

# ANALISIS DAN IMPLEMENTASI *WEB USAGE MINING* MENGUNAKAN ALGORITMA *SEQUENTIAL PATTERN* *DISCOVERY USING EQUIVALENCE CLASSES SPADE* (Studi Kasus: *Website iGracias Universitas Telkom*)

<sup>1)</sup>Asri Inna Khoirun Nissa, <sup>2)</sup>Ibnu Asror, <sup>3)</sup>Gede Agung Ary W

<sup>1,2,3)</sup>Fakultas Informatika

Universitas Telkom

Jl. Telekomunikasi No.1

<sup>1)</sup>[asriinna@gmail.com](mailto:asriinna@gmail.com), <sup>2)</sup>[iasror@telkomuniversity.ac.id](mailto:iasror@telkomuniversity.ac.id), <sup>3)</sup>[ary.wisudiawan@telkomuniversity.ac.id](mailto:ary.wisudiawan@telkomuniversity.ac.id)

## Abstrak

*Website Igracias Universitas Telkom* merupakan salah satu *website* di universitas Telkom yang sering digunakan oleh seluruh entitas Universitas Telkom. Pola pengunjung web dapat digunakan untuk mengetahui halaman apa saja yang telah dikunjungi oleh *user* dalam suatu *website*. Salah satu ilmu yang mempelajari pola navigasi *user* agar mendapatkan suatu informasi yang bermanfaat adalah *web usage mining*. Pada penelitian ini algoritma yang digunakan adalah *sequential pattern discovery using equivalence classes* (SPADE). Algoritma SPADE diterapkan untuk mencari pola perilaku pengguna *website* dengan cara melakukan *preprocessing* data untuk menyaring informasi yang dibutuhkan. Dilanjutkan dengan pembentukan data transaksi dan perhitungan SPADE dengan mengkombinasikan *itemset* dan menghitung *frequent*-nya untuk mendapatkan *rule* yang kemudian dicari kekuatan setiap *rule* dengan menghitung *lift ratio*-nya.

Kata kunci : *web usage mining*, SPADE.

## Abstract

*Igracias Telkom University is one of the Telkom University website that are frequently used by the entire entity of the by all entities that exist at Telkom University. The pattern of the web can be used recursively to find out what pages have been visited by the user on a website. One of the study of the patterns of user navigation in order to obtain a useful information is a web usage mining. On this final project, the algorithm used is a sequential*

*pattern discovery using equivalence classes (SPADE). The SPADE algorithm is applied to search behavior patterns by preprocessing data to find out the useful information or knowledge. Proceed with the establishment of the data transaction and calculation of the SPADE by combining itemset and calculate the frequent to get a rules which will then look for the strength of each rule by calculating the lift ratio.*

Keywords : *web usage mining*, SPADE

## I. PENDAHULUAN

Sebuah pengembangan pesat telah terjadi di dunia internet. Internet sekarang ini memiliki akses informasi yang jauh lebih luas dan fungsi yang lebih beragam. Perkembangan internet juga diiringi dengan pemanfaatan *website* dalam berbagai bidang. Terutama dibidang pendidikan seperti penggunaan *website* kampus untuk menampung segala informasi dan aktivitas *user* yang ada pada lingkungan tersebut. Untuk meningkatkan kualitas dari sebuah *website*, aktivitas *user* ini dapat digunakan untuk melihat pola navigasi *user* dengan menggunakan teknik *web mining*. *Web mining* pada dasarnya merupakan sebuah teknik untuk penggalian data yang berhubungan dengan *web*. *Web mining* terbagi menjadi tiga yaitu, *web content mining*, *web structure mining* dan *web usage mining*. Pada *web usage mining* atau biasa disebut dengan *web log mining* digunakan untuk menganalisis pola penggunaan *user* pada *web* yang berfungsi untuk pengembangan aplikasi *web*. Terdapat beberapa teknik untuk menentukan aturan

atau pola pada *web mining* diantaranya *statistical analysis*, *association rules*, *sequential rules*, *clustering* dan *classification* (Mehak, Kumar, & Aggarwal, 2013).

Pada penelitian ini dilakukan *web mining* pada *website* Igracias Universitas Telkom. *Website* Igracias merupakan sebuah *website* pendidikan yang digunakan oleh universitas Telkom. *Website* ini menampung semua data atau informasi dari setiap aktivitas elemen kampus diantaranya dosen, mahasiswa dan pegawai. Untuk mendukung aktivitas user ini, perlu dilakukan analisa untuk perbaikan dan pengembangan yang dapat dijadikan bahan evaluasi terhadap *website* igracias dengan menggunakan *event log*. Dikarenakan data *event log* ini merupakan data yang sekuensial atau berkelanjutan maka untuk memecahkan masalah ini digunakan sebuah metode yang bernama *sequential pattern mining* (Mooney & Roddick, 2013). *Sequential pattern mining* digunakan untuk mencari data yang memiliki urutan, data tersebut bisa merupakan urutan transaksi. Salah satu algoritma dari *Sequential Pattern Mining* adalah algoritma *Sequential Pattern Discovery Using Equivalence Classes* (SPADE) (Zaki, 2001). Algoritma *sequential pattern discovery using equivalent classes* (SPADE) merupakan salah satu algoritma dari *sequential pattern mining* yang digunakan untuk menemukan pola agar mendapatkan informasi yang berguna.

Algoritma SPADE menghasilkan performa yang jauh lebih baik dalam hal waktu komputasi dibandingkan algoritma-algoritma pencari frequent sequences sebelumnya seperti algoritma Apriori dan GSP (*Generalized Sequential Pattern*) (Gunawan, 2015). Dengan latar belakang tersebut maka metode *sequential pattern mining* dengan algoritma SPADE diharapkan cocok untuk menemukan pola perilaku pengguna *website* Igracias Universitas Telkom..

## II. KAJIAN LITERATUR

### II.1 Web Mining

*Data mining* merupakan sebuah proses yang digunakan untuk menemukan sebuah pola dalam data yang besar. Penerapan *data mining* dalam teknik data *web* untuk menemukan pengetahuan yang biasa disebut dengan *web mining* (Vijayarani & Suganya, 2015). Data yang sebenarnya lebih bervariasi dan memiliki pendekatan yang berbeda. Beberapa peneliti

menerapkan teknik pada *web logs* yang dikelola oleh server yang dapat digunakan untuk menemukan *user access* dan *traversals path*.

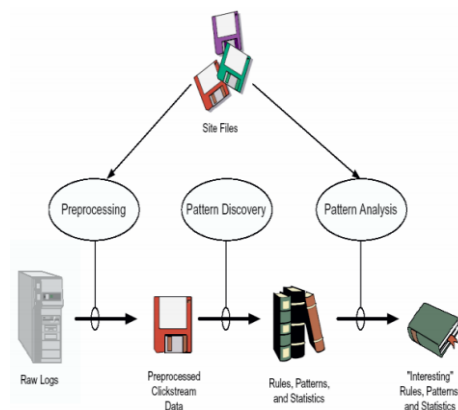
*Web mining* dikategorikan menjadi tiga jenis, yaitu:

1. *Web Content Mining*: melakukan *scanning* dan penambangan data teks, gambar dari sebuah halaman *web* untuk menentukan relevansi dari sebuah konten pencarian. Penelitian dibidang ini menggunakan teknik dari disiplin ilmu lain seperti *Information Retrieval* (IR) dan *Natural Language Processing* (NLP).
2. *Web Structure Mining*: merupakan alat yang digunakan untuk mengidentifikasi hubungan antar halaman *web*. Motif dari *structure mining* adalah menghasilkan ringkasan terstruktur mengenai informasi dari sebuah halaman *web*.

*Web Usage Mining*: memungkinkan pengumpulan informasi pada halaman *web*. Data pada *web usage mining* mengarah pada halaman *web* yang diakses. Dan *web server* menyimpan segala informasi ini secara otomatis dalam sebuah *access log*.

### II.2 Web Usage Mining

*Web Usage Mining* digunakan untuk menganalisis *web log files* untuk menemukan pola pengguna dalam mengakses sebuah halaman *web*. *Web usage mining* merupakan fokus utama dalam *web mining* yang mempelajari interaksi dari pengguna *web*. *Web usage mining* digunakan untuk menemukan pola yang bermakna dari data yang dihasilkan oleh transaksi *client-server* pada satu atau lebih *web server* (Mehak et al., 2013).



### Gambar 1. Proses pada Web Usage Mining

Web usage mining dikategorikan ke dalam tiga fase (Valera & Chauhan, 2013):

1. *Preprocessing*: Berdasarkan klien, server dan server proxy, *preprocessing* adalah pendekatan pertama untuk mengambil data mentah dari sebuah *web* dan memproses data. Secara otomatis mengubah data mentah untuk digunakan pada proses selanjutnya.
2. *Pattern Discovery*: Berdasarkan dari data *preprocessing*, data mentah yang ada digunakan untuk menemukan pengetahuan dan menerapkan teknik yang akan digunakan pada *machine learning* yang menggunakan prosedur dari data mining.

*Pattern Analysis*: *Pattern Analysis* adalah proses setelah penemuan pola. *Pattern Analysis* memeriksa apakah pola yang ada pada *web* benar dan memandu proses ekstraksi informasi atau pengetahuan dari *web*.

### II.3 Preprocessing

Beberapa *database* memiliki data yang tidak konsisten dan terdapat *noise*. *Preprocessing* digunakan untuk mentransformasi data tersebut. Hasil dari *preprocessing* membuat data menjadi lebih terintegrasi dan konsisten sehingga membentuk *database* yang stabil. Beberapa tahapan yang ada pada *preprocessing* (Varnagar, Madhak, Kodinariya, & Rathod, 2013):

1. *Data cleaning*: proses mengidentifikasi, memilih dan menghapus data dari hal-hal yang tidak diperlukan atau tidak relevan.
2. *User and Session Identification*: tugas utama pada tahap ini adalah untuk mengidentifikasi user yang berbeda sesi dari akses log. Beberapa metode digunakan untuk mengidentifikasi sesi dari alamat IP yang berbeda untuk membedakan user.

### II.4 Event Log

*Event log* merupakan catatan history yang memuat rangkaian aktivitas yang dilakukan pengguna dalam sebuah sistem informasi (Hermawan, 2014). *Event log* pada sistem informasi memuat catatan proses yang dilakukan oleh tiap pengguna. Setiap

*event log* juga memiliki berbagai atribut seperti *case\_id*, *activity*, *time*, *cost* dan *resource*. Namun tidak semua atribut tersebut selalu ada pada suatu *event log*. Misalnya adalah tidak adanya atribut *resource* maupun *cost*, tergantung dari data yang diperoleh dari suatu sistem informasi.

### II.5 Event Log

Sampai saat ini telah dikenal beberapa algoritma yang dikembangkan khusus untuk mendapatkan pola sekuensial. Masing-masing algoritma menggunakan pendekatan yang berbeda-beda. Algoritma apriori *Sequential Pattern Discovery using Equivalence Classes* adalah salah satu algoritma *sequential pattern mining* yang menggunakan format data vertikal pada *database sequence*. Dalam format data vertikal, *database sequence* menjadi berbentuk kumpulan urutan yang formatnya [*itemset*: (*sequence\_ID*, *event\_ID*)], dengan kata lain, untuk setiap *itemset* akan disimpan *sequence identifier* dan *event identifier* yang berkoresponden. *Event Identifier* berguna sebagai *timestamp* atau penanda waktu dari *itemset* tersebut. Sepasang (*sequence\_ID*, *event\_ID*) untuk setiap *itemset* membentuk *ID\_list* dari *itemset* tersebut (Zaki, 2001).

Langkah-langkah algoritma SPADE dalam mencari *frequent sequence* kemudian menentukan *rule* dari *frequent sequence* tersebut adalah sebagai berikut (Zaki, 2001) :

1. Menghitung *frequent 1-sequence* Untuk mencari *frequent 1-sequence* yang harus dilakukan pertama kali adalah mendaftar semua *item* yang terdapat pada *database* transaksi. Setiap *item* akan dicari nilai *support*-nya dengan cara menjumlahkan SID yang memiliki *item* tersebut. Jumlah dari SID merupakan nilai *support* dari *item* tersebut. Setelah semua *item* ditemukan nilai *support*-nya maka, *item* yang memiliki nilai *support* lebih dari sama dengan nilai minimum *support* merupakan *item frequent 1-sequence*.
2. Menghitung *frequent 2-sequence* Dalam mencari *frequent 2-sequence*, data yang digunakan adalah data dari *frequent 1-sequence*, sehingga tidak perlu mencari dari *sequence database* lagi. Untuk setiap masing-masing *frequent 1-sequence* akan digabungkan dengan semua *frequent 1-sequence* lainnya. Contohnya jika *1-sequence A* digabungkan dengan *1-sequence B* maka kemungkinan 2 *sequence* yang terjadi

adalah A,B dimana A dan B muncul bersamaan dalam satu transaksi (satu SID, satu EID), A->B dimana item B muncul setelah item A, dan B->A dimana item A muncul setelah item B (satu SID, beda EID). Untuk setiap masing-masing penggabungan *frequent 1-sequence* ini dilakukan pengecekan apakah eid dari *1-sequence A* sama dengan, kurang dari atau lebih dari eid *1-sequence B*. apabila sama maka id-listnya dimasukkan dalam *2-sequence A,B*. jika eid B lebih besar dari A maka id-listnya dimasukkan dalam *2-sequence A->B* dan jika eid A lebih besar dari B maka id-listnya dimasukkan dalam *2-sequence B->A*. kemudian seperti dalam *frequent 1-sequence id-list* akan ditambahkan supportnya untuk setiap masing-masing sid yang sebelumnya belum ditemui. Dari *2-sequence* itu kemudian dilakukan pengecekan apakah *support*-nya lebih dari nilai minimum *support*. Jika memenuhi syarat maka dimasukkan dalam *frequent 2-sequence*.

3. Menentukan *frequent k-sequence* Setelah mencari *frequent 2-sequence*, untuk mencari *frequent sequence- frequency sequence* berikutnya dilakukan proses yang sama, yaitu mencari *frequent k-sequence*. Untuk mencari *frequent k-sequence* ini dilakukan join pada *frequent (k-1) sequence* yang memiliki *prefix* yang sama. Contohnya untuk mencari *3-sequence* kita digabungkan *frequent sequence* dari *2-sequence* yang memiliki *prefix* yang sama, untuk mencari *4-sequence* digabungkan *frequent sequence* dari *3-sequence* yang memiliki *prefix* yang sama, dan seterusnya. Untuk mencari *prefix frequent (k-1), sequence* akan menghilangkan item terakhir dari *sequence* tersebut. Contoh jika terdapat *4-sequence A -> B-> C -> D*, maka *prefix*nya adalah *A->B->C*. untuk setiap penggabungan ini ada 3 kemungkinan hasil:
  - a. Jika A,B digabungkan dengan A,C maka kemungkinan hasilnya hanya A,B,C.
  - b. Jika A,B digabungkan dengan A->C, maka kemungkinan hasilnya A,B->C
  - c. Jika A->B digabungkan dengan A->C, maka ada 3 kemungkinan hasil: A->B, C, dan A->B->C dan A->C->B

Dari setiap kemungkinan ini kemudian mengecek nilai supportnya apakah memenuhi *min\_sup* atau tidak. Jika ya maka *sequence* itu

termasuk dalam *frequent k-sequence*. Pencarian *frequent sequence* dihentikan apabila tidak ada *frequent (k-1) sequence* yang bisa digabung atau sudah tidak ditemukan *frequent k-sequence* lagi.

Pembentukan *Rule* Setelah ditemukan semua *frequent sequence*, ditentukan *rule* dari *sequencesequence* tersebut. *1-sequence* tidak digunakan untuk membentuk *rule* karena hanya terdiri dari 1 item. Untuk *2-sequence* yang menjadi *antecedent* adalah item pertama dan consequentnya adalah item kedua. Contohnya untuk *sequence A->B* maka *rule* yang dibentuk adalah  $A \Rightarrow B$ . sedangkan untuk *sequence* yang panjangnya lebih dari 2 atau *k-sequence*, yang dijadikan *consequent* adalah item terakhir, sedangkan *antecedent*nya adalah semua item sebelum item terakhir. Contohnya pada *4-sequence A->B->C->D*, maka *rule* yang dihasilkan adalah  $A \Rightarrow B \Rightarrow C \Rightarrow D$ . Untuk masing-masing *rule* dihitung nilai *confidence*-nya serta nilai *support*. *Rule* yang memenuhi atau melebihi dari nilai minimum *confidence* dan minimum *support* merupakan *rule* yang digunakan dan kemudian untuk dihitung lift rasionya untuk menentukan *rule* mana yang paling kuat.

## II.5.1 SUPPORT AND CONFIDENCE

*Support* merupakan ukuran yang menunjukkan besar tingkat dominasi suatu *item* atau *itemset* keseluruhan transaksi. *Support* digunakan untuk *rule* yang memiliki nilai kurang dari *threshold* (Yunianto et al., n.d.). Untuk menentukan support dapat dilihat pada persamaan 1:

$$Supp(A \rightarrow B) = \frac{\text{Jumlah transaksi itemset}(A \rightarrow B)}{\text{Total transaksi}} \quad (1)$$

*Confidence* adalah ukuran yang menunjukkan hubungan antara dua *item* secara kondisional. *Confidence* adalah rasio antara jumlah transaksi yang meliputi semua item dalam *antecedent* dan *consequent* dengan jumlah transaksi meliputi semua item dalam *antecedent* (Yunianto et al., n.d.). Untuk mencari nilai *confidence* dapat dilihat pada persamaan 2:

$$Conf(A \rightarrow B) = \frac{\text{Jumlah Transaksi itemset}(A \rightarrow B)}{\text{Jumlah antecedent A}} \quad (2)$$

### II.5.2 LIFT RATIO

*Lift ratio* dapat digunakan untuk mengetahui kuat tidaknya sebuah *rule*. *Lift ratio* dapat dihitung dengan cara membandingkan nilai *confidence* dengan nilai *benchmark confidence*. Nilai dari *benchmark confidence* setiap *rule* didapatkan dari nilai jumlah transaksi *consequent* dibagi dengan jumlah transaksi. Untuk mencari nilai *confidence* dapat dilihat pada persamaan 3, sedangkan rumus untuk mencari nilai *lift ratio* dapat dilihat pada persamaan 4 (Yunianto et al., n.d.).

$$\text{Benchmark Confidence} = \frac{nc}{n} \quad (3)$$

nc = Jumlah transaksi item yang menjadi consequent.

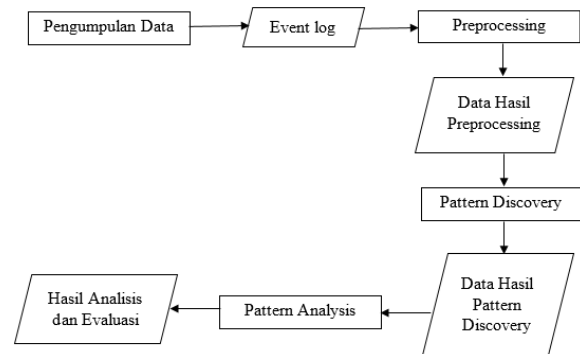
n = Jumlah transaksi basis data

$$\text{Lift Ratio} = \frac{\text{Confidence}}{\text{Benchmark Confidence}} \quad (4)$$

## III. ANALISIS DAN PERANCANGAN

### III.1 Gambaran Umum Sistem

Sistem yang akan dibangun merupakan sebuah sistem yang menerima input berupa *event log* dari Igracias Universitas Telkom. Data tersebut akan melalui tahap *preprocessing* untuk membersihkan data yang tidak dibutuhkan agar sesuai dengan kebutuhan sistem. Selanjutnya data hasil *preprocessing* masuk pada tahapan *pattern discovery* dengan algoritma *Sequential Pattern Mining Using Equivalence Classes (SPADE)*. Dan hasil dari proses *pattern discovery* adalah frekuensi pola pengaksesan *user*. Setelah mendapatkan hasil dari *pattern discovery*, dilakukan *pattern analysis* yaitu dengan membentuk *rule* yang diuji menggunakan nilai *lift ratio*, apabila nilai *lift ratio* lebih dari 1 maka *rule* dianggap kuat. Adapun tahapan proses sistem dapat dilihat pada gambar 2 dibawah ini.



Gambar 2 Tahapan Proses Sistem

### III.2 Gambaran Umum Sistem

Berikut adalah hasil pengujian yang dilakukan sesuai dengan skenario yang telah ditentukan dan disertai dengan analisis hasil pengujian.

- a. Pengujian dengan membandingkan perhitungan manual dan perhitungan sistem

Pengujian dengan membandingkan perhitungan manual dengan perhitungan sistem dilakukan dengan menggunakan beberapa sampel data dan kemudian dihitung akurasi secara manual dengan algoritma yang digunakan pada sistem. Data sample yang digunakan yaitu terdiri dari 36 *data record* yang diambil dari data mahasiswa. Dari data tersebut kemudian dilakukan *preprocessing*, *pattern discovery* dan *pattern analysis*. Dari *data sample* yang ada didapatkan 11 eksekutor yang berbeda dengan 14 jenis aktivitas. Pengujian pada perhitungan manual menghasilkan perhitungan seperti pada tabel 1.

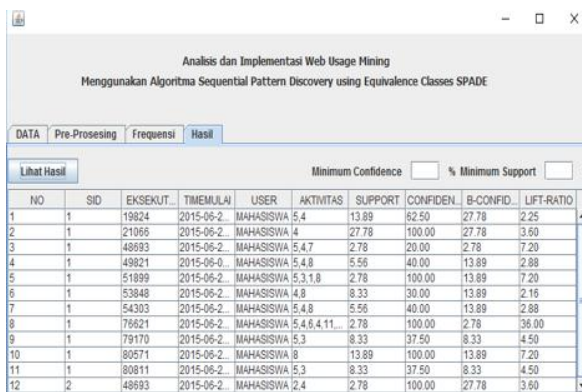
Pengujian perhitungan sistem dilakukan dengan *sample data* yang sama yaitu yang terdiri dari 36 *data record*. Output dari evaluasi ini menggambarkan bahwa perhitungan dari sistem dengan perhitungan manual yang dilakukan oleh penulis menghasilkan *output* yang sama. Hasil dari perhitungan sistem ditunjukkan pada gambar 3.

**Tabel 1 Hasil Perhitungan Manual SPADE pada Sample Data**

ID	Eksekutor	Support	Confidence	B-Conf	Lift Ratio
1	19824	14%	63%	28%	2,25
2	21006	28%	100%	28%	3,6
3	48693	3%	20%	3%	7,2
3	48693	3%	100%	28%	3,6
4	49821	6%	40%	14%	2,88
5	51899	3%	100%	14%	7,2
6	53848	8%	30%	14%	2,16
7	54303	6%	40%	14%	2,88
8	76621	3%	100%	3%	36
9	79170	8%	38%	8%	4,5
10	80571	14%	100%	14%	7,2
11	80811	8%	38%	8%	4,5

yaitu mahasiswa, dosen dan pegawai. Pengujian ini dilakukan untuk mendapatkan *minimum support* dan *minimum confidence* yang sesuai untuk pengujian selanjutnya.

Untuk data mahasiswa pada tabel 2 akan digunakan *Minimum support* 5%, 10%, dan 20% dan dengan nilai *minimum confidence* 60%, 70% dan 80%. Dari hasil uji coba data mahasiswa, perubahan yang terjadi tidak terlalu besar. Pada percobaan dengan *Minimum support* 5%, terdapat 6 jenis aktivitas. Pada *Minimum support* 10% terdapat 3 jenis aktivitas. Dan pada *Minimum support* 20% terdapat 2 jenis aktivitas. Dari ketiga percobaan tersebut terdapat 3 menu yang selalu ada. Namun terdapat 3 jenis aktivitas yang berbeda pada *minimum support* 5%. Untuk itu dari hasil percobaan tersebut diambil *minimum support* 5% untuk data mahasiswa. Dan pada percobaan dengan *minimum confidence* 60%, terdapat 6 jenis aktivitas. Pada *minimum confidence* 70% terdapat 6 jenis aktivitas. Dan pada *minimum confidence* 80% terdapat 5 jenis aktivitas. Pada *minimum support* 70% dan 80% terdapat perbedaan jumlah aktivitas yang dihasilkan. Untuk itu dari hasil percobaan tersebut diambil *minimum confidence* 70% untuk data mahasiswa.



**Gambar 3 Hasil Perhitungan Sistem SPADE pada Sample Data**

- b. Perhitungan dengan *minimum support* dan *minimum confidence*

Pengujian terhadap beberapa *Minimum support* dan *minimum confidence* dilakukan untuk melihat pengaruh nilai *Minimum support* dan *minimum confidence* terhadap *lift ratio* yang dihasilkan dan untuk mendapatkan *Minimum support* dan *minimum confidence* yang cukup. Hal ini dilakukan untuk melihat *lift ratio* mana yang kuat atau berpengaruh sehingga terlihat aktivitas mana yang sering dilakukan user. Pengujian berdasarkan *Minimum support* dan *minimum confidence* dilakukan untuk 3 kategori user

**Tabel 2 Data Mahasiswa dengan Minimum Support 5% dan Minimum Confidence 70%**

No.	Aktivitas	Support	Conf	B-conf	Lift ratio
1	32	5,39	100	5,39	18,56
2	92	7,11	100	7,11	14,06
3	51	7,82	100	7,82	12,78
4	45	21,57	100	21,57	4,64
5	13	29,13	100	29,13	3,43
6	45,13	16,86	78,15	29,13	2,68

Selanjutnya untuk data dosen pada tabel 3 akan digunakan *minimum support* 5%, 10%, dan 20% dan dengan nilai *minimum confidence* 40%, 50% dan 60%. Dari hasil uji coba data dosen, perubahan yang terjadi tidak terlalu besar. Pada percobaan dengan *minimum support* 5%, terdapat 9 jenis aktivitas. Pada *minimum support* 10% terdapat 5 jenis aktivitas. Dan pada *minimum support* 20%

terdapat 1 jenis aktivitas. Dari ketiga percobaan tersebut terdapat *minimum support* 5% menghasikan jumlah aktivitas yang lebih banyak sehingga terlihat aktivitas yang dilakukan oleh *user*. Untuk itu dari hasil percobaan tersebut diambil *minimum support* 5% untuk data dosen. Dan pada percobaan dengan *minimum confidence* 40%, terdapat 8 jenis aktivitas. Pada *minimum confidence* 50% terdapat 7 jenis aktivitas. Dan pada *minimum confidence* 60% terdapat 7 jenis aktivitas. Pada *Minimum support* 50% dan 60% memiliki jumlah aktivitas yang sama. Dan pada *minimum support* 40% terdapat 7 aktivitas yang dimiliki *Minimum support* 50% dan 60%. Untuk itu dari hasil percobaan tersebut diambil *minimum confidence* 40% untuk data dosen.

**Tabel 3 Data Dosen dengan Minimum Support 5% dan Minimum Confidence 40%**

No.	Aktivitas	Support	Conf	B-conf	Lift ratio
1	63	6,98	100	6,98	14,32
2	60	8,3	100	8,3	12,05
3	25	15,21	100	15,21	6,57
4	70	17,82	100	17,82	5,61
5	9	18,82	100	18,82	5,31
6	25,9	13,85	91,04	18,82	4,84
7	59	22,12	100	22,12	4,52
8	9,7	7,85	41,71	17,82	2,34

Selanjutnya untuk data pegawai pada tabel 4 akan digunakan *minimum support* 5%, 10%, dan 20% dan dengan nilai *minimum confidence* 70%, 80% dan 90%. Dari hasil uji coba data pegawai, perubahan yang terjadi tidak terlalu besar. Pada percobaan dengan *minimum support* 5%, terdapat 6 jenis aktivitas. Pada *minimum support* 10% terdapat 4 jenis aktivitas. Dan pada *minimum support* 20% terdapat 3 jenis aktivitas. Dari ketiga percobaan tersebut terdapat 3 aktivitas yang selalu ada. Namun terdapat 3 jenis aktivitas yang berbeda pada *minimum support* 5% dan 2 perbedaan pada *minimum support* 10%. Untuk itu dari hasil percobaan tersebut diambil *minimum support* 5% untuk data pegawai. Dan pada percobaan dengan *minimum confidence* 70%, terdapat 6 jenis aktivitas. Pada *minimum confidence* 80% terdapat 6 jenis aktivitas. Dan pada *minimum confidence* 90% terdapat 5 jenis aktivitas. Pada *minimum support* 80%

dan 90% terdapat perbedaan jumlah aktivitas yang dihasilkan. Untuk itu dari hasil percobaan tersebut diambil *minimum confidence* 80% untuk data pegawai.

**Tabel 4 Data Pegawai dengan Minimum Support 5% dan Minimum Confidence 80%**

No.	Aktivitas	Support	Conf	B-conf	Lift ratio
1	8	6,52	100	6,52	15,34
2	70	6,81	100	6,81	14,68
3	8,14	20,99	322,02	30,05	10,71
4	48	20,68	100	20,68	4,83
5	14	30,05	100	30,05	3,33
6	48,14	18,41	89,01	30,05	2,96

c. Pengujian dengan perbandingan kalender akademik

Dari hasil aktivitas tiap bulan dan kalender akademik didapatkan beberapa persamaan yaitu:

- Pada data mahasiswa bulan Januari hasil dari aktivitas sistem adalah aktivitas 80: Registrasi-Registrasi-Registrasi Mata Kuliah dan aktivitas 71: Registrasi-Cetak KSM sama dengan yang ada pada agenda kalender akademik tanggal 5-16 Januari 2015.
- Pada data mahasiswa bulan Maret hasil dari aktivitas sistem adalah aktivitas 85: Registrasi-Jadwal-Jadwal Ujian Mahasiswa sama dengan yang ada pada agenda kalender akademik 9-18 Maret 2015

Pada data mahasiswa bulan Mei hasil dari aktivitas sistem adalah aktivitas 32: Geladi-Pilih Lokasi dan aktivitas 92: Registrasi-Jadwal-Jadwal Ujian Mahasiswa. Aktivitas 32: Geladi--Pilih Lokasi berhubungan dengan agenda pada awal bulan juni dan aktivitas 92: Registrasi-Jadwal-Jadwal Ujian Mahasiswa sama dengan yang ada pada agenda kalender akademik 11-22 Mei 2015

#### IV. KESIMPULAN DAN SARAN

Berdasarkan dari hasil pengujian yang dilakukan. Hasil dari aktivitas yang dilakukan oleh sistem pada tiap bulannya berbeda dengan kalender akademik. Maka dari itu rekomendasi yang dapat dilakukan adalah dengan menambahkan menu input kalender akademik pada igracias. Menu ini nantinya berguna untuk membuat igracias menampilkan menu-menu yang sesuai berdasarkan inputan dari aktivitas yang ada pada kalender akademik. Berikut tampilan yang dapat diberikan berdasarkan hasil dari aktivitas yang ada pada sistem dari tiap user.

Tabel 3-5 Rekomendasi Menu Tiap User

Mahasiswa	Dosen	Pegawai
Cetak KSM	Berita Acara Perkuliahan	Data Pegawai
Jadwal Ujian Mahasiswa	Cetak Daftar Hadir	Detail Penggajian
Kehadiran	Cetak Presensi	Lihat Kehadiran
Lihat Nilai	Data Perwalian	Pendaftaran Rekreasi
Lihat Nilai Semester	Submit BAP	Pengajuan Cuti
Pilih Lokasi		Persetujuan Beban Kerja Pegawai
Registrasi Mata Kuliah		Ubah Password

Dari semua pengujian tersebut dihasilkan bahwa *minimum support* yang sesuai untuk data mahasiswa adalah 5% dan *minimum confidence* nya adalah 70%. *Minimum support* untuk data dosen adalah 5% dan *minimum confidence* nya adalah 40%. Sedangkan untuk data pegawai menggunakan *minimum support* 5% dan *minimum confidence* 80%. Hal ini dapat berbeda karena jumlah data dari tiap user berbeda dan pengaruh dari besarnya nilai support dan confidence pada setiap user.

#### REFERENSI

Gunawan, R. J. (2015). SEQUENTIAL PATTERN MINING DENGAN SPADE UNTUK PREDIKSI PEMBELIAN SPARE PART DAN AKSESORIS KOMPUTER PADA KEDATANGAN KEMBALI KONSUMEN, 314–325.

Hermawan, A. A. (2014). Analisis Konteks Proses Bisnis Berdasarkan “ Event Log ” Business Process Context

Analysis Based on “ Event Log ,” 4(3), 133–150.

Mehak, Kumar, M., & Aggarwal, N. (2013). Web usage mining: An analysis. *Journal of Emerging Technologies in Web Intelligence*, 5(3), 240–246. <https://doi.org/10.4304/jetwi.5.3.240-246>

Mooney, C. H., & Roddick, J. F. (2013). Sequential Pattern Mining – Approaches and Algorithms, 45(2). <https://doi.org/10.1145/2431211.2431218>.

Valera, M., & Chauhan, U. (2013). An efficient web recommender system based on approach of mining frequent sequential pattern from customized web log preprocessing. *2013 4th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2013*. <https://doi.org/10.1109/ICCCNT.2013.6726493>

Varnagar, C. R., Madhak, N. N., Kodinariya, T. M., & Rathod, J. N. (2013). Web usage mining: A review on process, methods and techniques. *Proceeding of the Information Communication and Embedded Systems (ICICES), 2013 International Conference*, 40–46. <https://doi.org/10.1109/ICICES.2013.6508399>

Vijayarani, S., & Suganya, M. E. (2015). Research issues in web mining. *International Journal of Computer-Aided Technologies (IJCAx)*, 2(3), 55–64. <https://doi.org/10.5121/ijcax.2015.2305>

Yunianto, D. R., Dewi, C., Yudistira, N., Studi, P., Informatika, T., Brawijaya, U., & Pola, P. (n.d.). SEQUENTIAL PATTERN MINING PADA PENCARIAN POLA PERILAKU PENGGUNA INTERNET MENGGUNAKAN ALGORITMA SPADE classes ) diterapkan untuk mencari pola perilaku pengguna internet dengan cara melakukan preprocessing data untuk menyaring informasi yang dibutuhkan . Di

Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, 42(1/2), 31–60. <https://doi.org/10.1023/A:1007652502315>