

**PEMBUATAN *PROTOTYPE***  
**PERANGKAT LUNAK *DATA MINING* BERBASIS WEB**  
**UNTUK PENGGALIAN KAIDAH ASOSIASI**  
**(*MINING ASSOCIATION RULES*)**  
**BERDASARKAN ALGORITMA APRIORI MENGGUNAKAN PHP**

**Tomi Listiawan, S.Si., M.Pd.**

Program Studi Pendidikan Matematika STKIP PGRI Tulungagung  
Jl. Mayor Sujadi Timur No.7 Tulungagung  
tomi.listiawan@gmail.com

**ABSTRAK**

*Ketersediaan data dari suatu database transaksional, telah mendorong pengembangan teknik-teknik yang secara otomatis dapat menemukan asosiasi produk atau item-item yang tersimpan dalam database. Teknik penemuan aturan asosiasi antar produk yang tersimpan dalam database inilah yang dikenal dengan teknik mining association rules.*

*Banyak teori dan algoritma yang dikembangkan untuk melakukan teknik mining association rules. Salah satu algoritma yang dikembangkan adalah Apriori. Metode atau algoritma apriori ini mempunyai tujuan utama untuk mencari maksimal frequent itemset. Frequent itemset inilah yang selanjutnya di-generate menjadi aturan-aturan asosiatif, yang sebelumnya sama sekali tidak muncul dalam database, menjadi suatu informasi berharga untuk bahan pertimbangan proses pengambilan keputusan.*

*Algoritma apriori yang merupakan interpretasi teknik mining association rules, akan di implementasikan dalam suatu perangkat lunak berbasis web. Perangkat lunak ini dibuat dengan menggunakan seperangkat komputer berprosesor Intel Celeron 1,86GHz, memori 512 MHz, dan dengan kapasitas hardisk 80 GB. Software (perangkat lunak) yang digunakan meliputi software desain web Macromedia Dreamweaver, pengolah database MySQL, webserver Apache, browser Mozilla Firefox, bahasa pemrograman PHP, serta editor PHPEdit-06.*

*Pada uji coba perangkat lunak yang diterapkan pada beberapa data yang berbeda, diperoleh suatu kesimpulan bahwa waktu yang diperlukan untuk proses penggalian kaidah asosiasi sangat tergantung pada jumlah kemunculan setiap item pada transaksi, jumlah transaksi, minimum support, dan minimum confidence. Sedangkan semakin kecil nilai minimum support dan minimum confidence yang dimasukkan, maka aturan asosiasi yang dihasilkan akan semakin banyak, demikian pula sebaliknya.*

**Kata Kunci :** *Data mining, mining association rules, apriori*

## **1. Pendahuluan**

### **A. Latar Belakang**

Seiring dengan meningkatnya jumlah konsumen atau pengunjung di suatu pasar swalayan, semakin meningkat pula jumlah transaksi yang ada di pasar swalayan tersebut, sehingga transaksi yang tersimpan di *database*

pun juga akan semakin besar, sehingga dalam jangka waktu yang cukup lama akan menimbulkan suatu *gunungan* data yang sangat besar. Untuk mendapatkan informasi-informasi berharga yang “tersembunyi” dalam “tumpukan data transaksi” yang ada dalam *database*

diperlukan suatu cara yang disebut *data mining*.

Ketersediaan *database* mengenai catatan transaksi pembelian para pelanggan suatu pasar swalayan, telah mendorong pengembangan teknik-teknik yang secara otomatis dapat menemukan asosiasi produk atau item-item yang tersimpan dalam *database*, seperti data mengenai transaksi yang terkumpul dari *bar-code scanner* di pasar swalayan. *Database* seperti ini mengandung *record* yang sangat besar. *Record* tersebut mendaftar item yang dibeli oleh seorang pelanggan pada suatu transaksi pembelian. Dari sini pengelola pasar swalayan dapat mengetahui apakah suatu kelompok item selalu dibeli secara bersama-sama, sehingga pengelola pasar swalayan bisa menggunakan informasi tersebut untuk membuat *lay-out* pasar swalayannya sehingga penyusunan item-item tersebut bisa optimal satu sama lainnya, untuk keperluan promosi, segmentasi pembeli, pembuatan katalog produk, atau melihat pola belanja.

### B. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, rumusan masalah dalam penelitian ini adalah bagaimana merancang dan membuat suatu perangkat lunak *data mining* berbasis teknologi web untuk penggalian kaidah asosiasi (*mining association rules*) berdasarkan algoritma apriori menggunakan bahasa pemrograman PHP.

### C. Batasan Masalah

Batasan masalah pada penelitian ini adalah :

- 1) Merancang dan membuat suatu perangkat lunak *data mining* berbasis teknologi web menggunakan bahasa pemrograman PHP.
- 2) Teknik *data mining* yang digunakan adalah teknik untuk menggali kaidah asosiasi (*mining association rules*).
- 3) Algoritma yang digunakan dalam penggalian kaidah asosiasi pada perangkat lunak ini adalah algoritma Apriori.
- 4) Tabel transaksi yang menjadi input adalah tabel yang telah diubah dalam bentuk *denormal*.

### D. Definisi Istilah

- 1) *Data mining* adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santosa, 2007:10).
- 2) Asosiasi adalah aturan untuk memeriksa semua kemungkinan hubungan if-then antar item dan memilih hanya yang paling mungkin (*most likely*) sebagai indikator dari hubungan ketergantungan antar item (Santosa, 2007:226).
- 3) *Mining Association Rules* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi item (Pramudiono, 2006).

## 2. Kajian Pustaka

### A. Data Mining

Istilah *data mining* sudah berkembang jauh dalam mengadaptasi setiap bentuk analisa data. Pada dasarnya *data mining* berhubungan dengan analisa data dan penggunaan teknik-teknik perangkat lunak untuk mencari pola dan keteraturan dalam himpunan data yang sifatnya “tersembunyi”.

Dengan diperolehnya informasi-informasi yang berguna dari data yang ada, hubungan antara item dalam transaksi, maupun informasi-informasi yang potensial, selanjutnya dapat diekstrak dan dianalisa serta diteliti lebih lanjut dari berbagai sudut pandang. Informasi yang ditemukan ini selanjutnya dapat diaplikasikan untuk aplikasi manajemen, melakukan query processing, pengambilan keputusan dan lain sebagainya. Dengan semakin berkembangnya kebutuhan akan informasi-informasi, semakin banyak pula bidang-bidang yang menerapkan konsep *data mining*.

Beberapa pengertian *data mining* yang dikemukakan oleh para ahli di bidang ini adalah sebagai berikut.

1. Secara sederhana *Data mining* dapat didefinisikan sebagai ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di *database* yang besar sehingga menjadi informasi yang sangat berharga (Suchahyo,2003).
2. *Data mining* merupakan proses penemuan yang efisien suatu pola terbaik yang dapat menghasilkan sesuatu yang bernilai dari suatu koleksi data yang sangat besar (Anonim).
3. *Data mining* adalah suatu pola yang menguntungkan dalam melakukan *search* pada suatu *database* yang terdapat pada suatu model. Proses ini dilakukan berulang-ulang (*iterasi*) hingga didapat satu set pola yang memuaskan yang dapat berfungsi sesuai yang diharapkan (Chung,2004).

4. *Data mining* adalah suatu *class* dari suatu aplikasi *database* yang mencari pola-pola yang tersembunyi di dalam group data yang dapat digunakan untuk memprediksi perilaku yang akan datang (Anonim).

Berdasarkan beberapa pengertian di atas dapat ditarik kesimpulan bahwa *data mining* adalah suatu teknik di dalam menggali informasi berharga yang terpendam atau tersembunyi pada suatu koleksi data (*database*) yang sangat besar sehingga ditemukan suatu pola yang menarik yang sebelumnya tidak diketahui.

#### **B. *Mining Association Rules (Market Basket Analysis)***

Analisis asosiasi atau *mining association rules* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi item. Contoh dari aturan asosiasi dari analisa pembelian di suatu pasar swalayan adalah bisa 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 untuk kombinasi barang tertentu. Karena analisis asosiasi menjadi terkenal karena aplikasinya untuk menganalisa isi keranjang belanja di pasar swalayan, analisis asosiasi juga sering disebut dengan istilah *market basket analysis*.

Ide dari aturan asosiasi adalah untuk memeriksa semua kemungkinan hubungan if - then (jika – maka) antar item dan memilih hanya yang paling mungkin (*most likely*), sebagai indikator dari hubungan ketergantungan antar

item. Biasanya digunakan istilah *antecedent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka”

Penting tidaknya suatu aturan asosiasi dapat diketahui dengan dua parameter, *support* (nilai penunjang/dukungan) dan *confidence* (nilai kepastian). *Support* dari suatu aturan adalah jumlah transaksi yang mengandung item, baik dalam *antecedent* maupun *consequent*. Disebut *support* karena ia mengukur seberapa tingkat dukungan data terhadap validitas aturan yang dikembangkan. *Support* biasanya dinyatakan dalam ukuran persen (%).

Dari jumlah besar aturan yang mungkin dikembangkan, kita perlu memilih aturan-aturan yang cukup kuat tingkat ketergantungan antar item dalam *antecedent* dan *consequent*. Selain *support* ada ukuran lain yang mengukur tingkat kepastian aturan “if-then” atau “jika-maka”. Ukuran tersebut adalah *confidence*. *Confidence* adalah rasio antara jumlah transaksi yang meliputi semua item dalam *antecedent* dan *consequent* dengan jumlah transaksi yang meliputi semua item dalam *antecedent*, atau

$$confidence = \frac{\text{jumlah transaksi dengan item dalam antecedent dan consequent}}{\text{jumlah transaksi dengan item dalam antecedent}}$$

Beberapa definisi formal terkait dengan bentuk dasar aturan asosiasi ini diantaranya,

Misalkan  $I = \{I_1, I_2, \dots, I_m\}$  adalah himpunan  $m$  item – item, dan misalkan  $D = \{T_1, T_2, \dots, T_n\}$  merupakan suatu *database* yang merupakan koleksi dari  $n$  transaksi dimana untuk setiap

$T \in D, T \subseteq I$ . Suatu himpunan item  $X \subseteq I$  disebut itemset. Suatu transaksi  $T$  memuat itemset  $X$  jika  $X \subseteq T$ . Setiap itemset  $X$  diasosiasikan dengan suatu himpunan transaksi  $TX = \{T \in D | T \supseteq X\}$  yang merupakan himpunan transaksi yang memuat itemset  $X$ . Misalkan  $X, Y \subseteq I$  adalah itemset tidak kosong dengan  $X \cap Y = \phi$ , maka aturan asosiasi (*association rule*) adalah suatu implikasi  $X \rightarrow Y$ , dan secara formal

$$supp(X) = \frac{|\{T \in D | X \subseteq T\}|}{|D|},$$

$$conf(X \rightarrow Y) = \frac{|\{T \in D | X \cup Y \subseteq T\}|}{|\{T \in D | X \subseteq T\}|}$$

$$, \text{ dan } supp(X \rightarrow Y) = supp(X \cup Y)$$

Aturan asosiasi biasanya dinyatakan dalam bentuk :

$$\{i_1, i_2, \dots, i_n\} \rightarrow \{k_1, k_2, \dots, k_m\} \text{ (support} = x \%, \text{ confidence} = y \%), i_n, k_m \in I; n, m, x, y \in \square$$

Berikut adalah contoh dari bentuk penulisan aturan asosiasi :  $\{\text{buku, pensil}\} \rightarrow \{\text{penghapus}\}$  (*support* = 20%, *confidence* = 75%) Yang artinya : "75% dari transaksi di *database* yang memuat item buku dan pensil juga memuat item penghapus. Sedangkan 20% dari seluruh transaksi yang ada di *database* memuat ketiga item itu."

Dapat juga diartikan : "Seorang konsumen yang membeli buku dan pensil punya kemungkinan 75% untuk juga membeli penghapus. Aturan ini cukup signifikan karena mewakili 20% dari catatan transaksi selama ini."

### C. Algoritma Apriori

Algoritma Apriori adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu *database* yang memiliki frekuensi atau *support* di atas ambang batas tertentu yang disebut dengan istilah *minimum support*. Pola frekuensi tinggi ini digunakan untuk menyusun aturan asosiasi.

Algoritma apriori dibagi menjadi beberapa tahap yang disebut iterasi atau *pass*. Tiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama dimulai dari *pass* pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Di iterasi pertama ini, *support* dari setiap item dihitung dengan men-*scan database*. Setelah *support* dari setiap item didapat, item yang memiliki *support* di atas *minimum support* dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disingkat 1-itemset. Singkatan k-itemset berarti satu set yang terdiri dari k item.

Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung *support*-nya dengan men-*scan database*. *Support* disini artinya jumlah transaksi dalam *database* yang mengandung kedua item dalam kandidat 2-itemset. Setelah *support* dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat *minimum support* dapat ditetapkan sebagai 2-itemset yang juga

merupakan pola frekuensi tinggi dengan panjang 2.

Secara umum pada iterasi ke-k dilakukan proses sebagai berikut :

1. Pembentukan kandidat itemset  
Kandidat k-itemset dibentuk dari kombinasi (k-1)-itemset yang didapat dari iterasi sebelumnya (*join*). Satu ciri dari algoritma Apriori adalah adanya pemangkasan (*prune*) kandidat k-itemset yang subset-nya 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 men-*scan database* untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k-itemset tsb. Ini adalah juga ciri dari algoritma apriori dimana 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*.
4. Bila tidak didapat pola frekuensi tinggi baru maka seluruh proses dihentikan. Bila tidak, maka k ditambah satu dan kembali ke bagian 1 (satu).

Apabila dituliskan dalam bentuk pseudocode, algoritma apriori adalah sebagai berikut :

```

L1 = {large 1-itemset};
k = 2; // k represent the
pass number
While ( Lk-1 ≠ ∅ ) do

```

```

    { $C_k$  = new candidate of
    size  $k$  generated from  $L_{k-1}$ ;
    Forall transaction
     $t \in D$  do
    increment the count of all
    candidates in  $C_k$  that are
    contained in  $t$ ;
     $L_k$  = all candidates in  $C_k$ 
    with minimum support;
     $k = k + 1$ ; }
    answer =  $\cup_k L_k$ 

```

Sedangkan pseudocode dari pembentukan kandidat itemset bersama pemangkasannya diberikan sebagai berikut :

#### JOIN STEP

```

Insert into candidate  $k$ -
itemset
Select  $p.item_1, p.item_2, \dots,$ 
 $p.item_{k-1}, q.item_{k-1}$ 
From large  $(k-1)$ -itemset  $p,$ 
large  $(k-1)$ -itemset  $q$ 
Where  $p.item_1=q.item_1, \dots,$ 
 $p.item_{k-2}=q.item_{k-2},$ 
 $p.item_{k-1} < q.item_{k-1}$ 

```

#### PRUNE STEP

```

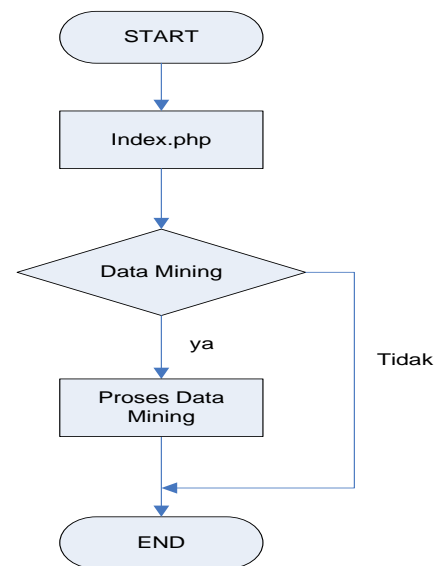
Forall itemset  $c \in$  candidate
 $k$ -itemset do
    Forall  $(k-1)$ -subsets  $s$ 
of  $c$  do
        If (  $s \notin$  large  $(k-1)$ -
itemset) then
            Delete  $c$  from
candidate  $k$ -itemset;

```

### 3. Perancangan Sistem

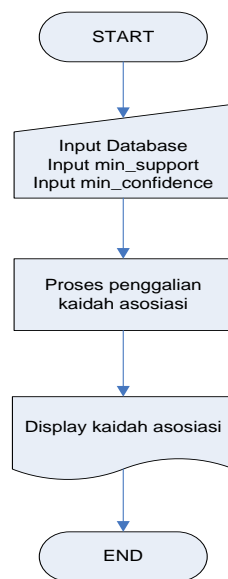
Perancangan proses dari perangkat lunak ini menggunakan pendekatan fungsional yang direpresentasikan menggunakan diagram alir. Diagram alir ini menunjukkan secara fisik alur proses dan data pada perangkat lunak *data mining* yang akan dibuat.

*Flowchart* program secara keseluruhan adalah sebagai berikut :



Gambar 3.1 *Flowchart* Program Secara Umum

Secara umum proses *data mining* menggunakan teknik *mining association rules* ini terdiri dari 3 bagian penting. Bagian tersebut terdiri dari proses input *minimum support* dan *minimum confidence*, proses penggalian kaidah asosiasi, dan proses menampilkan (*display*) output kaidah asosiasi. Proses ini ditunjukkan oleh *flowchart* berikut :

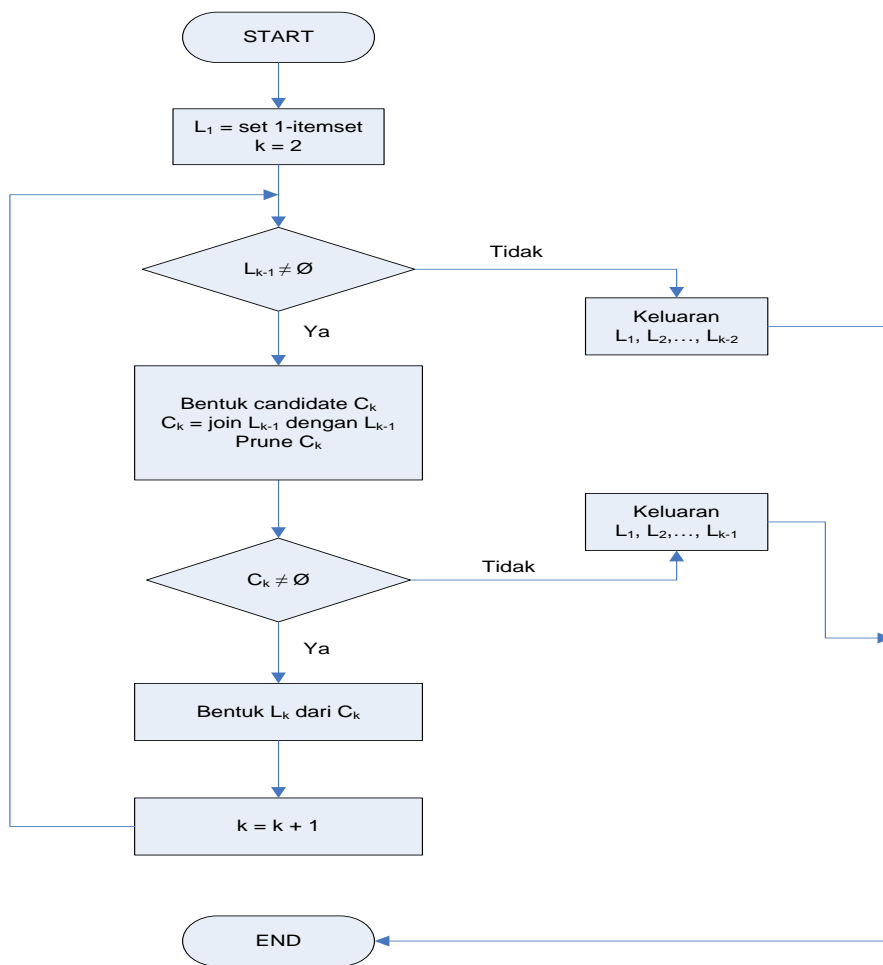


Gambar 3.2 *Flowchart* Teknik *Mining Association Rules*

Proses penggalian kaidah asosiasi memiliki dua tahap penting, yaitu pencarian *frequent* k-itemset yang direpresentasikan oleh algoritma apriori, dan penentuan semua aturan asosiasi. Data keluaran dari proses pencarian *frequent* k-itemset adalah terbentuknya semua *frequent* k-itemset. Data ini

nantinya akan digunakan sebagai data masukan untuk proses selanjutnya, yaitu proses pembentukan semua aturan asosiasi yang memenuhi *minimum confidence*.

Adapun *flowchart* proses pencarian *frequent* k-itemset ini adalah sebagai berikut :



Gambar 3.3 *Flowchart* Proses Penacarian *Frequent K-Itemset* / Algoritma Apriori

Seperti dijelaskan di atas bahwa hasil yang diperoleh dari proses pembentukan *frequent k-itemset* selanjutnya digunakan untuk membentuk aturan asosiasi berdasarkan nilai *minimum confidence* yang diberikan. Adapun algoritma yang digunakan untuk membentuk aturan asosiasi berdasarkan *frequent k-itemset* adalah algoritma Naïve.

*Pseudocode* algoritma Naïve adalah sebagai berikut :

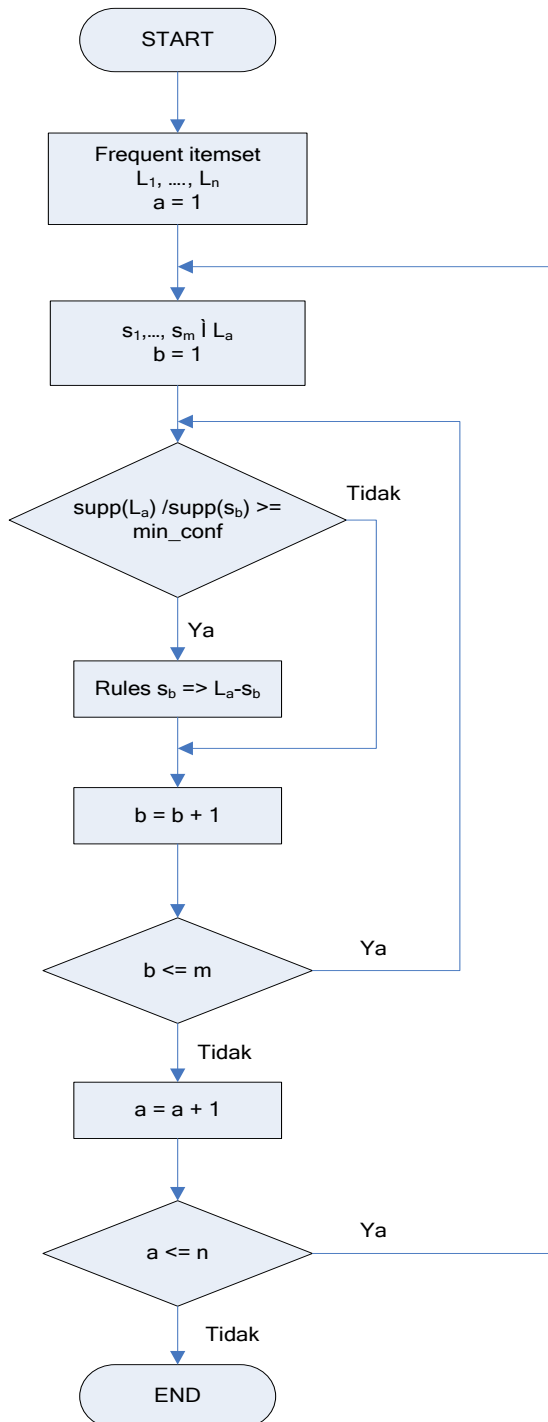
```

for each frequent itemset l
do for each nonempty proper
subset s of l do if
support(l)/support(s) >=
min_conf)output the rule s
→ l-s , with support
=support(l) and confidence
=support(l)/support(s)
  
```

Dari algoritma di atas terlihat bahwa proses pembentukan aturan asosiasi dilakukan dengan memeriksa semua *proper subset* dari *frequent k-itemset*. Kemudian menentukan nilai *confidence* dengan cara membagi nilai *support proper subset* dengan *support frequent k-itemset*. Semua aturan asosiasi yang memiliki nilai *confidence* di atas



nilai *minimum confidence* ditetapkan sebagai output dari proses penggalan kaidah asosiasi ini.



*Flowchart* dari algoritma Naïve adalah sebagai berikut :

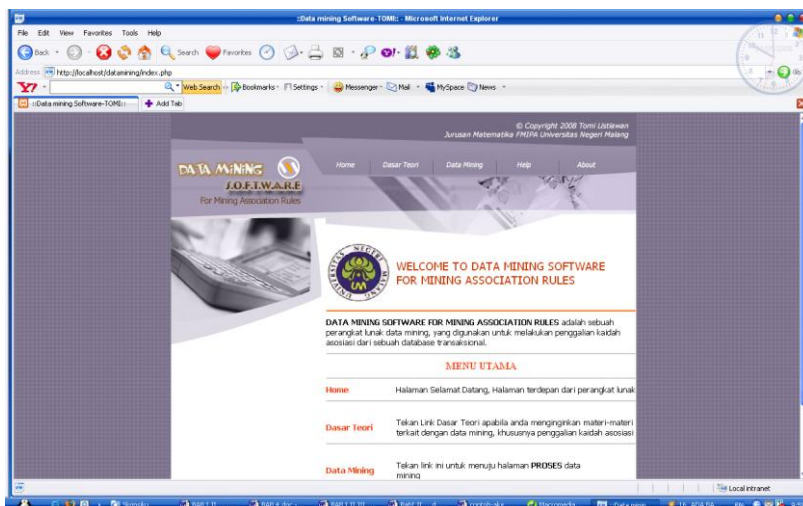
Gambar 3.4 *Flowchart* Algoritma Naïve

#### 4. Hasil Dan Pembahasan

Perangkat lunak *data mining* untuk penggalan kaidah asosiasi ini mempunyai dua bagian halaman utama, yaitu halaman pendahuluan dan halaman proses *data mining*.

##### A. Halaman Pendahuluan

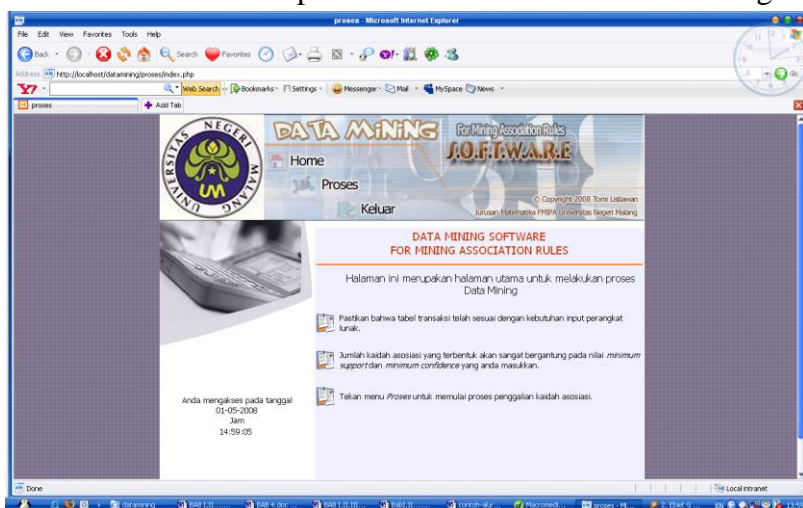
Halaman Pendahuluan ini merupakan halaman yang akan membantu pengguna untuk lebih mengenal tentang perangkat lunak yang akan digunakan. Untuk mengakses atau membuka perangkat lunak, digunakan browser dengan alamat URL [HTTP://localhost/datamining](http://localhost/datamining). Jika alamat URL tersebut dibuka, maka program akan menampilkan halaman utama yang bernama "index.php". Tampilan dari halaman utama tersebut adalah sebagai berikut



Gambar 4.1 Tampilan Halaman Utama Perangkat Lunak

**B. Halaman Proses *Data Mining***

Ketika menu *Data Mining* pada halaman pendahuluan ditekan, maka URL tersebut akan dieksekusi. Tampilan dari URL tersebut adalah sebagai berikut :



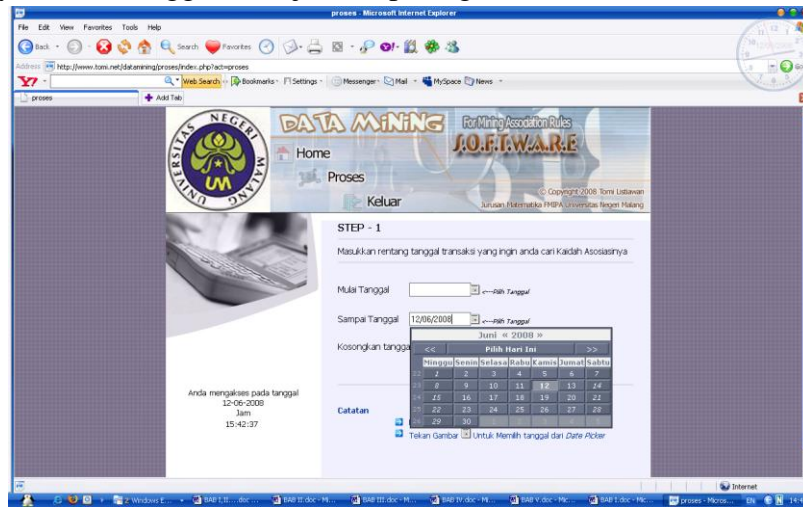
Gambar 4.2 Tampilan Halaman Proses *Data Mining*

Sebagai contoh apabila kita akan mengolah data berikut :

Tabel 4.1 Tabel Transaksi Pada *Database* Datamining\_db

Tanggal	<u>Kd transaksi</u>	<u>Kd barang</u>
1/1/2008	100	1
1/1/2008	100	3
1/1/2008	100	4
2/1/2008	101	2
2/1/2008	101	3
2/1/2008	101	5
2/1/2008	102	1
2/1/2008	102	2
2/1/2008	102	3
2/1/2008	102	5
3/1/2008	103	2
3/1/2008	103	5

Langkah pertama dalam meng-eksekusi proses ini adalah dengan menyeleksi tanggal transaksi yang akan dicari kaidah asosiasinya. Dalam perangkat lunak ini proses penyeleksian tanggal ditunjukkan pada gambar berikut :



Gambar 4.3 Tampilan Penyeleksian Tanggal Transaksi

Proses penyeleksian berdasarkan tanggal yang dimasukkan pada Tabel transaksi ditunjukkan oleh kode SQL berikut:

```

$SQL="SELECT
kd_barang,count(kd_barang) as
supp FROM transaksi
WHERE (tanggal>='$tgl_mulai')
and (tanggal<='$tgl_selesai')
GROUP BY kd_barang";
$eksekusi=mysql_query($SQL);
kemudian hasil query tersebut
dimasukkan kedalam Tabel C1 yang
merupakan Tabel kandidat 1-
frequent itemset. Sehingga Tabel C1
tampak sebagai berikut :

```

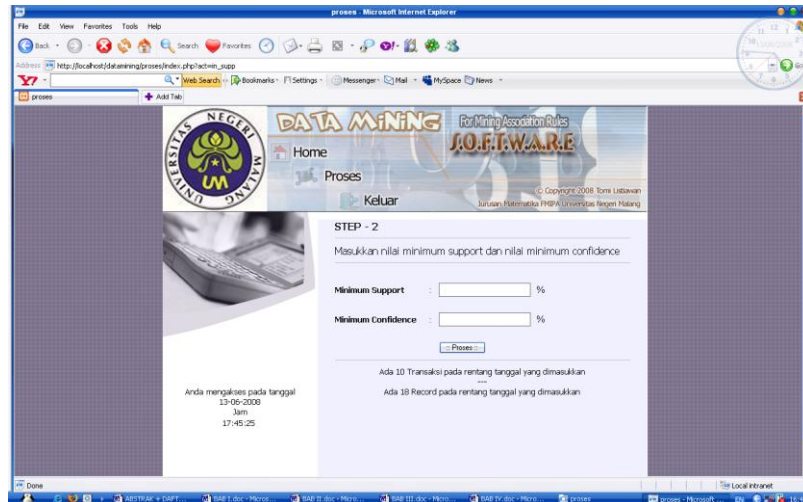
Tabel 4.2 Tabel C1

<u>d_barang1</u>	Support
1	2
2	3
3	3
4	1
5	3

Apabila tidak ditemukan transaksi pada rentang tanggal yang

dimasukkan maka pengguna akan diminta untuk memasukkan kembali rentang tanggal yang benar

Setelah rentang tanggal transaksi dimasukkan, berikutnya perangkat lunak akan meminta pengguna untuk memasukkan nilai *minimum support* dan nilai *minimum confidence*. Besar kecilnya nilai *minimum support* yang dimasukkan ini akan berpengaruh sekali secara keseluruhan bagaimana program akan berlangsung. Semakin *minimum support*, tentunya akan mengakibatkan kandidat *itemset* semakin banyak, yang tentunya akan mengakibatkan program menganalisa lebih banyak data. Tampilan halaman input *minimum support* dan *minimum confidence* ini adalah sebagai berikut



Gambar 4.4 Halaman Input *Minimum Support* dan *Minimum Confidence*

Dari Tabel 4.2, apabila dimasukkan nilai *minimum support* 50% maka akan dipilih item-item yang memiliki *support* di atas 50%, kemudian hasil pemilihan item tersebut dimasukkan ke dalam Tabel L1. Proses pemilihan item di atas *minimum support* dan pembentukan L1 tampak pada kode berikut :

```

$eksekusi=mysql_query("select
* from c1 where
support>=$min_supp");
while
($hasil=mysql_fetch_array($eks
ekusi))
{mysql_query("insert into
l1 (kd_barang1,support)
Values
('$hasil[kd_barang]','$hasil[s
upport]')");}

```

Sehingga Tabel L1 memiliki *record* sebagai berikut :

Tabel 4.3 *record* Tabel L1

Kd_barang1	Support
1	2
2	3
3	3
5	3

Setelah Tabel L1 terbentuk, maka tabel ini akan menjadi input dalam

proses pembentukan *frequent itemset* algoritma apriori. Algoritma apriori dimulai dengan membentuk tabel kandidat *frequent itemset* C2 yang merupakan kombinasi dari L1. Selanjutnya tabel C<sub>k</sub> terbentuk dari kombinasi item dalam L<sub>k-1</sub>, dan Tabel L<sub>k</sub> terbentuk dari Tabel C<sub>k</sub> yang mempunyai nilai *support* di atas nilai *minimum support*. Dalam pembentukan kandidat *frequent itemset* ini terdapat suatu langkah penting yang meningkatkan performansi algoritma apriori dari segi waktu komputasi. Langkah ini disebut *prune step* (pemangkasan). Langkah ini akan memeriksa semua k-1 proper subset anggota suatu kandidat *k-frequent itemset*, dan akan menghapus anggota tersebut jika ada k-1 proper subset yang bukan merupakan anggota dari k-1 *frequent itemset*. Setelah melalui langkah pemangkasan ini barulah kandidat C<sub>k</sub> terbentuk. Untuk tiap kandidat C<sub>k</sub>, dihitung *support*-nya dengan *men-scan database*. Pembentukan Tabel C<sub>k</sub> atau L<sub>k</sub> dalam

program tampak pada potongan kode berikut :

```

    $sql1="create table C/L$n
    (";//$n=$k+1.for ($i=1;$i<=$n;$
    i++){ $sql1.="kd_barang$i
    int(5), ";}$sql1.="primary key
    (";for ($i=1;$i<=$k;$i++)
    {$sql1=$sql1."kd_barang$i, ";}
    $sql1.="kd_barang$n)";
mysql_query($sql1);
mysql_query("alter table C/L$n
add support int(4) null");

```

Setelah tabel terbentuk, selanjutnya untuk memasukkan item  $L_k$  ke dalam  $C_{k+1}$  digunakan *join step* algoritma apriori. Langkah ini dilakukan dengan menggabungkan  $L_k$  dengan  $L_k$  dan memilih kombinasi yang memenuhi syarat *join step* untuk dimasukkan ke dalam  $C_{k+1}$ . Proses *join step* pada program ditunjukkan oleh potongan kode berikut ini :

```

    $sql="SELECT "; for
    ($i=1;$i<=$k-1;$i++)
    {$sql.="P.kd_barang$i, ";}
    $sql=$sql."P.kd_barang$k,Q.kd_ba
    rang$k as kd_barang$n ";
    sql="FROM L$k P, L$k Q WHERE ";
    for ($i=1;$i<=$k-1;$i++)
    {$sql.="P.kd_barang$i=Q.kd_baran
    g$i AND ";}
    $sql.="P.kd_barang$k <
    Q.kd_barang$k";$eksekusi=mysql_q
    uery($sql);

```

Hasil *query* kode di atas kemudian di masukkan kedalam Tabel  $C_{k+1}$ .

Isi Tabel  $C_{k+1}$  di atas merupakan record yang sementara. Artinya record tersebut masih dimungkinkan untuk dikurangi melalui proses *prune step*. *Prune step* dalam program ditunjukkan oleh potongan program berikut :

```

//PRUNE STEP
$kdbr="";
for ($i=1;$i<=$k-1;$i++)

```

```

    {$kdbr=$kd_br."kd_barang$i
    ,";}
    $kdbr=$kdbr."kd_barang$k";
    $array=range(1,$n);
    for ($i=0;$i<$n;$i++)
    {$array2=$array;
    $array2[0]=$array[$i];
    $array2[$i]=$array[0];
    $array3=array_shift($array
    2);//hapus array terdepan
    asort($array2);//mengurutkan
    array $subs="";while
    (list($indeks,$nilai)=each($ar
    ray2))
    {$subs=$subs."kd_barang$nilai
    ";}$pisah=explode(" ",$subs);
    $kd1="";for
    ($j=0;$j<=$k-2;$j++)
    {$kd1=$kd1."$pisah[$j],";}
    $kd1=$kd1.$pisah[$k-1];
    $sql_prune="delete from c$n
    where ($kd1) not in (select
    $kdbr from l$k )";
    mysql_query($sql_prune);}.

```

Tabel  $C_{k+1}$  yang terbentuk dari proses di atas adalah tabel kandidat yang belum diketahui nilai supportnya. Penghitungan support tiap *itemset* dalam tabel kandidat dilakukan dengan proses *scan database*. Proses ini tampak pada potongan program berikut .

```

//hitung jumlah support
melalui jumlah row dari
intersection
while
($hasil=mysql_fetch_array($sql
_C))
    {$sql="";$sql="select
    kd_transaksi from transaksi
    where ";for ($i=1;$i<=$k;$i++)
    {$kd="kd_barang$i";
    $sql.="kd_transaksi in
    (select kd_transaksi from
    transaksi where
    kd_barang=$hasil[$kd]) and ";}
    $kd="kd_barang$n";//$n=$k+1
    $sql.="kd_transaksi in
    (select kd_transaksi

```

```

from transaksi where
kd_barang=$hasil[$kd])
group by kd_transaksi";
$result=mysql_query($sql);//dapa
tкан support
$support=mysql_num_rows($result)
;{

```

Setelah proses pembentukan kandidat selesai, langkah berikutnya adalah proses pembentukan *frequent itemset* (L).  $L_k$  diperoleh dari  $C_k$  yang mempunyai nilai minimum di atas *minimum support*. Pembentukan  $L_k$  ditunjukkan oleh kode berikut .

```

$eksekusi=mysql_query("select
* from c$n where
support>=$min_supp");while
($hasil=mysql_fetch_array($eks
ekusi)){ $insert=""; $values="";
for ($i=1;$i<=$k;$i++)
{$insert=$insert."kd_barang$i,
"; $val="kd_barang$i"; $values=$
values."$hasil[$val], ";}
$insert=$insert."kd_barang
$n, support"; $val="kd_barang$n"

```

Tabel 4.6 Kandidat 3-Frequent Itemset (C3)

kd_barang1	Kd_barang2	Kd_barang3	Support
2	3	5	2

Tabel 4.7 3-Frequent Itemset (L3)

kd_barang1	Kd_barang2	Kd_barang3	Support
2	3	5	2

Setelah semua *frequent itemset* terbentuk langkah selanjutnya adalah pembentukan semua aturan atau kaidah asosiasi yang memenuhi nilai *minimum support* dan nilai *minimum confidence*. Pembentukan aturan asosiasi ini dilakukan dengan memeriksa semua proper subset  $s$  dari semua anggota *frequent itemset*  $l$ , jika  $(\text{supp}(l)/\text{supp}(s) \geq \text{minimum confidence})$ , maka aturan  $s \rightarrow (l - s)$  akan tersimpan, dengan *support* =

```

;$values=$values."$hasil[$val]
,$hasil[support]";
mysql_query("INSERT INTO l$n
($insert) VALUES ($values)");}

```

Dari Tabel  $L_1$  (Tabel 4.3) dapat terbentuk tabel kandidat dan *frequent itemset* sebagai berikut

Tabel 4.4 Kandidat 2-Frequent Itemset (C2)

Kd_barang1	Kd_barang2	Support
1	2	1
1	3	2
1	5	1
2	3	2
2	5	3
3	5	2

Tabel 4.5 2-Frequent Itemset (L2)

Kd_barang1	Kd_barang2	Support
1	3	2
2	3	2
2	5	3
3	5	2

*support(l)* dan *confidence* =  $\text{support}(l)/\text{support}(s)$

Dari contoh pembentukan kandidat *k-frequent itemset* ( $C_k$ ) dan pembentukan *frequent itemset* ( $L_k$ ) di atas, maka aturan asosiasi yang mungkin terbentuk adalah sebagai berikut :

Dari *frequent itemset*  $L_2$  :

1.  $\text{supp}(1,3)/\text{supp}(1) = 2/2=100\%$   
aturan yang terbentuk :  $1 \rightarrow 3$
2.  $\text{supp}(1,3)/\text{supp}(3) = 2/3=66\%$

- aturan yang terbentuk : 3->1
3.  $\text{supp}(2,3)/\text{supp}(2)=2/3=66\%$   
aturan yang terbentuk : 2->3
  4.  $\text{supp}(2,3)/\text{supp}(3) = 2/3=66\%$   
aturan yang terbentuk : 3->2
  5.  $\text{supp}(2,5)/\text{supp}(2)=3/3=100\%$   
aturan yang terbentuk : 2->5
  6.  $\text{supp}(2,5)/\text{supp}(5) = 3/3=100\%$   
aturan yang terbentuk : 5->2
  7.  $\text{supp}(3,5)/\text{supp}(3)=2/3=66\%$   
aturan yang terbentuk : 3->5
  8.  $\text{supp}(3,5)/\text{supp}(5) = 2/3=66\%$   
aturan yang terbentuk : 5->3

Dari *frequent itemset* L3 :

1.  $\text{supp}(2,3,5)/\text{supp}(2)=2/3=66\%$   
aturan yang terbentuk : 2->(3,5)
2.  $\text{supp}(2,3,5)/\text{supp}(3)=2/3=66\%$   
aturan yang terbentuk : 3->(2,5)
3.  $\text{supp}(2,3,5)/\text{supp}(5)=2/3=66\%$   
aturan yang terbentuk : 5->(2,3)
4.  $\text{supp}(2,3,5)/\text{supp}(2,3)=2/2=100\%$   
aturan yang terbentuk : (2,3)->5
5.  $\text{supp}(2,3,5)/\text{supp}(2,5)=2/3=66\%$   
aturan yang terbentuk : (2,5)->3
6.  $\text{supp}(2,3,5)/\text{supp}(3,5)=2/2=100\%$   
aturan yang terbentuk : (3,5)->2.

Jika ditetapkan nilai *minimum confidence* 75%, maka aturan yang tersimpan adalah aturan ke 1, 5, dan 6 dari L2, dan aturan ke 4, dan 6 dari L3.

Pemeriksaan semua *frequent itemset* (L) yang telah terbentuk ditunjukkan oleh kode berikut .

```
$sql_l=mysql_query("select *
froml$k");while
($hasil=mysql_fetch_array($sql
_l)) {.....
.....}
```

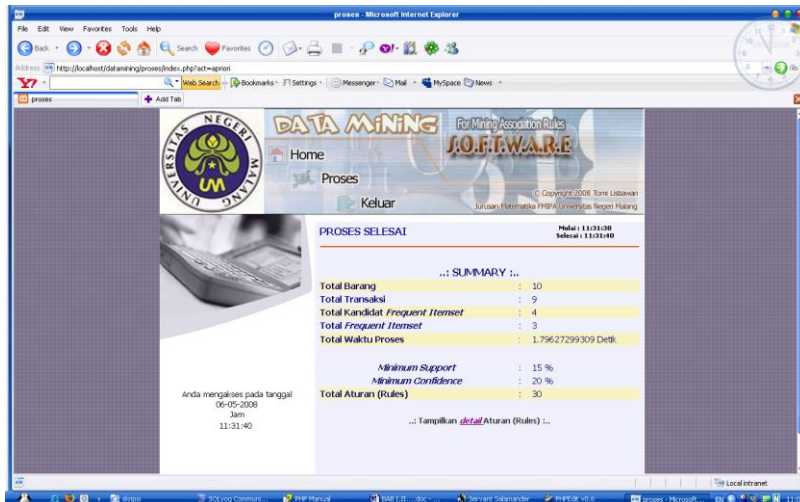
setiap *itemset* anggota L, akan diperiksa proper subsetnya. Pembentukan proper subset ini ditunjukkan oleh kode berikut :

```
//PERIKSA SETIAP PROPER SUBSET
L$inputArr=range(1,$k);
$num=0;$arrLen=count($inputArr);
for($i=0;$i<$arrLen;$i++)
{$str[$num]="kd_barang".$inputArr[$i];$idx=$num;$num++;
for($j=0;$j<$idx;$j++)
{$str[$num]=$str[$j].' ,kd_barang'.
$inputArr[$i];$num++;}}
$jum=1;for($i=1;$i<=$k;$i++)
{$jum=$jum*2;}unset($str[$jum-2]);
//hapus array terakhir
```

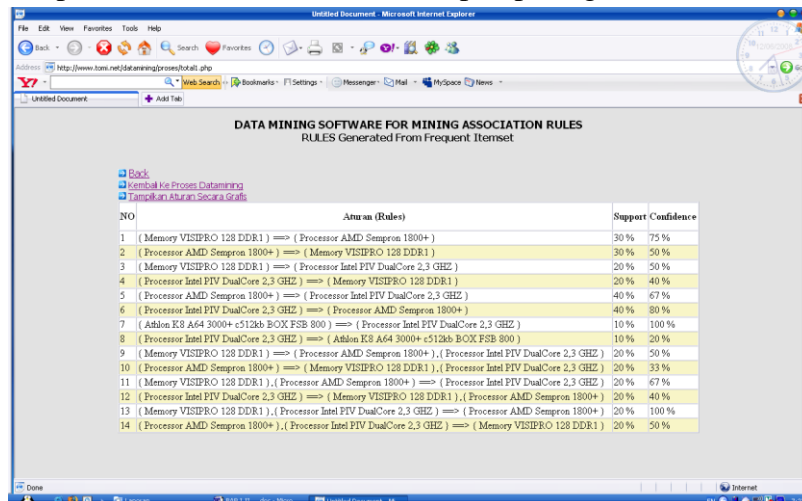
Proper subset yang memenuhi syarat pembentukan aturan akan disimpan ke dalam Tabel aturan, yang ditunjukkan oleh kode berikut.

```
$s[]=.....;// s ⊂ l ∈ L
$l[]=.....;
$skiri=.....;//subset yang
terambil saat iterasi
$conf=$suppl/$supps;$support_L=(
$suppl*100)/$total_transaksi;
$confidence=$conf*100;if
($conf>=$min_conf){$result =
array_diff($l, $s);//L
$kanan=explode(",",$result);
$jumlah_rules++;mysql_query("ins
ert into aturan
(jika,maka,support,confidence)
Values('$skiri','$kanan',$support
_L,$confidence)");}
```

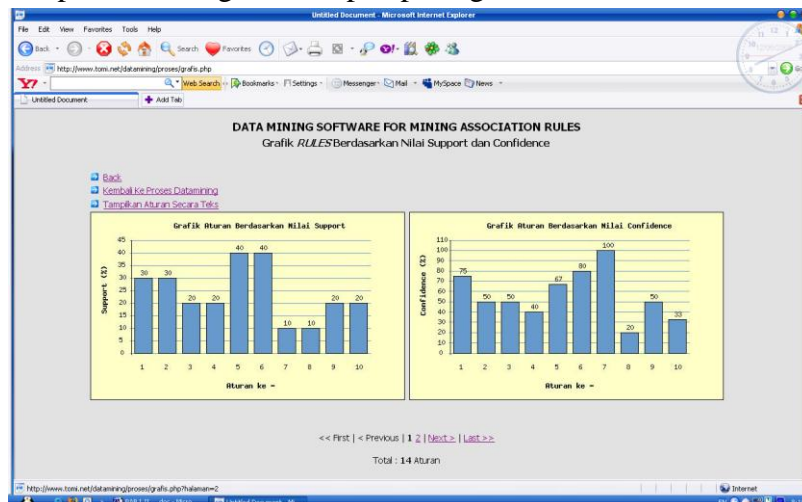
Proses input ke dalam Tabel Aturan tersebut merupakan akhir dari proses penggalian kaidah asosiasi ini. Tampilan pada program tampak pada gambar berikut.



Gambar 4.5 Tampilan Summary Proses Penggalan Kaidah Asosiasi  
 Apabila link *detail* diklik, maka akan ditampilkan seluruh aturan yang telah terbentuk. Tampilan halaman detail aturan tampak pada gambar berikut



Gambar 4.6 Tampilan Halaman Detail Aturan.  
 Sedangkan tampilan secara grafis tampak pada gambar berikut :



Gambar 4.7 Tampilan Halaman Detail Aturan Secara Grafis



### C. Uji Coba dan Analisa

Untuk mengetahui performansi perangkat lunak, dilakukan uji coba dengan beberapa data yang berbeda, dengan jumlah record yang relatif besar. Perangkat keras yang dipergunakan pada uji coba ini adalah komputer dengan prosesor Intel Celeron M 1,86 GHz dengan memori (RAM) sebesar 512 MB. Sedangkan sistem operasi yang digunakan adalah Windows XP Professional SP2.

Pada uji coba yang akan dilakukan, digunakan 3 data yang berbeda, yaitu data1, data2, dan data3. Data tersebut diperoleh dengan cara melakukan input pada Tabel transaksi dan Tabel detail transaksi secara random, dengan rentang data yang ditentukan dengan menggunakan perintah-perintah PHP. Untuk memperoleh data1, langkah pertama adalah membuat Tabel transaksi dengan jumlah record 301 transaksi, dan membuat Tabel barang dengan record berjumlah 20. Kemudian kode transaksi dan kode barang yang telah terbentuk, dirandom sebanyak 3676 kali untuk dimasukkan kedalam kolom *kd\_transaksi* dan kolom *kd\_barang* Tabel detail transaksi. Untuk membuat data yang sesuai untuk kebutuhan input perangkat lunak, langkah terakhir adalah menggabungkan Tabel transaksi dan detail transaksi berdasarkan kode transaksi yang sama. Demikian pula langkah yang digunakan untuk membuat data2 dan data3.

Spesifikasi ketiga data yang akan digunakan tampak pada tabel berikut.

Tabel 4.7 Spesifikasi Data Untuk Uji Coba

	Data1	Data2	Data3
<b>Jumlah record</b>	3676	2454	1198
<b>Jumlah transaksi</b>	301	201	101
<b>Jumlah barang</b>	20	20	20

Pada uji coba ini, yang akan dilakukan adalah menjalankan perangkat lunak dengan parameter yang sama pada tabel yang berbeda. Tabel yang digunakan ada 3 yaitu : data1, data2, dan data3. Sedangkan parameter yang digunakan adalah *minimum support* dan *minimum confidence*. Nilai *minimum support* yang dimasukkan adalah 25%, 35%, dan 40%. Sedangkan nilai *minimum confidence* yang dimasukkan adalah 65%.

Hasil uji coba untuk masing-masing tabel dapat dilihat pada tabel berikut.

Tabel 4.8 Hasil Uji Coba Menggunakan Data1

<b>Minimum Support</b>	25%	35%	40%
<b>Waktu proses (detik)</b>	8,80	5,86	1,187
<b>Jumlah Kandidat</b>	4	3	3
<b>Jumlah Frequent</b>	3	2	2
<b>Jumlah aturan</b>	286	38	24

Tabel 4.9 Hasil Uji Coba Menggunakan Data2

Minimum Support	25%	35%	40%
Waktu proses (detik)	11,1	2,6	1,36
Jumlah Kandidat	4	3	3
Jumlah Frequent	3	2	2
Jumlah aturan	590	98	64

Tabel 4.11 Hasil Uji Coba Menggunakan Data3

Minimum Support	25%	35%	40%
Waktu proses (detik)	6,8	1,64	0,90
Jumlah Kandidat	4	3	3
Jumlah Frequent	3	2	2
Jumlah aturan	389	68	40

Dari hasil uji coba di atas terdapat suatu temuan yang menarik, yaitu jumlah transaksi bukanlah satu-satunya faktor yang mempengaruhi waktu proses penggalian kaidah asosiasi. Artinya, jumlah transaksi yang lebih besar belum tentu membutuhkan waktu komputasi yang lebih besar juga. Beberapa faktor yang mempengaruhi waktu proses komputasi yaitu jumlah kemunculan setiap item pada transaksi, jumlah transaksi, input *minimum support* dan *minimum confidence*. Hal ini sesuai dengan kesimpulan yang diambil pada penelitian yang dilakukan oleh Andreas Handojo

(Staf pengajar Jurusan Teknik Informatika Universitas Kristen Petra), dengan judul "Aplikasi Data Mining Untuk Meneliti Asosiasi Pembelian Item Barang Di Supermarket Dengan Metode Market Basket Analysis".

## 5. Kesimpulan

Dari pembuatan perangkat lunak *data mining* ini, dapat diambil beberapa kesimpulan sebagai berikut Aturan asosiasi yang dihasilkan oleh perangkat lunak, bisa digunakan sebagai bahan pertimbangan dalam proses pengambilan keputusan di suatu toko atau pasar swalayan, seperti pengaturan tata letak barang, stok barang, segmentasi pembeli, merancang kampanye pemasaran untuk kombinasi suatu barang.

1. Aplikasi mampu mengolah data transaksi untuk menemukan kandidat dan *frequent itemset*, kemudian *generate association rules* untuk ditampilkan dalam bentuk teks dan grafis.
2. Pada analisa terhadap sejumlah data, ditemukan bahwa semakin kecil *minimum support* dan *confidence* yang ditentukan, semakin banyak pula aturan asosiasi yang dapat dihasilkan oleh aplikasi.
3. Dari pengujian ditemukan bahwa proses yang membutuhkan waktu paling lama, umumnya terjadi pada proses pembuatan kandidat 2-itemset. Semakin banyak jumlah kandidat 1-itemset yang memenuhi minimum support, maka semakin banyak pula jumlah kandidat 2-itemset yang harus dihasilkan.

4. Jumlah kandidat 1-itemset sulit dipastikan, karena bergantung sekali pada jumlah kemunculan setiap item pada transaksi, jumlah transaksi, dan input *minimum support* sehingga proses untuk menghasilkan kandidat 2-itemset pun sulit dipastikan yang menyebabkan waktu keseluruhan proses pun sulit untuk diperkirakan atau disimpulkan.

## 6. Daftar Pustaka

- Adam, Rinaldi. 2004. *Flowcharting*,(Online), ([http://www.cic.ac.id/modulkuliah/download/Flowchart%201\\_files.htm](http://www.cic.ac.id/modulkuliah/download/Flowchart%201_files.htm), diakses pada 21 Mei 2007)
- Chung, Christina. *Applying Data Mining to Data Security*,(Online), (<http://sirius.cs.ucdavis.edu/teaching/289F/> ,diakses pada 21 Mei 2007).
- Fathansyah.2001. *Basis Data*.Bandung:Penerbit Informatika.
- Handojo, Andreas. *Aplikasi Data Mining Untuk Meneliti Asosiasi Pembelian Item Barang Di Supermarket Dengan Metode Market Basket Analysis*. Penulisan penelitian. Tidak diterbitkan
- Jiawei Han and Micheline Kamber. *Data mining, concept and techniques*.(Online) (<http://www.cs.sfu.ca>, diakses pada 21 Mei 2007)
- Kadir, Abdul.2003. *Dasar Pemrograman Web Dinamis Menggunakan PHP*.Yogyakarta. PT. Andi.
- Microsoft Corporation. *OLEDB for Data Mining draft specification, version 0.9*.(Online) (<http://www.microsoft.com/dat> a/oledb/dm, diakses pada 21 Mei 2007).
- Nugroho, Bunafit.2004. *Aplikasi Pemrograman Web Dinamis Dengan PHP Dan MySQL*.Yogyakarta:Gava Media.
- Prasetyo, Didik Dwi.2005. *Solusi Menjadi web Master Melalui Manajemen web Dengan PHP*.Jakarta:Elex Media Komputindo
- Rantzau, Ralf. *Exrended Concept For Association Rules Discovery*.Penulisan Penelitian.Tidak diterbitkan
- Santosa, Budi.2007. *Data Mining Teknik Pemanfaatan Data Untuk Keperluan Bisnis*.Yogyakarta:Graha Ilmu.
- Son, Nguyen Hung. *Transaction data analysis and association rules*.(Online) (<http://www.mimuw.edu.pl/~son/datamining>, diakses pada 21 Mei 2007)
- Sucahyo, Yudo Giri.2003. *Data mining – Menggali Informasi Yang terpendam*.(Online) (<http://ikc.cbn.net.id/populer/yudho/yudho-datamining.zip>, diakses pada 21 Mei 2007)
- Syukur, Mark Ade. 1999. *Aplikasi Web dengan PHP*. Penulisan Penelitian. Tidak diterbitkan.