

# IMPLEMENTASI ALGORITMA HASH BASED TERHADAP ATURAN ASOSIASI UNTUK MENENTUKAN FREQUENT ITEMSET STUDY KASUS RUMAH MAKAN SEAFOOD “KITA”

Farha Ramadhan

Teknik Informatika STMIK AMIKOM Yogyakarta  
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281  
Email : farha8227@students.amikom.ac.id

## Abstrak

*Pada saat ini, teknik data mining dengan aturan asosiasi sudah banyak digunakan untuk menganalisa pola pembelian. Dengan memanfaatkan data – data transaksi pembelian yang kemudian diolah untuk menggali informasi berharga dari kumpulan data transaksi tersebut. Dari informasi yang tergalil tersebut kemudian dapat dijadikan menjadi suatu aturan untuk membuat kebijakan untuk meningkatkan penjualan.*

*Salah satu algoritma yang sering digunakan untuk asosiasi adalah algoritma apriori. Namun algoritma tersebut memiliki kelemahan dalam hal performa. Karena pada setiap menentukan frequent k-itemset harus melakukan scan pada database. Dan hal tersebut akan menjadi masalah serius apabila kandidat k-itemset memiliki jumlah sangat besar. Melakukan scan pada database yang besar akan memakan waktu yang banyak.*

*Algoritma hash based bisa menjadi solusi untuk mengatasi masalah dalam menentukan frequent k-itemset dari kandidat k-itemset dengan jumlah yang besar. Dengan menggunakan teknik hashing, kandidat k-itemset yang telah discan akan dimasukkan kedalam bucket pada tabel hash. Dari bucket tersebut akan digunakan untuk mencari frequent (k+1)-itemset sehingga scan database hanya dilakukan 1 kali pada iterasi pertama. Pada penilitan ini akan dijelaskan bagaimana algoritma hash bashed bekerja.*

**Kata kunci:** Data Mining, Asosiasi, Frequent Itemset, Algoritma Hash Based.

## 1. Pendahuluan

Jumlah rumah makan di Yogyakarta pada tahun 2011 menurut Badan Pusat Statistik Provinsi D.I.Y berjumlah 585 hingga pada tahun 2013 bertambah menjadi 745[1]. Bisnis kuliner merupakan bisnis yang cukup beresiko karena makanan merupakan barang yang tidak dapat bertahan lama. Sehingga para pengelola rumah makan dituntut untuk pintar dalam manajemen penyediaan bahan makanan agar mendapatkan keuntungan yang optimal. Adanya tuntutan tersebut mengharuskan para pengelola rumah makan untuk mencari cara untuk menganalisa kebutuhan konsumen terhadap lauk pauk yang disediakan. Salah satunya yaitu

dengan cara memanfaatkan teknologi informasi yang dapat memberikan informasi terkait pola pembelian konsumen.

Sekarang ini kita telah mengenal suatu teknik yang berguna dalam mengolah data dalam jumlah besar yang dikenal dengan data mining. Data mining digunakan untuk menggali pengetahuan dengan cara melakukan analisis pada data dalam jumlah yang sangat besar. Dengan menggunakan teknik pemodelan yang canggih. Mengkonversi data – data yang kurang berguna menjadi data yang bernilai yang berisi pengetahuan serta informasi[2]

Model data mining terdiri dari kumpulan aturan dan persamaan yang dapat digunakan untuk mengidentifikasi pola dari suatu kumpulan data dan juga untuk melakukan prediksi. Data Mining dikelompokkan menjadi 2 model berdasarkan tujuannya, yaitu Prediktif dan Deskriptif.[2] Pada penelitian ini menggunakan data mining dengan model prediktif. Model Deskriptif adalah model data mining yang hanya mempunyai field masukan tanpa mempunyai field keluaran. Pengenalan pola dilakukan secara langsung tidak terarahkan berdasarkan suatu atribut.[2]

Salah satu aturan pada model deskriptif adalah aturan asosiasi, yaitu aturan yang tidak melibatkan prediksi secara langsung dari suatu field tunggal. Semua field mempunyai peran ganda, yaitu menjadi masukan dan keluaran dalam waktu yang sama. Asosiasi digunakan untuk menemukan keterkaitan antara peristiwa, produk, dan atribut.[2] Asosiasi dikenal sebagai salah teknik data mining yang menjadi dasar bagi teknik data mining yang lainnya. Salah satu tahap asosiasi yang menarik yaitu analisis pola frekuensi tinggi. Analisis pola frekuensi tinggi adalah tahap untuk mencari itemset yang memenuhi syarat minimum dari nilai support count dalam database. Untuk menentukan itemset yang memenuhi minimum support count dari kandidat itemset dalam jumlah besar pada awal iterasi merupakan faktor yang mendominasi performa data mining secara keseluruhan.[3].

Algoritma yang sering digunakan untuk proses mining pada aturan asosiasi adalah algoritma apriori. Namun algoritma tersebut memiliki kelemahan ketika

menentukan frequent itemset dari kandidat itemset dalam jumlah besar. Semakin besar kandidat itemset maka akan semakin banyak melakukan iterasi dan pada algoritma apriori harus melakukan scan database setiap kali iterasi sehingga memerlukan waktu yang cukup lama.[4]

Algoritma hash based dapat menjadi solusi untuk menentukan frequent itemset dari kandidat itemset secara optimal. Algoritma ini mengurangi jumlah kandidat k-itemset pada awal. Jumlah itemset pada C2 yang dihasilkan menggunakan teknik hashing dapat menjadi kecil sehingga scan database yang dilakukan untuk menentukan L2 menjadi lebih efisien.[5]

Tahap pertama dari teknik Hashing adalah penelusuran pada data untuk melakukan perhitungan support count dari setiap kandidat k-itemset dan pada saat yang sama teknik ini mencari informasi (k+1)-itemset dengan menggunakan proses hash terhadap semua kemungkinan yang hasilnya akan disimpan pada tabel hash. Informasi yang disimpan pada tabel hash akan dimasukkan ke dalam bucket – bucket dimana pada setiap bucket akan menyimpan support count dari setiap itemset yang di hash ke dalam bucket tersebut. Selanjutnya dilakukan eliminasi untuk bucket yang tidak memenuhi minimum support.

**Tujuan**

Penelitian ini bertujuan untuk :

1. Mengetahui kelebihan algoritma hash based dibandingkan dengan algoritma apriori
2. Menentukan frequent itemset dari data rumah makan seafood “Kita” menggunakan algoritma hash based

**Batasan Penelitian**

Adapun pada penelitian ini mempunyai batasan yaitu:

1. Pada penelitian ini hanya sebatas pada proses untuk mencari frequent itemset
2. Variabel yang digunakan adalah item – item yang dibeli pada transaksi dan pada penelitian ini adalah menu makanan.

**Metode Penelitian**

Penelitian untuk mengimplementasikan algoritma hash based terdiri dari 4 tahap, yaitu :

1. Mencari referensi – referensi terkait dengan pengimplementasian data mining dan algoritma hash based. Referensi yang digunakan berupa buku, jurnal, paper, dan skripsi.
2. Mengumpulkan data transaksi dari rumah makan seafood “Kita” dengan mengambil sampel data transaksi sebanyak 10 transaksi. Data transaksi yang dikumpulkan terdiri dari 2 atribut, yaitu no transaksi dan makanan.
3. Melakukan preprocessing pada data transaksi yang dikumpulkan. Merubah value pada atribut makanan menjadi single value. Contoh, terdapat record dengan no transaksi 1 dan pada

atribut makanan berisi 3 value. Record tersebut akan dipecah menjadi 3 record dengan 1 value pada atribut makanan.

4. Melakukan Implementasi algoritma Hash Based dengan data yang sudah di preprocessing

**2. Pembahasan**

Data transaksi dikumpulkan dari rumah makan seafood dengan mengambil sampel sebanyak 10 data. Data transaksi ini sudah dilakukan preprocessing.

**Tabel 1. Data Transaksi**

Id Transaksi	Itemset
T1	Cumi, Nasi
T2	Udang, Capcay, Nasi
T3	Udang, Cumi, Nasi
T4	Kerang, Udang, Nasi
T5	Cumi, Mie
T6	Nasi, Capcay
T7	Sop, Nasi
T8	Cumi, Kerang, Udang, Nasi
T9	Udang, Ayam, Bihun
T10	Kwetiaw, Bihun

Selanjutnya dilakukan proses hashing terhadap kandidat 1-itemset untuk memasukkan setiap itemset kedalam bucket pada tabel hash. Proses hashing menggunakan rumus[5] :

$$h(x) = (\text{order of item } x) \bmod n \dots(1)$$

h = alamat bucket pada tabel hash

n = banyak alamat (n = 2m + 1, m = jumlah keseluruhan item)

**Tabel 2. Order of Item**

Item	Order
Cumi	1
Nasi	2
Udang	3
Capcay	4
Kerang	5
Mie	6
Sop	7
Bihun	8
Kwetiaw	9

Mencari alamat pada tabel hash untuk 1-itemset

- h (Cumi) = (1) mod 19 = 1
- h (Nasi) = (2) mod 19 = 2
- h (Udang) = (3) mod 19 = 3
- h (Capcay) = (4) mod 19 = 4
- h (Kerang) = (5) mod 19 = 5
- h (Mie) = (6) mod 19 = 6
- h (Sop) = (7) mod 19 = 7
- h (Bihun) = (8) mod 19 = 8
- h (Kwetiaw) = (9) mod 19 = 9

**Tabel 3. Tabel Hash 1-itemset**

Address	Itemset	Link	Support Count
0			
1	Cumi	T1,T3,T5,T8	4
2	Nasi	T1,T2,T3,T4,T6,T7,T8	7
3	Udang	T2,T3,T4,T8,T9	5
4	Capcay	T2,T6	2
5	Kerang	T4,T8	2
6	Mie	T5,T10	2
7	Sop	T7	1
8	Bihun	T9	1
9	Kwetiaw	T10	1

Setelah dilakukan penghitungan hash makan akan dihasilkan data seperti pada Tabel 2. Itemset menempati Alamat hash sesuai dengan hasil setelah dilakukan perhitungan menggunakan rumus hashing. Alamat – alamat yang ditempati oleh itemset akan menjadi node dan membuat link yang mengarah kepada transaksi – transaksi yang mengandung itemset tersebut secara berurutan sehingga terbentuk link list.

Dengan menentukan minimum support count = 2 dihasilkan frequent 1-itemset seperti pada Tabel 3.

**Tabel 4. Frequent 1-itemset**

Itemset	Support Count
Cumi	4
Nasi	7
Udang	5
Capcay	2
Kerang	2
Mie	2

Untuk langkah selanjutnya adalah mencari Frequent 2-itemset. Sebelumnya, ditentukan terlebih dahulu semua kemungkinan 2-itemset yang terbentuk dari Frequent 1-itemset. Kandidat 2-itemset yang

terbentuk dari frequent 1-itemset yaitu, (Cumi, Nasi), (Cumi, Udang), (Cumi, Capcay), (Cumi, Kerang), (Cumi, Mie), (Nasi, Udang), (Nasi, Capcay), (Nasi, Kerang), (Nasi, Mie), (Udang, Capcay), (Udang, Kerang), (Udang, Mie), (Capcay, Kerang), (Capcay, Mie), (Kerang, Mie). Selanjutnya memasukkan 2-itemset tersebut kedalam tabel hash dengan menggunakan rumus[5] hash berikut :

$$h(k)=((\text{order of } x) * 10 + \text{order of } y) \bmod n \dots(2)$$

Mencari alamat pada tabel hash untuk 2-itemset

- h(Cumi, Nasi) = ((1) \* 10 + 2) mod 19 = 12
- h(Cumi, Udang) = ((1) \* 10 + 3) mod 19 = 13
- h(Cumi, Capcay) = ((1) \* 10 + 4) mod 19 = 14
- h(Cumi, Kerang) = ((1) \* 10 + 5) mod 19 = 15\*
- h(Cumi, Mie) = ((1) \* 10 + 6) mod 19 = 16\*
- h(Nasi, Udang) = ((2) \* 10 + 3) mod 19 = 4
- h(Nasi, Udang) = ((2) \* 10 + 4) mod 19 = 5
- h(Nasi, Udang) = ((2) \* 10 + 5) mod 19 = 6
- h(Nasi, Mie) = ((2) \* 10 + 6) mod 19 = 7\*
- h(Udang, Capcay) = ((3) \* 10 + 4) mod 19 = 15\*
- h(Udang, Kerang) = ((3) \* 10 + 5) mod 19 = 16\*
- h(Udang, Mie) = ((3) \* 10 + 6) mod 19 = 17
- h(Capcay, Kerang) = ((4) \* 10 + 5) mod 19 = 7\*
- h(Capcay, Mie) = ((4) \* 10 + 6) mod 19 = 8
- h(Kerang, Mie) = ((5) \* 10 + 6) mod 19 = 18

Pada perhitungan hashing tersebut ditemukan adanya collision(Adanya lebih dari 1 itemset yang memiliki alamat hash sama). Pada kasus ini collision terjadi pada alamat ke-7 untuk (Nasi, Mie) dengan (Capcay, Kerang), alamat ke-15 untuk (Cumi, Kerang) dengan (Udang, Capcay), dan alamat ke-16 untuk (Cumi, Mie) dengan (Udang, Kerang).

Ketika sebuah collision terjadi maka harus segera dilakukan pengecekan untuk mencari alamat bucket yang masih kosong. Jika setelah dilakukan pengecekan ditemukan indikasi bahwa tabel hash sudah separuhnya terisi maka perlu dilakukan rehashing ulang dengan banyak alamat 2 kali dari banyak alamat sebelumnya. Untuk menyelesaikan masalah tersebut digunakan rumus[5] yang berbeda yaitu :

$$h(k)=((\text{order of } x) * 10 + \text{order of } y) \bmod j \dots(3)$$

j = banyak alamat setelah dilakukan penambahan (j = 2 \* m + 1, m = jumlah alamat pada tabel hash sebelum dilakukan penambahan )

$$h(\text{Cumi, Nasi}) = ((1) * 10 + 2) \bmod 39 = 12$$

$$\begin{aligned}
 h(\text{Cumi, Udang}) &= ((1) * 10 + 3) \bmod 39 = 13 \\
 h(\text{Cumi, Capcay}) &= ((1) * 10 + 4) \bmod 39 = 14 \\
 h(\text{Cumi, Kerang}) &= ((1) * 10 + 5) \bmod 39 = 15 \\
 h(\text{Cumi, Mie}) &= ((1) * 10 + 6) \bmod 39 = 16 \\
 h(\text{Nasi, Udang}) &= ((2) * 10 + 3) \bmod 39 = 23 \\
 h(\text{Nasi, Capcay}) &= ((2) * 10 + 4) \bmod 39 = 24 \\
 h(\text{Nasi, Kerang}) &= ((2) * 10 + 5) \bmod 39 = 25 \\
 h(\text{Nasi, Mie}) &= ((2) * 10 + 6) \bmod 39 = 26 \\
 h(\text{Udang, Capcay}) &= ((3) * 10 + 4) \bmod 39 = 34 \\
 h(\text{Udang, Kerang}) &= ((3) * 10 + 5) \bmod 39 = 35 \\
 h(\text{Udang, Mie}) &= ((3) * 10 + 6) \bmod 39 = 36 \\
 h(\text{Capcay, Kerang}) &= ((4) * 10 + 5) \bmod 39 = 6 \\
 h(\text{Capcay, Mie}) &= ((4) * 10 + 6) \bmod 39 = 7 \\
 h(\text{Kerang, Mie}) &= ((5) * 10 + 6) \bmod 39 = 17
 \end{aligned}$$

Setelah dilakukan penambahan alamat tabel hash, tidak ditemukan lagi adanya collision antar itemset. Setiap alamat terisi dengan 1 itemset. (Nasi, Mie) menempati alamat ke-26 dan (Capcay, Kerang) menempati alamat ke-6. (Cumi, Kerang) menempati alamat ke-15 dan (Udang, Capcay) menempati alamat ke-34. (Cumi, Mie) menempati alamat ke-16 dan (Udang, Kerang) menempati alamat ke-35. Tabel Hash 2 – Itemset dapat dilihat pada Tabel 4.

**Tabel 5. Tabel Hash 2 – Itemset**

Address	Itemset	Link	Support Count
0	-	-	-
..			
6	Capcay, Kerang	-	0
7	Capcay, Mie	-	0
..			
12	Cumi, Nasi	T1,T3,T8	3
13	Cumi, Udang	T3,T8	2
14	Cumi, Capcay	-	0
15	Cumi, Kerang	T8	1
16	Cumi, Mie	T5	1
17	Kerang, Mie	-	0
..			
23	Nasi, Udang	T2,T3,T4,T8	4
24	Nasi, Capcay	T2,T6	2
25	Nasi, Kerang	T4,T8	2
26	Nasi, Mie	-	-
..			
34	Udang, Capcay	T2	1
35	Udang, Kerang	T4,T8	2
36	Udang, Mie	-	0
..			
38	-	-	-

Dari tabel hash tersebut akan dilakukan eliminasi itemset terhadap alamat bucket dengan support count < 2 untuk mendapatkan frequent 2-itemset.

**Tabel 6. Frequent 2-itemset**

Itemset	Support Count
Cumi, Nasi	3
Cumi, Udang	2
Nasi, Udang	4
Nasi, Capcay	2
Nasi, Kerang	2
Udang, Kerang	2

Setelah mendapatkan frequent 2-itemset seperti pada tabel 5, langkah selanjutnya adalah mencari frequent 3-itemset dengan terlebih dahulu mencari semua kemungkinan 3 – itemset yang terbentuk. 3 – Itemset yang terbentuk yaitu : (Cumi, Nasi, Udang), (Cumi, Nasi, Capcay), (Cumi, Nasi, Kerang), (Cumi, Udang, Kerang), (Nasi, Udang, Capcay), (Nasi, Udang, Kerang), (Nasi, Capcay, Kerang). Untuk memasukkan 3 – itemset tersebut ke dalam tabel hash digunakan rumus[5] :

$$H(k) = ((\text{order of X}) * 100 + (\text{order of Y}) * 10 + \text{order of Z}) \bmod j \dots (4)$$

Mencari alamat pada tabel hash untuk 3 itemset :

$$\begin{aligned}
 H(\text{Cumi, Nasi, Udang}) &= ((1) * 100 + (2) * 10 + 3) \bmod 39 = 6 \\
 H(\text{Cumi, Nasi, Capcay}) &= ((1) * 100 + (2) * 10 + 4) \bmod 39 = 7 \\
 H(\text{Cumi, Nasi, Kerang}) &= ((1) * 100 + (2) * 10 + 5) \bmod 39 = 8 \\
 H(\text{Cumi, Udang, Kerang}) &= ((1) * 100 + (3) * 10 + 5) \bmod 39 = 18 \\
 H(\text{Nasi, Udang, Capcay}) &= ((2) * 100 + (3) * 10 + 4) \bmod 39 = 0 \\
 H(\text{Nasi, Udang, Kerang}) &= ((2) * 100 + (3) * 10 + 5) \bmod 39 = 1 \\
 H(\text{Nasi, Capcay, Kerang}) &= ((2) * 100 + (4) * 10 + 5) \bmod 39 = 11
 \end{aligned}$$

**Tabel 7. Tabel Hash 3 – Itemset**

Address	Itemset	Link	Support Count
0	Nasi, Udang, Capcay	T2	1
1	Nasi, Udang, Kerang	T4, T8	2
..			
6	Cumi, Nasi, Udang	T3, T8	2
7	Cumi, Nasi, Capcay	-	0
8	Cumi, Nasi, Kerang	T8	1

	Kerang		
..			
11	Nasi, Capcay, Kerang	-	-
..			
18	Cumi, Udang, Kerang	T8	1
..			
38	-	-	-

**Tabel 8. Frequent 3 – Itemset**

Itemset	Support Count
Nasi, Udang, Kerang	2
Cumi, Nasi, Udang	2

Setelah dilakukan 3 kali iterasi tersisa 2 alamat pada tabel hash dengan 3-itemset. Didapatkan frequent itemset untuk 3-itemset yaitu (Nasi, Udang, Kurang) dengan support count = 2 dan (Cumi, Nasi, Udang) dengan support count = 2.

### 3. Kesimpulan

Dari penelitian yang dilakukan didapatkan kesimpulan :

1. Algoritma hash based menjadi solusi bagi kelemahan algoritma apriori didalam menentukan frequent itemset. Tahap dalam menentukan frequent itemset merupakan tahap yang paling berpengaruh terhadap performa data mining, terutama jika kandidat itemset berjumlah sangat besar. Pada algoritma hash based hanya dilakukan 1 kali scan database yaitu pada awal iterasi. Untuk selanjutnya, itemset tersebut akan dimasukan kedalam bucket pada tabel hash menggunakan operasi hash dengan ketentuan setiap alamat bucket diisi oleh 1 itemset. Dengan begitu, performa data mining dapat meningkat karena scan database hanya dilakukan sekali pada awal iterasi sehingga tidak lagi dilakukan scan database pada setiap mencari frequent k-itemset. Dan pada setiap iterasi langsung dilakukan eliminasi terhadap alamat bucket yang tidak memenuhi minimum support count.
2. Dari 10 sampel data didapatkan 2 itemset yang memenuhi minimum support count setelah dilakukan iterasi hingga 3-itemset. 2 itemset tersebut adalah (Nasi, Udang, Kerang) dengan 2 support count dan (Cumi, Nasi Udang) dengan 2 support count.

### Daftar Pustaka

- [1] BPS Provinsi D. I. Yogyakarta, "Jumlah Biro Perjalanan, Pramuwisata, Restoran, dan Rumah Makan di D.I. Yogyakarta", *Daerah Istimewa Dalam Angka 2014*, Yogyakarta:BPS D. I. Yogyakarta, 2014, pp. 242.
- [2] Tsipsis, Konstantinos and Antonios Chorianopoulos, "Data Mining Techniques in CRM. Inside Customer Segmentation", WILEY, 2009, pp. 2 – 4.

- [3] Park JS, Chen MS, Yu PS (1995): "An effective hash-based algorithm for mining association rules". In: *Proceeding of the 1995 ACM-SIGMOD international conference on management of data (SIGMOD95)*, San Jose, CA, pp 175-186.
- [4] Pratama, Heruandika Cahyono, Martaleli Bettiza, and Tekad Matulatan, "PENERAPAN ALGORITMA APRIORI DALAM MENEMUKAN HUBUNGAN DATA AWAL MASUK MAHASISWA DENGAN PRESTASI AKADEMIK (STUDI KASUS : STAI Miftahul Ulum Tanjungpinang)", Fakultas Teknik UMRAH
- [5] Aguru, Sirisha, and Batteri Madhava Rao, "A Hash Based Frequent Itemset Mining using Rehashing", *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, issue 12, pp. 4201 – 4203. Desember 2014.

### Biodata Penulis

**Farha Ramadhan**, Saat ini menjadi mahasiswa di STMIK AMIKOM Yogyakarta.

