ADAPTIVE WEB SITE DENGAN METODE *FUZZY CLUSTERING*

Muchammad Husni, Waskito Wibisono, dan Wahyu Nugroho
Jurusan Teknik Informatika,
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5913804
E-mail : husni@its-sby.edu

ABSTRAK

Ledakan pertumbuhan dan perkembangan informasi dalam dunia maya menjadikan personalisasian informasi menjadi isu yang penting. Personalisasi informasi yang akan diberikan oleh situs web akan sangat mempengaruhi pola dan perilaku pengguna dalam pencarian informasi, terutama pada perdagangan elektronis (e-commerce).

Salah satu pendekatan yang memungkinkan dalam personalisasian web adalah mencari profil pengguna (user profile) dari data historis yang sangat besar di file web log. Pengklasifikasian data tanpa pengawasan (unsupervised clasification) atau metode metode clustering cukup baik untuk menganalisa data log akses pengguna yang semi terstruktur. Pada metode ini, didefinisikan "user session" dan juga ukuran perbedaan (dissimilarity) diantara dua web session yang menggambarkan pengorganisasian sebuah web site. Untuk mendapatkan sebuah profil akses pengguna, dilakukan pembagian user session berdasarkan pasangan ketidaksamaan menggunakan algoritma Fuzzy Clustering.

Kata kunci : *Adaptive Website, Fuzzy Clustering*, personalisasi informasi.

1. PENDAHULUAN

Perkembangan internet yang sangat pesat membuat pertukaran informasi menjadi sangat mudah dan cepat. Akibat dari pertukaran informasi itu, jumlah informasi yang ada pada *world wide web (WWW)* menjadi sangat banyak dan bertumpuk-tumpuk. Keuntungan dari banyaknya informasi itu pengguna bisa mendapatkan informasi yang dibutuhkan hanya dengan mencarinya di internet, tetapi untuk mencari informasi yang cocok seringkali pengguna harus menjelajahi beberapa halaman web. Waktu pengguna banyak tersita hanya untuk mencari informasi yang benar-benar cocok.

Personalisasi pengguna adalah salah satu cara yang cukup baik dalam mengatasi permasalahan ini. Personalisasi dapat dilakukan dengan dua macam cara yaitu malalui *information brokers* misalnya mesin pencari (*search engine*) dan pendekatan *end-to-end* dengan membuat situs web menjadi adaptif.

Suatu situs web yang tidak adaptif akan menampilkan halaman web yang sama setiap hari beserta *link link* yang tidak dibutuhkan. Sedangkan situs web yang adaptif akan mempelajari kebiasaan pengguna dan menampilkan suatu halaman web yang sudah diubah sesuai dengan ketertarikan pengguna. Dalam hal ini pengguna tidak perlu membaca berita politik pada halaman depan dan sebagai gantinya dapat melihat berita olahraga hanya dengan satu klik pada link di halaman depan tanpa perlu membuka link-link yang banyak untuk melihat berita olah raga.

Dari uraian di atas penelitian ini bertujuan untuk membuat dan menerapkan *data clustering* yang efektif pada weblog untuk mendapatkan pola penjelajahan dari pengguna sehingga dapat dihasilkan versi-versi personalisasi web page yang berbeda untuk pengguna yang berbeda.

2. PERANGKAT LUNAK WEB SERVER

Pada saat perangkat lunak web server dibuat, NCSA *(National Center for Supercomputing Application)* membuat Common Log-file Format (CLF), yang dipakai oleh sebagian besar software web server. Common Log-file Format merupakan format yang banyak dipakai oleh web server dan disepakati sebagai standar penulisan dalam file log pada web server. Common Log-file Format ini mempunyai susunan sebagai berikut

Rhost	-	Auth User	Date	Request	Status	Bytes

Perangkat lunak web server apache sekarang ini menjadi web server yang paling populer dan paling banyak dipakai dalam dunia internet sejak April 1996. Menurit survey pada februari 2001 dari *net craft web server survey* (http://www.netcraft.com/survey) menemukan bahwa 70% dari web site yang ada di internet menggunakan apache sebagai web servernya. Itulah salah satu alasan penggunaan Apache pada perangkat lunak ini

disamping Apache mempunyai sifat *Multi platform* dan tingkat keamanan yang bagus.

File access log pada apache *(access_log)* merekam semua permintaan yang diproses oleh web server. Lokasi dan isi dari file access log pada apache web server dikontrol oleh directive *CustomLog*. Directive *LogFormat* dapat digunakan untuk menyederhanakan dalam pemilihan isi dari file log yang digunakan. Bagian ini menjelaskan bagaimana untuk mengkonfigurasi server agar dapat menyimpan informasi yang terjadi dalam file access log. Beberapa versi dari apache httpd menggunakan modul-modul dan directive untuk mengontrol access log, termasuk *mod_log_referrer*, *mod_log_agent*, dan directive *TransferLog*.1

Sedang PHP merupakan script untuk pemrograman script web server-side, script yang membuat dokumen HTML secara *on-the-fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Salah satu alasan penggunaan PHP pada perangkat lunak ini adalah PHP dapat berjalan dengan baik pada Apache maupun IIS.

Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP.

PHP/FI merupakan nama awal dari PHP. PHP – Personal Home Page, FI adalah Form Interface. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web.

3. FUZZY CLUSTERING

Fuzzy Clustering adalah algoritma clustering untuk mendapatkan keangggotaan dari suatu cluster dengan menerapkan teori Fuzzy. Tingkat keanggotaan suatu pola terhadap cluster yang ada tergantung dari derajat keanggotaannya pada cluster tersebut. Teori ini sangat cocok diterapkan pada weblog karena data yang ada pada weblog bukanlah dari numerik melainkan URL dari suatu web. URL tersebut dapat dicari similarity dan dissimilaritynya sehingga dapat diolah dengan teori *Fuzzy Clustering* seperti yang akan diterangkan di bawah ini.

Algoritma *Fuzzy Clustering* yang ada kebanyakan tidak memperhatikan *Robustness* atau kekuatan dan menganggap jumlah komponen sudah diketahui. Teori robust tercipta tidaklah berdasarkan fuzzy, namun dua algoritma itu mempunyai banyak kesamaan, dan bila bersama mereka dapat saling melengkapi. Contohnya Fuzzy clustering dapat memecah persoalam perihal multiple cluster. Sebaliknya, Robust menekankan pada aspek kuatnya komponen tetapi hanya mengacu pada single komponen. Robust dapat menangani noise-noise, kesamar-samaran, dan ketidakpastian dan ketidaklengkapan informasi yang terdapat dalam data weblog.

Jadi *Fuzzy Clustering* dengan pendekatan robust adalah algoritma pencarian pola yang terdapat dalam data (weblog) secara fuzzy yang dapat menangani noise-noise, kesamar-samaran, dan ketidakpastian dan ketidalengkapan informasi yang terdapat dalam data weblog.

Untuk mengelompokkan (klasterisasi) session diperlukan algoritma yang dapat menerima derajat kesamaan/ketidaksamaan dalam data relasional, dan yang penting algoritma itu juga dapat menangani noise pada data. Untuk itu digunakan algoritma Fuzzy C-Medoids (FCMdd):

$$Jm(V;X)=\sum_{j=1}^{M}\sum_{i=1}^{c}u_{ij}\gamma(xj,vi) \quad (1)$$

dimana X = $\{x_i|\ i= 1,2,\ldots,n\}$ adalah kumpulan dari objek m. $\gamma(x_i,x_j)$ adalah dissimilarity antar dua objek. $V=\{v_1,v_2,\ldots,v_c\}$, $v_i \in X$ merupakan subset dari X dengan jumlah c. Dan u_{ij} merupakan fuzzy dari keanggotaan x_j pada cluster i.

$$u_{ij}=\frac{\left(\frac{1}{\gamma(xj,vi)}\right)^{1/(m-1)}}{\sum_{k=1}^{c}\left(\frac{1}{\gamma(xj,vk)}\right)^{1/(m-1)}}$$

dimana $m \in [1,\infty)$ adalah "*fuzzifier*". (2)

Penggunaan algoritma diatas memunculkan masalah kompleksitas yang tinggi untuk suatu pemrograman web karena harus menghitung semua kandidat n, *O(nn)*. Padahal dalam kenyataannya, dengan menyaring n untuk mendapatkan misalnya k buah objek yang mempunyai tingkat keanggotaan yang tinggi dalam cluster, kompleksitas dapat dikurangi menjadi *O(kn)*

Tetapi seperti diketahui algoritma di atas yang meminimalkan fungsi pendekatan tipe *least-squares* adalah tidak *Robust*. Dengan kata lain, (apabila ada objek yang sebenarnya diluar perhitungan ikut terhitung) hasil yang diperoleh bisa sangat berbeda dengan yang diharapkan. Untuk mengatasi hal ini, dibuat variasi *FCMdd* yang berdasarkan *Least Trimmed Square*. Dengan memasukkan u_{ij} kedalam fungsi J_m didapatkan :

$$J_m(\mathrm{V};\mathrm{X})=\sum_{j=1}^{n}\left(\sum_{i=1}^{c}\left(\gamma(x_j,v_i)\right)^{1/(1-m)}\right)^{1-m}=\sum_{j=1}^{n}(h_j) \quad (3)$$

$$h_j = \left(\sum_{i=1}^{c} \left(\gamma \left(x_j, v_i \right) \right)^{1/(1-m)} \right)^{1-m} \quad (4)$$

sehingga Algoritma Fuzzy c Trimmed Medoids dimodifikasi menjadi :

$$J_m^T(\mathrm{V};\mathrm{X}) = \sum_{k=1}^{s} h_{km} \quad (5)$$

4. IMPLEMENTASI PROSES

Implementasi proses pada dasarnya merupakan penulisan baris-baris perintah untuk menangani bentuk-bentuk pemrosesan yang telah dispesifikasikan. Dalam hal ini terdapat empat tahap utama, yang meliputi preparasi data, pencarian dissimilarity, dan pencarian cluster, serta pembuatan halaman web.

4.1. PREPARASI DATA

Preparasi data terdiri dari tiga proses yaitu penstrukturan data, pencarian IP dan tree URL, dan *sessioning*. Proses ini diimplementasikan kedalam tiga buah program. Penstrukturan data, pencarian IP dan tree URL, dan sessioning data. Penstrukturan data dilakukan dengan mengolah masukan menjadi data yang terstruktur kemudian hasilnya dimasukkan ke dalam tabel logasli.

Setiap masukan dari log berisi : Alamat IP pengguna, waktu akses, metode request(GET,POST,HEAD, dll), URL yang diakses, protocol (HTTP/1.*), return code, dll. Pertama-tama masukan itu disaring dan membuang yang tidak diperlukan dalam pengolahan preparasi data. Data yang dibuang meliputi data yang mempunyai nilai code error baik itu error client maupun server (4** dan 5**), metode request selain "GET", URL yang mengarah ke image file (.gif,.jpg,dll). Hasil dari penyaringan ini dimasukkan ke dalam tabel logasli yang nantinyadigunakan untuk proses selanjutnya.

Proses kedua adalah pencarian IP dan URL pada web site itu. IP diambil pada tabel logasli, dipilah dan dibedakan, dimasukkan ke dalam tabel list_IP. Begitu juga dengan tree_URL, setelah dipilah, dimasukkan ke dalam list_URL sebagai temporary storage untuk diurutkan lagi dan dimasukkan ke dalam tree_URL. Dengan mengurutkan abjad maka akan dihasilkan urutan URL yang menyerupai tree direktori URL itu sebenarnya.

Proses ketiga yaitu *sessioning*. Pada proses ini dilakukan pengambilan data log. Setiap URL diberi nomor id khusus sehingga $j \in \{1,2,...,N_u\}$, dimana N_u adalah jumlah total dari URL yang ada. Dan, *i* adalah session ke *i*. Maka User Sessioning dapat didefinisikan :

$S_j^{(i)}$ = 1 jika pengguna mengakses URL ke *j* dalam sesi ke *i*, dan

$S_j^{(i)}$ = 0 jika tidak.

Hasil dari Sessioning adalah suatu matrix N x 1, yang merepresentsikan pengaksesan URL URL tertentu (1,...,N) pada saat session ke i oleh ip satu IP. Pertama-tama matrix session yang terjadi adalah :

$S^{(i)}$ =[00001000000000000000000000000000000000

00000]

Matrix session itu dimasukkan ke dalam tabel sesibelum. Matrix tersebut belumlah final, karena belumlah menggambarkan suatu IP dalam satu session hanyalah satu, tetapi satu IP dalam satu session terdapat banyak. Untuk menjadikan suatu IP dalam satu session hanya satu dilakukan penggabungan antar matrix dalam satu session yang mempunyai IP sama.

$S^{(i)}$ = $S^{(i)}$ or $S^{(i)}$

4.2. PENCARIAN DISSIMILARITY

Pada proses ini dilakukan dalam dua tahap yaitu pencarian M1,dan pencarian M2,M dan dissimilarity. Pada pencarian M1, dilakukan pencarian similarity yang sederhana mengabaikan struktur hierarkhi dari web dan dianggap URL URL itu berdiri sendiri. Fungsi yang digunakan :

Pada tahap dua, formula yang digunakan adalah :

$$M_{2,kl} = \frac{\sum_{i=1}^{Nu} \sum_{j=1}^{N_u} s_i^{(k)} s_j^{(l)} Su(i,j)}{\sum_{i=1}^{Nu} s_i^{(k)} \sum_{j=1}^{Nu} s_j^{(l)}}$$

Hasil M didapat dari pencarian nilai yang terbesar M1 dan M2, keludian didapat dissimilarity dari tiap tiap session dengan :

$$d_s^2(k,l) = (1 - M_{kl})^2$$

4.3. PENCARIAN CLUSTER

Implementasi dari pencarian cluster dengan menggunakan algoritma *Fuzzy c trimmed Medoids* (FCTMdd) bisa dilihat di gambar 1.

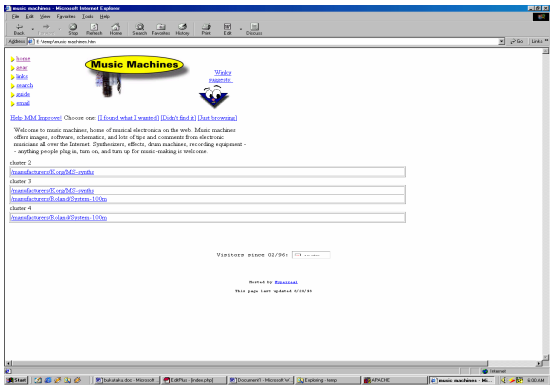
4.4. PEMBUATAN HALAMAN WEB

Pembuatan halaman web meliputi mendapatkan IP dari pengguna dan mencari kelas pengguna dalam tabel kelas dan menampilkan halaman web yang telah diubah sesuai dengan kelas pengguna tersebut. Contoh tampilan halaman web bisa dilihat pada gambar 2, dan 3.

```
<?
Connect to Database TA
Fix the number of cluster c;
Random pick initial set medoids;
Iter=0;

While (Vold = V or iter=max_iter)
        Compute harmonic dissimilarity
using …. (4);
        Sort Hj and keep s first object
of Hj;
        Compute membership for s
object:
        For j=1 to s do
                For I=1 to c do
                        Compute Uij
using …… (2);
                Endfor

        Endfor
        Store current medoids : Vold =
V;
        Compute mew medoids :
        Fori=1toc do
        q = arg min_{1≤k≤s} Σ_{j=1}^{s} uij_m^m γ(x_km, x_jm);
        vi=xq;i
        endfor
        Iter++
Endwhile

?>
```

Gambar 1.
Pseudocode Program untuk algoritma FCTMdd

Gambar 2.
Tampilan Halaman web sebelum diubah



Gambar 3.
Tampilan Halaman web setelah diubah

5. KESIMPULAN DAN SARAN

5.1. KESIMPULAN

Dari hasil ujicoba dan evaluasi yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut :

1. Klasterisasi data pada weblog dapat menghasilkan pola penjelajahan yang lebih efektif dari pengguna. sehingga dapat dihasilkan aplikasi web page yang dapat menghasilkan versi-versi web page yang berbeda (mungkin sama) untuk pengguna yang berbeda.
2. Aplikasi web yang telah dibuat dapat membantu pengguna mendapatkan *guide line* dalam menjelajahi internet sehingga dapat mengurangi upaya yang harus dikeluarkan oleh pengguna dalam mencari suatu informasi.
3. Aplikasi ini mengolah informasi dari *data semi structured* yang ada di access log web server, mencari pola penjelajahannya dengan menggunakan algoritma *fuzzy clustering* untuk mendapatkan kelompok kelompok kelas *(cluster)* yang mempunyai persamaan dengan batasan tertentu. Sehingga dapat menampilkan halaman web yang sesuai dengan kelas (*cluster*) akses pengguna sehingga dapat ditampilkan informasi dan *URL* yang mungkin diinginkan oleh pengguna.

5.2. SARAN

Beberapa saran untuk pengembangan selanjutnya dari perangkat lunak dijelaskan sebagai berikut :

Kemampuan pengenalan pengguna pada perangkat lunak ini hanyalah dari IP tiap tiap komputer pengguna. Untuk pengembangan lebih lanjut dapat dilakukan penganalisaan terhadap cookie yang dibuat secara otomatis oleh web sebagai pengganti IP pengguna. Hal ini berguna apabila ada pengguna yangmengakses dari suatu intranet yang mempunyai IP virtual.

Penggunaan PHP sebagai bahasa pemrograman web sebenarnya cukup tepat, akan tetapi untuk perhitungan dissimilarity yang sangat mahal bagi bahasa yang menggunakan sistem script, PHP tidaklah terlalu baik karena waktu yang dihabiskan untuk eksekusi sangat lama. Solusi untuk masalah ini adalah dengan menggunakan dua bahasa pemrograman sekaligus yaitu PHP untuk scripting dan pengambilan data, sedangka bahasa C (atau compiler yang lain) untuk pengolahan datanya.

Adaptive web server terbukti bisa mempersonalisasikan web sesuai dengan pola akses pengguna tanpa meminta profil pengguna. Sifat ini cocok untuk diaplikasikan pada web-web yang biasanya meminta personalisasi pengguna, misalnya portal-portal informasi, dan tidak menutup kemungkinan untuk web-web seperti koran, majalah.

6. DAFTAR PUSTAKA

1. Kamdar, T. and Joshi, A. , On Creating Adaptive Web Servers Using Weblog Mining,. 2000.
2. Joshi, A., Joshi, K. and Krishnapuram, R., On Mining Web Access Logs., In Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery 2000, pp. 63--69, 2000.
3. Nasraoui, O., Krishnapuram, R. and Joshi, A., Mining Web Access Log Using a Fuzzy Relational Clustering Algorithm Based on a Robust Estimator, 1999.
4. Gallegos, Maria Theresa, A Robust Method for Clustering Analysis,1999.
5. Nasraoui, O., Krishnapuram, R. and Joshi, A., Relational Clustering Based on a New Robust Estimator with Application to Web Mining, In Proceedings of NAFIPS 99, (New York), pp. 705-709, June 1999.
6. Joshi, A. and Krishnapuram, R. , Robust Fuzzy Clustering Methods to Support Web Mining, Proc. Workshop in Data Mining and knowledge Discovery, SIGMOD, pp. 15-1 -- 15-8, 1998.

7. Krishnapuram, R., Joshi, A. and Yi, L., A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering, In Proceedings of IEEE Intl. Conf. Fuzzy Systems-FUZZIEEE99, Korea, 1999.
8. Dave, R.N and Krishnapuram, R., Robust Clustering Methods: A Unified View, IEEE Trans. Fuzzy Systems, 5:2, pp 270-293, 1997.
9. Network Working Group, Request for Comments: 2616 on Hypertext Transfer Protocol -- HTTP/1.1 , Internet Society, June 1999.