

IMPLEMENTASI ALGORITMA APRIORI POSITIF NEGATIF DAN *BINARY HAMMING DISTANCE* UNTUK REKOMENDASI KULINER MAHASISWA UNIVERSITAS SEBELAS MARET SURAKARTA

Belladonna Shelly Agasti¹⁾ Ristu Saptono.²⁾ Hasan Dwi Cahyono³⁾

¹⁾ Jurusan Informatika Universitas Sebelas Maret
Jl. Ir. Sutami No-36 A, Ketingan, Surakarta

email : belladonashag@gmail.com, ristu.saptono@staff.uns.ac.id, hasandc@staff.uns.ac.id,

ABSTRACT

Students often feel confused in choosing restaurants to eat around the college. Students have many considerations to choose. From these consideration, we can be found an association patterns that can be used to make an appropriate restaurant recommendation for students. In the process of making a recommendation, we applied two algorithms association rule Apriori positive negative and Binary Hamming Distance. After applying this algorithm, of the seven criteria obtained from students interview (price, taste, portion, service, cleanliness, convenience, and choice of menu) we gained 10 association rule, but the rule which is used as the main rule is a rule that has a single antecedent with much consequent, namely portion to give effect to the price, taste, and service with a confidence value of 0.72 and minimal support 0.4. Three criteria are not prioritized in application development is the cleanliness, convenience, and menu options. Meaning that the user may choose to fill out or ignore. Binary hamming distance will match user input score with restaurant score and calculate its similarity. Restaurant with the nearest distance will be recommended to user.

Key words

Recommendation, culinary, college, Apriori, Binary Hamming Distance

1. Pendahuluan

Seiring dengan meningkatnya mobilitas masyarakat, jumlah penyedia jasa kuliner juga semakin berkembang. Tempat-tempat kuliner ini menjadi alternatif bagi orang-orang yang tidak punya waktu untuk makan di rumah. Setiap orang memiliki kriteria sendiri untuk memilih tempat makan, demikian juga dengan mahasiswa. Mahasiswa biasanya memilih makan di lingkungan sekitar

kampus, saat jeda di siang hari. Saat menentukan tempat makan, seringkali mahasiswa mengalami kebingungan. Hal ini disebabkan karena banyaknya pilihan tempat makan yang ada. Khususnya, mahasiswa senior juga sudah mulai jenuh dengan pilihan tempat makan yang sama. Padahal sebenarnya ada banyak tempat makan baru yang belum pernah mereka kunjungi. Mahasiswa yang kos juga mengalami kebosanan dan kesulitan dalam memilih tempat makan. Pada malam hari biasanya juga sedikit tempat makan yang masih buka, dengan menu yang kurang bervariasi. Akibatnya mahasiswa yang kos juga cukup kesulitan mencari tempat untuk makan malam, karena pilihan yang terbatas.

Meskipun saat ini sudah banyak *website* dan aplikasi yang menampilkan tentang informasi tempat kuliner, namun *website* dan aplikasi kuliner yang ada saat ini, kebanyakan hanya mampu menampilkan harga, lokasi, dan jenis makanan yang disajikan tanpa memperhatikan faktor lain yang menjadi pertimbangan *customer* khususnya mahasiswa. *Website* kuliner yang tersedia saat ini mungkin cenderung menampilkan tempat-tempat makan terkenal yang harganya kurang masuk jangkauan mahasiswa, belum lagi kebanyakan tempatnya berada jauh dari lingkungan kampus.

Berdasarkan hal yang telah dijelaskan diatas, dibutuhkan suatu algoritma terapan untuk membuat sistem rekomendasi yang dapat membantu mahasiswa memilih tempat makan sesuai dengan kriteria yang diinginkan mahasiswa itu sendiri. Dalam pengembangan aplikasi ini diimplementasikan algoritma Apriori positif negatif dan *Binary Hamming Distance*. Algoritma Apriori termasuk jenis aturan asosiasi atau *association rule mining* pada *data mining*. Algoritma Apriori adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu database yang memiliki frekuensi atau *support* di atas ambang batas tertentu yang disebut dengan istilah

minimum support dan *minimum confidence* atau disebut juga *threshold* [1]. Untuk menemukan kriteria-kriteria yang akan digunakan dalam sistem ini, dilakukan suatu wawancara mendalam dengan mahasiswa Universitas Sebelas Maret dari sembilan fakultas, laki-laki dan perempuan, baik yang kos maupun tidak. Dari hasil wawancara mendalam tadi dapat diketahui pertimbangan apa saja yang sering disebutkan mahasiswa dalam memilih tempat makan. Kemudian dari banyak pertimbangan tadi dicari pola asosiasinya dengan menggunakan algoritma Apriori positif dan negatif.

Selain itu untuk memproses rekomendasi digunakan algoritma *Binary Hamming Distance* untuk *matching process* antara hasil input user dengan tempat makan yang tersedia di *database*. *Binary Hamming Distance* merupakan bagian dari *Hamming Distance*. Algoritma ini adalah salah satu algoritma yang digunakan untuk mengukur kedekatan item. Jika nilai kedekatan makin kecil, maka kedua item itu semakin dekat dan berlaku sebaliknya [2]. Penggabungan dua metode ini akan membantu dalam menyampaikan rekomendasi tempat makan untuk mahasiswa.

2. Dasar Teori

2.1 Kaidah Asosiasi

Algoritma asosiasi merupakan suatu bentuk algoritma dalam *data mining* yang memberikan informasi hubungan antar item di basis data. Terdapat 2 persoalan dalam menggali kaidah asosiasi, yaitu:

1. Mencari semua *frequent itemset*, yaitu itemset (kumpulan item) yang memenuhi ketentuan nilai *minimum support*.
2. Mencari kaidah asosiasi yang memenuhi ketentuan nilai *minimum confidence*.

Kaidah asosiasi $A \rightarrow B$ dibaca "jika A maka B". Disini A berlaku sebagai *antecedent* dan B berlaku sebagai *consequent*. *Support* dan *confidence* adalah dua ukuran yang menunjukkan kepastian dan tingkat kegunaan suatu kaidah yang ditemukan. Pada umumnya, kaidah asosiasi yang ditemukan dikatakan menarik apabila kaidah tersebut memenuhi ketentuan nilai *minimum support* dan *minimum confidence* yang telah ditentukan.

Support (*sup*) adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu barang atau itemset dari keseluruhan transaksi. *Confidence* (*conf*) adalah suatu ukuran yang menunjukkan hubungan kondisional antar barang, misalnya seberapa sering barang B dibeli oleh orang yang membeli barang A. Yang dimaksud dengan *min_sup* adalah ketentuan nilai *minimum support* yang harus dimiliki oleh kaidah asosiasi, sedangkan *min_conf*

adalah ketentuan nilai *minimum confidence* yang harus dimiliki oleh kaidah asosiasi

$$\text{sup}(A \rightarrow B) = P(A, B) \quad (1)$$

$$\text{conf}(A \rightarrow B) = P(B|A) \quad (2)$$

$$= \frac{P(A, B)}{P(A)}$$

Kaidah asosiasi positif (*Positive Association Rule/PAR*) merujuk pada kaidah asosiasi antara item-item yang ada dalam transaksi. Misal, kaidah asosiasi yang menunjukkan barang apa saja yang dibeli secara bersamaan oleh pembeli. Kaidah jenis ini diwujudkan sebagai $A \rightarrow B$. Berikut ini adalah karakteristik yang dimiliki oleh PAR: [4]

1. $A \cap B = \emptyset$
2. $\text{sup}(A \rightarrow B) = \text{sup}(A, B) \geq \text{min_sup}$
3. $\text{conf}(A \rightarrow B) = \frac{\text{sup}(A, B)}{\text{sup}(A)} \geq \text{min_conf}$

Kaidah asosiasi negatif (*Negative Association Rule/NAR*) merujuk pada kaidah asosiasi antara item yang muncul dalam transaksi dengan item yang tidak muncul secara bersamaan. Misal, kaidah asosiasi yang menunjukkan customer yang membeli barang A tidak membeli B. Kaidah jenis ini diwujudkan sebagai $\{(\neg A \rightarrow B), (A \rightarrow \neg B)\}$. Meskipun kaidah $(\neg A \rightarrow \neg B)$ bisa terjadi, kaidah ini ekuivalen dengan PAR ($B \rightarrow A$) sehingga bentuk $(\neg A \rightarrow \neg B)$ tidak dianggap sebagai NAR [3]

PAR ($A \rightarrow B$) menjadi dasar dari NAR $\{(\neg A \rightarrow B), (A \rightarrow \neg B)\}$. Suatu transaksi t dikatakan memiliki support terhadap $(\neg A \rightarrow B)$ jika $A \notin t$ dan $B \in t$. Sebuah NAR $\{A\} \rightarrow \neg\{B, C\}$ memiliki arti bahwa kemunculan A dalam suatu transaksi mengindikasikan bahwa kemungkinan item B dan C tidak muncul. Dengan demikian, suatu transaksi yang mengandung item A dan B, tanpa adanya item C dikatakan memiliki support terhadap kaidah asosiasi tersebut.

Nilai *support* dan *confidence* dari NAR dihitung dengan menggunakan PAR yang terkait. $\text{sup}(\neg A) = 1 - \text{sup}(A)$. Dengan berpedoman pada hal ini, dapat diketahui cara penghitungan $\text{sup}(A \rightarrow \neg B)$ sebagai berikut:

$$\text{sup}(A \rightarrow \neg B) = \text{sup}(A, \neg B) = \text{sup}(A) - \text{sup}(A, B) \text{ dengan } A, B \subset I$$

Berikut ini adalah karakteristik yang dimiliki oleh NAR: (Wu et al., 2004)

1. $A \cap B = \emptyset$

- $$\sup(A) \geq \min \sup \geq, \sup(B) \geq$$
- $$\min \sup \geq, \text{ dan } \sup(A, B) < \min _sup$$
- 2.
 3. $\sup(A \rightarrow B) = \sup(A, \neg B) \geq \min_sup$
 4. $conf(A \rightarrow B) = \frac{\sup(A, \neg B)}{\sup(A)} \geq \min_conf$

2.2 Algoritma Apriori

Algoritma Apriori termasuk jenis aturan asosiasi atau *association rule mining* pada *data mining* [1]. Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi item. Algoritma ini diajukan oleh R. Agrawal dan R. Srikant tahun 1994. Algoritma Apriori adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu *database* yang memiliki frekuensi atau *support* di atas ambang batas tertentu yang disebut dengan istilah *minimum support* dan *minimum confidence* atau disebut juga *threshold* (Han, 2006). *Threshold* adalah batas minimum transaksi. Jika jumlah transaksi kurang dari *threshold* maka item atau kombinasi item tidak akan diikutkan perhitungan selanjutnya. Penggunaan *threshold* dapat mempercepat perhitungan. Algoritma Apriori melakukan pendekatan *iterative* yang dikenal dengan pencarian *level-wise*, dimana k-itemset digunakan untuk mengeksplorasi (k+1)-itemset. Algoritma Apriori dibagi menjadi beberapa tahap yang disebut iterasi. Tiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama dimulai dari pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Algoritma Apriori dalam menangani masalah asosiasi dengan mengurangi jumlah set item yang dipertimbangkan. Pengguna menetapkan *minimum support*. Jadi nilai $(A \cup C) \leq$

minimum support untuk setiap association rule $A \rightarrow C$

atau $A \rightarrow C \leq \text{minimum support}$.

Algoritma Apriori pertama menghasilkan semua himpunan item yang memenuhi minimum dukungan. himpunan item tersebut sering disebut *frequent itemset*. Dengan diberikan nilai *minimum support* dan *minimum confidence*, pencarian aturan-aturan asosiasi dengan menggunakan algoritma Apriori ada dua tahap (Han, 2006):

1. Join (penggabungan).
Pada proses ini setiap item dikombinasikan dengan item yang lainnya sampai tidak terbentuk kombinasi lagi.
2. Prune (pemangkasan).
Pada proses ini, hasil dari item yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan minimum support yang telah ditentukan oleh user.

Algoritma Apriori dibagi menjadi beberapa tahap yang disebut iterasi.

1. Pembentukan kandidat itemset, kandidat k-itemset dibentuk dari kombinasi (k-1)-itemset yang didapat dari iterasi sebelumnya. Satu ciri dari algoritma Apriori adalah adanya pemangkasan kandidat k-itemset yang subsetnya yang berisi k-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.
2. Perhitungan support dari tiap kandidat k-itemset. Support dari tiap kandidat kitemset di dapat dengan men-scan database untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k-itemset tersebut. Ini juga merupakan ciri dari algoritma Apriori dimana diperlukan perhitungan dengan scan seluruh database sebanyak k-itemset terpanjang.
3. Tetapkan pola frekuensi tinggi. Pola frekuensi tinggi yang memuat k item atau k-itemset ditetapkan dari kandidat k-itemset yang support-nya lebih besar dari minimum support.
4. Bila tidak didapat pola frekuensi tinggi maka seluruh proses dihentikan. Bila tidak, maka k tambah satu dan kembali ke bagian 1.

2.3 Binary Hamming Distance

Algoritma *Binary Hamming Distance* adalah bagian dari algoritma *Hamming distance*. *Hamming distance* adalah salah satu algoritma mengukur kedekatan item. Jika nilai kedekatannya makin kecil, maka kedua item itu semakin dekat dan berlaku sebaliknya. Yang biasanya dibandingkan adalah kata dan bilangan biner. Algoritmanya sebagai berikut :

1. Cek panjang kedua kata atau biner yang akan dibandingkan. Panjang keduanya harus sama. Jika tidak sama, maka perhitungan tidak bisa dilakukan. Misalkan ada contoh sebagai berikut:

$$\begin{array}{l} \text{salam} \\ \text{salim} \end{array} \text{ dan } \begin{array}{l} \underline{110101} \\ \underline{100011} \end{array}$$

Gambar 1 Contoh perbandingan Binary Hamming Distance

Dari kedua contoh dilakukan analisa panjang kata dan biner :

- a. Panjang kata salam berjumlah 5 dan panjang kata salim juga memiliki jumlah 5. Karena panjangnya sama, kedua kata ini bisa dibandingkan.
 - b. Panjang bilangan biner 110101 berjumlah 6 dan panjang 100011 juga 6. Karena panjangnya sama, maka kedua bilangan biner ini bisa dibandingkan.
2. Pilih salah satu kata atau biner sebagai pembanding. Dari contoh pada langkah 1 :
- a. Pada kata salam dan salim, dipilih kata salam sebagai pembanding
 - b. Pada bilangan biner 110101 dan 100011, dipilih bilangan biner 110101 sebagai pembanding.
 - c. Cek setiap karakter yang posisinya sama. Jika karakter kata atau biner yang dibandingkan berbeda dengan kata atau biner pembanding, maka memiliki nilai kedekatan 1. Nilai kedekatan ini akan bertambah untuk tiap karakter kata atau biner yang dibandingkan berbeda dengan karakter kata atau biner pembanding. Jika karakter kata atau biner yang dibandingkan sama dengan karakter kata atau biner pembanding, maka memiliki nilai kedekatan 0.
 - d. Kata salam bisa dipecah menjadi s.a.l.a.m dan kata s.a.l.i.m bisa dipecah menjadi s.a.l.i.m. Bila kedua kata itu dibandingkan menurut posisi hurufnya:

Tabel 1 Nilai Kedekatan Kata

SALAM	SALIM	Status	Nilai Kedekatan
S	S	Sama	0
A	A	Sama	0
L	L	Sama	0
A	I	Beda	1
M	M	Sama	0
TOTAL			1

- e. Bilangan biner 110101 dapat dipecah menjadi 1.1.0.1.0.1 dan bilangan biner 100011 dapat dipecah menjadi 1.0.0.0.1.1. Bila kedua bilangan biner itu dibandingkan menurut posisi binernya :

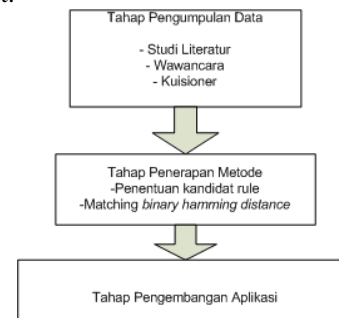
Tabel 2 Nilai kedekatan bilangan biner

110101	100011	Status	Nilai Kedekatan
1	1	Sama	0
1	0	Beda	1
0	0	Sama	0
1	0	Beda	1
0	1	Beda	1
1	1	Sama	0
TOTAL			3

Maka, total nilai kedekatan kata salam dan salim adalah 1 dan total nilai kedekatan bilangan biner 110101 dan 100011 adalah 3. Jadi, kata salam dan salim itu sangat mirip sedangkan bilangan biner 110101 dan 100011 cukup mirip

3. Metode Penelitian

Metode penelitian yang digunakan dalam melakukan implementasi metode ini dapat dilihat melalui alur kerja sebagai berikut:



Gambar 2 Alur metode penelitian

3.1 Tahap Pengumpulan Data

Dalam tahap pengumpulan data ini dilakukan studi literatur, wawancara, dan penilaian kuisisioner. Studi literatur untuk mendapatkan dasar-dasar teori yang berhubungan dengan sistem rekomendasi dan algoritma apriori positif negatif melalui jurnal, buku, karya ilmiah, dan sumber-sumber ilmiah lainnya.

Proses wawancara dilakukan untuk menemukan kriteria-kriteria yang digunakan mahasiswa dalam memilih tempat makan. Wawancara ini dilakukan kepada 30 mahasiswa Universitas Sebelas Maret dari sembilan fakultas, laki-laki dan perempuan, baik yang kos maupun tidak, serta dari angkatan 2010, 2011, 2012, dan 2013. Dari hasil wawancara mendalam tadi dapat diketahui pertimbangan apa saja yang sering disebutkan mahasiswa dalam memilih tempat makan.

Setelah proses wawancara, didapatkan kriteria-kriteria apa saja yang biasanya menjadi pertimbangan mahasiswa dalam memilih tempat makan. Kriteria-kriteria ini kemudian digunakan sebagai parameter untuk menilai 30 tempat makan yang biasa dikunjungi mahasiswa melalui kuisisioner. Kuisisioner yang telah dibuat tadi disebarkan ke mahasiswa Universitas Sebelas Maret melalui Google Form dan dapat diketahui penilaian mahasiswa terhadap 30 tempat makan tersebut. Dari hasil jawaban dari pertanyaan-pertanyaan diatas akan dilakukan analisa pada tahap selanjutnya untuk mendapatkan detail yang akan digunakan pada tahap penerapan metode.

3.2 Tahap Penerapan Metode

Pada tahap ini dilakukan penerapan dua metode yang akan digunakan, yaitu algoritma Apriori positif negatif dan *Binary Hamming Distance*. Dari hasil wawancara telah didapatkan beberapa kriteria yang akan digunakan untuk membuat rule. Namun, tidak semua kriteria digunakan. Kriteria yang dipilih adalah kriteria yang sering disebutkan responden. Kemudian kriteria yang terpilih ini diproses dengan algoritma Apriori untuk mendapatkan rule yang memenuhi *min_conf*. Kriteria-kriteria tersebut diberi nilai *true* dan *false* (1 dan 0). Kriteria yang terpilih ini nantinya akan digunakan untuk membentuk *rule* dengan algoritma Apriori. Setelah itu dipilih rule-rule dengan nilai *confidence* dan *min_sup* yang memenuhi untuk digunakan dalam sistem.

Algoritma *Binary Hamming Distance* digunakan untuk mengukur kedekatan antar item. Dalam penelitian ini, *binary hamming distance* digunakan untuk mengukur kedekatan kriteria yang diinputkan user dengan kriteria yang dimiliki suatu tempat makan. Kriteria yang dimiliki tempat makan didapatkan sebelumnya dari proses pengisian kuisioner. Karena menggunakan binary, maka nilai tiap kriteria berupa 0 dan 1. Setelah kriteria diinputkan oleh user, kriteria tersebut diproses dengan algoritma Apriori. Setelah didapatkan *score* untuk input user, kemudian dilakukan *matching* dengan *score* tempat makan yang tersedia menggunakan *binary hamming distance*. Tempat makan yang memiliki jarak paling dekat dengan input kriteria akan direkomendasikan ke user.

3.3 Tahap Pengembangan Aplikasi

Dalam tahap ini dilakukan proses penerapan metode yang digunakan ke dalam program. Dalam penerapan metode ke dalam aplikasi dibutuhkan suatu strategi pengembangan. Strategi pengembangan ini dibuat supaya aplikasi lebih *user-friendly* dan meminimalisir jumlah klik sebelum mendapatkan rekomendasi yang sesuai. Strategi ini khususnya dilakukan dalam proses menampilkan pertanyaan untuk user agar metode yang diterapkan dapat bekerja secara maksimal.

4. Hasil dan Pembahasan

Setelah dilakukan wawancara mendalam dengan mahasiswa Universitas Sebelas Maret, ditemukan tujuh kriteria yang sering disebutkan oleh mahasiswa ketika memilih tempat makan, yaitu harga, rasa, kebersihan, kenyamanan, pelayanan, pilihan menu, dan porsi, seperti ditunjukkan dalam tabel berikut:

Tabel 3 Kriteria memilih tempat makan

No.	Kriteria
1.	Harga
2.	Rasa
3.	Kebersihan
4.	Kenyamanan
5.	Pelayanan
6.	Pilihan Menu
7.	Porsi

Dari tujuh kriteria tadi, diujikan kepada 30 mahasiswa Universitas Sebelas Maret yang juga berpartisipasi dalam wawancara tadi, kemudian dari jawaban yang diberikan, dicari asosiasinya menggunakan algoritma Apriori positif negatif dengan nilai *min support* 0.4 dan *confidence* 0.7. Pemilihan *min support* dan *confidence* ini berdasarkan pada jumlah rule yang terbentuk dan rule yang dihasilkan, karena tidak ada aturan tertentu dalam menentukannya. Nilai *support* 0.4 dan *confidence* 0.72 dianggap menghasilkan rule yang menarik dan memiliki nilai *confidence* yang tinggi seperti ditunjukkan pada tabel 4.

Tabel 4 Rule Confidence

Rule	Rule Confidence
2	0.72
3	0.82
4	0.92
5	1
6	0.8125
9	0.76
10	0.76
11	0.86
13	0.8125
14	0.72

Setiap rule yang ditemukan di Tabel 4 memiliki *antecedent* dan *consequence*. *Antecedent* dalam Tabel 5 ditunjukkan dalam kolom Rule Condition, sedangkan *consequence* yang merupakan akibat dari *antecedent* ditunjukkan dalam kolom Rule Effect. Jadi setelah dilakukan pencarian pola asosiasi, dapat ditemukan suatu relasi antara pertimbangan-pertimbangan yang dimiliki mahasiswa.

Tabel 5 Rule Effect dan Rule Condition

Rule	Rule Condition	Score	Rule Effect	Score
2	HARGA	1	PELAYANAN	0
	RASA	1	PORSI	1
3	HARGA	1	RASA	1
	PELAYANAN	0	PORSI	1
4	HARGA	1	RASA	1
	PELAYANAN	0		
	PORSI	1		
5	HARGA	1	PORSI	1
	RASA	1		
	PELAYANAN	0		
6	HARGA	1	RASA	1
	PORSI	1	PELAYANAN	0
9	RASA	1	HARGA	1
	PELAYANAN	0	PORSI	1
10	RASA	1	HARGA	1
	PORSI	1	PELAYANAN	0
11	RASA	1	HARGA	1
	PELAYANAN	0		
	PORSI	1		
13	PELAYANAN	0	HARGA	1
	PORSI	1	RASA	1
14	PORSI	1	HARGA	1
			RASA	1
			PELAYANAN	0

Dari tabel 5 didapatkan 10 relasi antar kriteria yang akan diterapkan dalam pengembangan aplikasi. Kemudian untuk membuat suatu sistem rekomendasi yang menarik digunakan suatu strategi agar user tidak terlalu banyak melakukan klik, namun hasil yang didapatkan tetap efektif.

Dipilihlah satu *single condition* yang dapat memberikan banyak *effect*. Dari tabel 5, condition yang sesuai adalah rule 14 dimana jika user mempertimbangkan porsi dalam memilih tempat makan, maka otomatis dia akan mencari tempat makan dengan harga terjangkau, rasa enak, namun tidak mempertimbangkan faktor pelayanan. Sehingga untuk tampilan pertanyaan yang akan pertama muncul adalah pertanyaan mengenai porsi. Jika user menjawab “ya” maka tiga pertanyaan yaitu harga, rasa, dan pelayanan tidak akan ditampilkan karena sudah dijawab secara otomatis dengan aturan yang telah ditemukan sebelumnya, sehingga *user* hanya perlu menjawab tiga pertanyaan lagi, itu pun jika user tidak neglected. Tiga pertanyaan lain yaitu kebersihan, kenyamanan, dan pilihan menu sifatnya adalah opsional dan boleh dijawab ataupun tidak tergantung dari user.

Ketika user memilih untuk menjawab pertanyaan mengenai porsi dengan “no”, artinya dia tidak mempertimbangkan porsi dari suatu tempat makan, maka akan ditampilkan tiga antecedent lain yaitu, harga, rasa, dan pelayanan. Kemudian akan ditampilkan tempat makan

yang mendekati jawaban user. Berikut ini dalam Gambar 3 adalah daftar tempat makan dengan skor untuk masing-masing kriteria secara berurutan.

NAMA	HARGA	RASA	KEBERSIHAN	KENYAMANAN	PELAYANAN	PILIHAN MENU	PORSI
BALE ENAK	1	1	0	0	0	0	0
MIE AYAM JAMUR	1	1	0	0	1	0	0
WARUNG BU WARNI	1	1	0	0	1	1	1
RICA-RICA ISI	1	1	0	0	1	1	1
SPECIAL SAMBAL	0	1	1	1	0	1	0
MR. MESEM	1	1	1	1	1	0	1
ROAD DIMSUM	0	1	1	1	1	0	0
HAIKIE SOUP	1	1	1	1	0	0	0
WARUNG BU PUR	1	0	1	0	1	1	1
AYAM GEPREK CUMLOT	1	1	1	0	1	0	1
CHICKEN HUT	1	1	0	0	0	0	1
ARJIE'S KITCHEN	0	1	1	1	0	1	0
KEDAI CONEL	1	1	1	1	1	1	1
POPIPOP	0	1	1	1	0	1	0
SIMPLYCOOK CAFE	0	1	1	1	0	1	0
WARUNG TIADA TARA "TT"	1	1	1	1	0	1	1
NAGOYA JAPANESE FUSION	0	1	1	1	0	0	0
OLIVE CHICKEN	1	1	1	0	1	0	0
MOEN-MOEN STEAK	0	1	1	1	0	1	0
POPEYE CHICKEN EXPRESS	1	1	1	0	1	1	1
BHESUS	1	1	1	1	0	0	0
MIE AYAM SONY	1	1	1	0	0	0	0
MIE SUMATRA	0	1	1	1	0	0	1
KANTIN DANAU PERTANIAN	1	1	1	1	1	1	1
AYAM GORENG BLITAR	0	1	1	0	0	0	0
BANYUMASAN SURYA	1	1	0	0	1	1	1
MANG OMAN	1	1	1	0	0	1	0
SBC SPESIAL KANGKUNG	1	1	1	0	0	1	0

Gambar 3 Skoring tempat makan

Setelah didapatkan input dari user, maka akan dilakukan matching skor tempat makan menggunakan Binary Hamming Distance. Proses dari awal user menggunakan sistem adalah seperti skenario berikut:

1. User diberi pertanyaan apakah mempertimbangkan porsi yang banyak dalam memilih tempat makan.
2. Jika user menjawab “ya” maka di database disimpan nilai 1 untuk kriteria porsi, karena porsi telah memberikan efek ke tiga kriteria lain, maka otomatis tiga kriteria lain sudah dijawab dengan nilai harga 1, rasa 1, dan pelayanan 0. Kemudian disimpan ke database.
3. Sedangkan jika user menjawab “tidak” maka di database disimpan nilai 0 untuk porsi, dan akan ditampilkan tiga *antecedent* lain yang harus diisi oleh *user*.
4. Sebelumnya di database sudah terdapat nilai *default* yang dibuat 0 untuk tiga kriteria yang menjadi consequence, yaitu kebersihan, kenyamanan, dan pilihan menu. Sehingga walaupun user baru menjawab empat pertanyaan tadi (porsi, harga, rasa, dan pelayanan), namun aplikasi sudah dapat memberikan hasil rekomendasi karena tiga kriteria yang belum dijawab sudah diberi nilai.
5. Jika hasil rekomendasi yang diberikan belum sesuai dengan keinginan user, maka user dapat menambahkan nilai untuk kriteria kebersihan,

kenyamanan, atau, pilihan menu yang diletakkan di *checkbox*.

6. Hasil input akan dimatch dengan skor yang dimiliki 30 tempat makan dengan metode Binary Hamming Distance seperti pada tabel 2. Kemudian dapat diketahui nilai kedekatannya. Tempat makan yang memiliki nilai kedekatan paling kecil akan direkomendasikan ke user.
7. Hasil input user yang sudah menggunakan aplikasi ini nantinya akan disimpan untuk kemudian diolah di pengembangan lebih lanjut untuk proses *learning* sistem guna membentuk rule baru.

Transactions on Information Systems, Volume 22, pp. 381-405.

4. Kesimpulan dan Saran

Setelah melakukan penelitian dapat disimpulkan bahwa:

1. Dari 10 *rule* yang dibentuk, rule yang digunakan sebagai *main rule* adalah *rule* yang memiliki *single antecedent* namun memberikan banyak *consequence*, yaitu porsi yang memberikan efek untuk harga, rasa, dan pelayanan dengan nilai confidence 0.72.
2. Tiga kriteria yang tidak diprioritaskan dalam pengembangan aplikasi adalah kebersihan, kenyamanan, dan pilihan menu. Artinya user boleh memilih untuk mengisi atau mengabaikan.
3. Binary hamming distance dapat dipadukan bersama algoritma Apriori untuk mengukur nilai kedekatan nilai input user dengan nilai suatu tempat makan.

Adapun saran untuk penelitian lanjut bagi kasus ini adalah menggunakan jumlah data responden dan data tempat makan yang lebih banyak. Lingkup studi juga dapat diperluas wilayahnya sehingga dapat memberi rekomendasi kepada masyarakat umum lainnya. Selain itu mungkin dapat ditambahkan algoritma lain untuk menambah keefisienan sistem.

5. REFERENSI

- [1] Han, J., 2001. *Data Mining : Concepts and Techniques*. s.l.:Morgan Kaufmann Publishers.
- [2] Putro, L. S., 2013. Penerapan Kombinasi Algoritma Minhash dan Binary Hamming Distance pada Social-Content Filtering Rekomendasi Lagu. *Universitas Sebelas Maret Surakarta*, pp. 12-15.
- [3] Ramasubbareddy, B., Dr.A.Govardhan & Dr.A.Ramamohanreddy, 2010. *Mining Positive and Negative Association Rules*. Hefei, China, s.n.
- [4] Wu, X., Zhang, C. & Zhang, S., 2004. Efficient Mining of Both Positive and Negative Association Rule. *ACM*