

PENGGALIAN PENGETAHUAN DARI DATA RIWAYAT AKADEMIK UNTUK REKOMENDASI PEMILIHAN MATA KULIAH PILIHAN

Andhik Budi Cahyono¹, Sukrisno Mardiyanto²

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia¹⁾

Jalan Kaliurang Km.14,5 Sleman, Yogyakarta 55184.

Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung²⁾

Bandung, Jawa Barat.

Email : andhikbudi@uii.ac.id¹⁾, sukrisno@informatika.org²⁾

ABSTRACT

One of the higher education problems is to map out students potential effectively to achieve a more optimal education results. The mapping could be represented by choosing the right department, choosing thesis title, or choosing the right optional subject. Higher education is an institution that resulting many data including the student's academic profile which can be utilized to fulfil this objective. Data mining is a technology which can interface this need.

One of the data mining variant is association analysis and frequent itemset mining which are seeking the connection pattern between one attribute or item and the other. Data attribute that shows up often at the same time means that they have a strong association connection and can make a pattern used as information. Apriori algorithm is a popular algorithm that is used in association data mining. Although this algorithm have several disadvantages, this algorithm still commonly used because it's easiness to implement and it's flexibility to improved and adjusted with the purpose. As the addition, several publications have suggested some improvement for this algorithm, such as limiting the number of rules.

In this research, the utilization of Apriori Algorithm to extract knowledge from academic's profile data could not yet resulted in aimed recommendation due to the lack distribution of optional course which results in the lack of knowledge seeking result variation and only focused on certain optional course as the result of knowledge seeking.

Keywords : Academic's Profile Data, Data Mining, Apriori Algorithm, Choosing Course, Rule Limitation.

1. PENDAHULUAN

Perguruan tinggi merupakan salah satu organisasi yang menghasilkan data dalam jumlah yang besar, diantaranya adalah data riwayat akademik. Pengetahuan yang diperoleh dari data riwayat akademik tersebut bisa dimanfaatkan untuk membantu menyelesaikan beberapa persoalan yang ada dalam upaya meningkatkan kualitas pendidikan. Beberapa penelitian dengan tujuan yang berbeda-beda pernah dilakukan dengan memanfaatkan data riwayat akademik untuk membantu meningkatkan kualitas proses pendidikan di institusi pendidikan tinggi. Diantaranya adalah prediksi tingkat kelulusan mahasiswa dalam mengikuti mata kuliah tertentu seperti yang disebutkan dalam (baradwaj, 2001), (Kumar,

2011), dan (Yabing, 2013) , analisis kemungkinan pengunduran diri mahasiswa (Palanisamy, 2006) dan (Pujiantoro, 2011), prediksi waktu kelulusan dan kecepatan studi mahasiswa, prediksi prestasi mahasiswa seperti yang disampaikan dalam (Mangla, 2013), dll. Pada dasarnya dengan jumlah data yang sangat besar, potensi untuk menggunakan data tersebut dalam membantu berbagai persoalan yang dihadapi juga sangat besar.

Salah satu persoalan yang sering dihadapi di institusi pendidikan tinggi adalah bagaimana memetakan potensi mahasiswa dengan tepat. Bentuk pemetaan potensi bisa berupa : bagaimana menentukan bidang peminatan di jurusan, topik tugas akhir (penelitian) apa yang sesuai dengan profil

dan minat akademik mahasiswa, atau bagaimana menentukan mata kuliah pilihan yang sesuai dengan potensi akademiknya. Pertanyaan terakhir dari sub permasalahan tersebut akan menjadi fokus dalam makalah ini.

Dalam beberapa kasus mahasiswa sering kesulitan dalam memilih mata kuliah pilihan yang akan diambil. Hal ini bisa disebabkan antara lain karena kurangnya informasi mengenai mata kuliah pilihan yang akan diambil atau mahasiswa kurang menyadari potensi akademik yang dimilikinya. Salah satu indikator potensi akademik mahasiswa adalah riwayat nilai (transkrip) akademik. Transkrip akademik bisa dianggap menggambarkan, walau bukan ukuran yang selalu valid, potensi dan kompetensi mahasiswa dalam mengikuti perkuliahan. Nilai yang bagus dalam sebuah mata kuliah bisa diasumsikan bahwa mahasiswa punya kompetensi yang bagus pula dalam mata kuliah tersebut.

Selanjutnya dalam penelitian ini akan dicari hubungan antar potensi akademik mahasiswa dalam sebuah mata kuliah dengan mata kuliah lainnya. Dalam konteks pemilihan mata kuliah pilihan akan dicari hubungan antara potensi mahasiswa pada mata kuliah pilihan wajib dengan potensi mahasiswa dalam mata kuliah pilihan. Pengetahuan hasil pencarian tersebut akan digunakan untuk memberikan rekomendasi dalam menentukan mata kuliah pilihan yang akan diambil. Untuk mencari hubungan antara mata kuliah dalam data transkrip akademik tersebut akan digunakan Algoritma Apriori yang merupakan kelompok algoritma untuk tugas asosiasi data mining. Tugas Apriori yaitu mencari pengetahuan berupa aturan yang menyatakan hubungan munculnya mata kuliah wajib yang bernilai bagus dengan mata kuliah pilihan yang bernilai bagus seperti contoh aturan berikut ini :

jika (Alpro1 : bagus, Jarkom : bagus, Basisdata : bagus) maka (Jarkom Lanjut : bagus, Semantic Web : bagus)

Aturan di atas bisa diartikan mahasiswa yang sukses mendapat nilai bagus pada mata kuliah wajib Alpro 1, Jarkom, dan Basisdata maka akan mendapat nilai yang bagus juga dalam mata kuliah pilihan Jarkom Lanjut dan *Semantic Web*, dengan asumsi, misalnya, bahwa nilai yang tergolong bagus adalah nilai yang mendapat indeks A atau B.

Pengetahuan yang diperoleh dari teknik asosiasi dengan Algoritma Apriori akan dikombinasikan dan disesuaikan dengan aturan pengambilan mata kuliah pilihan pada studi kasus yaitu di Jurusan Informatika Universitas Islam Indonesia. Pengetahuan hasil kombinasi tersebut yang akan menjadi model untuk memberikan rekomendasi bagi mahasiswa yang akan mengambil mata kuliah pilihan

2. KAJIAN PUSTAKA

2.1. *Asosiation Rule Mining*

2.1.1. *Penggalian Pengetahuan dan Data Mining*

Menurut Han dan Kamber (2006) *data mining* bisa diartikan mencari pengetahuan dari sekumpulan data dalam jumlah yang besar. Karena tujuan utama dari proses penggalian adalah mencari pengetahuan, data mining juga dikenal dengan istilah *Knowledge Discovery from Data* (KDD). Tahapan penggalian pengetahuan menurut Han dan Kamber adalah sebagai berikut :

a. *Data Cleaning*

Proses untuk menghilangkan *noise* dan inkonsistensi pada data.

b. *Data Integration*

Pengkombinasian beberapa sumber data yang ada atau yang dibutuhkan.

c. *Data Selection*

Pemilihan data yang dibutuhkan untuk kebutuhan analisis.

d. *Data Transformation*

Transformasi atau konsolidasi data ke dalam format yang cocok terhadap kebutuhan analisis atau algoritma data mining.

e. *Data Mining*

Penggunaan methoda atau algoritma data mining untuk mencari pengetahuan dari data hasil proses sebelumnya.

f. *Pattern Evaluation*

Mengidentifikasi pengetahuan yang diperoleh apakah benar - benar menarik dan sesuai dengan kebutuhan.

2.2. Asosiasi Frequent Itemset Mining

Asosiasi merupakan salah satu jenis data mining yang bertujuan mencari hubungan antara atribut data pada basisdata sumber. Konsep asosiasi berkaitan erat dengan konsep *frequent itemset mining*. *Frequent itemsets* adalah *itemsets* yang sering muncul dalam basisdata atau nilai *support count (count)* nya lebih besar daripada *minimum support count (min_count)*. Istilah item pada awalnya mengacu pada item barang belanjaan dari aktivitas belanja seorang pembeli. Item - item belanjaan tersebut biasanya diletakkan dalam keranjang belanja. Oleh karena itu analisis terhadap data-data pembelian item ini dikenal dengan analisis keranjang belanja (*Market Basket Analysis*). Ide utama dari analisis keranjang belanja adalah mencari hubungan antara item yang berbeda yang dibeli pada saat bersamaan. Misalkan $I = \{ I_1, I_2, \dots, I_m \}$ adalah sebuah himpunan item dan **D** adalah **database transaksi**. *T* adalah notasi untuk transaksi yang ada dalam database *D* dengan isi dari *T* adalah himpunan bagian dari *I*. Setiap transaksi mempunyai identifier *TID* dengan nilai yang unik. Misalnya dalam database *D* ditemukan transaksi - transaksi berikut :

- $I = \{A, B, C, D, E, F\}$ merupakan himpunan item yang berada dalam database transaksi.

Tabel.1 Contoh Transaksi *T* dalam Database *D*

TID	Item
1001	A , C, D, E
1002	A, B, D, E, F
1003	B, D, F
1004	D, E, F
1005	B, D, E, F
1006	A, C, E, F

- Jumlah transaksi yang ada dalam database ada 6. Transaksi pertama dengan ID 1001 selanjutnya dinotasikan dengan T_1 . Berturut - turut T_2, T_3, \dots, T_6 .
- Setiap transaksi mengandung *itemset* yang terdiri dari item. Misalkan T_1 mempunyai *itemset*{A, C, D, E}.
- *Itemset* adalah himpunan tidak kosong item yang terbentuk dari item - item *I*. Misalkan {A, C, E}, {A, B, D}, {A,B}, {F}, dst.
- *K - itemset* adalah *itemset* yang terdiri dari *k* item. *1-Itemset* adalah *itemset* yang terdiri dari 1 item, contoh : {E}, {D}, {A}, dst. *2-Itemset* adalah *itemset* yang terdiri dari 2 item, contoh : {A, E} , {D, F}, {D, E}, dst.

Tujuan dari tugas asosiasi dan *frequent itemset mining* adalah menemukan *frequent itemsets*. *Frequent itemset* adalah *k - itemset* yang sering muncul dalam database transaksi *D*. Sering atau tidak kemunculan sebuah *itemset* ditentukan oleh nilai dua parameter yaitu : *support count (count)* dan minimal *support count (min_count)*. *Support count* menggambarkan beberapa jumlah kemunculan *itemsets* dalam transaksi. Sebuah *itemset* dinyatakan frequent apabila nilai *count* - nya lebih besar atau sama dengan *min_count*.

2.3. Pencarian Aturan Asosiasi

Tujuan dari analisis asosiasi tidak hanya mencari *frequent itemsets*. Pengetahuan dalam analisis asosiasi bisa berupa *frequent itemset* itu sendiri atau berupa aturan asosiasi (*association rule*). Aturan asosiasi dibentuk

dari *frequent itemsets* berdasarkan beberapa kriteria diantaranya : *support* dan *confidence*. Misalkan dari *frequent itemset* {B, D} bisa diperoleh dua aturan yaitu : B => D dan D => B. Aturan yang kuat (strong association rule) adalah aturan yang mempunyai nilai *support* dan *confidence* yang lebih besar dari minimum *support* (*min_sup*) dan minimum *confidence* (*min_conf*). Nilai *min_sup* dan *min_conf* ditentukan oleh penggali pengetahuan. Sedangkan perhitungan *support* dan *confidence* diformulasikan dalam formula 2.1 dan 2.2 :

$$Support (X \Rightarrow Y) = P (X \cup Y) \dots \dots \dots (1)$$

$$Confidence (X \Rightarrow Y) = P (X/Y) \dots \dots \dots (2)$$

Support menggambarkan persentase jumlah *itemset* yang membentuk aturan terhadap keseluruhan transaksi dalam basisdata . Sehingga formula atau rumusan (1) bisa dijelaskan lebih lanjut sebagai berikut :

$$support (X \Rightarrow Y) = \frac{support\ count (X \cup Y)}{\sum\ transaksi} \dots \dots \dots (3)$$

Sedangkan *confidence* menggambarkan persentase jumlah transaksi yang mengandung *itemset* dengan jumlah transaksi yang mengandung *antecedent* (bagian kiri) dari aturan.

$$confidence (X \Rightarrow Y) = \frac{support\ count (X \cup Y)}{support\ count (X)} \dots \dots (4)$$

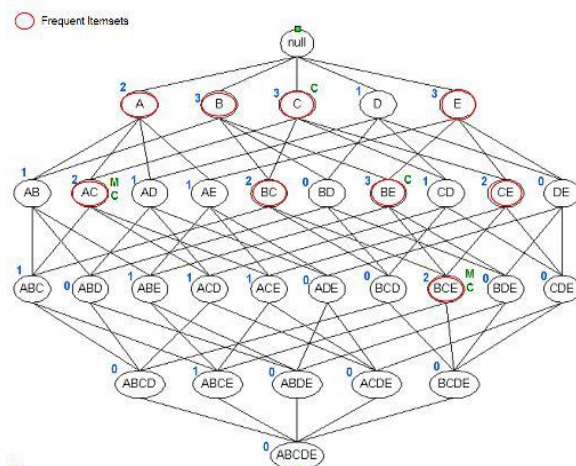
Nilai *support* dan *confidence* biasanya dinyatakan dengan nilai 0 % – 100 % bukan 0 – 1.

2.4. Algoritma Apriori

Menurut (Kaur, 2013) dan (Mangla, 2013) kelemahan dari pencarian *frequent itemsets* adalah ruang pencarian yang besar jika jumlah item banyak dan adanya pengecekan basisdata yang berulang-ulang.

Perlu dipahami dulu bahwa pencarian *frequent itemsets* pada setiap *k-itemsets* melibatkan (*k-1*)-*itemset*. *2-itemsets* merupakan hasil dari penggabungan dari item-item yang ada pada 1-*itemset*. Begitu juga ketika akan membentuk *3-itemsets*, caranya adalah dengan mengabungkan *itemsets - itemsets* pada 2-*itemsets*. Secara umum proses untuk mencari *frequent itemsets* pada *k-itemsets* adalah membentuk *k-itemsets* dari (*k-1*)-*itemsets* kemudian menghitung nilai *support count*-nya untuk masing-masing *itemsets* yang terbentuk. Jika lebih besar dari *min_count* maka termasuk *frequent itemsets*, jika lebih kecil akan dihilangkan dari daftar *frequent itemsets*.

Jika pada 1-*itemsets* ada 5 item, maka pembangkitan *itemsets* untuk setiap tahapnya bisa digambarkan seperti pada gambar.1 :



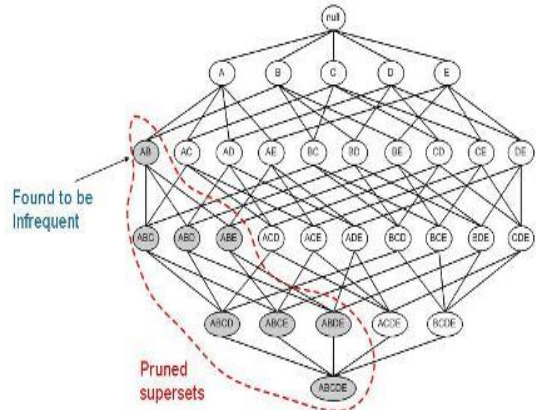
Gambar.1. Pembangkitan *Itemset* untuk Jumlah Item 5.

(Sumber : <http://www.codeproject.com/Articles/70371/Apriori-Algorithm>)

Dari gambar.1 bisa dilihat jumlah seluruh *itemsets* yang terbentuk dari lima item adalah 31. Sehingga diperoleh hubungan antara jumlah seluruh *itemsets* yang dibangkitkan dengan jumlah item adalah : **itemsets = 2^{item} – 1**. Tentu saja operasi *join* yang dilakukan lebih banyak lagi daripada jumlah *itemsets* yang dihasilkan. Cara pembangkitan *itemsets* seperti ini dikenal dengan istilah *Brute Force* (Han, 2006).

Algoritma Apriori bekerja dengan cara memberikan sugesti atau “level kebijaksanaan pencarian” atau **properti apriori** dalam pembangkitan *itemsets* untuk setiap *k-itemsets*. Berbeda dengan cara pembangkitan *itemsets* secara *Brute Force*, *itemsets* yang dibangkitkan untuk setiap *k-itemsets* dengan Apriori berasal dari penggabungan item-item yang ada pada **frequent (k-1) - itemsets** saja dan mempertimbangkan properti apriori-nya. Properti apriori berisi pengetahuan sebagai berikut : “Semua *subset* dari *frequent itemsets* adalah *frequent*”. Pernyataan di atas dapat dijelaskan dengan pernyataan sebagai berikut : “Jika sebuah *itemsets* memiliki *support* yang lebih kecil dari *min_sup*, maka semua *superset* yang mengandung *itemsets* tersebut juga memiliki *support* yang tidak lebih dari *min_sup*”. Algoritma apriori bekerja dalam dua tahap, yaitu :

1. *Join step*. Pencarian *frequent k-itemsets* pada Apriori dilakukan dengan melakukan join (penggabungan) *frequent itemsets* pada (k-1)-*itemset* yang nilai support count-nya lebih besar dari atau sama dengan *min_count*. *Frequent k-itemsets* dinotasikan dengan L_k sedangkan kandidat *itemsets* dinotasikan dengan C_k . C_k adalah hasil join *itemsets* yang ada pada L_{k-1} , sedangkan L_k berisi *frequent itemsets* yang ada pada C_k .
2. *Prune step*. C_k berisi *k-itemsets* baik yang *frequent* maupun yang tidak. Langkah selanjutnya adalah menghitung nilai count untuk masing-masing *itemsets* yang ada di C_k . *Itemsets* yang *frequent* akan masuk sebagai anggota dari L_k . Pemangkasan *itemsets* pada Apriori tidak hanya berdasar pada perhitungan count, tapi juga berdasar properti Apriori. *Itemsets* hasil join yang mengandung sub *itemsets* yang tidak *frequent* maka akan dipangkas terlebih dahulu karena sudah pasti tidak *frequent*. Dengan adanya pemangkasan tersebut jumlah kandidat menjadi lebih sedikit daripada menggunakan *Brute Force*.



Gambar.2. Pembangkitan Itemsets Dengan Algoritma Apriori.

(Sumber :

<http://philips.wordpress.com/2006/06/07/apriori/>)

2.5. Metarule Constraints Mining

Metarule constraints mining bertujuan mempersempit ruang pencarian *frequent itemset* dan kandidat aturan asosiasi yang dibangkitkan. *Metarule* memungkinkan pengguna untuk menentukan bentuk aturan yang diinginkan sehingga pencarian aturan lebih terarah daripada membangkitkan semua aturan yang mungkin. *Metarule* ditentukan berdasarkan pengalaman, harapan, atau intuisi dari analist terhadap data yang ada. Bentuknya berupa contoh (*template*) aturan yang ingin dihasilkan dari proses penggalian aturan. Misalkan seorang *analist* dari perusahaan *hardware* dan *software* ingin mencari hubungan antar penjualan barang dengan data pelanggan (penghasilan, umur, jenis kelamin, dsb) maka dibuatlah *template* seperti pada formula 2.5 :

$$P_1(X, Y) \wedge P_2(X, W) \dots \wedge P_k(X, Z) \implies \text{beli}(X, \text{software}) \dots \dots \dots (5)$$

Dimana P_1 , P_2 , dan P_k adalah variabel predikat yang juga merupakan atribut dari dataset pelanggan, X adalah variabel merepresentasikan pelanggan. Sedangkan Y , W , dan Z adalah nilai untuk masing - masing P_1 , P_2 dan P_k . Dengan adanya *template* tersebut diharapkan *data mining* bisa menghasilkan aturan yang sesuai dengan kebutuhan. Sebagai contoh aturan yang

dihasilkan dari pencarian berdasar *template* tersebut sebagai berikut :

$Umur(X, "30...39") \wedge Pendapatan(X, "> 50000000") == > Beli(X, "Office software")$

Maksud dari aturan di atas adalah jika pelanggan mempunyai umur antara 30 – 39 tahun dan pendapatan lebih besar dari Rp 5.000.000,00 maka pelanggan akan membeli *Office Software*.

3. METODE PENELITIAN

Dalam kurikulum terbaru yang dimiliki Jurusan Informatika UII (kurikulum 2010), disebutkan bahwa jumlah mata kuliah pilihan yang bisa diambil minimal sebanyak tujuh mata kuliah pilihan atau setara dengan 21 SKS. Ada aturan yang harus diperhatikan dalam mengambil mata kuliah pilihan. Pertama, dari mata kuliah pilihan yang akan diambil, empat diantaranya harus berasal dari fokus studi yang sama dan sisanya boleh mengambil dari fokus atau minat yang lain. Kedua, masing - masing mata kuliah pilihan juga mempunyai mata kuliah prasyarat yang harus dipenuhi (lulus) untuk bisa mengambilnya. Tujuan dari aturan ini adalah untuk mempertegas dan memperkuat kemampuan mahasiswa dalam bidang ilmu yang diminatnya serta untuk meningkatkan kualitas tugas akhir. Kedua aturan tersebut akan dikombinasikan dengan aturan yang dihasilkan dari Algoritma Apriori untuk rekomendasi mata kuliah pilihan.

3.1. Analisis Sumber Data

Data yang akan digunakan sebagai sumber pengetahuan adalah data riwayat akademik (transkrip nilai) mahasiswa dari tahun 2006 – 2012. Pertimbangan menggunakan data tersebut karena diperkirakan mahasiswa pada rentang tahun tersebut adalah mahasiswa yang pernah atau masih bersinggungan dengan kurikulum terakhir yang diterapkan di Jurusan Informatika yaitu Kurikulum 2010.

Perekomendasi yang akan dilakukan mengacu pada kurikulum 2010.

Data yang akan unduh sebagai data latihan merupakan gabungan dari berbagai tabel dari sumber data utama (*database* nilai). Karena data ini merupakan data yang memuat privasi dan skema basisdata juga merupakan privasi organisasi sehingga tidak bisa ditampilkan di penelitian ini skema basisdata nya. Data uji sudah merupakan data hasil join beberapa tabel dalam basisdata. Tabel.2 merupakan contoh sumber data latihan.

Tabel.2 Sampel Sumber Data

NIM	KURIKULUM	NAMA_MK	NILAI
1	2002	Algoritma dan Pemrograman I	C
1	2006	Algoritma dan Pemrograman I	C+
2	2002	Algoritma dan Pemrograman I	C
3	2002	Algoritma dan Pemrograman I	A-
1	2002	Algoritma dan Pemrograman II	C
2	2006	Algoritma dan Pemrograman II	C
2	2002	Algoritma dan Pemrograman II	C
3	2002	Algoritma dan Pemrograman II	C

Dari tabel.2 dapat diperoleh gambaran bahwa ada duplikasi data nilai untuk satu mata kuliah. Karena itu data yang mengandung duplikasi akan dibersihkan sebelum masuk dalam proses penggalian pengetahuan selanjutnya. Data latihan tersebut nantinya juga akan diubah format datanya sehingga lebih sesuai dengan format data untuk analisis asosiasi dengan menggunakan Algoritma Apriori. Jumlah data sumber adalah 200.000 baris dan terdapat kurang lebih 2.000 NIM (mahasiswa).

3.2. Desain Pemecahan Permasalahan

Pada bagian ini akan dibahas mengenai rancangan pemecahan masalah penggalian pengetahuan dan perekomendasi mata kuliah pilihan. Rancangan ini berdasar pada tahapan penggalian pengetahuan yang sudah dijelaskan pada bagian II. Alur penyelesaian permasalahan bisa dilihat pada gambar.3.

3.3. Data Cleaning

Pada tahap ini data sumber akan dibersihkan dari duplikasi baris data. Tabel 2 menunjukkan sumber data masih mengandung banyak duplikasi nilai mata kuliah. Pembersihan dilakukan dengan mengeliminasi baris data yang mengandung nama mata kuliah (nama_mk) dan nim yang sama, tetapi memiliki nilai yang berbeda. Sebagai contoh :

Tabel.3 Contoh Sampel Sumber Data 2

NIM	KURIKULUM	NAMA_MK	NILAI
1	2002	Algoritma dan Pemrograman I	C
1	2002	Algoritma dan Pemrograman I	A-

Data di atas menunjukkan adanya duplikasi pada tabel. Dari tabel baris data yang akan dibuang adalah baris data pertama s.d. ketiga. Sedangkan baris data yang dipertahankan adalah baris data dengan index nilai terbesar (A-) dengan asumsi nilai terbesar adalah nilai terakhir yang diperoleh oleh mahasiswa. Asumsi tersebut digunakan untuk seluruh proses pembersihan duplikasi data.

3.4. Transformasi Data

Transformasi data dilakukan sebanyak dua kali. Pertama yaitu merubah data dalam format tabel.2 ke dalam format tabel data

horizontal. Format data horizontal yang dimaksud adalah sebagai berikut :

Tabel.4 Contoh Hasil Transformasi Data Tahap Pertama

NIM	MK_1	MK_2	...	PIL_1	PIL_2	IPK
112	A	A-	...	C	A	3.00
113	B	A/B	...	NULL	B	3.02
116	C	C	...	A	C	2.90

Keterangan :

- Baris paling atas adalah nama kolom yang terdiri dari NIM, daftar mata kuliah wajib (MK_1, MK_2, dst.), daftar mata kuliah pilihan (PIL_1, PIL_2, dst.), serta IPK.
- MK_1 merupakan alias dari nama mata kuliah Algoritma dan Pemrograman 1. MK menandakan bahwa alias tersebut untuk mata kuliah wajib. Sedangkan PIL menandakan bahwa alias tersebut untuk mata kuliah pilihan. Alias digunakan untuk memperpendek nama kolom dan memudahkan proses transformasi data. Daftar alias dan nama mata kuliah ada di lampiran.
- Rentang indek nilai mahasiswa di Jurusan Informatika UII adalah : A, A-, A/B, B+, B, B-, B/C, C+, C, C-, C/D, D+, D, dan E. Jika baris data mengandung NULL menandakan mahasiswa tidak mengambil mata kuliah pilihan tersebut.

Setelah melalui transformasi tahap pertama selanjutnya dilakukan transformasi tahap kedua agar data lebih mudah digunakan ketika menjalankan Algoritma Apriori. Berikut contoh format data tahap kedua.

Tabel.5 Contoh Hasil Transformasi Data Tahap Pertama

NIM	MK_1	MK_2	...	PIL_1	PIL_2
112	1	1	...	0	1
113	0	1	...	0	0
116	0	0	...	1	0

Semua nilai A, A-, dan A/B diubah menjadi nilai “1” dan selainnya diubah menjadi nilai “0”. Hal ini dimaksudkan untuk menunjukkan bahwa nilai yang dianggap bagus adalah ketiga index nilai A, A-, dan A/B. Tabel di atas menunjukkan hubungan kemunculan nilai bagus antara mata kuliah. Sebagai contoh baris pertama dari tabel 5 bisa diartikan sebagai kemunculan item MK_1, MK_2, dan PIL_2 secara bersama - sama.

3.5. Pembangkitan Aturan Asosiasi

Pada tahap ini Algoritma Apriori akan digunakan untuk pencarian *frequent itemsets*. Tidak semua *frequent itemsets* akan digunakan untuk membangkitkan aturan asosiasi. Hanya *frequent itemsets* yang mengandung mata kuliah pilihan (*frequent constraint itemsets*) saja yang akan digunakan membangkitkan aturan.

Aturan asosiasi dibangkitkan dari *frequent itemset* yang sesuai dengan *constraint* yaitu yang mengandung mata kuliah pilihan. Sebagai contoh dari *frequent itemset* {MK_2, MK_5, PIL_5} dapat dibangkitkan aturan asosiasi :

if (MK_2 = bagus \wedge MK_5 = bagus) *then* PIL_5 = bagus

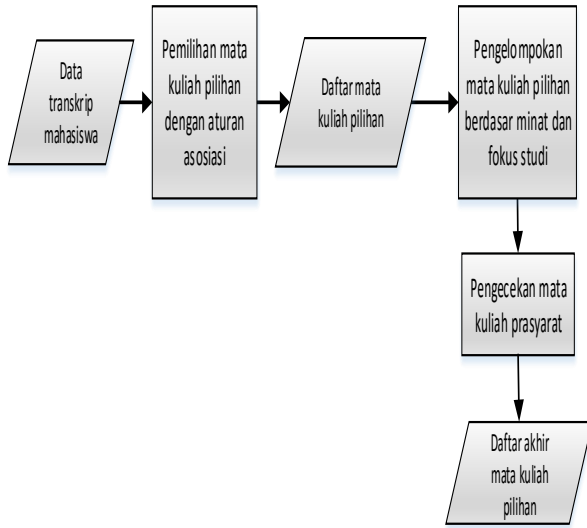
Dari itemset yang ada di atas sebenarnya ada aturan lain yang bisa dibentuk. Hanya saja aturan tersebut diabaikan karena tujuan yang ingin dicapai adalah pencarian aturan yang menempatkan mata kuliah pilihan sebagai *consequent*.

3.6. Proses Rekomendasi

Proses rekomendasi mata kuliah pilihan dilakukan dengan membandingkan data masukan dari pengguna yaitu data mata kuliah yang sudah diambil dengan aturan asosiasi. Setelah diperoleh daftar mata kuliah yang direkomendasikan, selanjutnya dibandingkan lagi dengan aturan pengambilan mata kuliah pilihan. Daftar mata kuliah yang direkomendasikan akan dikelompokkan berdasar fokus dan minat studi serta dicek apakah mata kuliah prasyaratnya lulus. Alur rekomendasi dapat dilihat pada gambar.4.

Aturan yang sudah diperoleh dengan menggunakan Algoritma Apriori akan disusun berdasar aturan sebagai berikut (Palanisamy, 2010) :

- Aturan dengan *confidence* lebih besar akan menempati urutan lebih tinggi (lebih kecil nomer urutnya).
- Jika dua atau lebih aturan memiliki *confidence* yang sama, maka akan dilihat nilai *support*-nya. Nilai *support* lebih besar akan menempati urutan lebih tinggi.
- Jika *confidence*, *support*, dan *lift* rasio sama besar maka aturan yang ditemukan lebih awal yang akan menempati urutan lebih tinggi.



Gambar.4 Proses Rekomendasi Berdasar Aturan Asosiasi dan Aturan Pengambilan Mata Kuliah.

Perekomendasi dilakukan dengan memetakan data transkrip mahasiswa pada setiap aturan. Kemudian mencocokkan setiap item pada data transkrip mahasiswa dengan bagian *antecedent*.

Tabel.6 Contoh Format Data Masukan Rekomendasi

Mk_1	Mk_2	Mk_3	Mk_4	Mk_5	...	Mk_N
A	B	C+	B/C	A-	.	A

Jika ada item yang sesuai dengan bagian *antecedent* maka bagian *consequent* dari aturan akan menjadi mata kuliah yang direkomendasikan. Tabel.5 menunjukkan format data transkrip sebagai masukan dari proses rekomendasi.

Proses rekomendasi mata kuliah pilihan dilakukan dengan membandingkan data mata kuliah yang sudah diambil dengan aturan asosiasi aturan pengambilan mata kuliah pilihan. Daftar mata kuliah yang direkomendasikan akan dikelompokkan berdasar fokus dan minat studi serta dicek apakah mata kuliah prasyaratnya lulus.

4. HASIL DAN PEMBAHASAN

4.1. Data uji dan skenario pengujian

Data uji dibagi menjadi dua yaitu data uji untuk menguji akurasi dari aturan asosiasi yang dihasilkan oleh Apriori dan data uji untuk merekomendasi. Data uji untuk menguji akurasi aturan adalah data dalam format biner seperti pada tabel 1. Data tersebut merupakan potongan data latih yang diambil secara acak. Sedangkan data uji untuk merekomendasi adalah contoh data transkrip nilai mahasiswa yang ingin memilih mata kuliah pilihan.

Pengujian dilakukan dengan tujuan sebagai berikut :

1. Melihat persentase jumlah kandidat *frequent constraint itemset* terhadap *frequent itemset*.
2. Melihat kemampuan Apriori menghasilkan aturan yang sesuai dengan template aturan yang sudah ditetapkan.
3. Mencari nilai *min_count*, *min_sup*, dan *min_conf* yang optimal sehingga bisa menghasilkan aturan yang paling baik dalam merekomendasikan.

Pengujian pertama dilakukan dengan mengubah nilai *min_count*, *min_sup*, dan *min_conf*. Hasil yang akan diamati adalah : jumlah *frequent itemsets*, jumlah *frequent constraint itemsets*, jumlah aturan yang dihasilkan, akurasi aturan, *consequent* yang dihasilkan oleh aturan, dan nilai *confidence* tertinggi yang mampu dihasilkan oleh aturan.

Pengujian kedua dilakukan untuk merekomendasikan data uji rekomendasi (data transkrip) untuk mendapatkan mata kuliah pilihan yang sesuai.

4.2. Hasil pengujian

Pengujian pertama menggunakan nilai *min_count* = 170. Nilai ini adalah nilai *min_count* terkecil yang bisa diuji dengan perangkat lunak yang telah dibangun. Nilai ini lebih kecil dari nilai rata - rata data yang mampu di-*cover* oleh mata kuliah pilihan pada tabel data biner.

4.2.1. Pengujian Dengan Min_Count = 170

Pengujian dilakukan dengan mengubah parameter *min_cup* dan *min_conf*. Nilai *min_sup* terkecil yang ditetapkan adalah 15 % (0,15) sampai dengan nilai tertinggi yang bisa dicapai. Nilai 15 % dianggap sudah cukup *representative* untuk menggambarkan bahwa sebuah itemset *frequent* (sering muncul) dalam basis data sumber. Hasil pengujian bisa dilihat pada tabel 7 s.d. 8.

Tabel.7 Hasil Pengujian Dengan *Min_Count* 170 dan *Min_Sup* 0,15

No	<i>Min_conf</i>	Jumlah Aturan	Akurasi	Nilai <i>Consequent</i> yang Muncul
1	40 %	108	48,02 %	pil_2, pil_5, pil_6, pil_15, pil_20, pil_24, pil_36
2	50 %	38	56,04 %	pil_2, pil_5, pil_15, pil_20, pil_24
3	60 %	14	63,01 %	Pil_20
4	70 %	1	71 %	Pil_20

Jumlah *frequent itemset* dan *frequent constraint itemset* yang dihasilkan dari percobaan diatas masing - masing adalah 4257 dan 969 (18 % dari *frequent itemsets*). Sedang nilai *confidence* tertinggi selama percobaan adalah : 0,714 (71,4 %). Sedangkan hasil pencarian aturan dengan *min_count* = 170 dan *min_sup* = 20 % dapat dilihat pada tabel.7.

Tabel.8 Hasil Pengujian Dengan *Min_Count* = 170 Dan *Min_Sup* = 0,2.

No	<i>Min_Conf</i>	Jumlah Aturan	Akurasi	Nilai <i>Consequent</i> yang Muncul
1	40	25	47,08 %	Pil_5, pil_20, pil_24
2		8	55,09 %	Pil_20, pil_24
3		2	62,08 %	Pil_20
4		0	0	Tidak ada

Hasil pencarian aturan dengan *min_count* = 170 dan *min_sup* = 30 % dapat dilihat pada tabel.9.

Tabel.9 Hasil Pengujian Dengan *Min_Count* = 170 dan *Min_Sup* = 0,3

No	<i>Min_Conf</i>	Jumlah Aturan	Akurasi	Nilai <i>Consequent</i> yang Muncul
1	40 %	1	51 %	Pil_20
2	50 %	1	51 %	Pil_20
3	60 %	0	0	Tidak ada
4	70 %	0	0	Tidak ada

Pencarian dengan *min_count* = 170 dan *min_sup* lebih dari 40 % tidak menghasilkan aturan sama sekali. Hal ini menunjukkan bahwa peluang munculnya mata kuliah pilihan dengan nilai bagus bersamaan dengan mata kuliah wajib dengan nilai yang bagus tidak lebih dari 40 % (+ 600 baris data) di dalam basisdata riwayat nilai mahasiswa.

4.2.2. Pengujian Dengan *Min_Count* = 250, 350, dan 450

Tabel.10 Hasil Pengujian Dengan *Min_Count* 250

<i>Min_Sup</i>	<i>Min_Conf</i>	Jumlah Rule	Akurasi	Nilai <i>Consequent</i> yang Muncul
0,15	40 %	102	48,03 %	pil_5, pil_20, pil_24, pil_36
	50 %	35	56,06 %	pil_5, pil_20, pil_24
	60 %	14	62,09 %	pil_20
	70 %	1	71,04 %	pil_20
0,2	40 %	25	47,08 %	pil_5, pil_20, pil_24
	50 %	8	55,09 %	pil_20, pil_24
	60 %	2	62,75 %	pil_20
	70 %	0	0	Tidak ada
0,3	40 %	1	51 %	pil_20
	50 %	1	51 %	pil_20
	60 %	0	0	Tidak ada
	70 %	0	0	Tidak ada
0,4	40 %	0	0	Tidak ada

Jumlah *frequent itemset* yang dihasilkan adalah 831 dan jumlah *frequent constraint itemset* adalah 168 (15 % dari *frequent itemsets*). Sedang nilai *confidence* tertinggi selama percobaan adalah : 0,64 (64 %).

Tabel.11 Hasil Pengujian Dengan *Min_Count* 350.

<i>Min_sup</i>	<i>Min_conf</i>	Jumlah Rule	Akurasi	Nilai <i>Consequent</i> yang Muncul
0,15	40 %	20	47,09 %	pil_5, pil_20, pil_24
	50 %	7	55,07 %	pil_20, pil_24
	60 %	2	62,75 %	pil_20
	70 %	0	0	Tidak ada
0,2	40 %	20	47,09 %	pil_5, pil_20, pil_24
	50 %	7	55,07 %	pil_20, pil_24
	60 %	2	62,75 %	pil_20
	70 %	0	0	Tidak ada
0,3	40 %	1	51 %	pil_20
	50 %	1	51 %	pil_20
	60 %	0	0	Tidak ada
	70 %	0	0	Tidak ada
0,4	40 %	0	0	Tidak ada

Jumlah *frequent itemset* yang dihasilkan adalah 196 dan *frequent constraint itemset* adalah 35 (16 % dari *frequent itemsets*). Sedang nilai *confidence* tertinggi selama percobaan adalah : 0,64 (64 %).

Tabel 12 Hasil Pengujian Dengan Min_Count 450.

Min_sup	Min_conf	Jumlah Rule	Akurasi	Nilai Consequent yang Muncul
0,15	40 %	6	48,03 %	pil_5, pil_20, pil_24
	50 %	2	55,04 %	pil_20, pil_24
	60 %	1	61,08 %	pil_20
	70 %	0	0	Tidak ada
0,2	40 %	6	48,03 %	pil_5, pil_20, pil_24
	50 %	2	55,04 %	pil_20, pil_24
	60 %	1	61,08 %	pil_20
	70 %	0	0	Tidak ada
0,3	40 %	1	51 %	pil_20
	50 %	1	51 %	pil_20
	60 %	0	0	Tidak ada
0,4	70 %	0	0	Tidak ada
	40 %	0	0	Tidak ada

Jumlah *frequent itemset* adalah 70 dan *frequent constraint itemset* adalah 14 (16 % dari *frequent itemsets*). Sedang nilai *confidence* tertinggi selama percobaan adalah : 0,64 (64 %). Jika pengujian diteruskan maka diperoleh nilai *min_count* maksimum sampai Apriori tidak menghasilkan *itemsets* yang mengandung *constraint* adalah 510. Jika *min_count* lebih besar dari nilai tersebut, Apriori tidak bisa menghasilkan *itemsets* yang mengandung *constraint*.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Aturan yang dihasilkan dari penggalian pengetahuan data latih menggunakan Apriori cenderung mempunyai nilai consequent yang tidak beragam. Dari percobaan dengan beberapa variasi nilai *min_count* dan *min_sup*, nilai consequent hanya mengarah pada tiga mata kuliah pilihan : pil_5 (*Cyber Law*), pil_20 (Pemrograman Non Prosedural), dan pil_24 (Pengamanan Sistem Komputer). Hal ini menyebabkan proses

perekomendasi menghasilkan rekomendasi yang tidak beragam juga.

2. Semakin besar nilai *min_count*, *min_sup*, dan *min_conf* keberagaman aturan semakin sedikit, tetapi akurasi semakin besar. Sebaliknya, semakin kecil nilai ketiga parameter tersebut keberagaman aturan semakin banyak, tetapi akurasi semakin kecil. Jika dipilih akurasi yang besar maka proses perekomendasi menjadi tidak berguna karena sudah pasti hasil perekomendasi hanya mengarah pada satu atau dua mata kuliah. Jika dipilih aturan dengan *consequent* yang beragam maka hasil rekomendasi menjadi kurang valid karena dihasilkan dari aturan yang memiliki akurasi yang kecil. Oleh karena itu dalam penelitian ini aturan yang dipilih sebagai dasar pemberian rekomendasi adalah aturan dengan consequent yang beragam dengan nilai akurasi yang bisa diterima ($\geq 50\%$) yaitu aturan yang dibentuk dari *min_count* = 170, *min_sup* = 0,15 dan *min_conf* = 50 % dengan besar akurasi 56,4 %.
3. Nilai *confidence* tertinggi dari hasil pengujian adalah 71,5 % dan aturan yang dihasilkan dengan nilai *min_conf* 60 % hanya berjumlah dua buah aturan. Sedangkan nilai *support* maksimum yang bisa dicapai aturan adalah 31 % dengan jumlah aturan yang memiliki nilai *support* di atas 30 % hanya satu aturan. Sedikitnya aturan yang mempunyai *confidence* lebih dari 60 % menunjukkan sedikitnya mata kuliah wajib dengan nilai yang bagus dan mata kuliah pilihan dengan nilai yang bagus memiliki hubungan yang kuat. Sedangkan sedikitnya jumlah aturan yang memiliki *support* di atas 30 % menunjukkan sedikitnya mata kuliah wajib dengan nilai yang bagus muncul bersama - sama dengan mata kuliah pilihan dengan nilai yang bagus pada basis data.

5.2. Saran

1. Data latih yang digunakan lebih banyak dan analisis terhadap data harus dilakukan dengan lebih mendalam sehingga data yang digunakan untuk proses *data mining* benar - benar bisa memberikan pengetahuan yang dibutuhkan.
2. Dengan sebaran data yang tidak merata dan dari fakta hasil pengujian, pilihan terakhir yang bisa dicoba adalah menurunkan *min_count* sekecil - kecilnya. Hanya saja hal ini bertentangan dengan prinsip *frequent itemset mining* yaitu mencari *itemset* yang sering muncul. Oleh karena itu perlu dicoba juga model perekomendasi yang lainnya misalnya dengan menggunakan teknik *data mining* klasifikasi.

DAFTAR PUSTAKA

- Aggarwal, S. dan Kaur, R. Comparative Study of Various Improved Versions of Apriori Algorithm. *International Journal of Engineering Trends and Technology (IJETT)*, Vol.4, Issue 4, 2013.
- Baradwaj , B.K. dan Pal, S. Mining Educational Data to Analyze Students Performance. *IJACSA*, Vol.2, No.6, 2011.
- Han, J. dan Kamber, M. *Data Mining, Concepts and Techniques*. Morgan Kaufmann Publisher. US, 227 – 272, 2006.
- Jha, J. dan Ragha, L. Educational Data Mining Using Improved Apriori Algorithm. *International Journal of Information and Computation Technology*, Vol.3, No.5, 411-418, 2013.
- Kaur, C. Association Rule Mining using Apriori Algorithm: A Survey. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol.2, issue 6, 2013.
- Kumar, V. dan Chadha, A. An Empirical Study of the Applications of Data Mining Techniques in Higher Education. *IJACSA*, Vol.2, No.3, 2011.
- Larose, D.T. *Discovering Knowledge In Data. A John Wiley & Son Publication*. US , 180 – 198, 2005.
- Mangla, V., Sarda, C., dan Madra, S. Improving the efficiency of Apriori Algorithm in Data Mining. *International Journal of Engineering and Innovative Technology (IJEIT) Vol. 3, Issue 3*, 2013.
- Mustafidah, H. dan Aryanto, D. Sistem Inferensi Fuzzy untuk Memprediksi Prestasi Belajar Mahasiswa Berdasarkan Nilai Ujian Nasional, Tes Potensi Akademik, dan Motivasi Belajar. *JUITTA ISSN : 2086 – 9398 Vol. II, No.1*, 2012.
- Palanisamy, S.K. Association Rule Based Classification. Worcester Polytechnic Institute. Tesis, 2006.
- Pudjiantoro, T.H. dkk. Penerapan Data Mining Untuk Menganalisa Kemungkinan Pengunduran Diri Calon Mahasiswa Baru. KNSI, 2011.
- Yabing, J. Research of an Improved Apriori Algorithm in Data Mining Association Rules. *International Journal of Computer and Communication Engineering*, Vol.2, No.1, 2013.
- Yadav, S.K. dan Pal, S. Data Mining Application in Enrollment Management: A Case Study. *International Journal of Computer*, 2012.