

# **Aplikasi Data Mining untuk Mengukur Tingkat Kelulusan Mahasiswa dengan Metode *Apriori* dan *k-Mean Clustering***

## **(Studi Kasus: Jurusan Teknik Informatika Universitas Trunojoyo Madura)**

**Mohammad Syarief**

Prodi D3 Manajemen Informatika – Fakultas Teknik, Universitas Trunojoyo Madura

Jl. Raya Telang, PO BOX 2 Kamal, Bangkalan

Email: ayibnya@gmail.com

### **Abstrak**

Perguruan tinggi memerlukan proses evaluasi dan penilaian secara komprehensif terhadap mutu pendidikan berdasarkan standar terakreditasi, sehingga perguruan tinggi harus mengetahui kondisi mahasiswa berdasarkan tingkat kelulusannya. Hal ini akan dijadikan bahan evaluasi untuk melakukan perbaikan dan mempertahankan mutu pendidikan. Penelitian ini akan menentukan informasi tingkat kelulusan mahasiswa dengan teknik data mining menggunakan Algoritma Apriori dengan menghitung nilai *support* dan *confidence* hubungan antara tingkat kelulusan dengan data induk mahasiswa dan mengelompokkan mahasiswa berdasarkan kategori tingkat kelulusannya menggunakan algoritma *k-Mean Clustering*. Dengan menggunakan nilai minimum transaksi tertentu atau *threshold* 4 diperoleh nilai yang terbaik dengan rata-rata kelulusan 11 semester masuk jalur PMDK1 dan berasal dari wilayah Bangkalan. Data hasil keluaran menggunakan algoritma apriori selanjutnya dikelompokkan dengan algoritma *k-Mean Clustering*. Pengelompokan tersebut berdasarkan segmentasi data nilai, penelitian ini menghasilkan bahwa mahasiswa Jurusan Informatika Universitas Trunojoyo Madura rata-rata lulus dengan IPK 2.75–3.5 atau predikat sangat memuaskan dan rata-rata menempuh jumlah sks 144.

**Kata kunci:** *k-mean clustering*, *algoritma apriori*, *support*, *confidence*, kelulusan mahasiswa.

### **Abstract**

*Colleges require process of comprehensive for evaluation the quality of education. They are based on standards accredited, so that college students should know condition based on graduation rates. This evaluation will be used as materials to repair quality of education. This study will be determined with graduation rate of student information. Data mining techniques has been using a Apriori Algorithm to calculate the value of support and confidence the relationship between graduation rates with parent data. Value of specific transaction or a minimum value threshold of 4 of the best values obtained with an average of 11 semesters of graduation and enter the path PMDK1 Bangkalan. Output data using a priori algorithm then grouped by k-Means Clustering algorithm. The grouping is based on the segmentation of data values, this study suggest that the Department of Informatics University student Trunojoyo average graduate with a GPA of 2.75-3.5 or honors and the average number of credits to take 144.*

**Key word:** *k-mean clustering*, *apriori algorithm*, *support*, *confidence*, *graduation*.

## **Pendahuluan**

Perguruan tinggi memerlukan proses evaluasi dan penilaian secara komprehensif terhadap mutu pendidikan berdasarkan standar terakreditasi yang telah ditetapkan oleh BAN PT. Sehingga perguruan tinggi harus terus menerus melakukan perbaikan dan mempertahankan mutu pendidikan. Di dalam peraturan akademik Universitas Trunojoyo Madura bidang pendidikan tahun 2010 pada BAB I pasal 1 ayat 2 disebutkan bahwa “Program Sarjana (S1) reguler adalah program pendidikan akademik setelah pendidikan menengah, yang memiliki beban studi sekurang-kurangnya 144 (seratus empat puluh empat)

sks (satu kredit semester) dan sebanyak-banyaknya 160 (seratus enam puluh) sks yang dijadwalkan untuk 8 (delapan) semester dan dapat ditempuh dalam waktu kurang dari 8 (delapan) semester dan paling lama 14 (empat belas) semester”. Berdasarkan buku wisuda angkatan ke-1,2...dst, ± 50 peserta wisuda Program Sarjana (S1) reguler di Fakultas Teknik Jurusan Informatika menempuh masa studi lebih dari delapan semester. Hal ini menunjukkan bahwa masih banyak mahasiswa Program Sarjana (S1) reguler di Fakultas Teknik yang menempuh lama studi lebih dari delapan semester dari yang dijadwalkan delapan semester. Oleh karena itu, dengan memanfaatkan data induk mahasiswa dan data kelulusan mahasiswa, dapat

diketahui informasi tingkat kelulusan mahasiswa melalui teknik *data mining*.

Penelitian ini akan menentukan informasi tingkat kelulusan mahasiswa dengan teknik *data mining*. Informasi yang ditampilkan berupa nilai *support* dan *confidence* hubungan antara tingkat kelulusan dengan data induk mahasiswa menggunakan Algoritma *Apriori* dan mengelompokkan mahasiswa berdasarkan kategori tingkat kelulusannya menggunakan algoritma *k-mean clustering*. Penelitian ini tidak membahas pada sistem pendukung keputusan maupun sistem informasi akademik.

Dalam membangun *data mining* membutuhkan suatu *data warehouse*, oleh karena itu dalam pembahasan penelitian ini dibahas mengenai pembangunan *data warehouse* sederhana yang dibangun untuk memenuhi kebutuhan dari proses *data mining*. *Data warehouse* yang dibangun bukan merupakan *data warehouse* yang menyimpan seluruh data transaksional, hanya merupakan *data warehouse* yang menunjang pembangunan *data mining*, sehingga data dan formatnya pun disesuaikan dengan kebutuhan *data mining*. Penelitian ini dibatasi pada studi kasus prodi Teknik Informatika, untuk menghasilkan model mahasiswa yang nantinya dapat diimplementasinya ke seluruh prodi di Universitas Trunojoyo Madura.

## Metodologi Penelitian

### Association Rules

Ide dasar dari algoritma ini adalah dengan mengembangkan *frequent itemset*. Dengan menggunakan satu item dan secara rekursif mengembangkan *frequent itemset* dengan dua item, tiga item dan seterusnya hingga *frequent itemset* dengan semua ukuran. Untuk mengembangkan *frequent set* dengan dua item, dapat menggunakan *frequent set item*. Alasannya adalah bila satu *itemset* tidak melebihi *support minimum*, maka sembarang ukuran *itemset* yang lebih besar tidak akan melebihi *support minimum* tersebut. Secara umum, mengembangkan *set* dengan *fc-item* menggunakan *frequent set* dengan *k-1 item* yang dikembangkan dalam langkah sebelumnya. Setiap langkah memerlukan sekali pemeriksaan ke seluruh isi *database*. Dalam asosiasi terdapat istilah *antecedent* dan *consequent*, *antecedent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka”. Dalam analisis ini, *antecedent* dan *consequent* adalah sekelompok item yang tidak punya hubungan secara bersama [1].

Dari jumlah besar aturan yang mungkin dikembangkan, perlu memiliki aturan-aturan yang cukup kuat tingkat ketergantungan antar *item* dalam *antecedent* dan *consequent*. Untuk mengukur kekuatan aturan asosiasi ini, digunakan ukuran *support* dan *confidence*. *Support* adalah rasio antara jumlah transaksi yang memuat *antecedent* dan *consequent* dengan jumlah transaksi. *Confidence* adalah rasio antara jumlah transaksi yang meliputi semua item dalam *antecedent* dan *consequent* dengan jumlah transaksi yang meliputi semua item dalam *antecedent*.

$$S = \frac{\sum(Ta+Tc)}{\sum(T)} \quad (1)$$

$$C = \frac{\sum(Ta+Tc)}{\sum(Ta)} \quad (2)$$

keterangan:

S = *Support*

$\sum(Ta+Tc)$  = Jumlah transaksi yang mengandung *antecedent* dan *consequent*

$\sum(T)$  = Jumlah transaksi

Untuk selanjutnya iterasi iterasi ke-*k* dapat dibagi lagi menjadi beberapa bagian:

1. Pembentukan kandidat *itemset*  
Kandidat *k-item set* dibentuk dari kombinasi (*k-1*)-*itemset* yang didapat dari iterasi sebelumnya. Satu ciri dari Algoritma *Apriori* adalah adanya pemangkasan kandidat *k-itemset* yang *subset*-nya yang berisi *k-1 item* tidak termasuk dalam pola frekuensi tinggi dengan panjang *k-1*.
2. Penghitungan support dari tiap kandidat *k-itemset*  
*Support* dari tiap kandidat *k-itemset* didapat dengan *scan database* untuk menghitung jumlah transaksi yang memuat semua *item* di dalam kandidat *k-itemset* tersebut. Ini adalah juga ciri dari Algoritma *Apriori* yaitu diperlukan penghitungan dengan *scan* seluruh *database* sebanyak *k-itemset* terpanjang.
3. Tetapkan pola frekuensi tinggi  
Pola frekuensi tinggi yang memuat *k item* atau *k-itemset* ditetapkan dari kandidat *k-itemset* yang *support*-nya lebih besar dari *minimum support*. Kemudian dihitung *confidence* masing-masing kombinasi *item*. Iterasi berhenti ketika semua *item* telah dihitung sampai tidak ada kombinasi *item* lagi. Sedangkan algoritma pemrograman ditunjukkan pada Gambar 2.

```

Create L1 = set of supported itemsets of cardinality one
Set k to 2
while (Lk-1 != ∅) {
    Create Ck from Lk-1
    Prune all the itemsets in Ck that are not
    supported, to create Lk
    Increase k by 1
}
The set of all supported itemsets is L1 ∪ L2 ∪ ... ∪ Lk

```

**Gambar 1.** Algoritma Apriori

### Algoritma K-Mean Clustering

*K-Mean* termasuk dalam *partitioning clustering* yaitu setiap data harus masuk dalam *cluster* tertentu dan memungkinkan bagi setiap data yang termasuk dalam *cluster* tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke *cluster* yang lain. *K-Mean* memisahkan data ke *k* daerah bagian yang terpisah, di mana *k* adalah bilangan integer positif. Algoritma *K-Mean* sangat terkenal karena kemudahan dan kemampuannya untuk mengklasifikasi data besar dan *outlier* dengan sangat cepat.

Berikut adalah langkah-langkah algoritma *K-Mean*:

1. Penentuan pusat *cluster* awal  
 Dalam menentukan *n* buah pusat *cluster* awal dilakukan pembangkitan bilangan random yang merepresentasikan urutan data *input*. Pusat awal *cluster* didapatkan dari data sendiri bukan dengan menentukan titik baru, yaitu dengan merandom pusat awal dari data.
2. Perhitungan jarak dengan pusat *cluster*  
 Untuk mengukur jarak antara data dengan pusat *cluster* digunakan *Euclidian Distance*. Algoritma perhitungan jarak data dengan pusat *cluster*. Jarak hasil perhitungan akan dilakukan perbandingan dan dipilih jarak terdekat antara data dengan pusat *cluster*, jarak ini menunjukkan bahwa data tersebut berada dalam satu kelompok dengan pusat *cluster* terdekat.

Untuk mendapatkan pusat *cluster* baru bisa dihitung dari rata-rata nilai anggota *cluster* dan pusat *cluster*. Pusat *cluster* yang baru digunakan untuk melakukan iterasi selanjutnya, jika hasil yang didapatkan belum konvergen. Proses iterasi akan berhenti jika telah memenuhi maksimum iterasi yang dimasukkan oleh *User* atau hasil yang dicapai sudah konvergen (pusat *cluster* baru sama dengan pusat *cluster* lama).

Algoritma penentuan pusat *cluster* baru:

1. Cari jumlah anggota tiap *cluster*.
2. Hitung pusat baru dengan rumus

$$\text{Pusat } cluster \text{ baru} = \frac{x_1 + x_2 + \dots + x_i + x_n + x_p}{i + 1} \quad (3)$$

keterangan:

$x_1, x_2, x_3, \dots, x_n$  = anggota *cluster*

$x_p$  = pusat lama

Selanjutnya dilakukan membuat aturan inferensinya sebagai berikut:

1. Komposisi Aturan

Dalam menentukan inferensi *fuzzynya* secara umum dapat dituliskan sebagai berikut:

$$\mu_f(x_i) = \max(\mu_f(x_i), \mu_k(x_i)) \quad (4)$$

di mana:

$\mu_f(x_i)$  nilai keanggotaan *fuzzy* sampai aturan ke  $-i$

$\mu_k(x_i)$  : nilai keanggotaan konsekuen *fuzzy* aturan ke  $-i$

2. Penegasan (*Defuzzifikasi*)

$$Z = \frac{\sum_{K=1}^M \mu_f X_i}{\sum_{K=1}^M \mu_f} \quad (5)$$

Keterangan:

$Z$  = nilai rata-rata terbobot

$\mu_f$  = nilai keanggotaan *fuzzy* sebagai hasil dari komposisi aturan

$X_i$  = nilai domain

## Hasil dan Pembahasan

### Analisa Perhitungan Menggunakan Algoritma Apriori

Algoritma *Apriori* digunakan untuk mengetahui hubungan antara beberapa *itemset*, seperti diperlihatkan pada perhitungan pada Tabel 1.

Ditetapkan *threshold* = 4, maka kandidat yang nilainya kurang dari 4 akan dihapus. Sehingga, didapat hasil seperti pada Tabel 1. Dari sejumlah *itemset* di-*scanning* data berdasarkan beberapa atribut berdasarkan lama kelulusan, jalur masuk dan daerah asal ditunjukkan pada Tabel 2 yang merupakan hasil kandidat dari kombinasi data tiga kriteria.

**Tabel 1.** Data Sampel Contoh Kedua

NIM	Kategori Kelulusan	Proses Masuk	Kota Asal Sekolah	IPK	SKS
M1A003001	L1	PMDK1	Surabaya	3.52	147
M1A003002	L2	SNMPTN	Pamekasan	2.85	144
M1A003003	L1	SNMPTN	Pamekasan	3.56	146
M1A003004	S1	SNMPTN	Surabaya	3.60	150
M1A003005	S3	PMDK2	Sidoarjo	2.6	153
M1A003006	S2	PMDK1	Bangkalan	2.98	144
M1A003007	S1	SNMPTN	Gresik	3.52	144
M1A003008	L2	SNMPTN	Pamekasan	3.01	146
M1A003009	S1	SNMPTN	Surabaya	3.50	150
M1A0030010	S1	PMDK1	Surabaya	3.54	150
M1A0030011	L2	SNMPTN	Lamongan	2.8	147
M1A0030012	L3	SNMPTN	Pamekasan	2.56	153
M1A0030013	S1	SNMPTN	Bangkalan	3.52	153
M1A0030014	S2	SPMP	Bangkalan	2.90	148
M1A0030015	S1	SPMP	Gresik	3.65	144
M1A0030016	L2	PMDK1	Pamekasan	3.10	144
M1A0030017	S1	PMDK1	Surabaya	3.70	146
M1A0030018	L2	PMDK1	Pamekasan	2.85	147
M1A0030019	L2	SNMPTN	Bangkalan	2.94	144
M1A0030020	L3	PMDK2	Sampang	2.70	148

**Tabel 2.** Hasil Kandidat Ketiga (C3) Berdasarkan Tiga Kriteria

Itemset	Count
L2, SNMPTN, Pamekasan	4
S1, SNMPTN, Pamekasan	4
S1, PMDK1, Surabaya	4

Berdasarkan hasil kandidat 3 (C3) Tabel 2 dilakukan perhitungan nilai *support* dan *confidence*.

Dari data-data tersebut dapat diambil hasil sebagai berikut:

$$\text{Support (L2, SNMPTN, Pamekasan)} = \text{Count (L2,SNMPTN, Pamekasan)}/\text{jumlah transaksi} = 4/20$$

$$\text{Support (S1, SNMPTN, Pamekasan)} = \text{Count(S1, SNMPTN, Pamekasan)}/\text{jumlah transaksi} = 4/20$$

$$\text{Support (S1, PMDK1, Surabaya)} = \text{Count(S1, PMDK1, Surabaya)}/\text{jumlah transaksi} = 4/20$$

$$\text{Confidence (L2, SNMPTN, Pamekasan)} = \text{Count (L2, SNMPTN, Pamekasan)}/\text{Count(L2)} = 4/6$$

$$\text{Confidence (S1, SNMPTN, Pamekasan)} = \text{Count (S1, SPMB, Pamekasan)}/\text{Count(S1)} = 4/7$$

$$\text{Confidence (S1, PMDK1, Surabaya)} = \text{Count (S1, PMDK1, Surabaya)}/\text{Count(S1)} = 4/7$$

Sedangkan hasil simulasi data penelitian yang telah dilakukan maka diperoleh hasil sebagaimana data dilihat pada tabel 3.

Berdasarkan Tabel 3 untuk mengukur kekuatan aturan asosiasi ini, digunakan ukuran *support* dan *confidence*. Nilai *support* dan *confidence* yang lebih tinggi mempunyai nilai kemunculan yang lebih dominan dibandingkan yang lain.

### Analisa Perhitungan Menggunakan Algoritma *k-Mean Clustering*

Penerapan algoritma *k-mean* dengan menggunakan data pada Tabel 4 sebagai contoh percobaan perhitungan manual.

Iterasi ke-1

#### 1. Penentuan pusat awal *cluster*

Tentukan pusat *cluster* secara acak, misalkan kita tentukan  $c_1 = (2.6, 144)$ ;  $c_2 = (3.1, 145)$ ; dan  $c_3 = (3.5, 146)$

#### 2. Perhitungan jarak pusat *cluster*

Untuk mengukur jarak antara data dengan pusat *cluster* digunakan *Euclidian Distance*, kemudian akan didapatkan matrik jarak sebagai berikut:

**Tabel 3.** Analisa Data Skenario Berdasarkan Kelulusan 2007–2010

Tahun	Keterangan	Support	Confidence
2007	L2, PMDK1, Surabaya	12/58	6/10
	S1, SNMPTN, Bangkalan	8/58	7/12
2008	L2, SNMPTN, Surabaya	15/60	6/12
	S1, PMDK1, Bangkalan	7/60	4/9
	L3, SNMPTN, Surabaya	6/60	8/14
2009	L2, SNMPTN, Bangkalan	10/60	8/10
	S1, PMDK1, Bangkalan	8/60	7/12
	L3, PMDK1, Surabaya	11/60	10/16
2010	L3, SNMPTN, Surabaya	12/48	6/14
	S1, PMDK1, Bangkalan	18/48	5/12
	L3, PMDK2, Pamekasan	8/48	6//15

Rumus *Euclidian Distance*:

$$d = |x-y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (6)$$

$x$  = pusat *cluster*  
 $Y$  = data

Sebagai contoh, perhitungan jarak dari data ke-1 terhadap pusat *cluster* adalah:

$$C1 = \sqrt{(3.52-2.6)^2 + (147-144)^2} = 3.16$$

$$C2 = \sqrt{(3.52-3.1)^2 + (147-145)^2} = 2.04$$

$$C1 = \sqrt{(3.52-3.5)^2 + (147-146)^2} = 1.00$$

Dan seterusnya dilanjutkan untuk data ke 2, 3, ... n

Kemudian akan didapatkan matrik jarak sebagai berikut:

Setiap kolom pada matrik menunjukkan nilai jarak data terhadap pusat *cluster*. Baris pertama pada matrik menunjukkan nilai jarak data terhadap titik pusat *cluster* pertama, baris kedua pada matrik menunjukkan nilai jarak data terhadap titik pusat *cluster* kedua dan seterusnya.

### 3. Pengelompokan data

Jarak hasil perhitungan akan dilakukan perbandingan dan dipilih jarak terdekat antara data dengan pusat *cluster*, jarak ini menunjukkan bahwa data tersebut berada dalam satu kelompok dengan pusat *cluster* terdekat. Berikut ini akan ditampilkan data matriks pengelompokan *group*, nilai 1 berarti data tersebut berada dalam *group*.

**Tabel 4.** Hasil Clusterisasi Mahasiswa pada Iterasi 1

Itemset	Kategori	IPK	SKS	d1	d2	d3	C1	C2	C3
M1A003001	L1	3.52	147	3.1685	2.0436	1.0002			x
M1A003002	L2	2.85	144	0.35	1.0307	2.1027	x		
M1A003003	S1	3.56	144	1.06	1.1007	2.0009	x		
M1A003004	S1	3.6	150	6.1	5.0249	4.0012			x
M1A003005	S3	2.6	153	9.0005	8.0156	7.05762			x
M1A003006	S2	2.98	144	0.48	1.0071	2.0664	x		
M1A003007	S1	3.52	144	1.02	1.0846	2.0001	x		
M1A003008	L2	3.01	146	2.06400	1.0040	0.49			x
M1A003009	S1	3.5	144	1	1.0770	2	x		
M1A0030010	S1	3.68	144	1.18	1.1560	2.0080		x	
M1A0030011	L2	2.8	147	3.01496	2.0223	1.2206			x
M1A0030012	L3	2.56	153	9.0002	8.0182	7.0628			x
M1A0030013	S1	3.7	144	1.2	1.1662	2.0099		x	
M1A0030014	S2	2.9	148	4.0199	3.0067	2.0881			x

Lanjutan Tabel 4.

Itemset	Kategori	IPK	SKS	d1	d2	d3	C1	C2	C3
M1A0030015	S1	3.65	144	1.15	1.1413	2.0056		x	
M1A0030016	L2	2.8	148	4.01125	3.0149	2.0881			x
M1A0030017	S1	3.7	146	2.3324	1.1662	0.2			x
M1A0030018	L2	2.85	147	3.0204	2.0155	1.1927			x
M1A0030019	L2	2.94	144	0.44	1.0127	2.0769	x		
M1A0030020	L3	2.7	148	4.0050	3.0265	2.1541			x

4. Penentuan pusat *cluster* baru

Setelah diketahui anggota tiap-tiap *cluster* kemudian pusat *cluster* baru dihitung berdasarkan data anggota tiap-tiap *cluster* sesuai dengan rumus pusat anggota *cluster*. Sehingga didapatkan perhitungan sebagai berikut: Karena  $C_1$  memiliki 10 anggota maka perhitungan *cluster* baru menjadi:

$$C_{11} = \frac{2.85 + 3.56 + 2.98 + 3.52 + 3.5 + 2.94}{6} = 3.225$$

$$C_{12} = \frac{144 + 144 + 144 + 144 + 144 + 144}{6} = 144$$

Karena  $C_2$  hanya mempunyai dua anggota maka *cluster* baru menjadi:

$$C_{21} = \frac{3.68 + 3.7 + 3.65}{3} = 3.675$$

$$C_{22} = \frac{144 + 144 + 144}{3} = 144$$

Karena  $C_3$  hanya mempunyai dua anggota maka *cluster* baru menjadi:

$$C_{31} = \frac{3.52 + 3.6 + 2.6 + 3.01 + 2.8 + 2.56 + 2.9 + 2.8 + 3.7 + 2.85 + 27}{11} = 3.0$$

$$C_{32} = \frac{147 + 150 + 153 + 146 + 147 + 153 + 148 + 148 + 146 + 147 + 148}{11} = 148.6$$

Iterasi Ke-2

1. Ulangi langkah ke 2 (kedua) hingga posisi data tidak mengalami perubahan.
2. Langkah selanjutnya sama dengan langkah pada nomor 3 jarak hasil perhitungan akan dilakukan

perbandingan dan dipilih jarak terdekat antara data dengan pusat *cluster*, jarak ini menunjukkan bahwa data tersebut berada dalam satu kelompok dengan pusat *cluster* terdekat.

3. Berikut ini akan ditampilkan data matriks pengelompokan group, nilai 1 berarti data tersebut berada dalam *group*. Karena  $G2 = G1$  memiliki anggota yang sama maka tidak perlu dilakukan iterasi/perulangan lagi. Hasil *clustering* telah mencapai stabil dan konvergen.

Berdasarkan uji coba data yang digunakan dalam percobaan ini adalah data mahasiswa Prodi Teknik Informatika yang masuk mulai dari angkatan 2003, 2004, 2005 dan 2006. Sehingga mahasiswa tersebut diperkirakan lulus sekitar tahun 2007, 2008, 2009 dan 2010.

**Analisa data berdasarkan Jumlah Iterasi**

Pada analisa data dilakukan iterasi sampai ke- $n$  sampai data memiliki anggota yang sama dan mencapai kestabilan atau konvergen, sehingga tidak perlu dilakukan iterasi/perulangan lagi. Berdasarkan hasil uji coba yang dilakukan seperti ditunjukkan pada Tabel 5.

**Tabel 5.** Analisa Data Mahasiswa Perangkatan berdasarkan Jumlah Iterasi

Skenario per Angkatan	Maks Iterasi
2003	25
2004	23
2005	24
2006	28

Sedangkan pada uji coba berdasarkan iterasi diperoleh data sesuai dengan Tabel 5.20.

Rata-rata iterasi dilakukan sampai memperoleh *cluster* yang sama dicapai 20–30 iterasi.

**Tabel 6.** Data Hasil Iterasi Menggunakan Algoritma *k-Mean Clustering*

Tahun lulus	Total Mahasiswa	Jumlah Cluster	Pusat Cluster	Rata-rata Status Kelulusan
2007	58	C1=38 C2=19 C3=11	$C_{11} = 3,12$ $C_{12} = 144$ $C_{21} = 2,94$ $C_{22} = 144$ $C_{31} = 3,59$ $C_{32} = 148$	<ul style="list-style-type: none"> <li>• <math>IP &lt; 2.75 =</math> Memuaskan</li> <li>• <math>IP 2.75 - 3.5 =</math> Sangat Memuaskan</li> <li>• <math>IP &gt; 3.5 =</math> dengan Pujian</li> </ul>
2008	60	C1=30 C2=10 C3=20	$C_{11} = 3,05$ $C_{12} = 144$ $C_{21} = 2,76$ $C_{22} = 146$ $C_{31} = 3,42$ $C_{32} = 150$	<ul style="list-style-type: none"> <li>• <math>IP &lt; 2.75 =</math> Memuaskan</li> <li>• <math>IP 2.75 - 3.5 =</math> Sangat Memuaskan</li> <li>• <math>IP &gt; 3.5 =</math> dengan Pujian</li> </ul>
2009	60	C1=31 C2=14 C3=15	$C_{11} = 2,9$ $C_{12} = 146$ $C_{21} = 3,01$ $C_{22} = 144$ $C_{31} = 3,32$ $C_{32} = 144$	<ul style="list-style-type: none"> <li>• <math>IP &lt; 2.75 =</math> Memuaskan</li> <li>• <math>IP 2.75 - 3.5 =</math> Sangat Memuaskan</li> <li>• <math>IP &gt; 3.5 =</math> dengan Pujian</li> </ul>
2010	48	C1=20 C2=10 C3=18	$C_{11} = 3,005$ $C_{12} = 144$ $C_{21} = 2,95$ $C_{22} = 146$ $C_{31} = 3,42$ $C_{32} = 148$	<ul style="list-style-type: none"> <li>• <math>IP &lt; 2.75 =</math> Memuaskan</li> <li>• <math>IP 2.75 - 3.5 =</math> Sangat Memuaskan</li> <li>• <math>IP &gt; 3.5 =</math> dengan Pujian</li> </ul>

Berdasarkan data pada Tabel 5 dan Tabel 6 diperoleh hasil bahwa mahasiswa Prodi Teknik Informatika Universitas Trunojoyo Madura rata-rata lulus dengan IPK 2.75–3.5 atau predikat sangat memuaskan dan rata-rata menempuh 144 sks. Sedangkan rata-rata total mahasiswa yang lulus per yudisium sejumlah 40-60 orang.

### Simpulan

Dari hasil penelitian, yang telah dilakukan dapat disimpulkan hal-hal sebagai berikut:

1. Aplikasi *data mining* dapat digunakan untuk mengetahui tingkat kelulusan mahasiswa yang berupa nilai *support* dan *confidence* yaitu hubungan antara tingkat kelulusan dengan data induk mahasiswa.
2. Semakin tinggi nilai *confidence* dan *support* maka semakin kuat nilai hubungan antar atribut. Data

induk mahasiswa yang diproses *mining* meliputi data proses masuk, data asal sekolah, data kota mahasiswa, dan data program studi.

3. Pada penggunaan algoritma *k-Mean* untuk mengelompokkan mahasiswa berdasarkan status kelulusannya.
4. Hasil pengelompokan tersebut adalah mahasiswa Prodi Teknik Informatika Universitas Trunojoyo Madura rata-rata lulus dengan IPK 2.75–3.5 atau predikat sangat memuaskan dan rata-rata menempuh 144 sks.

### Daftar Pustaka

- [1] Santosa, B., 2007, “Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis”, Graha Ilmu, Yogyakarta.