

Web Usage Mining Menggunakan Algoritma Adaptive Web Access Pattern Tree (AWAPT)

Fithratul Aini¹, Sri Widowati², Eko Darwiyanto³

^{1,2,3}Fakultas Teknik Informatika Institut Teknologi Telkom, Bandung

¹bundopito@yahoo.com, ²swd@ittelkom.ac.id, ³ekodarwiyanto@yahoo.com

Abstrak

Saat ini perkembangan website dalam World Wide Web semakin pesat dan data yang berhubungan dengan itu semakin banyak, para pengembang berusaha memberikan yang terbaik untuk mendapat performansi yang lebih baik dengan mengolah data yang didapatkan, sehingga muncullah ilmu Web mining. Salah satu kategori dalam Web Mining adalah Web Usage Mining (WUM), WUM merupakan proses untuk mengaplikasikan teknik web mining dalam melakukan analisa terhadap pola penggunaan dan analisa aktivitas pengunjung web yang terekam dalam server log dari website. WUM memiliki tiga tahapan pengerjaan yaitu *preprocessing data*, *pattern discovery*, dan *pattern analysis*.

Dalam Tugas Akhir ini akan diimplementasikan salah satu metode Frequent Sequential Pattern mining untuk melakukan pencarian pola sekuensial terhadap kecenderungan akses *user* dalam tahap *pattern discovery*, yaitu Adaptive Web Access Pattern Tree (AWAP Tree). Metode ini juga menggunakan teknik *Huffman code* sehingga metode ini akan lebih baik dari metode sebelumnya dalam penggunaan memory.

Kata kunci : Web Usage Mining, Frequent sequential Pattern mining, AWAPT.

Abstract

Now days, developing of websites and these data in World Wide Web grow rapidly. Developers of the site try to give the best effort to get better performance by mining all data that they get, so that there is Web Mining issue. One of Web mining's categories is Web Usage Mining (WUM). WUM is process to analyze pattern usage and user's activities recorded in server log of the website. WUM has three phases, they are *preprocessing data*, *pattern discovery*, dan *pattern analysis*.

In this final task, one of Frequent Sequential Pattern mining's method will be implemented to find sequential pattern of user access in *pattern discovery* phase, it is Adaptive Web Access Pattern Tree (AWAPT). This method uses *Huffman code* technic so that will be better in memory usage.

Keywords: Web Usage Mining, Frequent sequential Pattern mining, AWAPT.

1. Pendahuluan

Penggunaan internet mengalami peningkatan yang sangat pesat. Hal ini membuat para pengembang web berlomba-lomba untuk meningkatkan performa dan kualitas web masing-masing sehingga dapat memicu para pengembang untuk menggali *knowledge* yang ada dari aktivitas *user* dalam mengakses website, oleh karena itu munculah disiplin ilmu Web mining. Web mining adalah ilmu yang mempelajari penggalian data yang berhubungan dengan World Wide Web (WWW) [3].

Web mining dapat dikategorikan kedalam tiga kategori berdasarkan bagian mana dari sebuah website yang akan lakukan proses *mining* [2], Web Content Mining, Web Structured Mining, dan Web Usage Mining. Web Content Mining dan Web Structured Mining pada proses *mining*nya mencari informasi dari data yang tersedia secara *on-line*, sedangkan untuk Web Usage Mining data yang akan diolah dapat diperoleh secara *off-line* yaitu data log akses user yang tersimpan dalam *web log server* [4]. Pengolahan data secara *off-line* lebih mudah

dikarenakan dalam pelaksanaan proses *mining* tidak mewajibkan tersediannya koneksi internet.

Web Usage Mining (WUM) merupakan proses untuk mengaplikasikan teknik web mining dalam melakukan analisa terhadap pola penggunaan (*usage pattern*) dan analisa aktivitas pengunjung web yang terekam dalam server log dari website. WUM memiliki tiga tahapan pengerjaan yaitu *preprocessing data*, *pattern discovery* (pencarian pola), dan *pattern analysis* (analisis pola yang terbentuk). Walaupun WUM dalam proses *mining* tidak dibutuhkan koneksi internet tetapi data yang tersedia tidaklah sedikit, dan proses pencarian pola terhadap data yang sangat banyak tersebut membutuhkan sebuah teknik pembahasan yang khusus. Salah satu teknik yang dapat digunakan adalah Frequent Sequential Pattern (FSP) mining, yaitu teknik pencarian pola sekuensial terhadap kecenderungan akses *user*. Menemukan Frequent Sequential Pattern (FSP) merupakan masalah yang penting dalam WUM [2]. Oleh karena itu digunakan sebuah metode pencarian pola sekuensial yaitu Adaptive Web Access Pattern Tree (AWAP Tree). AWAP Tree merupakan sebuah cara

pencarian pola kecenderungan akses *user* dalam bentuk pohon sekuensial, dimana setiap transaksi akses user yang tercatat dalam web log akan dimasukkan kedalam pohon sekuensial. Metode ini juga menggunakan teknik *Huffman code* sehingga metode ini akan lebih baik dari metode sebelumnya dalam penggunaan memory.

Tujuan yang ingin dicapai dalam tugas akhir ini, melihat pengaruh nilai Minimum Support dan jumlah transaksi serta pengaruh waktu terhadap performansi system. Dan mengetahui keterkaitan antar item dengan melihat binary code item-item.

Metodologi penyelesaian masalah yang akan digunakan adalah:

1. Studi literature
2. Pengumpulan data
3. Analisis kebutuhan dan perancangan sistem
4. Implementasi rancangan sistem
5. Testing dan Evaluasi
6. Penarikan simpulan dan penyusunan laporan tugas akhir.

2. Landasan Teori

2.1 Web Mining

Web mining merupakan salah satu cabang ilmu *data mining*. *Web mining* melakukan penggalian terhadap data yang berhubungan dengan *World Wide Web* (WWW) [3]. *Web mining* melakukan pengolahan data web untuk mendapatkan knowledge sehingga dapat diaplikasikan dalam menangani permasalahan di dunia nyata.

Web mining dapat dikategorikan dalam tiga kelompok kelas berdasarkan bagian yang di-*mining* dari sebuah web. Ketiga kategori tersebut adalah *Web content mining*, *Web structure mining*, dan *Web usage mining*

2.2 Web Usage Mining

Web usage mining merupakan proses untuk mengaplikasikan teknik web mining dalam melakukan analisa terhadap pola penggunaan (*usage pattern*) dan analisa aktivitas pengunjung yang tersimpan dalam *web server log* [2]. Tipe data yang akan diolah berupa data *web server log* dalam web. Tujuan dari web usage mining adalah menemukan dan memprediksi tingkah laku user, agar dapat membantu *developer* untuk mengembangkan website, menarik pengunjung atau untuk mengelompokkan user berdasarkan kebiasaan pengaksesannya

2.3 Web Access Pattern Mining

Web access Pattern mining atau disebut juga Web usage mining dalam proses pelaksanaannya terdiri dari tiga langkah:

a. Pre-processing

Pre-processing merupakan tahap awal untuk melakukan pembersihan data sehingga mendapatkan inputan data yang berkualitas. Kegiatan yang dapat dilakukan yaitu:

- parsing data: memisahkan data log menjadi bagian-bagian tertentu yang lebih kecil. Data log dengan extension .log dibaca dan dipilih untuk ditempatkan ke dalam kolom-kolom yang sesuai dengan atribut format
- cleaning data: menghilangkan data-data yang tidak *relevant* digunakan saat proses pencarian pola dan analisis contohnya:
 - Menghilangkan *reference* tambahan yang ikut tersimpan ke dalam log seperti *reference* untuk *style file*, *graphics*, atau *sound file*.
 - Menghilangkan beberapa *data field*, seperti jumlah *byte* transfer data atau versi dari *HTTP protocol* yang digunakan, yang tidak mengandung informasi yang berguna untuk proses pencarian pola dan analisis [5].
- data transformation: mengubah dan menyimpan data kedalam penyimpanan data dengan format yang sesuai dengan kebutuhan data pada proses selanjutnya.

Berikut dapat digambarkan dengan diagram aktivitas dari proses preprocessingnya



Gambar 2-1 Aktivitas diagram *preprocessing data*

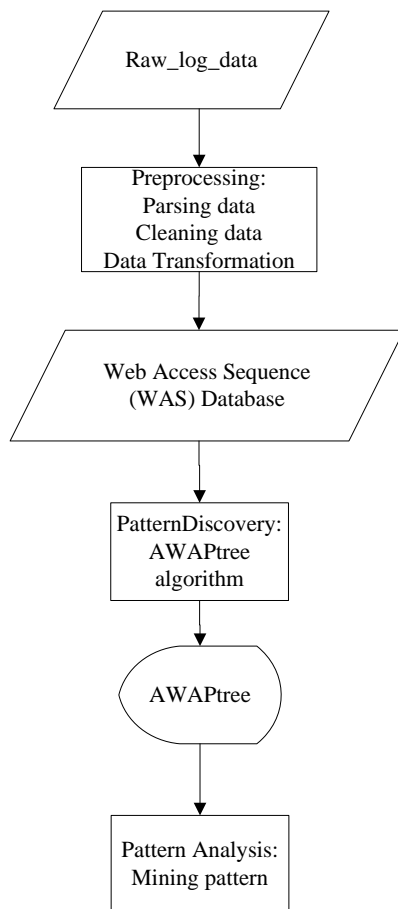
b. Pattern discovery

Pada tahap ini dilakukan proses pencarian pola akses user dengan menerapkan metode Frequent Sequential Pattern Mining yaitu Adaptive Web Access Pattern Tree.

c. Pattern analysis

Pada tahap pattern discovery ini kita melakukan proses analisis terhadap pola yang dihasilkan pada proses pattern discovery. Yaitu dengan melakukan proses mining terhadap pola pohon yang dihasilkan oleh algoritma Adaptive Web Access Pattern Tree untuk melihat bagaimana performansi atau kinerja pola yang dihasilkan.

Tahapan pada proses Web Usage Mining ini dapat digambarkan dalam flowchart seperti di gambar berikut



Gambar 2-2 Proses Web Usage Mining

Web usage mining bisa digunakan untuk banyak tujuan, misalnya dengan melihat sequence halaman dari yang diakses oleh seorang *user* maka *profile* dari *user* tersebut dapat dikembangkan, sehingga sangat membantu dalam proses *personalization*. Ada beberapa pengaplikasian dari Web usage mining seperti berikut:

- *Personalization* untuk seorang *user* dapat diperoleh dengan menelusuri halaman-halaman yang sebelumnya diakses. Halaman-halaman ini dapat digunakan untuk mengidentifikasi kebiasaan pengaksesan web dari seorang *user* kemudian juga mampu untuk memprediksi halaman-halaman yang diinginkan.
- Dengan menentukan kecenderungan pengaksesan yang paling sering dilakukan oleh *user*, kebutuhan *links* suatu halaman dapat diidentifikasi untuk meningkatkan keseluruhan performansi pengaksesan nantinya.

- Sebagai tambahan modifikasi terhadap struktur *link*, pengidentifikasian kecenderungan akses yang pada umumnya dilakukan, bisa digunakan untuk meningkatkan desain dari *Web page* dan untuk membuat modifikasi lain terhadap situs tersebut.
- Web usage mining bisa digunakan dalam business intelligence untuk meningkatkan penjualan dan promosi.

2.4 Frequent Sequential Pattern Mining

Sequential Pattern mining adalah salah satu bagian dalam data mining yang mencoba untuk menemukan pola yang paling sering dilakukan dari sekumpulan *sequential data*. *Sequential data* adalah data yang sangat mempertimbangkan urutan kejadian. Sequence Pattern Mining pertama kali dikenalkan oleh (Agrawal R., and Srikant R.(1994)) dengan penjelasan sebagai berikut *Sequence Pattern Mining* awalnya terdapat sebuah *sequential database* dimana tiap *sequence*-nya merupakan sebuah *list* transaksi yang terurut berdasarkan waktu pelaksanaannya dan tiap transaksi mengandung sekumpulan item [7].

Sequential Pattern mining pada dasarnya terdiri atas dua pendekatan[1]:

- Apriori-based mining algorithms

Algoritma AprioriAll (Pei J., Han J., Mortazaviasl B., dan Zhu Ha (2000)) menjelaskan bahwa terdapat tiga langkah untuk melakukan *mining* pada *sequential pattern* yaitu:

- a. Mencari semua *frequent itemset* yang ada. *Frequent itemset* adalah himpunan *item(event)* yang seringkali muncul, atau sering dijelaskan dengan *itemset* yang nilainya $support \geq minimum\ support$. *Support* adalah perbandingan kemunculan transaksi yang mengandung *item-item* tertentu atau persentase transaksi yang memiliki pola tertentu.
- b. Men-transformasi-kan database yang tiap transaksi awal kemudian diganti dengan semua *frequent itemset* yang terkandung dalam transaksi tersebut.
- c. Menemukan *sequencial pattern*-nya

Algoritma ini tidak bersifat *scalability* dikarenakan *cost* yang tinggi pada tahap *transformasi*.

- WAP Mining-based Algorithm

Algoritma WAP Tree merupakan kompresi struktur data yang efektif yang didisain untuk menyimpan data yang terkandung dalam web log. Langkah-langkah pengerjaannya:

- a. Melakukan *scanning* pada *web access sequence database*, mencari *frequent individual event*. *Frequent individual event* atau disebut juga *1-sequence frequent event* adalah *event* atau *item* yang paling sering diakses yang panjangnya 1 *event*.
- b. Melakukan *scanning database* lagi untuk membuat WAP tree dari *frequent subsequent*, transaksi yang hanya mengandung *1-sequence frequent event*, seperti pembangunan *Frequent Pattern tree (FP-Tree)* pada data *non-sequential*, yaitu pembangunan pohonnya secara *prefix* dan disaat yang sama semua *node* yang mengandung *frequent event* yang sama akan terhubung ke sebuah *event queue* dan pada *Header table* tercatat *head* dari tiap-tiap *queue* yang terbentuk

2.5 Algoritma Adaptive Web Access Pattern Tree

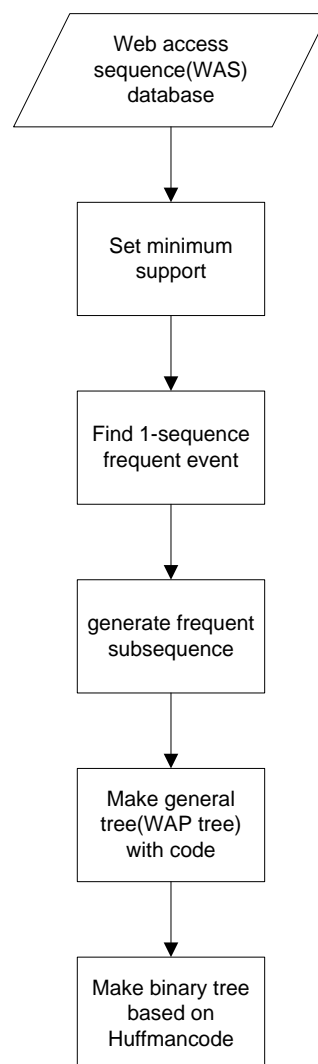
Tahapan pada proses pembangunan algoritma Adaptive Web Access Pattern Tree ini dapat digambarkan dalam flowchart seperti di gambar 2-3

Algoritma AWAP Tree pada dasarnya hampir sama dengan algoritma WAP Tree. Web Access Pattern Tree (WAP Tree) mengalami penambahan teknik pembentukan tree menjadi Adaptive Web Access Pattern Tree (AWAP Tree) dengan mengadaptasi teknik pengkodean binary *Huffman code* dalam pembentukan *tree*-nya. Adapun langkah-langkah AWAP Tree seperti berikut:

- a. Menginputkan *minimum support* sebagai batas atau *threshold* pemilihan jumlah *1-sequence frequent event*.
- b. Menemukan *1-sequence frequent event*
- c. Dari *1-sequence frequent event* yang ditemukan scan database sekali lagi untuk mendapatkan *frequent subsequent*.
- d. Buat *general tree* dengan metode *prefix tree* serta *queues* dan *header tablenya*. Sejauh ini pembuatan AWAP tree sama dengan pembuatan WAP tree. Tetapi untuk AWAP tree, *general tree* tersebut diberi *binary code* dengan metode *Huffman code* seperti berikut:
 - *Current node* = Root, dimana root kodenya 0
 - Anak paling kiri dari *Current node* ditambahkan bit 1 di belakang

kode dari *Current node*, sedangkan untuk sibling ke arah kanan dari anak paling kiri *Current node*, akan ditambahkan *bit 0* setelah kode anak paling kiri *Current node* sebanyak *n* buah dari *sibling* terdekat.

- Update current node sebagai anak paling kiri
 - Begitu seterusnya
- e. Buat binary tree dengan menyusun node pada general tree sesuai aturan huffman



Gambar 2-3 Proses Adaptive Web Access Pattern Tree

Perbedaan yang mendasar pada teknik Huffman code yang digunakan dalam algoritma ini dibandingkan teknik Huffman code yang pada umumnya, adalah dibagian penentuan *bit-bit* binary codenya, untuk lebih jelasnya terlihat pada tabel dibawah ini

Tabel 2-1 Perbedaan metode Huffman

Metode Huffman biasa	Metode Huffman AWAP Tree
Pembentukan <i>code</i> berdasarkan frekuensi event yang diurutkan, semakin besar frekuensi, akan semakin pendek kode huffmannya	Pembentukan <i>code</i> berdasarkan posisi suatu node dalam <i>general tree</i> -nya. Karena dalam AWAPTree, urutan sequensial suatu node merupakan hal yang sangat penting.

3. Analisis dan Perancangan Sistem

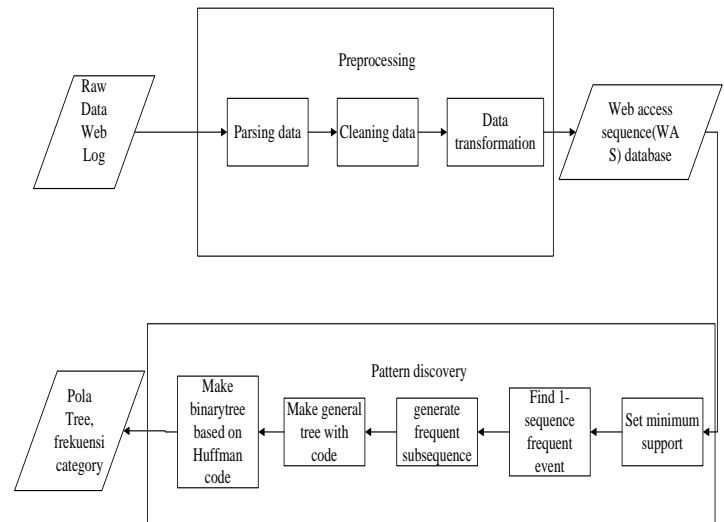
Secara umum, sistem yang akan dibangun meliputi beberapa fungsionalitas untuk melihat kecenderungan pola pengaksesan halaman web <http://www.ittelkom.ac.id/id> menggunakan metode *Adaptive Web Access Pattern Tree*. Sistem ini meliputi proses *data preprocessing* dan *pattern discovery*. Dimana pada *data preprocessing* dilakukan tiga tahapan yaitu *parsing data*, *cleaning data*, serta *data transformation* sedangkan pada tahap *pattern discovery* dilakukan pembentukan pola pengaksesan user dengan AWAP Tree.

Raw log file yang didapat dari web server ittelkom diolah terlebih dahulu untuk mendapatkan data-data yang relevan untuk nantinya digunakan sebagai inputan pada proses pencarian pola pengaksesan web IT Telkom.

Pada tahap awal yaitu *data preprocessing* dilakukan beberapa proses preprocessing yang dianggap perlu dilakukan terhadap data web log, yang masih berupa *raw log file*. Pada tahap *pattern discovery* akan dilakukan proses pencarian pola pengaksesan halaman yang paling sering diakses. Dalam pencarian pola tersebut menggunakan disiplin ilmu Frequent Pattern Mining. Lalu baru dilakukan tahap berikutnya analisis terhadap pola akses user dalam tahapan Pattern Analysis.

Pada proses algoritma AWAP Tree terdapat lima langkah pengerjaan. Pertama *set minimum support*, disini *user* diminta untuk menginputkan minimum support dalam rentang nilai $0 < \text{minimum support} < 1$. Langkah berikutnya adalah dengan mencari 1-sequence frequent event. 1-sequence frequent event dicari dengan cara mengurutkan event yang panjangnya 1 sequence dari yang nilai supportnya paling besar, lalu diambil event yang memiliki nilai $\text{support} \geq \text{minimum support}$. Nilai support suatu event adalah banyaknya transaksi yang mengandung suatu event dibandingkan dengan seluruh transaksi yang ada. Jika dalam sebuah transaksi mengandung event yang dicari maka dihitung satu transaksi, walaupun kemunculan event tersebut lebih dari satu kali. Langkah ketiga yaitu *men-generate frequent subsequent*, pada langkah ini akan dipilih transaksi yang paling sering diakses (*frequent subsequent*) dengan memilih hanya transaksi yang mengandung 1-sequence frequent

event dalam sequencenya. Langkah keempat membuat *general tree* dari transaksi *frequent subsequent* serta memberi label *binary code*. Langkah terakhir yaitu dengan membuat konversi *general tree* dalam bentuk *binary tree*-nya sesuai dengan aturan *Huffman code*.



Gambar 3-1 Deskripsi Proses Sistem

4. Implementasi dan Pengujian

Dataset yang digunakan berupa data *raw log* web access server aplikasi website Institut Teknologi Telkom dengan alamat URL nya <http://www.ittelkom.ac.id/id> namun sekarang telah bisa diakses dengan alamat URL <http://www.ittelkom.ac.id>. *Raw log* digunakan sebagai inputan tahapan *Preprocessing* yang kemudian akan diubah ke menjadi data yang bertipe *discrete sequence* sebagai inputan pada tahapan pencarian pola. Web log ini memiliki format log bertipe NCSA combined log format yang memiliki atribut IP address, waktu, request, status, size, referrer, dan *user_agent*. Web log yang digunakan merupakan web log periode mei 2011-juni 2011 yang telah dipartisi dalam beberapa ukuran dataset.

4.1 Skenario Pengujian pada *Preprocessing data*

Pada tahap ini sistem akan meload file *web server log* dari *database file* di *harddisk* dan kemudian melakukan pemilahan data yang akan digunakan untuk tahap berikutnya. Parsing dilakukan dengan mengecek pada tiap baris yang dibagi kedalam kolom-kolom atribut yang formatnya sama dimana baris yang hanya akan digunakan adalah baris yang merequest file halaman web yang hanya mengandung `"/id/index.php?categoryid= "` dalam URLnya. Proses *Cleaning* dilakukan terhadap baris dan atribut yang tidak relevan sehingga yang tersisa adalah atribut-atribut yang dibutuhkan untuk proses pencarian pola

Setelah itu dilakukan proses transformasi untuk mengubah *cleaned log* tersebut menjadi data yang bertipe *discrete sequence* yang merupakan inputan pada proses pencarian pola. maka bentuk Web Access Sequence Database (WAS Database) dapat dilihat pada Tabel 4-1

Tabel 4-1 Contoh Web Access Sequence Database (WAS Database)

No.	IP	Sequence Category Halaman
1	67.195.110.167	22 12 27 44
2	180.178.106.233	44 33 27
3	118.96.255.186	22 12 33

4.1 Skenario Pengujian pada Pencarian Pola dengan AWAPT

Pada proses pencarian pola kecenderungan akses user dalam mengakses halaman-halaman website dibutuhkan inputan berupa data sekuensial yang tersimpan dalam *Web Access Sequence Database* (WAS Database). Hasil dari algoritma berupa pohon kecenderungan akses user yang tiap nodenya merupakan sebuah item yang telah diberi *binary code*. Pengujian dilakukan dengan dua skenario pengujian, yaitu:

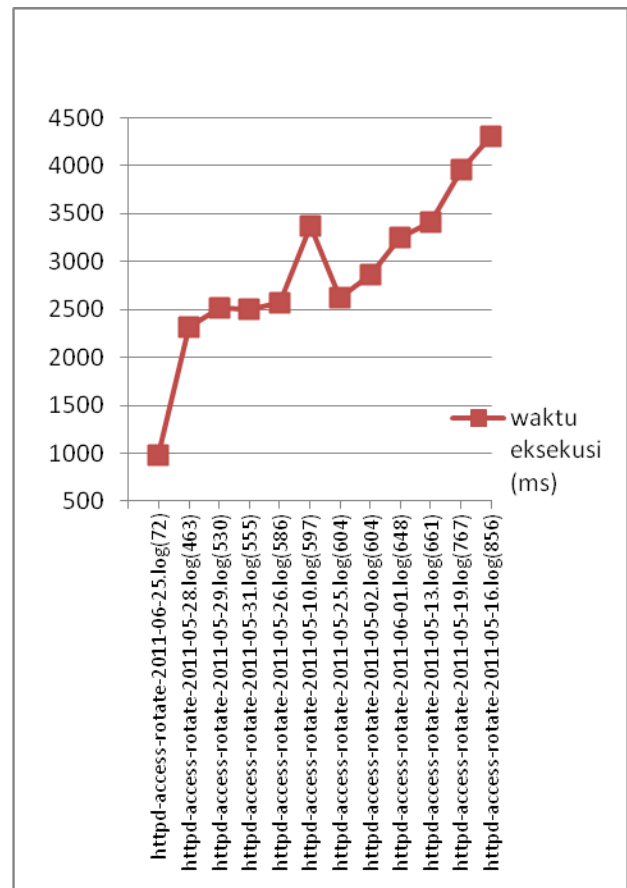
- Skenario pertama, dilakukan untuk melihat pengaruh jumlah transaksi *user* terhadap waktu eksekusi. Pengujian dilakukan terhadap 12 dataset yang berbeda tetapi dengan ukuran dataset yang sama (100 MegaByte) dengan inputan *minimum support* yang sama yaitu 0.1.

Skenario pengujian kedua, dilakukan untuk melihat pengaruh nilai *minimum support* terhadap performansi sistem. Pengujian dilakukan dengan mengubah nilai *minimum support* 0.05, 0.1 , 0.15 ,dan 0.2 dan melihat pengaruhnya terhadap performansi sistem yang dilihat dari pola sekuensial yang dihasilkan.

4.2 Analisis Hasil Pengujian Jumlah Transaksi dengan Waktu

Pengujian ini dilakukan untuk melihat pengaruh jumlah transaksi terhadap waktu eksekusi, maka digunakan dataset yang ukurannya sama tetapi memiliki jumlah transaksi yang berbeda untuk melihat pengaruhnya pada waktu eksekusi algoritma AWAPTTree.

Dari grafik pada Gambar 4-1 maka pada umumnya dapat dilihat bahwa semakin besar jumlah transaksi suatu dataset maka waktu eksekusi algoritma yang dibutuhkan akan semakin lama.



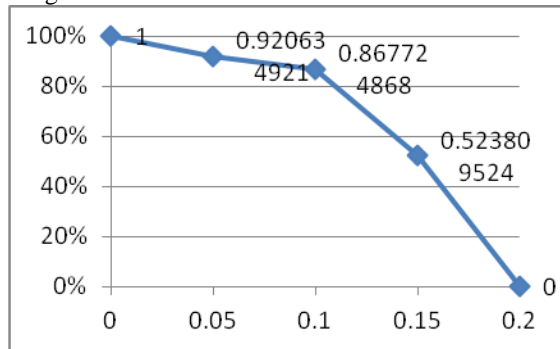
Gambar 4-1 Hasil Pengujian terhadap Jumlah Transaksi

Waktu eksekusi pada hasil pengujian akan semakin besar seiring bertambahnya jumlah transaksi yang diuji namun terjadi sebaliknya pada dataset “httpd-access-rotate-2011-05-10.log” dan dataset “httpd-access-rotate-2011-05-25.log” ,yang jumlah transaksinya semakin besar, menghasilkan waktu eksekusi yang semakin kecil. Hal ini dapat dikarenakan pengaruh “jumlah 1-seq frequent item” yang terbentuk pada saat eksekusi algoritma AWAPTTree. Saat proses algoritma AWAPTTree terjadi besarnya “jumlah 1-seq frequent item” yang terbentuk akan mempengaruhi pembangunan pohonnya. Jadi dapat dilihat pada Tabel 4-1 untuk jumlah transaksi yang sama pada dataset “httpd-access-rotate-2011-05-25.log” dan “httpd-access-rotate-2011-05-02.log” , 604 transaksi, dengan “jumlah 1-seq frequent item” yang semakin besar akan menghasilkan waktu eksekusi yang semakin besar pula.

4.3 Analisis Hasil Pengujian Pengaruh Nilai Minimum Support terhadap Performansi sistem

Pengujian ini dilakukan untuk melihat pengaruh nilai *minimum support* terhadap performansi sistem dilihat dari pola sekuensial yang dihasilkan, maka digunakan *minimum support* yang berbeda-beda nilainya untuk melihat

pengaruhnya pada pola sekuensial pada data dengan jumlah transaksi yang sama yaitu 378 transaksi dengan total event 73.



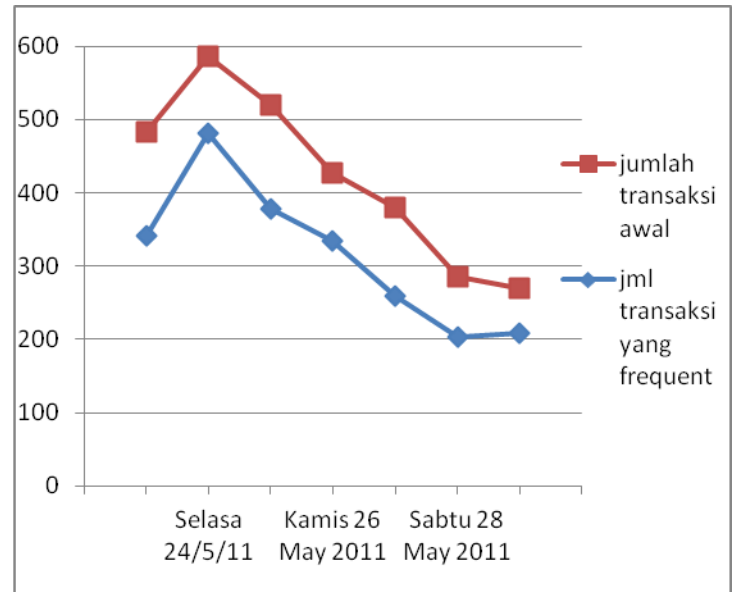
Gambar 4-2 Grafik Perbandingan nilai *minimum support* terhadap performansi sistem

Dari grafik diatas maka dapat disimpulkan bahwa semakin besar nilai *minimum support* maka pola, yang paling sering diakses, yang dihasilkan akan semakin sedikit yang nantinya juga berpengaruh pada waktu eksekusi juga akan semakin sedikit. Pengujian tidak menghasilkan hasil yang optimal pada nilai *minimum support* 0,2 – 1 karena tidak terdapat satupun event yang memiliki nilai $support \geq minimum\ support$ tersebut. Sedangkan untuk *range* nilai *minimum support* digunakan 0,05 karena setelah melakukan pengujian untuk rentang nilai yang lebih kecil akan menghasilkan pola yang sama.pada grafik tersebut.

4.4 Analisis Hasil Pengujian Pengaruh Waktu pada Performansi Sistem

Analisis ini dilakukan untuk melihat pengaruh waktu pada performansi atau kinerja sistem. Pengujian pada waktu yang dimaksudkan disini adalah melihat hasil eksekusi algoritma pada dataset yang telah dipartisi perhari selama seminggu dimulai dari tanggal 23 May 2011 sampai 29 May 2011. Pengujian dilakukan untuk melihat prosentase transaksi sekuensial yang frequent dari keseluruhan transaksi awal.

Grafik tersebut menampilkan besar transaksi sekuensial yang frequent dimana transaksi sekuensial yang frequent tersebut merupakan pola hasil keluaran sistem. Terlihat bahwa pola yang dihasilkan tiap harinya berbeda-beda, tetapi pada hari kerja Senin – Jumat pola transaksi pengaksesan lebih banyak dibandingkan non hari kerja (*weekend*).

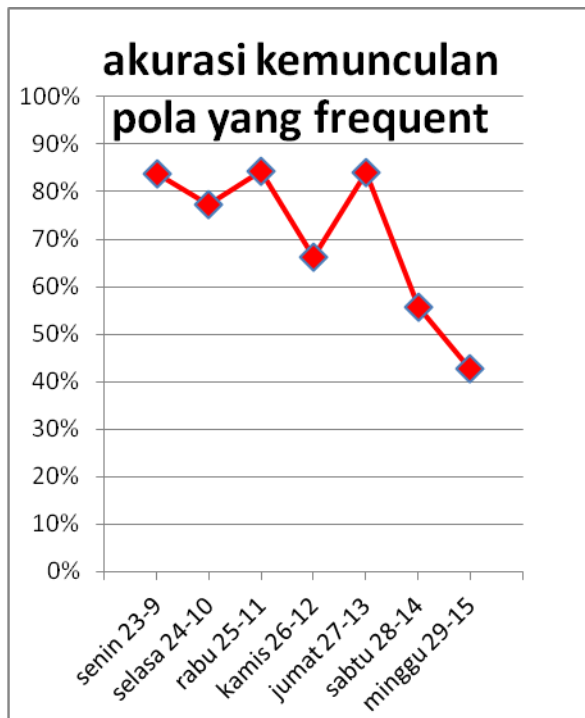


Gambar 4-3 jumlah transaksi sekuensial yang frequent

4.5 Analisis Akurasi Kemunculan Pola yang dihasilkan

Analisis dilakukan untuk melihat seberapa besar akurasi pola-pola transaksi sekuensial yang dihasilkan. Pengujian dilakukan pada dua jenis data, pertama dataset selama satu minggu yaitu Senin tanggal 9 May 2011 sampai dengan Minggu 15 May 2011, data tersebut akan digunakan sebagai *data training* untuk membangun model dari pola yang dihasilkan. Sedangkan untuk *data testing*nya digunakan dataset selama satu minggu yaitu Senin tanggal 23 May 2011 sampai dengan Minggu 30 May 2011. Lalu akan dilihat akurasi kemunculan pola pada *data testing* sesuai dengan model yang telah dibangun oleh *data training*. Akurasi dihitung dengan melihat seberapa banyak transaksi pada data testing yang terprediksi benar,sesuai dengan model yang dibangun, dari keseluruhan transaksinya

Dari grafik terlihat akurasi kemunculan pola frequent yang dihasilkan oleh data testing sehingga terlihat bahwa akurasi pola pengaksesan pada hari kerja Senin – Jumat lebih besar dibandingkan non hari kerja Sabtu- Minggu (*weekend*).



Gambar 4-4 akurasi kemunculan pola yang frequent

4.6 Analisis Keterkaitan antar item melalui binary code

Analisis dilakukan dengan mengecek apakah suatu item memiliki keterkaitan dengan item lain. Maksud dari keterkaitan antar item ini adalah apakah suatu item termasuk kedalam sekuens pola akses halaman web yang sering dikunjungi. Dengan melihat binary code suatu item bisa dibandingkan binary code tersebut dengan binary code item lain dan dapat menyatakan bahwa item tersebut bagian dari sebuah sekuens pola akses, baik itu berupa *ancestornya* ataupun *descendantnya*. Untuk mengecek hal tersebut ditentukan beberapa node dengan binary codenya, lalu dibandingkan binary codenya.

Misal diambil beberapa node hasil eksekusi algoritma dengan binary codenya seperti yang terlihat pada Tabel 4-6

Tabel 4-9 Contoh Node dengan posisi Binary Codenya

No.	Node	Binary Code
1	27	101
2	12	1010
3	1	101010

Pengecekan dilakukan dengan melihat teknik pengkodean huffman yang digunakan pada algoritma ini untuk menentukan binary code yaitu:

$$dcnode = ac + 1 + h$$

(1)

Dapat diketahui keterangan dari Gambar 4-3 yaitu :

dcnode : descendant code node
ac : ancestor code node
h : bit-bit ekstra atau tambahan lain

Tabel 4-10 Tabel bit binary code 1

Node	Binary code					
	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
27	1	0	1			
12	1	0	1	0		
1	1	0	1	0	1	0

Kotak yang ditandai biru adalah *AncestorCode(ac)* yang sama tapi node 12 bukan *Descendant node* dari node 27 karena bit keempat dari node 12 bukanlah bit 1 dan tidak memenuhi persamaan (1).

Tabel 4-11 Tabel bit binary code 2

Node	Binary code					
	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
27	1	0	1			
12	1	0	1	0		
1	1	0	1	0	1	0

Sedangkan untuk node 12 dan 1 merupakan node yang sama-sama berada pada sebuah *frequent access sequence* karena memenuhi persamaan pada gambar 4-3. Dengan *ancestornya* node 12 dan *descendantnya* node 1. Dengan *Frequent sequential Patternnya*: 12 → 1.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis terkait dengan penelitian Tugas Akhir yang telah dilakukan, diperoleh beberapa kesimpulan antara lain :

1. AWAP Tree dibangun dari sekuens halaman yang paling sering diakses dengan membentuk general tree, yang mempunyai *binary code*, lalu dikonversi ke binary tree sesuai aturan *Huffman coding tree*
2. Nilai parameter nilai *minimum support* yang optimal adalah $0 < \text{minimum support} < 0,2$. Pengaruh nilai *minimum support* pada performansi sistem dilihat dari pola yang dihasilkan adalah semakin besar nilai *minimum support* maka semakin sedikit pola yang dihasilkan. Begitu juga sebaliknya.
3. Parameter jumlah transaksi yang semakin kecil akan semakin cepat waktu eksekusinya. Tetapi terdapat faktor lain yang mempengaruhi perbandingan jumlah transaksi dataset terhadap waktu, yakni jumlah 1-sequence frequent item yang

- dihasilkan. Untuk besarnya jumlah transaksi yang sama dari dataset yang berbeda maka jika jumlah 1-sequence frequent item semakin besar maka waktu eksrksi akan semakin besar, beitu juga sebaliknya.
4. Pengaruh waktu penggunaan data, dalam hal ini dipartisi perhari selama seminggu pola yang dihasilkan tiap harinya berbeda-beda, tetapi pada hari kerja Senin – Jumat pola transaksi pengaksesan lebih banyak dibandingkan non hari kerja (*weekend*).
 5. Akurasi pola pengaksesan pada hari kerja Senin – Jumat lebih besar dibandingkan non hari kerja Sabtu- Minggu (*weekend*).
 6. Hubungan keterkaitan antar item dapat diketahui dengan melihat *Huffman codenya*.
- [11] Vijayalakshmi. S, V.Mohan and S.Suresh Raja, 2010, *Mining of Users' access behaviour for frequent sequential pattern from web logs*, International Journal of Database Management Systems (IJDMS) Vol.2, No.3, August 2010
 - [12] Zhou. Baoyao, Siu Cheung Hui, Kuiyu Chang, *An Intelligent Recommender System using Sequential Web Access Patterns*, Singapore

Daftar Pustaka:

- [1] Gomathi. C.M. Moorthi, K.Duraiswamy, 2008, *Binary Coded Web Access Pattern Tree in Education Domain*. www.cccsenet.org/journal.html Vol.1,No.4 , November 2008
- [2] Gomathi. C.M. Moorthi, K.Duraiswamy, 2008, *Web Access Pattern Algorithms in Education Domain*.
- [3] <http://swa.co.id/2010/10/penggunaan-mobile-web-terus-meningkat-setiap-bulan/> diakses pada tanggal 28 September 2010
- [4] Iváncsy Renáta, István Vajk, 2006, *Frequent Pattern Mining in Web Log Data*. Acta Polytechnica Hungarica Vol.3, No.1, 2006.
- [5] Margaret H. Dunham, 2003, *Data Mining Introductory and Advanced Topics*, New Jersey , Prentice Hall.
- [6] Mobasher Bamshad, 2008, *Chapter 12: Web Usage Mining*
- [7] Ningtyas. Dian Kusuma, Farah Virnawati, Prasetyo, Tirta Paramitta, I Wayan Simri, 2008, Analisis Perilaku pengguna sistem e-learning universitas gunadarma, Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008).
- [8] Paramita. Ayudya ,2009, Analisis pola kunjungan pengguna situs web IPB menggunakan algoritma Totally Fuzzy, Bogor, IPB
- [9] Sathiyamoorthi. V., V. Murali Bhaskaran, 2009, *Data Preparation Techniques for Web Usage Mining in World Wide Web-An Approach*, International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009 , Academy Publisher
- [10] Vasumathi. D, A. Govardhan, 2009, *BC-WASPT Web Access Sequential Pattern Tree Mining*, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009