

# Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

Satrio Prasajo ([st.prasajo@gmail.com](mailto:st.prasajo@gmail.com)), Shaufiah, ST., MT ([ufi@telkomuniversity.ac.id](mailto:ufi@telkomuniversity.ac.id)), Hetti Hidayati, S.Kom., MT ([htt@telkomuniversity.ac.id](mailto:htt@telkomuniversity.ac.id)), Telkom University

---

## Abstrak

Pada proses perkuliahan, setiap mahasiswa pada proses registrasi wajib melakukan tahapan perwalian. Mahasiswa biasanya mengajukan mata kuliah apa saja yang akan diambilnya kepada dosen wali, setelah itu dosen wali akan memberikan saran terkait mata kuliah apa saja yang sebaiknya diambil, terutama mata kuliah pilihan. Demi memudahkan proses tersebut, dibuatlah sebuah sistem yang menangani perekomendasi. Sistem ini memanfaatkan teknik *data mining* menggunakan algoritma *Recursive Elimination* (Relim). Data yang diproses akan dipadatkan menggunakan parameter *minimum support*. Implementasi yang dilakukan terbukti dapat menghasilkan rekomendasi mata kuliah pilihan dengan aturan asosiasi yang menggunakan nilai *support* dan *confidence* sebagai acuannya.

**Kata kunci :** rekomendasi, Relim, *minimum support*, *support*, *confidence*

---

## Abstract

Every college student at the registration stage should have a consultation with their lecturer. The student usually submit their course plan for the next semester to their lecturer, then the lecturer would give a recommendation especially for the certain courses. To make it simple, we will build a system that could give a course recommendation. The data mining technique, that called Recursive Elimination Algorithm (Relim) will be planted in the system. Processed data will be denser because of minimum support parameter. This implementation proves can produce a course recommendation using association rules with support and confidence value as a reference.

**Keywords :** recommendation, Relim, minimum support, support, confidence

---

## 1. Pendahuluan

Rencana studi seorang mahasiswa dalam satu semester ke depan sangat ditentukan oleh pengambilan mata kuliah yang dilakukannya. Bagi mahasiswa tingkat akhir, tidak hanya mata kuliah wajib saja yang perlu diambil, namun juga mata kuliah pilihan sebagai dasar pengerjaan Tugas Akhir nantinya. Pada jurusan Teknik Informatika IT Telkom terdapat 3 buah Kelompok Keahlian (KK), yaitu SIDE, ICM, dan Telematik. Tiap kelompok keahlian memiliki mata kuliah penunjang yang berbeda satu sama lain. Di sinilah sering terjadi permasalahan dimana mahasiswa terkadang mengambil mata kuliah pilihan tanpa melihat riwayat nilai mata kuliah wajib pendukungnya. Contohnya adalah saat seorang mahasiswa mengambil Jaringan Komputer Lanjut (Jarkomlan) padahal nilai mata kuliah Jarkom pada semester sebelumnya

kurang memuaskan. Dosen wali seharusnya dapat mencegahnya pada saat proses perwalian, namun kejadian seperti itu seringkali lolos begitu saja.

Sistem yang ada saat ini masih berupa perwalian manual, di mana seorang dosen wali memeriksa riwayat nilai mahasiswanya satu per satu terlebih dulu, barulah ia dapat memberi rekomendasi mata kuliah pilihan apa saja yang sebaiknya diambil. Sistem seperti ini jelas menguras waktu dan tenaga. Kendala lain dihadapi para mahasiswa yang harus kembali dari daerah asalnya hanya sekedar untuk bertatap muka dengan dosen wali. Seperti yang kita ketahui, jadwal perwalian biasanya dilakukan pada saat libur semester. Mengingat pekerjaan seorang dosen wali tidak mudah, karena rata-rata seorang dosen wali memiliki setidaknya 40 anak wali, diperlukan sebuah

## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

sistem yang dapat merekomendasikan mata kuliah pilihan secara otomatis.

Sistem yang dibangun akan memanfaatkan teknik *data mining* yang telah terbukti dalam beberapa tahun terakhir dapat menangani jumlah data yang begitu besar, seperti pada analisis pasar, deteksi penipuan, pengendalian produksi, dan masih banyak lagi [1]. Salah satu topik terpenting dalam data mining yaitu *association rules* / aturan asosiasi yang sangat penting bagi pengambilan keputusan [2]. Berdasarkan studi literatur, aturan asosiasi telah terbukti dapat menyelesaikan beberapa masalah perkomendasi, seperti pada jurnal yang ditulis oleh Nugroho Wandy, Rully Hendrawan, dan Akhmad Mukhlason berjudul "Pengembangan Sistem Rekomendasi Penelusuran Buku dengan Penggalian *Association Rule* Menggunakan Algoritma Apriori (studi Kasus Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur)". Metode dan algoritma tersebut menghasilkan transaksi-transaksi peminjaman buku dengan strong association (keterkaitan kuat) antar buku dalam transaksi yang digunakan sebagai rekomendasi peminjaman buku.

Algoritma asosiasi yang digunakan pada Tugas Akhir ini yaitu algoritma *Recursive Elimination (Relim)* karena kesederhanaan strukturnya dengan langsung memproses transaksi. Pada dasarnya, seluruh pekerjaan pemrosesan transaksi dilakukan dalam satu fungsi rekursif sederhana, yang cukup ditulis dengan beberapa baris kode saja [3]. Diharapkan sistem yang baru ini nantinya akan dapat memberi rekomendasi mata kuliah pilihan dengan cepat dan tepat sehingga dapat menghemat waktu dan tenaga baik dosen wali maupun mahasiswa.

## 2. Landasan Teori

### 2.1 Data Mining

Berdasarkan jurnal Seminar Nasional Informatika 2009 yang ditulis oleh Azhari dan Anshori dari FMIPA UGM dengan judul "Pendekatan Aturan Asosiasi untuk Analisis Pergerakan Saham", *data mining* adalah eksplorasi dan analisis terhadap sejumlah data dengan tujuan untuk menemukan pola dan aturan yang sangat penting (Dunham, 2003; Truban, dkk. 1988).

Menurut Han, dkk. (2001), *data mining* merupakan penerapan bidang ilmu

interdisipliner, statistik, *database*, *machine learning*, *pattern recognition*, kecerdasan buatan, dan visualisasi. Bidang-bidang tersebut memiliki peranan yang saling berhubungan, sehingga sangat sulit untuk memberikan batasan yang jelas diantara masing-masing disiplin ilmu tersebut dengan *data mining* [2].

### 2.2 Association Rules Mining

Masih berdasarkan jurnal Azhari dan Anshori dari FMIPA UGM "Pendekatan Aturan Asosiasi untuk Analisis Pergerakan Saham", *association rules mining* (aturan mining asosiasi) berfungsi untuk menemukan asosiasi antar variabel, korelasi atau suatu struktur diantara item atau objek-objek didalam database transaksi, database relasional, maupun pada penyimpanan informasi lainnya (Agrawal dan Srikant, 1994; Tan dkk., 2006).

Sebuah aturan asosiasi adalah implikasi dari  $X \Rightarrow Y$ , dimana  $X \subseteq I$ ,  $Y \subseteq I$ , dan  $X \cap Y = \emptyset$ . Aturan  $X \Rightarrow Y$  berada di dalam himpunan transaksi  $D$  dengan kepercayaan (confidence)  $c$ , jika  $c\%$  dari transaksi dalam  $D$  yang ada  $X$  terdapat juga  $Y$ . Aturan  $X \Rightarrow Y$  memiliki dukungan (support)  $s$  di dalam set transaksi  $D$ , jika  $s\%$  transaksi dalam  $D$  terdapat  $X \cup Y$  (1).

Support *item set* (1) adalah jumlah transaksi  $T$  yang didalamnya terdapat (1). Confidence mengukur seberapa besar ketergantungan suatu item dengan item yang lainnya. Frequent *item set* adalah *item set* yang memiliki nilai support lebih atau sama besar dengan nilai *minimum support* yang ditentukan (2).

Dimisalkan suatu aturan  $R : X \Rightarrow Y$ , maka:

$$\text{Support}(X \Rightarrow Y) = \frac{\text{support}(\{X \cup Y\})}{\text{support}(D)} \quad (1)$$

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{support}(\{X \cup Y\})}{\text{support}(X)} \quad (2)$$

Sebuah aturan (rule) biasanya terdiri dari dua bagian, yaitu kondisi (condition) dan hasil (result), biasanya disajikan dalam pernyataan sebagai berikut :

Jika **kondisi** maka **hasil** (If **condition** then **result**) (3)

## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

### 2.3 FP-Growth Algorithm

Menurut Christian Borgelt dalam jurnalnya yang berjudul “*Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination*”, algoritma FP-Growth merupakan inspirasi dari algoritma Relim. FP-growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. FP-growth menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori [7].

#### 2.3.1 Analisis Pola Frekuensi Tinggi dengan Algoritma FP-Growth

Pada penentuan *frequent itemset* terdapat 2 tahap proses yang dilakukan yaitu: pembuatan FP-tree dan penerapan algoritma FP-growth untuk menemukan *frequent itemset*. Struktur data yang digunakan untuk mencari *frequent itemset* dengan algoritma FP-growth adalah perluasan dari penggunaan sebuah pohon *prefix*, yang biasa disebut adalah FP-tree. Dengan menggunakan FP-tree, algoritma FP-growth dapat langsung mengekstrak *frequent itemset* dari FP-tree yang telah terbentuk dengan menggunakan prinsip *divide and conquer*.

#### 2.3.2 Pembentukan FP-Tree

FP-tree merupakan struktur penyimpanan data yang dimampatkan. FP-tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-tree. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data FP-tree semakin efektif. Adapun FP-tree adalah sebuah pohon dengan definisi sebagai berikut:

- a. FP-tree dibentuk oleh sebuah akar yang diberi label null, sekumpulan sub-tree yang beranggotakan item-item tertentu, dan sebuah tabel frequent header.
- b. Setiap simpul dalam FP-tree mengandung tiga informasi penting, yaitu label item, menginformasikan jenis item yang direpresentasikan simpul tersebut, support count, merepresentasikan jumlah lintasan

transaksi yang melalui simpul tersebut, dan pointer penghubung yang menghubungkan simpul-simpul dengan label item sama antar-lintasan, ditandai dengan garis panah putus-putus.

#### 2.3.3 Penerapan Algoritma FP-Growth

Setelah tahap pembangunan FP-tree dari sekumpulan data transaksi, akan diterapkan algoritma FP-growth untuk mencari *frequent itemset* yang signifikan. Algoritma FP-growth dibagi menjadi tiga langkah utama, yaitu:

1. Tahap Pembangkitan Conditional Pattern Base  
*Conditional Pattern Base* merupakan *subdatabase* yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-tree yang telah dibangun sebelumnya.
2. Tahap Pembangkitan Conditional FP-tree  
 Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar sama dengan *minimum support count* akan dibangkitkan dengan *conditional FP-tree*.
3. Tahap Pencarian *Frequent itemset*  
 Apabila *Conditional FP-tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi item untuk setiap *conditional FP-tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan FP-growth secara rekursif.

Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

2.4 Recursive Elimination Algorithm (Relim)

Algoritma Recursive Elimination (Relim) merupakan salah satu algoritma asosiasi yang pertama kali diperkenalkan oleh Christian Borgelt dari Department of Knowledge Processing and Language Engineering Universitas Magdeburg di Jerman sekitar tahun 2005. Algoritma ini terinspirasi dari teknik asosiasi lainnya, yaitu FP-growth dan H-mine, berfungsi untuk menemukan frequent item sets. Relim bekerja tanpa memerlukan prefix trees ataupun struktur data yang rumit, dengan kata lain algoritma ini langsung mengeksekusi transaksi [3].

2.4.1 Preprocessing

Pada tahap ini, dilakukan penghitungan frekuensi tiap item transaksi. Minimum support yang diinginkan kita tentukan, sehingga selanjutnya kita dapat menghilangkan item yang infrequent.

a d f		a d
c d e	g 1	e c d
b d	f 2	b d
a b c d	e 3	a c b d
b c	a 4	c b
a b d	c 5	a b d
b d e	b 7	e b d
b c e g	d 8	e c b
c d f		c d
a b d		a b d

Gambar 0.1 Database transaksi (kiri), frekuensi item (tengah), dan database yang telah direduksi (kanan)

2.4.2 Recursive Processing

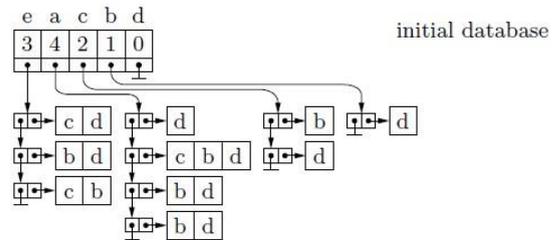
Pada tahap ini dimulailah pemrosesan secara rekursif terhadap database yang telah direduksi tadi.

Langkah pertama adalah dengan menghitung jumlah frequent item yang terdapat pada prefix (awalan) dari seluruh item set.

a	d
e	c d
b	d
a	c b d
c	b
a	b d
e	b d
e	c b
c	d
a	b d

Gambar 0.2 Sorted Transaction

Terbentuklah initial database yang nantinya akan dilakukan recursive elimination terhadapnya. Frequent item disusun dari yang berjumlah kecil ke besar seperti pada gambar 2.2. Saat ini, posisi pointer berada pada item „e“.



Gambar 0.3 Initial Database

Selanjutnya dilakukan eliminasi terhadap prefix e, pointer berpindah ke item setelah prefix e. Item-item yang semula ber-prefix e kemudian bergabung dengan item set yang memiliki prefix sama. Proses ini berlangsung terus menerus hingga hanya tersisa 1 buah item set saja.

## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

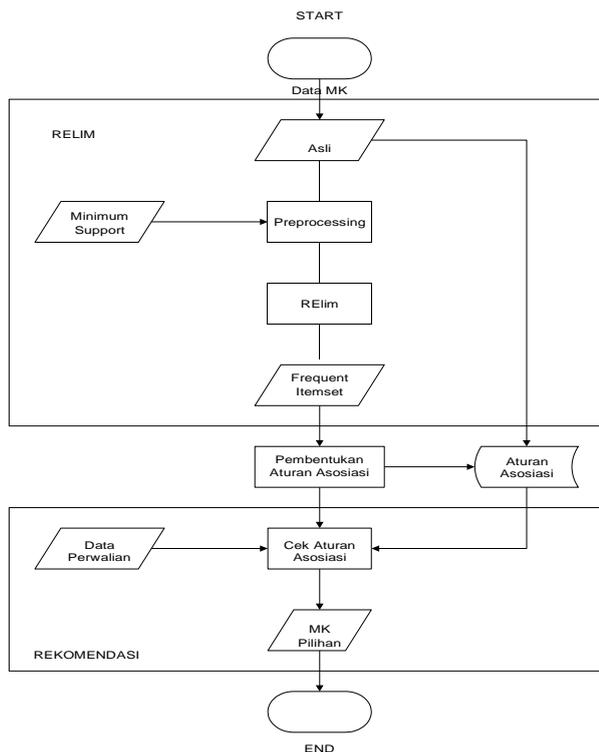
### 3. Perancangan Sistem

#### 3.1 Deskripsi Umum Sistem

Pada jurnal ini akan dirancang suatu sistem yang memudahkan perwalian, khususnya dalam proses pengambilan mata kuliah pilihan oleh mahasiswa. Sistem akan membaca mata kuliah beserta nilai tiap mahasiswa yang tersimpan dalam *database*. Selanjutnya data tersebut akan di-*preprocessing* agar memenuhi *minimum support* sekaligus menghilangkan mata kuliah yang tidak relevan dengan mata kuliah pilihan yang ada. Algoritma *Relim* digunakan untuk mendapatkan *frequent itemset* yang akan dicari *rule*-nya. *Rule* inilah yang nantinya akan digunakan digunakan untuk per Rekomendasi.

#### 3.2 Perancangan Proses

Perancangan proses dari sistem yang dibuat dalam bentuk flowchart dapat dilihat pada gambar 3.1 berikut.



Gambar 0.2 Gambaran Umum Sistem

### 3.3 Deskripsi Kasus

#### 3.3.1 Preprocessing

Tahap ini berupa *data cleaning*, *data integration*, *data transformation*, dan *data reduction* untuk menghilangkan *noise* dan inkonsistensi, serta mereduksi ukuran data agar lebih sesuai dengan kebutuhan sistem. Data mentah didapat dari UPT Sisfo IT Telkom (sekarang Telkom University).

Berikut ini adalah contoh data mentah yang didapat.

Tabel 0.1 Contoh Data Mentah

NIM	NIL AI	KODE KULIAH	THN AJARAN	SEMES TER	NIL AI
4689582	A	CS3113	1112	2	A
4689582	A	CS3243	1112	2	A
b39156826	C	DU2002	1011	2	C
b39156826	C	SE2423	1011	2	C
.....	.....	.....	.....	.....	.....
N	.....	.....	.....	.....	.....

Dari tabel di atas terlihat bahwa masih ada mata kuliah yang bukan merupakan mata kuliah informatika, yaitu DU2002 (Pengling). Mata kuliah yang tidak relevan inilah yang akan dihilangkan. Selain itu, untuk memudahkan pembacaan data, NIM yang berupa kode enkripsi tersebut diubah ke dalam format 113000000, 113000001, dan seterusnya. Setelah itu dilakukan tahapan preprocessing selanjutnya, yaitu

mengeliminasi data yang tidak memenuhi *minimum support*. Sebelumnya dilakukan inisiasi terlebih dahulu terhadap data yang sudah diolah sebelumnya. Hal ini bertujuan untuk menjadikan kode mata kuliah dan nilai sebuah *item*

Tabel 0.2 Contoh Data Akademik yang Telah Diinisiasi

NIM	Mata Kuliah & Nilai			
113000000	CDG4I3-B	CSG2A3-B	CS4713-C	
113000001	CS3113-B	CSG2A3-B	CS1113-A	
113000002	CS4333-A	CSG2A3-B		
113000003	CDG4I3-B	CS4333-A	CS3113-B	CSG2A3-B
113000004	CS4333-A	CS3113-B		
113000005	CDG4I3-B	CS4333-A	CSG2A3-B	
113000006	CS4333-A	CSG2A3-B	CS1113-A	
113000007	CS4333-A	CS3113-B	CS1113-A	CS4383-C
113000008	CS3113-B	CSG2A3-B	CS4713-C	
113000009	CDG4I3-B	CS4333-A	CSG2A3-B	

Guna memudahkan penulisan di buku,

Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination  
Algorithm (Relim)

penulis lalu melakukan inisiasi kembali

Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

terhadap mata kuliah dan nilai. Inisiasi yang dilakukan dapat dilihat pada tabel 3.3 berikut.

Tabel 0.3 Hasil Inisiasi

NIM	Mata Kuliah & Nilai			
113000000	A	D	F	
113000001	C	D	E	
113000002	B	D		
113000003	A	B	C	D
113000004	B	C		
113000005	A	B	D	
113000006	B	D	E	
113000007	B	C	E	G
113000008	C	D	F	
113000009	A	B	D	

Tentukan nilai *minimum support* untuk menghilangkan *infrequent item*. Pada contoh berikut ditentukan bahwa *minsup*=3.

Tabel 0.4 Frekuensi Item

Makuni	Support
G	1
F	2
E	3
A	4
C	5
B	7
D	8

Setelah menghilangkan *infrequent items*, dibentuklah *reduced database* dengan menyusun kembali NIM serta mata kuliah dan nilai seperti semula namun dengan urutan *item set* yang berbeda. *Item set* disusun secara *ascending*, yaitu mulai dari *item* berfrekuensi terendah hingga tertinggi. *Item* yang tidak memenuhi *minimum support* tidak disertakan kembali.

Tabel 0.4 Data Akademik yang Telah Direduksi

NIM	Mata Kuliah & Nilai			
113000000	A	D		
113000001	E	C	D	
113000002	B	D		
113000003	A	C	B	D
113000004	C	B		

113000005	A	B	D	
113000006	E	B	D	
113000007	E	C	B	
113000008	C	D		
113000009	A	B	D	

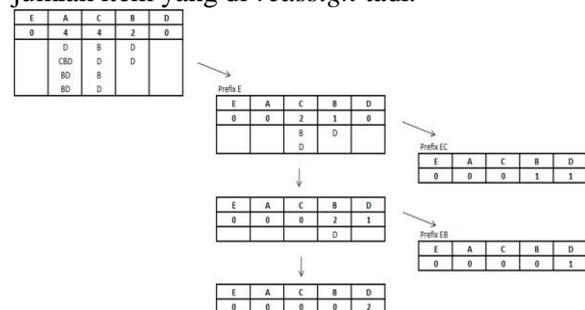
3.3.2 Recursive Processing

Pada tahap ini akan dilakukan pengolahan terhadap data akademik baru yang tadi dibentuk menggunakan algoritma Relim. Tujuannya yaitu untuk menemukan *frequent itemset*. Data akademik tersebut dapat dibentuk ke dalam *initial database* berupa *array* sebagai berikut.

Tabel 0.5 Initial Database

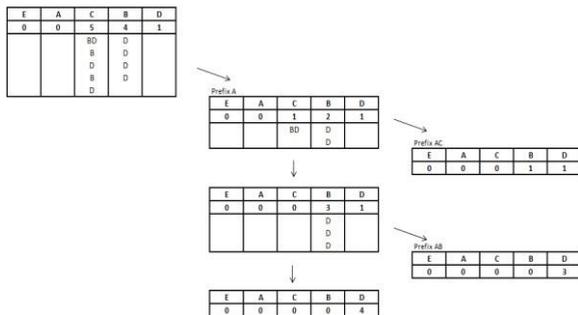
E	A	C	B	D
3	4	2	1	0
CD	D	B	D	
BD	CBD	D		
CB	BD			
	BD			

*Recursive elimination* dilakukan dengan mengeliminasi *prefix* satu per satu mulai dari kiri ke kanan *array*. Pada kasus ini, yang pertama dieliminasi adalah *prefix* E. Gambar 3.2 berikut menunjukkan proses eliminasi yang terjadi. *Counter* pada *prefix* yang dieliminasi akan menunjukkan angka 0, sedangkan *item-item* yang mengikutinya akan di-*reassign* ke dalam *prefix-prefix* yang letaknya di kanan. *Counter* akan menunjukkan jumlah *item* yang di-*reassign* tadi.

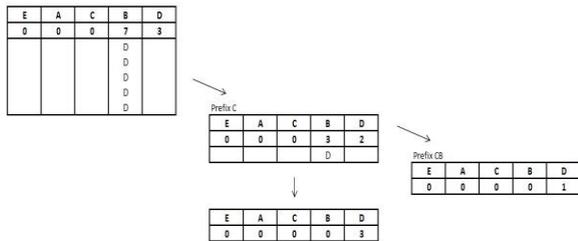


Gambar 0.3 Eliminasi Prefix E

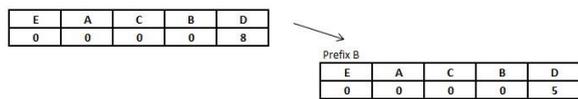
## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)



Gambar 0.4 Eliminasi Prefix A



Gambar 0.5 Eliminasi Prefix C



Gambar 0.6 Eliminasi Prefix B

Akhirnya didapat seluruh *frequent itemset* yaitu E, A, C, B, D, AB, AD, CB, CD, BD, dan ABD. *Frequent itemset* ini akan digunakan pada proses selanjutnya, yaitu pencarian aturan asosiasi.

### 3.3.3 Mencari Aturan Asosiasi

Setelah *frequent itemset* diketahui, kita bisa mencari aturan asosiasinya. Setiap aturan asosiasi haruslah berbentuk  $R : X \Rightarrow Y$ , maka kita dapat menghilangkan *frequent itemset* yang hanya terdiri dari 1 buah item saja, yaitu E, A, C, B, dan D, sehingga tersisa AB, AD, CB, CD, BD, dan ABD. Untuk contoh kasus ini penulis mengambil contoh 3 buah *frequent itemset* saja, yaitu AB, CD, dan ABD.

Tabel 0.6 Pencarian Aturan Asosiasi

Kandidat Aturan	Confidenc
A → B	—
B → A	—
C → D	—

D → C	—
AB → D	—
AD → B	—
BD → A	—

### 3.3.4 Proses Rekomendasi

Proses rekomendasi dilakukan dengan terlebih dahulu menyaring aturan yang merekomendasikan hanya mata kuliah pilihan. Pada kasus di atas terdapat 2 aturan yang memberi rekomendasi mata kuliah pilihan, yaitu  $A \rightarrow B$  dan  $AD \rightarrow B$ . mata kuliah pilihan dengan kode inisiasi B tersebut yaitu CS4333-A (Data Mining).

Mata kuliah yang telah diambil oleh mahasiswa selanjutnya dimasukkan ke dalam aturan-aturan tersebut. Misal seorang mahasiswa telah mengambil mata kuliah A (CDG4I3-B / AUSI) dan D (CSG2A3-A / Alstrukdat) maka dia akan mendapat rekomendasi mata kuliah pilihan CS4333 (Data Mining).

## 4. Pengujian dan Analisis Sistem

### 4.1 Data Set

Data set yang digunakan pada pengujian ini adalah data akademik mahasiswa IT Telkom angkatan 2008 s/d 2011. Di dalam data ini terdapat 1532 NIM mahasiswa dan 592 jenis kombinasi mata kuliah dan nilai. Nantinya data ini akan dibagi menjadi *data training* dan *data testing* di dalam sistem.

### 4.2 Pengujian Sistem

Pada tahap ini dilakukan beberapa skenario pengujian untuk mengetahui kapabilitas sitem yang telah dibangun. Adapun tujuan dari skenario pengujian ini adalah sebagai berikut.

1. Mengetahui pengaruh nilai *minimum support* dan pembagian data *training* serta *testing* terhadap waktu eksekusi, jumlah *frequent itemset* serta jumlah aturan asosiasi yang dihasilkan.
2. Mengetahui pengaruh nilai *minimum support* dan pembagian data *training* serta

Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

testing terhadap akurasi berdasarkan support dan confidence-nya.

- Menentukan nilai *minimum support* dan pembagian data yang optimal untuk sistem ini.

Untuk mencapai tujuan di atas, dilakukanlah beberapa skenario pengujian sebagai berikut.

- Menganalisis pengaruh nilai *minimum support* dan pembagian data *training* serta *testing* terhadap waktu eksekusi, jumlah *frequent itemset* dan aturan asosiasi yang dihasilkan serta akurasi. Seluruh data akademik mahasiswa akan diuji dengan nilai *minimum support* antara 20 hingga 100 dan persentase data *training* antara 30% hingga 90%. Melalui pengujian tersebut nantinya akan didapat perbedaan hasil waktu eksekusi dan jumlah *frequent itemset* serta aturan asosiasi yang terbentuk.
- Melakukan analisis nilai *minimum support* dan persentase data *training* yang optimal untuk sistem. Nilai optimal yang dimaksud yaitu di mana kombinasi kedua nilai masukan tersebut dapat menghasilkan tingkat akurasi yang paling besar.

4.3 Analisis Hasil Pengujian

4.3.1 Pengujian dengan Nilai Minimum Support dan Persentase Data Training Berubah

4.3.1.1 Pengujian Waktu Terbentuknya Aturan Asosiasi

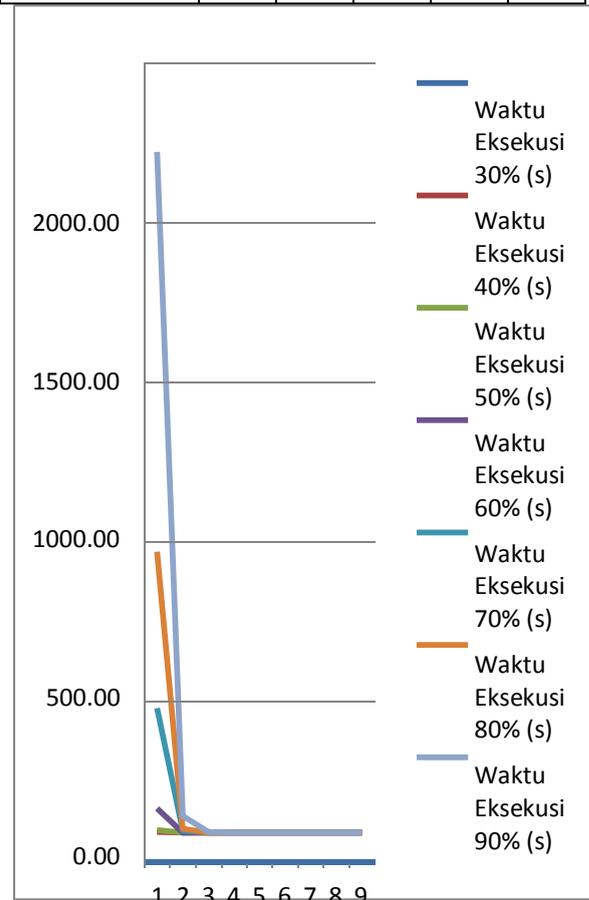
Pengujian ini dilakukan guna menganalisis pengaruh perubahan nilai

*minimum support* terhadap waktu terbentuknya aturan asosiasi. Adapun hasil pengujiannya dapat dilihat pada tabel dan grafik di bawah berikut ini.

Tabel 0.2 Waktu Terbentuknya Aturan Asosiasi dengan Data Training 30%

M i n i m u m s u p p	Wakt u Ekse kusi 30% (s)	Wakt u Ekse kusi 40% (s)	Wakt u Ekse kusi 50% (s)	Wakt u Ekse kusi 60% (s)	Wakt u Ekse kusi 70% (s)	Wakt u Ekse kusi 80% (s)	Wakt u Ekse kusi 90% (s)
2				76.3	390.	880.	2133
0	0.07	1.76	8.76	0	18	63	.40
3						13.1	53.1
0	0.01	0.05	0.24	0.98	5.47	6	5
4							
0	0.00	0.01	0.04	0.09	0.41	1.13	2.36
5							
0	0.00	0.00	0.01	0.02	0.09	0.14	0.38

0							
6							
0	0.00	0.00	0.01	0.01	0.02	0.07	0.10
7							
0	0.00	0.00	0.00	0.01	0.01	0.02	0.04
8							
0	0.00	0.00	0.00	0.00	0.01	0.01	0.01
9							
0	0.00	0.00	0.00	0.00	0.01	0.01	0.01
1							
0							
0	0.00	0.00	0.00	0.00	0.00	0.01	0.01



Gambar 0.7 Grafik Waktu Terbentuknya Aturan Asosiasi

Setelah melihat tabel di atas beserta dengan grafiknya yang menunjukkan pengaruh *minimum support* terhadap lamanya waktu terbentuk aturan asosiasi, kita dapat menarik kesimpulan bahwa semakin besar nilai *minimum support*, maka waktu komputasi yang dibutuhkan untuk membentuk aturan asosiasi semakin kecil. Nilai *minimum support* yang semakin besar mengakibatkan semakin banyak pula itemset yang tereliminasi sehingga hanya menyisakan *frequent itemset* saja. Dengan jumlah *frequent itemset* yang makin sedikit inilah, aturan asosiasi yang

Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

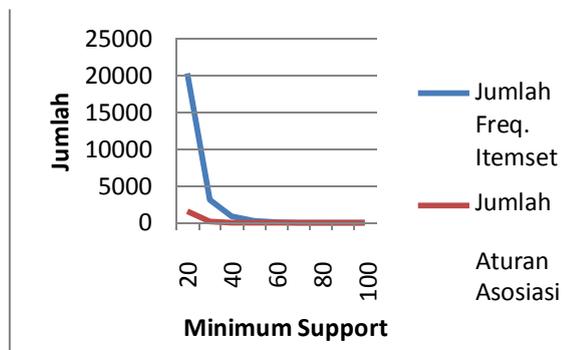
dihasilkan pun semakin sedikit sehingga waktu komputasi yang dibutuhkan semakin cepat.

**4.3.1.2 Pengujian Jumlah Frequent Itemset dan Aturan Asosiasi yang Terbentuk**

Pengujian ini dilakukan untuk mengetahui pengaruh nilai *minimum support* terhadap jumlah *frequent itemset* serta aturan asosiasi yang dihasilkan. Adapun hasil pengujiannya dapat dilihat pada tabel grafik berikut ini.

Tabel 0.2 Jumlah *Frequent itemset* dan Aturan Asosiasi yang Terbentuk dengan Data *Training* 50%

Minsu p	Jumlah Freq. Itemset	Jumlah Aturan Asosiasi
20	20324	1578
30	3173	209
40	915	37
50	279	6
60	94	3
70	32	0
80	4	0
90	1	0
100	0	0



Gambar 0.2 Grafik Jumlah *Frequent itemset* dan Aturan Asosiasi yang Terbentuk dengan Data *Training* 50%

Dari hasil pengujian yang ditunjukkan oleh tabel dan grafik di atas menunjukkan bahwa nilai *minimum support* berpengaruh terhadap jumlah *frequent itemset* yang dihasilkan. Semakin besar nilai *minimum support*, maka akan semakin sedikit *frequent itemset* yang dihasilkan. Sedangkan jumlah *frequent itemset* yang dihasilkan berpengaruh terhadap jumlah aturan asosiasi yang terbentuk. Semakin sedikit jumlah *frequent*

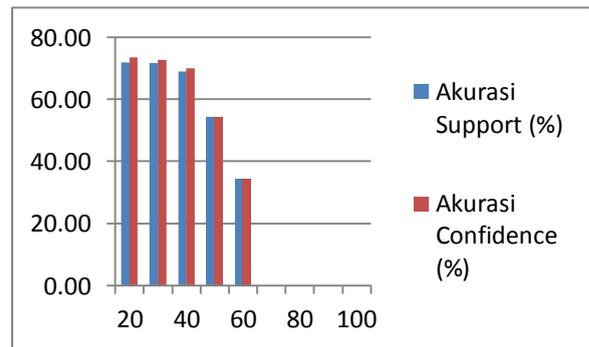
*itemset*, maka akan semakin sedikit pula aturan asosiasi yang terbentuk.

**4.3.1.3 Pengujian Akurasi**

Pengujian ini dilakukan dengan membandingkan kecocokan antara mata kuliah pilihan yang telah diambil mahasiswa dengan mata kuliah pilihan yang direkomendasikan sistem. Hasil pengujiannya adalah sebagai berikut.

Tabel 4.3 Akurasi pada Data *Training* 50%

Minsu p	Akurasi Support (%)	Akurasi Confidence (%)
20	71.79	73.48
30	71.71	72.68
40	68.93	69.97
50	54.11	54.11
60	34.48	34.48
70	0.00	0.00
80	0.00	0.00
90	0.00	0.00
100	0.00	0.00



Gambar 4.3 Grafik Akurasi pada Data *Training* 50%

**4.3.1.4 Analisis Nilai Minimum Support dan Persentase Data Training yang Optimal**

Nilai *minimum support* dan persentase data *training* yang optimal dilihat dari kombinasi dari dua variabel tersebut yang menghasilkan tingkat akurasi/kecocokan yang paling besar. Dapat dilihat pada sub-bab 4.3.1.3 mengenai pengujian akurasi bahwa nilai *minimum support* terendah (20) selalu menghasilkan akurasi yang paling besar.

Tabel 0.4 Akurasi Tertinggi Tiap Pengujian

Minsu p	Data Training (%)	Akurasi Support (%)	Akurasi Confidence (%)
20	71.79	73.48	73.48

## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

20	30	73.72	72.04
20	40	71.31	71.71
20	50	71.79	73.48
20	60	71.71	72.20
20	70	71.63	73.08
20	80	71.71	72.76
20	90	71.63	72.76

Sedangkan untuk persentase data *training*, tidak ditemukan selisih yang signifikan untuk menentukan salah satu nilai persentase data *training* sebagai yang paling optimal. Namun, pada pengujian diketahui bahwa data *training* sebesar 50% dapat mencakup mata kuliah pilihan paling banyak dibanding lainnya.

#### 4.4 Rekomendasi

Berdasarkan pengujian yang telah dilakukan pada sub-bab sebelumnya, dihasilkan nilai *minimum support* yang dianggap sebagai nilai paling optimal. Data aturan asosiasi yang akan digunakan adalah yang memiliki *minimum support* 20 dan dengan persentase data *training* sebanyak 50%. Hal ini dikarenakan kondisi tersebut dapat mencakup mata kuliah pilihan paling banyak yaitu sebanyak 13 dari 15 mata kuliah pilihan.

Dari data aturan asosiasi yang dihasilkan, terdapat mata kuliah wajib yang diambil mahasiswa dengan mata kuliah pilihan tertentu. Artinya, mahasiswa yang telah mengambil mata kuliah wajib dengan nilai tertentu juga akan mengambil mata kuliah pilihan tertentu. Adapun beberapa aturan asosiasi didapat yang dari pengujian yang nantinya akan dijadikan acuan untuk rekomendasi adalah sebagai berikut.

Tabel 0.5 Rekomendasi Mata Kuliah Pilihan

	MK yang Telah Diambil	Support (%)	Confidence (%)
CS4 113	CS4002-A	7.02	32.31
CS4 333	MA1223-A CS4102-A CS1911-A CS3243-A	10.40	77.78
CS4 343	CS4002-B	10.42	41.67
CS4 363	CS4002-A MA2713-C	6.69	71.43

CS4 383	CS4102-B CS3214-C	7.02	41.18
IS4 453	CS3214-C CS3623-B	6.69	31.25
CS4 573	SE3773-B	7.02	16.94
CS4 603	CS1911-B	9.90	40.82
CS4 633	SE3414-C SE2423-C CS2223-B	5.83	63.64
CS4 643	CS2223	10.00	20.34
CS4 323	SE3773-B CS2223-B CS2624-C	6.69	57.14
CS4 713	IS3013-B CS2624-C	18.60	63.16
CS4 723	CS4102-A CS3931-A CS1931-A	5.56	22.73

Tabel 4.5 di atas merupakan beberapa aturan rekomendasi yang mengacu pada aturan asosiasi pada tabel sebelumnya. Untuk memahami tabel di atas, dengan contoh kasus apabila seorang mahasiswa mengambil mata kuliah dengan kode CS4002 dengan nilai A, maka ia direkomendasikan untuk mengambil mata kuliah pilihan dengan kode CS4113.

Pada tabel tersebut, tidak semua mata kuliah pilihan dapat direkomendasikan. Hal ini bergantung pada kemunculan mata kuliah pilihan yang terdapat pada saat pembentukan aturan asosiasi. Untuk mata kuliah CS4313 dan CS4613 tidak dapat ditemukan aturan asosiasi yang memiliki unsur *item* rekomendasi dikarenakan kedua mata kuliah tersebut memiliki itemset yang sangat sedikit sehingga tidak terbentuk pada pola aturan asosiasi.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan pada Bab 4, maka didapatkan beberapa kesimpulan sebagai berikut.

1. Algoritma *Recursive Elimination* (Relim) dapat digunakan untuk menyelesaikan masalah perekomendasi mata kuliah pilihan informatika. Aturan asosiasi yang dihasilkan algoritma ini berupa mata kuliah wajib beserta nilainya dan mata kuliah pilihan yang telah diambil mahasiswa beserta nilainya.
2. Nilai *minimum support* pada kaitannya dengan waktu eksekusi, tidak memiliki

## Rekomendasi Pengambilan Mata Kuliah Pilihan Menggunakan Recursive Elimination Algorithm (Relim)

pengaruh signifikan terhadap durasi preprocessing, namun berpengaruh terhadap waktu terbentuknya aturan asosiasi. Semakin besar nilai *minimum support*, maka waktu yang dibutuhkan untuk menghasilkan aturan asosiasi semakin kecil karena *frequent itemset* yang semakin berkurang.

3. Jumlah *frequent itemset* yang dihasilkan berhubungan erat dengan aturan asosiasi yang terbentuk. Semakin sedikit *frequent itemset*-nya, maka semakin sedikit pula jumlah aturan asosiasi karena berkurangnya jumlah itemset yang dapat disusun ke dalam aturan asosiasi.
4. Nilai *minimum support* paling optimal yaitu 20, karena merupakan nilai yang dapat menghasilkan persentase akurasi terbesar. Sedangkan untuk persentase data *training*, tidak terlihat salah satu nilai yang cukup menonjol untuk dijadikan nilai optimal.

### 5.2 Saran

Adapun saran yang diperlukan untuk perbaikan jurnal ini adalah sebagai berikut.

1. Sistem dapat diakses secara *online* untuk memudahkan penggunaanya, kapanpun dan dimanapun.
2. Proses rekomendasi tidak hanya untuk jurusan Informatika saja, tapi juga mencakup jurusan lain di Telkom University.
3. Menggunakan data mahasiswa yang akurat, yaitu mahasiswa hanya mengambil mata kuliah pilihan dalam satu kelompok keahlian.

### Daftar Pustaka

- [1] Jiawei Han and Micheline Kamber. 2006. *Data Mining Concepts and Techniques*, Second Edition
- [2] Farrah Hanna Al-Zawaidah, Yosef Hasan Jbara, and Marwan Al-Abed Abu-Zanona. 2011. *An Improved Algorithm for Mining Association Rules in Large Databases*. World of Computer Science and Information Technology Journal
- [3] Borgelt, Christian. *Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination*. University of Magdeburg, Germany

- [4] Borgelt, Christian. *Simple Algorithms for Frequent Item Set Mining*. European Center for Soft Computing, Spain
- [5] Azhari and Anshori. 2009. *Pendekatan Aturan Asosiasi Untuk Analisis Pergerakan Saham*. Intelligent System Research Group, Computer Science, FMIPA UGM
- [6] Dubey, Pallavi. 2012. *Association Rule Mining on Distributed Data*
- [7] Goldie Gunadi and Dana Indra Sensuse. Maret 2012. *Penerapan Metode Data Mining Market Basket Analysis terhadap Data Penjualan Produk Buku dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-Growth) : Studi Kasus Percetakan PT. Gramedia*. Jurnal TELEMATIKA MKOM Vol.4 No.1
- [8] Kusriani dan Emha Taufiz Luthfi. 2009. *Algoritma Data Mining*. Yogyakarta : CV ANDI OFFSET (Penerbit ANDI)
- [9] Widodo, 21-23 Mei 2008, *Prediksi Mata Kuliah Pilihan dengan Aturan Asosiasi*, Jurusan Teknik Elektro – Fakultas Teknik Universitas Negeri Jakarta, Konferensi dan Temu Nasional Teknologi Informasi dan Komunikasi untuk Indonesia, Jakarta
- [10] Raden Selamet. 2008. *Association Rule*. Media Informatika volume 7 no. 1
- [11] R. Agrawal and R.Srikant.Sept,1994. *Fast Algorithms for Mining Association rules*. In Proc. Of the 20th Int'l Conf. On Very Large Databases, Santiago,Chile
- [12]

[http://www.itelkom.ac.id/plug-ins/file/file\\_download/INSTITUSI/BP%202011.pdf](http://www.itelkom.ac.id/plug-ins/file/file_download/INSTITUSI/BP%202011.pdf) (Date : 30/11/12)