

PENEMUAN POLA ASOSIASI PADA DATA RESTORAN MENGGUNAKAN ALGORITMA HASH BASED

Triaji Doni Prakoso¹, Iin Ernawati², Henki Bayu Seta³
Program Studi Informatika Fakultas Ilmu Komputer
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Jawa Barat 12450
Email : triajidonip2@gmail.com
iinerti@gmail.com
henkiseta@upnvj.ac.id

Abstrak. Restoran Waroeng Djamet merupakan salah satu restoran yang bergerak dalam bisnis kuliner. Banyaknya restoran yang bermunculan mengakibatkan ketatnya persaingan bisnis, sehingga restoran harus terus melakukan inovasi baik dalam segi rasa maupun strategi penjualan oleh pihak manajemen. Hal tersebut dapat diketahui dengan menggunakan metode data mining yaitu *frequent itemset mining* atau algoritma *Hash Based* dan *association rule* yang dapat menjadi solusi untuk mengatasi masalah penentuan *frequent k-itemset* dari kandidat *k-itemset* dengan jumlah yang besar dan juga mempercepat pemrosesan atau pengolahan informasi tentang kebiasaan pelanggan mengenai makanan atau minuman yang dibeli. Dengan pola kombinasi tersebut dapat menghasilkan informasi penting yang bermanfaat untuk yang akan diambil oleh Restoran Waroeng Djamet. Hasil penelitian ini berupa rekomendasi itemset yang dapat menganalisa pola transaksi yang dibeli, terdapat pola yang dihasilkan dan digunakan untuk strategi penjualan pada restoran Waroeng Djamet, seperti desain katalog penjualan, eliminasi produk, mengetahui segmentasi pelanggan, dan penjualan silang (cross-selling).

Kata Kunci: *Data Mining, Association Rule, Algoritma Hash Based, Frequent Itemset, Restoran Waroeng Djamet.*

1 Pendahuluan

Latar Belakang

Perkembangan teknologi di era saat ini memungkinkan pengolahan data yang ada disimpan pada basis data. Kemudian data-data dari data transaksi penjualan yang ada akan dianalisis maka dari itu penulis menggunakan data mining sebagai dasarnya. Data mining merupakan proses penelusuran data yang dapat menjadi informasi yang belum diketahui dengan cara manual, melakukan pengolahan data sehingga menjadi pengetahuan juga bisa dilakukan dengan cara mengetahui pola-pola dengan metode penggalian [1].

Algoritma apriori memiliki kelemahan diantaranya dalam menentukan *frequent k-itemset*, algoritma apriori melakukan *scan* basis data pada tiap iterasi sehingga banyak memakan memori yang ada. Hal inilah yang melatar belakangi penulis menggunakan algoritma *hash-based* yang lebih efisien dan cenderung memakan memori lebih sedikit yang hasilnya dituangkan dalam judul penelitian “Penemuan Pola Asosiasi Pada Data Restoran Menggunakan Algoritma *Hash Based*” [3].

Tujuan Penelitian

Tujuan pada penelitian ini adalah:

- Menerapkan algoritma *Hash Based* pada data transaksi restoran Waroeng Djamet.
- Untuk menemukan pola asosiasi terhadap item menu yang banyak dipesan dalam transaksi maupun tidak.
- Membantu memberikan informasi dari data transaksi yang diolah dengan algoritma *Hash Based*.
- Pola yang ditentukan dapat dijadikan rekomendasi untuk pengambilan keputusan.

Batasan Masalah

Untuk menghindari terjadinya penyimpangan pada pokok masalah maka peneliti memberikan batasan masalah, yaitu:

- Penerapan data mining hanya pada data transaksi penjualan restoran Waroeng Djamet.
- Pengolahan data hanya dengan menggunakan algoritma *Hash Based*.
- Penggunaan algoritma *Hash Based* untuk menemukan *frequent* itemset, kemudian diolah kedalam aturan asosiasi.

2 Landasan Teori

Data Mining

Data mining merupakan proses penelusuran data yang dapat menjadi informasi yang belum diketahui dengan cara manual, melakukan pengolahan data sehingga menjadi pengetahuan juga bisa dilakukan dengan cara mengetahui pola-pola dengan metode penggalian [1].

Penyimpangan data transaksi juga perlu di telusuri lebih lanjut agar dapat mengambil jalan kesimpulannya. Hal tersebut disinyalir cara pembentukan langkah yang tepat dalam mendukung kegiatan pengolahan data transaksi restoran yang ada [4].

Data mining adalah sebuah otomatisasi dalam pencarian informasi pada wadah penyimpanan data yang berukuran besar, biasanya juga disebut dengan KDD yang merupakan *knowledge discovery (mining) in database*. Data mining juga dapat menemukan pola yang tidak disadari keberadaannya [5].

Analisa Asosiasi

Aturan asosiasi atau *association rule* adalah suatu *itemset disjoint* yang terdiri dari A dan B, dikatakan bahwa pemenuhan persyaratan $(A, B) = \emptyset$ terbentuk dari ekspresi implikasi $A \rightarrow B$. Aturan asosiasi memiliki kekuatan utama yang terdiri dari *support* dan *confidence*. *Support* merupakan pentapan dari banyaknya pembentukan aturan yang diterapkan pada data set, definisi formal untuk metrik tersebut adalah sebagai berikut [7]:

$$\text{Support, } s(X) = \frac{\text{Jumlah Transaksi Mengandung } A}{\text{Total Transaksi}}$$

$$\text{Support, } s(A, B) = \frac{\text{Jumlah Transaksi Mengandung } A \text{ dan } B}{\text{Total Transaksi}}$$

(2.1) (2.2)

Setelah semua pola frekuensi itemset ditemukan, kemudian dicari aturan asosiasi yang memenuhi minimum confidence, untuk menghitung confidence aturan asosiasi $(A \rightarrow B)$, Nilai confidence dari aturan $(A \rightarrow B)$ diperoleh dari rumus berikut:

$$\text{Confidence, } c(A \rightarrow B) = \frac{\text{Jumlah Transaksi Mengandung } A \text{ dan } B}{\text{Jumlah Transaksi Mengandung } A}$$

(2.3)

Dijelaskan bahwa terdapat beberapa variasi untuk meningkatkan efisiensi dari algoritma apriori. Salah satu varian tersebut adalah teknik Hash Based. Algoritma Hash Based dapat mengurangi jumlah dari kandidat k-itemset. Efisiensi juga terjadi pada frequent itemset ketika pembangkit kandidat itemset terutama frequent 2-itemset. Hal tersebut dapat meningkatkan performa data mining [6].

Algoritma Hash Based merupakan algoritma yang termasuk dalam golongan frequent itemset mining yang bertujuan untuk mencari itemset yang memenuhi minimum support (Prasetyo, 2012:315). Dalam algoritma *Hash Based* penggunaan teknik *hashing* dalam pembangkitan *itemset* selanjutnya dengan cara menyaring *itemset* yang tidak penting.

Kemudian kandidat k-itemset dari support count dihitung dengan menelusuri basis data, algoritma Hash Based akan mengumpulkan informasi tentang (k+1)-itemset dengan cara seluruh kemungkinan (k+1)-itemset di hash kedalam tabel hash dengan menggunakan sebuah bilangan prima untuk operasi modular. Didalam setiap bucket tersebut pada hash table berisi berapa kali itemset telah di hash. Berdasarkan hash table tersebut akan dibuat node dimana node bernilai 1 jika angka pada bucket yang bersangkutan lebih besar atau sama dengan minimum support. Pembangkitan kandidat yang telah menghitung $C_k = L_{k-1} * L_{k-1}$, pada setiap k-itemset diperiksa apakah itemset tersebut dapat di hash kedalam bucket yang memiliki node yang sama dengan yang pertama. Namun itemset tidak dipergunakan bila tidak ada yang sama dengan yang pertama. Dengan menggunakan tabel hash dapat mengurangi

jumlah kandidat k-itemset kemudian juga dapat mengurangi nilai komputasi dari pembangkit itemset pada setiap iterasinya.

Untuk melakukan pembangkitan frequent itemset pada algoritma Hash Based, terbagi menjadi tiga bagian diantaranya [2]:

- Bahwa algoritma Hash Based akan menghasilkan C1 –atau kandidat 1-itemset dan L1 atau large 1-itemset dari basis data. Pada kandidat 1-itemset, bahwa seluruh transaksi akan ditelusuri untuk menghitung support count dari itemset. Kemudian pada tahap hash tree untuk C1 dibangun dengan tujuan agar perhitungan support count menjadi efisien. Algoritma Hash Based sendiri akan memeriksa apakah setiap item sudah berada didalam hash tree. Jika belum maka item dengan jumlah yang sama akan dimasukkan kedalam hash tree, jika sudah ada, maka jumlah tersebut akan ditambahkan satu. Pada tahap ini algoritma Hash Based akan membangun tabel hash dengan fungsi hash dengan 2-itemset yang berfungsi untuk mengurangi kandidat dari banyaknya itemset pada C2.
- Sekumpulan kandidat itemset Ck akan dibandingkan berdasarkan tabel hash yang telah di buat pada iterasi sebelumnya. Kemudian akan ditentukan frequent itemset Lk lalu akan mengurangi ukuran basis data untuk melakukan pembangkitan selanjutnya. Selain itu pada tabel hash untuk kandidat (K+1)-itemset pada algoritma ini dibagi menjadi dua fase. Fase pertama dilakukan pada pembangkitan kandidat k-itemset Ck berdasarkan pada Hash Based Hk, proses ini dilakukan ketika melakukan prosedur gen_candidate atau kandidat generasi. Pada algoritma ini dapat membangkitkan k-itemset dengan Lk-1, tetapi uang unik pada bagian ini digunakan bit vectr untuk menguji validitas dari setiap k-itemset. Untuk seluruh itemset dari Lk-1*Lk-1, algoritma Hash Based hanya akan menambahkan k-itemset yang melewati penyaringan untuk dimasukkan pada Ck yang merupakan hash tree. Fase yang kedua bahwa algoritma ini akan menghitung support data kandidat itemset dan akan mengurangi ukuran dari setiap transaksi-transaksi yang ada, proses ini terjadi pada prosedur count support. Kemudian akan dibangkitkan tabel hash untuk (k+1)-itemset dan dilakukan lagi pemangkasan data transaksi.
- Seperti pada bagian kedua penggunaan tabel hash memang mirip dengan algoritma apriori. Pada bagian kedua akan dilakukan selama nilai hash bucket lebih besar dari minimum support. Setelah batasan terlewati, algoritma Hash Based akan digantikan dengan algoritma apriori dikarenakan tidak lebih efisien dibandingkan dengan algoritma apriori.

Algoritma Hash Based

Pada jurnal “A Hash Based Frequent Itemset Mining Using Rehashing”, menyatakan bahwa mengusulkan algoritma aturan asosiasi mining baru yang di sebut RBFi dimana teknologi *hashing* yang ada digunakan untuk menyimpan basis data kedalam format data vertical [2]. Cara kerjanya dimana semua basis data akan direpresentasikan kedalam basis data vertikal, untuk menemukan dan menghasilkan *frequent* itemset baru dimana:

$$n = 2 * m + 1 \quad (2.4)$$

Keterangan:

m = jenis makanan pada seluruh pesanan n
 = banyaknya alamat

Setelah didapatkan nilai n, perhitungan akan dilanjutkan, namun apa bila terjadi tabrakan (collision) harus segera dilakukan pengecekan untuk mencari alamat *bucket* yang masih kosong. Kemudian melakukan teknik *rehashing* ulang dengan banyak alamat 2 kali dari alamat sebelumnya. Untuk menyelesaikan masalah tersebut dipergunakan rumus yang berbeda yaitu:

$$j = 2 * n + 1 \quad (2.5)$$

Setelah alamat berubah karena terdapatnya *collision* maka rumus *hash* menjadi:

$$h(k) = ((\text{order of item X}) * 10 + \text{order of item Y}) \bmod j \quad (2.6)$$

Keterangan:

h = alamat *bucket* pada tabel *hash*

Order of item X = perwakilan nilai X

Order of item Y = perwakilan nilai Y

n = banyaknya alamat awal

j = banyaknya alamat setelah dilakukan penambahan

Hash Function

Pengurangan ukuran dengan *hash function* pada bagian kedua dan tiga terdiri dari rumus sebagai berikut [2]:

- Rumus *hash* untuk proses tabel *hash* bagian iterasi satu

$$h(k) = (\text{Order of item } W) \bmod n \quad (2.7)$$

b. Rumus *hash* untuk proses tabel *hash* bagian iterasi dua

$$h(k) = ((\text{order of item } W) * 66 + \text{order of item } X) \bmod n \quad (2.8)$$

c. Rumus *hash* pemrosesan tabel *hash* pada bagian itersi tiga

$$h(k) = ((\text{order of item } W) * 4356 + (\text{order of item } X) * 66 + (\text{order of item } Y) \bmod n \quad (2.9)$$

d. Rumus *hash* pemrosesan tabel *hash* pada bagian itersi empat

$$h(k) = ((\text{order of item } W) * 4356 + (\text{order of item } X) * 66 + (\text{order of item } Y) * 287496 + (\text{order of item } Z) \bmod n \quad (2.10)$$

Keterangan:

h = alamat *bucket* pada tabel *hash*

order of item X = perwakilan nilai X

order of item Y = perwakilan nilai Y

order of item Z = perwakilan nilai Z

n = banyaknya alamat

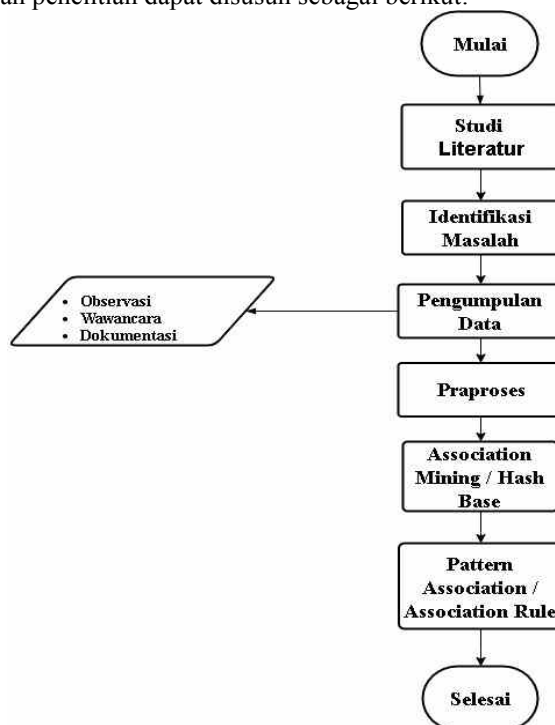
66 = merupakan konstanta yang digunakan karena total menu pada restoran

4356 = merupakan konstanta hasil dari 66^2

287496 = merupakan konstanta hasil dari 66^3

Metodologi Penelitian

Perumusan dan penetapan tujuan penelitian dapat disusun sebagai berikut:



Gambar 1 Kerangka Pikir

Data Praproses

Pada tahap ini data akan dilakukan proses *cleaning*, *selection*, *integration*, dan *transformation*. Dimana data akan diseleksi agar mendapatkan data yang bersih dan siap digunakan dalam penelitian. Tahap yang dilakukan adalah dengan melakukan integrasi terhadap data transaksi penjualan menurut *field* *kd_transaksi* pada tabel transaksi, kemudian melakukan perubahan terhadap beberapa tipe data pada atribut dengan tujuan untuk mempermudah pemahaman terhadap isi data, dengan memperhatikan konsistensi data, *missing value*, dan *redudant* pada data. Selanjutnya dilakukan transformasi data sebagai *input* dari model yang akan melakukan praproses.

Association Mining (Hash Based)

Pada tahap ini dilakukan proses pengolahan data untuk mendapatkan kombinasi item dengan aturan *assosiatif*. Pencarian kombinasi itemset mining menggunakan algoritma hash based.

Selanjutnya dilakukan proses *hashing* terhadap kandidat 1-itemset dengan tujuan memasukan itemset kedalam *bucket* menggunakan rumus (2.7) pada halaman 11. Dimana nilai n yang merupakan jumlah alamat dicari dengan rumus (2.4) pada halaman 11.

Kemudian setelah mendapatkan alamat selanjutnya alamat akan di masukan kedalam C1. Setelah dilakukan perhitungan *hash*, itemset akan menempati alamat *hash* sesuai dengan hasil perhitungan menggunakan rumus *hashing*. Kemudian alamat-alamat akan menjadi *node* dan membuat link yang mengarah kepada transaksi yang memiliki itemset tersebut. Selanjutnya dengan menentukan minimal *support count* = dihasilkan *frequent* 1-itemset (L1). Selanjutnya melakukan pencarian *frequent* 2-itemset (L2), sebelumnya akan ditentulan kemungkinan 2-itemset yang terbentuk dari. Kandidat 2-itemset yang terbentuk melalui *frequent* 1-itemset kemudian memasukan 2-itemset tersebut kedalam tabel *hash* dengan menggunakan rumus (2.8) pada halaman 11.

Pada saat melakukan perhitungan *hashing* masih terdapat tabrakan (*collision*) itemset, yang berarti terdapat 1 itemset yang memiliki alamat *hash* yang sama. Untuk alamat yang sama diberikan tanda (*). Karena terdapat sebuah *collision* maka harus segera dilakukan pengecekan untuk mencari alamat *bucket* yang masih kosong. Untuk menyelesaikan permasalahan tersebut dengan menggunakan rumus: (2.6). Dimana nilai $mod\ j$ merupakan banyaknya alamat setelah dilakukan penambahan dengan rumus (2.5) pada halaman 11. Setelah diketahui bahwa sudah tidak terdapat *collision* dan semua *bucket* sudah terisi maka didapatkan hasil C2. Dari tabel *hash* tersebut akan di lakukan eliminasi itemset terhadap alamat *bucket* dengan *support* ≥ 2 untuk mendapatkan *frequent* 2-itemset.

Setelah mendapatkan *frequent* 2-itemset (L2), langkah selanjutnya adalah mencari *frequent* 3-itemset (L3) dengan terlebih dahulu mencari semua kemungkinan 3-itemset yang untuk memasukan 3-itemset tersebut ke dalam tabel *hash* digunakan rumus (2.9) pada halaman 12. Setelah mendapatkan *frequent* 3-itemset (L3), langkah selanjutnya adalah mencari *frequent* 4-itemset (L4) dengan terlebih dahulu mencari semua kemungkinan 3-itemset yang untuk memasukan 4-itemset tersebut ke dalam tabel *hash* digunakan rumus (2.10). Setelah dilakukan 4 kali iterasi dan yang tersisa 3 alamat pada tabel *hash* dengan 3-itemset (L3), kemudian dilakukan perhitungan untuk mencari nilai *support* dari tabel iterasi ke pertama, kedua, ketiga dan keempat dengan rumus (2.1) dan (2.3) pada halaman 8.

3 Hasil dan Pembahasan

Data mining

Hasil dari data pola kombinasi *hash based*, merupakan data yang telah dilakukan prosesnya dengan algoritma *hash based*. Data yang akan ditampilkan merupakan data pola kombinasi dengan periode data sejak 1 Januari 2018 sampai 31 Desember 2018. Dibawah akan ditampilkan contoh perhitungan pada bulan Januari terdapat 538 transaksi. Langkah awal penentuan tabel 11 sebelum melakukan proses perhitungan di iterasi 1.

Tabel 1 Order of Item

Item	Order
Kopi tubruk	1
Es teh manis	2
Teh manis hangat	3
Teh tawar	4
...	...
Lemon tea	63
Es teh Tarik	64
Teh Tarik hangat	65

Tabel 1 menjelaskan item yang terdapat pada seluruh itemset dibulan januari dan memberikan kode order yang akan digunakan untuk perhitungan mencari *adres* di setiap iterasi. Penentuan nilai mod menggunakan rumus (2.4), jumlah keseluruhan item pada bulan januari yaitu $65 \times 2 + 1$ menjadi 131. Nilai 131 ini menjadi mod yang akan digunakan tabel 12, sebagai berikut:

Tabel 2 Mencari Adres Hash 1 Itemset

No	Item	Formula	Hasil
----	------	---------	-------

1	Kopi tubruk	(1) mod 131	1
2	Es teh manis	(2) mod 131	2
3	Teh manis hangat	(3) mod 131	3
4	Teh tawar	(4) mod 131	4
...
63	Lemon tea	(63) mod 131	63
64	Es teh tarik	(64) mod 131	64
65	Teh Tarik hangat	(65) mod 131	65

Tabel 2 menghasilkan nilai address untuk urutan address pada tabel hash 1 itemset.

Tabel 3 Hash 1 Itemset

No	Address	Itemset	Link	SCount
1	0			
2	1	Kopi tubruk	NOTA0000004 NOTA0000009 NOTA0000010 ... NOTA0000534 NOTA0000536	120
3	2	Es teh manis	NOTA0000020 NOTA0000030 NOTA0000039 ... NOTA0000492 NOTA0000528	35
4	3	Teh manis hangat	NOTA0000011 NOTA0000069 NOTA0000074 ... NOTA0000435 NOTA0000472	16
...
64	63	Lemon tea	NOTA0000145	1
65	64	Es teh tarik	NOTA0000009 NOTA0000021 NOTA0000028 ... NOTA0000521 NOTA0000529	16
66	65	Teh Tarik hangat	NOTA0000315 NOTA0000336 NOTA0000357 NOTA0000374 NOTA0000455 NOTA0000535	6

Tabel 3 menjelaskan hasil dari tabel 12 menempati field *address*, itemset akan menjadi *node* dan membuat link kepada transaksi yang ada pada bulan januari. jumlah link akan masuk ke SCount atau *Support Count*. Nilai SCount yang dibawah minimum *support* yaitu 20 akan dieliminasi.

Tabel 4 Frequent 1 Itemset

No	Itemset	SCount	Support
1	Kopi tubruk	120	23%
2	Es teh manis	35	7%
5	Milk shake coklat	31	6%
7	Milk shake choco oreo	29	6%
...
59	Chicken bento	54	11%
60	Nasi goreng spesial	54	11%
62	Air mineral	27	6%

Tabel 4 menghasilkan itemset yang sudah bebas dari minimum *support*, nilai *support* didapatkan dengan rumus (2.1) pada halaman 8. Kemudian dilakukan pada perhitungan iterasi 2 dengan pola kombinasi item dari hasil tabel 14, menggunakan rumus (2.8) pada halaman 11 untuk mencari address 2 itemset.

Tabel 5 Mencari Address 2 Itemset

No	Itemset	Formula	hasil
1	Kopi tubruk, Es teh manis	$(1 * 66 + 2) \text{ mod } 131$	68
2	Kopi tubruk, Milk shake coklat	$(1 * 66 + 5) \text{ mod } 131$	71
3	Kopi tubruk, Milk shake choco oreo	$(1 * 66 + 7) \text{ mod } 131$	73
...
187	Kopi saring susu, Fanta float	$(12 * 66 + 18) \text{ mod } 131$	24
...
433	Chicken bento, Nasi goreng spesial	$(59 * 66 + 60) \text{ mod } 131$	24

434

	Nasi goreng spesial, Air mineral	$(60 * 66 + 62) \bmod 131$	92
435	Chicken bento, Air mineral	$(59 * 66 + 62) \bmod 131$	26

Tabel 5 menghasilkan jumlah hash yang masih *collosion* salah satunya terjadi pada baris 429 dan 433. *Collosion* atau tumbukan merupakan hasil dari pencarian address yang hasilnya sama. Langkah yang harus dilakukan ialah dengan merubah nilai mod dengan rumus (2.5) pada halaman 11 menjadi $131 * 2 + 1 = 263$. *Collosion* terjadi 5 kali hingga nilai mod berubah beberapa kali menjadi 263, 527, 1055, 2111, 4223.

Tabel 6 Loop 5 Mencari Address 2 itemset

No	Itemset	Formula	hasil
1	Kopi tubruk, Es teh manis	$(1 * 66 + 2) \bmod 4223$	68
2	Kopi tubruk, Milk shake coklat	$(1 * 66 + 5) \bmod 4223$	71
3	Kopi tubruk, Milk shake choco oreo	$(1 * 66 + 7) \bmod 4223$	73
4	Kopi tubruk, Milk shake choco hazelnut	$(1 * 66 + 8) \bmod 4223$	74
...
433	Chicken bento, Nasi goreng spesial	$(59 * 66 + 60) \bmod 4223$	3954
434	Chicken bento, Air mineral	$(59 * 66 + 62) \bmod 4223$	3956
435	Nasi goreng spesial, Air mineral	$(60 * 66 + 62) \bmod 4223$	4022

Tabel 6 sudah tidak terdapat *collosion* sehingga hasil bisa masuk kedalam tabel hash. Dibawah ini ditampilkan tabel 17 *field* hasil pada tabel 16 menjadi urutan address seperti berikut berikut.

Tabel 7 Hash 2 Itemset

No	Address	Itemset	Link	Scount
1	0	-	-	-
...
69	68	Kopi tubruk, Es teh manis	NOTA0000020 NOTA0000030 ... NOTA0000192 NOTA0000312	11
...
72	71	Kopi tubruk, Milk shake coklat	NOTA0000084 NOTA0000100 ... NOTA0000507	5
73	72	-	-	-
74	73	Kopi tubruk, Milk shake choco oreo	NOTA0000020 NOTA0000029	2
75	74	Kopi tubruk, Milk shake choco hazelnut	NOTA0000014 NOTA0000176	2
76	75	Kopi tubruk, Milk shake choco caramel	NOTA0000066 NOTA0000287	2
77	76	Kopi tubruk, Ice shake coklat	NOTA0000037 NOTA0000191 ... NOTA0000467	5
...
2111	2110	-	-	-

Tabel 7 akan melakukan eliminasi pada nilai Scount yang kurang dari minimum *support* (20). Hasilnya akan masuk dalam tabel 18.

Tabel 8 Frequent 2 Itemset

No	Itemset	SCount	Support
1	Kopi tubruk, Kopi Saring Susu	31	6%
2	Kopi tubruk, Kentang + sosis goreng	20	4%
3	Kopi tubruk, Kopi Saring Krim Susu	24	5%
...
14	Sosis goreng, Kopi Saring Krim Susu	34	7%
15	Kentang + sosis goreng, Kopi Saring Krim Susu	22	5%
16	Ice lemon tea, Nasi goreng spesial	20	4%

Tabel 8 merupakan hasil dari itemset yang sudah lebih atau sama dengan minimum *support*. Selanjutnya 2 itemset akan di kombinasikan menjadi 3 itemset.

Tabel 9 Mencari Address 3 Itemset

No	Itemset	Formula	hasil
----	---------	---------	-------

1

	Kopi tubruk, Kopi saring susu, Kentang sosis goreng	$(1 \times 4356 + 12 \times 66 + 22) \bmod 4223$	947
2	Kopi tubruk, Kopi saring susu, Kopi saring krim susu	$(1 \times 4356 + 12 \times 66 + 23) \bmod 4223$	948
3	Kopi tubruk, Kopi saring susu, Pisbak coklat keju	$(1 \times 4356 + 12 \times 66 + 43) \bmod 4223$	968
4	Kopi tubruk, Kopi saring susu, Es teh manis	$(1 \times 4356 + 12 \times 66 + 2) \bmod 4223$	927
...
110	Kentang sosis goreng, Pisbak coklat keju, Nasi goreng spesial	$(22 \times 4356 + 43 \times 66 + 60) \bmod 4223$	1601
...
192	Es teh manis, Kentang goreng, Ice shake red velvet	$(2 \times 4356 + 20 \times 66 + 15) \bmod 4223$	1601
...
219	Sosis goreng, Ice shake thai tea latte, Ice lemon tea	$(21 \times 4356 + 17 \times 66 + 55) \bmod 4223$	3970
220	Ice shake red velvet, Ice shake thai tea latte, Ice lemon tea	$(15 \times 4356 + 17 \times 66 + 55) \bmod 4223$	3172

Tabel 9 pada hasil masih terdapat *collosion* salah satunya terjadi pada baris 5 dan 121 maka nilai mod nya berubah dengan rumus (2.5). $2111 \times 2 + 1 = 4223$. Terjadi 4 kali *collosion* sehingga nilai mod menjadi 4223, 8447, 16895, 33791.

Tabel 10 Loop 4 Mencari 3 Itemset

No	Itemset	Formula	hasil
1	Kopi tubruk, Kopi saring susu, Kentang sosis goreng	$(1 \times 4356 + 12 \times 66 + 22) \bmod 33791$	5170
2	Kopi tubruk, Kopi saring susu, Kopi saring krim susu	$(1 \times 4356 + 12 \times 66 + 23) \bmod 33791$	5171
3	Kopi tubruk, Kopi saring susu, Pisbak coklat keju	$(1 \times 4356 + 12 \times 66 + 43) \bmod 33791$	5191
...
219	Sosis goreng, Ice shake thai tea latte, Ice lemon tea	$(21 \times 4356 + 17 \times 66 + 55) \bmod 33791$	25071
220	Ice shake red velvet, Ice shake thai tea latte, Ice lemon tea	$(15 \times 4356 + 17 \times 66 + 55) \bmod 33791$	32726

Tabel 10 sudah tidak terdapat *collosion* sehingga hasil dapat masuk kedalam tabel hash. Hasil dari tabel ini akan masuk kedalam address pada tabel 21.

Tabel 11 Hash 3 Itemset

No	Address	Itemset	Link	Scount
1	-	-	-	-
...
151	150	Kopi saring krim susu, Kentang goreng, Ice shake red velvet	NOTA0000122	1
152	151	-	-	-
153	152	Kopi saring krim susu, Kentang goreng, Ice shake thai tea latte	NOTA0000032 NOTA0000475 NOTA0000499	3
...
157	156	Kopi saring krim susu, Kentang goreng, Sosis goreng	NOTA0000003 NOTA0000016 ... NOTA0000524	30
...

33784	33783	Kopi saring krim susu, Ice shake thai tea latte, Ice lemon tea		0
...
33791	33790	-	-	-

Tabel 11 akan mengeliminasi nilai SCount yang kurang dari 2, hasilnya akan masuk kedalam tabel frequent 3 itemset.

Tabel 12 Frequent 3 Itemset

No	Itemset	SCount	Support
1	Kopi saring krim susu, Kentang goreng, Sosis goreng	30	6%
2	Kopi saring susu, Kentang goreng, Sosis goreng	24	5%

Tabel 12 merupakan hasil dari itemset yang sudah bebas dari minimum *support*. Selanjutnya itemset akan dikombinasikan menjadi 4 itemset seperti pada tabel 23 dibawah ini.

Tabel 13 Mencari Address 4 Itemset

No	Itemset	Formula	hasil
1	Kopi saring krim susu, Kentang goreng, Sosis goreng, Kopi saring susu	$(23 \times 287496 + 20 \times 4356 + 21 \times 66 + 12) \bmod 33791$	10308

Tabel 13 hasil pola kombinasi 4 item hanya menghasilkan 1 itemset dan tidak terdapat *collosion*, hasil dari tabel 23 menjadi address pada tabel 24 yang berfungsi untuk menunjukkan *link* item yang di transaksikan.

Tabel 14 Hash 4 Itemset

No	Address	Itemset	Link	Scount
1	0	-	-	0
...
10309	10308	Kopi saring krim susu, Kentang goreng, Sosis goreng, Kopi saring susu	-	0
...
16894	16893	-	-	-
16895	16894	-	-	-

Tabel 14 karena hasil itemset tidak terdapat pada seluruh transaksi dibulan januari maka tabel frequent 4 itemset tidak terbentuk. Untuk menentukan nilai *confidence* menggunakan rumus (2.3) pada halaman 8, dengan nilai *support* pada tiap iterasi. Ditentukan bahwa nilai minimum *confidence* yaitu 50% maka nilai kurang dari minimum *confidence* akan tereliminasi.

Tabel 15 Confidence 2 kombinasi

Itemset { A U B }	<i>support</i> { A U B }	<i>support</i> { A }	Confidence
Sosis goreng THEN Kentang goreng	10%	12%	83.33%
Pisbak coklat keju THEN Kopi tubruk	7%	10%	70%
Kentang goreng THEN Sosis goreng	10%	17%	58.82%
Sosis goreng THEN Kopi saring krim susu	7%	12%	58.33%
Es teh manis THEN Nasi goreng spesial	4%	7%	57.14%

Sosis goreng THEN Kopi saring susu	6%	12%	50%
Ice lemon tea THEN Nasi goreng spesial	4%	8%	50%

Tabel 15 menjelaskan 2 kombinasi itemset dengan hasil nilai lebih besar sama dengan minimum *confidence*.

Tabel 16 Confidence 3 Kombinasi

Itemset { A U B }	<i>support</i> { A U B }	<i>support</i> { A }	Confidence
Sosis goreng, Kopi saring susu THEN Kentang goreng	5%	6%	83.33%
Kopi saring susu, Kentang goreng THEN Sosis goreng	5%	7%	71.43%
Kentang goreng, Sosis goreng THEN Kopi saring krim susu	6%	10%	60%
Kentang goreng, Sosis goreng THEN Kopi saring susu	5%	10%	50%

Tabel 16 menjelaskan 3 kombinasi itemset dengan hasil nilai yang diatas minimum *confidence*.

4 Kesimpulan

Berdasarkan pembahasan penelitian pada 3 bulan periode yang telah di tentukan yaitu bulan januari, april, juli, dan oktober. Bahwa menu yang diminati dengan pola asosiasi 3 itemset If Sosis goreng Then Kopi saring krim susu Then Kentang goreng memiliki nilai *confidence* terdapat pada bulan juli dan oktober 85.71% kemudian pola kombinasi yang tidak diminati seperti If Kentang goreng Then Sosis goreng Then Kopi saring susu hanya terdapat pada bulan januari 50%.

Untuk pola asosiasi 2 itemset yang diminati If Sosis goreng Then Kentang goreng memiliki nilai *confidence* terdapat pada bulan januari 83.33% kemudian juli 64.71% lalu bulan april 62.70% dan oktober 58.82% dan untuk yang tidak diminati Kentang goreng Then Kopi saring krim susu memiliki nilai *confidence* terdapat pada bulan juli 52.63% kemudian oktober 50% kemudian april 47.37% dan januari 47.06%. Pembentukan pola kombinasi 3 itemset dengan itemset yang tidak diminati seperti If Sosis goreng Then Kentang goreng Then Cappucino pada pola kombinasi ini ditambahkan item Cappucino dengan nilai support sebesar 12% dan pembentukan pola kombinasi 2 itemset dengan itemset yang tidak diminati seperti If Kopi saring susu Then Kentang goreng Then Kopi saring krim susu Then Chicken bento pada pola kombinasi ini ditambahkan item Chiken bento dengan nilai support sebesar 11% diantara periode 3 bulan.

5 Referensi

- [1] Herawati, Fajar, A. 2013. Data Mining. (hlm. 3). Yogyakarta: CV ANDI OFFSET.
- [2] Aguru, Sirisha. Rao, B., M. 2014. A Hash Based Frequent Itemset Mining using Rehashing. India: Sri Vasavi Engineering Collage.
- [3] Ramadhan, Farha. 2017. Implementasi Algoritma Hash Based Terhadap Aturan Asosiasi untuk Menentukan Frequent Itemset Study Kasus Rumah Makan Seafood "Kita". Yogyakarta: STMIK AMIKOM.
- [4] Prasetyo, Eko. 2012. Data Mining Konsep dan Aplikasi menggunakan MATLAB. (hlm. 2 dan 313-315). Yogyakarta: CV ANDI OFFSET.
- [5] Sitanggang, LF. Dkk. 2014. Aplikasi Data Mining Association Rule Dengan Algoritma Fp-Growth Untuk Mengenal Pola Pembelian Pelanggan Pada Toko Kue (Studi Kasus: L'cheese Factory)6 Politeknik Caltex Riau.
- [6] Han, J, Kamber, M, Pei, J 2011, Data mining concepts and techniques, (ed. 3), Morgan Kaufmann, USA.
- [7] Aritonang, P 2012, Implementasi data mining dengan association rule dalam pengambilan keputusan untuk korelasi pengambilan produk menggunakan algoritma apriori, Skripsi, Universitas Sumatera Utara.