

Algoritma Apriori Untuk Menemukan Pola Aturan Asosiasi

Wahju Tjahjo Saputro^{1*}, Hamid Muhammad Jumasa¹, Murhadi¹

¹Teknologi Informasi, Universitas Muhammadiyah Purworejo, Purworejo 54111, Indonesia

Abstrak

Asosiasi merupakan salah satu metode pada data mining, yaitu mencari suatu kemiripan pola pada data transaksi atau item tertentu yang sering muncul. Sehingga pihak manajer supermarket dapat menentukan dalam mengambil keputusan terhadap pola item barang yang dibeli bersamaan dan sering muncul pada database. Guna menghasilkan pola dari aturan asosiasi ada beberapa algoritma yang dapat digunakan. Pertama dipelopori oleh Algoritma AIS, kemudian peningkatan kinerja memori oleh algoritma Apriori. Proses algoritma Apriori meliputi beberapa tahap. Seperti menentukan *minimum support*, *confidence* dan kandidat itemset. Proses tersebut berlangsung sampai diperoleh pola sesuai kriteria *support* dan *confidence* dari nilai ambang batas minimal. Sebagian besar algoritma yang digunakan untuk menemukan pola aturan asosiasi berawal dari algoritma Apriori, selanjutnya berkembang ke algoritma Apriori series. Namun ada dua hambatan dari algoritma Apriori. Salah satunya adalah proses pembuatan kandidat yang kompleks dimana sebagian besar membutuhkan waktu, ruang, dan memori. Masalah lain adalah pemindaian berulang-ulang pada database berskala besar.

Kata kunci: Apriori, Support, Confidence, Asosiasi

Abstract

Association is one of the methods in data mining, which is looking for a similar pattern in data transactions or certain items that often appear. So that the supermarket manager can determine in making decisions on the pattern of items purchased together and often appears in the database. In order to generate patterns from association rules there are several algorithms that can be used. First pioneered by the AIS Algorithm, then increased memory performance by the Apriori algorithm. The Apriori algorithm process involves several steps. Such as determining the minimum support, confidence, candidate itemset. The process continues until the pattern is obtained according to the support and confidence criteria of the minimum threshold value. Most of the algorithms used to find patterns of association rules start with the Apriori algorithm, then progress to the Apriori series algorithm. But there are two obstacles from the Apriori algorithm. One of them is the process of making complex candidates that mostly require time, space and memory. Another problem is repeated scanning of large-scale databases.

Keywords: Apriori, Support Confidence, Association

1. PENDAHULUAN

Dua dekade ini sejak 1990an banyak transaksi bisnis proses pembayaran telah dilakukan menggunakan alat memindai barcode. Mayoritas digunakan pada minimarket, supermarket, plaza, mall, hingga berkembang sampai ke hypermart, transmart. Menurut (Agrawal et al., 1993) dengan pemindaian barcode terhadap data transaksi pada kode barang memudahkan data transaksi tumbuh pesat menjadi sebuah bongkahan data yang besar. Karena bongkahan data yang besar tersedia dengan mudah, maka muncul kebutuhan untuk mempelajari pola konsumsi

konsumen yang dapat membantu dalam meningkatkan infrastruktur pemasaran dan disiplin ilmu terkait misal pemasaran yang ditargetkan (Agrawal et al., 1993).

Di pusat perbelanjaan sering terjadi seorang konsumen membeli barang tertentu bersamaan dengan barang lainnya. Baik barang tersebut saling mendukung maupun tidak. Misal bila seseorang membeli roti tawar maka peluang besar juga membeli mentega, coklat atau keju. Kegiatan ini dapat dikategorikan sebagai asosiasi dalam proses bisnis. Pada data mining terdapat sebuah metode yang disebut Asosiasi. Menurut (Agrawal et al., 1993), (Handojo et al., 2004)

Association Rule Mining (ARM) adalah suatu teknik menggali data yang sesuai untuk memperoleh dataset pada Market Basket Analysis (MBA). (Handojo et al., 2004), (Pasa, Ike Y. & Saputro, Wahyu T., 2018) mengatakan metode MBS mampu mencari dan menemukan pola yang berasosiasi diantara produk yang dibeli konsumen. Misal barang A biasanya dibeli bersamaan dengan barang B pada waktu tertentu. Teknik ini disebut juga *association rule analysis*.

Hasil yang didapat dari proses data mining dijadikan suatu pengetahuan baru yang dapat digunakan perusahaan dalam meningkatkan produksi maupun produk penjualannya (Handojo et al., 2004). Misal pada rak sebuah supermarket, bila manajer penjualan telah mengetahui barang A dibeli bersamaan dengan barang B dapat diletakkan berdekatan atau berjauhan. Tergantung dari maksud setiap supermarket. Bila konsumen ingin berlama-lama di supermarket, tentu letak barang A dan B perlu berjauhan. Namun dari aspek kenyamanan berbelanja, konsumen lebih senang barang A letaknya berdekatan dengan barang B.

Menurut (Handojo et al., 2004), (Anisa, Siti & Saputro, Wahyu. T., 2012), (Pasa, Ike Y. & Saputro, Wahyu T., 2018) menjelaskan keputusan lain dari manajer penjualan adalah meningkatkan promosi terhadap item-item barang tertentu agar saling berasosiasi. Hal ini tergantung dari konsep pemasaran supermarket tersebut. Maka dengan adanya algoritma Apriori pola asosiasi dapat terbentuk.

ARM merupakan salah satu teknik penambahan data yang paling baik, penting dan sering digunakan dalam penelitian asosiasi menurut (Agrawal et al., 1993). Pola ARM bertujuan untuk mengekstrak korelasi antar item yang menarik, pola yang sering muncul atau struktur kasual di antara set item dalam database transaksi atau repositori data lainnya (Kaur & Madan, 2015).

Dengan frekuensi data transaksi yang cukup pesat saat ini, (Agrawal et al., 1993) beberapa organisasi sangat mudah memiliki data dalam jumlah besar. Kumpulan data tersebut biasanya disimpan pada penyimpanan tersier dan sangat lambat bermigrasi ke sistem basisdata. Salah satu alasan utama keterbatasan sistem database adalah tidak menyediakan fungsionalitas yang diperlukan oleh manajer guna mengambil suatu keputusan dari informasi berbasis pengetahuan.

(Agrawal et al., 1993) menjelaskan penelitian ARM memperkenalkan masalah sekumpulan transaksi berskala besar jenis data market basket untuk aturan asosiasi antara set item dengan beberapa spesifikasi tertentu, dan menyajikan algoritma yang efisien. (Fürnkranz, 2013) menjelaskan aturan asosiasi ditemukan tahun 1993 oleh (Agrawal et al., 1993), yang mengidentifikasi hubungan item dari sekumpulan dalam database. Hubungan ini tidak berdasarkan properti data tersebut, namun berdasarkan peristiwa yang terjadi pada data tersebut.

Contoh aturan asosiasi yaitu pernyataan bahwa 90% transaksi membeli roti dan mentega juga membeli susu. Anteseden peraturan ini terdiri dari roti dan mentega dan konsekuensinya hanya terdiri dari susu saja. Angka 90% adalah faktor confidence dari aturan tersebut (Agrawal et al., 1993). Sebuah toko mencoba mencari hubungan antara suatu barang dengan barang lainnya, misal selai. Pihak toko menemukan bahwa selai 30% dibeli bersama roti tawar. Namun selai 40% dibeli bersama agar-agar. Berdasarkan fakta ini pihak toko akan menempatkan penjualan barang berdekatan supaya terjadi peningkatan barang yang dibeli bersamaan.

Proses identifikasi keterkaitan antar barang akan mengarah ke pola asosiasi dimana (Agrawal et al., 1993):

- Temukan semua aturan yang memiliki A sebagai *consequent*. Aturan ini mampu membantu apa yang harus dilakukan toko untuk meningkatkan penjual barang A.
- Temukan semua aturan yang memiliki B sebagai *antecedent*. Aturan ini membantu menentukan produk apa yang mungkin berpengaruh jika toko tidak menjual barang B.
- Temukan semua aturan yang mengandung C sebagai *antecedent* dan D sebagai *consequent*. Hal ini dapat diutarakan sebagai alternatif tambahan bila item C dan D juga dijual bersamaan.
- Temukan semua aturan yang terkait barang-barang yang terletak di rak X dan Y. Aturan ini membantu perencanaan tataletak rak yang terkait dengan barang A dan B.
- Temukan aturan k terbaik yang memiliki B sebagai *consequent*. Di sini, “terbaik” dapat dirumuskan sebagai faktor *confidence* atau sebagai *support*, yaitu transaksi yang memenuhi aturan.

2. PEMBAHASAN

2.1. Permasalahan Aturan Asosiasi

Permasalahan aturan asosiasi dalam (Kaur & Madan, 2015) direpresentasikan dengan contoh yang lazim digunakan pada pernyataan ARM yang dijelaskan pada (Agrawal et al., 1993). Misal diberikan $I = I_1, I_2, I_3, \dots, I_m$ adalah satu himpunan atribut, dimana T merupakan sebuah transaksi yang mengandung rangkaian item barang maka dapat dikatakan $T \subseteq I$, dimana D merupakan database yang berisi transaksi yang berbeda-beda pada T s. Aturan asosiasi adalah implikasi yang ditulis sebagai berikut $X \Rightarrow Y$, di mana $X, Y \subset I$ merupakan himpunan yang terdapat pada sebuah itemset, dan $X \cap Y = \emptyset$. X disebut *antecedent* dan Y disebut *consequent*, sehingga aturannya menjelaskan bawah X menyiratkan Y .

Ada dua hal utama dalam aturan asosiasi yaitu *support* (s) dan *confidence* (c) (Kaur & Madan, 2015) (Agrawal et al., 1993) (Hegland, 2007) (Fürnkranz, 2013). Karena database supermarket berukuran besar dan manajer hanya memikirkan barang yang dibeli konsumen, biasanya nilai ambang batas *support* dan *confidence* ditentukan oleh manajer supermarket dengan cara menghilangkan pola-pola asosiasi yang tidak menarik. Dua ambang batas itu disebut *minimum support* dan *minimum confidence* oleh (Agrawal, 2013) (Kaur & Madan, 2015), bila ada pola yang menarik maka manajer supermarket dapat memberikan tambahan *constraints*. Sehingga dua parameter utama dari ARM adalah: *support* dan *confidence*.

2.2. Support

Pada (Agrawal et al., 1993) (Hegland, 2007) (Kaur & Madan, 2015) (Pasa, Ike Y. & Saputro, Wahyu T., 2018) *Support* (s) pada aturan asosiasi ditulis dalam bentuk prosentase dari perhitungan data yang mengandung $X \cup Y$ terhadap jumlah total dalam database. Perhitungan setiap item bertambah satu, ketika proses pemindaian item dijumpai dalam transaksi T yang berbeda pada database D . Artinya *support count* tidak memperhitungkan jumlah item. Misal dalam sebuah transaksi, seorang pelanggan membeli tiga botol air mineral tetapi manajer supermarket hanya menaikkan *support count* {air mineral} satu, dengan kata lain jika sebuah transaksi mengandung sebuah item maka *support count* tersebut

bertambah satu. Maka *Support* (s) dihitung menggunakan rumus berikut:

$$support(XY) = \frac{support\ count\ dari\ XY}{total\ transaksi\ database\ D}$$

Dari perhitungan tersebut dapat diketahui bahwa *support* suatu item merupakan pergerakan statistik yang signifikan dalam aturan asosiasi. Contoh *support* air mineral = 0.1%, artinya hanya 0.1 persen dari semua transaksi yang mengandung pembelian air mineral. Dengan demikian manajer supermarket tidak akan memperhatikan barang lain dengan prosentase yang cukup rendah. Tampak jelas bahwa manajer supermarket lebih menyukai barang yang dibeli konsumen memiliki *support* yang tinggi dalam aturan asosiasi, sehingga tampak lebih menarik untuk diketahui. Sebelum menggali informasi dari dataset supermarket, manajer perlu menentukan *support minimum* sebagai ambang batas. Artinya manajer hanya tertarik pada aturan asosiasi dengan pola pembelian tertentu yang diperoleh dari hasil dataset dengan nilai *support* di atas ambang batas yang diinginkan. Namun, terkadang dijumpai itemset yang tidak sesuai keinginan manajer, dimana itemset tidak sering muncul dengan hasil di bawah ambang batas. Meskipun demikian asosiasi tersebut tetap dianggap penting karena beberapa hal. Misal, beberapa barang cukup mahal karena harga tinggi, sehingga barang tersebut tidak sering dibeli. Namun aturan asosiasi pada barang mahal sama penting dengan barang lain yang sering dibeli oleh pengecer, karena pertimbangan segmen konsumen yang berbeda.

2.3. Confidence

Confidence adalah aturan asosiasi yang ditulis dalam bentuk persentase dari sejumlah transaksi yang mengandung $X \cup Y$, dari total transaksi yang mengandung X , dimana jika persentase yang dihasilkan melebihi ambang batas *confidence* dari aturan asosiasi $X \Rightarrow Y$.

$$confidece(X|Y) = \frac{support(XY)}{support(X)}$$

Confidence merupakan tolok ukur atau sebuah kekuatan pada aturan asosiasi. Misal jika ada *confidence* dari $X \Rightarrow Y$ sebesar 80%, artinya transaksi barang yang mengandung X juga mengandung Y sebesar 80% dalam (Kaur &

Madan, 2015). Hal ini memastikan ketertarikan satu sama lain antara X dan Y. Untuk *support minimum* ditentukan oleh manajer supermarket.

2.4. Association Rule Mining

Pada (Agrawal et al., 1993), (Kaur & Madan, 2015) ARM adalah menemukan aturan asosiasi yang memenuhi *minimum support* dan *confidence* dari sebuah database. Permasalahan dari ARM dapat dipecah menjadi dua yaitu menemukan pola itemset dan menghasilkan aturan asosiasi. Menemukan pola itemset adalah menganalisa dan menggali kemunculan pola yang melebihi nilai ambang batas dari database yang telah ditentukan. Itemset yang sering muncul tersebut dapat dikatakan *frequent itemset* atau *large itemset* (Agrawal et al., 1993) (Fürnkranz, 2013) (Kaur & Madan, 2015).

Guna menghasilkan aturan asosiasi dari itemset dengan menggunakan batasan *confidence* minimal. Misal, sebuah *frequent itemset* dimisalkan Lk, maka $L_k = \{I_1, I_2, I_3, \dots, I_{k-1}, I_k\}$. Dimana aturan asosiasi dari itemsets yang dihasilkan mengikuti langkah aturan pertama yaitu $\{I_1, I_2, I_3, \dots, I_{k-1}\} \Rightarrow \{I_k\}$, berdasarkan aturan pertama ini sebuah aturan dapat ditentukan apakah pola tersebut menarik atau tidak (Agrawal et al., 1993) (Kaur & Madan, 2015). Kemudian aturan kedua dihasilkan dengan menghapus item terakhir pada *antecedent* dan menambahkan ke *consequent*, selanjutnya *confidence* dari aturan baru diperiksa untuk menentukan apakah menarik atau tidak. Proses tersebut diulang sampai *antecedent* kosong. Karena sub permasalahan kedua aturan asosiasi memiliki langkah yang jelas, biasanya peneliti fokus pada sub permasalahan pertama yaitu mencari *frequent itemsets*.

2.5. Pendekatan Aturan Asosiasi

Pada (Kaur & Madan, 2015) menjelaskan bahwa sub permasalahan kedua sangat mudah, maka sebagian besar pendekatan tersebut fokus pada sub permasalahan pertama. Maka Sub permasalahan pertama dapat dibagi lagi menjadi dua sub permasalahan yaitu: proses menghasilkan *candidate large itemset* dan proses menghasilkan *frequent itemset*. Itemset yang nilai *support* melebihi ambang batas disebut *frequent*, sedangkan itemset yang memiliki harapan nilai *support*nya besar disebut *candidate itemset*. Menurut (Kaur & Madan, 2015) sebagian besar algoritma pada aturan asosiasi yang di survei cukup mirip, perbedaannya pada perluasan

kinerja hingga dimana perbaikan algoritma tersebut ada improvisasi.

2.6. Seri Pendekatan Algoritma Apriori

Pada pembahasan ini digunakan database yang sama sebagai contoh, guna memudahkan dalam membandingkan setiap series Apriori.

2.6.1. Algoritma AIS

Pada (Kaur & Madan, 2015) algoritma AIS (Agrawal, Imielinski, Swami) merupakan sebuah algoritma pertama yang diusulkan untuk aturan asosiasi (Agrawal et al., 1993). Algoritma Ini fokusnya pada peningkatan kualitas database dan fungsionalitas yang diperlukan untuk memproses permintaan manajer dalam mengambil keputusan. Algoritma AIS, hanya satu item aturan asosiasi yang dihasilkan, artinya consequent dari aturan tersebut berisi satu item. Misal hanya menghasilkan pola $X \cap Y \Rightarrow Z$ bukan pola aturan sebagai $X \Rightarrow Y \cap Z$ (Kaur & Madan, 2015).

TID	Daftar Items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3
T000	I1, I2, I5, I6

(a) Database asli

Items	Jml Count
I1	7
I2	8
I3	6
I4	2
I5	3
I6	1

(b) Tahap C1

Frequent Items
I1
I2
I3
I5

(c) Tahap L1

Items	Jml Count
I1, I2	5
I1, I3	4
I1, I4	1
I1, I5	3
I1, I6	1
I2, I3	4
I2, I4	2
I2, I5	3
I2, I6	1
I3, I4	0
I3, I5	1
...	...

(d) Tahap C2

Frequent Items
I1, I2
I1, I3
I1, I5
I2, I3
I2, I5

(e) Tahap L2

Items	Jml Count
I1, I2, I3	2
I1, I2, I4	1
I1, I2, I5	3
I1, I2, I6	1
I2, I3, I4	0
I2, I3, I5	1
I2, I3, I6	0
I3, I4, I5	0
...	...

(e) Tahap C3

Gambar 1. Proses AIS Mining

Pada algoritma AIS database dipindai beberapa kali untuk memperoleh *frequent itemset* yang sering muncul. Selama membaca data tahap pertama, jumlah *support* dari setiap item di akumulasi seperti yang ditunjukkan pada Gambar 1 (b), misal ambang batas *minimum support* 30%, besarnya satu item yang dihasilkan ditunjukkan pada Gambar 1 (c). Sesuai *minimum support*, itemset dengan jumlah *support* ≤ 3 dihapus dari *frequent itemsets*. Berdasarkan 1 *frequent itemsets*, untuk memperoleh kandidat 2-item (C-2) dengan cara memperluas *frequent itemset* satu dengan item lain dalam satu database. Untuk

menghindari hasil itemset yang sama dan berulang kali, kandidat itemset yang dihasilkan digabung dengan itemset bernilai besar. Misal, berdasarkan transaksi T100 I1, I2, I5, sesuai hasil kandidat 2-itemsets I1 dapat diperluas dengan mengkombinasikan I2 sampai I5. Hasilnya ditunjukkan pada Gambar 1 (d). Selama proses pembacaan data transaksi kedua, *support count* dari kandidat 2-itemsets telah terakumulasi kemudian melihat nilai ambang batasnya. Demikian proses tersebut berjalan sampai kandidat (k+1)-itemsets dihasilkan, sampai *frequent itemsets* menghasilkan nilai kosong. Hasil fre-

quent itemsets hanya satu 3-itemset yaitu {I1, I2, I5} (Kaur & Madan, 2015).

Untuk membuat algoritma AIS lebih efisien, Metode Estimasi diperkenalkan guna memangkas kandidat item yang tidak memiliki harapan menjadi itemset yang besar, dampaknya perhitungan yang tidak perlu dapat dihindari. Karena semua itemset yang ditentukan tersimpan di memori utama (RAM), maka manajemen memori juga diusulkan dalam penelitian tersebut sebagai dampak dari penggunaan Algoritma AIS. Salah satu pendekatannya yaitu menghapus kandidat itemet yang belum pernah dikombinasi. Pendekatan lain dengan menghapus kandidat itemset yang telah memiliki jumlah maksimum itemsets, dan menyimpan itemset ini pada memori utama sebagai calon itemset yang akan diproses tahap berikutnya (Agrawal et al., 1993).

Kelemahan utama dari algoritma AIS adalah terlalu banyak kandidat itemset bernilai kecil turut dihitung dan muncul dalam hasil akhir, yang membutuhkan lebih banyak ruang dan waktu yang ternyata tidak berguna.

2.6.2. Algoritma Apriori

Apriori adalah peningkatan kinerja sebuah algoritma yang cukup signifikan pada Metode Asosiasi, algoritma Apriori pertama kali diusulkan oleh Agrawal dalam (Agrawal et al., 1993). Dimana algoritma AIS menggunakan pendekatan langsung yang membutuhkan banyak proses pembacaan data pada database, sehingga proses menemukan kandidat itemset cukup lama. Algoritma Apriori lebih efisien dalam menghasilkan kandidat itemset karena dua hal, yaitu Apriori menggunakan metode pembuatan kandidat yang berbeda dan teknik pemangkasan baru (Hegland, 2007) (Kaur & Madan, 2015).

Ada dua tahap menemukan *frequent itemset* dari sebuah database menggunakan Algoritma Apriori. Pertama menghasilkan kandidat itemset, dimana database melakukan pemindaian terhadap itemset secara langsung yang memenuhi *support*. Selama proses pemindaian pertama, *support count* setiap item dihitung dan dijadikan sebagai kandidat 1-itemset dengan memangkas item yang memiliki *support* di bawah nilai ambang batas yang telah ditentukan. Misal ditentukan nilai *support* ≤ 2 . Seperti ditunjukkan pada Gambar 2 (a) dan Gambar 2 (b).

Items	Jml Count
I1	7
I2	8
I3	6
I4	2
I5	3
I6	1

(a) C1

Frequent Items
I1
I2
I3
I5

(b) L1

Items	Jml Count
I1, I2	5
I1, I3	4
I1, I5	3
I2, I3	4
I2, I5	3

(c) C2

Frequent Items
I1, I2
I1, I3
I1, I5
I2, I3
I2, I5

(d) L2

Items	Jml Count
I1, I2, I3	2
I1, I2, I5	3

(e) C3

Gambar 2. Proses Apriori

Calon k-itemset diperoleh setelah melewati (k-1), kemudian digabungkan dengan k-1 *frequent itemsets*. Semua kandidat k-item dipangkas dengan memeriksa (k-1)-itemset, jika setiap item (k-1) tidak terdapat dalam daftar *frequent* (k-1)-itemset, karena k-itemset tidak memiliki harapan menjadi *frequent* menurut aturan Apriori. Aturan Apriori menjelaskan bahwa setiap (k-1) itemset dari *frequent* k-itemset tersebut harus sering muncul. Misal, diambil hasil kandidat 3-itemset sebagai contoh. Pertama semua kandidat itemset yang dihasilkan digabung dengan 2-itemset yang sering muncul, meliputi (I1, I2, I3) (I1, I2, I5) (I1, I3, I5) (I2, I3, I5).

Itemset tersebut kemudian diperiksa sub itemnya, karena (I3, I5) tidak sering muncul pada *frequent 2-itemset*. Dua 3-itemset terakhir dihilangkan dari daftar kandidat 3-itemset atau kandidat 3-itemset yang tidak memenuhi nilai ambang batas *support* dihilangkan, lihat Gambar 2 (e). Semua proses tersebut dijalankan secara iteratif guna menemukan semua *frequent item-*

sets. Proses pseudocode algoritma Apriori ditunjukkan Gambar 3.

```

Input:
  Database D
  Minimum support  $\epsilon$ 
  Minimum confidence  $\xi$ 
Output:
  Rt All AR
Method:
01 L1 = Large 1-itemset;
02 for (k=2; Lk-1 $\neq\emptyset$ ; k++) do begin
03   Ck=apriori-gen(Lk-1)
04   for all trans T $\in$ D do begin
05     Ct=subset(Ck, T);
06     for all cand C  $\in$  Ct do
07       count(C)=count(C)+1;
08     end
09     Lk={C $\in$ Ct | Count(C)  $\geq\epsilon \times |D|$ }
10   end
11 Lf =  $\cup_k L_k$ ;
12 Rt = generate rules (Lf,  $\xi$ )

```

Gambar 3. Algoritma Apriori

Dalam proses menemukan *frequent itemset*, algoritma Apriori berusaha menghindari pemborosan dalam menghitung kandidat itemset yang diketahui tidak akan sering muncul. Selanjutnya kandidat itemset yang dihasilkan digabung dengan itemset lain, kandidat yang tidak sesuai dipangkas berdasarkan aturan Apriori. Gambar 2 menunjukkan proses keseluruhan algoritma Apriori. Dengan membandingkan Gambar 1 dan 2 dapat dilihat jumlah kandidat berubah secara signifikan. Secara rinci fungsi Apriori-gen dan Generate Rules diuraikan dalam (Agrawal et al., 1993).

Algoritma Apriori masih mewarisi kelemahan AIS dari aspek pemindaian data transaksi berkali-kali. Berdasarkan algoritma Apriori, saat itu mulai bermunculan algoritma baru yang dirancang, dimodifikasi dan mengalami perbaikan dari aspek kinerja, kecepatan maupun proses peningkatan dalam pemindaian. Secara umum ada dua pendekatan yaitu mengurangi jumlah pemindaian data di dalam database dan merubah semua atau hanya sebagian data berdasarkan item yang sering muncul. Pendekatan lain adalah mengeksplorasi berbagai teknik pemangkas data supaya jumlah kandidat itemset lebih sedikit. Beberapa algoritma yang meningkatkan kinerja Apriori yaitu Apriori-TID dan Apriori

Hybrid oleh (Agrawal et al., 1993), DHP (Park Jong et al., 1995), SON (Savasere et al., 1995).

3. KESIMPULAN

Sebagian besar algoritma yang diperkenalkan di atas didasarkan pada algoritma Apriori dan mencoba untuk meningkatkan efisiensi dengan membuat beberapa modifikasi. Seperti mengurangi jumlah pemindaian data transaksi pada database; mengurangi ukuran database yang akan dipindai, memangkas kandidat dengan teknik yang berbeda dan menggunakan teknik pengambilan sampel. Namun ada dua hambatan dari algoritma Apriori. Salah satunya adalah proses pembuatan kandidat yang kompleks yang sebagian besar membutuhkan waktu, ruang, dan memori. Hambatan lain adalah pemindaian berulang-ulang pada database.

DAFTAR PUSTAKA

- Agrawal, R. (2013). Fast Algorithms For Mining Association Rules In Datamining. *International Journal of Scientific & Technology Research*, 2(12), 13–24.
- Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. *ACM SIGMOD Record*, 22(2), 207–216. <https://doi.org/10.1145/170036.170072>
- Anisa, Siti, & Saputro, Wahyu. T. (2012). *Implementasi Data Mining Menggunakan Metode Deskripsi Untuk Mengetahui*. 1–12.
- Fürnkranz, J. (2013). Association Rule. *Encyclopedia of Systems Biology*, 7(1), 47–47. https://doi.org/10.1007/978-1-4419-9863-7_838
- Handojo, A., Budhi, G. S., & Rusly, H. (2004). Aplikasi Data Mining Untuk Meneliti Asosiasi Pembelian Item Barang di Supermarket Dengan Metode Market Basket Analysis. *Seminar Nasional Teknologi Informasi 2004*, 10–17.
- Hegland, M. (2007). *the Apriori Algorithm – a Tutorial*. 209–262. https://doi.org/10.1142/9789812709066_0006

- Kaur, J., & Madan, N. (2015). Association Rule Mining: A Survey. *International Journal of Hybrid Information Technology*, 8(7), 239–242.
<https://doi.org/10.14257/ijhit.2015.8.7.22>
- Park Jong, S., Chen, M., & Yu, P. S. (1995). *An Effective Hash-Based Algorithm for Mining Association Rules*. 175–186.
- Pasa, Ike Y., & Saputro, Wahyu T. (2018). Pendekatan Algoritma Apriori pada Data Mining untuk Menentukan Pola Belanja Konsumen. *Intek*, 1 No 1(2009), 1–9.
- Savasere, A., Omiecinski, E., & Navathe, S. (1995). *An Efficient Algorithm for Mining Association Rules in Large Databases*. 368.