

## PENERAPAN ALGORITMA APRIORI ASSOCIATION RULE UNTUK ANALISA NILAI MAHASISWA DI UNIVERSITAS GUNADARMA

Margi Cahyanti<sup>1)</sup>, Maulana Mujahidin<sup>2)</sup>, Ericks Rachmat Swedia<sup>3)</sup>

<sup>1)</sup> Sistem Informasi Universitas Gunadarma

<sup>2)</sup> Teknik Komputer Universitas Gunadarma

<sup>3)</sup> Teknik Informatika Universitas Gunadarma

Jl Margonda Raya no 100, Depok

Email : margi@staff.gunadarma.ac.id<sup>1)</sup>, mujahidin@staff.gunadarma.ac.id<sup>2)</sup>, ericks\_rs@staff.gunadarma.ac.id<sup>3)</sup>

### Abstrak

Pada penelitian ini diterapkan metode apriori association rule untuk melihat aturan asosiasi nilai dan matakuliah pada mahasiswa universitas gunadarma jenjang diploma tiga jurusan teknik komputer dari tahun 2000 sampai tahun 2005 untuk nilai D dan nilai E, atau berarti mencari aturan asosiasi untuk nilai yang mengulang pada 75 matakuliah yang telah diambil. Min\_support yang diinput adalah 100 ( $\pm 5\%$  dari rata-rata jumlah data yang sebanyak 1800), dan min\_confidence yang diinput adalah 60%.

Dari hasil analisa terlihat bahwa aturan yang memiliki nilai support vs confidence terbesar terjadi pada aturan IF Praktikum Algoritma dan Pemrograman THEN Pengantar Sistem Komputer. Hal ini berarti bahwa kecenderungan mahasiswa jika mengulang matakuliah praktikum algoritma dan pemrograman juga mengulang matakuliah pengantar sistem komputer, dengan confidence rata-rata sebesar 83,84%. Confidence terbesar terdapat pada tahun 2004 yaitu sebesar 88,67%.

**Kata kunci:** Association Rule, Data Mining, Apriori, Market-Basket Analysis.

### 1. Pendahuluan

Association Rule biasanya dimanfaatkan secara luas dalam proses bisnis diantaranya dalam proses penjualan. Data mining algoritma asosiasi dapat membantu dalam proses penjualan dengan memberikan hubungan antar data penjualan yang dilakukan pelanggan sehingga akan didapat pola pembelian pelanggan atau sering disebut dengan Market-Basket Analysis.[6]

Analisis asosiasi dikenal juga sebagai salah satu teknik data mining yang menjadi dasar dari berbagai teknik data mining lainnya. Tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (*frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien[7].

Association rule mining adalah suatu prosedur untuk mencari hubungan antar item dalam suatu data set yang ditentukan [1]. Association rule meliputi dua tahap [2] :

1. Mencari kombinasi yang paling sering terjadi dari suatu itemset dan 2. Mendefinisikan Condition dan Result (untuk conditional association rule).

Dalam menentukan suatu association rule, terdapat suatu ukuran kepercayaan yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu [3] : 1. *Support* : suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item/itemset dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu item/itemset layak untuk dicari confidence-nya (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa item A dan B dibeli bersamaan) dan 2. *Confidence* : suatu ukuran yang menunjukkan hubungan antar 2 item secara kondisional (misal, seberapa sering item B dibeli jika orang membeli item A).

Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan (threshold) yang ditentukan oleh user. Batasan tersebut umumnya terdiri dari *min\_support* dan *min\_confidence* [3].

Penerapan aturan asosiatif dari analisa pembelian di suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu.

Aturan asosiatif biasanya dinyatakan dalam bentuk :

{roti, mentega} {susu}

(support = 40%, confidence = 50%)

Yang artinya : "50% dari transaksi di database yang memuat item roti dan mentega juga memuat item susu, sedangkan 40% dari seluruh transaksi yang ada di database memuat ketiga item itu." Dapat juga diartikan : "Seorang konsumen yang membeli roti dan mentega punya kemungkinan 50% untuk juga membeli susu. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini." [6]

Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan assosiatif yang memenuhi syarat minimum untuk support (minimum support) dan syarat minimum untuk confidence (minimum confidence).

Dengan menggunakan output dari aplikasi ini, yang berupa aturan asosiasi, ingin diketahui tingkat asosiasi nilai dan matakuliah yang berguna untuk membantu pihak pengambil keputusan dalam menentukan kebijakan.

“Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules.” [1]. Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian frequent Itemset dengan association rule.

Algoritma ini menggunakan knowledge mengenai frequent itemset yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Algoritma apriori memakai pendekatan iterative (level-wise search), dimana k-itemset dipakai untuk menyelidiki (k+1)-itemset. Langkah-langkah dari algoritma ini adalah sebagai berikut [4] :

1. Set k=1 (menunjuk pada itemset ke-1).
2. Hitung semua k-itemset (itemset yang mempunyai k item).
3. Hitung support dari semua calon itemset – filter itemset tersebut berdasarkan perhitungan *minimum support*.
4. Gabungkan semua k-sized itemset untuk menghasilkan calon itemset k+1.
5. Set k=k+1.
6. Ulangi langkah 3-5 sampai tidak ada itemset yang lebih besar yang dapat dibentuk.
7. Buat final set dari itemset dengan menciptakan suatu union dari semua k-itemset.
8. Menghitung *support* / analisa pola frekuensi tinggi. Support aturan asosiasi  $A \Rightarrow B$  adalah :

$$\text{Support } P(A \cap B) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}} \quad (1)$$

9. Menghitung *confidence*. Confidence aturan asosiasi  $A \Rightarrow B$  adalah :

$$\text{Confidence } P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}} \quad (2)$$

10. Menghitung *support vs confidence* dengan rumus :  
 Support vs Confidence = Support x Confidence

Nilai dengan *support vs confidence* terbesar yang merupakan aturan terbaik.

**2. Pembahasan**

Aplikasi dibuat dengan bahasa pemrograman C#. Aplikasi ini terkoneksi dengan server akademik universitas gunadarma. Data yang dibutuhkan harus memiliki struktur seperti tabel dibawah[5] :

**Tabel 1.**Struktur data nilai mahasiswa

NPM	KD_MK	NAMA_MK	NILAI
50401001	KK013210	Pengantar Sistem Komputer	D
50401001	KK013215	Algoritma dan Pemrograman I	B
50401002	KK013210	Pengantar Sistem Komputer	C
50401002	KK013215	Algoritma dan Pemrograman I	E
.....	.....		

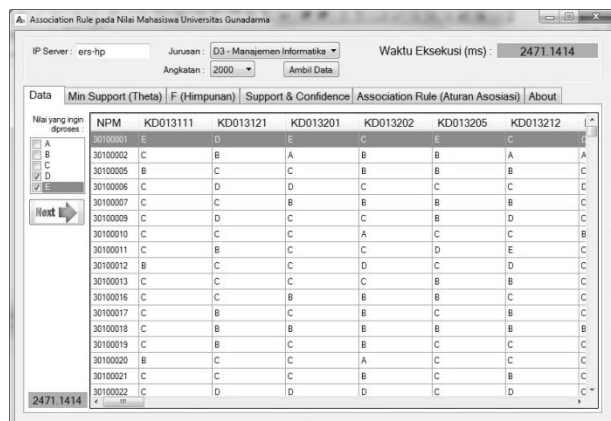
Langkah pertama aplikasi ini adalah mengambil data ke database nilai dan mentransformasi pivot dari struktur data diatas menjadi struktur seperti gambar 1. dibawah (data NPM sebagai baris, data matakuliah dijadikan kolom dan data nilai sebagai isi), seperti contoh tabel 2 dibawah[8].

**Tabel 2.**Struktur Pivot data nilai mahasiswa

NPM	KK013210	KK013215
50401001	D	B
50401002	C	E
.....	.....	

Berikut adalah hasil setelah transformasi pivot dari datatable[7].

Datatable dtProses = Pivot(ref dtNilai, "NPM", "KD\_MK", "NILAI");



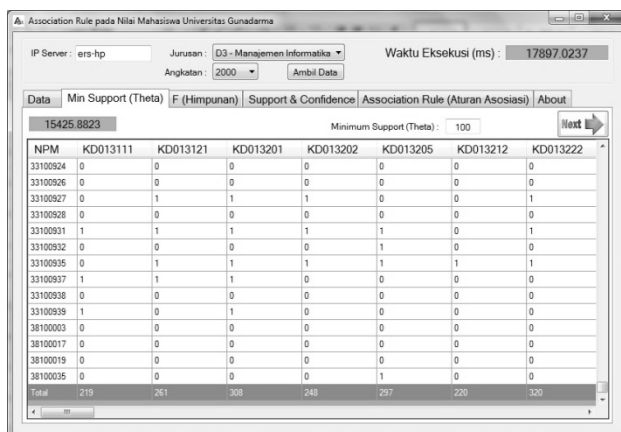
**Gambar 1.**Tampilan hasil pivot data nilai mahasiswa.

Dari data pada gambar 1 diatas setiap nilai yang ingin dilihat association rule-nya ditransform menjadi 1, selain itu 0, misal jika ingin melihat relasi matakuliah yang nilainya D dan E, maka setiap nilai D dan E diubah menjadi 1, sedangkan nilai A, B, dan C diubah menjadi 0, contoh terlihat pada tabel 3 dibawah, kemudian memasukkan *minimum support* yang diinginkan dan menghitung jumlah setiap kolom. Hasil dapat dilihat di gambar 2.

Tabel 3. Transformasi data nilai mahasiswa

NPM	KK013210	KK013215
50401001	1	0
50401002	0	1
Total :	1	1

Berikut adalah output aplikasi untuk merubah nilai A, B, C menjadi 0 dan D, E menjadi 1 dari data table dan menghitung total keseluruhan nilai 1 dari setiap kolom matakuliah.

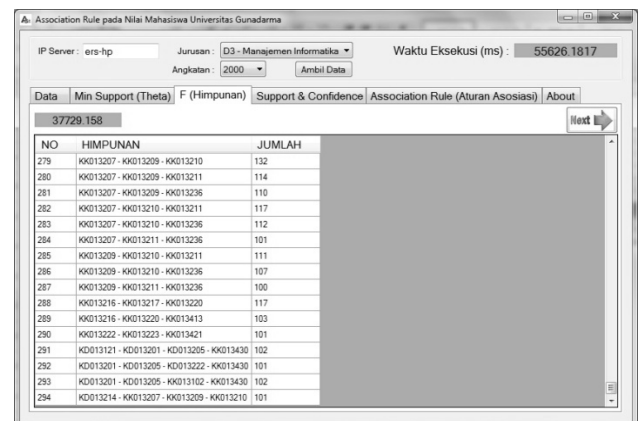


Gambar 2. Tampilan data pivot nilai mahasiswa

Kemudian membentuk k-itemset dan mencari *minimum support*. Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam database. *Minimum support* yang dimasukkan adalah 100 ( $\pm 5\%$  dari rata-rata jumlah data yang sebanyak 1800). Proses dimulai dengan mencari kombinasi himpunan data matakuliah dengan item-set berjumlah 2 : {KD013111, KD013121}, {KD013111, KD013201}, {KD013111, KD013202}, {KD013111, KD013205} ... dst sampai semua kombinasi terpenuhi, kemudian menghitung total dari data yang bernilai 1 dari setiap himpunan. Jika terdapat total data lebih besar atau sama dengan *min\_support*, maka proses berlanjut mencari kombinasi himpunan data matakuliah dengan item-set berjumlah 3 : {KD013111, KD013121, KD013201}, {KD013111, KD013121, KD013202}, {KD013111, KD013121, KD013205} ... dst, kemudian menghitung

total dari data yang bernilai 1 pada masing-masing kolom dari setiap himpunan. Kemudian kembali mengecek total setiap himpunan dengan *min\_support*. Jika terdapat total data lebih besar atau sama dengan *min\_support*, maka proses berlanjut mencari kombinasi himpunan data matakuliah dengan item-set berjumlah 4,5,6, dst, sampai tidak ada total data yang lebih besar atau sama dengan *min\_support*. Hasil terlihat pada gambar 3.

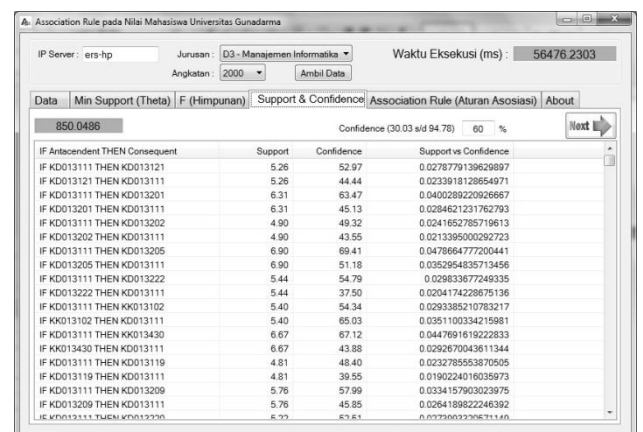
Berikut adalah output dari hasil perhitungan kombinasi data dimulai dari item-set berjumlah 2, menghitung total setiap himpunan dan mengecek total dengan dengan *min\_support* yang diinput (data himpunan akan tersimpan di datatable dtHimpunan).



Gambar 3. Tampilan himpunan dan jumlah masing-masing himpunan

Setelah semua pola frekuensi tinggi (*minimum support*) telah ditemukan, langkah selanjutnya adalah menghitung *confidence* dan menghitung *support vs confidence*, kemudian memasukkan *confidence* yang diinginkan.

Pada gambar 4 adalah output aplikasi untuk menghitung *support* dan *confidence* dan menghitung *support vs confidence* (data ditampilkan pada komponen Listview lvwSupport).

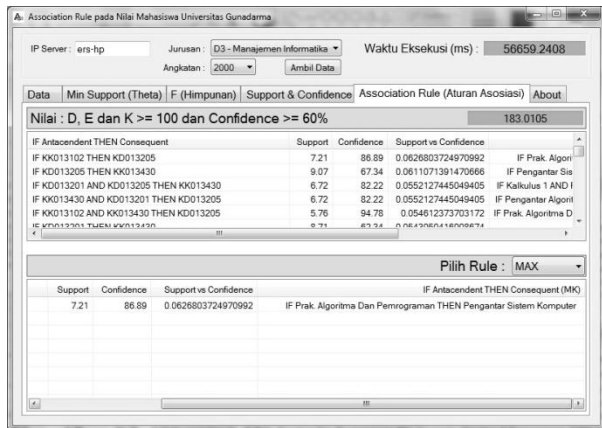


Gambar 4. Tampilan Support, Confidence dan Support vs Confidence.

Setelah nilai *confidence* didapat, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk

*confidence*, dan mencari nilai terbesar *support vs confidence* dari masing-masing aturan. Nilai terbesar dari *support vs confidence* ini yang dijadikan aturan asosiasi. Hasil aturan asosiasi dapat dilihat pada gambar 5.

Berikut adalah output aplikasi untuk mengambil nilai *confidence* yang diinginkan (data ditampilkan pada komponen Listview *lvwFilterConfidence*).



Gambar 5. Tampilan Aturan asosiasi yang dihasilkan.

Analisis dilakukan pada mahasiswa jenjang diploma tiga jurusan manajemen informatika universitas gunadarma dari tahun 2000 sampai tahun 2005 untuk nilai D dan nilai E, atau berarti mencari aturan asosiasi untuk nilai yang mengulang pada 75 matakuliah yang telah diambil. *Min\_support* yang diinput adalah 100 ( $\pm 5\%$  dari rata-rata jumlah data yang sebanyak 1800), dan *min\_confidence* yang diinput adalah 60%.

Tabel 4. Hasil aturan asosiasi dari tahun 2000 – 2005.

ANGKATAN	$\Sigma$ DATA	$\Sigma$ HIMPUNAN	CONFIDENCE (%)	SUPPORT vs CONFIDENCE	ATURAN ASOSIASI
2000	2204	294 Maks F(3)	86,69	0,062	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer
2001	2433	662 Maks F(4)	83,59	0,056	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer
2002	2148	1012 Maks F(5)	87,21	0,092	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer
2003	1454	182 Maks F(3)	69,92	0,079	IF Pengantar Sistem Komputer THEN Algoritma dan Pemrograman I
			78,74	0,075	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer

ANGKATAN	$\Sigma$ DATA	$\Sigma$ HIMPUNAN	CONFIDENCE (%)	SUPPORT vs CONFIDENCE	ATURAN ASOSIASI
2004	1214	116 Maks F(3)	88,67	0,098	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer
2005	1351	168 Maks F(4)	78,14	0,141	IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer

Dari hasil diatas terlihat bahwa dari tahun 2000 sampai 2005 aturan yang memiliki nilai *support vs confidence* terbesar terjadi pada aturan “IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer”. Hal ini berarti bahwa kecenderungan mahasiswa jika mengulang matakuliah “Praktikum Algoritma dan Pemrograman” juga mengulang matakuliah “Pengantar Sistem Komputer”.

Sedikit perbedaan terjadi pada tahun 2003 yang nilai *support vs confidence* terbesar 0,079 adalah pada aturan “IF Pengantar Sistem Komputer THEN Algoritma dan Pemrograman I” dengan *confidence* 69,92%, namun pada aturan “IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer” walaupun memiliki nilai *support vs confidence* yang sedikit lebih kecil yaitu sebesar 0,075 (selisih 0,004) tapi memiliki tingkat *confidence* yang paling tinggi yaitu sebesar 78,74%. Jika *minimum confidence* diubah menjadi 75% maka aturan “IF Pengantar Sistem Komputer THEN Algoritma dan Pemrograman I” tidak akan didapat, karena hanya memiliki *confidence* sebesar 69,92%, karena itu pada tahun 2003 ini aturan yang diambil adalah “IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer”.

Rata-rata *confidence* yang terjadi dari tahun 2000-2005 adalah  $(86,69 + 83,59 + 87,21 + 78,74 + 88,67 + 78,14)/6$  sebesar 83.84%, dengan nilai *confidence* paling tinggi sebesar 88,67% terjadi pada tahun 2004, dan nilai *confidence* terkecil sebesar 78,14% terjadi pada tahun 2005.

### 3. Kesimpulan

Algoritma apriori association rule dapat juga diterapkan untuk melihat asosiasi nilai dan matakuliah pada mahasiswa. Metode ini diterapkan untuk mencari aturan asosiasi pada nilai yang mengulang (nilai D dan nilai E) di 75 matakuliah yang telah diambil pada jurusan manajemen informatika jenjang diploma tiga universitas gunadarma tahun 2000 sampai 2005.

Dari penerapan metode ini, dihasilkan aturan asosiasi. Aturan asosiasi yang terbentuk adalah :

*IF Prak. Algoritma dan Pemrograman THEN Pengantar Sistem Komputer*

dengan confidence sebesar 83,84% dan minimum support 100, sehingga dapat disimpulkan bahwa dengan tingkat kepercayaan  $\pm 80\%$  mahasiswa yang mengulang matakuliah “Praktikum Algoritma dan Pemrograman” juga akan mengulang matakuliah “Pengantar Sistem Komputer”. Hasil ini dapat digunakan sebagai referensi untuk mahasiswa angkatan mendatang.

### Daftar Pustaka

- [1] Jiawei, H, Kamber, Micheline *Data Mining: Concepts and Techniques* Morgan Kaufmann, 2001.
- [2] Ulmer, David; “*Mining an Online Auctions Data Warehouse. The Mid-Atlantic Student Workshop on Programming Languages and Systems*”. 19 April 2002. Pace University. <<http://csis.pace.edu/csis/masplas/p8.pdf>>. [Diakses: 25 November 2016]
- [3] Budhi G.S, Lim, R dan Prayitno, “*Penggunaan Metode Fuzzy C-Covering untuk Analisa Market Basket pada Supermarket*”, Jurnal Informatika Vol. 6, No. 1. Mei 2005.
- [4] Larose, D.T, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, Mei 2005.
- [5] Tang, Z dan MacLennan, J, . *Data Mining with SQLServer 2005*, Wiley Publishing, Indianapolis, 2005.
- [6] Dilly, Ruth, *Data Mining: An Introduction*, Queens University Belfast, 2004.
- [7] R Agrawal; R Srikant “*Fast Algorithms for Mining Association Rules*”, Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.
- [8] R. Agrawal, T. Imielinski, and A. Swami December “*Database mining: A performance perspective*”, IEEE Transactions on Knowledge and Data Engineering, 1993.

### Biodata Penulis

**Margi Cahyanti**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi Universitas Gunadarma, lulus tahun 1992. Memperoleh gelar Magister Manajemen Sistem Informasi (MMSi) Program Pasca Sarjana Magester Manajemen Sistem Informasi Universitas Gunadarma, lulus tahun 2000. Saat ini menjadi Dosen di Universitas Gunadarma

**Maulana Mujahidin**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Komputer Universitas Gunadarma, lulus tahun 2003. Memperoleh gelar Magister Teknik (MMSi) Program Pasca Sarjana Magester Teknik Universitas Gunadarma, lulus tahun 2009. Saat ini menjadi Dosen di Universitas Gunadarma

**Ericks Rachmat Swedia**, memperoleh gelar Sarjana Teknik (ST), Jurusan Teknik Informatika Universitas Gunadarma, lulus tahun 1999. Memperoleh gelar Magister Manajemen Sistem Informasi (MMSi) Program Pasca Sarjana Magester Manajemen Sistem Informasi Universitas Gunadarma, lulus tahun 2008. Saat ini menjadi Dosen di Universitas Gunadarma.

