

Penggunaan Metode Algoritma FIN Untuk *Mining Frequent Access Pattern* Pada Menu iGracias Universitas Telkom

Rizki Nastiti Ambarwati¹, Yanuar Firdaus Arie Wibowo², Ibnu Asror³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹rizkinastiti@student.telkomuniversity.ac.id, ²yanuar@telkomuniversity.ac.id, ³iasror@telkomuniversity.ac.id

Abstrak

Bagi sebuah universitas, penggunaan sistem informasi *web* sangat diperlukan. Seperti halnya iGracias yang digunakan oleh seluruh civitas akademik Universitas Telkom untuk menyimpan berbagai informasi dan aktivitas pengguna yang ada pada lingkungan universitas tersebut. Untuk itu, perlu dilakukan perbaikan struktur untuk memelihara kualitas dari iGracias. Salah satu teknik yang dapat digunakan yaitu dengan *web usage mining*. Pada penelitian ini menggunakan metode algoritma FIN yang digunakan untuk melakukan *mining frequent access pattern* dan menggunakan data iGracias, dimana pengguna iGracias yang digunakan adalah data mahasiswa. Metode ini memiliki kelebihan yaitu dapat melakukan pencarian pola terhadap kecenderungan akses pengguna dan juga dengan adanya algoritma FIN dapat menyempurnakan pada memori yang lebih rendah serta performansi eksekusi yang lebih cepat. Dari implementasi metode ini guna mendapat pola pengaksesan pengguna iGracias yang sesuai dengan *user id* dari *user group* mahasiswa. Pola tersebut direkomendasikan untuk menampilkan *frequent itemset* berdasarkan dari riwayat aktivitas pengguna.

Kata kunci : algoritma FIN, *web usage mining*, *mining frequent access pattern*

Abstract

For a university, the use of web information systems is indispensable. Like iGracias, which is used by the entire academic community of Telkom University to store various information and user activities in the university environment. Therefore, it is necessary to improve the structure to maintain the quality of iGracias. One of the techniques that can be used is web usage mining. This study uses the FIN algorithm method which is used to mine frequent access patterns and uses iGracias data, where the iGracias users used are student data. This method has the advantage of being able to search for patterns on user access trends and also with the FIN algorithm it can improve on lower memory and faster execution performance. From the implementation of this method in order to get iGracias user access patterns that match the user id of the student user group. The pattern is recommended to display frequent itemset based on the user activity history.

Keywords: FIN algorithm, *web usage mining*, *mining frequent access pattern*

1. Pendahuluan

Latar Belakang

Pada saat ini, pertumbuhan internet memiliki akses informasi yang jauh lebih luas dan fungsi yang lebih beragam. Banyak bidang yang menjadi efek pengembangan internet, terutama dalam hal pendidikan seperti penggunaan iGracias untuk menyimpan berbagai informasi dan aktivitas pengguna yang ada pada lingkungan universitas tersebut. Penggunaan sistem informasi web pada universitas merupakan salah satu contoh penerapan internet pada bidang pendidikan. Sistem informasi berbasis web merupakan sarana digital yang dapat diakses seluruh civitas akademik Universitas Telkom. Untuk itu mengetahui pola pengaksesan menjadi hal penting. Agar dapat mengurutkan menu pengaksesan iGracias sesuai dengan *user id* dari *user group* data mahasiswa. Dan juga untuk mengetahui jumlah *frequent itemset* untuk setiap bulannya. Untuk menggali *knowledge* yang ada dari aktivitas pengguna dalam mengakses sistem informasi, oleh karena itu terdapat teknik *web mining* [1].

Web mining terbagi menjadi tiga yaitu, *web content mining*, *web structure mining*, dan *web usage mining*. Pada *web usage mining* digunakan untuk menganalisis pola perilaku pengguna pada sistem informasi web yang berfungsi untuk pengembangan sistem informasi. *Web usage mining* bertujuan untuk menemukan informasi atau pengetahuan yang bermanfaat dari pola navigasi pengguna pada *web* [2]. Untuk mendukung aktivitas pengguna, perlu dilakukan pengembangan dengan menggunakan *event log*. Dikarenakan data *event log* ini merupakan data yang berkelanjutan maka untuk memecahkan masalah ini digunakan sebuah metode algoritma FIN untuk *mining frequent access pattern* pada suatu sistem informasi dan dengan studi kasus iGracias Universitas Telkom.

Algoritma FIN merupakan sebuah cara pencarian pola dalam bentuk pohon yang disebut dengan *POC-tree*, dimana setiap transaksi akses pengguna akan dimasukkan ke dalam *nodesets*. Metode ini digunakan karena algoritma FIN merupakan algoritma pengembangan dari algoritma PPV dan *Prepost*, dengan penyempurnaan pada memori yang lebih rendah serta performa eksekusi komputasi yang cepat [3]. Metode ini menggunakan *minimum support* sebagai batas frekuensi minimum yang harus ditentukan terlebih dahulu oleh penggunaannya. *Minimum support* digunakan untuk mengeliminasi item-item yang tidak *infrequent* pada setiap data transaksi [4].

Topik dan Batasannya

Topik yang dibahas pada tugas akhir ini adalah *mining frequent access pattern* pada iGracias Universitas Telkom. Metode yang digunakan adalah dengan algoritma FIN. Penelitian ini terdapat beberapa batasan yakni data yang digunakan merupakan data dari *event log* pada <http://igracias.telkomuniversity.ac.id/> dari bulan April - Juni 2020. Data yang digunakan terdiri dari kategori pengguna yaitu mahasiswa.

Tujuan

Berdasarkan identifikasi masalah yang telah dirumuskan maka tujuan dari penelitian ini adalah:

1. Menerapkan metode algoritma FIN untuk *mining frequent access pattern* pada *web usage mining* untuk membuat pemodelan pola pengaksesan pengguna pada iGracias sesuai dengan *user id*,
2. Melakukan analisis dari metode algoritma FIN untuk *mining frequent access pattern* pada iGracias

Organisasi Tulisan

Bab 2 pada penelitian ini memberikan tinjauan mengenai studi terkait dengan *mining frequent access pattern* untuk menemukan pola pengaksesan pada iGracias menggunakan algoritma FIN. Bab 3 menjelaskan mengenai sistem yang di bangun dalam penelitian ini. Bab 4 menjelaskan mengenai hasil analisis dan evaluasi pada penelitian ini dan bab 5 memberikan kesimpulan dalam penelitian ini.

2. Studi Terkait

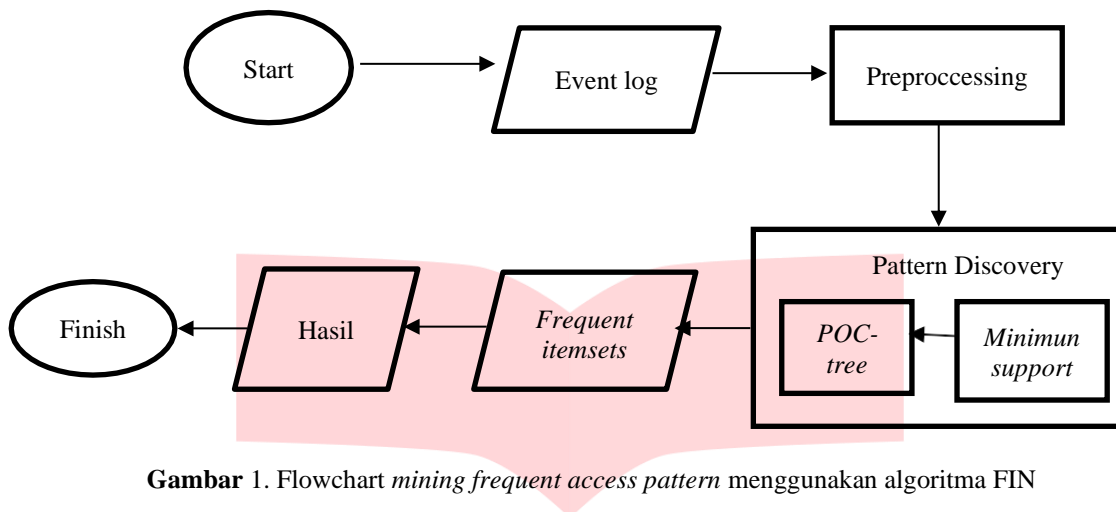
Penelitian sebelumnya mengenai pencarian *frequent pattern mining* menggunakan algoritma *PrePost*, *FIN*, *H-Mine*. Penelitian ini dilakukan oleh M.Ranjanisindu [5] memberikan gambaran tentang beberapa algoritma yang biasa digunakan dalam *frequent pattern mining*. Karena pada dasarnya setiap algoritma tersebut memiliki kelebihan dan kekurangan masing-masing, tergantung pada kasus penggunaannya. Pada penelitian ini juga menggunakan berbagai macam *tree* yang digunakan dalam setiap algoritma tersebut. Setiap algoritma memiliki *tree* untuk hasil penyelesaian yang terbaik dan terburuk, tergantung pada ukuran dataset dan tipe data yang ada. Penelitian oleh Chen Lin [6] menjelaskan bahwa mengusulkan PFIN untuk mewakili *itemset*. Algoritma ini menggunakan strategi yang disebut proyeksi untuk membagi *set-enumeration tree* sehingga setiap partisi dapat dieksekusi untuk menemukan item yang sering secara paralel. PFIN juga menggunakan strategi pemangkasan yang disebut *promotion* untuk mengurangi ruang pencarian. Dan hasil yang didapat dari penelitian ini adalah bahwa PFIN dapat digunakan untuk mengatasi permasalahan FIM dengan skala yang besar.

Penelitian sebelumnya dengan judul "*FIN Algorithm for Generating Frequent Itemset in Big Data*" dilakukan oleh R. Prakash [7]. Pada penelitian tersebut algoritma FIN digunakan untuk menganalisis kebiasaan konsumen dalam membeli suatu produk. *mining frequent itemset* diproses secara paralel menggunakan *Map-Reduce* yang diimplementasikan dengan menggunakan algoritma FIN. Hasil yang didapatkan pada penelitian tersebut adalah performansi yang dihasilkan oleh algoritma FIN lebih cepat dari pada algoritma Apriori maupun FP-Growth. Penelitian yang lainnya berjudul "*Sentiment Analysis using FP-Growth and FIN Algorithm*" [8]. Pada penelitian tersebut dilakukan oleh Prajacta Lobo menggunakan algoritma FIN untuk menganalisis sentimen konsumen berdasarkan review konsumen pada suatu produk. Mengidentifikasi kata yang sering digunakan untuk semua review yang dikirim konsumen setelah membeli suatu produk melalui *mining frequent itemset*. Algoritma FIN menghasilkan polaritas positif dengan konsumsi maksimal memori 37,94 MB.

Algoritma FIN merupakan salah satu algoritma yang diusulkan Oleh Deng Et Al pada tahun 2014 [3]. *Frequent pattern mining* pertama kali dilakukan oleh Gawali [8] untuk menganalisis ulasan pengguna. Pada penelitian yang dilakukan oleh Deng Et Al membandingkan kinerja *PPC-tree* dengan *POC-tree*. Hasil penelitian ini menunjukkan bahwa menggunakan *nodesets* untuk *frequent pattern mining*. Karena *nodesets* hanya membutuhkan *pre-order* dari setiap node, yang membuatnya menghemat setengah memori dibandingkan dengan *N-list* dan *Node-list*. Berdasarkan *nodesets* untuk digunakan dalam algoritma yang efisien yang disebut algoritma FIN untuk *frequent pattern mining*. Setelah di bandingkan dengan yang lainnya hasil percobaan menunjukkan bahwa algoritma FIN lebih efisien dalam hal konsumsi memori.

3. Sistem yang Dibangun

Bagian ini membahas bagaimana sistem yang akan dibangun. Dimana sistem tersebut yang menerima *input* berupa *event log* dari iGracias. Data yang digunakan terdiri dari kategori pengguna yaitu mahasiswa yang didapat dari iGracias Universitas Telkom dari bulan April 2020 sampai Juni 2020. Berikut adalah gambaran sistem yang akan dibangun untuk melakukan *mining frequent access pattern* pada iGracias menggunakan algoritma FIN.



Gambar 1. Flowchart *mining frequent access pattern* menggunakan algoritma FIN

3.1 Dataset

Dataset yang digunakan merupakan *event log* yang akan diteliti. Pada tugas akhir ini *event log* yang digunakan berasal dari iGracias Universitas Telkom. *Event log* tersebut memiliki 3500 data yang didapatkan dari April hingga Juni 2020. Namun dataset tersebut tidak dipergunakan semuanya melainkan hanya mengambil 1 jenis pengguna, yaitu mahasiswa. Dari jenis pengguna tersebut terdapat perbedaan jumlah data untuk setiap bulannya.

3.2 Event Log

Event log merupakan catatan histori yang memuat rangkaian aktivitas yang dilakukan pengguna dalam sebuah sistem informasi. *Event log* pada sistem informasi memuat catatan proses yang dilakukan oleh setiap pengguna [9]. Setiap *event log* memiliki berbagai atribut. Tetapi tidak semua atribut selalu ada pada suatu *event log*. Tergantung dari data yang diperoleh dari suatu sistem informasi.

3.3 Preprocessing

Beberapa *database* memiliki data yang tidak konsisten. *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* [10]:

1. *Data cleaning*: proses mengidentifikasi, memilih dan menghapus data dari hal-hal yang tidak diperlukan atau tidak relevan.
2. *User and Session Identification*: pada identifikasi pengguna akan digunakan informasi ID login sebagai identitas unik dari tiap pengguna. Sedangkan identifikasi *session*, dilakukan untuk mengetahui sesi tunggal yang dilakukan oleh pengguna. Umumnya sesi akan dibagi berdasarkan waktu.

3.4 Pattern Discovery

Setelah melakukan *preprocessing* tahap selanjutnya adalah *pattern discovery*. Pada *pattern discovery* akan dilakukan proses pencarian pola pengaksesan halaman yang paling sering diakses. Dalam pencarian pola tersebut menggunakan disiplin ilmu *mining frequent access pattern*. Metode algoritma FIN merupakan algoritma *frequent itemset* yang menggunakan struktur data *nodeset* dan juga salah satu metode yang akan digunakan untuk menentukan seberapa besar frekuensi sebuah *pattern* pada saat mengakses sebuah sistem informasi web.

Karena Nodesets didasarkan oleh *POC-tree* (*pre-order coding tree*). Pre-order merekam ketika node N diakses selama traversal pre-order. Konstruksi *POC-tree* dipergunakan untuk mengidentifikasi *frequent-1 itemset*, guna mencari *frequent-1 itemset* harus mendaftarkan semua *itemsets* yang terdapat dalam *database* transaksi. Mencari *Frequent-2 itemset* yang berdasar pada *frequent-1 itemset*, sehingga tidak perlu mencari dari *sequence database* lagi. Hingga mencari *frequent-k itemset*, untuk mencari *frequent-k itemset* harus memiliki *prefix* yang sama. Kemudian, setelah membangun *nodesets* ini, *POC-tree* tidak berguna dan dapat dihapus. *Post-order* juga dapat digunakan untuk merepresentasikan setiap node dan membangun *tree* yang sama untuk menghasilkan *nodeset*. Artinya, *post-order* setara dengan *pre-order* untuk metode ini, namun penelitian ini menggunakan *pre-order tree* [3].

Pada proses algoritma FIN terdapat beberapa langkah pengerjaan. Pertama set *minimum support*, dimana pengguna diminta untuk menginputkan *minimum support*, pada setiap item dalam rentang $0\% < \text{minimum support} < 100\%$. *minimum support* digunakan untuk *eliminator* item yang tidak *infrequent* pada setiap data transaksi [4]. Setelah menentukan *minimum support*, kemudian mengurutkan item berdasarkan dari nilai *support* terbesar sampai terkecil. Setelah urut selanjutnya mengeliminasi item-item pada setiap transaksi yang memiliki nilai kurang dari *minimum support*. item yang nilai *support*-nya telah terurut, maka dimasukkan ke *POC-tree* untuk menyusun *frequent itemset*. Dari *frequent itemset* dapat diketahui jumlah nilai *support* dalam satu transaksi tersebut. Jumlah nilai *support* dari setiap *minimum support* yang telah ditentukan sebelumnya akan berbeda sesuai dengan jumlah data transaksi dalam database.

3.5 Pattern Analysis

Pada proses *pattern analysis*, hasil *frequent sequential access pattern* yang ada kemudian di analisa berdasarkan *minimum support* yang digunakan. Dari proses ini akan diketahui *behavior* pola akses untuk tiap *usergroup* dan juga menampilkan jumlah *support* dari setiap pola tiap *usergroup*.

4. Evaluasi

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

4.1 Data yang digunakan

Dataset yang digunakan pada penelitian ini menggunakan data *event log* dari iGracias Universitas Telkom. Dari *group user* pengguna yaitu mahasiswa yang terdiri dari 3.500. Contoh data yang digunakan pada penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Contoh data yang digunakan

No	Contoh data
1	313853 Registrasi JadwalMahasiswaJadwalUjianMahasiswaStatusRegistrasiMahasiswa Nilai LihatNilaiLihatEkivalensiLihatNilaiSemester Survey Beranda Nilai LihatNilai Survey Kuesioner
2	306994 Presensi Kehadiran Survey BerandaKuesioner Presensi Kehadiran Survey KuesionerKuesionerBerandaBerandaKuesioner Presensi Kehadiran Survey KuesionerKuesionerBeranda
3	228574 Nilai LihatNilaiLihatNilaiSemesterStatistikNilaiLihatNilai Survey Beranda Nilai LihatNilai Survey Beranda Nilai LihatNilai SemesterLihatEkivalensi Survey KuesionerKuesionerKuesioner Presensi Kehadiran

4.2 Skenario Uji

Skenario uji bertujuan untuk menguji model sistem yang dibangun dengan mengubah nilai *minimum support*. Perubahan nilai *minimum support* mengakibatkan perbedaan jumlah pola pengaksesan yang terbentuk. Semakin kecil nilai *minimum support* maka pola yang terbentuk semakin banyak. Sebaliknya apabila semakin besar nilai *minimum support* maka pola yang terbentuk semakin sedikit.

4.3 Hasil Pengujian

4.3.1 Hasil Pengujian pada data mahasiswa

Pengujian dilakukan terhadap beberapa *minimum support* yang dilakukan di kategori pengguna yaitu mahasiswa. Pola yang dihasilkan berangsur-angsur berkurang pada nilai *minimum support* yang lebih besar. Pada 3500 data yang digunakan sebagai data untuk melakukan *mining frequent itemset* menunjukkan bahwa data memiliki sifat *devergen*, memiliki sedikit pengulangan item yang sama. Apabila diuji dengan nilai *minimum support* 0.005 pada bulan april akan menghasilkan pola pengaksesan pengguna dengan hasil konsumsi memori yang lebih rendah serta performansi eksekusi komputasi yang cepat. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.005 dapat dilihat ditabel 2.

Tabel 2. Pola yang terbentuk sesuai *user id* bulan april

No	User id	Pola	support	confidence
1	227828	Presensi KehadiranRFIDLogKehadiranRFIDLog	0.006	0.007
2	226394	Registrasi JadwalUjianMahasiswaCetakJadwalUjianMahasiswa Presensi KehadiranRFIDLog	0.007	0.03
3	282750	Registrasi JadwalUjianMahasiswaCetakJadwalUjianMahasiswa	0.006	0.003
4	145334	Registrasi StatusRegistrasiMahasiswaJadwalMahasiswaJadwalUjian MahasiswaCetakJadwalUjianMahasiswa Presensi Kehadiran	0.008	0.2
5	179432	Registrasi JadwalUjianMahasiswa Presensi KehadiranRFIDLog	0.006	0.03

Pada Tabel 2 menampilkan hasil $k(\geq 2)$ *frequent itemset* untuk mendapatkan pola pengaksesan pengguna. Pemilihan hasil $k(\geq 2)$ menyesuaikan dengan teorema algoritma FIN. Hasil pembentukan pola pengaksesan ditentukan oleh *minimum support* sebagai *eliminator* item-item yang tidak *infrequent* pada setiap data transaksi. Digunakan nilai *minimum support* 0.015, sehingga menghasilkan pola pengaksesan yang representatif dengan kebiasaan mahasiswa. Urutan menu yang sering diakses adalah Registrasi, Jadwal Ujian Mahasiswa, dan Cetak Jadwal Ujian Mahasiswa dikarenakan pada bulan april mahasiswa melakukan ujian. Apabila digunakan nilai *minimum support* 0.50 akan menampilkan hasil yang akan mengeliminasi banyak item *infrequent* dalam data transaksi tetapi belum dapat menampilkan *user id* dikarenakan jumlah *minimum support* belum memenuhi nilai *minimum support*. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.50 dapat dilihat ditabel 3.

Tabel 3. Pola yang terbentuk bulan april

No	Pola	support	confidence
1	Presensi Kehadiran	0.72	0.92
2	Kehadiran Registrasi	0.78	0.88
3	Presensi	0.76	0.98
4	Presensi Registrasi	1.44	1.84
5	Registrasi	1.18	0.80

Apabila diuji dengan nilai *minimum support* 0.005 pada bulan mei akan menghasilkan pola pengaksesan pengguna dengan hasil konsumsi memori yang lebih rendah serta performansi eksekusi komputasi yang cepat. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.50 dapat dilihat ditabel 4.

Tabel 4. Pola yang terbentuk sesuai *user id* bulan mei

No	User id	Pola	support	confidence
1	272689	Presensi KehadiranRFIDLog	0.005	0.007
2	311454	Presensi KehadiranRFIDLog Survey Kuesioner	0.006	0.07
3	316645	Survey BerandaKuesioner Registrasi StatusRegistrasi	0.006	0.01
4	275253	Survey BerandaKuesionerKuesionerBeranda	0.007	0.18
5	227178	Presensi KehadiranRFIDLog Survey BerandaKuesioner	0.005	0.07

Pada Tabel 4 menampilkan hasil $k(\geq 2)$ *frequent itemset* untuk mendapatkan pola pengaksesan pengguna. Pemilihan hasil $k(\geq 2)$ menyesuaikan dengan teorema algoritma FIN. Hasil pembentukan pola pengaksesan ditentukan oleh *minimum support* sebagai *eliminator* item-item yang tidak *infrequent* pada setiap data transaksi.

Apabila dipilih nilai *minimum support* 0.015 akan mengeliminasi banyak *item infrequent* dalam data transaksi. Sehingga menghasilkan pola pengaksesan yang representatif dengan kebiasaan mahasiswa. Urutan menu yang sering diakses adalah Survey, Kuesioner, dan Beranda dikarenakan setelah melakukan ujian di bulan mei mahasiswa harus mengisi survei yang telah disediakan. Apabila digunakan nilai *minimum support* 0.50 akan menampilkan hasil yang akan mengeliminasi banyak *item infrequent* dalam data transaksi tetapi belum dapat menampilkan *user id* dikarenakan jumlah *minimum support* belum memenuhi nilai *minimum support*. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.50 dapat dilihat ditabel 5.

Tabel 5. Pola yang terbentuk bulan mei

No	Pola	<i>support</i>	<i>confidence</i>
1	Kehadiran Survey	0.62	0.85
2	Presensi Kehadiran Survey	0.60	1.24
3	Presensi Kehadiran	1.23	2.55
4	Survey Presensi	0.90	1.23
5	Presensi	0.72	0.98

Apabila diuji dengan nilai *minimum support* 0.005 pada bulan juni akan menghasilkan pola pengaksesan pengguna dengan hasil konsumsi memori yang lebih rendah serta performansi eksekusi komputasi yang cepat. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.005 dapat dilihat ditabel 6.

Tabel 6. Pola yang terbentuk sesuai *user id* bulan juni

No	<i>User id</i>	Pola	<i>support</i>	<i>confidence</i>
1	279403	Presensi KehadiranRFIDLog	0.005	0.04
2	311454	Nilai LihatNilaiLihatNilaiLihatNilaiSemester Survey BerandaKuesioner	0.005	1.16
3	330242	Nilai StatistikNilaiLihatNilaiLihatNilaiSemester	0.006	0.35
4	275253	Nilai LihatNilaiLihatNilaiSemester TAKMahasiswa DaftarTAKMahasiswaTAKProses	0.008	0.29
5	227178	Presensi KehadiranRFIDLog Nilai LihatNilai	0.008	0.12

Pada Tabel 6 menampilkan hasil $k(\geq 2)$ *frequent itemset* untuk mendapatkan pola pengaksesan pengguna. Pemilihan hasil $k(\geq 2)$ menyesuaikan dengan teorema algoritma FIN. Hasil pembentukan pola pengaksesan ditentukan oleh *minimum support* sebagai *eliminator* item-item yang tidak *infrequent* pada setiap data transaksi. Apabila dipilih nilai *minimum support* 0.005 akan mengeliminasi banyak *item infrequent* dalam data transaksi. Sehingga menghasilkan pola pengaksesan yang representatif dengan kebiasaan mahasiswa. Urutan menu yang sering diakses adalah Nilai Lihat Nilai Semester dan Lihat Nilai dikarekan di bulan juni apabila sudah melakukan ujian mahasiswa lebih sering unruk melihat nilai. Apabila digunakan nilai *minimum support* 0.50 akan menampilkan hasil yang akan mengeliminasi banyak *item infrequent* dalam data transaksi tetapi belum dapat menampilkan *user id* dikarenakan jumlah *minimum support* belum memenuhi nilai *minimum support*. Bagian pola yang dihasilkan untuk nilai *minimum support* 0.50 dapat dilihat ditabel 7.

Tabel 7. Pola yang terbentuk bulan juni

No	Pola	<i>support</i>	<i>confidence</i>
1	Nilai Beranda	1.11	0.28
2	Nilai LihatNilai	1.21	0.31
3	Registrasi Nilai	0.82	0.86
4	Nilai Presensi	1.52	0.39
5	Nilai	1.07	0.27

4.4 Analisis Hasil Pengujian

Hasil yang ditunjukkan dalam Tabel 8. menunjukkan bahwa hasil perhitungan dengan menggunakan algoritma FIN. Hasil tersebut sangat terkait dengan jumlah data yang akan diproses. Kinerja algoritma FIN akan menghasilkan memori yang lebih rendah serta performa eksekusi yang cepat. Berikut merupakan kinerja waktu komputasi untuk menampilkan pola pengaksesan pengguna iGracias berdasarkan *user id*.

Tabel 8. Hasil kinerja algoritma FIN

April	Mei	Juni
Minsup = 6 Number of transactions: 1055 Number of frequent itemsets: 7140 Total time: 154 ms Max memory:48.083 MB	Minsup = 6 Number of transactions: 1155 Number of frequent itemsets: 4960 Total time: 162 ms Max memory:48.655 MB	Minsup = 6 Number of transactions: 1195 Number of frequent itemsets: 10754 Total time: 224 ms Max memory:54.145MB

Secara umum pengujian yang dilakukan pada penelitian ini yaitu untuk mengetahui pola pengaksesan pengguna pada iGracias sesuai dengan *user id*. Tidak semua menu pada data dapat digunakan, karena dari hasil pemilihan dan *minimum support* yang ditentukan. Contoh data yang tereliminasi adalah jumlah *frequent itemset* nya tidak lebih sama dengan *minimum support* yang telah ditentukan. Dari hasil pengujian tersebut setiap *itemset* dinotasi dengan *support*.

Nilai *minimum support* berbanding terbalik dengan jumlah pola pengaksesan pengguna iGracias. Pemilihan *minimum support* dengan nilai tinggi akan membuat mengeliminasi banyak item *infrequent* pada data transaksi. Sehingga dapat menghasilkan pola yang representatif dengan kebiasaan pengguna iGracias. Hasil *mining* dengan menampilkan urutan pola pengaksesan terdapat perbedaan kinerja waktu komputasi, apabila data yang diproses semakin banyak waktu yang dibutuhkan juga semakin lambat dan memori yang digunakan semakin besar. Tetapi perbedaan tersebut tidak terlalu berbeda, dikarenakan dengan menggunakan algoritma FIN dapat memperkecil pada konsumsi memori serta performa eksekusi yang lebih cepat .

5. Kesimpulan

Berdasarkan dari hasil pengujian yang dilakukan. Algoritma FIN diimplementasikan untuk menentukan pola pengaksesan pada iGracias. Perhitungan nilai *support* digunakan untuk mengevaluasi hasil variasi pola yang dihasilkan oleh sistem. Nilai *minimum support* sebagai *eliminator* terhadap item menu yang tidak *infrequent*. Nilai *minimum support* menentukan bentuk *POC-tree* dalam menghasilkan pola pengaksesan iGracias. Apabila menggunakan *minimum support* yang kecil akan menghasilkan pola pengaksesan yang banyak. Sedangkan menggunakan *minimum support* yang besar menghasilkan pola pengaksesan yang sedikit. Tetapi dengan menggunakan *minimum support* yang besar tidak dapat menampilkan *user id* yang telah diinputkan karena jumlah *frequent itemset* nya tidak lebih sama dengan *minimum support* yang telah ditentukan.

Pada Tabel 2, 4, dan 6 terlihat pola pengaksesan iGracias yang berbeda untuk *group user* yakni mahasiswa. Pada hasil pengujian tersebut, menggunakan *minimum support* 0.005. Pada Tabel 3, 5, dan 7 menggunakan *minimum support* 0.50. *Group user* tersebut menunjukkan hasil kinerja waktu komputasi yang berbeda-beda. Tergantung dengan *input* data yang akan diproses. Semakin banyak data yang akan diproses maka waktu dan memori yang digunakan akan semakin tinggi. Tetapi perbedaan yang dihasilkan tidak terlalu berbeda dikarenakan dengan menggunakan algoritma FIN dapat menghemat konsumsi memori serta komputasi yang lebih rendah

Dengan metode algoritma FIN dapat mendapatkan pola pengaksesan *group user sesuai dengan user id*. Hal ini dapat terlihat dari perbedaan masing-masing *group user* tiap bulannya dan sesuai dengan nilai *minimum support*. Untuk *group user* akan mendapatkan pola yang berbeda setiap bulannya. Maka dari itu pola pengaksesan tersebut digunakan untuk rekomendasi menu yang sesuai dengan *user id* data mahasiswa. Ini nantinya berguna untuk menampilkan urutan menu yang sesuai *user id* berdasarkan pengaksesan dari aktivitas pengguna yang ada dan juga mengetahui *frequent itemset* untuk setiap bulannya yang dimana untuk setiap bulan tersebut jumlah *frequent itemset* tersebut berbeda-beda dikarenakan pada bulan-bulan tersebut sudah terdapat perbedaan pola kebiasaan yang diakses oleh mahasiswa. Semua tergantung dari data tiap pengguna yang berbeda dan pengaruh besarnya nilai *minimum support* pada setiap pengguna.

Daftar Pustaka

- [1] T. Srivastava, P. Desikan, and V. Kumar, Web Mining – Concepts, Applications and Research Directions, vol. 307, no. 2005, pp. 275–307, 2005.
- [2] C. Gomathi, M. Moorthi, and K. Duraiswamy, Web Access Pattern Algorithms in Education Domain, *Comput. Inf. Sci.*, vol. 1, no. 4, pp. 183–186, 2008.
- [3] Z. H. Deng and S. L. Lv, Fast mining frequent itemsets using Nodsets, *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4505–4512, 2014.
- [4] R. Nova Saputra, M. Tanzil Furqon, and Indriati, “Implementasi Association Rule Mining Untuk Menentukan Menu Paket Makanan Dengan Algoritma FIN Menggunakan Nodsets,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3962–3967, 2018.
- [5] Ranjanisindu, S. Gunasekaran, and N.Deepa, Frequent Pattern Mining Algorithms PrePost, FIN, H-

- Mine – A Survey, *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 8, no. 2, 2019.
- [6] C. Lin and J. Gu, PFIN : A Parallel Frequent Itemset Mining Algorithm Using Nodesets, vol. 9, no. 6, pp. 81–92, 2016.
- [7] R. Prakash and D. Prabha, Fin Algorithm For Generating Frequent Itemset In Big Data, *J. Adv. Comput. Commun. Technol.*, vol. 4, no. 3, pp. 58–61, 2016.
- [8] G. Rajendra and P. Lobo, Association data mining in sentiment analysis, *Int. J. Comput. Trends Technol.*, vol. 39, no. 2, 2016.
- [9] A. A. Hermawan, “Analisis Konteks Proses Bisnis Berdasarkan ‘ Event Log ’ Business Process Context Analysis Based On " Event Log ”,” *J. Penelit. dan Pengemb. Komun. dan Inform.*, vol. 4, no. 3, pp. 133–150, 2014.
- [10] C. R. Varnagar, N. N. Madhak, T. M. Kodinariya, and J. N. Rathod, “An efficient web recommender system based on approach of mining frequent sequential pattern from customized web log preprocessing,” *Int. Conf. Comput. Commun. Netw. Technol. ICCCNT*, 2013.

