list(supp = 0.1, confidence = 0.5))
Apriori

Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen
 0.5      0.1      1 none FALSE          TRUE         5      0.1      1
maxlen target  ext
 10      rules FALSE

Algorithmic control:
filter tree heap memopt load sort verbose
 0.1 TRUE TRUE  FALSE TRUE  2    TRUE

Absolute minimum support count: 1

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[4 item(s), 10 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [16 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> inspect(subset(mba, (lhs %in% "Teh Celup" | rhs %in% "Teh Celup") & lift>1))
lhs      rhs      support confidence lift
[1] {Pet Food,Teh Celup} => {Sirup} 0.2      0.6666667 1.1111111
```

Gambar 11. *Filter lhs* dan *rhs* dengan Item Teh Celup dan Lift > 1

Operator `%in%` yang sebelumnya telah digunakan sudah efektif. Namun operator ini tidak cocok jika ingin melakukan *filter* itemset dengan logika *AND*. Artinya seluruh item harus muncul dalam itemset yang terpilih. Untuk keperluan tersebut, digunakan operator `%ain%` sebagaimana berikut:

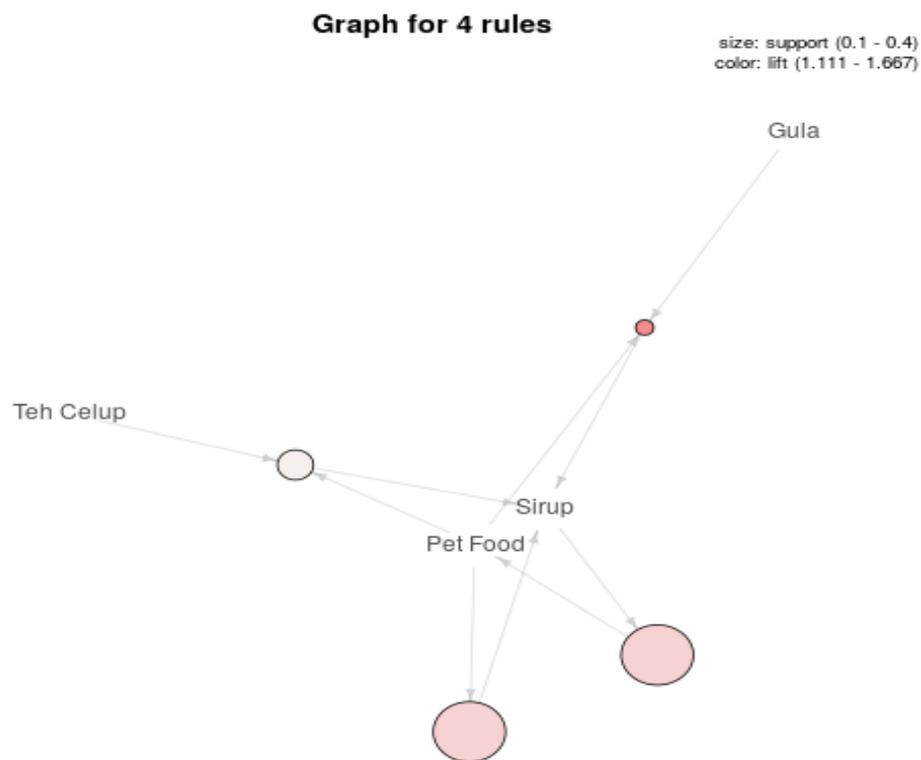
```
subset(mba, (lhs %ain% c("Pet Food", "Gula" )))
```

Ekspresi ini menjadi kunci untuk membuat rekomendasi. Hasil dari perintah ini ditunjukkan pada Gambar 12.

lhs	rhs	support	confidence	lift	count
[1] {Gula,Pet Food} =>	{Sirup}	0.1	1	1.666667	1

Gambar 12. *Filter lhs dan rhs dengan Item Teh Celup dan Lift > 1 Menggunakan Operator %ain%*

Ketika item semakin banyak, maka diperlukan visualisasi untuk membantu analisa. Jenis visualisasi cukup banyak, contohnya *graph* yang terdapat pada *package arulesViz*, yaitu suatu *package* dengan visualisasi khusus untuk *association rules*. Berikut adalah visualisasi yang dihasilkan dari *rules* yang telah dilakukan *filter* dengan *lift* di atas nilai 1.1 yang diperlihatkan pada Gambar 13.



Gambar 13. Visualisasi Rules dengan Graph

IV. KESIMPULAN

Setelah melalui tahapan pembacaan dataset transaksi, implementasi dan beragam pengujian, dapat diambil beberapa kesimpulan sebagai berikut:

1. Dari hasil pengujian, sistem telah terbukti berhasil mengimplementasikan algoritma apriori untuk *MBA (Market Basket Analysis)*,
2. Hanya terdapat satu (1) *rule* sebagai hasil *filter*, dan dengan *lift* di atas 1 dan *support* 0,1 (10%). *Rule* ini bisa dianggap layak untuk menghasilkan rekomendasi item, yaitu Sirup, sehingga orang yang membeli Gula dan Pet Food bisa ditawarkan dengan item Sirup.

DAFTAR PUSTAKA

- [1] Nursikuwagus, T. Hartono, "Implementasi Algoritma Apriori Untuk Analisis Penjualan Dengan Berbasis Web," Jurnal SIMETRIS, vol. 7, no. 2, hal. 701-706, 2016.
- [2] S.F. Rodiyansyah, "Algoritma Apriori untuk Analisis Keranjang Belanja pada Data Transaksi Penjualan," Infotech Journal, vol. 1, no. 2, 2016.
- [3] W. Aprianti, K.A. Hafidz, M.R. Rizani, "Implementasi Association Rules dengan Algoritma Apriori pada Dataset Kemiskinan," Limits Journal, vol. 14, no. 2, hal. 57-67, 2017.
- [4] A.A. Arafah, I. Mukhlash, "The Application of Fuzzy Association Rule on Co-Movement Analyze of Indonesian Stock Price," Procedia Computer Science, vol. 59, hal. 235-243, 2015.
- [5] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Edisi kedua, San Francisco, CA: Elsevier inc., 2006.
- [6] R.T. Vulandari, Data Mining Teori dan Aplikasi Rapidminer, Edisi pertama, Yogyakarta: CV. Gava Media, 2017, hal. 65.
- [7] Y. Cohen, J.Y. Cohen, Statistics and Data with R: An applied approach through examples. United Kingdom: John Wiley & Sons Ltd., 2008.
- [8] J.M. Zelle, Python Programming: An Introduction to Computer Science. Version 1.0rc2. Wartburg College Printing Services, 2002.
- [9] Hendri, Cepat Mahir Python. 2003, <http://www.IlmuKomputer.com>.
- [10] H.N. Wulandari, N.W. Rahayu, "Pemanfaatan Algoritma Apriori untuk Perancangan Ulang Tata Letak Barang di Toko Busana," Seminar Nasional Aplikasi Teknologi Informasi (SNATI), vol. 1, no. 1, 2014.
- [11] S. Susanto, D. Suryadi, *Pengantar Data Mining: Menggali Pengetahuan dari Bongkahan Data*. Yogyakarta: Andi Offset, 2010.