Sistem Rekomendasi Penambahan Link pada Website berdasarkan Data Log Website

Kurniawan Aji Saputra, Annisa

Departemen Ilmu Komputer FMIPA-IPB, Bogor, West Java, 16680, Indonesia

***Abstract*---A good website structure is needed by the website visitors. The website structure which is created by the website developer should be in accordance with user preferences. The easier the user can find the target page, the better the website structure is. However if the user still get some difficulties in** *finding the location of the target page, website developer must change the website structure. Data logs from the website can help developers to improve the structure of the website. Data log should pass through the process of data preprocessing which are data cleaning, user identification, session identification, path completion and transaction identification. After that, expectation location will be searched in the data log by using Find Expectation Location algorithm. Location expectations are then processed using the optimization algorithm (FirstOnly and OptimizeBenefit) to get the recommendation of adding a page link. Finally the improved structure will be displayed using a scalable vector graphic (SVG).*
Keyword : website structure, find expectation location algorithm, data preprocessing, visualitation, scalable vector graphic (svg).

I. PENDAHULUAN

A. Latar Belakang

Website yang baik adalah yang mampu menyediakan layanan yang baik kepada setiap pengunjungnya. Pengunjung akan tetap bertahan pada *website* tersebut sampai sesuatu yang mereka inginkan sudah berhasil didapatkan. Pembangunan sebuah *website* yang baik harus memperhatikan tujuan dari pengembangan *website* tersebut. Layanan apa saja yang nantinya akan diberikan kepada pengunjungnya. Sedapat mungkin layanan yang ditujukan sebagai layanan utama ini diletakkan di halaman yang mudah ditemukan. Jika tidak ditampilkan pada halaman utama maka sebaiknya diberikan jalan pintas ataupun *hyperlink* ke halaman tujuan tersebut.

Pada penelitian Nurdian Setyawan (2008) yang berjudul Rekomendasi Penambahan *Link* pada Web Berdasarkan Pola Akses Pengguna sudah berhasil mengimplementasikan algoritme yang tepat untuk menentukan halaman target yang pengguna cari. Namun hasil penelitian tersebut masih berupa tabel-tabel yang sulit dimengerti oleh pengembang ataupun pengguna biasa. Penelitian ini akan memvisualisasikan hasil rekomendasi penambahan *hyperlink* tersebut dengan menggunakan *website* yang sederhana sehingga mudah dimengerti oleh pengguna maupun pengembang *website*.

Data yang digunakan pada penelitian sebelumnya masih menggunakan data *dummy*, maka pada penelitian ini akan menggunakan data *log* asli *website* yang sebenarnya. Dengan demikian, sistem hasil penelitian ini diharapkan dapat dimanfaatkan oleh pengembang *website* untuk dapat memperbaiki struktur *web*-nya agar sesuai dengan harapan pengguna yaitu dengan menambahkan *hyperlink* dari lokasi harapan ke lokasi target.

B. Tujuan

Tujuan dari penelitian ini adalah membuat halaman rekomendasi *website* yang diterapkan ke dalam sebuah *website artificial* dengan menggunakan data *log* asli *website*.

C. Ruang Lingkup

Penelitian ini dibatasi pada data *path* halaman dari sebuah *website* yang dikunjungi oleh pengguna yang diperoleh dari data *log* asli *website* yang menggunakan a*pache-server*.

D. Manfaat

Simulasi hasil rekomendasi penambahan *hyperlink* pada *website* berdasarkan data *log website* ini diharapkan mampu membantu para pengembang *website* untuk membuat struktur *web* yang baik sehingga dapat meningkatkan kepuasan terhadap pengguna *website* itu sendiri.

II. TINJAUAN PUSTAKA

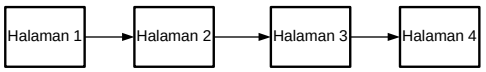
A. Struktur *Website*

Struktur *website* perlu diorganisasikan dengan baik agar pengunjung puas terhadap layanan yang diberikan dan memudahkan dalam pencarian informasi yang diperlukan. Struktur *website* tergantung dari tujuan pengembangannya seperti berikut ini (*Veer et al*. 2004)

a. Struktur Berurut

Pada halaman utama hanya terdapat *hyperlink* ke halaman berikutnya dan untuk halaman yang lainnya terdapat *hyperlink* ke halaman sebelumnya, *hyperlink* ke halaman selanjutnya dan *hyperlink* ke halaman utama. Bentuk struktur berurut seperti pada Gambar 1

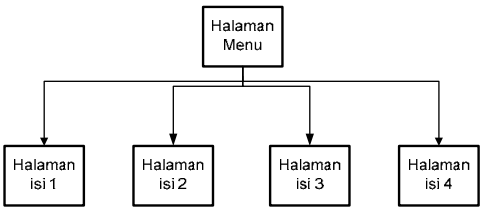
.



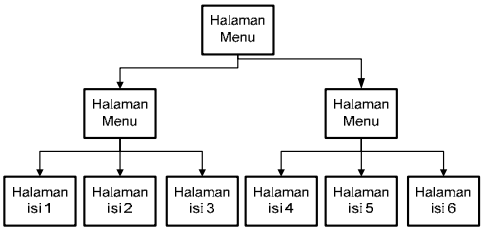
Gambar 1. Struktur Berurut (Veer *et al*. 2004).

b. Struktur Bertingkat

Struktur ini yang paling banyak digunakan oleh para pengembang *website* sekarang ini. Halaman-halaman dibagi menjadi halaman menu dan halaman isi sesuai dengan kategorinya. Tingkatan struktur ini bisa 2 (tingkat) seperti pada Gambar 2 atau lebih seperti pada Gambar 3.



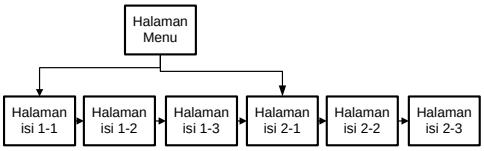
Gambar 2. Struktur Bertingkat dua (Veer *et al.* 2004)



Gambar 3. Struktur Bertingkat tiga (Veer *et al.* 2004).

c. Struktur Kombinasi Berurut dan Bertingkat

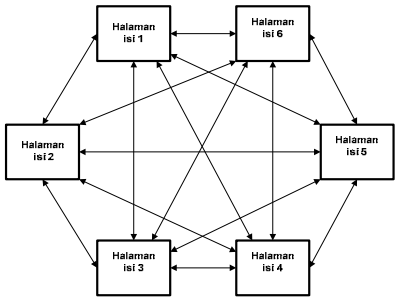
Struktur ini perpaduan atau gabungan antara struktur berurut dengan struktur bertingkat. Perpaduan struktur ini dapat dilihat pada Gambar 4 berikut ini.



Gambar 4. Struktur Kombinasi Berurut dan Bertingkat (Veer *et al.* 2004)

d. Struktur Jaringan

Pada struktur ini *hyperlink* suatu halaman tidak teratur penempatannya dan setiap halaman pasti terdapat *hyperlink* dengan halaman lainnya. Struktur jaringan ini menyulitkan pengembang *website* untuk menelusuri tingkah laku pengunjung. Struktur ini dapat dilihat seperti pada Gambar 5 berikut ini.



Gambar 5 Struktur Jaringan (Veer *et al.* 2004)

B. Data *Log*

Data *log* adalah data yang mencatat setiap *request* halaman *website* oleh pengguna saat membuka suatu *website* tertentu. Data *log* ini dihasilkan oleh *website* atau *server proxy* berupa berkas untuk masing-masing transaksi HTTP. Biasanya nama *file* data *log* ini adalah *access.log*. Data log ini terdiri dari tiga jenis yaitu data *log* yang tersimpan di *apache-server*, *client* , dan *server proxy*. Format data *log* ini berbeda-beda untuk setiap jenis informasi yang disimpan di dalamnya. Perbedaan ini biasanya tergantung pada *web server* yang digunakan (Ivancy & Vajk 2006).

Sebagai contoh berikut ini adalah format yang lebih dikenal dengan CLF (*Common Log Format*) data *log* yang menggunakan *Apache-server* :

114.4.12.15 - - [29/Jul/2009:15:05:58 +0700] "GET /index.php?page=detail&id=25 HTTP/1.0" 200 12493

Dari data *log* tersebut ada beberapa informasi yang bisa didapatkan. Informasi yang pertama adalah IP pengguna yang melakukan *request* terhadap halaman *website* tersebut yaitu 114.4.12.15. Kemudian setelah itu dapat terlihat informasi yang diberikan adalah mengenai waktu pengguna melakukan *request* yaitu tanggal 29 Juli tahun 2009 pada pukul 15:05:58. GET menunjukkan metode *request*, */index.php?page=detail&id25* menunjukkan halaman yang diminta, sedangkan *HTTP/1.0* menunjukkan protokol yang digunakan. 200 menunjukkan status *request* (200 berarti status setuju), 12493 adalah respon yang dikirimkan ke *client* dalam satuan *byte.*

C. *Data Preprocessing*

Beberapa yang dilakukan pada tahap *data preprocessing* ini dapat dilihat pada Gambar 6. Tahap-tahap yang dilakukan yaitu (Cooley *et al*. 1999):

1. *Data Cleaning*

Langkah pertama yang harus dilakukan pada data awal adalah *data cleaning*. Data mentah yang didapatkan masih banyak mengandung baris-baris yang tidak bersangkutan dengan *web mining* yang akan dilakukan. Baris-baris tersebut harus dihapus sebelum kita mengaplikasikan teknik untuk *mining* data.

Ketika pengguna meminta sebuah halaman maka yang diberikan oleh *server* tidak hanya halaman tersebut. Di dalam kode HTML diberikan juga gambar-gambar yang berkaitan dengan halaman yang diminta. Format dari gambar yang dihilangkan antara lain gif, jpeg, jpg, GIF, JPEG, JPG (Cooley *et al.* 1999). Misalkan di dalam satu halaman memuat 4 buah gambar, maka yang terjadi pada data *log* terdapat 5 baris di dalam data *log*. Baris-baris yang memuat gambar ini yang harus dihapus. Karena pada umumnya pengguna tidak meminta semua grafik pada satu halaman *website*. Grafik-grafik tersebut secara otomatis ter-*download* karena dicantumkan pada kode HTML halamannya. Format multimedia seperti *file-file* video tidak dihilangkan, karena ada kemungkinan *file-file* video ini tidak ter-*download* otomatis tetapi karena memang diminta oleh pengguna.

2. *User Identification*

Setelah tahap *data cleaning*, muncul masalah baru yaitu bagaimana kita menentukan halaman permintaan dari pengguna yang sama. Banyak metode yang dapat digunakan, namun kebanyakan membutuhkan tambahan informasi yang terkadang tidak ada di dalam data *log* Di dalam setiap baris data *log* terdapat alamat IP dan *browser* yang digunakan pengguna. Salah satu metode yaitu dengan menggunakan kombinasi alamat IP dan *browser* yang digunakan.

Asumsinya adalah semua baris dalam data *log* dengan alamat IP dan *browser* yang sama dilakukan oleh pengguna yang sama.

Akan tetapi kenyataannya alamat IP tidak selalu menjadi sebuah identitas yang unik untuk pengguna dan dalam data *log* CLF (*Common Log Format*) tidak tercatat *browser* yang digunakan pengguna. Kadang banyak pengguna saling berbagi satu alamat IP. Metode yang akan digunakan adalah jika pengguna-pengguna ini membuka *website* yang sama dengan satu alamat IP maka mereka dianggap satu pengguna yang sama.

3. *Session Identification*

Setelah data *log* dikelompokkan berdasarkan penggunanya, pada langkah ini data *log* akan dikelompokkan berdasarkan *session* kunjungannya. Hal ini tidak mudah untuk mencari kapan pengunjung meninggalkan *website* karena tidak ada informasi tersebut di dalam data *log.* Metode yang paling banyak digunakan untuk hal ini adalah metode yang berdasarkan pada *time-out.* Jika rentang waktu sangat lama diantara subbagian permintaan dari pengguna yang sama maka *session* baru telah dimulai. Pada penelitian Catledge dab Pitkow (1995), mereka menemukan *time-out* yang optimal adalah 25.5 menit dan menghasilkan waktu standar 30 menit.

4. *Path Completion*

Browser menyimpan halaman yang telah dikunjungi di dalam memori. Teknik ini disebut dengan *caching* dan tujuannya adalah mengurangi waktu respon ketika meminta sebuah berkas. Sebagai contoh, ketika pengguna memilih tombol *back* di *browser* maka versi *cached* dari halaman akan ditampilkan dan tidak ada permintaan baru yang dikirim ke *server* halaman. Hal ini menyebabkan *request* halaman yang dilakukan pengguna tidak tercatat di dalam data *log.* Tujuan *path completion* adalah untuk menambahkan *request* halaman yang tidak tercatat tersebut untuk setiap sesinya sehingga dihasilkan *path* perilaku pengguna yang lengkap.

5. *Transaction Identification*

Transaction Identification hanyalah sebuah langkah pilihan yang dapat dilakukan setelah langkah-langkah sebelumnya telah dilakukan. Contoh jika kita menganalisa sebuah *website* berita maka bentuk dari transaksi yang ditangkap oleh setiap *session* misalkan hanya halaman dengan berita olahraga. Setiap transaksi kemudian akan mengandung semua halaman dengan berita internasional dan yang lainya. Untuk *website e-commerce* bentuk traksaksinya mengandung halaman dengan informasi produk yang ditawarkan atau semua halaman yang berkaitan dengan pembayaran dan pesanan.

D. *Scalable Vector Graphic* (SVG)

Scalable Vector Graphic (SVG) merupakan format *file* baru untuk menampilkan grafik dalam pengembangan web yang berbasis XML (*eXtensible Markup Language*). *Image* SVG berekstensi svg yang hanya bisa dibaca oleh *browser* yang sudah mendukung *plugin* SVG atau kontrol *ActiveX*. Fungsi SVG untuk menampilkan grafik 2 dimensi dalam kode XML dan juga dapat mengkreasikan sebuah grafik yang terdiri dari banyak vektor yang berbeda-beda. Pada dasarnya, SVG dapat digunakan untuk membuat tiga jenis objek grafik, yaitu *path*, gambar dan teks. Kelebihan yang paling utama adalah image tidak akan kehilangan kualitasnya apabila diperbesar atau diperkecil(*scalable*), karena dibuat berdasarkan metode vektor bukan pixel seperti pada format grafik umumnya yaitu GIF, JPEG dan PNG. Sehingga memungkinkan pengembangan web dan juga *designer* untuk membuat grafik dengan mutu tinggi.

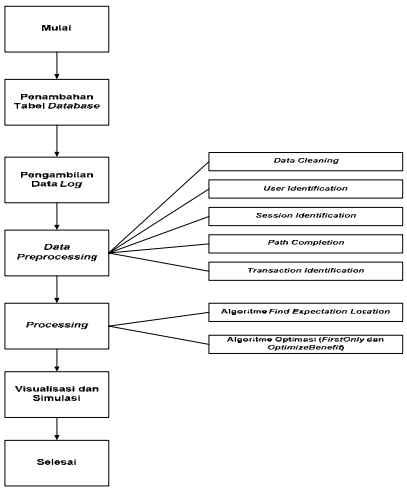
Struktur penulisan dokumen SVG mempunyai sedikit aturan sederhana. Aturan dasar yang paling penting adalah dokumen SVG dimulai dengan elemen <SVG> dan diakhiri dengan elemen </SVG>. Contoh :

```
<svg width='800px' height='450' viewBox='10 0 0 0'
  xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'>

…..

</svg>
```

SVG menetapkan enam bentuk dasar, termasuk juga *path* dan teks, yang dapat digabungkan untuk membentuk image yang mungkin. Setiap bentuk ini mempunyai properti yang menjelaskan posisi dan ukuran dari bentuk. Warna dan garis ditentukan oleh properti *fill* dan *stroke*.

III. METODE PENELITIAN

Penelitian ini dilaksanakan dalam beberapa tahap yaitu : (1) Penambahan tabel *database* (2) Pengambilan Data Log asli *Website* (3) *Data Preprocessing* (4) *Data Processing* dan (5) Visualisasi dan Simulasi Sistem. Tahap-tahap yang dilakukan pada penelitian diilustrasikan pada Gambar 6 di bawah ini.



Gambar 6. Metode Penelitian.

Pada gambar metode penelitian di atas tidak keseluruhan tahapnya dilakukan pada penelitian ini karena beberapa sudah dilakukan pada penelitian Nurdian Setyawan (2008). Tahap-tahap yang dilakukan pada penelitian ini adalah penambahan tabel *database*, pengambilan data *log*, *data preprocessing* (*data cleaning, user identification, session identification, path completion*), visualisasi dan simulasi sistem.

A. Penambahan Tabel *Database*

Tabel-tabel *database* yang ada pada penelitian sebelumnya hanya digunakan untuk tahap *data processing*. Sedangkan untuk tahap *data preprocessing* belum ada, sehingga perlu ditambahkan beberapa tabel *database* untuk *data preprocessing*. Tabel-tabel yang ditambahkan ini juga harus menyesuaikan dengan tabel-tabel yang sudah ada sehingga memudahkan dalam menghubungkan keterkaitan antara tahap *data preprocessing* dan tahap *data processing*.

B. Pengambilan Data *Log* asli *Website*

Pada tahap pengambilan data *log* ini dibagi menjadi 2 langkah yaitu proses *input* data dan penentuan halaman target dari *website*.

1. Proses *input* data

Pada penelitian sebelumnya masih menggunakan data *dummy* atau data yang dihasilkan oleh suatu algoritme. Pada penelitian kali ini data yang akan digunakan adalah data *log* asli *website* dari www.eramuslim.com yang merupakan data *log apache-server*.

Data ini pada awalnya berupa *file* berekstensi *log*. Karena ukuran *file* ini sangat besar dan tidak mampu dibuka oleh teks editor maka untuk memperoleh informasi yang dibutuhkan dalam data *log* tersebut dapat menggunakan cara *parsing* data menggunakan *script* bahasa pemrograman PHP. Tujuan lain dari penggunaan *parsing* data ini adalah untuk mengambil informasi yang dibutuhkan, tidak semua atribut yang ada di dalam data akan berguna dalam penelitian. Informasi yang diambil antara lain adalah alamat IP (*Internet Protocol Address*), waktu, dan *path* halaman *website* yang dibuka oleh pengunjung. Informasi yang yang diperlukan tersebut akan disimpan ke dalam *database* yang sudah dibuat.

2. Penentuan halaman target.

Pada sebagian besar *website* yang disebut dengan halaman target adalah halaman-halaman isi yang mengandung informasi yang dibutuhkan oleh para pengunjung *website*. Penentuan halaman target ini digunakan untuk menelusuri langkah-langkah yang dilakukan oleh pengunjung dalam mencari informasi yang dibutuhkan. Langkah yang dilakukan bisa saja langsung dapat menemukan halaman target, namun banyak langkah-langkah yang harus dilakukan pengunjung tersebut dan mungkin ada juga yang tidak dapat menemukan halaman target tersebut. Dalam penelitian ini untuk menentukan halaman menggunakan asumsi bahwa halaman target itu adalah *leaf* dari struktur *website* tersebut dan hanya diambil 15 halaman target yang paling banyak dikunjungi oleh pengunjung.

C. *Data Preprocessing*

Data yang diperoleh untuk penelitian ini masih berupa data mentah ataupun data yang belum terstruktur dengan jelas. Oleh sebab itu diperlukan *data preprocessing* agar data yang diperoleh dapat mendukung penelitian

1. *Data Cleaning*

Pada langkah pertama dalam *data praprocessing* ini data mentah yang sudah dimasukkan ke dalam tabel *database* melalui proses *parsing* kemudian akan diperiksa *path-path* halaman tersebut. *Path* halaman yang mengandung informasi selain halaman *website* akan dihapus. *Path* halaman yang dihapus diantaranya yang mengandung format gambar seperti jpg, jpeg, gif, JPG, JPEG, GIF (Cooley *et al.* 1999). dan format lain yang tidak mengandung informasi yang dibutuhkan (misalnya : css, exe, js).

2. *User Identification*

Pada tahap ini akan diidentifikasi pengguna berdasarkan alamat IP pengguna di dalam setiap baris data *log*. Dalam hal ini digunakan asumsi bahwa permintaan dengan alamat IP yang sama maka informasi berasal dari pengguna yang sama. Sebagai contoh jika banyak pengguna membagi satu alamat IP maka dalam hal ini akan memberikan indikasi bahwa hanya ada satu pengguna.

Kelemahan pada metode ini adalah jika banyak pengguna yang mengunjungi sebuah *website* dengan IP yang sama maka hanya dianggap hanya satu orang saja. Walaupun sebenarnya tingkah laku orang-orang tersebut berbeda-beda namun masih tetap dianggap hanya satu orang.

3. *Session Identification*

Informasi yang ada di dalam data *log website* terkadang berbeda-beda. Hal ini juga yang membuat karakteristik dari data *log* tersebut juga pasti akan berbeda. *Session* dapat berarti menunjukkan lamanya pengunjung berada di dalam *website* dan juga dapat berarti banyaknya kunjungan yang dilakukan pengunjung.

Pada penelitian ini untuk menentukan *session* dari setiap pengguna menggunakan asumsi bahwa setiap *session* dari pengguna berada pada rentang waktu 25,5 menit. Perhitungan rentang waktu dimulai saat pengguna melakukan *request* pertama kali pada *website* sampai waktu 25,5 menit. Apabila waktu lebih dari 25,5 menit maka kemudian akan dihitung sebagai *session* yang baru sehingga pengunjung akan mempunyai lebih dari satu *session*. Jika rentang waktu kurang dari 25.5 menit maka pengunjung tersebut hanya memiliki satu *session* pada kunjungannya tersebut.

4. *Path Completion*

Setiap pengunjung mempunyai tingkah laku yang berbeda di dalam mengunjungi sebuah *website* yang sama. Ada pengunjung yang hanya membuka satu halaman saja kemudian meninggalkan *website* tersebut. Namun ada juga yang membuka banyak halaman sampai menemukan halaman target yang dicarinya. Pada tahap ini akan diurutkan halaman yang diminta oleh pengunjung di dalam satu kali *session* kunjungannya. Sehingga akan terlihat halaman dan urutan pencarian dari pengunjung dalam mencari informasi di dalam *website* tersebut. Jika pengunjung sama namun *session* berbeda maka *path completion* yang dihasilkan akan berbeda pula.

5. *Transaction Identification*

Tahap selanjutnya adalah *transaction identification* yaitu akan dicari urutan halaman yang diminta oleh pengunjung sampai pengunjung tersebut menemukan halaman target yang sudah ditentukan pada tahap sebelumnya atau jika memang

tidak menemukan halaman target yaitu sampai dengan *path completion* berakhir. Proses yang dilakukan pada tahap ini hampir sama dengan *path completion*. Perbedaannya hanyalah pada *transaction identification* berganti setelah ditemukan halaman target pada proses *path completion*.

D. *Data Processing*

Setelah *data preprocessing* selesai kemudian data akan diproses agar mendapatkan hasil rekomendasi *hyperlink* pada halaman *website*. Pada tahap ini dilakukan 2 langkah yaitu pencarian lokasi harapan dan penentuan halaman rekomendasi.

1. Pencarian lokasi harapan

Lokasi harapan adalah lokasi atau *path* halaman yang ada di dalam *website* yang diminta oleh pengunjung sebelum menemukan halaman targetnya namun tidak mempunyai *hyperlink* dengan halaman target. Lokasi harapan yang dihasilkan dapat lebih dari satu karena perbedaan tingkah laku dari pengunjung tersebut. Pencarian lokasi harapan ini menggunakan algoritme *Find Expectation Locations* yang sudah digunakan pada penelitian sebelumnya yaitu oleh Nurdian Setyawan (2008). Pada penelitian ini hanyalah memasukkan data hasil *preprocessing* ke dalam algoritme tersebut. Algoritme ini membatasi jumlah lokasi harapan hanya 4 (empat) lokasi yaitu untuk lokasi pertama (E1), lokasi kedua (E2), lokasi ketiga (E3) dan lokasi keempat (E4).

2. Penentuan Halaman Rekomendasi

Pada penentuan halaman rekomendasi ini juga akan digunakan 2 (dua) buah algoritme optimasi yaitu *FirstOnly* dan *OptimizeBenefit* yang juga sudah digunakan pada penelitian sebelumnya oleh Nurdian Setyawan (2008). Hasil dari tahap ini adalah halaman-halaman mana saja yang perlu ditambahkan sebagai *hyperlink* ke dalam halaman targetnya. Pernambahan *hyperlink* ini disimpan dalam bentuk tabel *database mysql*.

E. Visualisasi

Proses visualisasi akan menggunakan gambar dengan format SVG (*Scalable Vector Graphic*) yang berbasis XML sehingga visualiasasi ini akan bersifat dinamis mengikuti data *log website* yang dimasukkan ke dalam sistem. Agar visualisasi ini mudah dipahami oleh pengunjung maka akan dibedakan struktur awal *website* dan halaman yang ditambahkan oleh sistem.

IV. HASIL DAN PEMBAHASAN

A. Penambahan Tabel *Database*

Pada penelitian yang menggunakan data *log* asli dari *website* ini membutuhkan penambahan tabel pada *database*. Namun ada beberapa tabel *database* yang digunakan pada penelitian sebelumnya oleh Nurdian Setyawan (2008) masih tetap akan digunakan pada penelitian kali ini. *Database* yang masih tetap akan digunakan antara lain : *tb_exp_locations, tb_hsl_firstonly* dan, *tb_hsl_optbenefit*. Penambahan tabel yang dilakukan dapat dilihat pada Tabel 1.

Tabel 1 Tabel *database* tambahan

Nama Tabel	Keterangan
data_cleaning	Tabel untuk menyimpan data *log* hasil tahap *data cleaning.*
session_identifcation	Tabel untuk menyimpan hasil tahap *session identification* yaitu sesi kunjungan dari masing-masing pengunjung.
path_completion	Tabel untuk menyimpan hasil dari proses *path completion.*
transaction_identification	Tabel untuk menyimpan hasil *transaction identification.*
path_url	Tabel untuk menyimpan jenis-jenis halaman yang diminta oleh pengunjung dan jumlah total kunjungan terhadap halaman tersebut
struktur_web	Tabel untuk menyimpan struktur awal *website* yang akan digunakan untuk visualisasi.

Dengan demikian total dari tabel *database* yang digunakan dalam pengembangan sistem ini adalah 9 (sembilan) buah tabel. Namun dari kesembilan tabel tersebut saling berdiri sendiri atau tidak mempunyai *relationship* antar tabel.

B. Pengambilan Data *Log* asli *Website*

1. Proses *input* data

Pada penelitian ini data *log* yang digunakan adalah data *log website* www.eramuslim.com. Data *log* ini merupakan data *log* hasil penyimpanan pada *apache-server* bulan januari 2009. Jumlah *record* yang ada pada data *log* ini adalah 112.219 baris. Jumlah data yang akan digunakan sebagai *record* di dalam ini dibatasi hanya 10.000 baris. Hal ini dikarenakan keterbatasan perangkat keras dalam pengembangan sehingga untuk data yang lebih besar dari itu akan membutuhkan waktu untuk *data processing* yang sangat lama. Selanjutnya dilakukan partisi sesuai kebutuhan yaitu 10.000 baris per *file* yang akan digunakan sebagai data *input.* Dengan demikian, rata-rata besarnya ukuran *file input* data

tersebut adalah 1MB. Untuk penelitian ini hanya akan diambil 5 *file* pertama hasil partisi.

Sistem untuk pertama kali akan meminta *user* untuk memasukkan *file* atau data tersebut untuk di-*upload* ke dalam sistem. Setelah proses *input* data berhasil kemudian akan dilakukan *parsing* terhadap data tersebut. Pada proses *parsing* ini hanya beberapa informasi saja yang akan diambil yaitu alamat IP, *path* atau halaman dan waktu.

2. Penentuan Halaman Target

Dalam penelitian ini untuk menentukan halaman menggunakan asumsi bahwa halaman target itu adalah *leaf* dari struktur *website* tersebut dan hanya diambil 15 halaman target yang paling banyak dikunjungi oleh pengunjung. Untuk itu harus dilakukan penghitungan terhadap kunjungan *website* untuk masing-masing halaman *website.*

Dari tahap ini dihasilkan 15 halaman yang akan digunakan sebagai halaman target. Dari hasil penentuan halaman target tersebut dapat dilihat dari 10.000 baris halaman yang paling banyak diminta oleh pengguna adalah "/berita/palestina/hamas-hancurkan-tujuh-tank-zionis.htm" dengan nilai kunjungan sebanyak 313 kali. Halaman yang paling banyak dikunjungi kedua adalah "/berita/palestina/israel-gencatan-senjata-sepihak-gagal-tumbangkan-hamas.htm" dengan kunjungan sebanyak 253 kali. Begitu juga untuk 15 halaman target yang lainnya.

C. *Data Preprocessing*

1. Data Cleaning

Untuk tahap *data cleanin*g ini sudah dilakukan secara sekaligus saat proses *parsing* data. Sehingga sampai tahap ini data sudah dianggap bersih dari *path* yang diminta pengunjung yang formatnya bukan format halaman.

2. User Identification

Alamat IP dari data *log* yang diproses sudah mengalami proses pengkodean / enkripsi, sehingga alamat IP yang ditunjukkan bukan alamat IP yang sebenarnya dari pengunjung. Hal ini untuk menjaga kerahasiaan dari pengunjung *website* itu sendiri. Untuk melakukan simulasi percobaan pada sistem ini digunakan 5 bagian dari data *log*. Untuk bagian 1 terdapat 5.748 pengunjung yang berbeda. Sedangkan untuk bagian 2 terdapat 5.719 pengunjung. 5.722 pengunjung untuk bagian 3, 5.599 pengunjung untuk bagian 4 dan untuk bagian 5 terdapat 5.835 pengunjung yang mempunyai alamat IP berbeda.

3. Session Identification

Tahap *session identification* dilakukan dengan cara menghitung. Satu pengunjung dapat memiliki lebih dari satu *session*, Namun satu *session* hanya dimiliki oleh satu pengunjung. Seperti pengguna dengan alamat IP 1022457008 mempunyai 2 *session* yaitu "1232266477" dan "1232266267". Sedangkan untuk *session* "1231140272" hanya dimiliki oleh pengguna dengan alamat IP 1023754728. Tidak ada selain pengguna tersebut yang mempunyai *session* "1231140272" walaupun pengguna tersebut mempunyai banyak *session*.

4. Path Completion

Urutan halaman yang diminta oleh pengunjung setiap *session*-nya berbeda-beda. Dari tabel *data_cleaning* kemudian dikelompokkan berdasarkan *session* dan urutan-urutan yang dihasilkan dipisahkan dengan menggunakan tanda koma (,) dan selanjutnya akan disimpan dalam tabel *path_completion*.

Dari hasil penelitian dapat dilihat bahwa pengguna dengan alamat IP 2085692385 dengan *session* "1231140062" hanya membuka 1 (satu) saja pada *website* ini yaitu halaman /berita/dakwah-mancanegara/amira-mayorga-yesus-bahkan-tidak-menyuruh-umatnya-untuk-menyembah-dirinya.htm". Sementara untuk pengguna dengan alamat IP 2107713143 dan *session* "1231140066" membuka lebih dari satu halaman. Halaman yang dibuka yaitu /berita/dunia/mahasiswa-indonesia-di-mesir-gelar-aksi-solidaritas.htm" kemudian pengguna melanjutkan kunjungannya dengan membuka "/manhaj-dakwah/hujatan-terhadap-dakwah-al-banna.htm" dan "berita/palestina/hamas-hancurkan-tujuh-tank-zionis.htm".

5. Transaction Identification

Data dari hasil *path completion* akan diproses pada tahap selanjutnya yaitu tahap *transaction identification*. Dari urutan-urutan tingkah laku pengunjung pada *path completion* akan dilihat halaman target pada urutan tersebut. Jika pengunjung sudah menemukan halaman target maka kemudian proses akan mulai dari awal namun proses pembacaan data hasil *path completion* tetap akan diteruskan.

Hasil dari tahap *transaction identification* ini selanjutnya akan disimpan ke dalam tabel *transaction_identification* yang nantinya akan menjadi data *input* untuk tahap *processing* menggunakan algoritme optimasi. Pengunjung tidak perlu bersusah payah untuk menemukan halaman target yang dicarinya. Bahkan sering kali pengunjung dapat langsung menemukan halaman target tersebut. Hal ini dapat sebagai indikator awal bahwa struktur dari *website* ini sudah cukup baik yaitu dengan menempatkan *hyperlink* ke halaman target pada halaman utama atau halaman pertama pengunjung mengunjungi *website* tersebut.

Pengguna dengan alamat IP 1009985486 membuka halaman "/berita/palestina/hari-ke-21-22-pertemuan-doha-dunia-arab-bekukan-hubungan-dengan-israel.htm". Halaman tersebut oleh sistem diidentifikasi sebagai halaman target. Maka pengguna ini sudah langsung dapat menemukan halaman target dengan mudah dan dengan satu kunjungan saja. Kemudian pengguna langsung meninggalkan *website*. Berbeda dengan pengguna dengan alamat IP 1039598537. Pengguna ini membuka 5 (lima) halaman pada kunjungannya. Halaman pertama yang diminta adalah "/berita/dunia/yusuf-qardawi-teruslah-turun-ke-jalan.htm". halaman berikutnya adalah "//nasihat-ulama/israel-harus-dihapus-dunia-islam-perlu-direformasi.htm, /konsultasi/konspirasi/kontroversi-natal-25-desember-2.htm, /berita/dunia/yahudi-menentang-yahudi.htm" dan /berita/dunia/yordania-akan-bekukan-hubungan-diplomatik-dengan-israel.htm" karena akhir dari ketiganya tersebut adalah halaman target maka pengguna ini dapat dikatakan telah melakukan 3 (tiga) kali transaksi.

D. *Data Processing*

1. Pencarian Lokasi Harapan

Langkah pertama dari tahapan *data processing* adalah pencarian lokasi harapan. Lokasi harapan sebagai indikator awal ada atau tidaknya pengunjung yang tersesat di dalam *website* untuk pencarian halaman targetnya. Semakin banyak lokasi harapan yang dihasilkan maka semakin banyak pula pengunjung yang tersesat. Hal ini berarti struktur dari *website* tersebut masih kurang baik. Sebaliknya jika lokasi harapan yang dihasilkan sedikit atau tidak ada sama sekali maka artinya semakin sedikit pula pengunjung yang tersesat yang berarti ada indikasi bahwa struktur *website* tersebut sudah baik.

Pada penelitian ini ditemukan beberapa halaman lokasi harapan. Contoh halaman target adalah "/berita/dunia/iran-bantu-gaza-bentuk-pasukan-islam-dan-hentikan-ekspor-minyak-ke-as.htm" dan kolom E1 yaitu "/konsultasi/sehat/alergi-kulit.htm". Sedangkan pada kolom aktual adalah "/konsultasi/sehat/dima-klinik-bekam-yang-baik.htm". Hal ini berarti ada pengguna yang mengharapkan halaman target "/berita/dunia/iran-bantu-gaza-bentuk-pasukan-islam-dan-hentikan-ekspor-minyak-ke-as.htm" pada lokasi harapan "/konsultasi/sehat/alergi-kulit.htm" sedangkan *hyperlink* halaman target tersebut berada pada lokasi aktual "/konsultasi/sehat/dima-klinik-bekam-yang-baik.htm". Begitu juga seterusnya untuk baris-baris yang lain dalam tabel *tb_exp_locations*.

2. Penentuan Halaman Rekomendasi

Penelitian ini tidak menemukan halaman rekomendasi walaupun ditemukan lokasi harapan. Hal ini dikarenakan perbedaan karakteristik data yang digunakan pada penelitian ini dengan data *dummy* pada penelitiaan sebelumnya. Karakteristik data yang digunakan pada saat ini masih susah membedakan halaman indeks dan halaman target. Sehingga untuk menentukan halaman target harus menggunakan asumsi bahwa halaman tersebut berada pada *leaf* struktur *website* dan mempunyai jumlah kunjungan halaman paling banyak. Jika dilihat dari *website* aslinya ternyata memang pada halaman awal sudah terdapat *hyperlink* menuju ke halaman target, sehingga pengunjung tidak perlu bersusah payah mencari halaman targetnya.

Apabila sistem tidak menghasilkan halaman rekomendasi hal ini dapat berarti struktur *website* tersebut sudah baik atau bisa juga disebabkan oleh perbedaan karakteristik data pada algoritme *Find expectation Location* dan algoritme optimasi. Faktor lain yang dapat mempengaruhi halaman rekomendasi adalah panjang pendeknya *path completion*. Jika terlalu pendek kemungkinan sistem tidak akan menghasilkan halaman rekomendasi. Kemudian faktor yang berikutnya adalah kemungkinan sudah ada *hyperlink* ke halaman target pada halaman utama.

E. Visualisasi

Pada tahapan visualisasi struktur *website* digunakan gambar dengan format SVG (*Scalable Vector Graphic*) sebagai salah satu format gambar yang bersifat dinamis. Visualisasi struktur *website* hanya dapat memperlihatkan struktur awal dari *website* tanpa adanya penambahan hasil rekomendasi. Hal ini dikarenakan data pada penelitian ini tidak menghasilkan halaman rekomendasi. Visualisasi ini menggunakan simbol lingkaran (*circle*) untuk mewakili *path* atau halaman. Untuk hubungan keterkaitan atau *hyperlink* antarhalaman disimbolkan dengan menggunakan lurus berwarna biru. Nama-nama halaman *website* disimbolkan dengan angka. Angka yang digunakan tergantung dari banyaknya halaman yang divisualisasikan. Sedangkan untuk *hyperlink* hasil rekomendasi akan digambarkan dengan garis putus-putus dengan warna merah.

Karena penelitian ini tidak menghasilkan halaman rekomendasi maka visualisasi hasil penelitian data *log* hanya menampilkan struktur awal *website* tanpa halaman rekomendasi seperti dapat dilihat pada Lampiran 10. Untuk menguji visualisasi struktur *website* awal dan struktur *website* yang sudah ditambahkan *hyperlink* rekomendasi maka kemudian digunakan data dari hasil penelitian sebelumnya oleh Nurdian Setyawan (2008) yang menggunakan data *dummy*. Hasil dari kedua algoritme optimasi (*FirstOnly, OptimizeTime*) akan diambil untuk menguji visualisasi. Pada hasil algoritme *FirstOnly* nilai maksimal dari *threshold* adalah 44 dengan nilai maksimal dari *minsup* adalah 10%. Sedangkan untuk algoritme *OptimizeBenefit* nilai maksimal *threshold* adalah 52 dengan nilai maksimal *minsup* juga 10%. Parameter yang digunakan adalah *threshold* 37 dan *minsup* 7%.

Hasil kedua algoritme ini menunjukkan halaman rekomendasi yang berbeda-beda sehingga menghasilkan struktur *website* yang berbeda juga. Perbedaan pada algoritme ini diantaranya adalah pada algorirme *FirstOnly* lokasi harapan yang dilibatkan hanya 1 (satu) lokasi harapan yaitu E1. Sedangkan pada algoritme *OptimizeBenefit* melibatkan 4 (empat) lokasi harapan yaitu E1, E2, E3, dan E4. Sehingga waktu yang dibutuhkan oleh kedua algoritme sangat berbeda.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini telah berhasil melakukan tahapan *data preprocessing* dengan menggunakan data *log* asli dari *website* yang disimpan dalam bentuk *database mysql*, sehingga data yang nantinya akan diproses menggunakan algoritme optimasi (*FirstOnly*, *OptimizeBenefit*) harus melalui tahapan *data preprocessing* terlebih dahulu. Penelitian ini juga sudah dapat menampilkan struktur dari *website*, baik struktur awal *website* maupun struktur yang sudah ditambahkan dengan halaman rekomendasi pada *website* tersebut. Visualisasi menggunakan SVG (*Scalable Vector Graphic*) membuat visualisasi lebih dapat dimengerti oleh pengguna sistem dan lebih dinamis. Dengan demikian, jika dilakukan percobaan terhadap data *log* dari *website* lain, visualisasi ini tetap dapat digunakan.

B. Saran

Pada penelitian ini hanya digunakan data *log* dari satu *website*. Untuk itu perlu dicoba menggunakan data dari beberapa *website* yang lain (dinamis maupun statis) yang

sudah dapat dibedakan halaman target dan halaman indeks dari *website* tersebut.

DAFTAR PUSTAKA

[1] Castellano G, Fanelli A M, Torsello M A. 2007. *Log Data Preparation For Mining Web Usage Patterns*. maya.cs.depaul.edu/~classes/ect584/papers/cms-kais.pdf [27 Maret 2009].

[2] Catledge L, Pitkow JE. 1995. *Characterizing Browsing Strategies in the World-Wide Web*. http://smartech.library.gatech.edu/dspace/bitstream/1853/3558/1/95-13.pdf [8 April 2008]

[3] Cooley R, Mobasher B, Srivastava J. 1999. *Data Preparation for Mining World Wide Web Browsing Pattern.* http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=ECF66FEC62532CF4833FIE2AC36C5C93?doi=10.1.1.33.2792&rep=rep1&type=pdf[8 April 2008]

[4] Ivancy R, Vajk I. 2006. *Frequent Pattern Mining in Web Log Data*. Acta Polytechnica Hungarica, Vol. 3, No. 1.

[5] Setyawan N. 2008. *Rekomendasi Penambahan Link Pada Web Berdasarkan Pola Akses Pengguna* [Skripsi]. Bogor. Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam.

[6] Srikant R, Yang Y. 2001. *Mining Web Log to Improve Website Organization.* http://www.almaden.ibm.com/quest/papers/www10_weblog.pdf [22 September 2007].

[7] Veer EV *et al*. 2004. *Creating Web Pages All-In-One Desk Reference for Dummies* Edisi Ke-2. Indianapolis; Wiley Publishing, Inc.