IMPLEMENTASI METODE ASSOCIATION RULE PADA APLIKASI PARIWISATA BANGKA BELITUNG DI PLATFORM ANDROID

Nio Somalo[1]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : niosomalo22@gmail.com[1]

ABSTRAK

Provinsi Bangka Belitung adalah provinsi kepulauan yang terdiri dari dua buah pulau utama yaitu pulau Bangka dan pulau Belitung. Provinsi ini menyimpan berbagai potensi wisata, seperti wisata bahari, sejarah dan kuliner. Namun sedikitnya informasi tentang tempat wisata membuat provinsi Bangka Belitung belum terlalu dikenal oleh wisatawan lokal maupun mancanegara. Berdasarkan hasil wawancara sulitnya mendapatkan informasi dan rekomendasi tempat wisata menjadi salah satu faktor sedikitnya wisatawan yang datang ke provinsi Bangka Belitung.

Salah satu cara untuk mengatasi permasalahan tersebut adalah dengan menggunakan data mining. Teknik data mining yang dapat digunakan untuk menemukan rekomendasi tempat wisata dan aktivitas terkait adalah *association rule*, yaitu sebuah teknik *data mining* untuk menemukan sebuah hubungan korelasi atau pola yang menarik dari sebuah data yang besar. Salah satu algoritma yang dapat digunakan untuk mencari suatu pola yang menarik adalah algoritma *apriori*.

Pembangunan sistem babel kite dengan menggunakan algoritma *apriori* mampu mengahasilkan rekomendasi tempat wisata dan aktivitas terkait. Berdasarkan hasil pengujian *black box*, *beta* dan *data mining* mampu menghasilkan rekomendasi tempat wisata.

Kata kunci*:* apriori, data mining, association rule, android

1 PENDAHULUAN

Informasi mengenai tempat wisata merupakan hal yang dibutuhkan oleh wisatawan. Dengan mendapatkan informasi lebih, terhadap suatu tempat wisata akan lebih memudahkan bagi para wisatawan untuk dapat melakukan persiapan dan kebutuhan lainnya. Berdasarkan hasil wawancara dengan wisatawan yang pernah berkunjung ke provinsi Bangka Belitung, ditemukan beberapa fakta tentang masalah yang sering dialami ketika berkunjung ke tempat wisata. Masalah yang sering terjadi adalah para wisatawan sulit mendapatkan informasi mengenai tempat wisata terdekat, informasi tempat penginapan, dan informasi layanan transportasi yang tersedia untuk mengunjungi tempat wisata tersebut. Hal ini disebabkan karena sedikitnya informasi dari dinas provinsi Bangka Belitung maupun dari agen perjalanan.

Fenomena lain yang terjadi yaitu wisatawan mengalami kebingungan dalam menentukan kegiatan apa yang bisa dilakukan pada suatu tempat wisata atau pada beberapa tempat wisata, karena ada tipe wisatawan yang tidak mencari tempat tapi aktivitas. Berdasarkan hasil wawancara wisatawan menyatakan bahwa kebingungan ini disebabkan karena belum tersedia media yang bisa membantu wisatawan untuk memberitahu kira-kira ada aktivitas apa yang bisa dilakukan pada suatu tempat wisata, wisatawan biasanya hanya melihat dari kebiasaan orang-orang yang ada di dalam tempat wisata tersebut, sehingga wisatawan merasa kesulitan untuk mencari media-media informasi dan biasanya wisatawan hanya melakukan dengan cara bertanya. Dari fenomena tersebut diperlukan rekomendasi kegiatan atau aktivitas.

Rekomendasi kegiatan dan tempat wisata dapat diberikan kepada wisatawan jika sudah ada riwayat dari wisatawan lain serta melihat pola keterkaitan antara kegiatan dan tempat wisata yang sering dilakukan oleh para wisatawan sebelumnya. Untuk dapat menentukan keputusan yang tepat dalam merekomendasikan wisata dan kegiatan kepada wisatawan, maka diperlukan suatu metode untuk dapat mencari keterkaitan antara tempat wisata dan kegiatan yang sering dilakukan oleh wisatawan yaitu dengan metode *association rule*.

1.1 Association Rule

Association rule merupakan suatu prosedur untuk mencari hubungan antar-*item* dalam suatu *dataset* [1]. Association rule (aturan asosiasi) akan menemukan pola tertentu untuk mengasosiasikan data yang satu dengan data yang lain. Untuk mencari association rule dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari frequent itemset terlebih dahulu. Frequent itemset adalah sekumpulan item yang sering muncul secara bersamaan. Setelah semua pola frequent itemset ditemukan, barulah

mencari aturan assosiatif atau aturan keterkaitan yang memenuhi syarat yang telah ditentukan [1].

Jika diasumsikan bahwa barang yang dijual di swalayan adalah semesta, maka setiap barang akan memiliki boolean variabel yang akan menunjukkan keberadaannya atau tidak barang tersebut dalam satu transaksi atau satu keranjang belanja. Pola boolean yang didapat digunakan untuk menganalisa barang yang dibeli secara bersamaan. Pola tersebut dirumuskan dalam sebuah association rule. Sebagai contoh konsumen biasanya akan membeli komputer dan *software* antivirus pada saat yang sama yang ditujukkan sebagai berikut [1]:

Komputer → *software* antivirus [*support* = 2, *confidence* = 60%]

Association rule diperlukan suatu variabel ukuran yang ditentukan sendiri oleh *user* untuk menentukan batasan sejauh mana atau sebanyak apa *output* yang diinginkan *user*. *Support* dan *confidence* adalah sebuah ukuran kepercayaan dan kegunaan suatu pola yang telah ditemukan. Nilai *support* 2 menunjukkan bahwa keseluruhan dari total transaksi konsumen membeli komputer dan *software* antivirus secara bersamaan yaitu sebanyak 2. Sedangkan *confidence* 60% yaitu menunjukkan bila konsumen membeli komputer dan pasti membeli *software* antivirus sebesar 60% [1].

1.1.1. Algoritma Apriori

Algoritma *apriori* adalah salah satu algoritma yang melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule*. Algortima *apriori* menggunakan pengetahuan frekuensi atribut yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Pada algoritma *apriori*, menentukan kandidat yang mungkin muncul dengan cara memperhatikan *minimum support* dan *minimum confidence*. *Support* adalah nilai pengunjung atau persentase kombinasi sebuah item dalam database [1].

Nilai *support* sebuah *item* diperoleh dengan rumus sebagai berikut: [1]

$$Support = (A \cap B) = \frac{\sum \text{Jumlah Transaksi yang Mengandung A dan B}}{\sum \text{Total Transaksi}} * 100\% \quad (2.1)$$

Sedangkan *confidence* adalah nilai kepastian yaitu kuatnya hubungan antar *item* dalam sebuah apriori. *Confidence* dapat dicari setelah pola frekuensi munculnya sebuah item ditemukan. Rumus untuk menghitung *confidence* adalah sebagai berikut [1]:

$$Confidence = \frac{\sum \text{Jumlah transaksi yang mengandung A dan B}}{\sum \text{Transaksi yang mengandung A}} * 100\% \quad (2.2)$$

Proses utama yang dilakukan dalam algoritma *apriori* untuk mendapat

frequent itemset yaitu [1]:

1. *Join* (Penggabungan)

Proses ini dilakukan dengan cara pengkombinasian item dengan item yang lainnya hingga tidak dapat terbentuk kombinasi lagi.

2. *Prune* (Pemangkasan)

Proses pemangkasan yaitu hasil dari item yang telah dikombinasikan kemudian dipangkas dengan menggunakan minimum support yang telah ditentukan oleh *user*. Pada iterasi ke-k akan ditemukan semua itemset yang memiliki k item, disebut dengan k-itemset. Tiap iterasi terdiri dari dua tahap, yaitu :

a. Gunakan *frequent* (k-1) itemset untuk membangun kandidat *frequent* k- *itemset*.

b. Gunakan *scan database* dan pencocokan pola untuk mengumpulkan hitungan untuk kandidat *itemset*.

Adapun langkah-langkah proses perhitungan *association rule* dengan algoritma *apriori* sebagai berikut [1]:

1. Sistem men-*scan database* untuk mendapat kandidat 1-*itemset* (himpunan *item* yang terdiri dari 1 *item*) dan menghitung nilai *support*-nya. Kemudian nilai *support*-nya tersebut dibandingkan dengan minimum *support* yang telah ditentukan. Jika nilainya lebih besar atau sama dengan *minimum support*, maka *itemset* tersebut termasuk *large itemset*.
2. *Itemset* yang tidak termasuk dalam *large itemset* tidak diikutkan dalam iterasi selanjutnya (di pangkas).
3. Pada iterasi kedua, sistem akan menggunakan hasil *large itemset* pada iterasi pertama (L1) untuk membentuk kandidat *itemset* kedua (L2). Pada iterasi selanjutnya sistem akan menggunakan hasil *large itemset* pada iterasi sebelumnya (Lk-1) untuk membentuk kandidat itemset berikut (Lk). Sistem akan menggabungkan (*join*) Lk-1 dengan Lk-1 untuk mendapatkan Lk. Seperti pada iterasi sebelumnya sistem akan menghapus (memangkas) kombinasi *itemset* yang tidak termasuk dalam *large itemset*.
4. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru hasil proses *join* tersebut dihitung *support*-nya.
5. Proses pembentuk kandidat yang terdiri dari proses penggabungan dan pemangkasan akan terus dilakukan hingga himpunan kandidat *itemset*-nya *null*, atau sudah tidak ada lagi kandidat yang akan terbentuk.
6. Setelah itu, dari hasil *frequent itemset* tersebut dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.
7. Pada pembentukan associaton rule, nilai yang sama dianggap sebagai satu nilai.
8. Association rule yang terbentuk harus memenuhi nilai minimum yang telah ditentukan.
9. Untuk setiap large itemset L, cari himpunan bagian L yang tidak kosong. Untuk setiap himpunan bagian

tersebut, dihasilkan rule dengan bentuk aB(L-a) jika support-nya (a) lebih besar dari minimum support.

1.2 Pengujian

Pengujian sistem dilakukan untuk menemukan kekurangan atau kesalahan yang ada dalam sistem yang sedang diuji. Pengujian bermaksud untuk mengetahui sistem yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan. Terdapat tiga buah pengujian yang dilakukan pada penelitian ini, yaitu pengujian *black box*, pengujian *beta*, dan pengujian *acceptance*.

1.2.1 Pengujian *Black box*

Pengujian *black box* dilakukan dengan mengamati fungsional-fungsional yang terdapat dalam sistem dengan tujuan untuk mengetahui kesalahan pada sistem dan mengetahui apakah sistem telah sesuai dengan tujuan yang diharapkan.

1.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif. pengujian dilakukan secara langsung kepada responden selaku pengguna sistem untuk mengetahui sejauh mana kelayakan sistem.
Dengan melakukan pengujian beta, letak kesalahan maupun kekurangan dari sistem dapat diketahui untuk diperbaiki dalam tahap pengembangan yang akan datang. Pengujian ini melalui dua tahap, yaitu demo program, setelah itu pemberian kuesioner.

1.2.3 Pengujian Data Mining

Rencana pengujian *data mining* yang akan dilakukan adalah dengan cara membandingkan hasil rekomendasi tempat wisata dan aktivitas yang dihasilkan dari perhitungan manual menggunakan *microsoft excel* dengan hasil dari perangkat lunak yang akan dibangun. Pengujian ini menggunakan keseleruhan data dengan nilai *minimum support* 2 dan nilai *minimum confidence* 70%.

2 ISI PENELITIAN

2.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang muncul pada pembangunan sistem. Analisis tersebut meliputi analisis masalah, analisis aplikasi sejenis, analisis arsitektur sistem, analisis sumber data, analisis *preprocessing* data, analisis penerapan metode *association rule*, analisis spesifikasi kebutuhan perangkat lunak, analisis kebutuhan non fungsional, analisis kebutuhan fungsional.

2.1.1 Analisis Masalah

Berikut adalah analisis masalah yang timbul dari penelitian ini yaitu:
1. Sulitnya wisatawan mendapatkan informasi fasilitas yang tersedia
2. Sulitnya wisatawan mendapatkan informasi rekomendasi tempat wisata dan kegiatan di tempat wisata oleh wisatawan.

2.1.2 Gambaran Umum Sistem

Secara umum sistem yang dibangun adalah suatu perangkat lunak untuk menemukan pola keterkaitan antara tempat wisata ke tempat wisata lain beserta aktivitasnya. Parameter yang digunakan dalam analisa adalah nilai minimum *support*, nilai minimum *confidence*, pengguna, waktu transaksi, tempat wisata, dan aktivitas.
Metode yang digunakan dalam sistem yang akan dibangun adalah metode *association rule* untuk melakukan *frequent itemset mining*. Sedangkan untuk mencari *frequent itemset* dan menentukan nilai *support* dalam sistem yang akan dibangun menggunakan algoritma apriori.

2.1.3 Analisis Data Selection

Pada penelitian ini akan dilakukan dua atribut yang akan di *mining* yaitu atribut tempat wisata dan aktivitas. Sebelum tahapan *mining* akan dilakukan terlebih dahulu proses data *selection* untuk memisahkan atribut tempat wisata dan aktivitas. Untuk hasil dari proses *selection* atribut tempat wisata dapat dilihat pada Tabel 1

Tabel 1 Data Selection

Timestamp	Nama	Tempat Wisata
11/1/2015 15.59.59	Mochamad riezky nazrudin	Pantai Tanjung Kiras , Pantai Air Anyer, Pantai Batu Bedaun
11/1/2015 16.14.54	Muhammad Iqbal	Alun-Alun Taman Merdeka
11/1/2015 16.32.00	hendry wijaya	Pantai Tanjung Kalian
11/1/2015 16.46.26	Rafiq zafir	Pulau Buku Limau, Vihara Dewi Kwan Yin,
11/1/2015	I Gede Puja Arjana	Kampung Bali

Timestamp	Nama	Tempat Wisata
17.19.05		
11/1/2015 17.39.34	Humaira	Vihara Dewi Kwan Yin, Pantai Buku Limau
11/1/2015 17.50.45	Cepy Setia Nugraha	Pemandian Air Panas Tirta Tapta Pemali
……	……	……

2.1.4 Analisis Data Cleaning

Data *Cleaning* adalah proses menghilangkan *noise*. Pada penelitian ini dilakukan proses *data cleaning* untuk menghilangkan di mana satu transaksi yang hanya memiliki satu item akan dihilangkan. Untuk data tempat wisata sebelum dilakukan proses *cleaning* dapat dilihat pada Tabel 2 . Berikut hasil dari proses *data cleaning* berdasarkan atribut tempat wisata.

Tabel 2 Data Cleaning

Timestamp	Nama	Tempat Wisata
11/1/2015 15.59.59	Mochamad riezky nazrudin	Pantai Tanjung Kiras, Pantai Air Anyer , Pantai Batu Bedaun
11/1/2015 16.46.26	Rafiq zafir	Pulau Buku Limau, Vihara Dewi Kwan Yin,
11/1/2015 17.39.34	Humaira	Vihara Dewi Kwan Yin, Pantai Buku Limau,
11/1/2015 22.17.03	Agung Faishal Faris	Pantai Batu Bedaun , Pantai Batu Belubang, Pantai Mabai
11/1/2015 23.02.10	bolly	Gua Maria, Pantai Batu Bedaun, Pantai Bukit Batu
11/2/2015 14.52.36	Silo	Pantai Tanjung Kalian, Pantai Air Anyer
11/2/2015	Asep Sarifudin	Pantai Romodong, Pantai Air Anyer
14.52.57		
…….	……	…….

Untuk data aktivitas sebelum dilakukan proses *cleaning* dapat dilihat pada Tabel 1. Berikut hasil dari proses *data cleaning* berdasarkan atribut aktivitas:

Timestamp	Nama	Aktivitas
11/5/2015 12.10.56	Bintang Januari Haliri	Mengambil gambar, Makan-makan, Santai, Berenang
11/5/2015 13.04.29	Agus Purwanto	Mengambil gambar, Makan-makan, Santai, Berenang
11/5/2015 13.56.13	Desy	Mengambil gambar, Santai, Berenang
11/5/2015 13.56.51	Sandro	Mengambil gambar, Santai, Berenang
11/5/2015 13.57.23	Husnaisa	Mengambil gambar, Santai, Berenang
11/14/2015 23.16.14	Gea	Mengambil gambar, Makan-makan
11/14/2015 23.16.56	Cicilya	Santai, Berenang
……..	……	…….

2.1.5 Analisis Data Mining

Penerapan metode *association rule* pada penelitian ini akan dilakukan berdasarkan tempat wisata dan aktivitas wisatawan dengan *minimum support* yang digunakan adalah dua, dan untuk *minimum confidence 70%.* Adapun harapan dalam menentukan *minimum support* dua dan *minimum confidence* 70% adalah agar terbentuknya *rule* yang efesien.

2.1.5.1 Tempat Wisata

Adapun tahapan proses pengerjaan algoritma *apriori* dalam penelitian ini berdasarkan atribut tempat wisata adalah sebagai berikut:

1. *Generate* 1-*itemset Frequent Pattern*

Langkah pertama adalah men-*scan* data tempat wisata yang ada pada Tabel 2 untuk mendapatkan kandidat 1-*itemset* dan menghitung nilai *support count* dari setiap tempat wisata yang ada. Berikut hasil proses *scan* yang dilakukan:

Tabel 3 Kandidat 1-*itemset*

Tempat Wisata	Support Count
Pantai Tanjung Kiras	2
Pantai Air Anyer	4
Pantai Batu Bedaun	6
Pulau Buku Limau	4
Vihara Dewi Kwan Yin	3
Pantai Batu Belubang	2
Pantai Mabai	4
……	….

Hilangkan (*prune*) data yang memiliki *support count* kurang dari *minimum support* yang telah ditentukan. Dapat dilihat pada Tabel 3 terdapat calon kandidat 1-*itemset* yang memiliki *support count* kurang dari *minimum support*, sehingga perlu dihilangkan. Berikut hasil kandidat yang memenuhi *minimum support*:

Tabel 4 Large 1-*itemset*

Tempat Wisata	Support Count
Pantai Tanjung Kiras	2
Pantai Air Anyer	5
Pantai Batu Bedaun	6
Vihara Dewi Kwan Yin	3
Pantai Batu Belubang	2
Pantai Mabai	4
Gua Maria	2
…….	…….

2. *Generate* 2-*itemset Frequent Pattern*
Selanjutnya untuk menemukan *frequent pattern* 2-*itemset* atau L2 gabungkan (*join)* setiap *item* yang ada pada Tabel 4,kemudian hitung *support count*-nya sehingga akan didapatkan hasil sebagai berikut:

Tabel 5 Kandidat 2-*itemset*

Tempat Wisata	Support Count
Pantai Tanjung Kiras, Pantai Air Anyer	1
Pantai Tanjung Kiras, Pantai Batu Bedaun	2
Pantai Tanjung Kiras, Vihara Dewi Kwan Yin	0
Pantai Tanjung Kiras, Pantai Batu Belubang	0
Pantai Tanjung Kiras, Pantai Mabai	0
Pantai Tanjung Kiras, Gua Maria	0
Pantai Tanjung Kiras, Pantai Tanjung Kalian	0
……	……

Setelah *support count* dari setiap kombinasi 2-*itemset* didapat, maka diketahui kombinasi tempat wisata mana saja yang tidak memenuhi *minimum support*, sehingga harus dihilangkan (*prune)*. Berikut hasil kombinasi 2-*itemset* yang memenuhi aturan *minimum support*:

Tabel 6 Large 2-*itemset*

Tempat Wisata	Support Count
Pantai Tanjung Kiras, Pantai Batu Bedaun	2
Pantai Air Anyer, Pantai Tanjung Kalian	3
Pantai Batu Bedaun, Pantai Batu Belubang	2
Pantai Batu Bedaun, Pantai Mabai	2
Pantai Batu Bedaun, Gua Maria	2
Pantai Batu Belubang, Pantai Mabai	2
Pantai Mabai, Pantai Tanjung Pesona	2

3. *Generate* 3-*itemset Frequent Pattern*
Proses selanjutnya adalah gabungkan (*join*) kembali tempat wisata yang ada pada Tabel 6 untuk menghasilkan kandidat 3-*itemset*, kemudian dihitung *support count*-nya, sehingga akan didapat hasilnya sebagai berikut:

Tabel 7 Kandidat 3-*itemset*

Tempat Wisata	Support Count
Pantai Tanjung Kiras, Pantai Batu Bedaun, Pantai Air Anyer	1
Pantai Tanjung Kiras, Pantai Batu Bedaun, Pantai Tanjung Kalian	0
Pantai Tanjung Kiras, Pantai Batu Bedaun, Pantai Batu Belubang	0
Pantai Tanjung Kiras, Pantai Batu Bedaun, Pantai Mabai	0
Pantai Tanjung Kiras, Pantai Batu Bedaun, Gua Maria	0

Tempat Wisata	Support Count
Pantai Tanjung Kiras, Pantai Batu Bedaun, Pantai Tanjung Pesona	0
Pantai Air Anyer, Pantai Tanjung Kalian, Pantai Batu Bedaun	0
……	….

Hilangkan (*prune)* data yang memiliki *support count* kurang dari *minimum support* yang telah ditentukan. Dapat dilihat pada Tabel 7 terdapat calon kandidat 3-*itemset* yang memiliki *support count* kurang dari *minimum support*, sehingga perlu dihilangkan. Berikut hasil kandidat yang memenuhi *minimum support*:

Tabel 8 Large 3-*itemset*

Tempat Wisata	Support Count
Pantai Batu Bedaun, Pantai Batu Belubang, Pantai Mabai	2

Hasil dari Tabel 8 menunjukkan bahwa sudah tidak ada kombinasi tempat wisata yang dapat dilakukan penggabungan kembali, sehingga proses pencarian *itemset* berhenti. Maka didapat *frequent itemset* sebagai berikut:

Tabel 9 Frequent itemset

Frequent Itemset
{Pantai Tanjung Kiras}
{Pantai Air Anyer}
{Pantai Batu Bedaun}
{Vihara Dewi Kwan Yin}
{Pantai Batu Belubang}
{Pantai Mabai}
{Gua Maria}
.....

4. Setelah didapatkan *frequent itemset* yang lebih besar atau sama dengan *minimum support*, maka selanjutnya adalah menghitung nilai *confidence* dari *frequent itemset* yang dimana nilai *minimum confidence* 70%. Berikut hasil perhitungannya , dimana perhitungan nilai *confidence* yang memiliki hasil berupa nilai desimal dengan ketentuan jika nilai angka dibelakang koma lebih besar lebih besar sama dengan lima, maka dibulatkan ke atas, sedangkan nilai angka yang dibelakang koma lebih kecil dari lima akan dibulatkan dibawah. Untuk perhitungan nilai *confidence* dan menentukan pola *association rule* dapat dilihat pada tabel 10.

Tabel 10 Perhitungan nilai confidence dan menentukan pola association rule

Tempat Wisata	Confidence	Rule
Jika ke {Pantai Batu Bedaun} maka ke {Pantai Batu Belubang dan Pantai Mabai}	2/6*100 %=33%	Rule Rejected
Jika ke {Pantai Batu Belubang} maka ke {Pantai Batu Bedaun dan Pantai Mabai}	2/4*100 %=50%	Rule Rejected
Jika ke {Pantai Mabai} maka ke {Pantai Batu Bedaun dan Pantai Batu Belubang}	2/4*100 %=50%	Rule Rejected
Jika ke {Pantai Batu Belubang dan Pantai Mabai} maka ke {Pantai Batu Bedaun}	2/4*100 %=50%	Rule Rejected
Jika ke {Pantai Batu Bedaun dan Pantai Mabai} maka ke {Pantai Belubang}	2/4*100 %=50%	Rule Rejected
Jika ke {Pantai Batu Bedaun dan Pantai Batu Belubang} maka ke {Pantai Mabai}	2/4*100 %=50%	Rule Rejected
Jika ke {Pantai Tanjung Kiras} maka ke {Pantai Batu Bedaun}	2/2*100 %=100 %	Rule Selected

2.1.6 Hasil Pengujian Data Mining

Berikut adalah hasil perhitungan algoritma *apriori* secara *manual* dibandingkan dengan hasil oleh sistem babel kite dimana nilai *minimum support* dua dan *minimum confidence 70%* yang telah dimasukan didalam sistem.

Tabel 11 Hasil Pengujian data mining

***Manual* (*Microsoft excel*)**	**Babel Kite**
jika ke pantai batu belubang maka ke pantai batu bedaun dan pantai mabai	jika ke gua maria maka ke pantai batu bedaun

jika ke pantai tanjung kiras maka ke pantai batu bedaun	jika ke pantai batu belubang maka ke pantai mabai
jika ke pantai batu belubang maka ke pantai batu bedaun	jika ke pantai tanjung pesona maka ke pantai mabai
jika ke gua maria maka ke pantai batu bedaun	jika ke pantai tanjung kiras maka ke pantai batu bedaun
jika ke pantai batu belubang maka ke pantai mabai	jika ke pantai batu belubang maka ke pantai batu bedaun dan pantai mabai
jika ke pantai tanjung pesona maka ke pantai mabai	jika ke pantai batu belubang maka ke pantai batu bedaun

3 KESIMPULAN

***Manual* (*Microsoft excel*)**	**Babel Kite**
jika ke pantai batu belubang maka ke pantai batu bedaun dan pantai mabai	jika ke gua maria maka ke pantai batu bedaun
jika ke pantai tanjung kiras maka ke pantai batu bedaun	jika ke pantai batu belubang maka ke pantai mabai
jika ke pantai batu belubang maka ke pantai batu bedaun	jika ke pantai tanjung pesona maka ke pantai mabai
jika ke gua maria maka ke pantai batu bedaun	jika ke pantai tanjung kiras maka ke pantai batu bedaun
jika ke pantai batu belubang maka ke pantai mabai	jika ke pantai batu belubang maka ke pantai batu bedaun dan pantai mabai
jika ke pantai tanjung pesona maka ke pantai mabai	jika ke pantai batu belubang maka ke pantai batu bedaun

DAFTAR PUSTAKA

[1] Indonesia Travel. [Online]. HYPERLINK "http://www.indonesia.travel/destinations/destination/26/bangka-belitung" http://www.indonesia.travel/destinations/destination/26/bangka-belitung

[2] Chandra. Babelprov. [Online]. HYPERLINK "http://babelprov.go.id/content/11298-wisatawan-kunjungi-babel" http://babelprov.go.id/content/11298-wisatawan-kunjungi-babel

[3] Micheline Kamber Jiawei Han, "Data Mining: Concepts and Technique," in *Data Mining: Concepts and Technique*. San Francisco: Diane Cerra, 2006, pp. 227-229.

[4] IDC Analyze the Future. [Online]. HYPERLINK "http://www.idc.com/prodserv/smartphone-os-market-share.jsp" http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[5] MPA., Ph.D, Surya Dharma, *Pendidikan, Jenis, dan Metode Penelitian Pendidikan*. Jakarta: Direktorat Tenaga Kependidikan Department Pendidikan Nasional, 2008.

[6] Ian Sommerville, *Software Engineering*, Eight Edition ed.: Addison Wesley, 2007.

[7] Sommerville Ian, *Software Engineering*, Eight Edition ed.: Addison Wesley, 2007.

[8] KBBI. (2015, Desember) KBBI. [Online]. HYPERLINK "http://kbbi.web.id/" http://kbbi.web.id/

[9] Ida Nuraida, *Manajemen Administrasi Perkantoran*. Yogyakarta: Kanisius, 2008.

[10] Chr. Jimmy L. Gaol, *Sistem Informasi Manajemen*. Yogyakarta: Grasindo, 2008.

[11] Komaruddin Sastradipoera, *Asas-asas menejemen perkantoran*. Bandung: Kappa Sigma, 2001.

[12] gsbipb. (2013, Oct.) gsbipb. [Online]. HYPERLINK "http://gsbipb.com/?p=821" http://gsbipb.com/?p=821

[13] Nong Ye, *Data Mining: Theoris Algorithms and Examples*. NW, US: Taylor & Francis, 2014.

[14] Agus Wahadyo and Sudarma S, *Tip Trik Android untuk pengguna Tablet & Handphone*. Jakarta: Penerbit Mediakita, 2012.

[15] Putri Atalapu, "Implementasi Location Based Service Berbasis Cell Id Untuk Anjungan Provinsi Sulawesi Selatan Taman Mini Indonesia Indah (Tmii) Memanfaatkan Teknologi Augmented Reality Pada Perangkat Bergerak Android," p. 11, 2012.

[16] Harry J. Rosenblatt Gary Shelly, *Systems Analysis and Design*. Boston: Course technology cengange learning, 2010.

[17] Hanif Fatta, *Rekayasa Sistem Pengenalan Wajah*. Yogyakarta: Andi, 2009.

[18] Tanjung Hendri, Prabowo Haryi Marimin, *Sistem Infromasi Manajemen : Sumber Daya Manusia*. Bogor: Grasindo, 2006.

[19] Hendra Divayana, *Konsep OOAD*. Jakarta: STMIK Eresha, 2010.

[20] Julius Hermawan, *Analisa Desain & Pemrograman Berorientasi Objek dengan UML dan Visual Basic.NET*. Yogyakarta: Andi, 2010.

[21] Kim Hamilton and Russell Miles, *Learning UML 2.0*. United States of America: O'Reilly, 2006.

[22] Janner Simarmata, *Rekayasa Web*. Yogyakarta: ANDI, 2010.

[23] json org. json org. [Online]. HYPERLINK "http://json.org/json-id" http://json.org/json-id

[24] Marsela dan Veronica S. Moertini Yulita, *Analisis Keranjang Pasar dengan Algoritma Hash-Based pada transaksi penjualan di apotek*, 2002.

[25] S.Kom., M.Kom. Dicky Nofriansyah, *Konsep Data Mining Vs Sistem Pendukung Keputusan*. Yogyakarta, Indonesia: Deepublish, 2015.

[26] Twinkle Puri, Binita Shah, Ishaan Bajaj, Binita Parekh Prof. Prashasti Kanikar, "Comparative Study of Apriori Algorithm Performance on Different Datasets," *Comparative Study of Apriori Algorithm Performance on Different Datasets*, vol. 4, no. 4, p. 40, April 2014.