

ANALISIS INTERAKSI AKTIVITAS PEMBELAJARAN DARING BERDASARKAN DATA LOG AKTIVITAS PADA LEARNING MANAGEMENT SYSTEM (LMS) MENGUNAKAN EDUCATIONAL PROCESS MINING

Novriani Dewi Anwar¹, Nanang Yudi Setiawan^{*2}, Welly Purnomo³

^{1,2,3} Universitas Brawijaya, Malang

¹nov_anwar@student.ub.ac.id, ²nanang@ub.ac.id, ³wepe@ub.ac.id

^{*}Penulis Korespondensi

(Naskah SENTRIN, Naskah masuk: 06 November 2023, diterima untuk diterbitkan: 30 Oktober 2024)

Abstrak

Perpaduan perkembangan teknologi di bidang pendidikan telah mengembangkan suatu inovasi yang disebut LMS (*Learning Management System*) yang dapat diakses dari mana saja dan kapan saja tanpa harus melakukan pertemuan secara fisik di kelas. Kegiatan interaksi pada LMS memiliki *event log* yang dapat diekstraksi untuk mengidentifikasi analisis menggunakan *Educational Process Mining* (EPM). Kami mengekstrak *event log* dari ELING, platform LMS yang disediakan oleh fakultas kami. Selanjutnya, kami melakukan *Process Mining* menggunakan metode *Heuristic Miner* dari tiga dosen kelas mata kuliah *Data Warehouse* yang berbeda. Analisis yang dilakukan yaitu melihat gambaran kegiatan pembelajaran yang diberikan oleh masing-masing dosen dan pendekatannya dalam mengarahkan mahasiswa untuk berinteraksi dengan modul materi pada ELING apakah sesuai urutan dalam Rencana Pembelajaran Semester (RPS). Kami memvariasikan *dependency threshold* untuk mengidentifikasi modul pembelajaran mana yang memiliki tingkat ketergantungan yang lebih tinggi antar *event* yang menjadi inti pembelajaran mata kuliah tersebut. Pada *dependency threshold* 0,6, jumlah kegiatan dari model yang ditemukan untuk semua dosen adalah 21. Sementara pada *threshold* 0,8, jumlah kegiatan bervariasi antara 12 dan 20. Dengan variasi *dependency threshold* ini juga akan mengidentifikasi modul mana yang tetap muncul yang berhubungan dengan bentuk model aktivitas pembelajaran dan apakah model yang dihasilkan sesuai dengan urutan RPS.

Kata kunci: *Learning Management System, Process Mining, Heuristic Miner, Dependency Threshold, Educational Process Mining*

ANALYSIS OF ONLINE LEARNING ACTIVITY INTERACTIONS BASED ON ACTIVITY LOG DATA IN THE LEARNING MANAGEMENT SYSTEM (LMS) USING EDUCATIONAL PROCESS MINING

Abstract

The combination of technological developments in the field of education has developed an innovation called LMS (*Learning Management System*) that can be accessed from anywhere and anytime without having to conduct physical meetings in class. Interaction activities in the LMS have event logs that can be extracted to identify analysis using *Educational Process Mining* (EPM). We extract event logs from ELING, an LMS platform provided by our faculty. Next, we conducted *Process Mining* using the *Heuristic Miner* method from three different *Data Warehouse* course class lecturers. The analysis carried out is to see an overview of the learning activities provided by each lecturer and their approach in directing students to interact with the material modules on ELING whether in the order in the *Semester Learning Plan* (RPS). We vary the *dependency threshold* to identify which learning modules have a higher level of dependency between events that are at the core of the course's learning. At a *dependency threshold* of 0.6, the number of activities from the model found for all lecturers was 21. While at the *threshold* of 0.8, the number of activities varies between 12 and 20. With *dependency threshold* variations, it will also identify which modules still appear that relate to the shape of the learning activity model and whether the resulting model matches the RPS sequence.

Keywords: *Learning Management System, Process Mining, Heuristic Miner, Dependency Threshold, Educational Process Mining*

1. PENDAHULUAN

Learning Management System (LMS) dapat didefinisikan sebagai platform perangkat lunak berbasis web yang menyediakan lingkungan pembelajaran online interaktif dan mengotomatiskan administrasi, pengorganisasian, pengiriman, dan pelaporan konten pendidikan dan hasil pelajar (Turnbull, Chugh & Luck, 2020), yang memungkinkan pengguna (kebanyakan guru dan peserta didik) untuk melihat, berbagi, berdiskusi, mengontrol, melacak kegiatan belajar online, catatan, dll. (Aldahwan & Alsaeed, 2020). LMS digunakan dalam pembelajaran dan pengajaran kelas online, dengan guru dan peserta didik yang mendukung layanan pembelajaran (Aldahwan & Alsaeed, 2020). LMS telah berkembang sebagai tanggapan terhadap permintaan akan produk pendidikan inovatif yang memanfaatkan kemajuan teknologi informasi dan telekomunikasi (Turnbull, Chugh & Luck, 2020). Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya memiliki LMS yang telah dikembangkan menjadi media dalam kegiatan belajar mengajar secara daring yang disebut sebagai ELING.

Pada ELING, satu mata kuliah memiliki tim *teaching* yang terdiri dari beberapa dosen pengampu yang mengajar dan menyusun Rencana Pembelajaran Semester (RPS) untuk mata kuliah tersebut. Sehingga urutan pemberian materi kuliah tidak berbeda pada setiap kelas dengan mata kuliah yang sama, dengan perbedaan pada tugas, kuis, ujian dan pengumuman yang berbeda tergantung dosen pengampu kelas tersebut. ELING seperti LMS lainnya memiliki *data log* aktivitas yang menyimpan interaksi mahasiswa dalam ELING yang bisa dianalisis untuk keperluan *Educational Process Mining*. Interaksi disini ialah urutan mahasiswa dalam mengakses modul, pengulangan akses akan dilihat secara signifikan melalui status pada *data log* aktifitas. *Data log* aktivitas yang dikumpulkan berisi data yang tidak beraturan yang kemudian akan di-*preprocessing* untuk diolah menjadi *event log* agar dapat di analisis. *Event log* mewakili kumpulan peristiwa, menggambarkan kegiatan apa yang telah dilakukan untuk berbagai contoh proses yang diteliti (Berti, van Zelst & van der Aalst, 2019).

Process Mining (PM) adalah cabang ilmu pengetahuan yang muncul menyediakan seperangkat alat yang komprehensif untuk memberikan wawasan berbasis fakta dan untuk mendukung peningkatan proses (van der Aalst, 2011) dengan menggunakan *event log* dari sistem yang ada. PM yang diterapkan pada data pendidikan disebut *Educational Process Mining* (EPM) (Bogarín, Cerezo & Romero, 2017). *Educational Process Mining* menggunakan alat dan metode *Process Mining* untuk mengekstrak pengetahuan dalam kegiatan data pendidikan. Teknik ini akan digunakan dalam pengolahan *event log* aktivitas pada ELING menggunakan *event log* dari interaksi yang akan diproses untuk mendapatkan model proses yang dihasilkan dan melihat pola urutan

interaksi pembelajaran sesuai dengan Rencana Pembelajaran Semester (RPS) menggunakan teknik *Process Mining*. Penelitian ini berfokus untuk proses penelusuran menggunakan *Process Mining* dan mengidentifikasi pola interaksi penyelesaiannya pada model proses dan juga apakah sudah sesuai dengan RPS yang telah dirumuskan atau belum.

2. TINJAUAN PUSTAKA

Process Mining (PM) yang diterapkan pada data pendidikan disebut *Educational Process Mining* (EPM) (Bogarín, Cerezo & Romero, 2017). Bogarín juga menyebutkan bahwa EPM berfokus pada *process-centric* yang membuat proses tidak diketahui (atau sebagian besar tidak diketahui) menjadi jelas dan mengasumsikan berbagai tipe data *events*, seperti setiap *events* milik *single* proses yang disebut *case* dan *events* yang berhubungan dengan *activities*. Dia juga menjelaskan dalam EPM *process discovery* dari PM menjadi dapat memvisualisasikan model perilaku siswa. Terdapat beberapa referensi yang digunakan untuk penelitian dalam mengeksplorasi informasi serta bahan pembelajaran dalam melakukan penelitian ini.

Pada penelitian sebelumnya (Chanifah, Andreswari & Fauzi, 2021) menganalisis pola pembelajaran berdasarkan frekuensi dalam mengakses LMS menggunakan 3 jenis *threshold Heuristic Miner*. Hasilnya dari frekuensi urutan akses menunjukkan pola aktivitas yang sesuai dengan *Process Mining* yaitu dosen mengarahkan/memberikan mahasiswa materi terlebih dahulu, lalu diikuti tugas kemudian kuis, dan pada penelitian lain (Hachicha et al., 2021) menganalisis model proses *event log* berdasarkan *multi-features* pada profil pelajar dengan menggunakan tiga algoritma *Process Mining* yaitu yaitu Inductive Mining, Heuristic Mining dan Alpha Mining dan membagi *event log* menjadi lulus dan tidak lulus. Lalu dilakukan *clustering* pada profil fitur pelajar yang memilih pembelajaran tatap muka, pembelajaran online dan hybrid. Pada evaluasinya menghasilkan bahwa *Heuristic Mining* memiliki kecenderungan yang stabil dan mendekati keakuratan model process pada matriks Fitness, Precision dan Generalization.

2.1 Process Mining

Penggunaan *Process Mining* dalam bidang pendidikan berupa *educational process mining* yang membantu mengidentifikasi masalah yang mungkin terjadi dalam proses pembelajaran dan mencari solusi untuk meningkatkan efisiensi dan kualitas proses pembelajaran dengan *Process Mining*, menggunakan *event log* untuk mendapatkan wawasan terkait proses yang diekstraksi. Ada tiga persyaratan yang harus dipenuhi oleh *event log* agar dapat diolah menggunakan *Process Mining*. Pertama, *event log* berisi pengidentifikasi kasus yang menghubungkan beberapa *event* dengan *case* tertentu. Kedua, setiap

event memiliki label yang menghubungkan peristiwa dengan langkah atau contoh aktivitas yang terdefinisi dengan baik dalam proses, yang diidentifikasi dengan nama aktivitas. Ketiga, setiap *event* memiliki stempel waktu (*timestamp*) sedemikian rupa sehingga ada urutan total antara *event* yang terkait dengan satu case (Jans, Soffer & Jouck, 2019).

2.2 Heuristic Miner

Algoritma ini mengekstrak informasi dari data yang diamati dan menghasilkan pola-pola proses yang berbeda yang mungkin tidak diketahui sebelumnya. Parameter yang digunakan dalam penelitian ini adalah *dependency threshold* yaitu parameter yang menunjukkan apakah hubungan diantara dua aktivitas bisa dimasukkan dalam *dependency graph* (model proses) atau tidak. *Dependency Graph* merupakan metrik berbasis frekuensi yang digunakan untuk menunjukkan seberapa jelas bahwa benar-benar ada hubungan ketergantungan antara dua peristiwa A dan B yaitu model yang mewakili ketergantungan antar *events a* ke *b* (notasi $a \rightarrow_w b$), yaitu frekuensi *event a* diikuti dengan *event b* dengan menentukan hubungan nilai *dependency* yang dihitung dengan *dependency measure* untuk melihat frekuensi dependency pada aktivitas. Nilai dependency antar *events* mempengaruhi aktivitas yang muncul dalam *dependency graph* jika nilai *dependency measurement*-nya lebih tinggi atau sama dengan *threshold* maka *event log* akan muncul pada model proses.

Rumus *dependency measure* untuk menentukan *dependency graph* adalah:

$$a \rightarrow_w b = \left(\frac{|a >_w b| - |b >_w a|}{|a >_w b| + |b >_w a| + 1} \right) \quad (1)$$

dengan $a \rightarrow_w b$ adalah *dependency measure* dari *a* ke *b*, $|a >_w b|$ adalah frekuensi *event a* diikuti *event b*, $|b >_w a|$ adalah frekuensi *event a* diikuti *event b* (Van Der Aalst, 2006)

Nilai default *dependency threshold* adalah 0.45 atau 0.5 pada framework PM4PY, agar suatu *event* muncul pada *dependency graph* maka nilai *dependency measure* harus diatas nilai *dependency threshold* yang digunakan (Van Der Aalst, 2006).

Dependency Heuristic Miner tidak secara langsung dapat diinterpretasikan, tetapi karena inilah *Heuristic Miner* memiliki keunikan *dependency* metrik dalam menemukan model atau *event log soundness* (Saint et al., 2021). *Threshold* yang akan digunakan dalam penelitian ini yaitu 0.6, 0.7 dan 0.8 karena pada percobaan menggunakan *threshold* 0.5 *dependency graph* menjadi sulit dibaca karena hasil model menjadi spageti, dan tidak menggunakan *threshold* 0.9 karena menghasilkan model proses dengan aktivitas yang kuat tetapi menghilangkan aktivitas yang infrequent karena dianggap *noise*

namun penting untuk pembacaan analisis model (Maharani et al, Kurniati & Atastina, 2015).

2.3 PM4PY

PM4Py merupakan *library* khusus untuk *Process Mining* dan dapat digunakan dalam proses data mining berbasis Python yang memungkinkan untuk mengeksekusi berbagai proses mining secara otomatis dalam struktur kode Python. PM4Py menyediakan konversi objek *event* data dari format ke format lainnya dan mendukung penggunaan *panda* data frame yang mana efisien jika menggunakan *event* data yang lebih besar serta mendukung *heuristic nets*, menerima *petri nets*, *process tree* dan *transition systems* (Berti, van Zelst & van der Aalst, 2019). *Framework* ini menyediakan beberapa algoritma yang dapat digunakan antara lain *Inductive Miner*, *Alpha Miner* dan *Heuristic Miner*. Pada penelitian sebelumnya (Filipe et al., 2022.) yang mencoba membandingkan 3 *tools* dalam penggunaan *Process Mining* yaitu PM4PY, ProM dan Disco yang menghasilkan bahwa PM4PY menjadi *tools* yang memiliki kelebihan berbiaya rendah dan dapat memaksimalkan kustomisasi dalam kebutuhan *process mining*.

3. IMPLEMENTASI

3.1 Ekstraksi Data

Data diunduh melalui ELING berupa *data log* aktivitas, data yang akan dijadikan objek penelitian adalah kelas mata kuliah Data Warehouse dengan data file format XLS berisi 115.001 rows dari 69 mahasiswa yang mengambil mata kuliah Data Warehouse. Setiap mata kuliah pada ELING memiliki tim *teaching* yaitu tim dosen yang merumuskan Rencana Pembelajaran Semester (RPS) yang menjadi acuan dalam pengajaran mata kuliah tersebut. Selama fase ekstraksi data, disimpulkan bahwa pendekatan ekstraksi *event log* yang ditinjau dapat dikelompokkan berdasarkan jenis sistem informasi yang berbeda (Dakic et al., 2019). Seperti kelas yang akan dijadikan objek penelitian adalah kelas dosen N, dosen S dan dosen NY yang memiliki lebih dari 200 *data log* aktivitas dengan data yang diambil berfokus pada aktivitas modul materi dengan masing-masing data berisi 8 kolom yang terdiri dari kolom Time menunjukkan waktu ketika aktivitas dilakukan, kolom User full name berisi nama user yang mengakses modul, Affected User berisi user yang terdampak ketika suatu aktivitas dilakukan, Event Context berisi nama modul yang diakses, Component menunjukkan letak Event Context yang diakses, Event Name berisi aktivitas yang dilakukan, Description menjelaskan aktivitas yang dilakukan user secara lengkap dengan menggunakan ID user dan ID modul yang diakses, Origin berisi aplikasi yang diakses dan IP Address berisi alamat IP pengguna.

3.2 Preprocessing

Menurut Marin-Castro, dalam konteks *preprocessing event log*, penting untuk mengidentifikasi masalah yang terkait erat dengan kualitas data yang disimpan dalam *event log*. Di satu sisi, dalam *discovery* model proses, *preprocessing* dapat mengurangi kompleksitas model yang didapatkan melalui identifikasi, koreksi, dan penghapusan kesalahan yang terkait dengan *event log* untuk identifikasi model yang benar dan karenanya memungkinkan penemuan model yang lebih terstruktur. Ini akan memfasilitasi interpretasi model yang ditemukan dengan mencoba mempertahankan perilaku asli *event log*. Di sisi lain, teknik *preprocessing* telah digunakan untuk tugas verifikasi kesesuaian antara *event log* dan model yang ditemukan (Marin-Castro and Tello-Leal, 2021).

Preprocessing dengan melakukan pembersihan data dilakukan dengan mengecek status akses pada kolom Event Context dan Event Name, pada proses ini dibuat *sheet* 2 sebagai *sheet* pengecekan yang dilakukan secara manual satu persatu dengan cara melihat Event Name sebagai aktivitas yang relevan terhadap Event Context seperti untuk materi hanya perlu melihat “Course activity completion updated” yang menandakan mahasiswa mengklik link materi/men-download materi tersebut, kemudian Event Context aktivitas melihat link kanal komunikasi baik video dan chat akan dihapus. Setelah itu akan disesuaikan dengan *sheet* 1 untuk mengeliminasi aktivitas yang tidak relevan menggunakan fitur *remove duplicate* untuk menghapus data yang sama pada Event Context, kemudian akan dilakukan pengecekan satu persatu pada Event Name untuk melihat status akses “Course activity completion updated” tersebut, selanjutnya menghapus kolom Affected User, Componen, Description, Origin dan IP Address. Kemudian data hanya diambil dari Februari 2022 sejak dimulainya semester genap 2021/2022 hingga akhir Juni 2022 setelah selesainya ujian akhir semester genap.

Tabel 1 Kolom Awal Data log Raw		
No	Fitur	Keterangan
1.	Time	Timestamp user saat akses LMS (dd/MM/yy, HH:mm).
2.	User full name	Nama user yang mengakses <i>event</i> LMS.
3.	Affected User	User yang terdampak pada aktivitas yang dilakukan user lain.
4.	Event Context	Menu aktivitas yang diakses oleh user.
5.	Component	Label aktivitas <i>event log</i> yang diakses oleh user.
6.	Event Name	Status aktivitas yang dilakukan user saat mengakses <i>event</i> context.
7.	Description	Detail aktivitas yang dilakukan user saat mengakses <i>event log</i> .
8.	Origin	Situs yang digunakan dalam mengakses LMS.
9.	IP Address	Alamat IP user.

Berikut adalah penyesuaian kolom tabel yang akan dipakai:

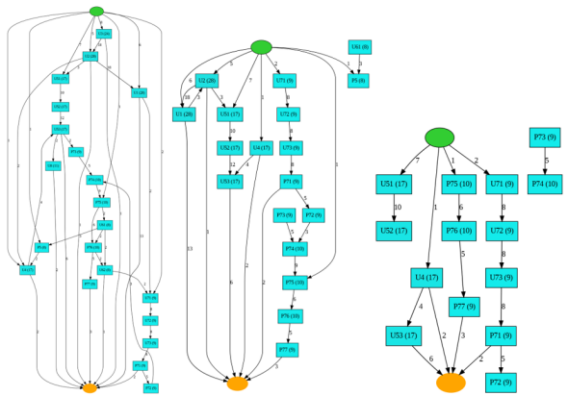
Tabel 2 Kolom Data log Setelah Di-Preprocessing			
No	Fitur	Penyesuaian	Keterangan
1.	User full name	CASE_ID	Digunakan sebagai identifikasi unik dari user.
2.	Event Context	event_context	Menu aktivitas yang diakses oleh user.
3.	Event Name	event_name	Status aktivitas yang dilakukan user saat mengakses Event Context.
4.	Time	Time	Timestamp user saat akses LMS (dd/MM/yy, HH:mm).
5.	Groups	Groups	Kelas yang diikuti oleh user.

3.3 Pengelompokkan Data

Insight dapat diperoleh dari menilai aktivitas mana yang sesuai untuk sub-kelompok atau profil siswa yang berbeda (Fischer et al., 2020). Pada data ini dikelompokkan berdasarkan dosen pengajar di mata kuliah yang sama, dengan *data log* aktivitas pada kelas Dosen N menjadi 276 *log*, kelas Dosen S menjadi 258 *log*, kelas Dosen NY menjadi 395 *log*. Kemudian nama modul materi yang diakses pada kolom *event_context* akan disesuaikan dari topik 1—8 akan diganti menjadi U1—U8 dan P1—P8 yaitu U sebagai materi Utama dan P sebagai materi Penunjang.

3.4 Pemodelan Event Log

Dalam proses pemodelan, data Excel akan diimpor dari Google Drive kemudian diekspor ke CSV untuk dijadikan *event log* dan data frame setelah itu diurutkan aktivitas di dalamnya yang kemudian akan diubah menjadi XES. Ekstraksi *data log* ke dalam data frame menggunakan media Google Colaboratory dengan Python menggunakan *library* PM4PY untuk pemrosesan *Heuristic Miner*. Setelah itu, diproses menjadi model *Heuristic Miner* dengan *dependency threshold* 0.6, 0.7 dan 0.8. Model yang dihasilkan dari masing-masing kelas dosen dalam mengakses materi yang diberikan dapat dilihat pada gambar berikut.



Gambar 3.1 (kiri ke kanan) Kelas Dosen N dengan *Threshold* 0.6, 0.7, 0.8

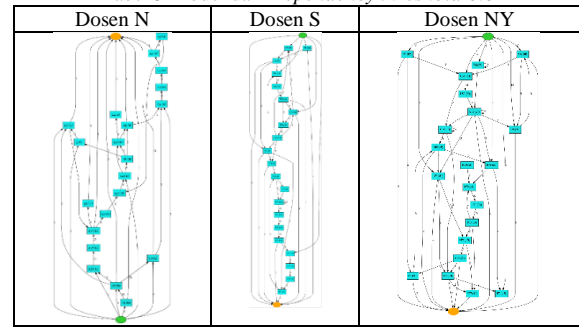
4. HASIL DAN ANALISIS

Dari gambar yang ditampilkan memiliki beragam bentuk pada setiap kelas dosen pengampu dari *threshold* 0.6—0.8 yang menunjukkan bahwa nilai *threshold* mempengaruhi bentuk model yang dihasilkan. Semakin tinggi *threshold* bentuk model yang dihasilkan semakin jelas karena garis varian yang muncul semakin sedikit. Angka pada garis model menunjukkan frekuensi variasi akses *events a* ke *b*, dan angka di dalam *events* menunjukkan frekuensi akses pada modul materi pengajaran. Pada kelas Dosen N, *event* P73 dan P74 terpisah dari *event* lain dalam model, hal ini berarti bahwa suatu *event* tidak harus bergantung pada keseluruhan *event log* selama ada dua *event* ($a \rightarrow_w b$) dengan *dependency* yang kuat maka akan tetap muncul walau terpisah dari keseluruhan model. Perubahan *threshold* mempengaruhi bentuk model yang dihasilkan, dengan angka pada garis model menunjukkan frekuensi variasi akses *events a* ke *b* dan angka di dalam *events* menunjukkan frekuensi akses pada modul materi. Model yang dihasilkan oleh *dependency threshold* yang berbeda-beda menampilkan jumlah *events* yang bervariasi, hasil yang ditunjukkan pada model dipengaruhi oleh *dependency* antar *events* dengan frekuensi siswa dalam mengakses modul tersebut. Model (*dependency graph*) yang dihasilkan dibangun dari *dependency measurement* yang memberikan nilai minimum pada *dependency threshold* agar *events* activity terbaca, maka nilai *dependency measurement* harus berada diatas *dependency threshold* (Van Der Aalst, 2006). Pada tiga nilai *threshold* yang berbeda, dapat diidentifikasi jika nilai *dependency* antar varian (garis pada model) *events a* ke *b* tidak berada diatas nilai *dependency threshold* maka garis tersebut akan hilang tidak peduli betapa banyaknya frekuensi akses pada *events* tersebut. Contohnya seperti pada case U3 dan U51 pada model, dapat dilihat U3 memiliki frekuensi 28 akses case dengan garis varian akses (*events a* ke *b*) dari start to end yang cukup jauh yaitu U3 (8 dan 1), jika *threshold* dinaikkan menjadi 0.7 dan 0.8 maka pada model U3 tidak terlihat tetapi U51 yang memiliki 17 akses case dan start to end yang relative konsisten (7 dan 10 pada U51) tetap muncul pada *threshold* 0.8. Hal ini mengidentifikasikan garis varian akses antar *event* yang cenderung konsisten akan tetap memunculkan case activities yang dimodelkan.

4.1 Threshold 0.6

Pada *threshold* 0.6, ketiga dosen memiliki 21 *log* aktivitas yang terbaca. Alur kerja modul pembelajaran di 0,6 menunjukkan model yang paling kompleks mengingat *event* yang lebih sering harus ditangkap. Pada model ini terlihat bahwa semua kelas mengakses modul materi dari U1—U8 walaupun tidak secara berurutan.

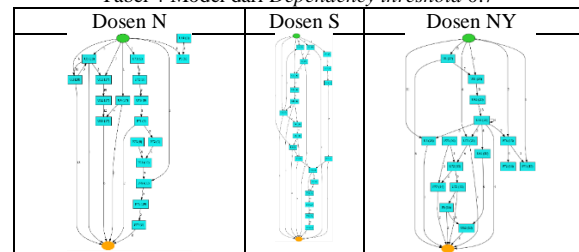
Tabel 3 Model dari *Dependency threshold* 0.6



4.2 Threshold 0.7

Pada *threshold* 0.7, kelas Dosen N memiliki 18 *log* aktivitas yang terbaca, kelas Dosen S memiliki 21 *log* aktivitas yang terbaca dan kelas Dosen NY memiliki 16 *log* aktivitas yang terbaca. Kelas Dosen NY menunjukkan persentase penurunan tertinggi (23%), yang berarti sebagian besar siswa mengambil jalur pembelajaran berbeda yang meningkatkan banyak aliran variasi sehingga ketergantungan antar *event* kurang kuat. Sedangkan, pada kelas Dosen S terlihat bahwa jumlah *event log* tidak berubah yang menunjukkan urutan variasi akses modul materi cenderung seragam.

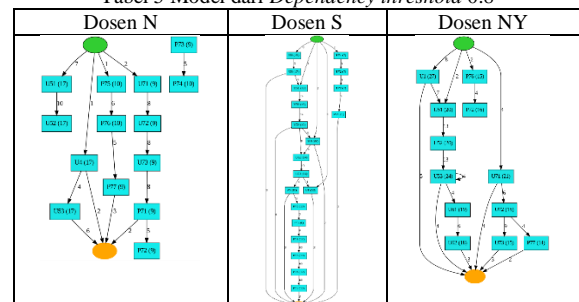
Tabel 4 Model dari *Dependency threshold* 0.7



4.3 Threshold 0.8

Pada *threshold* 0.8, kelas Dosen N memiliki 14 *log* aktivitas kelas Dosen S memiliki 20 *log* aktivitas yang terbaca dan kelas Dosen NY memiliki 12 *log* aktivitas yang terbaca. Kelas Dosen NY mengalami penurunan sebesar 42% dari *threshold* 0.6. *Threshold* yang lebih tinggi tidak memunculkan varian interaksi dengan *dependency* yang tidak kuat. Dengan nilai ini, kelas Dosen S mendapat persentase penurunan terendah dari jalur yang difilter.

Tabel 5 Model dari *Dependency threshold* 0.8



Tabel 6 Ringkasan Jumlah *Event Log* Aktivitas yang Muncul

NO	Informasi Dosen Pengampu	Jumlah Log Activities	Threshold	Jumlah Aktivitas yang dimunculkan
1	Dosen N	276	0.6	21
			0.7	18
			0.8	14
2	Dosen S	258	0.6	21
			0.7	21
			0.8	20
3	Dosen NY	395	0.6	21
			0.7	16
			0.8	12

Model yang dihasilkan dari 3 kelas pada awalnya tidak memiliki karakteristik tertentu, pada *threshold* 0.8 model kelas Dosen S terlihat lebih teratur dan mudah terbaca dibanding dua kelas lainnya. Secara umum, dari ketiganya, urutan kegiatan yang muncul belum berurutan sesuai dengan Rencana Pembelajaran Semester (RPS) yang mempengaruhi bentuk model proses yang dihasilkan dari penyelesaian modul materi. Kelas Dosen S yang memiliki jumlah *log* aktivitas yang terkecil diantara ketiganya memiliki *dependency* yang paling kuat yang terlihat pada jumlah penurunan pada model hampir tidak ada, yang dapat diidentifikasi bahwa mahasiswa pada kelas dosen S mengakses modul materi dengan urutan hampir secara seragam. Tetapi dari ketiga kelas tersebut memiliki kesamaan yaitu modul materi pada U5 dan U7 diakses lengkap bahkan pada kelas Dosen NY yang memiliki penurunan frekuensi kemunculan *event* paling besar. Ini dapat diidentifikasi bahwa materi tersebut menjadi inti dari mata kuliah yang diajarkan sehingga ketiga dosen mata kuliah tersebut telah menekankan dan memandu pentingnya materi dari topik itu. Pada kelas Dosen S dapat dilihat model yang dihasilkan lebih teratur dan semua topik modul materi bahkan pada *threshold* 0.8 tetap muncul dalam model. Hal ini mengidentifikasi gaya mengajar Dosen S lebih menekankan mahasiswa untuk mengakses ELING secara bertahap dalam mempelajari materi yang diberikan, sehingga topik utama U2—U8 bahkan materi penunjang seperti URL dan Repository masih terbaca pada model dan lebih konsisten walau tidak berurutan. Sedangkan pada kelas Dosen N dan Dosen NY, topik utama yang muncul secara konsisten adalah U5 dan U7 yang mengidentifikasi gaya mengajar kedua dosen ini mirip, lebih membebaskan siswa dalam mengakses urutan topik yang diajarkan tetapi tetap menekankan pada konten inti dari mata kuliah tersebut.

KESIMPULAN DAN SARAN

Pada *threshold* dengan nilai terendah, semua topik (topik 1 sampai topik 8) tercakup dalam model di semua kelas, sedangkan dalam *threshold* 0.8 banyak topik yang tidak terbaca pada model, semakin berkurangnya *event log* yang muncul dapat diartikan mahasiswa jarang mengakses modul tersebut secara berulang atau jumlah mahasiswa yang mengakses

modul tersebut lebih sedikit dibanding modul lain. Kemudian, dengan kelas Dosen S yang diidentifikasi lebih menekankan mahasiswa mengakses ELING siswa secara bertahap dalam setiap pertemuan untuk mempelajari konten yang diberikan dan lebih mirip dengan urutan RPS yang diharapkan dalam mengakses ELING tetapi *event* yang muncul belum sesuai urutan yang diharapkan pada RPS, dan jika semua mahasiswa dalam suatu kelas cenderung mengakses urutan materi secara seragam maka aktivitas yang muncul pada model proses yang dihasilkan cenderung stabil dan tidak berkurang ketika *threshold* dinaikkan, sehingga memperkuat bentuk model untuk melihat gaya mengajar dosen. Dalam hal ini restriksi dalam mengakses topik seperti mengakses topik 1 terlebih dahulu untuk membuka kunci restriksi topik 2, dst. bisa menekankan mahasiswa menyelesaikan modul materi secara berurutan. Dengan ini selanjutnya dapat mengidentifikasi proses belajar mahasiswa yang lebih akurat menggunakan lebih banyak bentuk model untuk melihat perbedaan pada model proses dengan menyertakan data kuisioner terhadap metode pembelajaran yang diberikan dosen dan pengaruhnya pada evaluasi belajar untuk meningkatkan proses pengajaran.

DAFTAR PUSTAKA

- ALDAHWAN, N.S. dan AISAEED, N.I., 2020. *Use of Artificial Intelligent in Learning Management System (LMS): A Systematic Literature Review*. *International Journal of Computer Applications*, .
- BERTI, A., VAN ZELST, S.J. dan VAN DER AALST, W., 2019. *Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science*.
- BOGARIN, A., CEREZO, R. dan ROMERO, C., 2017. A survey on *Educational Process Mining*.
- CHANIFAH, S., ANDRESWARI, R. dan FAUZI, R., 2021. Analysis of Student Learning Pattern in Learning Management System (LMS) using *Heuristic Mining a Process Mining Approach*. In: *Proceeding - ICERA 2021: 2021 3rd International Conference on Electronics Representation and Algorithm*. Institute of Electrical and Electronics Engineers Inc. pp.121–125.
- DAKIC, D., VUČKOVIĆ, T., DUSANKA, D., DARKO, S., TEODORA, L., DAJANA, N. dan NENAD, S., 2019. *Event Log Extraction for the Purpose of Process Mining: A Systematic Literature Review Unapređenje konkurentnosti Srbije u procesu pristupanja Srbije Evropskoj uniji View project Designing an Joint Master in Cybersecurity (DJM-CYBER) View project Event Log Extraction for the Purpose of Process Mining: A Systematic Literature Review*.
- FILIPPE, A., GOMES, D., WANZELLER, C. dan FIALHO, J., 2021. *Comparative Analysis of*

Process Mining Tools Sports Dentistry View project Association for Information Systems Association for Information Systems AIS Electronic Library (AISeL) AIS Electronic Library (AISeL) CAPSI 2021 Proceedings Portugal (CAPSI) Comparative Analysis of Process Mining Tools Comparative Analysis of Process Mining Tools.

- FISCHER, C., PARDOS, Z.A., BAKER, R.S., WILLIAMS, J.J., SMYTH, P., YU, R., SLATER, S., BAKER, R. dan WARSCHAUER, M., 2020. Mining Big Data in Education: Affordances and Challenges. *Review of Research in Education*, 44(1), pp.130–160.
- HACHICHA, W., GHORBEL, L., CHAMPAGNAT, R., ZAYANI, C.A. dan AMOUS, I., 2021. Using *Process Mining* for learning resource recommendation: A Moodle case study. In: *Procedia Computer Science*. Elsevier B.V. pp.853–862.
- JANS, M., SOFFER, P. dan JOUCK, T., 2019. Building a valuable *event log* for *Process Mining*: an experimental exploration of a guided process. *Enterprise Information Systems*, 13(5), pp.601–630.
- MARIN-CASTRO, H.M. dan TELLO-LEAL, E., 2021. *Event log preprocessing for Process Mining: A review. Applied Sciences (Switzerland)*.
- SAINT, J., FAN, Y., SINGH, S., GASEVIC, D. dan PARDO, A., 2021. Using *Process Mining* to analyse self-regulated learning: A systematic analysis of four algorithms. In: *ACM International Conference Proceeding Series*. Association for Computing Machinery. pp.333–343.
- TURNBULL, D., CHUGH, R. dan LUCK, J., 2020. Learning Management Systems, An Overview. In: *Encyclopedia of Education and Information Technologies*. Springer International Publishing. pp.1052–1058.
- VAN DER AALST, W., 2006. *Process Mining with the heuristics miner-algorithm*.
- VAN DER AALST, W.M.P., 2011. *Process Mining. Process Mining*. Springer Berlin Heidelberg.
- VERADISTYA MAHARANI, H., KURNIATI, A.P., dan ATASTINA, I., 2015. *Process Mining pada Proses Pengadaan Buku dengan Algoritma Heuristic Miner (Studi Kasus : Perpustakaan Universitas Telkom)*.

Halaman ini sengaja dikosongkan.