SEQUENTIAL PATTERN UNTUK PEMBANGUNAN SISTEM PEMBERI REKOMENDASI TAUTAN PADA *WEBSITE*

Novi Setiani
Teknik Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
JL. Kaliurang KM 14,5 Sleman, DIY, 55584
115230101@uii.ac.id

ABSTRAK

Sistem pemberi rekomendasi tautan merupakan salah satu bentuk personalisasi pengguna pada aplikasi berbasis web. Sistem ini akan memberikan rekomendasi untuk pengunjung berupa tautan ke suatu halaman berdasarkan histori pola pengaksesan web. Untuk membangun pola pengaksesan web ini, digunakan salah satu teknik data mining yaitu sequential pattern mining. Teknik ini akan menghasilkan serangkaian pola penelusuran pengguna dalam menjelajahi sebuah website. Dalam penelitian ini, algoritma sequential pattern mining yang digunakan adalah PrefixSpan. Algoritma ini lebih efisien dalam pembangunan pola sekuensial dibandingkan algoritma general sequential pattern lainnya.
Parameter yang mempengaruhi performansi sistem adalah support threshold, panjang session pengguna yang sedang online, dan nilai confidence minimum. Secara umum, tingkat akurasi berbanding lurus dengan ketiga parameter ini meskipun secara khusus, sistem dengan ukuran session dinamis tidak terlalu dipengaruhi oleh ukuran session window. Secara fungsional, sistem rekomendasi dengan ukuran session dinamis lebih baik dibandingkan sistem yang menggunakan ukuran session statis karena lebih mampu menghindari kegagalan dalam memberikan rekomendasi meskipun tingkat akurasinya sedikit lebih rendah (67%) dibandingkan ukuran session statis (76%). Waktu rata-rata untuk memberikan rekomendasi pada setiap user session adalah 0.1 - 0.13 detik.

Kata kunci : *sequential pattern mining*, prefixspan, *web usage mining*, sistem pemberi rekomendasi

PENDAHULUAN

Pertumbuhan *website*, *e-commerce* dan *web-based information system* yang dikunjungi oleh jutaan pengguna menjadikan sebuah alasan pentingnya personalisasi sebagai salah satu kesuksesan sebuah website. Sebuah situs yang sangat kompleks karena konten atau halaman yang disediakan sangat banyak (contoh: *e-commerce*), jika tidak dilengkapi personalisasi, maka pengunjung akan kesulitan memperoleh informasi yang diinginkan. Sedangkan melalui personalisasi, seorang pengunjung situs dapat memperoleh rekomendasi mengenai sebuah halaman, konten, jasa atau produk yang sesuai dengan karakteristik atau profilnya. Salah satu bentuk personalisasi adalah dengan memberikan rekomendasi link kepada pengguna.
Link recommender system dapat dikembangkan dengan menggunakan arsitektur *memory- based* dan *model-based* (Mobasher, 2007). Salah satu keuntungan *model-based* adalah dapat mengatasi permasalahan skalabilitas *karena* pembentukan profil pengguna dilakukan secara offline. Teknik *data mining* yaitu *sequential patern mining* dapat dimanfaatkan untuk membentuk profil pengguna berupa pola penelusuran terhadap halaman-halaman dalam sebuah situs.
Tujuan penelitian ini adalah menemukan konfigurasi terbaik berdasarkan parameter algoritma sequential pattern mining dan algoritma rekomendasi. Untuk menghasilkan akurasi yang lebih baik, dilakukan modifikasi terhadap algoritma rekomendasi sehingga pengaruh sejarah penelusuran pengguna akan bersifat dinamis.

METODOLOGI PENELITIAN

Metodologi yang digunakan dalam penelitian ini terdiri dari empat tahap yaitu sebagai berikut.

1. Melakukan analisis untuk memahami pemanfaatan teknik *sequential pattern* pada pembangunan aplikasi *recommender system*. Pengembangan aplikasi *recommender system* yang terdiri dari analisis, perancangan, dan implementasi aplikasi sesuai dengan pendekatan yang telah didefinisikan pada tahap sebelumnya.
2. Pengujian aplikasi *recommender system* pada aspek performansi dan fungsional. Uji performansi akan dilakukan dengan menghitung akurasi sistem dan waktu yang diperlukan dalam memberikan rekomendasi kepada pengguna.

Sequential pattern untuk Sistem Pemberi Rekomendasi Tautan

Sequential pattern mining adalah salah satu proses *data mining* untuk menghasilkan pengetahuan mengenai serangkaian kejadian yang frekuensi kemunculannya melebihi suatu nilai minimum yang ditentukan (Jian dan Jiawei, 2001). Fungsinya dalam *link recommender system* adalah untuk menghasilkan serangkaian pola penelusuran pengguna dalam menjelajahi sebuah situs. Pola ini akan digunakan untuk memberikan rekomendasi kepada pengguna yang sedang mengunjungi sebuah situs.

Link Recommender System yang dikembangkan oleh (Mobasher, 2002) memberikan hasil bahwa teknik yang menggunakan *less constrained patterns*, seperti *frequent itemsets* atau *general sequential patterns* cocok digunakan untuk melakukan personalisasi web dan *recommender system*. Penelitian ini melakukan eksperimen pada *website* Universitas Depaul dan memberikan hasil akurasi maksimum 70%. Pada penelitian ini, akan dimanfaatkan teknik *sequential pattern mining* yang memiliki performansi dan efisiensi *memory* yang lebih baik yaitu algoritma PrefixSpan (Jian, 2001) supaya akurasi sistem dapat meningkat.

Pemanfaatan Sequential Pattern Mining untuk Menemukan Pola Penelusuran

Untuk melakukan *sequential pattern mining*, diperlukan input berbentuk basisdata sekuens. dimana tiap sekuens adalah daftar akses *user* yang diurutkan berdasarkan waktu akses dan tiap akses terdiri dari kumpulan informasi yang berguna, kemudian dicari semua pola akses dengan minimum *support*, yaitu jumlah dari database sekuens yang mengandung pola tersebut.

Dalam konteks *recommender system* ini, sekuens yang dihasilkan dari basis data transaksi hanya akan terdiri dari satu *item* karena terdapat sebuah asumsi bahwa dalam satu satuan waktu hanya ada satu *pageview* (sebuah halaman unik dalam sebuah situs) yang dikunjungi oleh pengguna. Bentuk himpunan pageview atau daftar akses *user* yang dihasilkan dari tahap *data preparation* dapat dipandang sebagai sebuah basisdata sekuens. Setelah tersedia *sequence database*, dapat dilakukan mekanisme *sequential pattern mining* berdasarkan suatu nilai *support threshold* yang merepresentasikan jumlah dari database sekuens yang mengandung pola tersebut. *Sequential pattern* dalam konsep analisis penggunaan web, menggambarkan pola penelusuran web yang sering dikunjungi oleh pengguna, sesuai dengan urutannya.

Untuk memperoleh sequential pattern dari daftar akses *user*, digunakan algoritma PrefixSpan (*Prefix-Projected Sequential Pattern Growth*) karena performansi dan efisiensinya lebih baik dibandingkan algoritma *sequential pattern* lainnya (Jian, 2001). Prefixspan memakai pendekatan pengembangan *sequence* untuk mencari *sequential pattern* . PrefixSpan akan mencari *frequent sequence* satu elemen dan kemudian mengembangkan *sequence-sequence* tersebut dengan cara menambahkan elemen satu persatu. PrefixSpan dirancang sedemikian rupa sehingga *sequence* hasil penambahan elemen tersebut tetap merupakan *frequent sequence*. Dengan cara ini, tidak diperlukan pembangkitan dan pengujian kandidat. Ide utama dari PrefixSpan adalah dengan memproyeksi basisdata yang memiliki prefix *frequent*. Ide ini dikembangkan karena adanya prinsip yang menyatakan bahwa setiap *frequent sequence* selalu dapat dihasilkan dari *prefix* yang *frequent*.

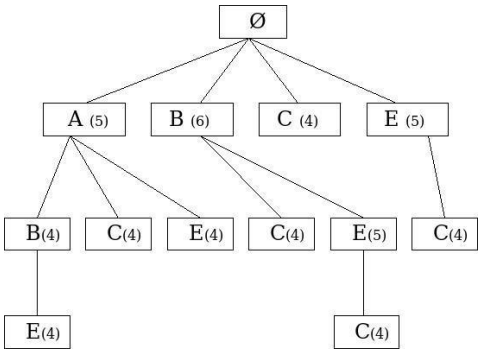
Terdapat dua parameter algoritma prefixspan yaitu *support threshold* dan *confidence*. Diberikan satu himpunan transaksi T dan satu himpunan *sequential pattern* $S = \{S_1, S_2, S_3, ..., S_n\}$ yang terjadi pada T. *Support threshold* untuk setiap S_i didefinisikan pada persamaan (1).

$\sigma(S_i) = \frac{|\{t \in T : Si \text{ adalah subsekuens dari } t\}|}{|T|}$ (1)

Confidence dari rule X → Y, di mana X dan Y adalah *sequential pattern* , didefinisikan pada persamaan (2), dimana o merepresentasikan operator konkatenasi sekuens.

$\alpha (X \rightarrow Y) = \sigma(X \text{ o } Y) / \sigma(X)$

Sequential pattern yang dihasilkan selanjutnya disimpan dalam pohon supaya memudahkan penelusuran profil pengguna secara *online*. Representasi profil pengguna berupa struktur data pohon akan disimpan dalam basisdata relasional dengan menggunakan metode *Modified Preorder Tree Traversal* . Metode ini cukup efektif untuk melakukan penelusuran pohon karena jumlah *query* yang diperlukan tidak tergantung pada kedalaman pohon. Hal ini akan menghemat waktu yang diperlukan untuk memberikan rekomendasi melalui penelusuran *sequential pattern tree* yang sudah disimpan secara persisten dalam basidata relasional.



Gambar 1. Sequential Pattern Tree

Setiap simpul pada Gambar 2.1 merepresentasikan sebuah halaman yang dikunjungi oleh pengguna, dan jumlah kemunculannya dalam setiap transaksi (Mobasher, 2001). Notasi dalam sebuah simpul (*A:4*), *A* menunjukkan halaman yang dikunjungi dan *4* menunjukkan jumlah kemunculannya. Penelusuran dari simpul akar menuju simpul tertentu akan menghasilkan sebuah pola (*pattern*) transaksi yang dilakukan sebagian besar pengguna dalam menggunakan situs..

Eksploitasi Profil dengan Menggunakan Algoritma Rekomendasi

Tahap eksploitasi profil pengguna merupakan tahap untuk memberikan rekomendasi kepada pengguna dengan memanfaatkan profil yang sudah dibangun. Algoritma ini memerlukan masukan berupa sejarah aktivitas pengguna yang sedang *online (session)* dan nilai minimum *confidence* yang harus dipenuhi. *Session* ini diperoleh saat pengguna mulai membuka *web browser* sampai dia menutupnya.

Untuk mengambil kedalaman sejarah aktivitas *current user* digunakan sebuah *sliding windows* pada *session* yang saat ini sedang aktif. Sebagai contoh, jika active *session* adalah <A, B, C> dengan ukuran *window* = 3, dan *user* mengakses halaman D, maka active *session* yang baru adalah <B, C, D>. Jadi, *sliding window* dengan ukuran *n* pada *session* yang sedang aktif akan mengijinkan *n* halaman terakhir yang dikunjungi *user* untuk mempengaruhi himpunan rekomendasi yang dihasilkan.

Dalam *link recommender system,* cukup sulit menentukan jumlah *sequential pattern* yang dapat digunakan untuk memberikan rekomendasi. Hal ini dipengaruhi oleh *threshold* dan *coverage* yang tarik menarik. Menurunkan nilai *threshold* akan mengharuskan untuk mengurangi ukuran *session window*. Pada umumnya, menggunakan ukuran *window session* yang lebih besar akan menaikkan tingkat akurasi dari rekomendasi yang diberikan. Namun, jika digunakan *threshold* yang lebih tinggi, maka ukuran *window session* yang besar akan mengakibatkan penurunan cakupan rekomendasi (*recommendation coverage*).

Untuk mengatasi permasalahan ini, dapat digunakan eksploitasi profil dengan memanfaatkan model *all-kth-order* Markov sehingga ukuran *window* akan bersifat dinamis. *Order* k dianalogikan sebagai ukuran *session window*, dan state merepresentasikan keadaan *user* saat ini. Implementasinya adalah sebagai berikut:

a. *Recommendation engine* menggunakan ukuran active *session window* yang paling besar dan memungkinkan sebagai input.

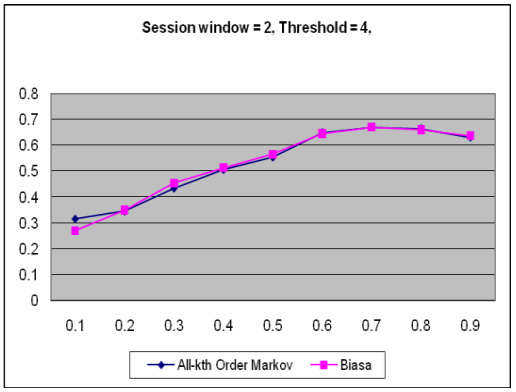
b. Jika *engine* tidak dapat membangkitkan rekomendasi, maka ukuran active *session window* akan dikurangi secara iteratif sampai menghasilkan rekomendasi atau ukuran *window* diubah menjadi 0.

HASIL DAN PEMBAHASAN

Data yang digunakan adalah dua file weblog dari situs microsoft.com yang didapat dari (OKA, 09) dengan karakteristik sebagai berikut.

1. Data A : terdiri dari 295 pageview, 5000 *session user*, dan 15191 *clickstream*
2. Data B : terdiri dari 295 pageview, 28166 *session user*, dan 72764 *clickstream*

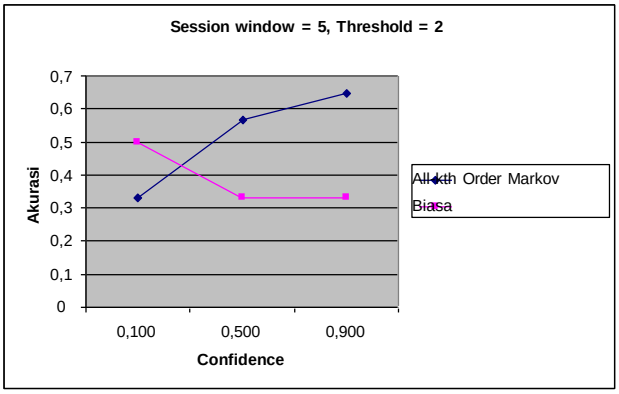
Setiap data akan dibagi menjadi 90% data latih dan 10% data tes. Data latih akan digunakan untuk membangun *sequential pattern trie* dalam basisddata. Data tes akan dibagi menjadi dua bagian, yaitu sebagai pembangkit rekomendasi dan sisanya sebagai penguji kebenaran rekomendasi yang dihasilkan. Misal terdapat sebuah *record* t dalam data tes, dan parameter panjang *session window* adalah n, maka n *sequence* pertama dari *record* tersebut akan digunakan sebagai pembangkit rekomendasi dan pageview sisanya (|t| - n) akan dibandingkan dengan rekomendasi yang dihasilkan.



Gambar 2. Hasil Pengujian Akurasi Data B

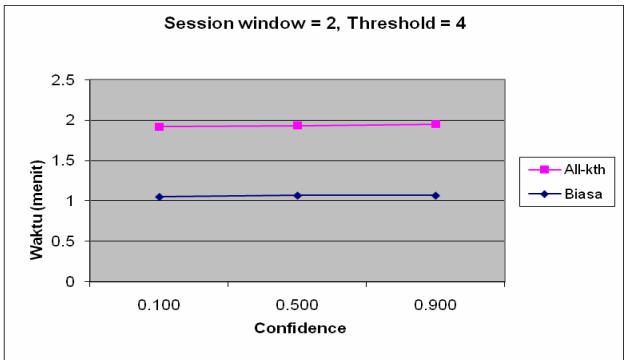
Tingkat akurasi algoritma All-kth Order Markov Model lebih stabil dibandingkan algoritma rekomendasi biasa, baik menggunakan data A maupun B. Yang mempengaruhi tingkat akurasi algoritma All-kth Order Markov Model adalah nilai confidence, sedangkan algoritma rekomendasi biasa lebih mudah terpengaruh oleh nilai support threshold dan *session* window.

Untuk ukuran *session* window yang lebih besar, ternyata algoritma rekomendasi biasa mampu memberikan performansi yang cukup baik jika dibandingkan dengan algoritma All-kth Order Markov. Hal ini dikarenakan algoritma rekomendasi biasa benar-benar memberikan rekomendasi berdasarkan data *session user* saat itu, sedangkan algoritma All-kth Order Markov berusaha memberikan rekomendasi dengan mengurangi pengaruh *session window* secara iteratif seperti terlihat pada Gambar 3.2.



Gambar 3. Hasil Pengujian dengan Data A

Pengujian terhadap waktu pemberian rekomendasi dilakukan terhadap data A dengan jumlah *session user* adalah 500 (10% x 5000). Waktu yang diperlukan untuk menghasilkan rekomendasi dengan menggunakan algoritma All-kth Order Markov Model memerlukan waktu yang lebih lama. Untuk satu *session user* dengan menggunakan algoritma All-kth Order Markov Model, diperlukan 64 detik/500 = 0,128 sedangkan menggunakan algoritma rekomendasi biasa diperlukan sekitar 53 detik/500 = 0,106 (Gambar 3.3).



Gambar 4. Performansi dari segi Waktu

KESIMPULAN

Secara umum, tingkat akurasi sistem berbanding lurus dengan nilai minimum *confidence*, ukuran *session window* dan *support threshold* meskipun secara khusus, sistem dengan ukuran *session* dinamis tidak terlalu terpengaruh oleh ukuran *session window*. Jika nilai *support threshold* pada proses *sequential pattern mining* terlalu kecil, maka nilai minimum *confidence*-nya harus dinaikkan supaya sistem tetap memiliki akurasi yang baik. Hal ini untuk menghindari penurunan *recommendation coverage.*

Secara fungsional, sistem rekomendasi dengan ukuran *session* dinamis lebih baik dibandingkan sistem yang menggunakan ukuran *session* statis. Hal ini dikarenakan sistem dengan ukuran *session* dinamis menghindari kegagalan dalam memberikan rekomendasi meskipun tingkat akurasinya sedikit lebih rendah (67%) daripada sistem yang menggunakan ukuran *session* statis (76%).

Untuk penelitian selanjutnya, sebaiknya digunakan algoritma pembentukan *sequential pattern trie* yang lebih efisien supaya waktu yang diperlukan untuk melakukan tahap *offline mining* tidak terlalu lama. Hal lain yang perlu diperhatikan adalah pemeliharaan profil pengguna yang masih dilakukan secara *offline*. Sebaiknya dilakukan mekanisme pemeliharaan representasi profil dengan cara memperhitungkan pilihan pengguna yang sedang *online* terhadap hasil rekomendasi.

DAFTAR PUSTAKA

Mobasher, Bamshad et al. (2002). *Using Sequential Patternand Non-Sequential Patternin Predictive Web Usage Mining Tasks*. Proceedings of IEEE International Conference on Data Mining (p. 669-672).

Mobasher, Bamshad. (2007). *Recommender System*. German Journal on Artificial Intelligence, Künstliche Intelligenz, Band 21 (p. 41-43).

Okane, Data Akses User Microsoft.com. Situs http://www.cs.uni.edu/~okane/souce/ISR/ diakses pada tanggal 5 Januari 2009 pada pukul 12:26 PM.

Pei, Jian, Han, Jiawei, et. Al. (2001) *PrefixSpan: Mining Sequential PatternEfficiently by Prefix-Projected Pattern Growth*. Proceedings of 17th International Conference on Data Engineering (p. 215-224).

BIODATA PENULIS

Novi Setiani. Penulis mengajar di Teknik Informatika UII, setelah menyelesaikan pendidikan S1 dan S2-nya di Informatika ITB. Area riset yang menjadi fokus utama penulis adalah bidang *data mining* dan *software engineering.*