

SISTEM REKOMENDASI KATALOG PRODUK ONLINE
MENGGUNAKAN METODE ASSOCIATION RULE DENGAN
ALGORITMA FOLD-GROWTH

SKRIPSI

oleh:
HENDY KUSDARMANTO
0210960029-96



PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009

UNIVERSITAS BRAWIJAYA



**SISTEM REKOMENDASI KATALOG PRODUK ONLINE
MENGGUNAKAN METODE ASSOCIATION RULE DENGAN
ALGORITMA FOLD-GROWTH**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer dalam bidang Ilmu Komputer

oleh:

HENDY KUSDARMANTO
0210960029-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

SISTEM REKOMENDASI KATALOG PRODUK ONLINE MENGGUNAKAN METODE ASSOCIATION RULE DENGAN ALGORITMA FOLD-GROWTH

oleh:

HENDY KUSDARMANTO
0210960029-96

Setelah dipertahankan di depan Majelis Pengaji
pada tanggal 14 Agustus 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I

Pembimbing II

Drs. A. Ridok, M.Kom
NIP. 132 090 392

Dian Eka R., S.Si, M.Kom
NIP. 132 300 224

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Drs. Agus Suryanto, M.Sc
NIP. 132 126 049

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama

: Hendy Kusdarmanto

NIM

: 0210960029-96

Jurusan

: Matematika

Program Studi

: Ilmu Komputer

Penulis skripsi berjudul

: Sistem Rekomendasi Katalog produk Online Menggunakan Metode *Association Rule* dengan Algoritma *Fold-Growth*

Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.
3. Kode program dalam skripsi ini dibuatkan oleh, Nama: Arief Rachmansyah, NIM: 0210960011.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 14 Agustus 2009

Yang menyatakan,

Hendy Kusdarmanto

NIM. 0210960029

UNIVERSITAS BRAWIJAYA



SISTEM REKOMENDASI KATALOG PRODUK ONLINE MENGGUNAKAN METODE ASSOCIATION RULE DENGAN ALGORITMA *FOLD-GROWTH*

ABSTRAK

Informasi yang cepat dan tepat sangat berperan penting pada perkembangan pada dunia bisnis. Bagi produsen, mengetahui kebutuhan konsumen akan sebuah produk merupakan hal yang penting. Begitu juga dengan konsumen, mengetahui informasi sebuah produk yang sesuai dengan kebutuhannya merupakan hal yang penting juga. Sistem rekomendasi merupakan sebuah terobosan yang menarik untuk menjembatani akan sebuah informasi yang bermanfaat baik bagi produsen maupun konsumen. Skripsi ini dibuat sistem rekomendasi yang didasarkan pada data akses *user* yang terdapat pada server atau biasa disebut dengan *data access log*. Metode yang digunakan adalah *association rule* dengan algoritma *fold-growth*. Dengan algoritma ini akan dicari set *item* (*itemset*) dari produk – produk yang sering diakses oleh konsumen yang dapat dilihat melalui *data access log*. Studi kasus skripsi ini dilakukan pada katalog produk online dan dengan menggunakan data percobaan yang ditempatkan pada *data access log*.

UNIVERSITAS BRAWIJAYA



RECOMMENDATION SYSTEM PRODUCT ONLINE CATALOG USING ASSOCIATION RULE METHODE WITH FOLD GROWTH ALGORITHM

ABSTRACT

A fast and accurate information has a significant role to business aspect. To the producer, knowing what consumer need for a certain product is a significant thing. So as the consumer, knowing the infomation of a product that match their need is an important thing too. Recommendation system is an interesting breakthrough for connecting usefull information to the producer and so to the consumer too. This observation are created based on user data access on a server or data access log. Methode that being used are association rule with fold-growth algorithm. This algorithm will looked for an itemset from product-product that oftenly accessed by a user which is seen through the data access log. Observation are being used on product online catalog and using experiment data which is placed on data access log.

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Puji syukur penyusun panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul "**Sistem Rekomendasi Katalog produk Online Menggunakan Metode Association Rule dengan Algoritma Fold-Growth**".

Skripsi ini diajukan sebagai syarat untuk memperoleh gelar Sarjana Komputer di Fakultas MIPA, Jurusan Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaiannya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Drs. Achmad Ridok, M.Kom, selaku Dosen Penasehat Akademik sekaligus Dosen Pembimbing I penulisan skripsi.
2. Dian Eka Ratnawati, S.Si, M.Kom, selaku Dosen Pembimbing II penulisan skripsi.
3. Wayan Firdaus Mahmudy, S.Si., MT, selaku Ketua Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
4. Dr. Agus Suryanto, M.Sc, selaku Ketua Jurusan Matematika FMIPA Universitas Brawijaya.
5. Arief Rachmansyah, selaku teman ilmu komputer 2002 yang telah membantu dalam membuat kode program dengan PHP.
6. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
7. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Orang tua dan keluarga Penulis atas segala dukungan materi dan doa restunya kepada Penulis.

9. Rekan-rekan Ilmu Komputer 2002 yang telah memberikan dukungannya dalam penyusunan skripsi ini.
10. Rekan-rekan Unit Aktivitas Universitas Brawijaya khususnya Korps Sukarela yang telah memberikan dukungannya dalam penyusunan skripsi ini.
11. Semua pihak yang telah membantu terselesaikannya laporan ini yang tidak dapat kami sebutkan satu per satu.

Penulis menyadari bahwa laporan ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat Penulis harapkan dari berbagai pihak guna peningkatan kualitas penelitian serupa di masa mendatang.

Malang, 14 Agustus 2009

Penulis

DAFTAR ISI

Halaman

KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii

BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
 BAB II DASAR TEORI.....	 5
2.1 Data Mining.....	5
2.1.1 <i>Association Rule</i>	6
2.2 Array	7
2.2.1 Array dalam PHP	7
2.3 <i>Web Mining</i>	8
2.3.1 <i>Web Usage Mining</i>	10
2.4 Fold-Growth.....	11
2.4.1 FODARM	12
2.4.2 FP-Growth	13
 BAB III METODE & PERANCANGAN SISTEM.....	 15
3.1 Preprocessing dan Transformasi Data	16
3.1.1 Struktur Website.....	16
3.1.2 Sorting Hyperlink	17
3.1.3 Pemrosesan <i>Access log</i>	17
3.1.4 Pengelompokan Data <i>Log</i>	19
3.1.5 Pembuatan <i>Session</i>	20
3.2 <i>Data Mining</i>	21
3.2.1 Penggalian L_1 dan L_2 dengan menggunakan SOTrieIT ..	21
3.2.2 Tahap Pemangkasan <i>Item-item</i> yang Tidak <i>Frequent</i> ..	23
3.2.3 Tahap Pembangunan FP-Tree.....	24

3.2.4 Tahap Penggalian <i>Itemset Frequent</i>	25
3.2 Perancangan tabel	31
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....	33
4.1 Implementasi	33
4.1.1 Persiapan Data.....	33
4.1.2 <i>Parsing Data Access Log</i>	35
4.1.3 Pembuatan <i>Session</i>	36
4.1.4 Pembentukan <i>Itemset</i>	40
4.2 Analisa Hasil	47
BAB V PENUTUP	51
5.1 Kesimpulan	51
5.1 Saran	51
DAFTAR PUSTAKA	53

DAFTAR TABEL

Halaman

Tabel 2.1 Array function	8
Tabel 2.2 Bentuk umum data <i>log</i>	9
Tabel 3.1 Data hyperlink pada database.....	17
Tabel 3.2 Contoh data log CLF	18
Tabel 3.3 Contoh data log ECLF.....	18
Tabel 3.4 Isi dan arti Elemen Extended Common Log Format.....	19
Tabel 3.5 <i>Transformation Data access log</i>	19
Tabel 3.6 Contoh data hyperlink pada session	21
Tabel 3.7 Data hyperlink setelah pembersihan duplikasi	22
Tabel 3.8 Data pembentukan itemset L_1	22
Tabel 3.9 Data pembentukan itemset L_2	23
Tabel 3.10 Data pembentukan itemset L_2 dalam bentuk matrix	23
Tabel 3.11 Data transaksi setelah pemangkasan L_1 dan L_2	24
Tabel 3.12 Data transaksi setelah dipotong, dgn min sup=60%	24
Tabel 3.13 Hasil Frequent Itemset, dgn min sup=60%	30
Tabel 3.14 Hasil Frequent Itemset yang telah diurutkan.....	30
Tabel 3.15 Hasil rekomendasi yang didapat.....	31
Tabel 3.16 Atribut tabel hyperlink	31
Tabel 3.17 Atribut tabel k_hs	31

UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

Halaman

Gambar 2.1 Tahapan Proses KDD	6
Gambar 2.2 Taksonomi <i>web mining</i>	10
Gambar 2.3 Algoritma Fold-Growth.....	12
Gambar 2.4 Pohon SOTrieIT	13
Gambar 2.5 Transaksi dengan min sup 3	14
Gambar 2.6 Pohon FP-Tree dengan transaksi yang telah dipotong .	14
Gambar 3.1 Blok diagram preprocessing dan transformation data ..	15
Gambar 3.2 Blok diagram <i>data mining</i>	16
Gambar 3.3 Struktur website katalog produk.....	17
Gambar 3.4 FP-Tree setelah pemangkasan	25
Gambar 3.5 <i>Conditional FP-Tree for H6</i>	26
Gambar 3.6 <i>Conditional FP-Tree for (H3,H6)</i>	26
Gambar 3.7 <i>Conditional FP-Tree for H4</i>	27
Gambar 3.8 <i>Conditional FP-Tree for H5</i>	29
Gambar 4.1 Interface <i>Parsing hyperlink</i>	34
Gambar 4.2 Interface hasil <i>parsing hyperlink</i>	34
Gambar 4.3 Input data access log.....	38
Gambar 4.4 Hasil proses <i>parsing data log</i> dan <i>session</i>	39
Gambar 4.5 Grafik kemunculan <i>itemset</i>	49

UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan adanya katalog produk *online* pada dunia usaha diharapkan dapat mengantarkan sebuah terobosan baru dalam perdagangan global. Bagi produsen, katalog produk *online* dapat dijadikan sarana memperkenalkan produk baru kepada konsumen di seluruh penjuru dunia yang terhubung dengan internet. Dengan adanya katalog produk *online*, pencarian terhadap suatu barang/produk dapat dilakukan secara cepat. Bisa dikatakan bahwa katalog produk *online* dapat menjadi sarana komunikasi antara produsen dan konsumen.

Bagi sebuah katalog produk *online*, dibutuhkan sebuah sistem yang mampu beradaptasi dengan masing-masing konsumen dan dapat memberi saran atau sugesti kepada konsumen tentang sebuah produk yang cocok. Dan juga dapat dijadikan pedoman bagi produsen akan produk – produk apa yang banyak diminati oleh konsumen. Dalam kajian *web mining*, sistem personalisasi web semacam ini disebut dengan *recommender system*.

Recommender system (RSs) bekerja untuk membantu *user* dalam memilih item yang akan dibeli dengan cara memberi informasi atau sugesti berdasarkan tingkat ketertarikan atau selera masing-masing *user* (Barneveld, 2003). Ada beberapa hal untuk mengetahui kecocokan antara *user* dan produk, misalnya data dari masing – masing *user* dan produk itu sendiri, cara akses *user* terhadap halaman web, dari rating yang diberikan oleh *user* kepada produk, ataupun dari data transaksi pembelian produk pada waktu lampau. Dengan adanya *recommender system* diharapkan *user* akan menemukan produk yang cocok atau produk yang sedang dicari/dibutuhkan, bahkan dapat memberikan saran terhadap produk lain yang kemungkinan akan dibutuhkan oleh *user*.

Ada banyak metode yang dapat digunakan dalam RSs, yaitu *statistical approach*, *collaborative filtering*, *association rules*, *content based*, dan *demographic recommendation* (P. Kazienko, 2006). Dari beberapa metode yang ada tersebut, yang cocok digunakan dalam menemukan hubungan antar produk dalam sebuah katalog *online* berdasarkan tingkah laku konsumen adalah

association rule. *Association rules* merupakan suatu prosedur untuk menemukan aturan asosiatif antara suatu kombinasi item dalam suatu dataset yang ditentukan (Weiyang, 2000). Karena awalnya berasal dari studi tentang database transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan *market basket analysis* (Santosa, 2007). Berbagai penelitian berbasis *association rules* telah banyak dilakukan dengan berbagai teknik. Salah satunya adalah algoritma *FOLDARM* yang menggunakan struktur data *SOTrieIT* yang memiliki kinerja cepat pada saat ukuran *itemset frequent* maksimum (k_{max}) adalah kecil atau $k_{max} \leq 10$. Dan ada algoritma *FP-Growth* yang dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree* yang memiliki kinerja cepat pada saat $k_{max} > 10$ (R. Soelaiman, 2006).

Dalam penelitian ini, metode asosiasi (*association rule*) yang digunakan adalah algoritma *Fold-Growth*. Algoritma *Fold-Growth* merupakan hasil gabungan dari algoritma *FOLDARM* dan *FP-growth* dalam proses penggalian *itemset* yang *frequent*. Menurut penelitian yang telah dilakukan sebelumnya, durasi eksekusi, skalabilitas, reabilitas, dan utilisasi memori *Fold-Growth* lebih baik daripada *FP-Growth* (R. Soelaiman, 2006). Algoritma *Fold-Growth* cocok digunakan pada katalog produk online, baik untuk jumlah produk yang sedikit seperti di elektronik commerce dan baik juga untuk jumlah produk yang banyak seperti di amazon atau ebay. Algoritma *Fold-Growth* pada penelitian ini akan dibagi dalam empat tahapan utama yaitu, tahapan penggalian *1-itemset frequent* dan *2-itemset frequent*, tahap pemangkasan item-item yang tidak *frequent*, membangun *FP-Tree*, dan tahapan penggalian semua *itemset frequent*. Dengan adanya empat tahapan tersebut, diharapkan *itemset frequent* yang terbentuk sesuai dengan tingkah laku konsumen dalam mengakses katalog produk *online*, sehingga akan diketahui item-item apa saja yang sering diakses bersamaan dengan item yang sedang diakses.

1.2 Rumusan Masalah

Rumusan masalah yang dapat diambil adalah bagaimana menerapkan metode assosiasi (*association rule*) dengan algoritma *Fold-Growth* pada sistem rekomendasi katalog produk *online*.

1.3 Batasan Masalah

Sistem mengenali konsumen melalui nomor *IP*. Sehingga diperlukan pembatasan masalah pada nomor *IP* yang dipakai bersama seperti pada *hotspot* maupun warnet yang hanya akan dihitung satu nomor *IP* saja pada tiap session dan digunakan rentang waktu per 30 menit untuk setiap sessionnya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sistem rekomendasi dengan menggunakan metode assosiasi (*association rule*) dengan algoritma *Fold-Growth* pada katalog produk *online*.

1.5 Manfaat Penelitian

Manfaat yang didapat dari hasil skripsi ini adalah untuk memberikan rekomendasi kepada konsumen terhadap produk - produk yang kemungkinan dibutuhkan ketika mencari suatu produk.

1.6 Sistematika Penulisan

Penulisan skripsi ini dibagi dalam lima bab, yaitu :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi pemecahan masalah, dan sistematika penulisan.

BAB II : DASAR TEORI

Bab ini berisi tentang teori-teori yang menjadi acuan untuk pelaksanaan penulisan skripsi yang meliputi teori tentang *data mining*, *association rule*, dan juga tentang algoritma *Fold-Growth*.

BAB III : PERANCANGAN DAN METODOLOGI

Bab ini berisi tentang aliran proses atau alur dari sistem yang akan dibuat serta algoritma dari program serta pembahasannya.

BAB IV : ANALISIS DAN PEMBAHASAN

Bab ini membahas tentang analisis dan pembahasan tentang kinerja dari program yang telah dibuat baik dari antar muka maupun proses.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari sistem yang dirancang serta saran dari hasil skripsi ini untuk pengembangan selanjutnya.



BAB II

DASAR TEORI

2.1 Data Mining

Penggalian atau penambangan data (Bahasa Inggris: *data mining*, DM) adalah proses pencarian otomatis terhadap pola dalam data dalam jumlah besar dengan menggunakan perangkat seperti klasifikasi, penggugusan (*clustering*), dan *association rule*. Penambangan data adalah suatu topik yang kompleks dan berpautan dengan berbagai bidang inti seperti ilmu komputer dan memberikan nilai tambah dari teknik komputasi lain seperti statistika, pengambilan informasi, pembelajaran mesin, dan pengenalan pola. *Data mining* diterapkan dengan paradigma untuk melihat informasi yang tersembunyi.

Terdapat beberapa pengertian tentang *data mining* sebagai berikut:

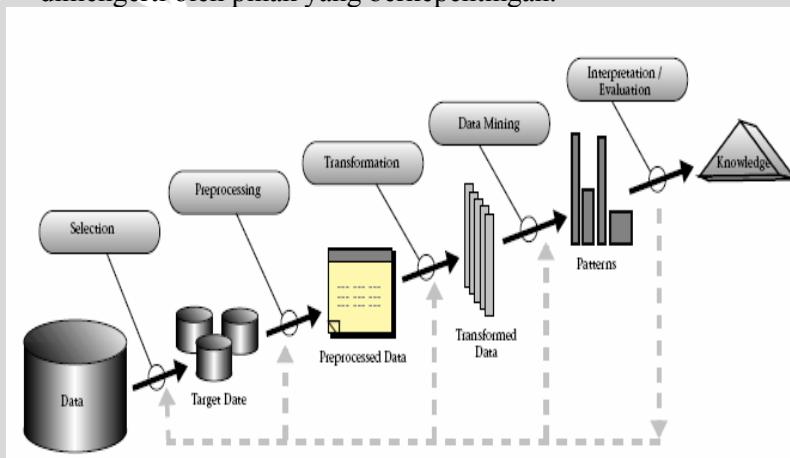
1. *Data mining* adalah suatu proses dalam mencari korelasi, pola, atau tren yang bermanfaat dari sebuah data yang berukuran besar dengan menggunakan teknik statistika ataupun matematika (Larose, 2005).
2. *Data mining* adalah salah satu teknik dalam ilmu komputer yang digunakan untuk menggali dan mengambil suatu informasi pada banyak data (Rennolls, 2004).
3. *Data mining* adalah mencocokkan data dalam suatu model untuk menemukan informasi yang tersembunyi dalam basis data (J.Kamber, 2001).

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, akan tetapi berkaitan satu sama lain, dimana salah satu tahapan dalam keseluruhan proses KDD adalah *data mining* (Masruri, 2006).

Secara garis besar beberapa tahap proses KDD adalah sebagai berikut (J.Kamber. 2001):

1. *Data selection* (pemilihan data), pemilihan data relevan yang didapat dari basis data.

2. *Preprocessing/data cleaning* (pembersihan data), proses menghilangkan *noise* dan data yang tidak konsisten atau data tidak relevan. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang tidak konsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak.
3. *Transformation* (transformasi data), data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*.
4. *Data mining*, suatu proses di mana metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.
5. *Interpretation/ Evaluation*, pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.



Gambar 2.1 Tahapan proses KDD

Sumber: (J.Kamber, 2001)

2.1.1 Association Rule

Association rule adalah teknik *data mining* untuk menemukan aturan hubungan *assosiatif* antara suatu kombinasi item dengan tujuan akhir mendapatkan kombinasi item yang dikatakan menarik (*interesting association rules*). Penting tidaknya suatu aturan *assosiatif* dapat diketahui dengan dua parameter (F. Perdana, 2008), yaitu:

1. *Support*, yaitu prosentase kombinasi atribut dalam *basisdata*.

2. *Confidence*, yaitu kuatnya hubungan antar atribut dalam aturan asosiasi.

Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rules*, yaitu nilai suatu *item* dengan melihat peluang munculnya *item* berdasarkan *support* dari *item* tersebut dan kemudian dibandingkan dengan batasan (*threshold*) yang ditentukan oleh *user*. Batasan tersebut umumnya terdiri dari *minimum support* dan *minimum confidence* (F. Perdana, 2008)..

Contoh penerapan teknik ini dapat dilakukan pada swalayan. Pada swalayan dapat dilakukan penempatan barang secara berdekatan untuk barang yang memiliki kecenderungan pembelian secara bersamaan. Misalnya seseorang akan memiliki kecenderungan untuk membeli roti dan susu secara bersamaan. Contoh lainnya adalah pembelian pasta dan sikat gigi. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang strategi pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu (Masruri, 2006). Metode ini disebut dengan *market basket analysis*.

Market basket analysis adalah teknik *data mining* dalam mengidentifikasi asosiasi antar item dalam sebuah *shopping basket* atau catatan transaksi penjualan. Proses analisis ini banyak dimanfaatkan untuk mengetahui item-item mana saja yang sering dibeli secara bersamaan sehingga hasil analisis yang didapat dapat digunakan untuk mengembangkan strategi pemasaran melalui promosi (Masruri, 2006).

2.2 Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun *array* disebut *elemen array*, yang masing-masing elemen dapat diakses tersendiri melalui indeks *array*. Indeks *array* dimulai dari 0. Jadi indeks *array* 0 menyatakan elemen pertama dari *array*, indeks *array* 1 menyatakan elemen *array* kedua, dan seterusnya (Anonymous, 2006)

2.2.1 Array dalam PHP

Pada bahasa pemrograman PHP (*Personal Home Page*) sudah banyak disediakan fungsi *array* (*array function*) dan langsung dapat digunakan untuk melakukan operasi data hanya dengan memanggil fungsi tersebut. Dalam skripsi ini, tidak seluruh *array function* yang disediakan oleh PHP digunakan, adapun beberapa *array function* yang nantinya digunakan ditunjukkan pada Tabel 2.1.

Tabel 2.1 Array function

Sumber: (Mehdi, 2006)

1. <code>\$arra1=Array()</code>	Deklarasi awal pembuatan <i>array</i> kosong
2. <code>count (\$arra1)</code>	Menghitung jumlah elemen dalam <i>array</i>
3. <code>print_r (\$arra1)</code>	Mencetak hasil <i>array</i>
4. <code>array_unique (\$arra1)</code>	Menghilangkan duplikasi data/element pada suatu <i>array</i>
5. <code>array_push (\$arra1, elemen)</code>	Memasukkan suatu elemen ke dalam <i>array</i>
6. <code>in_array (elemen, \$arra1)</code>	Memeriksa apakah suatu elemen terdapat dalam sebuah <i>array</i>
7. <code>array_search (elemen, \$arra1)</code>	Mencari sebuah elemen dalam <i>array</i>
8. <code>array_merge (\$arra1, \$arra12)</code>	Menggabungkan elemen pada masing-masing <i>array</i> ; <code>\$arra12=array()</code>
9. <code>explode ("string pemisah", elemen)</code>	Untuk memecah elemen dalam <i>array</i> berdasarkan <i>string</i> tertentu
10. <code>mysql_fetch_array</code>	Digunakan untuk manipulasi data dalam memanggil data pada <i>database</i>

2.3 Web Mining

Pendekatan teknik-teknik dalam *data mining* dapat juga dilakukan pada *website*, dan disebut dengan *web mining*. Dengan kata lain *web mining* adalah suatu teknik *data mining* yang

diimplementasikan pada *website*. Data *web* itu sendiri adalah data yang berkaitan dengan *web*, mulai dari teks, halaman HTML, data *multimedia*, data transaksi, hingga data *web access log*, yaitu data yang mencatat dan menunjukkan *user session* (aktivitas *user*) pada *website* (Peter I, 2006). Terdapat perbedaan makna session pada bahasan *web content* dan *web usage*. Pada *web content*, *session* biasanya digunakan untuk mengetahui *user* yang sedang mengunjungi sebuah *website* pada saat *login* atau *online*. Sedangkan pada *web usage*, *session* merupakan pembagian satuan transaksi yang terdapat pada *server* ke dalam satuan waktu tertentu.

Bentuk umum *data access log* yang disimpan oleh *web servers* adalah *remotehost*, *rfc931*, *authuser*, *date*, *request*, *status*, *bytes*, *referrer*, *user_agent*.

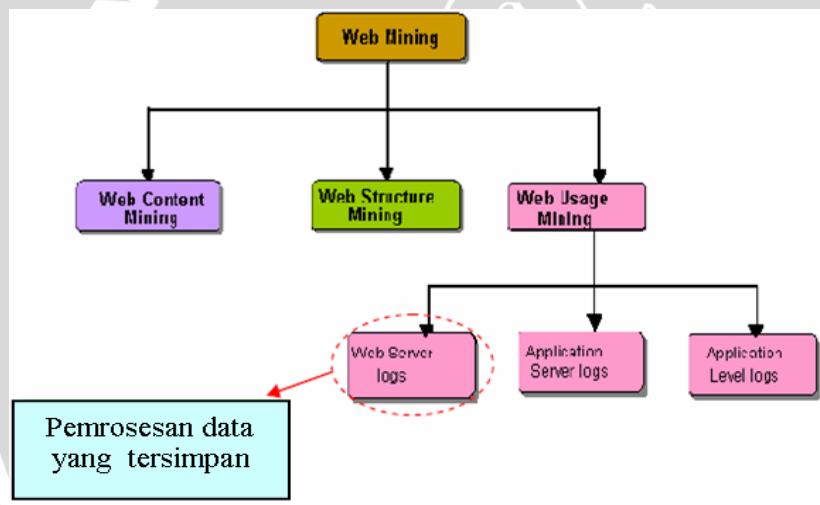
Tabel 2.2 Bentuk umum data *log*

Sumber: (Peter I, 2006)

Bentuk umum fields untuk data access log yang disimpan oleh web server	
Nama field	Deskripsi field disertai contoh
<i>remotehost</i>	Remote hostname (atau IP number jika tidak terdapat DNS hostname) Contoh: 127.0.0.1
<i>rfc931</i>	remote login untuk sebuah <i>user name</i> . Contoh: -
<i>authuser</i>	<i>username</i> yang melakukan otentifikasi terhadap dirinya Contoh: -
<i>date</i>	Waktu (tanggal dan jam) request berdasarkan zona waktu <i>web server's</i> Contoh: [29/May/2007:14:50:04 +0700]
<i>request</i>	Untuk data request, adalah data yang sebenarnya diperoleh dari aktivitas <i>user</i> . Data request ini dibagi menjadi tiga bagian, yaitu: method, alamat data yang diminta, dan jenis protocol yang digunakan. Contoh: "GET / HTTP/1.1"
<i>status</i>	HTTP status code yang dikembalikan ke <i>user</i> . Contoh: 200

<i>bytes</i>	Besarnya dokumen yang ditransfer Contoh: 12079
<i>referrer</i>	URL yang dikunjungi <i>user</i> sebelum adanya permintaan terhadap URL yang baru Contoh: “-”
<i>user_agent</i>	Software yang digunakan oleh <i>user</i> Contoh: "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"

Pada bahasan *web mining*, secara umum *web mining* dikategorikan ke dalam 3 bahasan, yaitu *web content mining*, *web structure mining*, dan *web usage mining*, seperti ditunjukkan Gambar 2.2.



Gambar 2.2 Taksonomi *web mining*
Sumber: (Desikan, 2003)

2.3.1 Web Usage Mining

Web usage mining mempelajari tentang *user navigation pattern* (pola *user*) dalam menelusuri halaman-halaman suatu website (Desikan, 2003). Hal ini telah meningkatkan kebutuhan *admin* atau pemilik website akan pemahaman terhadap kebutuhan

user dan perilaku mereka dalam menelusuri halaman *website*. Dan saat ini *web usage mining* telah menjadi salah satu topik bahasan yang memiliki perhatian lebih dalam mempelajari kebiasaan *user* tersebut (Desikan, 2003).

Pada tingkat selanjutnya, *web usage* dibagi menjadi tiga bagian tergantung pada data yang diproses, yaitu:

1. *Web server logs*

web server logs berhubungan dengan data *access logs* yang ada pada *web server*. Data yang disediakan oleh *web server* berisi informasi tentang *remotehost (IP address)*, *rfc931*, *authuser*, *timestamp*, *request*, *status*, dan *size* serta jenis agen yang digunakan untuk *mengakses website*.

2. *Application server log*

Data yang dihasilkan secara dinamis oleh berbagai macam *aplikasi server*. Seperti file *.asp dan *.php, terdapat aplikasi-aplikasi tertentu yang hanya bisa dijalankan dengan kedua bahasa ini, seperti penggunaan *database* untuk informasi yang dihasilkan dan dikumpulkan berdasar perilaku seorang *user* terhadap aplikasi yang ada.

3. *Application level logs*

Sebuah data yang disediakan oleh *user* untuk sebuah aplikasi, seperti data demografi. Data semacam ini dapat dikumpulkan dari setiap *user* atau *event* tertentu dan data ini dapat digunakan sebagai informasi yang berguna untuk suatu kebutuhan di masa mendatang.

2.4 Fold-Growth

Algoritma Fold-*Growth* merupakan hasil gabungan dari algoritma FOLDARM dan FP-*Growth*. Pada algoritma Fold-*Growth* ini, diharapkan dapat menggabungkan keuntungan dari algoritma FOLDARM yang memiliki kinerja cepat pada saat ukuran *itemset frequent* maksimum (k_{max}) adalah kecil atau $k_{max} \leq 10$ dengan keuntungan dari algoritma FP-*Growth* yang memiliki kinerja yang cepat pada saat $k_{max} > 10$. Algoritma Fold-*Growth* ditunjukkan pada gambar 2.3. (R. Soelaiman, 2006)

```

1: Menggunakan SOTrie untuk
   Menemukan  $L_1$  dan  $L_2$  dengan cepat
2: if  $L_1 = 0$  v  $L_2 = 0$  then
3:   algoritma dihentikan
4: end if
5: for transaction  $T \in D$  do
6:   Hilangkan item yang tidak
      Memberikan kontribusi pada  $L_k$ 
      dimana  $k > 2$ , menggunakan
       $L_1$  dan  $L_2$ 
7: urutkan item berdasarkan
   support terbesar
8: Bangun/Ubah FP-Tree dengan  $T$ 
   yang telah dipangkas dan diurutkan
9: end for
10: Jalankan algoritma FP-Growth
   Pada FP-Tree yang dibangun

```

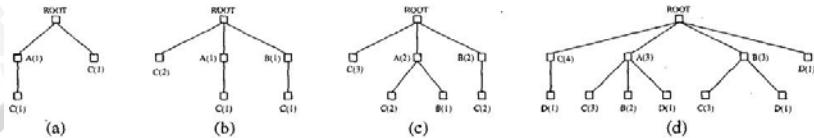
Gambar 2.3 Algoritma Fold-Growth

Sumber: (R. Soelaiman, 2006)

2.4.1 FODARM

FODARM (*Fast Online Dynamic Association Rule Mining*) merupakan algoritma yang dapat melakukan penambangan data secara cepat. Dalam melakukan penambangan data, FODARM menggunakan struktur data SOTrieIT. Struktur data SOTrieIT digunakan pada tahap penggalian L_1 dan L_2 . SOTrieIT memiliki dua tingkatan dari node-node *tree* yang menyatakan bahwa tiap node w memiliki sebuah label l yang dinyatakan sebagai *item* dan sebuah notasi j yang menyimpan nilai support count yang berhubungan. Setiap node pohon terhubung pada beberapa *item* yang terdapat dalam *itemset* I (dinotasikan dengan $a_i \in I$) maka, untuk w_i mengacu pada node yang memiliki hubungan dengan $a_i \in I$. Himpunan SOTrieIT dimungkinkan memiliki *parent node* yang berbeda-beda seperti w_1 , w_2 ,..., , w_N , yang dibangun dari sebuah basis data untuk menyimpan *support count* dari semua 1-*itemset* dan

2-itemset. Maka, digunakan node khusus yang dinamakan ROOT untuk menghubungkan semua SOTrie bersama-sama. (R. Soelaiman, 2006). Gambar 2.4 menunjukkan pembentukan tree dari SOTrieIT.



Gambar 2.4 Pohon SOTrieIT

sumber: (Y. Won, 2001)

2.4.2 FP-Growth

Salah satu algoritma yang tercepat dan terpopuler saat ini dalam penambangan *frequent itemset* adalah FP-Growth (J.Han, 2000). Algoritma FP-Growth menggunakan struktur pohon untuk mengelola database transaksi (yang disebut FP-Tree) yang dapat menyimpan sejumlah memori yang cukup dalam melakukan pemrosesan transaksi. FP-Tree merupakan struktur penyimpanan data yang dimampatkan. FP-Tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-Tree. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki *item* yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki *item* yang sama, maka proses pemampatan dengan struktur data FP-Tree semakin efektif. Kelebihan dari FP-Tree adalah hanya memerlukan dua kali pemindai data transaksi yang terbukti sangat efisien. Misal $I = \{a_1, a_2, \dots, a_n\}$ adalah kumpulan dari *item*. Dan basis data transaksi $DB = \{T_1, T_2, \dots, T_n\}$, di mana T_i ($i \in [1..n]$) adalah sekumpulan transaksi yang mengandung *item* di I . Sedangkan support adalah penghitung (*counter*) frekuensi kemunculan transaksi yang mengandung suatu pola. Suatu pola dikatakan sering muncul (*frequent pattern*) apabila support dari pola tersebut tidak kurang dari suatu konstanta ξ (batas ambang minimum support) yang telah didefinisikan sebelumnya. Permasalahan mencari pola *frequent* dengan batas ambang minimum support count ξ inilah yang dicoba untuk dipecahkan oleh FP-Growth dengan bantuan Struktur FP-Tree.

(R. Soelaiman, 2006). Contoh transaksi dan *tree* dari FP-*Tree* ditunjukkan pada gambar 2.5 dan gambar 2.6.

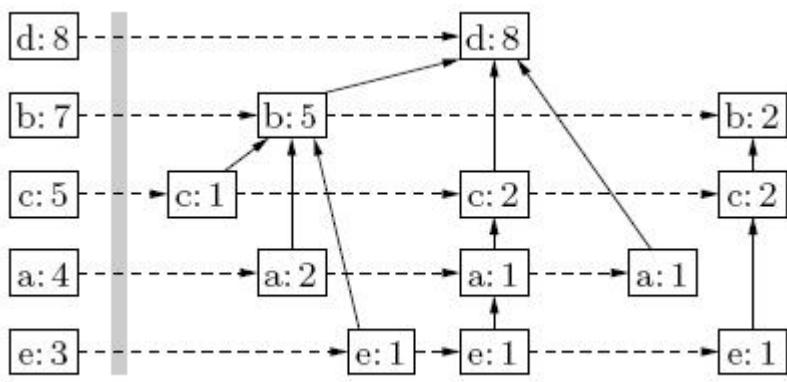
a d f
a c d e
b d
b c d
b c
a b d
b d e
b c e g
c d f
a b d

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

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

Gambar 2.5 Transaksi dengan min sup 3

Sumber : (Borgelt, 2005)



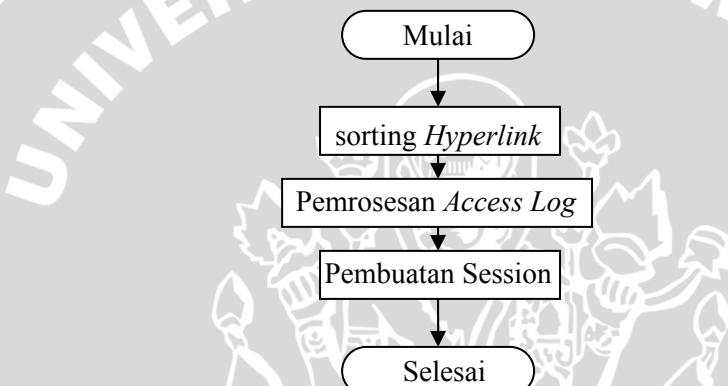
Gambar 2.6 Pohon FP-Tree dengan transaksi yang telah dipotong

Sumber : (Borgelt, 2005)

BAB III

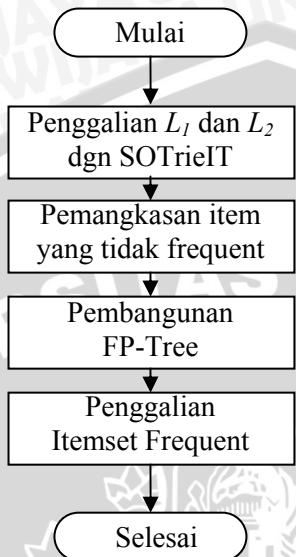
METODE & PERANCANGAN SISTEM

Pada sistem rekomendasi ini akan dibagi menjadi beberapa tahap, mulai dari tahap *preprocessing* dan *transformation data*, sampai tahap *data mining*. Pada tahap *preprocessing* dan *transformation data*, data yang digunakan adalah data produk/item yang berdasarkan *hyperlink pages*, dan data transaksi yang berdasarkan *session* pada data *access log*. Lihat pada gambar 3.1.



Gambar 3.1 Blok diagram *preprocessing* dan *transformation data*

Setelah pembuatan *session* dilakukan, langkah berikutnya adalah tahap *data mining*. Tahap ini dimulai dengan penggalian L_1 dan L_2 dengan menggunakan SOTrieIT sampai dengan penggalian *itemset frequent*. Lihat gambar 3.2.



Gambar 3.2 Blok diagram *data mining*

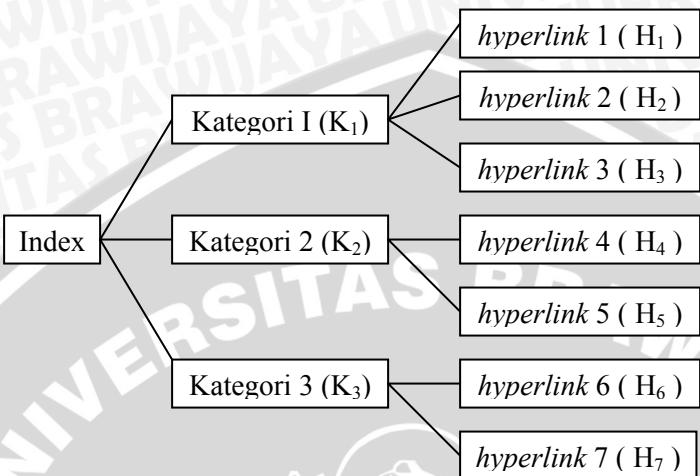
3.1 Preprocessing dan Transformation Data

Dalam persiapan data ini, hal pertama yang dilakukan adalah menentukan *website* untuk penempatan data produk yang diwakili oleh *hyperlink*. Disini akan dibuat simulasi *website* yang memuat *hyperlink – hyperlink* yang dibagi dalam beberapa kategori.

Untuk proses transformasi data, proses ini dilakukan pada *data log*, karena tidak semua bagian pada *data log* akan digunakan.

3.1.1 Struktur Website

Struktur website yang akan dipakai mempunyai beberapa bagian, yaitu: halaman utama (*index*), halaman kategori, dan halaman untuk produk (*hyperlink*), dapat dilihat pada gambar 3.3.



Gambar 3.3 Struktur website katalog produk

3.1.2 Sorting Hyperlink

Sorting hyperlink dibutuhkan untuk menemukan struktur *hyperlink* yang ada sehingga dapat disimpan dalam *database*, dan dapat dituliskan dalam bentuk tabel 3.1.

Tabel 3.1 Data hyperlink pada database

Id	url	rec_1	rec_2	rec_3	rec_4	rec_5
1	H ₁					
2	H ₂					
3	H ₃					
4	H ₄					
dst						
n	H _n					

3.1.3 Pemrosesan Access Log

Data *log* pada *web server* menunjukkan aktifitas pengunjung pada suatu *website*. Ada 2 format *web server log*, yaitu CLF (*Common Log Format*) dan ECLF (*Extended Common Log Format*).

Beda dari 2 format itu adalah pada penambahan informasi yang ada pada ECLF yaitu informasi mengenai halaman *referrer* dan *user agent* (G, Ridowati, 2003).

Tabel 3.2 Contoh data log CLF

No	Data
1	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/ HTTP/1.1" 200 604
2	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/head.php HTTP/1.1" 200 1393
3	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/start.php HTTP/1.1" 200 1074
4	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/navi.php HTTP/1.1" 200 3982
5	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/img/blank.gif HTTP/1.1" 200 43
6	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/xampp.css HTTP/1.1" 200 4178
7	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/img/head-fo.gif HTTP/1.1" 200 791
8	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/img/xampp-logo-new.gif HTTP/1.1" 200 4878
9	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/xampp.js HTTP/1.1" 200 573
10	127.0.0.1 - - [04/Jul/2009:09:21:46 +0700] "GET /xampp/img/head-windows.gif HTTP/1.1" 200 1370

Pada studi kasus dalam skripsi ini, data yang dipakai adalah data dengan format ECLF.

Tabel 3.3 Contoh data log ECLF

Data log
114.123.197.116 - - [29/Jul/2009:11:24:40 +0700] "GET /?p=56 HTTP/1.1" 200 8151 "http://katalog.gubbug.com/?cat=3" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.9) Gecko/2009040821 Firefox/3.0.9"

Tabel 3.4 Isi dan arti Elemen Extended Common Log Format

Nama elemen	Isi Elemen
Host	114.123.197.116
Ident	-
Authuser	-
Time	29/Jul/2009:11:24:40 +0700
Request	GET /?p=56
Status	200
Byte	8151
Referrer	http://katalog.gubbug.com/?cat=3
User Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.9) Gecko/2009040821 Firefox/3.0.9

Format data pada Tabel 3.3, dapat diasumsikan untuk setiap barisnya adalah sebuah *array* dimana elemen-elemennya dipisahkan dengan spasi. Dengan pola data tersebut, maka proses transformasi data (*transformation*) dapat dilakukan. Pada studi kasus ini, setiap baris data yang digunakan untuk *transformation* ditunjukkan pada Tabel 3.5.

Tabel 3.5 Transformation Data access log

No	Index Array	Jenis Data	Contoh Data
1	0	IP	114.123.197.116
2	1	Tanggal	2009-07-29
3	2	Jam	11:24:40
4	6	URL	/?p=56
6	16	Status data	200, 404

3.1.4 Pengelompokan Data Log

Pada pengelompokan data ini data asli dari *web server log* disaring sehingga hanya data yang *valid* saja yang akan diolah. Proses persiapan yang dilakukan adalah sebagai berikut:

1. Menghapus seluruh *request* dari data-data dengan jenis file grafik, yaitu file-file *request* yang berekstensi *gif*, *jpeg*, *jpg*, *bmp*,

- ico*, dan lain-lain. Juga menghapus *request* untuk jenis file *sound*, *video*, dan file lainnya..
2. Menghapus seluruh *request* yang *error*. Untuk mengetahui apakah sebuah *request error* atau tidak, dapat dilihat dari isi elemen status dan kata awal dari isi elemen *request*. Kode status menunjukan respon dari server terhadap permintaan *user*. Jika kode statusnya adalah 200 berarti bahwa *request* adalah *valid*, sedangkan kode status lainnya, seperti misalnya 404 berarti bahwa dokumen yang diminta tidak ada pada *web site*. Kata awal dari elemen *request* menunjukan permintaan *user* kepada *server*. Jika diawali dengan kata *GET* berarti *user* meminta sebuah dokumen untuk ditampilkan. Sedangkan jika kodennya adalah *HEAD* berarti bahwa *user* hanya menginginkan informasi dari dokumen, tetapi bukan dokumennya itu sendiri. Arti kode status dan kata awal dari *request* merupakan ketentuan dari protokol **HTTP** (Hyper Text Transfer Protocol).

3.1.5 Pembuatan Session

Setelah pengelompokan, *data log* yang ada akan dibagi kedalam *session - session*. Jika halaman web (*hyperlink*) merupakan identitas dari produk/*Item*. *Item* sendiri adalah nama *URI* (Universal Resource Identifier) hasil *request* dari *user*. Satu *session* adalah urutan *URI* yang diakses *user* dalam satu kunjungan tunggal. Cara yang dilakukan untuk pembentukan *session* adalah dengan melakukan pengelompokan berdasarkan *IPUser*-nya dan waktu melakukan akses. Secara berurutan *session* dibentuk dengan cara melihat *IPUser* pemakai. Jika *IPUser* berbeda, maka akan dianggap sebagai *session* yang berbeda. Sering kali terjadi satu buah IP akan digunakan bersama-sama oleh banyak *user*. Bila satu IP dipakai bersama, maka akan dianggap hanya satu *session*. Setelah melakukan pengecekan terhadap IP *user*, selanjutnya dilakukan pengecekan terhadap waktu. Untuk satu *session* dalam penelitian ini digunakan rentang waktu 30 menit. Artinya lamanya seorang *user* pada umumnya melakukan akses dalam satu kali kunjungan adalah 20 menit. Pembentukan *session* seperti di atas akan mulai terjadi jika jumlah *request* dari setiap IP lebih dari satu.

3.2 Data Mining

Untuk proses selanjutnya adalah proses *data mining*. Dimana tahap pertama adalah pembentukan *itemset*. *Itemset* adalah himpunan *item-item*. *Item – item* sendiri dikumpulkan menjadi satu berdasarkan *IP* yang akan kita sebut *session* (*S*). Dalam algoritma Fold-Growth ini dibagi menjadi empat tahapan utama yaitu (a) penggalian L_1 dan L_2 dengan menggunakan SOTrieIT, (b) pemangkasan *item-item* yang tidak *frequent*, (c) membangun FP-Tree menggunakan transaksi-transaksi yang telah dipangkas dan, (d) penggalian *itemset frequent* dengan algoritma FP-Growth.

3.2.1 Penggalian L_1 dan L_2 dengan menggunakan SOTrieIT

Tabel 3.6 Contoh data hyperlink pada session

Set of hyperlink SHL(<i>Sj</i>) pada session S1- S10	
<i>Sj</i>	SHL(<i>Sj</i>)
S1	{H1, H2, H3, H4, H5, H6, H2, H3, H4}
S2	{H1, H2, H4, H5, H6, H11, H12, H13, H3, H1}
S3	{H1, H2, H3, H4, H5, H6, H7, H8, H12, H6, H7}
S4	{H1, H2, H3, H8, H9, H1, H1, H3}
S5	{H1, H3, H4, H5, H6, H5, H4, H6, H7, H8, H12}
S6	{H2, H3, H5, H7, H8, H9, H3, H10}
S7	{H2, H3 , H4, H5, H6, H8, H11, H12}
S8	{H1, H3, H7, H8, H9, H1, H3, H8, H10, H2}
S9	{H1, H4, H5, H3, H4, H6, H7, H8, H9, H10}
S10	{H2, H3, H4, H5, H6, H11, H12, H13}

*Set of hyperlink SHL(*Sj*)* pada Tabel 3.6 masih belum memenuhi *format* yang dibutuhkan untuk proses *data mining*, sehingga perlu dilakukan pembersihan duplikasi data terlebih dahulu. Data yang sudah tidak memiliki duplikasi ini dapat dituliskan ke dalam bentuk tabel, ditunjukkan pada Tabel 3.7

Tabel 3.7 Data hyperlink setelah pembersihan duplikasi

Set of hyperlink SHL(Sj) pada session S1- S10	
Sj	SHL(Sj)
S1	{H1, H2, H3, H4, H5, H6}
S2	{H1, H2, H3, H4, H5, H6, H11, H12, H13}
S3	{H1, H2, H3, H4, H5, H6, H7, H8, H12}
S4	{H1, H2, H3, H8, H9}
S5	{H1, H3, H4, H5, H6, H7, H8, H12}
S6	{H2, H3, H5, H7, H8, H9, H10}
S7	{H2, H3 , H4, H5, H6, H8, H11, H12}
S8	{H1, H2, H3, H7, H8, H9, H10}
S9	{H1, H4, H5, H6, H7, H8, H9, H10}
S10	{H2, H3, H4, H5, H6, H11, H12, H13}

Setelah pembersihan duplikasi *hyperlink*, proses selanjutnya adalah pembangkitan *itemset L₁* / *hyperlink set - 1* menggunakan struktur data SOTrieIT. Hasil dari pembentukan dapat dilihat pada tabel 3.8.

Tabel 3.8 Data pembentukan *itemset L₁*

Id_link	kemunculan	Support
1	7	0.7
2	8	0.8
3	9	1.0
4	7	0.7
5	8	0.8
6	7	0.7
7	5	0.5
8	7	0.7
9	4	0.4
10	3	0.3
11	3	0.3
12	5	0.5
13	2	0.2

Hasil pembangkitan *itemset* L_2 / *hyperlink set – 2* dapat dilihat pada tabel 3.9.

Tabel 3. 9 Data pembentukan *itemset* L_2

Id_link 1	Id_link 2	Support
1	2	0.5
1	3	0.6
1	4	0.5
⋮		
11	13	0.2
12	13	0.2

Tabel 3. 10 Data pembentukan *itemset* L_2 dalam bentuk matrix.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0,5	0,6	0,5	0,5	0,5	0,4	0,5	0,3	0,2	0,1	0,3	0,2
2	0,5	0	0,8	0,5	0,6	0,5	0,3	0,5	0,3	0,2	0,3	0,4	0,2
3	0,6	0,8	0	0,6	0,7	0,6	0,4	0,6	0,3	0,2	0,3	0,5	0,2
4	0,5	0,5	0,6	0	0,7	0,7	0,3	0,4	0,1	0,1	0,3	0,5	0,2
5	0,5	0,6	0,7	0,7	0	0,7	0,4	0,5	0,2	0,2	0,3	0,5	0,2
6	0,5	0,5	0,6	0,7	0,7	0	0,3	0,4	0,1	0,1	0,3	0,5	0,2
7	0,4	0,3	0,4	0,3	0,4	0,3	0	0,5	0,3	0,3	0	0,2	0
8	0,5	0,5	0,6	0,4	0,5	0,4	0,5	0	0,4	0,3	0,1	0,3	0
9	0,3	0,3	0,3	0,1	0,2	0,1	0,3	0,4	0	0,3	0	0	0
10	0,2	0,2	0,2	0,1	0,2	0,1	0,3	0,3	0,3	0	0	0	0
11	0,1	0,3	0,3	0,3	0,3	0,2	0	0,1	0	0	0	0,3	0,2
12	0,3	0,4	0,5	0,5	0,5	0,5	0,2	0,3	0	0	0,3	0	0,2
13	0,2	0,2	0,2	0,2	0,2	0,2	0	0	0	0	0,2	0,2	0

3.2.2 Tahap Pemangkasan Item-item yang Tidak Frequent

Dalam tahap ini, akan dilakukan pemangkasan pada setiap transaksi yang ada dalam basis data dengan menggunakan L_1 dan L_2 . Untuk setiap transaksi T , pada *itemset* L_k yang terdapat dalam transaksi tersebut dimana panjang k lebih dari 2, akan dilakukan pengecekan dengan menggunakan L_1 dan L_2 . Sehingga, untuk *item-item* yang dianggap tidak *frequent* akan dilakukan pemangkasan. Sebuah *item* dikatakan tidak *frequent* apabila nilai *support count*-nya

kurang dari batas *minimum support* yang telah ditentukan oleh pengguna. Setelah dilakukan pemangkasan terhadap transaksi T , maka *item-item* yang terdapat dalam transaksi tersebut akan diurutkan berdasarkan nilai *support count* yang paling besar. Dengan menggunakan L_1 dan L_2 yang didapatkan melalui SOTrieIT, maka akan dihasilkan *Ordered Frequent Items* yang telah dipangkas. Apabila kita tentukan *minimum support* sebesar 60%, maka akan didapatkan hasil transaksi yang telah dipangkas dan diurutkan, terlihat pada tabel 3.11.

Tabel 3. 11 Data transaksi setelah pemangkasan L_1 dan L_2

Id_link	kemunculan	Support
3	9	0.9
2	8	0.8
5	8	0.8
1	7	0.7
4	7	0.7
6	7	0.7
8	7	0.7

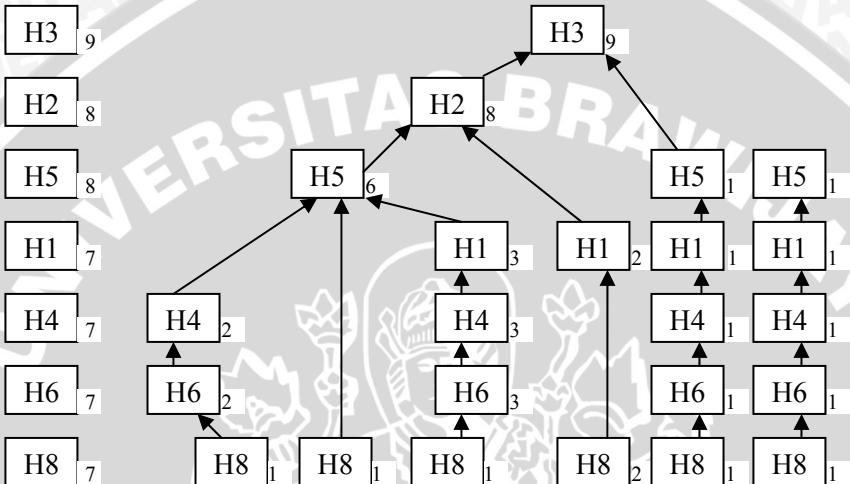
3.2.3 Tahap Pembangunan FP-Tree

Pada tahapan ini, akan dilakukan pembangunan FP-Tree dengan menggunakan data transaksi T yang telah dipangkas dan diurutkan berdasarkan nilai *support count*. Dengan perolehan *Frequent Items* setelah dipangkas dan diurutkan (lihat tabel 3.12), maka akan dibangun FP-Tree, lihat gambar 3.2.

Tabel 3. 12 Data transaksi setelah dipotong, dgn min sup=60%

Transaksi	
Tj	STj
T1	{H3, H2, H5, H1, H4, H6}
T2	{H3, H2, H5, H1, H4, H6}
T3	{H3, H2, H5, H1, H4, H6, H8}
T4	{H3, H2, H1, H8}
T5	{H3, H5, H1, H4, H6, H8}
T6	{H3, H2, H5, H8}
T7	{H3, H2 , H5, H4, H6, H8}

T8	{H3, H2, H1, H8}
T9	{H5, H1, H4, H6, H8}
T10	{H3, H2, H5, H4, H6}



Gambar 3.4 FP-Tree setelah pemangkasan

3.2.4 Tahap Penggalian Itemset Frequent

Setelah tahap pembangunan FP-Tree selesai, dilanjutkan dengan tahap penggalian itemset frequent dengan menggunakan algoritma FP-Growth pada FP-Tree tersebut.

Minimum support 60% x 10 =6

Conditional pattern base for H8 : 7
 (H3, H2, H5, H1, H4, H6, H8) : 1,
 (H3, H2, H1, H8) : 2,
 (H3, H5, H1, H4, H6, H8) : 1,
 (H3, H2, H5, H8) : 1,
 (H3, H2, H5, H4, H6, H8) : 1,
 (H5, H1, H4, H6, H8) : 1
 F-list : H3:6

Conditional FP-Tree for H8

H3:6

Frequent Pattern

H3:6, (H3,H8):6

Conditional pattern base for H6:7

(H3, H2, H5, H1, H4, H6): 3,

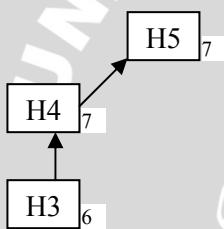
(H3, H2, H5, H4, H6): 2,

(H3, H5, H1, H4, H6): 1,

(H5, H1, H4, H6): 1,

F-list : H5:7, H4:7, H3:6

Conditional FP-Tree for H6



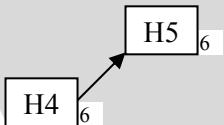
Gambar 3.5 Conditional FP-Tree for H6

Recursion

Conditional pattern base for (H3,H6) : 6

(H5,H4): 6

Conditional FP-Tree for (H3,H6) (gambar 3.4)



Gambar 3.6 Conditional FP-Tree for (H3,H6)

Recursion

Conditional pattern base for (H4,H3,H6) : 6

H5: 6

Conditional FP-Tree for (H4,H3,H6)

H5: 6

Frequent Pattern

H5:6, (H5,H4,H3,H6):6

Conditional pattern base for (H5,H3,H6) : 6

NULL

Conditional FP-Tree for (H5,H3,H6)

NULL

Frequent Pattern

(H5,H3,H6):6, and (H3,H6):6

Conditional pattern base for (H4,H6) : 7

H5: 7

Conditional FP-Tree for (H4,H6)

H5: 7

Frequent Pattern

(H5,H4,H6):7

Conditional pattern base for (H5,H6) : 7

NULL

Conditional FP-Tree for (H5,H6)

NULL

Frequent Pattern

(H5,H6):7 and H6:7

Conditional pattern base for H4:7

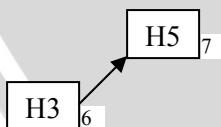
(H3, H2, H5, H1, H4): 3,

(H3, H2, H5, H4): 2,

(H3, H5, H1, H4): 1, (H5, H1, H4): 1,

F-list : H5:7, H3:6

Conditional FP-Tree for H4 (gambar 3.5)



Gambar 3.7 Conditional FP-Tree for H4

Recursion

Conditional pattern base for (H3,H4) : 6

H5:6

Conditional FP-Tree for (H3,H4)

H5:6

Frequent Pattern

H5:6, (H5,H3,H4):6

Conditional pattern base for (H5,H4) : 7

NULL

Conditional FP-Tree for (H5,H4)

NULL

Frequent Pattern

(H5,H4):6, and H4:6

Conditional pattern base for H1:7

(H3, H2, H5, H1): 3,

(H3, H5, H1): 1,

(H3, H2, H1): 2,

(H5, H1): 1

F-list : H3:6,

Conditional FP-Tree for H1

H3:6

Frequent Pattern

H3:6, (H3,H1):6

Conditional pattern base for H5:8

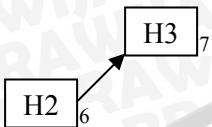
(H3, H2, H5): 6,

(H3, H5): 1,

(H5): 1

F-list : H3:7, H2:6

Conditional FP-Tree for H5 (gambar 3.6)



Gambar 3.8 Conditional FP-Tree for H5

Recursion

Conditional pattern base for (H2,H5) : 6

H3:6

Conditional FP-Tree for (H2,H5)

H3:6

Frequent Pattern

H3:6, (H3,H2,H5):6

Conditional pattern base for (H3,H5) : 7

NULL

Conditional FP-Tree for (H3,H5)

NULL

Frequent Pattern

(H3,H5):6, and H5:6

Conditional pattern base for H2:8

(H3, H2): 8,

F-list : H3:8

Conditional FP-Tree for H5

H3:8

Frequent Pattern

H3:8, (H3,H2):8

Conditional pattern base for H3:9

NULL

Conditional FP-Tree for H3

NULL

Frequent Pattern

H3:9

Hasil *frequent Pattern / frequent Itemset* yang terbentuk dengan *minimum support* 60% untuk beberapa akhiran (*suffix*) ditunjukkan pada tabel 3.13

Tabel 3. 13 Hasil *Frequent Itemset*, dgn min sup=60%

Prefix	Frequent Pattern
H8	(H8), (H3,H8)
H6	(H6), (H5,H4,H3,H6), (H5,H3,H6), (H3,H6), (H5,H4,H6), (H5,H6)
H4	(H4), (H5,H3,H4), (H5,H4)
H1	(H1), (H3,H1)
H5	(H5), (H3,H2,H5), (H3,H5)
H2	(H2), (H3,H2)
H3	(H3)

Tabel 3. 14 Hasil *Frequent Itemset* yang telah diurutkan

Frequent Pattern	kemunculan	Support
(H3,H2)	8	0.8
(H5,H4,H6)	7	0.7
(H5,H6)	7	0.7
(H3,H6)	6	0.6
(H3,H8)	6	0.6
(H5,H4,H3,H6)	6	0.6
(H5,H3,H6)	6	0.6
(H5,H3,H4)	6	0.6
(H5,H4)	6	0.6
(H3,H1)	6	0.6
(H3,H2,H5)	6	0.6
(H3,H5)	6	0.6

Dari tabel 3.14 diatas dicari mulai dari panjang *itemset* yang paling besar, dan dimasukkan rekomendasinya pada tabel *hyperlink*.

Tabel 3.15 Hasil rekomendasi yang didapat

url	rec1	rec2	rec3	rec4	rec5
H3	H5	H4	H6		
H4	H5	H3	H6		
H5	H4	H3	H6		
H6	H5	H4	H3		
H2	H3	H5			
H8	H3				
H1	H3				

3.3 Perancangan tabel

Berikut adalah tabel – tabel yang digunakan pada penelitian ini. Untuk tabel k_hs, jumlah field url disesuaikan dengan *itemset* 1_hs s/d k_hs.

Tabel 3.16 Atribut tabel hyperlink

Field	Type	Null	Default
id	int(11)	Yes	NULL
url	varchar(255)	Yes	kosong
rec_1	varchar(255)	Yes	kosong
rec_2	varchar(255)	Yes	kosong
rec_3	varchar(255)	Yes	kosong
rec_4	varchar(255)	Yes	kosong
rec_5	varchar(255)	Yes	kosong

Keterangan :

- id : kode *hyperlink* (*auto increment*)
- url : alamat url *hyperlink* (format data log)
- rec_ : rekomendasi 1 s/d 5

Tabel 3.17 Atribut tabel k_hs

Field	Type	Null	Default
id	int(11)	Yes	NULL
url	varchar(255)	Yess	
num	int(10)	Yes	
supp	float(4,2)	Yes	NULL

Keterangan :

- id : id (*auto increment*)
- url : nama url (*hyperlink*)
- num : banyak kemunculan pada *session*
- supp : nilai *support hyperlink*

UNIVERSITAS BRAWIJAYA



BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

4.1.1 Persiapan Data

Langkah pertama yang dilakukan dalam pesiapan data yaitu data apa yang dipakai dalam sistem rekomendasi. Dalam hal ini data yang dipakai adalah data *hyperlink* yang menunjukkan alamat suatu item/produk. Data *hyperlink* diambil dari tabel wp_posts pada database db_wp yang ada pada sistem CMS Wordpress. Dari tabel wp_posts diambil field *guid* dan dimasukkan kedalam tabel *hyperlink* pada bagian field *url*. Lihat source code dibawah ini:

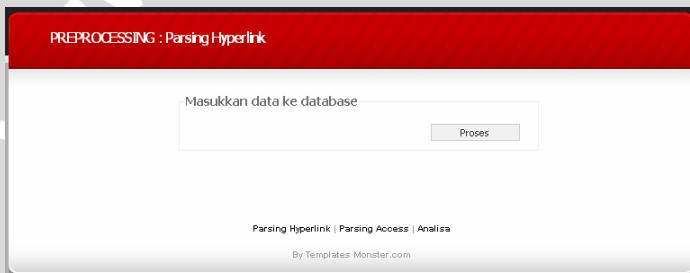
```
018 function parsing_hyperlink()
019 {
020     $temp = array();
021     $result = mysql_query("select guid from
022         wp_posts where post_type = 'post'");
023     while ($row = mysql_fetch_array($result))
024     {
025         $masuk = $row['guid'];
026         array_push($temp, $masuk);
027     }
028     foreach($temp as $guid)
029     {
030         $sql_check = "select * from
031             `hyperlink` where `url` = '$guid'
032             limit 0,1";
033         $query = mysql_query($sql_check);
034         $num = mysql_num_rows($query);
035         if($num === 0)
{
            $str = "insert into `hyperlink`
                (`id`, `url`) values ('',
                '$guid')";
            mysql_query($str);
            echo "<em>$guid</em> dimasukkan
                kedalam database<br>";
        }
    }
}
```

```

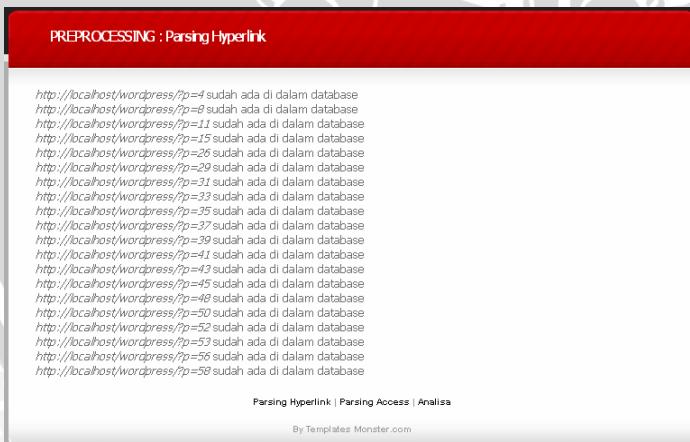
036      }
037      else
038      {
039          echo "<em>$guid</em> sudah ada di
040          dalam database<br>" ;
041      }
042  }

```

Untuk *interfacenya* dapat dilihat pada gambar 4.1 dan gambar 4.2 dibawah ini..



Gambar 4.1 Interface *Parsing hyperlink*



Gambar 4.2 Interface hasil *parsing hyperrlink*

4.1.2 Parsing Data Access Log

Parsing data access log ini dimulai dengan menginputkan file, setelah itu di data yang ada dalam file *disorting* apakah *requestnya* berupa "Get" serta yang direquest berupa *file posting* "?p=" dan statusnya < 400. Lihat *source code* dibawah ini:

```
046     $polasort = array();
047     $dataurl = array();
048     $handle = @fopen("$data_log", "r");
049     if ($handle)
050     {
051         while (!feof($handle))
052         {
053             $buffer = fgets($handle, 4096);
054             $row_file = explode(" ", $buffer);
055             if ($row_file[5] == '"GET' &&
056             $row_file[8] < 400 && strstr($row_file[6],
057             "/?p="))
058             {
059                 $ip = $row_file[0];
060                 $datetime = str_replace("[", " ",
061                 $row_file[3]);
062                 $datetime = str_replace("2009:", "2009 ",
063                 $datetime);
064                 $datetime = str_replace("/", "-", $datetime);
065                 $datetime = strtotime($datetime);
066                 $url = "http://localhost/wordpress" .
067                 $row_file[6];
068                 $polo = array($ip, $datetime, $url);
069                 array_push($dataurl, $polo);
070                 array_push($polasort, $datetime);
071             }
072         }
073     }
074     fclose($handle);
075 }
```

4.1.3 Pembuatan Session

Pembuatan *session* dimulai dengan membuat *range* waktu yang akan dipakai dalam pembuatan *session*. Setiap *session* adalah 30 menit, jadi dalam waktu 30 menit data akan dikelompokkan menjadi 1 *session*, 30 kemudian merupakan *session* berikutnya. Lihat *source code* dibawah ini:

```
081     $max = 0;
082     $min = 0;
083     foreach($raw_url as $key=>$urls)
084     {
085         if($max < $urls[1])
086             $max = $urls[1];
087         if($key == 0)
088             $min = $urls[1];
089         if($min > $urls[1])
090             $min = $urls[1];
091     }
092     $starttime = mktime(date("H", $min), 0,
0, date("m", $min), date("d", $min),
date("Y", $min));
093     $start_counter = 0;
094     $session_counter = 0;
095     $session_array = array();
096     while($starttime < $max)
097     {
098         $i = $start_counter;
099         $endtime = $starttime + 1800;
100         for($i = $start_counter; $i < count($raw_url); $i++)
101         {
102             if($raw_url[$i][1] < $endtime)
103             {
104                 $session_array[$session_counter][] =
$raw_url[$i];
105             }
}
```

```

106         else
107     {
108         $start_counter = $i;
109         $i = count($raw_url);
110     }
111 }
112 $session_counter++;
113 $starttime = $endtime ;
114 if($starttime > $max)
115     break;
116 }

```

Berikutnya adalah pemangkasan data yang sama (sama *ip* dan sama *url*) dalam satu *session*. Lihat *source code* dibawah ini:

```

125     foreach($session_array as $key_root =>
126 $url_array)
127     {
128         foreach($url_array as $key_b =>
129 $val_b)
130         {
131             $temp[0] = $val_b[0];
132             $temp[1] = $val_b[2];
133             $url_array2 = $url_array;
134             unset($url_array2[$key_b]);
135             foreach($url_array2 as $key_c =>
136 $val_c)
137             {
138                 if($val_c[0] == $temp[0] &&
139 $val_c[2] == $temp[1])
140                 {
141                     unset
142                     ($url_array[$key_c]);
143                     unset
144                     ($session_array[$key_root][$key_c]);
145                 }
146             }
147         }
148     }

```

Setelah proses pemangkasan data yang sama atau bisa disebut pembersihan data, maka langkah selanjutnya adalah mengelompokkan data berdasarkan no *ip*. Lihat *source code* dibawah ini:

```

151  $_SESSION['grouped_session'] = array();
152      foreach($session_array as $keyz =>
153      $valz)
154      {
155          $_SESSION['grouped_session'][$keyz]
156          = array();
157          foreach($valz as $valz2)
158          {
159              $ip = $valz2[0];
160              $_SESSION['grouped_session'][$keyz][$ip][] =
161              $valz2[2];
162          }
163      }

```

Untuk tampilan *interface* proses *input* data *access log* dan hasil dari proses sampai pembentukan *session* dapat dilihat pada gambar 4.3 dan gambar 4.4.

PREPROCESSING : Parsing Data Log

Input

File Log

[Parsing Hyperlink](#) |
 [Parsing Access](#) |
 [Analisa](#)

By Templates: Monster.com

Gambar 4.3 Input data access log

PREPROCESSING : Parsing Data Log

data session sebelum eleminasi ::

```
session 1 mempunyai : 2 data
session 3 mempunyai : 2 data
session 4 mempunyai : 1 data
session 5 mempunyai : 19 data
session 6 mempunyai : 5 data
session 7 mempunyai : 17 data
session 8 mempunyai : 4 data
session 9 mempunyai : 14 data
session 10 mempunyai : 7 data
session 11 mempunyai : 33 data
```

data session setelah eleminasi ::

```
session 1 mempunyai : 2 data
session 3 mempunyai : 2 data
session 4 mempunyai : 1 data
session 5 mempunyai : 17 data
session 6 mempunyai : 2 data
session 7 mempunyai : 13 data
session 8 mempunyai : 2 data
session 9 mempunyai : 14 data
session 10 mempunyai : 7 data
session 11 mempunyai : 19 data
```

data session setelah digrouping ::

```
session 1 :
    IP 114.123.10.11 : 2 data

session 3 :
    IP 114.123.10.11 : 2 data

session 4 :
    IP 114.123.10.10 : 1 data

session 5 :
    IP 114.123.10.17 : 4 data
    IP 114.123.10.12 : 1 data
    IP 114.123.10.10 : 3 data
    IP 114.123.10.13 : 5 data
    IP 114.123.10.15 : 3 data
    IP 114.123.10.16 : 1 data

session 6 :
    IP 114.123.10.15 : 1 data
    IP 114.123.10.13 : 1 data

session 7 :
    IP 114.123.10.15 : 2 data
    IP 114.123.10.16 : 3 data
    IP 114.123.10.13 : 4 data
    IP 114.123.10.10 : 2 data
    IP 114.123.10.17 : 1 data
    IP 114.123.10.11 : 1 data

session 8 :
    IP 114.123.10.15 : 1 data
    IP 114.123.10.17 : 1 data
```

Gambar 4.4 Hasil proses parsing data log dan session

4.1.3 Pembentukan Itemset

Setelah data terbentuk dan dibagi persession, maka proses berikutnya adalah pembentukan *itemset*. Pembentukan *itemset* dimulai dengan penggalian L_1 dan L_2 . Lihat *source code* dibawah ini:

```
230     $temp = $set_root;
231     $set_level1 = array();
232     $x = 0;
233     foreach ($set_root as $key => $val)
234     {
235         unset($temp[$x]);
236         $temp2 = array();
237         foreach ($temp as $keyz => $valz)
238         {
239             $temp3['node_1'] = $val;
240             $temp3['node_2'] = $valz;
241             $temp3['val'] = 0;
242             array_push($temp2, $temp3);
243         }
244         $set_level1 =
245         array_merge($set_level1, $temp2);
246         $x++;
247     }
248     foreach ($_SESSION[grouped_session] as
249     $key => $val)
250     {
251         foreach ($val as $key2 => $val2)
252         {
253             foreach ($set_level1 as $key3 =>
254             $val3)
255             {
256                 $node1 = $val3['node_1'];
257                 $node2 = $val3['node_2'];
258                 $node_val = $val3['val'];
259                 if((array_search($node1,
$val2) !== FALSE) && (array_search($node2,
```

```

        $val2) !== FALSE))
258     $set_level1[$key3]['val']++;
259     }
260   }
261 }
```

Kemudian dilakukan pemangkasan terhadap *item – item* yang mempunyai *support* dibawah *minimum support* pada *itemset* ke 2 dan diurutkan berdasarkan *support* yang paling besar. Lihat *source code* dibawah ini:

```

$support_limit =
$_POST['input_support'];
$sql_str = "select * from `2_hs` where
`supp` >= $support_limit order by `supp` desc";
$sql_q = mysql_query($sql_str);
$levl = array();
$url1 = array();
while($r = mysql_fetch_assoc($sql_q))
{
    array_push($levl, $r);
    $u_1 = $r['url_1'];
    $u_2 = $r['url_2'];
    if(array_search($u_1, $url1) ===
FALSE)
        array_push($url1, $u_1);
    if(array_search($u_2, $url1) ===
FALSE)
        array_push($url1, $u_2);
}
natsort($url1);
printr($url1);

$sql_str = "select * from `1_hs` order
by `supp` desc";
$sql_q = mysql_query($sql_str);
$url2 = array();
```

```
325     while($r = mysql_fetch_assoc($sql_q))
326     {
327         $temp= array();
328         if(array_search($r[url_1], $url1)
329 != FALSE)
330         {
331             $url2[$r[url_1]] = $r[supp];
332         }
333         printr($url2);
334
335         $temp_session =
336 $_SESSION[grouped_session];
337         $sliced_session = array();
338
339         foreach($temp_session as $key1 => $val1)
340         {
341             $sliced_session[$key1] = array();
342             foreach ($val1 as $key2 => $val2)
343             {
344                 $sliced_session[$key1][$key2] =
345 array();
346                 foreach ($url2 as $key_url =>
347 $val_url)
348                 {
349                     $this_url = $key_url;
350                     if(array_search($this_url,
351 $val2) != FALSE)
352                     {
353                         $sliced_session[$key1][$key2][] =
354 $this_url;
355                     }
356                 }
357                 if
358 (count($sliced_session[$key1][$key2]) < 1)
359 unset($sliced_session[$key1][$key2]);
360             }
361         }
362     }
```

Langkah terakhir adalah penggalian *itemset – itemset* dari data yang telah dipangkas. Sehingga didapatkan *itemset* yang mempunyai *support* berdasarkan *minimum support* yang telah ditentukan. Langkah – langkahnya dimulai dari memproses node yang paling bawah/terakhir. Lihat *source code* dibawah ini:

```
357 echo "<h4>==Mulai Proses==</h4>";
358 $url3 = array_keys($url2);
359 while (count($url3) > 0)
360 {
361     unset($new_support);
362     unset($new_support2);
363     unset($support);
364     $the_last = array_pop($url3);
365     echo "<h5 style='text-align: left;
padding-left:20px;'><b>Proses untuk
$the_last </b></h5>";
366     $temp_trx = array();
367     foreach($sliced_session as $session)
368     {
369         foreach($session as
370 $transaction)
371         {
372             if(array_search($the_last,
373 $transaction))
374                 $temp_trx[ ] = $transaction;
375         }
376         $support = array();
377         foreach($temp_trx as $transaction)
378         {
379             foreach($transaction as $url)
380             {
381                 if(!isset($support[$url]))
382                     $support[$url] = 1;
```

```

383         else
384             $support[$url]++;
385     }
386 }
387
388 foreach($support as $url => $value)
389 {
390     $val = round((($value / $pembagi
391 ) * 100));
392     if($val <
393     $_POST['input_support'])
394         unset ($support[$url]);
395
396     printr($support);
397     $flag = count($support);
398     $limit = 0;

```

Hasil akhir dari analisa ini adalah jika telah ditemukan *itemset* yang *frequent*. Lihat *source code* dibawah ini:

```

551 echo "<b>Hasil akhir </b><br><br> ";
552 $x=0;
553 if(isset($new_support))
554 {
555     foreach($new_support as $ksupport => $nsupport)
556     {
557         $x++;
558         echo "<b>Set #".$x."</b>";
559         echo "<ul style='padding-left:50px;'>";
560         echo "<li>" . $nsupport[0] . "</li>";
561         echo "<li>" . $nsupport[1] . "</li>";
562         echo "</ul>";
563         if (is_array($new_support2[$ksupport]))
564         {
565             echo "        <b>Subset level
2<br></b> ";

```

```
566 foreach($new_support2[$ksupport] as $ksupport2 =>
567 $nsupport2)
568 {
569 echo "      &nbsp; &gt; Subset
570 #$ksupport2";
571 echo "<ul style='padding-left:75px;'>";
572 echo "<li> . $nsupport2[0] . "</li>";
573 echo "<li> . $nsupport2[1] . "</li>";
574 echo "<li> . $nsupport2[2] . "</li>";
575 echo "</ul>";
576 $semua_hasil[] =
577 $sub_support_hasil[$ksupport][$ksupport2];
578 foreach($sub_support_hasil[$ksupport][$ksupport2]
579 as $sssupport_hasil)
580 {
581 echo "<li> . $sssupport_hasil . "</li>";
582 echo "</ul><br>";
583 }
584 echo "&nbsp; &gt; Hasil";
585 echo "<ul style='padding-left:50px;'>";
586 $semua_hasil[] = $new_support_hasil[$ksupport];
587 foreach($new_support_hasil[$ksupport] as
588 $nsupport_hasil)
589 echo "<li> . $nsupport_hasil . "</li>";
590 echo "</ul><br>";
591 }
592 }
593 else if (isset($support))
594 {
595 echo "Set 1";
596 foreach($support as $key => $val)
597 {
```

```
598         echo "<li>$key</li>";
599     }
600
601     if(count($support) == 0)
602     {
603         echo "<li>$the_last</li>";
604         $semua_hasil[] = $the_last;
605     }
606     else
607     {
608         $semua_hasil[] = $the_last;
609     }
610 }
611 else
612 {
613     echo "Set 1";
614     echo "<li>$the_last</li>";
615     $semua_hasil[] = $the_last;
616 }
617 echo "<br><br>";
618
619 foreach($sliced_session as $key1 =>
620 $val1)
621 {
622     foreach($val1 as $key2 => $val2)
623     {
624         foreach($val2 as $key3 => $val3)
625         {
626             if ($val3 == $the_last)
627                 unset($sliced_session[$key1][$key2][$key3]);
628         }
629     }
}
```

4.2 Analisa Hasil

Berikut adalah spesifikasi *hardware* yang digunakan untuk melakukan proses rekomendasi *hyperlink* beserta pengambilan data-data yang dibutuhkan:

1. Spesifikasi komputer sebagai alat uji untuk memproses pembuatan sistem rekomendasi adalah sebagai berikut:
 - Prosesor Intel(R) Celeron(R) 550 2.0 GHz
 - Memori 512 MB share to vga 64 MB
 - Sistem operasi Windows XP Professional SP2
 - XAMPP Control Version 2.5
 - Web Server Apache versi 2.2.11
 - Database server MySQL versi 5.1.33
2. CMS (*Content Management System*) menggunakan wordpress 2.7. Dengan *themes* oleh *mg12*.
3. Data *access log* yang digunakan adalah *data log* percobaan yang disesuaikan dengan perhitungan manual.

Data yang digunakan dalam penelitian ini adalah data penelitian yang terdiri dari 13 *hyperlink*, yaitu:

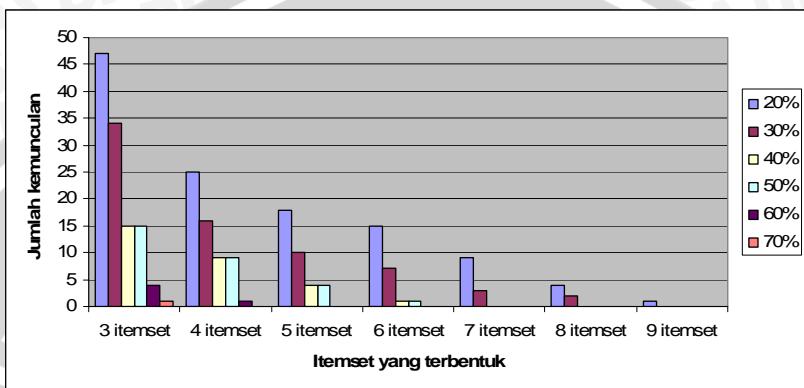
1. <http://localhost/wordpress/?p=4>
2. <http://localhost/wordpress/?p=8>
3. <http://localhost/wordpress/?p=11>
4. <http://localhost/wordpress/?p=15>
5. <http://localhost/wordpress/?p=26>
6. <http://localhost/wordpress/?p=29>
7. <http://localhost/wordpress/?p=31>
8. <http://localhost/wordpress/?p=33>
9. <http://localhost/wordpress/?p=35>
10. <http://localhost/wordpress/?p=37>
11. <http://localhost/wordpress/?p=39>
12. <http://localhost/wordpress/?p=41>
13. <http://localhost/wordpress/?p=43>

Dari data pelatihan, didapatkan bahwa *itemset* yang terbentuk pada saat minimum support 60%, mempunyai kesamaan dengan *itemset* yang didapatkan ketika dilakukan perhitungan manual.

Berikut adalah data yang diperoleh sesuai dengan *minimum support* yang ditentukan:

- *Minimum support 70%*
Hasil dengan 3 *itemset* : 1 hasil
- *Minimum support 60%*
Hasil dengan 3 *itemset* : 4 hasil
Hasil dengan 4 *itemset* : 1 hasil
- *Minimum support 50%*
Hasil dengan 3 *itemset* : 15 hasil
Hasil dengan 4 *itemset* : 9 hasil
Hasil dengan 5 *itemset* : 4 hasil
Hasil dengan 6 *itemset* : 1 hasil
- *Minimum support 40%*
Hasil dengan 3 *itemset* : 15 hasil
Hasil dengan 4 *itemset* : 9 hasil
Hasil dengan 5 *itemset* : 4 hasil
Hasil dengan 6 *itemset* : 1 hasil
- *Minimum support 30%*
Hasil dengan 3 *itemset* : 34 hasil
Hasil dengan 4 *itemset* : 16 hasil
Hasil dengan 5 *itemset* : 10 hasil
Hasil dengan 6 *itemset* : 7 hasil
Hasil dengan 7 *itemset* : 3 hasil
Hasil dengan 8 *itemset* : 2 hasil
- *Minimum support 20%*
Hasil dengan 3 *itemset* : 47 hasil
Hasil dengan 4 *itemset* : 25 hasil
Hasil dengan 5 *itemset* : 18 hasil
Hasil dengan 6 *itemset* : 15 hasil
Hasil dengan 7 *itemset* : 9 hasil
Hasil dengan 8 *itemset* : 4 hasil
Hasil dengan 9 *itemset* : 1 hasil

Dari data diatas dapat dilihat bahwa semakin kecil nilai *minimum support* maka akan dihasilkan jumlah *itemset* yang semakin banyak. Lihat gambar 4.5.



Gambar 4.5 Grafik kemunculan *itemset*

Detail dari data diatas untuk *minimum support* 60% adalah sebagai berikut:

Itemset A (support = 0.6)

- <http://localhost/wordpress/?p=26>
- <http://localhost/wordpress/?p=15>
- <http://localhost/wordpress/?p=11>
- <http://localhost/wordpress/?p=29>

Itemset B (support = 0.7)

- <http://localhost/wordpress/?p=26>
- <http://localhost/wordpress/?p=11>
- <http://localhost/wordpress/?p=29>

Itemset C (support = 0.6)

- <http://localhost/wordpress/?p=26>
- <http://localhost/wordpress/?p=15>
- <http://localhost/wordpress/?p=29>

Itemset D (support = 0.6)

- <http://localhost/wordpress/?p=26>

- <http://localhost/wordpress/?p=11>
- <http://localhost/wordpress/?p=15>

Itemset E (support = 0.6)

- <http://localhost/wordpress/?p=11>
- <http://localhost/wordpress/?p=8>
- <http://localhost/wordpress/?p=26>

Berikutnya adalah nilai *confidence* tiap *item* untuk *itemset A* dengan *support* 60% yang akan diwakilkan berdasarkan nomor *pages* dari tiap *hyperlink* melalui persamaan (1).

$$\text{Confidence } (A \Rightarrow B) = \frac{\text{Support_count}(A \Rightarrow B)}{\text{Support_count}(A)} \quad (1)$$

$$\text{Confidence } ((26) \Rightarrow (15)(11)(29)) = \frac{0.6}{0.8} = 0.75$$

$$\text{Confidence } ((15) \Rightarrow (26)(11)(29)) = \frac{0.6}{0.7} = 0.86$$

$$\text{Confidence } ((11) \Rightarrow (26)(15)(29)) = \frac{0.6}{0.9} = 0.66$$

$$\text{Confidence } ((29) \Rightarrow (26)(11)(15)) = \frac{0.6}{0.7} = 0.86$$

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari uji coba yang telah dilakukan pada implementasi sistem rekomendasi adalah sebagai berikut:

1. Sistem rekomendasi dapat membentuk *itemset* dari *hyperlink* yang ada pada data *access log*. Sehingga dapat diketahui pasangan *item* mana saja yang diakses secara bersamaan.
2. Semakin kecil nilai *minimum support* maka akan dihasilkan jumlah *itemset* yang semakin banyak
3. Nilai dari *confidence* berbanding lurus dengan nilai *minimum support* yang ditentukan terlebih dahulu.
4. Jika pada panjang *itemset* yang sama terdapat lebih dari satu *itemset*, maka dicari nilai *support* terbesar. Dari nilai *support* yang sama, pembobotan berikutnya adalah dengan nilai *confidence* dari masing – masing *item* terhadap *itemset*.

5.2 Saran

Berikut ini merupakan beberapa saran yang dapat diberikan untuk pengembangan penelitian sistem rekomendasi yang mengambil data dari *access log* lebih lanjut:

1. Pembagian *session* yang berdasarkan nomor *ip*, dapat menggunakan *agent* untuk mengetahui tiap *user* pada warnet atau *hotspot* yang mempunyai nomor *ip* yang sama.
2. Nilai *minimum support* dapat diubah – ubah sesuai dengan kebutuhan.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Anonymous. 2006. *Tutorial PHP: Array*. <http://www.Klik-Kanan.com>, tanggal akses: 27 Juli 2007.
- Barneveld, J.J.F. 2003. *User Interfaces for Personalized Information Systems*. Telematica Instituut.
- Borgelt, Christian. 2005. *An Implementation of the FP-growth Algoritma*. Department of Knowledge Processing and Language Engineering School of Computer Science, Magdeburg, Germany.
- Desikan, P. dan Srivastava J. 2003. *Mining Information from Temporal Behavior of Web Usage*. Department of Computer Science. University of Minnesota. USA.
- G. Ridowati. 2003. *Pencarian Pola dari Data Web Server Log dengan Menggunakan Teknik Association Rule*. Jurusan Teknik Informatika, Fakultas Teknik, Universitas Sanata Dharma, Yogyakarta.
- Han, J dan Kamber, M. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufman. San Fransisco.
- J. Han, H. Pei, and Y. Yin. 2000. *Mining Frequent Patterns without Candidate Generation*. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA.
- Larose, Daniel T. Discovering. 2005. *Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons. Inc. New Jersey.
- Masruri, Farid. 2006. *Perancangan Recommender System untuk personalisasi web e-commerce menggunakan metode Item-Based Collaborative Filtering*. Universitas Brawijaya.
- F. Perdana A.S. 2008. *Sistem Rekomendasi untuk Pengaturan Hyperlink pada Halaman Utama Website Menggunakan Metode Association Rule dan Hyperlink Clustering*. Universitas Brawijaya.

- Peter I. Hofgesang. 2004. *Web Usage Mining Structuring semantically enriched click stream data*. Vrije Universiteit Amsterdam, The Netherlands.
- Pramudiono, Iko. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. Ilmukomputer.com. 2003.
- P. Kazienko and P. Kolodziejksi. 2006. *Personalized Integration of Recommendation Methods for E-commerce*. Wroclaw University of Technology, Institute of Applied Informatics Wybrzeze S. Wyspianskiego 27, 50-370 Wroclaw, Poland.
- Rennolls, K. 2004. *Introduction to Data Mining- Chapter 2*.
- R. Soelaiman and N. M. Arini WP. 2006. *Analisis Kinerja Algoritma Fold-Growth dan FP-Growth pada Penggalian Pola Asosiasi*. Fakultas Teknologi Informasi, ITS, Surabaya.
- Santosa, Budi. 2007. *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta.
- Weiyang, Lin. 2000. *Association Rule Mining for Collaborative Recommender Systems*. Thesis. Worcester Polytechnic Institute.
- Y. Won; W. Ng; A. Das. 2001. *Fast Online Dynamic Association Rule Mining*. Nanyang Technological University, Singapore.