JURNAL TEKNOLOGI TECHNOSCIENTIA ISSN: 1979-8415
Vol. 2 No. 1 Agustus 2009

ANT-WUM: ALGORITMA BERBASIS ANT COLONY OPTIMIZATION UNTUK *WEB* USAGE MINING

[1]Abdurrahman, [2]Bambang Riyanto T., [3]Rila Mandala, [4]Rajesri Govindaraju
[1,2,3]*Sekolah Teknik Elektro & Informatika-ITB,* [4]*Fakultas Teknologi Industri-ITB*

Masuk: 15 April 2009, revisi masuk : 20 Juli 2009, diterima: 24 Juli 2009

ABSTRACT

This paper is continuity research from our previous work in Ant-Miner implementation for web user classification. In our previous work, we implemented Ant-Miner algorithm for web user classification same with Ant-Miner for classification task in data mining domain. In this paper, we propose modification of heuristic function of Ant-Miner based on web usage mining (WUM) problem, that we name Ant-WUM. The heuristic function ACO is based on local problem domain. Information theory is common heuristic function used in classification task, such as implemented in C4.5 algorithm and ant-miner algo-rithm. Ant-WUM uses heuristic function based on closeness principle that implemented in clustering problem in WUM. We propose to use data from web access log, profile user, and transaction data to provide some attributes as term candidate of classification rule by Ant-WUM algorithm. We compared Ant-WUM algorithm with Ant-Miner algorithm. The result indicates that Ant-WUM has competitive result in term of accuracy rate, amount of rules, and computation time.

Keywords: *Ant-Miner*, *Ant-WUM*, heuristic function, *web* usage mining.

INTISARI

Paper ini merupakan kelanjutan riset sebelumnya dalam pemanfaatan algoritma *Ant-Miner* untuk melakukan klasifikasi pengguna *web*. Riset sebelumnya pemanfaatan algoritma *Ant-Miner* yang dimanfaatkan dalam domain data mining untuk permasalahan klasifikasi pengguna *web* dalam *web usage mining*. Dalam paper ini diusulkan modifikasi fungsi heuristik dalam algoritma *Ant-Miner* yang disesuaikan dengan permasalahan *web usage mining* (WUM), yang dinamakan algoritma *Ant-WUM*. Fungsi heuristik dalam algoritma ACO tergantung dari domain permasalahan yang akan diselesaikan. Teori informasi merupakan fungsi heuristik yang umum diimplementasikan dalam algoritma untuk fungsi klasifikasi sebagaimana diterapkan dalam algoritma C4.5 dan dalam algoritma *ant-Miner.* Fungsi heuristik *Ant-WUM* adalah modifikasi fungsi heuristik *ant-Miner* dengan menggunakan prinsip kedekatan yang digunakan dalam fungsi klusterisasi WUM dan menggunakan teori informasi untuk menentukan nilai informasi suatu term yang akan menjadi variabel untuk penghitungan jaraknya. Dan dalam paper ini, diusulkan penggunaan gabungan data yang terdiri dari hasil ekstraksi *web access log*, profil pengguna, dan data transaksi. Dalam riset ini telah dilakukan pengujian perbandingan antara algoritma *Ant-Miner* dengan *Ant-WUM.* Dari hasil uji menunjukkan bahwa *Ant-WUM* cukup kompetitif dengan *Ant-Miner* dalam tingkat akurasi, dari jumlah kaidah yang dihasilkan dan aspek waktu komputasi.

Kata Kunci: *Ant-Miner*, *Ant-WUM*, fungsi heuristik, *web usage mining*.

PENDAHULUAN

Interaksi pengguna (*user*) *web* menghasilkan data akses *web* yang maha besar dalam periode waktu yang tersimpan di file *web access log* di server. Data hasil interaksi ini diharapkan dapat memberikan informasi yang bernilai bagi pengelola *web* dalam rangka memasar-

[1]*mr.indonesian@gmail.com,*[2]*briyanto@lskk.ee.itb.ac.id,*[3]*rila@informatika.org,*
[4]*rajesri_g@mail.ti.itb.ac.id*

kan keberadaan *web*-nya dan produk ini yang dijualnya. Dalam kontek ini *web usage mining* (WUM) mempunyai peran dalam menemukan pengetahuan (*knowledge discovery*) dari data penggunaan *web* tersebut. Selain data *web access log,* data yang terbentuk dari interaksi pengguna dengan *web* e-commerce adalah data profil pengguna dan data transaksi.

Business intelligence (BI) merupakan salah satu fungsi WUM ini untuk membantu dalam kegiatan pemasaran keberadaan *web* dan produk yang dijualnya (Abraham,2003). Untuk membantu fungsi BI dalam WUM, maka diusulakan klasifikasi pengguna *web* dalam kategori pengguna potensial, retensi, dan baru dengan memanfaatkan tiga data yaitu *web access log,* profil pengguna, dan data transaksi. Masih sedikit peneliti dalam riset WUM yang memanfaatkan ketiga data tersebut (Jaideep, 2000). Parameter yang digunakan untuk melakukan klasifikasi pengguna *web* ini adalah sebagai berikut (Abdurrahman, 2006):

- *Recency*: berapa waktu lama pengguna berinteraksi dengan *web* sejak terakhir mengakses web.
- *Frequency*: berapa kali lama pengguna mengakses *web* dalam satuan waktu.
- *Intencity*: berapa total transaksi pembelian produk melalui *web*.

Klasifikasi pengguna *web* ini diharapkan dapat membantu aktivitas pemasaran *web* dan produknya dalam melakukan aktifitas (Abdurrahman, 2004):
Akuisisi pelanggan baru dari para pengguna. Retensi terhadap pelanggan eksisting. Penetrasi terhadap pelanggan eksisting dalam kerangka meningkatkan nilai transaksi penjualan.

Pengembangan algoritma *Ant-Miner yang dilakukan ini adalah* melakukan cara modifikasi fungsi heuristik untuk menghasilkan kaidah klasifikasi pengguna *web ini*, yang diberi nama *Ant-WUM*. Algoritma Ant-WUM merupakan metode untuk tugas klasifikasi pengguna *web* dalam *web usage mining.* Algorima ini merupakan pengembangan dari algoritma *Ant-Miner* yang berbasis pada *Ant Colony Optimization* (ACO) (Parpinelli, 2002). Hingga kini, belum ada peneliti yang memanfaatkan ACO dalam WUM untuk tugas klasifikasi pengguna *web* dan peneliti lain memanfaatkan ACO untuk klusterisasi pengguna *web* dan klasifikasi halaman *web* (Abrahaham, 2005; Spiliopoulou, 2007; Holden,2007). Algoritma *Ant-WUM* mempunyai fitur sebagai berikut:

- Ekstraksi kaidah klasifikasi bersumber dari data dengan atribut kategori, sehingga untuk atribut kontinyu akan dilakukan diskritisasi.
- Fungsi heuristik yang digunakan adalah prinsip kedekatan yang dimplementasikan dalam fungsi klusterisasi dalam WUM (Grear, 2006) dan pemanfaatan teori informasi untuk mengukur kualitas suatu informasi (Parpinelli, 2002).

Untuk menguji performansi algoritma Ant-WUM, telah dilakukan perbandingan performansi kedua algoritma tersebut mencakup aspek:

- Tingkat akurasi (*accuracy*): kemampuan algoritma dalam menghasilkan kaidah tingkat kesalahan rendah.
- Efisiensi komputasi(*computational efficiency*): waktu yang dibutuhkan algoritma untuk melakukan proses ini pembelajaran model klasifikasi pada data training maupun data uji.
- Kemudahan memahami (*interpretability*): kaidah yang dihasilkan dapat dipahami secara mudah oleh pengguna dan dapat digunakan untuk pengambilan keputusan.

Algoritma *Ant-Miner* merupakan pengembangan dari algoritma *Ant Colony Optimization* (ACO), yang difungsikan untuk tugas klasifikasi dalam *data mining* (Parpinelli, 2002). ACO merupakan sistem berbasis agen yang mensimulasikan perilaku natural sekelompok semut, *term*asuk di dalamnya mekanisme bekerjasama dan adaptasi. Dalam paper (Dorigo, 1999) penggunaan sistem ini merupakan metaheuristik baru untuk memecahkan masalah optimasi kombinasi yang kokoh dan serbaguna.

Pada dasarnya, disain algoritma ACO terdiri dari spesifikasi sebagai berikut (Bonabeu, 1999):

- Representasi masalah, dimana sekelompok semut akan membangun/

modifikasi solusi melalui pemanfaatan aturan transisi probabilistik (*probabilistic transition rule*) berdasarkan jumlah *pheromone* dan *local problem-dependent heuristik.*

- Sebuah metode untuk membangun solusi yang valid.
- Fungsi heuristik terkait dengan masalah yang didefinisikan (η) yang mengukur kualitas item-item yang dapat ditambahkan dalam pilihan solusi yang sedang dipilih.
- Kaidah updating *pheromone* yang menspesifikasikan modifikasi nilai *pheromone* (τ).
- Kaidah transisi probabilistik yang didasarkan pada nilai fungsi heuristik (η) dan nilai *pheromone* (τ) yang digunakan secara berulang untuk membangun solusi.

Kaidah klasifikasi yang akan dipecahkan oleh *Ant-Miner* dapat dipresentasikan dalam bentuk kaidah sebagai berikut:

IF <*term*1 AND *term*2 AND...> THEN <class>

Masing-masing *term* terdiri dari tiga bagian (atribut, operator, nilai), dimana nilai ini adalah nilai yang dimiliki oleh suatu atribut. Bagian operator adalah operator penghubung "=". *Ant-miner* ini hanya mengakomodasi atribut kategori (*categorical attribute*). Untuk atribut yang bernilai kontinyu didiskritkan pada tahap *preprocessing*.

Deskripsi umum algoritma *Ant-Miner* dapat dideskripsikan dalam *pseodo code* berikut ini (Parpineli, 2002):

TrainingSet = {all training cases};
DiscoveredRuleList = [ ]; / rule list is initialized with an empty list */*
WHILE (TrainingSet >Max_uncovered_cases)
t = 1; / ant index */*
j = 1; / convergence test index */*
Initialize all trails with the same amount of pheromone;
REPEAT
Ant_t starts with an empty rule and incrementally constructs a classification rule R_t; by adding one term at a time to the current rule; Prune rule R_t;Update the pheromone of all trails by increasing pheromone in the trail followed by Ant_t (proportional to the quality of R_t) and decreasing pheromone in the other trails (simulating pheromone evaporation);
*IF (R_t is equal to R_t–1) /*update convergence test */*
THEN j = j + 1;
ELSE j = 1;
END IF
t = t + 1;
UNTIL (i ≥ No_of_ants) OR (j ≥ No_rules_converg)
Choose the best rule R_{best} among all rules R_t constructed by all the ants;
Add rule R_{best} to DiscoveredRuleList;
TrainingSet = TrainingSet - {set of cases correctly covered by R_{best}};
END WHILE

Web mining merupakan aplikasi teknik *data mining* untuk mengekstrak pengetahuan (*knowledge*) dari data *web* (Abraham, 2003). Ada dua pendekatan yang digunakan untuk mendefinisiskan *web mining,* yaitu pendekatan berbasis proses (*process-centric view*) yang mendefinisikan *web mining* sebagai sekumpulan suatu aktivitas (*sequence of tasks*) Yang kedua adalah pendekatan berbasis data (*data-centric view*) yang mendefinisikan *web mining* sebagai *term*inologi tipe data *web* yang digunakan untuk proses *data mining.* Dalam paper ini pendekatan yang digunakan adalah pendekatan kedua.

Web mining dapat dibagi dalam tiga kategori berdasarkan jenis data yang diekstrak, yaitu (Abraham, 2003): *Web content mining* (WCM); merupakan penemuan informasi terhadap content *web*, yang terdiri dari teks, gambar, audio, video, metadata, dan *hyperlinks*.

- *Web structure mining* (WSM); merupakan penemuan model yang berkaitan dengan struktur hubungan *web* yang meliputi *intra-page structure* dan *inter-page structure.*
- *Web Usage Mining* (WUM) ini yang menjadi fokus paper merupakan proses untuk mengaplikasikan teknik *data mining* dalam melakukan penemu-

an pengetahuan berupa pola penggunaan (*usage pattern*) dari *web*.

Adapun fungsi dari WUM adalah sebagai berikut (Pramudiono, 2004):

- *Personalization*; melakukan personalisasi *website* sesuai dengan keinginan user yang didasarkan dari perilaku penggunaan *web*.
- *System improvement*; meningkatkan performansi sebuah *web* sebagai tujuan untuk mendapatkan kepuasan bagi penggunaya. WUM menyediakan fasilitas kunci untuk memahami perilkau trafik *web*, hal ini akan dijadikan sebagai landasan membuat kebijakan *web chaching,* transmisi jaringan, *load balancing,* dan distribuís data.
- *Site modification*; menyediakan umpan balik (*feed back*) dari perilaku akses penguna terhadap statu *website* kepada *designer*. Hal ini dimaksudkan untuk membuat *website* yang adaptif dengan pola perubahan struktur *website* yang dinamis berdasarkan pola penggunaan.
- *Busines Intelligence*; menyediakan informasi bagaimana para pelanggan memanfaatkan *website* sebagai informasi yang fundamental bagi e-Commerce. Hal ini dijadikan sebagai proses penemuan pengetahuan (*knowledge discovery process*) sebagai *marketing intelligence* dari data *web*.
- *Usage Characterization*; menyediakan informasi tentang perilaku interaksi user dengan *website ini*, dalam konteks interaksi dengan *web* content dan atributnya serta dengan *web browser*. Hal ini dimaksudkan untuk meningkatkan performansi skalabilitas dan kemampuan *load balancing* di *server web*.

PEMBAHASAN

Dalam pembahasan dijelaskan mengenai tahap *preprocessing* sebagai tahap penyiapan data, algoritma Ant-WUM, dan pengujian. WUM terdiri dari proses utama sebagai berikut:

- Proses *preprocessing* meliputi proses konversi penggunaan (*usage*) yang ada dalam *web access log* ke level abstraksi data yang dibutuhkan dalam *pattern discovery.*
- *Pattern discovery* menggambarkan metode algoritma yang dibangun untuk melakukan penemuan pola penggunaan *web*, hal ini dengan menggunakan *Ant-Miner.*
- *Pattern analysis* merupakan tahapan terakhir dalam proses WUM, dimana dalam proses ini dilakukan penyaringan (*filter*) terhadap kaidah-kaidah atau pola yang tidak relevan dari kumpulan data yang ditemukan dalam tahapan *pattern discovery*.

Pada tahap *preprocessing,* telah dilakukan pengembangan suatu metode yang dapat menyiapkan data untuk tugas klasifikasi pengguna *web* dengan menggunakan algoritma *Ant-WUM.* Tahap *preprocessing* ini dapat dijelaskan sebagai berikut:

Format *web access log* standar seperti dalam format sebagai berikut:

```
127.0.0.1 - - [11/Jan/2009:13:32:21
+0700] "GET /xampp/xampp.css
HTTP/1.1" 200 4178
127.0.0.1 - - [11/Jan/2009:13:32:21
+0700] "GET /xampp/img/xampp-logo.jpg
HTTP/1.1" 200 19738
127.0.0.1 - - [11/Jan/2009:13:32:21
+0700] "GET /xampp/img/blank.gif
HTTP/1.1" 200 43
127.0.0.1 - - [11/Jan/2009:13:32:22
+0700] "GET /favicon.ico HTTP/1.1" 200
30894
127.0.0.1 - - [11/Jan/2009:13:32:26
+0700] "GET /xampp/lang.php?en
HTTP/1.1" 302 -
127.0.0.1 - - [11/Jan/2009:13:32:26
+0700] "GET /xampp/index.php
HTTP/1.1" 200 604
```

Dilakukan *parsing* dan pembersihan data sebagai berikut:

- Halaman URL harus tidak mengandung ekstensi gambar (*.png, *.jpeg, *.gif, dll).
- Status koneksi yang diambil hanya yang berkode 200 (akses ke halaman *web* sukses) dan 301 (melakukan transaksi *login* atau *logout.*

Melakukan ekstraksi data *session* tabel hasil *parsing web access log* untuk mendapatkan atribut jumlah akses, jumlah *login,* dan rata-rata *login* dalam format data kontinyu.

Melakukan diskritisasi data kontinyu menjadi data kategori, sehingga mendukung untuk kebutuhan algoritma *Ant-WUM.* Diskritisasi dilakukan meng-

gunakan aplikasi *data mining* WEKA dengan metode MDL Fayyad dan Irani.

Ektraksi data dilakukan untuk membuat tabel untuk mendapatkan informasi: Jumlah akses yang dilakukan user dalam periode tertentu (*A*), yang dihitung dari akumulasi ini penjumlahan waktu akses per halaman dalam kurun waktu tertentu, selama waktu akses per halaman lebih kecil daripada waktu *time out* yang didefinisikan oleh penguna (lihat Persamaan 1). Durasi akses yang dilakukan pengguna dalam periode tertentu (*D*), yang dihitung dari penjumlahan sekuensial waktu akses dalam rentang waktu yang didefiniskan oleh user (lihat Persamaan 2). Rata-rata waktu akses pengguna dalam periode tertentu ($\overline{D}$), yang dihitung dari durasi akses dibagi dengan jumlah waktu akses dalam rentang waktu yang didefinisikan oleh user (lihat Persamaan 3). Jumlah waktu login yang dilakukan pengguna dalam periode tertentu (*L*), yang dihitung dari sekuensial akses halaman sampai bertemu dengan halaman *LOGIN* dan waktu akses halaman tidak sama dengan waktu *time out* yang didefiniskan oleh user (lihat Persamaan 4). Rata-rata waktu *login* pengguna ($\overline{L}$), yang dihitung dari total waktu *login* (persamaan (5)) dibagi dengan jumlah penguna melakukan *login* dalam rentang waktu yang didefinisikan oleh pengguna (lihat Persamaan 6).

Untuk melakukan ekstraksi data di atas menggunakan formula sebagai berikut:

$$A(t) = \sum_{p=1}^{p=n} t_p, t_p < t_{out} \quad \text{.....} \quad (1)$$

$$D = A(t)_1 + A(t)_2 + A(t)_n \quad \text{.....} \quad (2)$$

$$\overline{D} = \frac{D}{\sum A(t)} \quad \text{.....} \quad (3)$$

$$L(t) = \sum_{p=1}^{p=n} t_p, t_p < t_{out}, p_n = \text{"LOGIN"} \quad \text{..} \quad (4)$$

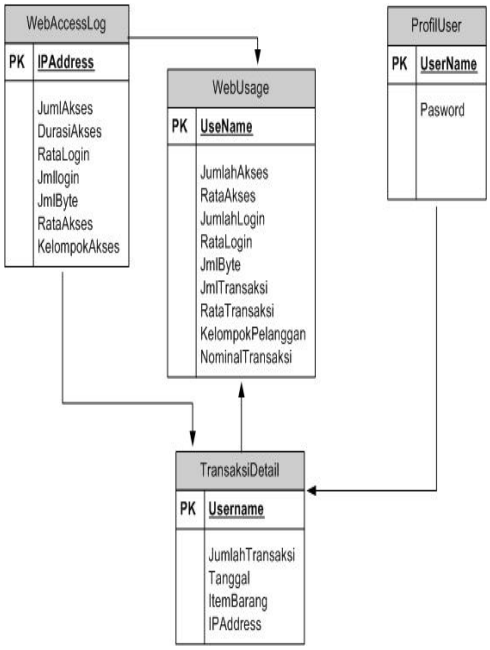
$$TotL = L(t)_1 + L(t)_2 + L(t)_n \quad \text{.....} \quad (5)$$

$$\overline{L} = \frac{TotL}{\sum L(t)} \quad \text{. ……..} \quad (6)$$

dimana

- t_p menunjukkan waktu akses per halaman
- t_{out} menunjukkan waktu *time out* sebagai acuan waktu toleransi masing-masing halaman *web*
- *p* menunjukkan halaman *web.*

Membangun relasi tabel *webacceslog* tersebut dengan tabel transaksi dan profil pengguna sebagaimana digambarkan sebagai berikut:



Gambar 1. Hubungan Antar Data WUM

Dari Gambar 1 dapat dijelaskan sebagai berikut: Tabel *webacesslog* akan digunakan sebagai masukan dalam algoritma *Ant-WUM* untuk menghasilkan kaidah klasifikasi pengguna *web* dalam kategori pengguna dengan akses frekuensi tinggi atau rendah. Dalam hal ini atribut yang akan digunakan sebagai kandidat *term* adalah jumlah akses, durasi akses, rata-rata akses, jumlah byte dan atribut kelompok akses sebagai kelas prediktor. Data pengguna yang direpresentasikan dalam *IP address* di tabel ini, merupakan daftar pengguna *web* yang belum melakukan transaksi pembelian dalam *web e-commerce*. Sedangkan *IP address* yang dipakai oleh pengguna *web* yang melakukan transaksi *login* dan atau pembelian akan dimasukan dalam tabel *webusage.*

Tabel *webusage* sebagai tabel yang dbentuk dari relasi *webaccesslog*,

profil pengguna, dan transaksi akan digunakan sebagai masukan dalam algoritma *Ant-Miner* untuk menghasilkan kaidah klasifikasi pengguna *web* dalam kategori pengguna potensial, retensi maupun baru. Atribut yang akan digunakan sebagai calon *term* dalam kaidah klasifikasi adalah jumlah akses, rata-rata akses, jumlah login, rata-rata login, jumlah *byte*, jumlah transaksi, rata-rata transaksi, nominal transaksi, dan kelompok pelanggan sebagai kelas prediktor.

Data *IP address* pengguna *web* dalam tabel *webusage* merupakan bagian dari data *IP address* dalam tabel *webaccesslog.* Hubungan antara keduanya dapat dijelaskan dalam persamaan 7 sebagai berikut:

$$\sum_{i=1}^{n} I_a \in \sum_{i=1}^{n} I_u \quad \text{.....} \qquad (7)$$

dimana:

- *i* menunjukkan urutan data IP *address*
- I_a menunjukkan daftar IP *address* dalam tabel *webusage*
- I_u menunjukkan daftar IP *address* dalam tabel *webaccesslog*.

Algoritma *Ant-WUM* merupakan pengembangan algoritma *Ant-Miner* dari sisi fungsi heuristik. Dalam pembahasan ini akan dijelaskan secara utuh mengenai hal tahapan-tahapan algoritma *Ant-WUM* yang mengadopsi algoritma *Ant-Miner*. Algoritma ini menggunakan pendekatan sekuensial untuk menemukan sejumlah kaidah klasifikasi untuk melingkupi data latih (*training data*). Iterasi pada pengulangan REPEAT-UNTIL pada *pseudo code a*lgoritma *Ant-Miner* terdiri dari tiga tahapan, yaitu pembuatan kaidah, *rule pruning* (pembuangan kaidah yang tidak sesuai) dan *updating pheromone*.

Pertama, Ant_t dimulai dengan kaidah kosong, yaitu kaidah yang yang tidak mempunyai *term* pada *antecedent*nya dan menambahkan satu *term* pada kaidah yang sedang dibangun. Kesamaannya, pilihan sebuah *term* yang akan ditambahkan pada kaidah yang berjalan terkait dengan pilihan penunjukan jalur yang akan dikembangkan. Pilihan *term* yang akan ditambahkan pada kaidah yang berjalan tergantung pada fungsi heuristik (η) dan nilai p*heromone* (τ). Ant_t akan selalu menambah sebuah *term* pada kaidah yang sedang berjalan dan akan berhenti sampai bertemu dengan kedua kriteria berikut ini:

- Beberapa *term* yang ditambahkan dalam kaidah yang mengakibatkan kaidah tersebut melingkupi kasus lebih kecil daripada nilai *threshold* yang didefinisikan user pada *Min_cases _ per_rule* (jumlah minimum dari kasus yang harus dilingkupi per kaidah).
- Semua atribut sudah digunakan oleh agen semut (*ant*), sehingga sudah tidak ada atribut yang akan ditambahkan dalam *antecedent*. Dalam hal ini berlaku aturan bahwa masing-masing atribut hanya digunakan satu kali dalam satu kaidah, hal ini untuk menghindari kaidah yang tidak valid, seperti "IF (Sex=male) AND (Sex= Female…"

Kedua, kaidah R_t ini yang dibangun oleh Ant_t dilakukan pembabatan (*pruning*) untuk memindahkan *term-term* yang tidak relevan. *Term-term* yang tidak relevan kemungkinan terjadi pada metode variasi stokastik pada prosedur pemilihan *term* dan atau pada fungsi heuristik yang hanya mengijinkan penggunaan satu atribut pada satu *term*.

Ketiga, jumlah *pheromone* pada masing-masing jalur diupdate, penambahan nilai *pheromone* pada jalur diikuti Ant_t (mengikuti kualitas R_t) penurunan nilai *pheromone* pada jalur lain (mensimulasikan penguapan *pheromone*) ini. Kemudian agen semut yang lain mulai membangun kaidah menggunakan jumlah *pheromone* yang baru untuk mengarahkan pencariannya.

Proses ini akan dilakukan secara berulang sampai dengan bertemu dengan salah satu dari kedua kondisi berikut ini: (1). Jumlah kaidah yang dibangun sama dengan atau lebih besar daripada nilai *threshold* pengguna *No_ of_ants*.(2) Ant_t eksisting telah membangun sebuah kaidah yang sama dengan kaidah yang telah dibangun *ant No_ru-les_ converg–1,* dimana *No_rules_ con-verg* merepresentasikan jumlah kaidah yang

digunakan untuk menguji konver gensi sekumpulan agen semut.

Ketika pengulangan REPEAT-UNTIL selesai, kaidah terbaik di antara kaidah-kaidah yang terbangun oleh semua agen semut akan ditambahkan dalam daftar kaidah penemuan dan sistem akan memulai pengulangan baru dengan WHILE dengan melakukan inisialisasi ulang semua jalur dengan jumlah *pheromone* yang sama.

Tahap pertama ini pengulangan REPEAT-UNTIL dalam algoritma *Ant-Miner* adalah sebuah agen semut eksisting secara iterasi menambahkan sebuah *term* pada kaidah yang sedang dibangun. Jika $term_{ij}$ merupakan sebuah kondisi kaidah dalam bentuk $A_i = V_{ij}$ dimana A_i merupakan atribut ke-I dan V_{ij} merupakan nilai ke=j pada domain A_i. Probabilitas $term_{ij}$ akan dipilih dan ditambahkan dalam kaidah dengan Persamaan 7 sebagai berikut:

$$P_{ij} = \frac{\eta_{ij} \cdot \tau_{ij}(t)}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b_i} (\eta_{ij} \cdot \tau_{ij}(t))} \quad \ldots\ldots\ldots \quad (8)$$

dimana:

- η_{ij}= nilai fungsi heuristik untuk $term_{ij}$ yang dihasilkan oleh persamaan (14). Semakin besar nilai η_{ij} maka semakin relevan untuk klasifikasi $term_{ij}$ semakin besar probabilitasnya untuk dipilih.
- $\tau_{ij}(t)$= jumlah *pheromone* yang diasosiakan dengan $term_{ij}$ pada ite-rasi *t* berkorespondensi dengan jumlah *pheromone* eksisting yang tersedia pada posisi jalur *i,j* yang diikuti oleh agen semut eksisting.
- *a* = jumlah atribut.
- x_i = bernilai 1 jika atribut A_i belum digunakan oleh agen semut eksisting, dan bernilai 0 jika sebaliknya.
- b_i = jumlah nilai dalam domain atribut ke-i.

Masing-masing $term_{ij}$ dapat ditambahkan dalam kaidah eksisting berdasarkan hasil komputasi nilai η_{ij} menggunakan fungsi heuristik untuk melakukan estimasi kualitas *term* ini dan meningkatkan tingkat akurasi prediksi kaidah. Fungsi heuristik yang digunakan berbasis teori informasi (Cover, 1991). Nilai η_{ij} untuk $term_{ij}$ mengandung sebuah ukuran entropi (jumlah informasi) yang diasosiasikan dengan *term* tersebut. Nilai entropi untuk masing-masing $term_{ij}$ dalam format $A_i = V_{ij}$ adalah sebagai berikut:

$$H(W|A_i = V_{ij}) = -\sum_{w=1}^{k} \left(P(w|A_i = V_{ij}) \cdot \log_2 P(w|A_i = V_{ij}) \right) \quad (9)$$

dimana:

- *W* adalah kelas atribut (atribut yang domainnya ini terdiri dari sekumpulan kelas yang diprediksi)
- *k* adalah jumlah kelas
- $P(w|A_i = V_{ij})$ adalah probabilitas empirik untuk observasi kondisi kelas *w* yang sesuai dengan $A_i = V_{ij}$

Semakin besar untuk peningkatan nilai $H(W|A_i = V_{ij})$ maka semakin seragam dapat didistribusikan ke kelas-kelas semakin kecil peluang agen semut eksisting menambah $term_{ij}$ tersebut ke dalam kaidah yang sedang dibangun. Hal ini memungkinkan untuk dilakukan normalisasi nilai fungsi heuristik untuk memfasilitasi penggunaanya pada Persamaan 1. Untuk mengimplementasikan normalisasi ini, maka perlu dibatasi bahwa nilai $H(W|A_i = V_{ij})$ berkisar pada rentang nilai $0 \le H(W|A_i = V_{ij}) \le \log_2 k$, dimana *k* adalah jumlah kelas. Normalisasi fungsi heuristik dapat dituliskan dalam persamaan sebagai berikut:

$$\eta_{ij} = \frac{\log_2 k - H(W|A_1 = V_{ij})}{\sum_{i=1}^{a} x_1 \cdot \sum_{j=1}^{b_i} \left(\log_2 k - H(W|A_1 = V_{ij}) \right)} \quad \ldots.(10)$$

dimana:

- a, x_i dan b_i mengandung arti yang sama dengan persamaan (1).
- Jika nilai V_{ij} pada atribut $A_i$tidak ada dalam data training $H(W|A_i = V_{ij})$ adalah berisi nilai maksimum yaitu $\log_2 k$ dengan demikian mempunyai peluang kecil untuk diprediksi dan jika semua kasus mempunyai kelas yang sama maka $H(W|A_i = V_{ij})$ diberi nilai 0 dan tentunya $term_{ij}$ tersebut mempunyai peluang yang besar untuk diprediksi masuk dalam kaidah.

Dari Persamaan 9 dan 10 di atas dapat dijelaskan dengan ilustrasi penerapan persamaan tersebut dalam contoh data latih PLAY, sebagai berikut:

Tabel 1. Tabel Play Untuk Penghitungan Heuristik Informasi

Out look	Temp	Humi-dity	Windy	Play
Sunny	85	85	False	Don't Play
Sunny	80	90	True	Don't Play
Over-cast	83	78	False	Play
Rain	70	96	False	Play
Rain	60	80	False	Play
Over-cast	64	65	True	Play
Sunny	72	95	False	Don't Play
Sunny	69	70	False	Play
Rain	75	70	True	Play
Sunny	75	70	True	Play
Over-cast	72	90	True	Play
Over-cast	81	75	False	Play
Rain	71	80	True	Don't Play

Untuk menghitung nilai informasi term "outlook= sunny" untuk kelas PLAY berdasarkan persamaan (9) dan (10), adalah sebagai berikut:

- P(Play|outlook=sunny= 2/14 = 0.143,
- P(Don't Play|outlook=sunny) = 3/14 = 0.214
- H(W,outlook=sunny)=-0.143.log(0.143)-0.214.log(0.214)=0.877
  $\eta = log_2 k - H(W, outlook = sunny) = 1 - 0.8777 = 0.123$

Fungsi heuristik dalam algoritma Ant-WUM ini didasarkan pada fungsi kedekatan (*closeness principal*) yang digunakan oleh (Grear, 2006) untuk melakukan klustering profil pengguna *web* dalam WUM. Fungsi ini adalah untuk mengukur jarak antara waktu yang digunakan oleh pengguna eksisting dalam mengakses halaman *web* dengan waktu akses yang telah ada dalam kelompok kluster. Adapun persamaan yang digunakan adalah:

$$Distance\ (t1, t2) = 1 - \cos(t2 - t1) \quad (11)$$

Dimana, *t*1 menunjukkan jumlah waktu yang digunakan oleh pengguna *web* eksisting dalam mengakses sekumpulan halaman *web*, dan *t*2 merupakan jumlah waktu dalam masing-masing kluster.

Waktu akses *web* adalah akumulasi waktu akses masing-masing halaman *web* yang bersifat sekuensial sebagaimana dalam Persamaan 1 Penggunaan fungsi heuristik ini untuk menyelesaikan permasalahan klasifikasi pengguna *web* dalam WUM, dikarenakan beberapa atribut data WUM sebagaimana dijelaskan dalam bagian *preprocessing* adalah sama dengan yang digunakan oleh (Grear, 2006) yaitu waktu *session* masing-masing halaman yang diakses oleh pengguna *web*. Perbedaannya adalah dalam penyiapan data untuk fungsi klasifikasi pengguna web dilakukan akumulasi jumlah waktu untuk masing-masing pengguna yang bersifat unik, sedangkan dalam (Grear, 2006) waktu *session* digunakan untuk masing-masing pengguna *web* dalam satu siklus akses *web*, tanpa memperhatikan eksistensi pengguna *web*, jadi satu pengguna *web* mempunyai beberapa total waktu *session* yang berbeda.

Implementasi Persamaan 11 untuk fungsi heuristik dalam algoritma *Ant-WUM* dapat diformulasikan dalam persamaan sebagai berikut:

$$D(H1, H2) = 1 - \cos(H2 - H1) \quad \ldots (12)$$

dimana:

- *D* merupakan jarak nilai heuristi H1 ke H2
- *H*1 merupakan nilai heuristik $term_{ij}$ dalam format $A_i = V_{ij}$, sebagaimana Persamaan 9.
- *H*2 merupakan suatu nilai heuristik $term_{notij}$ dalam format $A_i = V_{notij}$.

Nilai heuristic *H*2 ini diperoleh dari nilai informasi *term* yang berbeda dengan $term_{ij}$ pada kelas yang sama. Adapun persamaan untuk mengitung *H*2 adalah:

$$H2(W|A_i = V_{Nij}) = -\sum_{w=1}^{k} \left(P(w|A_i = V_{Nij}) \cdot \log P(w|A_i = V_{Nij}) \right) \quad \ldots 13)$$

dimana:

- W, k dan P adalah sama dengan deskripsi Persamaan 9
- V_{Nij} merupakan nilai atribut selain V_{ij} untuk kelas yang sama

Hasil penghitungan menunjukan perbandingan jarak yang dilakukan normalisasi dengan Persamaan 14 sebagai berikut:

$$\eta_{ij} = \frac{\log_2 k - H(W|A_i = V_{ij})}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b} \left(\log_2 k - H(W|A_i = V_{ij})\right)} / 1 - \cos(H2 - H1) \quad \ldots \quad (14)$$

Variabel pada normalisasi mempunyai arti yang sama pada Persamaan 10 dan 12. Hasil normalisasi ini yang menjadi nilai input dalam Persamaan 8.

Ilustrasi fungsi heuristik prinsip kedekatan algoritma *Ant-WUM* ini dapat dijelaskan contoh berikut. Untuk menghitung nilai jarak informasi term "outlook=sunny" untuk kelas PLAY berdasarkan prinsip kedekatan adalah adalah sebagai berikut:

- P(Play|outlook=sunny)=2/14= 0.143,
- P(Don't Play|outlook=sunny) = 3/14 = 0.214
- *H*1(W,outlook=sunny)=0.143.log(0.143)-0.214.log(0.214)=0.877
- P(Play|outlook=NOTsunny)=7/14=0.5
- P(Don't Play|outlook=NOTsunny) =1/14= 0.071
- *H*2(W,outlook=NOTsunny)=-0.5.log(0.5)-0.071.log(0.071)=0.232
- *D* (*H*1,*H*2) = 1 – cos(0.232 – 0.877)=0.200
- $\eta = log_2 k - H(W, outlook = sunny) / (1 - \cos(H2, H1) = 0.123 / 0.200 = 0.613$

Proses selanjutnya adalah *rule pruning.* Tujuan utama dari *rule pruning* adalah menghilangkan *term-term* yang tidak relevan yang akan dimasukkan dalam kaidah. Fungsi dari *rule pruning* adalah untuk meningkatkan kemampuan prediksi kaidah dan tingkat kemudahan kaidah sehingga mudah dipahami oleh pengguna. Prosedur *rule pruning* dieksekusi ketika agen semut eksisting selesai membangun kaidah. Prosedur ini dilakukan secara berulang sampai mendapatkan kualitas suatu kaidah yang didasarkan pada Persamaan 11.

Setelah melakukan *rule pruning,* maka dilakukan proses *updating pheromone.* Sebagaimana dijelaskan dalam algoritma *Ant-Miner* ini, bahwa semua $term_{ij}$ akan diinisialisasi dengan jumlah *pheromone* yang sama sehingga pada saat agen semut pertama melakukan pencarian, semua jalur mempunyai nilai *pheromone* yang sama. Inisialisasi jumlah *pheromone* yang disimpan dalam masing-masing jalur adalah berbanding proporsional dengan jumlah nilai semua atribut, dan didefinisikan dalam Persamaan 15 berikut ini:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \quad \ldots\ldots\ldots\ldots \quad (15)$$

Dimana *a* adalah jumlah atribut dan b_i adalah jumlah yang dimiliki atribut A_i.

Nilai dalam persamaan ini dinormalisasi untuk digunakan dalam Persamaan 7 yaitu kombinasi antara nilai ini dengan nilai fungsi heuristik. Ketika sebuah agen semut membangun kaidahnya dan kaidah itu di-*pruning,* maka jumlah *pheromone* dalam semua jalur harus di*update*. *Updating pheromone* dilakukan dengan dua hal, yaitu: Pertama, penambahan Jumlah *phe-romone* yang diasosiasikan dengan masing-masing $term_{ij}$ dalam kaidah yang ditemukan oleh agen semut (setelah proses *pruning*) akan ditambahkan sehingga secara proporsional dapat menambah kualitas kaidah. Kualitas suatu kaidah dinotasikan dengan *Q,* yang dihitungan dengan formula *Q* = *sensitifity* * *specifity* (Parpinelli, 2002) yang didefinisikan dengan persamaan sebagai berikut:

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN} \quad (16)$$

dimana:

- *TP* (*True Positive*) merupakan jumlah kasus yang dilingkupi oleh kaidah yang mempunyai kasus yang telah diprediksi oleh kaidah tersebut
- *FP* (*False Positive*) merupakan jumlah kasus yang dilingkupi oleh kaidah yang mempunyai kelas yang berbeda dengan yang diprediksi oleh kaidah
- *FN* (*False Negative*) merupakan jumlah kasus yang tidak dilingkupi oleh kaidah tetapi mempunyai kelas yang diprediksi oleh kaidah
- *TN* (*True Negative* merupakan jumlah kelas yang tidak dilingkupi oleh kaidah dan tidak mempunyai kelas yang diprediksi oleh kaidah tersebut

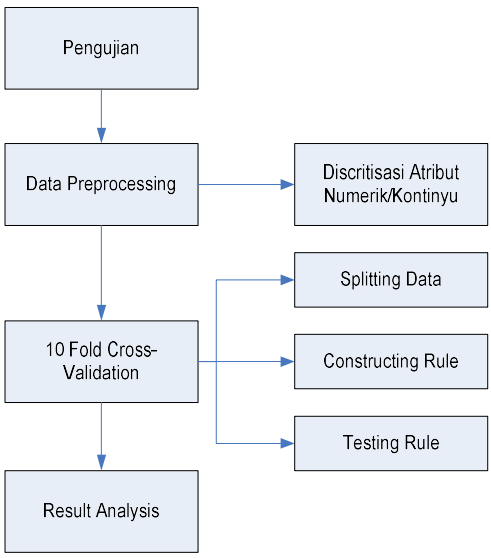
Nilai *Q* berada dalam rentang nilai $0 \leq Q \leq 1$, dimana semakin besar nilai

Q maka semakin besar kualitas sebuah kaidah. Updating *pheromone* untuk sebuah $term_{ij}$ dilakukan dengan Persamaan 17 sebagai berikut:

$$\tau_{ij}(t+1)=\tau_{ij}(t)+\tau_{ij}(t)\cdot Q, \forall i,j \in R \ldots. \quad (17)$$

Dimana *R* merupakan himpunan *term* yang terbentuk dalam kaidah yang dibangun oleh agen semut pada pengulangan *t.* Pengurangan Jumlah *Pheromone* yang diasosiasikan dengan masing-masing $term_{ij}$ yang tidak ada dalam kaidah dikurangi. Pengurangan jumlah *pheromone* untuk *term* yang tidak dipakai dilakukan dengan melakukan normalisasi nilai masing-masing *pheromone* τ_{ij} dengan melakukan penjumlahan semua $\tau_{ij}, \forall ij$. Pada waktu normalisasi, jumlah *pheromone term* yang tidak dipakai dan akan dihitung dengan membagi nilai eksisting ini dengan total penjumlahan *pheromone* semua *term*.

Pengujian ini dilakukan dengan skenario seperti Gambar 2: Sedangkan data yang digunakan dalam pengujian ini adalah *data repository* UCI (University of California at Irvine) yang dapat diakses di http://www. ics.uci.edu/~mlearn/ MLRepository. html, data *e-commerce.* Seting parameter algoritma *Ant-WUM* dan *Ant-Miner* untuk melakukan pengujian ini adalah sebagai berikut:



Gambar 2. Skenario Pengujian

Tabel 2. Setting Parameter Algoritma

Folds	10
Number of Ants	5
Min-cases Per Rule	5
Max-uncovered cases	10
Rules of convergence	10
Number of Iterations	100

Tabel 3. Nama Data UCI dan Data WUM Untuk Pengujian

Nama Data	Jumlah Atribut	Jumlah Data
Breast Cancer	10	286
Diabetes	9	768
Lymph	19	148
Breast W	10	699
Hepatitis	20	155
WUM	9	498

Hasil pengujian terhadap data UCI dan WUM dapat digambarkan sebagai berikut: Tabel 4. Tingkat Akurasi yang dihasilkan dari Ant-Miner dan Ant-WUM Pada Data Test UCI & WUM.

Adapun contoh kaidah klasifikasi pengguna *web* yang terbangun adalah sebagai berikut:

IF durasiakses = '\(120.1-126.8]\' THEN 'Potensial'.
IF jmlakses='\(11.3-12.2]\' THEN 'Potensial'; IF jmlakses = '\(9.5-10.4]\' THEN 'Poten-sial';IF JK = 'F' THEN 'NP'
IF JK = 'M' THEN 'NP'
Default rule: NP

Pada Tabel 4 menunjukkan bahwa algoritma *Ant-WUM* menghasilkan kaidah dengan tingkat akurasi yang lebih tinggi pada empat data uji coba,

Tabel 4 Rata-rata Tingkat Akurasi Pengujian Ant- WUM

Nama Data	Average Predictive Accuracy (%)	
	Ant-Miner	Ant-WUM
Breast Cancer	74.15 ± 2.28	76.15 ± 2.72
Diabetes	68.23 ± 1.93	67.98 ± 1.76
Lymph	72.46 + 5.34	73.69 ± 3
Breast W	91.84 ± 0.88	92.12 ± 1.4
Hepatitis	83.82 ± 2.24	78.48 ± 3.38
WUM-Potensial	88.72 ± 1.32	88.78 ± 1.74

Tabel 5. Rata-rata Jumlah Kaidah Yang Dihasilkan Ant-Miner &Ant-WUM pada Data Test UCI & WUM

Nama Data	Average Number of Rules	
	Ant-Miner	Ant-WUM
Breast Cancer	6.4 ± 0.16	6.3 ± 0.21
Diabetes	9.2 ± 0.25	8.5 ± 0.37
Lymph	5.9 ± 0.31	6.8 ± 0.2
Breast W	12.4 ± 0.31	12.1 ± 0.23
Hepatitis	5.1 ± 0.18	5.6 ± 0.16
WUM-Potensial	6 ± 0	6 ± 0

Tabel 6. Rata-rata Jumlah Term per Kaidah yang Dihasilkan Ant-Miner & Ant-WUM pada Data Test UCI & WUM

Nama Data	Average Number of Terms Per Rule	
	Ant-Miner	Ant-WUM
Breast Cancer	9.2 + 0.44	9.4 ± 0.56
Diabetes	8.5 ± 0.27	7.8 ± 0.39
Lymph	9.9 ± 0.75	11.8 ± 0.9
Breast W	12.4 ± 0.4	12.2 ± 0.51
Hepatitis	9.1 ± 0.5	9.7 ± 0.4
WUM-Potensial	5 ± 0	5 ± 0

yaitu pada data Breast Cancer, Lymph, Breast-W dan data WUM. Sedangkan pada data Hepatitis dan Diabetes, algoritma *Ant-Miner* menghasilkan tingkat akurasi yang lebih tinggi dibandingkan algoritma Ant-WUM.

Sedangkan pada Tabel 5 dan Tabel 6 menunjukkan bahwa algoritma Ant-WUM menghasilkan kaidah yang lebih sederhana pada data Breast Cancer, Diabetes, dan Breast-W, sedangkan pada data WUM kedua algoritma ini menghasilkan kaidah yang sama dari sisi simplifikasi. Algoritma *Ant-Miner* menghasilkan kaidah yang lebih sederhana pada data Lymph dan Hepatitis.

KESIMPULAN

Dari hasil uji coba menunjukkan, bahwa *algoritma Ant-WUM* yang menggunakan fungsi kedekatan dan pemanfaatan teori informasi untuk menentukan nilai masing-masing heuristik yang dihitung jaraknya, menghasilkan tingkat performansi yang cukup kompetitif dengan algoritma *Ant-Miner*. Fungsi kedekatan sebagai heuristik WUM merupakan metode untuk menyelesaikan masalah klasifikasi pengguna *web* dalam WUM. Heuristik ini juga digunakan dalam fungsi klusterisasi WUM. Yang membedakan keduanya adalah dari sisi variable yang dihitung nilai jaraknya.

Pada algoritma *Ant-WUM* ini, teori informasi masih digunakan untuk mengukur nilai informasi antar dua nilai heuristik. Dengan memadukan pemanfaatan fungsi kedekatan dan teori informasi dalam algoritma Ant-WUM telah menghasilkan kaidah klasifikasi yang mempunyai tingkat akurasi dan tingkat simplifikasi yang lebih tinggi pada enam data uji coba di atas.

Saran, dalam rangka pengembangan riset ini di masa mendatang, diusulkan pengembangan dalam dua hal, yaitu:

- Pemanfaatan Ant-WUM untuk menghasilkan kaidah dari atribut kontinyu secara langsung, sehingga tidak perlu didiskritkan pada tahap *preprocessing*.
- Pemanfaatan fungsi heuristik berbasis prinsip kedekatan dengan menggunakan fungsi selain teori informasi dalam menghitung nilai antar dua vatiable (H1 dan H2). Uji coba ini diharapkan dapat menghasilkan kaidah klasifikasi dengan tingkat akurasi yang lebih tinggi

DAFTAR PUSTAKA

Abdurrahman, et al, 2006, *Pemodelan Data Webhouse sebagai Tahap Preprocessing Web Usage Mining untuk Business Intelligence*, Konferensi Nasional Sistem Informasi, Universitas Pasundan – Bandung.

Abdurrahman, 2004, *Pemodelan Customer Churn Management Berbasis CRM Studi Kasus PT. Telekomunikasi Indonesia, Tbk,* Tesis Magister Teknik Informatika, ITB.

Abraham, A., 2003, *Business Intelligence From Web Usage Mining*, Journal of Information & Knowledge Management, Vol. 2, No. 4, p. 375-390.

Abraham, A., et al, 2005, *Web Usage Mining Using Artificial Ant Colony Clustering and Linear Genetic Programming*, cs.okstate.edu

Bonabeu, E., et al, 1999, *Swarm Intelligence: From Natural to Artificial Systems,* New York, NY, Oxford University Press.

Cover, T.M., et al, 1991, *Elements of Information Theory,* New York, NY, John Willey & Sons.

Dorigo, M., et.al, 1999, *The Ant Colony Optimization Meta-heuristik,* new Ideas in Optimization, D.Corn, M. Dorigo and F. Glover Eds. London, McGrawHill, p.11 -32.

Grear, M., 2006, *User Profiling : Web Usage Mining*, http://www.ijs.si.

Holden, N., et.al, 2007, *Web Page Classification with an Ant Colony Optimization*, kent.ac.uk.

J.R Quinlan, 1993, C4.5: *Programs for Machine Learning,* San Fransisco, CA : Morgan Kaufmann.

Jaideep, S., et al, 2000, *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data*, ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data

Lopes, 1998, *An Evolution Approach to Simulate Cognitive Learning in Medical Domain in Genetic Algorithm and Fuzzy Logic Systems: Soft Computing Perspctive,* Singapore World Scientific, p. 193-207.

Parpinelli, R.S, et al, 2002, *Data Mining with an Ant Colony Optimization Algorithm*, IEEE Transaction on Evolutionary Computation, special issue on Ant Colony Algorithm, v.6, p.321-332.

Pramudiono, I., 2004, *Parallel Platform for Large Scale Web Usage Mining*, Tesis Ph.D, Universitas Tokyo.

Padmajavalli, R., 2006, *An Overview of Data Pre-Processing in Web Usage Mining*, The ICFAI Journal of Information Technology, Vol. 2, No. 3, pp. 55-66.

Ramadhan, H., et al, 2005, *A Classification on Techniques for Web Usage Analysis*, Journal of Computer Science 1(3), p. 413-418, Science Publication.

Spiliopoulou, M., et al, 2007, *A Data Miner Analyzing the Navigational Behavior of Web Users*, wiwi.huberlin.de.