

**Implementasi Algoritma RuleGrowth Untuk
Memprediksi Urutan Penjualan Barang
Dari Data Penjualan Barang
Pusat Perkulakan**

Proposal Tugas Akhir



Diajukan oleh:
Michael Roynaldo
71130086

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA

2016

Implementasi Algoritma RuleGrowth Untuk Memprediksi Urutan Penjualan
Barang Dari Data Penjualan Barang Pusat Perkulakan

Bidang Minat	:	Data Mining	
Mata kuliah pendukung	:	1. Pemrograman Berorientasi Objek	A-
		2. Desain dan Analisis Algoritma	A-
		3. Sistem Basis Data	A

Nama Mahasiswa	:	Michael Roynaldo
N I M	:	71130086
IPK	:	3,86

Proposal ini diajukan dalam kolokium sebagai persyaratan untuk memulai penulisan tugas akhir.

Proposal I
Semester Gasal 2016/2017
2016

1. Latar Belakang

Di tengah pesatnya perkembangan dunia bisnis sekarang ini, banyak bisnis pedagang eceran (*retail*) dan pasar (*market*) yang banyak bermunculan di Indonesia. Meningkatnya jumlah bisnis ritel dan pasar ini muncul dikarenakan kebutuhan hidup manusia yang semakin meningkat dari hari ke hari. Oleh karena jumlah kebutuhan yang makin meningkat tersebut barang kebutuhan pokok sehari-hari yang harus disediakan oleh toko ritel (pedagang eceran) dan pasar pun semakin meningkat. Efek dari hal tersebut dirasakan oleh perusahaan perkulakan yang harus selalu mempersiapkan barang kebutuhan pokok sehari-hari agar toko ritel dan *market* selalu memiliki stok yang cukup untuk memenuhi kebutuhan hidup masyarakat dari hari ke hari. Oleh karena itu, diperlukan pengelolaan barang yang baik oleh perusahaan perkulakan agar dapat menyediakan barang kebutuhan pokok sehari-hari sesuai dengan kebutuhan pembeli seperti toko ritel, pedagang eceran, pasar (*market*) atau pembeli perorangan.

Dalam kasus ini, perusahaan perkulakan yang akan dibahas merupakan perusahaan dagang di bidang ritel perkulakan yang menyediakan barang kebutuhan pokok sehari-hari, melayani pedagang eceran (*retailers*) maupun konsumen rumah tangga (*end users*). Perusahaan perkulakan pun akan kesulitan bila tidak menggunakan cara yang tepat untuk memprediksi barang-barang kebutuhan pokok sehari-hari yang dibutuhkan oleh pedagang eceran ataupun konsumen. Bila memprediksi barang yang dibutuhkan secara manual pun akan dibutuhkan waktu yang sangat lama karena proses transaksi penjualan barang yang selalu bertambah dari waktu ke waktu. Oleh karena itu, data transaksi penjualan barang tersebut dapat dimanfaatkan dengan menggunakan bantuan algoritma *data mining*. *Data mining* sendiri adalah suatu cara untuk mencari informasi / pengetahuan yang terkandung dari kumpulan *data* mentah. Data yang digunakan pada penelitian ini adalah data transaksi penjualan barang dan data barang dalam jangka waktu 2 tahun. Data-data tersebut kemudian diproses menggunakan algoritma *data mining*, hasil yang didapat kemudian dianalisis agar dapat digunakan sebagai dasar dalam mengambil keputusan.

Untuk melakukan proses *data mining*, diperlukan *Data Warehouse* agar dapat memproses data transaksi penjualan barang yang jumlahnya sangatlah banyak (*Big Data*) dengan efektif dan efisien. Setelah itu data tersebut akan diimplementasikan menggunakan proses *Data Mining* yang menggunakan algoritma *sequential rule mining* adalah algoritma *RuleGrowth*. Algoritma *RuleGrowth* ini digunakan untuk mencari keterkaitan antar barang dari waktu ke waktu. Hasil yang didapat dari algoritma tersebut kemudian dianalisis agar dapat digunakan untuk memprediksi urutan penjualan barang dari waktu ke waktu.

2. Rumusan Masalah

Rumusan masalah yang muncul dari latar belakang dilakukannya penelitian ini antara lain :

1. Bagaimana sistem dapat menerapkan algoritma *RuleGrowth* untuk membantu menganalisis keterkaitan antar barang dan memprediksi urutan penjualan barang kebutuhan pokok sehari-hari dari data transaksi penjualan barang yang terjadi pada pusat perkulakan barang kebutuhan pokok sehari-hari?
2. Seberapa baikkah hasil yang didapat dari penggunaan algoritma *RuleGrowth* dalam menunjukkan keterkaitan antar barang serta memprediksi penjualan urutan barang dari waktu ke waktu?

3. Batasan Masalah

Batasan masalah yang digunakan oleh penulis dalam pembuatan sistem pada penelitian ini adalah sebagai berikut :

1. Data yang digunakan adalah data yang terkait dengan transaksi penjualan barang pada pusat perkulakan barang kebutuhan pokok sehari-hari dalam jangka waktu 2 (dua) tahun terhitung dari 2014-2016. Beberapa data tersebut meliputi data transaksi penjualan barang dan data barang kebutuhan pokok sehari-hari dari salah satu cabang pusat perkulakan.

2. Pembangunan dan pengembangan *Data Warehouse* dalam mengolah *Big Data* akan menggunakan aplikasi Hadoop dan Hive.
3. Untuk bagian *back-end* diolah pada *Java Server Pages* menggunakan Tomcat Apache Server.
4. Algoritma *sequential rule mining* yang akan digunakan adalah algoritma *RuleGrowth* yang disediakan pada *library* SPMF, dilengkapi dengan fungsi *left expansions* dan *right expansions*.
5. Output sistem akan berupa tabel yang berisikan keterkaitan antar barang kebutuhan pokok sehari-hari dari waktu ke waktu. Bersamaan dengan tabel output tersebut, sistem akan memberikan hasil prediksi dan seberapa kuatnya urutan penjualan barang dari keterkaitan antar barang tersebut.

4. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat suatu aplikasi yang dapat menampilkan bagaimana keterkaitan antar barang menggunakan data transaksi penjualan barang oleh pusat perkulakan barang kebutuhan pokok sehari-hari menggunakan algoritma *RuleGrowth* dan dapat memberikan hasil analisis berupa prediksi urutan penjualan barang dari waktu ke waktu berdasarkan keterkaitan antar barang tersebut agar dapat membantu untuk mengambil keputusan, meningkatkan penjualan, pemasaran dan pengelolaan barang dari pusat perkulakan.

5. Manfaat Penelitian

Manfaat dari penelitian ini bagi penulis adalah untuk menerapkan dan memperluas wawasan dalam menerapkan algoritma *data mining* pada kasus yang nyata dan juga menyelesaikan tugas akhir perkuliahan.

Manfaat bagi pusat perkulakan barang kebutuhan pokok sehari-hari adalah untuk meningkatkan penjualan, pemasaran dan pengelolaan barang dengan bantuan sistem yang sudah dibangun pada penelitian ini. Pusat perkulakan juga dapat melihat hasil pengolahan data mentah menjadi data yang dapat berguna bagi

kemajuan perusahaan. Diharapkan sistem ini juga menjadi alat bantu perusahaan dalam memprediksi kebutuhan stok suatu barang serta membantu dalam pemetaan atau pengelompokkan barang dalam penjualan barang. Perusahaan juga dapat lebih mudah dalam melakukan promosi-promosi untuk menjual suatu barang berdasarkan hasil prediksi dari sistem yang akan dibangun ini.

Manfaat bagi peneliti selanjutnya, diharapkan dengan dilakukannya penilitan ini dapat menjadi bahan atau dasar bagi penelitian-penelitian selanjutnya yang masih berkaitan dengan *data mining*. Selain itu, penelitian ini juga dapat dikembangkan kembali menggunakan algoritma yang berbeda, atau algoritma yang digunakan sama namun pada kasus yang berbeda, atau bisa juga melakukan pendekatan yang berbeda dalam melakukan analisis.

6. Landasan Teori

6.1. Data Mining

Menurut Yulianton (2008), Definisi *data mining* secara formal adalah proses mengekstrak informasi atau pengetahuan yang valid, bermanfaat, tak dikenal, dan dapat dipahami dari data - data mentah yang berskala besar. Informasi yang diekstrak pun harus benar dan valid agar dapat digunakan sebagai pendukung dalam mengambil keputusan. Validitas informasi berarti kebenaran dan juga kelengkapan informasi dari pengolahan data. Informasi tersebut kemudian digunakan sebagai alat bantu dalam mengambil keputusan bisnis. Biasanya penggunaan *data mining* ini digunakan pada sistem-sistem yang ditujukan untuk menganalisis data-data yang dapat digunakan sebagai pertimbangan dalam mengambil keputusan bisnis.

Data mining sendiri dipahami sebagai suatu proses yang memiliki beberapa tahapan yang pada setiap tahapannya memberikan hasil yang dapat diolah kembali hingga informasi yang valid didapatkan. Kesuksesan proses *Mining* dalam suatu sistem akan sangat ditentukan oleh setiap tahapan-tahapan dalam memproses data.

Tahapan-tahapan yang dilalui data mentah hingga menjadi informasi/pengetahuan antara lain (Yulianton, 2008)

- Pembersihan Data

Pembersihan data adalah proses menghilangkan *noise* dan data yang tidak konsisten atau tidak relevan. Tahap ini dilakukan dikarenakan *database* suatu perusahaan maupun hasil eksperimen pada umumnya memiliki isian yang tidak sempurna, seperti ada data yang hilang, tidak valid dan bisa juga dikarenakan kesalahan pengetikan. Pembersihan data ini menjadi penting agar proses *mining* menjadi lebih akurat dalam menyajikan informasi.

- Integrasi Data

Integrasi data adalah proses menggabungkan data dari berbagai *database* ke dalam satu *database* baru. Integrasi data ini bisa juga berasal dari *file* teks atau *file* tabel seperti *file* excel. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasikan entitas yang unik seperti nama, nomor pelanggan, nomor transaksi, dan sebagainya. Integrasi data perlu dilakukan secara teliti agar tidak terjadi kesalahan, bila terjadi kesalahan maka hasil informasi akan menyimpang dan menyesatkan dalam mengambil keputusan.

- Seleksi Data

Pada umumnya, data yang terdapat di *database* tidak semuanya akan digunakan dalam proses *mining*. Oleh karena itu hanya data yang sesuai kebutuhan yang akan diambil dari *database*.

- Transformasi Data

Data diubah atau digabung ke dalam format yang sesuai agar dapat diproses dalam proses *Mining*. Transformasi data diperlukan karena ada beberapa metode *mining* yang membutuhkan format khusus sebelum bisa diaplikasikan.

Seperti contohnya menggunakan metode *clustering* yang berkaitan dengan angka-angka (tipe *number*).

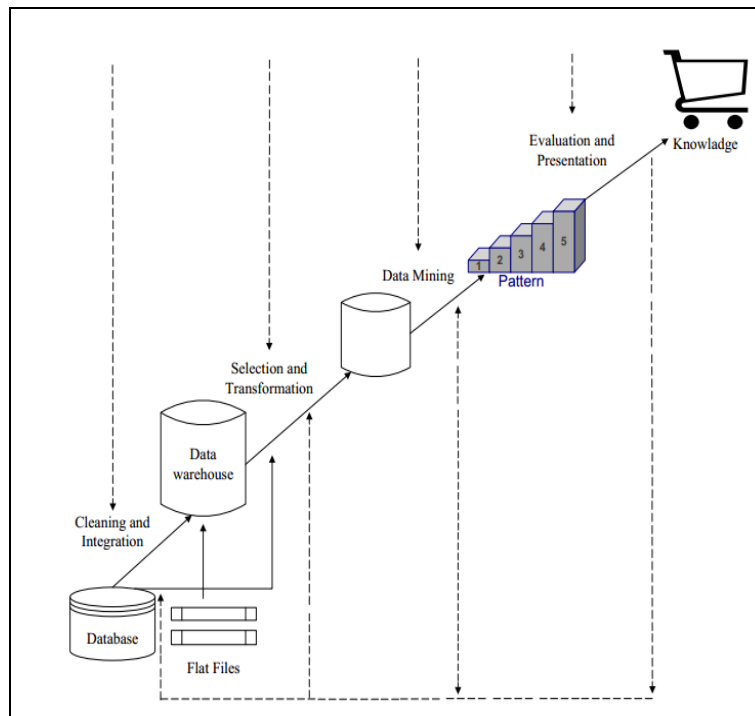
- Proses *Data Mining*

Proses ini adalah proses utama dimana data yang sudah siap akan diuji menggunakan metode *Mining* agar dapat diperoleh informasi atau pengetahuan yang berharga dan tersembunyi dari data. Proses *Mining* yang umum digunakan pada sebuah sistem adalah *association rule mining*, *sequential pattern mining*, *sequential rule mining*, *frequent itemset mining*, *clustering*, *classification*, dan *forecasting*.

- Presentasi Pengetahuan

Proses ini ditujukan untuk mengidentifikasi pola-pola unik yang dapat digunakan sebagai bahan keputusan. Bila didapat hasil yang tidak sesuai dengan hipotesa maka bisa dilakukan perbaikan pada proses *mining*, atau mencoba metode lain yang lebih sesuai agar informasi yang didapat lebih akurat.

Gambaran tahapan data mining dalam memproses data hingga menjadi informasi/pengetahuan dapat dilihat pada gambar 1.



Gambar 1. Tahapan Data Mining. (Yulianton, 2008)

6.2. Sequential Pattern Mining dan Sequential Rule Mining

Sequential pattern mining digunakan untuk mencari pola data yang memiliki urutan, data tersebut bisa berupa urutan transaksi. Pola tersebut akan mudah didapatkan apabila data yang disimpan dalam jumlah yang besar dan setiap objeknya muncul secara berulang kali. (Juliastio, 2015)

Contoh algoritmanya adalah *Generalizes Sequential Pattern* (GSP), *FreeSpan*, *SPADE*, dan sebagainya. Untuk contoh proses *sequential pattern mining* dapat digambarkan sebagai berikut, terdapat tabel transaksi penjualan yang berisikan data-data customer (pembeli), tanggal pembelian dan item yang sudah dibeli. Dari tabel transaksi penjualan tersebut kemudian dibentuk *sequence* (urutan) transaksi berdasarkan customer (pembeli) dan diurutkan berdasarkan tanggal pembelian sehingga membentuk beberapa *sequence*. Kemudian dari beberapa *sequence* tersebut diolah menggunakan metode yang dapat menentukan pola apa saja yang terbentuk pada transaksi tersebut. (Juliastio, 2015)

Sequential rule mining berbeda dengan *sequential pattern mining*. Pada penggunaan metode *sequential pattern mining* memiliki keterbatasan dalam melakukan *predictions* (prediksi). Sebagai contoh dari pola X Y, ada kemungkinan bahwa Y akan muncul setelah X, namun ada kemungkinan muncul kasus X muncul tidak selalu diikuti oleh Y. Untuk itu, diperlukan pengukuran yang lebih menyakinkan bila X muncul maka Y akan muncul juga. Oleh karena itu, digunakanlah *Sequential Rule Mining* yang dapat membantu meningkatkan prediksi pola munculnya *item*.

Pada *sequential rule* ditandai dengan tanda panah (\rightarrow) untuk menunjukkan kelanjutan, misal $\{X\} \rightarrow \{Y\}$ dan juga memperhitungkan *confidence* dan *support*. Algoritma yang biasanya digunakan antara lain CMRules, CMDeo dan RuleGrowth. (Fournier-Viger, Wu, Tseng dan Cao, 2015).

Contoh penentuan *rule* dari *sequence database* dapat dilihat pada gambar 2.

Consider *minSup*= 0.5 and *minConf*= 0.5:

ID	Sequences
<i>seq1</i>	$\{a, b\}, \{c\}, \{f\}, \{g\}, \{e\}$
<i>seq2</i>	$\{a, d\}, \{c\}, \{b\}, \{a, b, e, f\}$
<i>seq3</i>	$\{a\}, \{b\}, \{f\}, \{e\}$
<i>seq4</i>	$\{b\}, \{f, g\}$

A sequence database

\rightarrow

ID	Rule	Support	Confidence
r1	$\{a, b, c\} \Rightarrow \{e\}$	0.5	1.0
r2	$\{a\} \rightarrow \{c, e, f\}$	0.5	0.66
r3	$\{a, b\} \rightarrow \{e, f\}$	0.5	1.0
r4	$\{b\} \rightarrow \{e, f\}$	0.75	0.75
r5	$\{a\} \rightarrow \{e, f\}$	0.75	1.0
r6	$\{c\} \rightarrow \{f\}$	0.5	1.0
r7	$\{a\} \rightarrow \{b\}$	0.5	0.66
...

Some rules found

Gambar 2. Contoh Penentuan Rule dari Sequence Database.

(Fournier-Viger, Nikambou, Tsemh, 2011).

6.3. Algoritma RuleGrowth

Algoritma *RuleGrowth* merupakan salah satu algoritma yang digunakan untuk memprediksi urutan kemunculan data dari waktu ke waktu. Algoritma *RuleGrowth* termasuk dalam algoritma *sequential rule mining*.

Algoritma *RuleGrowth* ini dikenal lebih cepat dan baik dalam segi penggunaan memori dari pada algoritma *sequential rule* mining yang lainnya, algoritma CMRules dan CMDeo. (Fournier-Viger, Nikambou dan Tsemh 2011).

Gambaran langkah pada algoritma *RuleGrowth* dalam menyusun *rule* antara lain (Fournier-Viger, Nikambou dan Tsemh, 2011) :

- Pertama algoritma ini akan membaca semua *item* yang terdapat dalam *sequences database*. *Sequences database* sendiri dibuat berdasarkan transaksi penjualan barang dari waktu ke waktu oleh konsumen/*customer*.
- Kemudian dari semua *item* yang ditemukan, susun semua kemungkinan *rule* $X \rightarrow Y$ dengan 2 *item* yang berbeda.
- Dari setiap *rule* $X \rightarrow Y$, temukan *rule* yang lebih besar secara rekursif dengan menambahkan satu *item* ke sisi kiri ataupun kanan, menggunakan fungsi *left expansion* dan *right expansion*.

Catatan :

Fungsi *right expansion*

- Selama pembuatan *rule-rule* tersebut, cari *rule-rule* yang memiliki *support* dan *confidence* lebih besar sama dengan *minimum support* dan *minimum confidence*.

Untuk algoritma *RuleGrowth* dapat dilihat pada gambar 3. Sedangkan untuk fungsi *left expansion* dan *right expansion* dapat dilihat dari gambar 4 dan 5.

```

RULEGROWTH(database, minsup, minconf)
1. Scan the database one time. During this scan, for each
   item  $c$ , record the  $sids$  of the sequences that contains  $c$ 
   in a variable  $sids\_c$  and the position of the first and last
   occurrence of  $c$  for each  $sid$  in hash tables respectively
   named  $firstOccurences\_c$  and  $lastOccurences\_c$ .
2. FOR each pairs of items  $i, j$  such that  $|sids\_i| \geq minsup$ 
   and  $|sids\_j| \geq minsup$ :
3.    $sidsi \sqcup j := \{\}$ .
4.    $sidsj \sqcup i := \{\}$ .
5.   FOR each  $sid$   $s$  such that  $s \in sids\_i$  and  $s \in sids\_j$ 
6.     IF  $firstOccurences\_i(s)$  is before
        $lastOccurences\_j(s)$ , THEN  $sidsi \sqcup j := sidsi \sqcup j \cup \{s\}$ .
7.     IF  $firstOccurences\_j(s)$  is before
        $lastOccurences\_i(s)$ , THEN  $sidsj \sqcup i := sidsj \sqcup i \cup \{s\}$ .
8.   END FOR
9.   IF  $(|sidsi \sqcup j| / |database|) \geq minsup$  THEN
10.    EXPANDLEFT( $\{i\} \Rightarrow \{j\}$ ,  $sids\_i$ ,  $sidsi \sqcup j$ ,
        $lastOccurences\_j$ ).
11.    EXPANDRIGHT( $\{i\} \Rightarrow \{j\}$ ,  $sids\_i$ ,  $sids\_j$ ,  $sidsi \sqcup j$ ,
        $firstOccurences\_i$ ,  $lastOccurences\_j$ ).
12.    IF  $(|sidsi \sqcup j| / |sids\_i|) \geq minconf$  THEN OUTPUT
       rule  $\{i\} \Rightarrow \{j\}$  with its confidence and support.
13.  END IF
14.  ... [lines 9 to 13 are repeated here with  $i$  and  $j$ 
       swapped]...
15. END FOR

```

Gambar 4. Algoritma RuleGrowth.

(Fournier-Viger, Nikambou, Tsemh, 2011).

```

EXPANDLEFT(ruleIJ, sidsI, sidsI  $\sqcup$  J, lastOccurences_J)
1. FOR each  $sid \in sidsI \sqcup J$ , scan the sequence  $sid$  from the
   first itemset to the itemset before the last occurrence of
   J. For each item  $c$  appearing in these sequences that is
   lexically larger than items in I, record in a variable
    $sidsIc \sqcup J$  the  $sids$  of the sequences where  $c$  is found.
2. END FOR
3. FOR each item  $c$  such that  $(sidsIc \sqcup J / |S|) \geq minsup$  :
4.    $sidsIc := \{\}$ .
5.   FOR each  $sid \in sidsI$ 
6.     IF( $sid \in sids\_c$ ) THEN  $sidsIc := sidsIc$ 
        $\cup \{sid\}$ .
7.   END FOR
8.   EXPANDLEFT( $I \cup \{c\} \Rightarrow J$ ,  $sidsIc$ ,  $sidsIc \sqcup J$ ,
        $lastOccurences\_J$ ).
9.   IF  $(|sidsIc \sqcup J| / |sidsI|) \geq minconf$  THEN OUTPUT
       rule  $I \cup \{c\} \Rightarrow J$ .
10.  END FOR

```

Gambar 5. Fungsi Left Expansion.

(Fournier-Viger, Nikambou, Tsemh, 2011).

EXPANDRIGHT(ruleIJ, <i>sidsI</i> , <i>sidsJ</i> , <i>sidsI</i> ■ <i>J</i> , firstOccurences_I, lastOccurences_J)			
1.	FOR each <i>sid</i> ∈ <i>sidsI</i> ■ <i>J</i> , scan the sequence <i>sid</i> from the itemset after the first occurrence of I to the last itemset. For each item <i>c</i> appearing in these sequences that is lexically larger than items in J, record in a variable <i>sidsI</i> ■ <i>Jc</i> the <i>sids</i> of the sequences where <i>c</i> is found.		
2.	END FOR		
3.	FOR each item <i>c</i> such that $ sidsI \blacksquare Jc \geq minsup * S $:		
4.	<i>sidsJc</i> = {}.		
5.	<i>lastOccurences_Jc</i> := <i>lastOccurences_J</i> .		
6.	FOR each <i>sid</i> ∈ <i>sidsJ</i>		
7.	IF(<i>sid</i> ∈ <i>c.sids</i>) THEN		
8.	<i>sidsJc</i> := <i>sidsJc</i> ∪ { <i>sid</i> }.		
9.	Update(<i>lastOccurences_Jc</i> , <i>lastOccurences_J</i> , <i>c</i> , <i>sid</i>).		
10.	END IF		
11.	END FOR		
12.	EXPANDLEFT($I \Rightarrow JU\{c\}$, <i>sidsI</i> , <i>sidsI</i> ■ <i>Jc</i> , <i>lastOccurence_Jc</i>).		
13.	EXPANDRIGHT($I \Rightarrow JU\{c\}$, <i>sidsI</i> , <i>sidsJC</i> , <i>sidsI</i> ■ <i>Jc</i> , firstOccurences_I, lastOccurences_Jc).		
14.	IF $(sidsI \blacksquare Jc / sidsI) \geq minconf$ THEN		
15.	OUTPUT rule $I \Rightarrow JU\{c\}$.		
16.	ENDIF		
17.	END FOR		

Gambar 6. Fungsi Righth Expansion.

(Fournier-Viger, Nikambou, Tsemh, 2011).

6.4. Studi Kasus RuleGrowth

Untuk lebih memahami cara penggunaan algoritma RuleGrowth, penulis memberikan sebuah contoh untuk menggambarkan proses jalannya algoritma RuleGrowth.

Tabel 1. Contoh Sequence Database

Seq ID	Sequences
seq1	{a, b}, {c}, {f }, {g}, {e}
seq2	{a, d}, {c}, {b}, {a, b, e, f}
seq3	{a}, {b}, {f}, {e}
seq4	{b}, {f, g}

Tabel 1 di atas menggambarkan sebuah database yang saling berhubungan dengan 4 *sequences* (urutan transaksi per konsumen), dengan masing-masing *sequences* memiliki identitas yang berbeda-beda (seq1, seq2, seq3, seq4). *Sequence* seq1 menunjukkan transaksi yang dilakukan oleh konsumen 1, seq2 menunjukkan transaksi yang dilakukan oleh konsumen 2, dst.

Setiap kata yang ada di *database* menunjukkan sebuah barang ('a' untuk pensil, 'b' untuk buku, 'c' untuk marker, dll). Setiap *sequences* terdiri dari sekumpulan barang yang dibeli dari waktu ke waktu.

Contohnya pada 'seq2', konsumen 2 membeli barang 'a' dan 'd' bersama, kemudian membeli barang 'c', kemudian membeli barang 'b' dan terakhir membeli barang 'a', 'b', 'e' dan 'f' bersama-sama.

Sebuah *sequential rule* memiliki struktur $X \rightarrow Y$. *Rule* tersebut mendeskripsikan sebuah hubungan antara 2 barang X dan Y, yang berarti barang Y akan muncul atau dibeli setelah barang X dibeli terlebih dahulu. Contoh diketahui *rule* seperti berikut ini, $\{a\} \rightarrow \{e, f\}$. *Rule* tersebut berarti jika konsumen membeli barang 'a', maka konsumen akan membeli barang 'e' dan 'f'. Untuk $\{e, f\}$ berarti kedua barang tersebut dibeli bersamaan pada satu transaksi.

Untuk menemukan *sequential rule* pada *sequence database*, terdapat 2 alat pengukuran yaitu *support* dan *confidence*.

- *Support* dari *rule* $X \rightarrow Y$

Banyaknya *sequence* yang di dalamnya memiliki urutan pembelian barang dari X ke Y.

Sebagai contoh, *support* dari *rule* $\{a\} \rightarrow \{e, f\}$ pada tabel 1 adalah 3. *Rule* $\{a\} \rightarrow \{e, f\}$ tersebut terdapat di seq1, seq2 dan seq3. Contoh lain, *support* dari *rule* $\{a\} \rightarrow \{b\}$ pada tabel 1 adalah 2, yaitu pada seq2 dan seq3.

- *Confidence* dari rule $X \rightarrow Y$

Hasil perbandingan antara *support* dari rule $X \rightarrow Y$ dan banyaknya *sequence* yang mengandung barang 'X'. Dapat juga dihitung dengan probabilitas $P(Y|X)$.

Contohnya *confidence* dari rule $\{a\} \rightarrow \{e,f\}$ pada tabel 1 adalah 100%, karena pada *sequence* yang terdapat barang 'a' (seq1, seq2 dan seq3) terdapat rule $\{a\} \rightarrow \{e,f\}$. Contoh lainnya, *confidence* dari rule $\{a\} \rightarrow \{b\}$ pada tabel 1 adalah 66%, karena *support*-nya adalah 2 dan jumlah *sequence* yang memiliki barang 'a' adalah 3. Jadi, $2/3 = 0.66$ atau 66%.

Langkah pertama adalah menentukan semua *item* yang muncul dari semua *sequences database* pada tabel 1.

Tabel 2. Semua *item* yang muncul dari tabel 1.

Item	{a}	{b}	{c}	{d}	{e}	{f}	{g}
------	-----	-----	-----	-----	-----	-----	-----

Langkah kedua, menentukan semua *rule* yang dapat terbentuk antara 2 *item* dari semua item yang ditemukan pada tabel 2.

Tabel 3. Semua *rule* yang terbentuk dari *item* pada tabel 2.

$\{a\} \rightarrow \{b\}$	$\{a\} \rightarrow \{c\}$	$\{a\} \rightarrow \{d\}$	$\{a\} \rightarrow \{e\}$	$\{a\} \rightarrow \{f\}$	$\{a\} \rightarrow \{g\}$
$\{b\} \rightarrow \{a\}$	$\{b\} \rightarrow \{c\}$	$\{b\} \rightarrow \{d\}$	$\{b\} \rightarrow \{e\}$	$\{b\} \rightarrow \{f\}$	$\{b\} \rightarrow \{g\}$
$\{c\} \rightarrow \{a\}$	$\{c\} \rightarrow \{b\}$	$\{c\} \rightarrow \{d\}$	$\{c\} \rightarrow \{e\}$	$\{c\} \rightarrow \{f\}$	$\{c\} \rightarrow \{g\}$
$\{d\} \rightarrow \{a\}$	$\{d\} \rightarrow \{b\}$	$\{d\} \rightarrow \{c\}$	$\{d\} \rightarrow \{e\}$	$\{d\} \rightarrow \{f\}$	$\{d\} \rightarrow \{g\}$
$\{e\} \rightarrow \{a\}$	$\{e\} \rightarrow \{b\}$	$\{e\} \rightarrow \{c\}$	$\{e\} \rightarrow \{d\}$	$\{e\} \rightarrow \{f\}$	$\{e\} \rightarrow \{g\}$
$\{f\} \rightarrow \{a\}$	$\{f\} \rightarrow \{b\}$	$\{f\} \rightarrow \{c\}$	$\{f\} \rightarrow \{d\}$	$\{f\} \rightarrow \{e\}$	$\{f\} \rightarrow \{g\}$
$\{g\} \rightarrow \{a\}$	$\{g\} \rightarrow \{b\}$	$\{g\} \rightarrow \{c\}$	$\{g\} \rightarrow \{d\}$	$\{g\} \rightarrow \{e\}$	$\{g\} \rightarrow \{f\}$

Dari tabel 3, tentukan *rule-rule* yang memiliki *support* lebih besar atau sama dengan *minimum support* = 0.5 ($0.5 * 4 = 2$).

Tabel 4. *Rule* yang memenuhi minsup 0,5 (tidak memenuhi diberi warna merah).

$\{a\} \rightarrow \{b\}$	$\{a\} \rightarrow \{c\}$	$\{a\} \rightarrow \{d\}$	$\{a\} \rightarrow \{e\}$	$\{a\} \rightarrow \{f\}$	$\{a\} \rightarrow \{g\}$
$\{b\} \rightarrow \{a\}$	$\{b\} \rightarrow \{c\}$	$\{b\} \rightarrow \{d\}$	$\{b\} \rightarrow \{e\}$	$\{b\} \rightarrow \{f\}$	$\{b\} \rightarrow \{g\}$
$\{c\} \rightarrow \{a\}$	$\{c\} \rightarrow \{b\}$	$\{c\} \rightarrow \{d\}$	$\{c\} \rightarrow \{e\}$	$\{c\} \rightarrow \{f\}$	$\{c\} \rightarrow \{g\}$
$\{d\} \rightarrow \{a\}$	$\{d\} \rightarrow \{b\}$	$\{d\} \rightarrow \{c\}$	$\{d\} \rightarrow \{e\}$	$\{d\} \rightarrow \{f\}$	$\{d\} \rightarrow \{g\}$
$\{e\} \rightarrow \{a\}$	$\{e\} \rightarrow \{b\}$	$\{e\} \rightarrow \{c\}$	$\{e\} \rightarrow \{d\}$	$\{e\} \rightarrow \{f\}$	$\{e\} \rightarrow \{g\}$
$\{f\} \rightarrow \{a\}$	$\{f\} \rightarrow \{b\}$	$\{f\} \rightarrow \{c\}$	$\{f\} \rightarrow \{d\}$	$\{f\} \rightarrow \{e\}$	$\{f\} \rightarrow \{g\}$
$\{g\} \rightarrow \{a\}$	$\{g\} \rightarrow \{b\}$	$\{g\} \rightarrow \{c\}$	$\{g\} \rightarrow \{d\}$	$\{g\} \rightarrow \{e\}$	$\{g\} \rightarrow \{f\}$

Pada tabel 4, ditemukan 10 *rule* yang memenuhi minsup = 0,5. Dari kesepuluh *rule* tersebut lakukan penambahan satu item ke sisi kiri (fungsi *left expansion*) atau sisi kanan (fungsi *right expansion*) secara rekursif. Jangan lupa untuk melakukan pengecekan support.

Tabel 5. Contoh *rule* hasil *left expansion* dan *right expansion*.

Rule	Left expansion	right expansion
$\{a\} \rightarrow \{b\}$	$\{a, c\} \rightarrow \{b\}$	$\{a\} \rightarrow \{b, c\}$
	$\{a, c, d\} \rightarrow \{b\}$	$\{a, c\} \rightarrow \{b, d\}$
	$\{a, c, d, e\} \rightarrow \{b\}$	$\{a, c, d\} \rightarrow \{b, e\}$
	$\{a, c, d, e, f\} \rightarrow \{b\}$	$\{a, c, d, e\} \rightarrow \{b, f\}$
	$\{a, c, d, e, f, g\} \rightarrow \{b\}$	$\{a, c, d, e, f\} \rightarrow \{b, g\}$
$\{a\} \rightarrow \{c\}$

Hasil akhirnya dapat dilihat pada tabel 6 dengan minsup = 0,5 dan minconf = 60%.

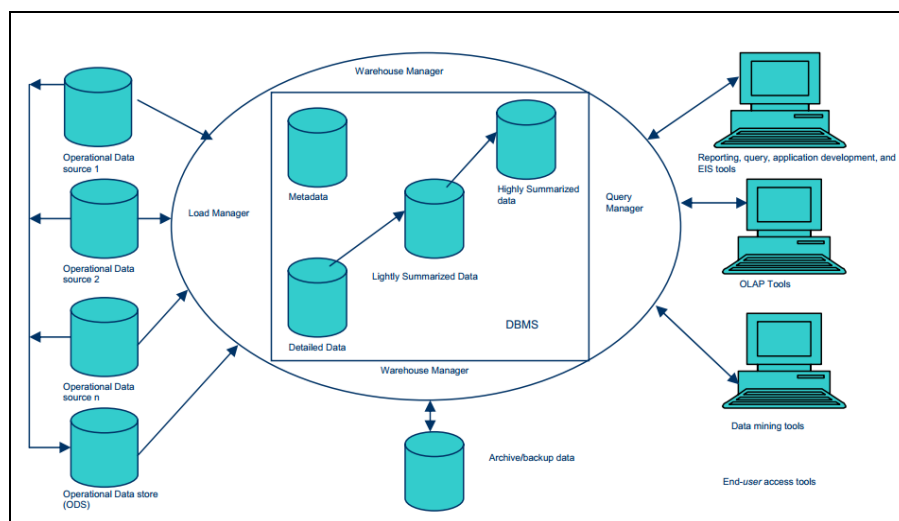
Tabel 6. *Rule* yang memenuhi minsup = 0,5 dan minconf = 60%.

Rule ID	Rule	Support	Confidence
r1	$\{a, b, c\} \rightarrow \{e\}$	2	100 %
r2	$\{a\} \rightarrow \{c, e, f\}$	2	66 %
r3	$\{a, b\} \rightarrow \{e, f\}$	3	100 %
r4	$\{b\} \rightarrow \{e, f\}$	3	75 %
r5	$\{a\} \rightarrow \{e, f\}$	3	100 %
r6	$\{c\} \rightarrow \{e, f\}$	2	100 %
r7	$\{a\} \rightarrow \{b\}$	2	66 %
...

6.5. Data Warehouse

Menurut Inmon (2002), *Data Warehouse* adalah kumpulan data yang telah diringkas dan terintegrasi dari data operasional maupun data *external*, yang memiliki karakteristik *subject-oriented* (berorientasi subjek), *integrated* (terintegrasi), *nonvolatile* (tidak dapat berubah) dan *time variant* (variasi waktu / menyimpan sejarah data) yang berguna dalam pengambilan keputusan. *Data Warehouse* digunakan juga untuk memudahkan proses *report* dan *query data* dari jumlah data yang sangat besar (*Big Data*).

Arsitektur dari *data warehouse* dapat dilihat pada gambar 7.



Gambar 7. Arsitektur Data Warehouse (Inmon, 2002).

Berikut adalah tugas-tugas yang dilakukan oleh *data warehouse* menurut Inmon (2002).

- *Pembuatan Laporan (Reporting)*
Digunakan untuk menghasilkan informasi dari *query-query* yang dijalankan.
- *On-Line Analytical Processing (OLAP)*
OLAP menggunakan konsep data multidimensi dan memungkinkan pemakai untuk menganalisa data sampai mendetail, tanpa mengetikkan satu pun perintah SQL. Hal ini dimungkinkan karena pada konsep data multidimensi, data berupa fakta yang sama bisa dilihat dengan menggunakan dimensi yang berbeda.
- *Data Mining*
- *Proses Informasi Eksekutif*
Digunakan untuk mencapai ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data.

6.6. Business Intelligence

“Business Intelligence adalah suatu cara untuk mengumpulkan, menyimpan, mengorganisasikan, membentuk ulang, meringkas data serta menyediakan informasi, baik berupa data aktifitas bisnis internal maupun eksternal perusahaan, maupun data aktifitas bisnis para pesaing yang mudah diakses serta dianalisis untuk berbagai kegiatan manajemen” (Husni dan Mukhlash , 2004)

Menurut Husni dan Mukhlash (2014), BI dapat digunakan untuk (1) membantu pembuatan keputusan dengan kecepatan dan kualitas yang baik, (2) mempercepat operasional, (3) memperpendek siklus pengembangan produk, (4) memaksimalkan nilai dari produk yang tersedia dan mengantisipasi peluang baru, dan (5) menciptakan pasar yang lebih baik dan terfokus.

6.7. SPMF

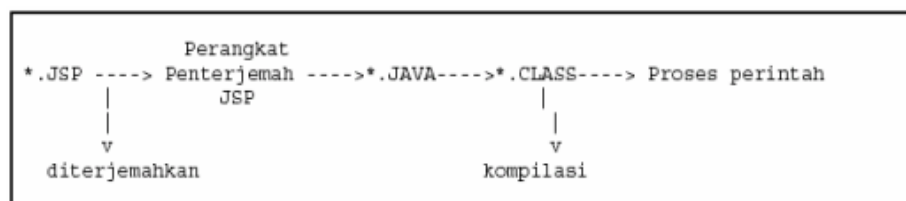
SPMF adalah singkatan dari *Sequential Pattern Mining Framework*. Seperti namanya, SPMF adalah *open-source data mining library* yang mengimplementasikan algoritma-algoritma *mining* lebih dari 55 algoritma. SPMF ini diimplementasikan menggunakan bahasa Java, dan dikhususkan untuk mengenali pola dari transaksi dan sekuensial *database*, seperti algoritma *frequent itemsets*, *association rules*, *sequential pattern*, *sequential rule*, *clustering*, dan sebagainya. Algoritma dari SPMF dapat juga diintegrasikan dengan program java lainnya. (Fournier-Viger, Gomariz, Gueniche, Soltani, Wu dan Tseng, 2014).

6.8. Java Server Pages (JSP)

Java Server Pages (JSP) adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di *Platform* Java, serta merupakan bagian teknologi J2EE (*Java 2 Enterprise Edition*). JSP sangat sesuai dan tangguh untuk menangani presentasi di *website*. (Sunento, n.d.)

JSP memerlukan *Java Virtual Machine* (JVM) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal JVM di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut *Web Container*. Teknologi JSP didesain untuk membuat lebih mudah dan cepat dalam membuat aplikasi web yang bekerja dengan berbagai macam *web server*, *application server*, *browser* dan *development tool*.

JSP merupakan bahasa scripting untuk *web programming* yang bersifat *server side* seperti halnya PHP.

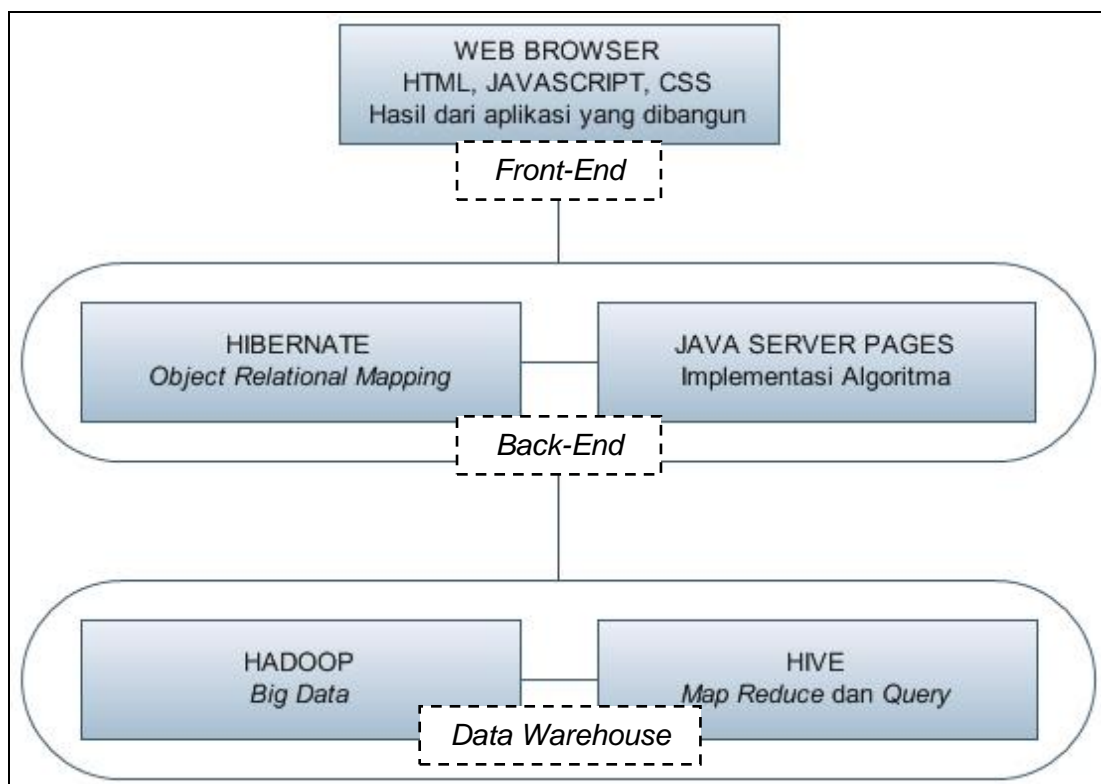


Gambar 8. Proses Kerja JSP. (Sunento, n.d.)

7. Blok Diagram Sistem

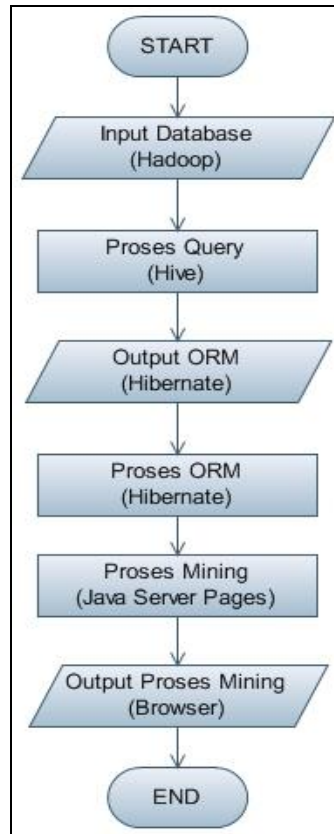
Pada penelitian ini data mentah penjualan barang dan barang akan dikumpulkan terlebih dahulu dari perusahaan pusat perkulakan. Dari data besar tersebut kemudian diolah menggunakan bantuan Hadoop. Setelah itu, proses *Map Reduce* dan *query* data akan ditangani menggunakan Hive. Kemudian dilakukannya *Object Relational Mapping* menggunakan Hibernate. Untuk bagian *back-end* yang mengimplementasikan algoritma *RuleGrowth* akan diolah pada *Java Server Pages* dengan Tomcat Apache sebagai server. Dan hasil algoritma dan analisis dari sistem akan ditampilkan menggunakan *web browser*.

Pada gambar 9 dapat dilihat bagaimana blok diagram sistem yang dibangun dalam penelitian ini.



Gambar 9. Diagram Sistem

Sedangkan untuk langkah-langkah pengerjaan sistme ini dapat dilihat pada *flowchart* pada gambar 10 di bawah ini.



Gambar 10. Flowchart.

8. Metodologi Penelitian

Metode-metode yang digunakan penulis dalam menyelesaikan penelitian ini antara lain :

1. Studi Pustaka

Pada tahap ini, penulis melakukan studi pustaka dengan cara mengumpulkan dan mempelajari teori-teori dan literatur yang dapat membantu penyelesaian penelitian. Khususnya teori-teori yang berhubungan dengan *Data Mining*, *Data Warehouse*, *Sequential Rule Mining*, algoritma *RuleGrowth*, *Business Intelligence* serta *software-software* yang akan dipakai untuk membangun sistem.

2. Pengumpulan Data

Penulis akan mengumpulkan data transaksi penjualan dan data barang kebutuhan pokok sehari-hari dari pusat perkulakan. Data-data yang dikumpulkan akan diolah terlebih dahulu menggunakan Hadoop dan Hive agar diperoleh data bersih yang kemudian dapat digunakan sebagai bahan penelitian.

3. Perancangan dan Pembangunan Sistem

Tahap ini adalah tahap dimana penulis melakukan perancangan mengenai sistem yang akan dibuat, mulai dari *Data Warehouse* yang digunakan sebagai tempat penyimpanan *Big Data* dan rancangan program yang mengimplementasikan algoritma *RuleGrowth*. Kemudian dari perancangan tersebut, penulis membangun sistem yang dapat menampilkan tabel yang berisikan keterkaitan antar barang dari data transaksi penjualan barang, serta memberikan hasil analisis dari keterkaitan antar barang tersebut untuk didapatkan prediksi urutan penjualan barang dari waktu ke waktu.

4. Konsultasi

Tahap ini dilakukan penulis dengan melakukan konsultasi dengan dosen pembimbing agar dapat membantu proses penyelesaian penelitian ini dengan baik dan sistem dapat berjalan sesuai dengan tujuan penelitian.

5. Implementasi dan Testing

Tahap ini ditujukan agar data transaksi penjualan barang dan data barang dapat diuji menggunakan sistem yang sudah dibangun dan sudah mengimplementasikan algoritma *RuleGrowth* pada tahap sebelumnya. Output yang dihasilkan oleh sistem kemudian akan dianalisis dan digunakan sebagai bahan evaluasi sistem agar dapat berjalan menjadi lebih baik.

6. Analisis dan Evaluasi

Pada tahap analisis dan evaluasi, output yang dihasilkan oleh sistem diuji kembali apakah sudah mencapai tujuan dari dilakukannya penelitian ini, dan apakah sudah benar dalam mengimplementasikan algoritma RuleGrowth.

9. Daftar Pustaka

- Fournier-Viger, P., Wu, C., Tseng, V.S., Cao, L., & Nikambou, R. (2015). Mining Partially-Ordered Sequential Rules Common to Multiple Sequences. *IEEE Transactions on Knowledge and Data Engineering*. Retrieved from <http://dx.doi.org/10.1109/TKDE.2015.2405509>
- Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C., & Tseng, V.S. (2014). SPMF : A Java Open-Source Pattern Mining Library. *Journal Machine Learning Research*, 15, 3569-3573.
- Fournie-Viger, P., Nikambou, R., & Tseng, V.S. (2011). RuleGrowth: Mining Sequential Rules Common to Several Sequences by Pattern-Growth. *Proceedings of the 26th Symposium on Applied Computing (ACM SAC 2011)*. ACM Press, pp. 956-961.
- Husni, Z.N., & Mukhlash, I. (2014). Implementasi Business Intelligence Pada Manajemen Report Bank XYZ. *Journal Sains dan Senni Pomits*, 3(2), A16-A21.
- Inmon, W. (2002). *Building the Data Warehouse*. Canada: Wiley & Sons.
- Juliastio, R., & Gunawan. (2015). *Sequential Pattern Mining Dengan Spade Untuk Prediksi Pembelian Spare Part Dan Aksesoris Komputer Pada Kedatangan Kembali Konsumen*. Seminar Nasional “Inovasi dalam Desain dan Teknologi”. IDEaTech.
- Sunento. (n.d.). *Module 1 : Pengenalan Dasar Tentang JSP*. Retrieved from <http://kambing.ui.ac.id/onnopurbo/library/library-ref-ind/ref-ind-2/application/programming/Module%201%20-%20Pengenalan%20Dasar%20JSP.pdf>
- Yulianton, H. (Januari 2008). Data Mining untuk Dunia Bisnis. *Jurnal Teknologi Informasi DINAMIK*, 13(1), 9-15.