

Implementasi Heuristic Miner pada Mata Kuliah Algoritma Pemrograman untuk Mengetahui Pola Belajar Mahasiswa

1st Muhammad Tegar Zharfan

Humam Setiabakti

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

muhammadtegarzhs@student.tel

komuniversity.ac.id

2nd Gede Agung Ary Wisudiawan

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

degunk@telkomuniversity.ac.id

3rd Angelina Prima Kurniati

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

angelina@telkomuniversity.ac.id

Abstrak—*Learning Management System* merupakan sistem yang membantu dosen dalam mengelola kegiatan belajar secara mandiri di instansi. LMS adalah sistem manajemen pembelajaran Universitas Telkom yang berbasis Moodle. Penelitian ini dilakukan *process mining* pada *event log* dari LMS pada mata kuliah Algoritma Pemrograman tahun ajaran 2021/2022 semester genap. Metode *process mining* digunakan untuk mengetahui pola belajar mahasiswa yang lulus dan tidak lulus selama perkuliahan online dengan melihat pola mahasiswa yang digambarkan dalam suatu model proses, sehingga dosen mengetahui pola belajar mahasiswa selama perkuliahan menggunakan LMS. Penerapan algoritma *heuristic miner* menggunakan *Process Mining for Python* (PM4PY). Setelah mendapatkan model proses, dilakukan *conformance checking* nilai *fitness*, mahasiswa yang lulus nilai *fitness* 0,98461 dan untuk mahasiswa yang tidak lulus nilai *fitness* 0,98535. Meskipun nilai *precision* yang dihasilkan tidak cukup baik yaitu, mahasiswa yang lulus nilai *precision* 0,18111 dan untuk mahasiswa yang tidak lulus nilai *precision* 0,27291, Hasilnya *underfitting*, behavior dari model proses yang dihasilkan tidak menggambarkan *behavior* yang ada pada *event log*. Nilai *generalization* yang dihasilkan menunjukkan nilai cukup baik. Mahasiswa yang lulus nilai *generalization* 0,70013 dan untuk mahasiswa yang tidak lulus nilai *generalization* 0,57563. Penelitian ini diharapkan dapat memberikan kontribusi perspektif baru mengenai implementasi *proses mining* di Universitas Telkom menggunakan algoritma *heuristic miner*.

Kata kunci—*learning management system* (LMS), *process mining*, *heuristic miner*,

Conformance Checking, PM4PY

Abstract—*Learning Management System* is a system that helps lecturers in managing learning activities independently in the institution. LMS is a learning management system at Telkom University based on Moodle. This research was carried out by *process mining* on the *event log* from the LMS in the *Programming Algorithm* course for the even semester 2021/2022 academic year. The *process mining* method is used to determine the learning patterns of students who pass and do not pass during online lectures by looking at student patterns depicted in a process model, so that lecturers know student learning patterns during lectures using LMS. The implementation of the *heuristic miner* algorithm uses *Process Mining for Python* (PM4PY). After getting the process model, *conformance checking* of the fitness value is carried out, for students who pass the fitness value of 0.98461 and for students who do not pass the fitness value of 0.98535. Although the resulting precision value is not good enough, namely, students who pass the precision value of 0.18111 and for students who do not pass the precision value of 0.27291, the result is *underfitting*, the behavior of the resulting process model does not describe the behavior in the event log. The resulting *generalization* value shows a fairly good value. Students who pass the *generalization* score is 0.70013 and for students who do not pass the *generalization* score is 0.57563. This research is expected to contribute a new perspective on the implementation of the mining process at Telkom University using the *heuristic miner* algorithm.

Keywords—*learning management system* (LMS), *process mining*, *heuristic miner*, *conformance checking*, PM4PY

I. PENDAHULUAN

Universitas Telkom merupakan salah satu universitas yang menggunakan *Learning Management System* (LMS) untuk media pembelajaran. LMS memiliki fitur-fitur pendukung dengan metode *e-learning* seperti sistem

manajemen pembelajaran memungkinkan mahasiswa untuk melihat kuliah multimedia, berkomunikasi dengan dosen mereka dan satu sama lain dalam komunitas belajar, mengunduh materi pelajaran, mengerjakan kuis online, mengumpulkan tugas, menyediakan materi serta membantu dalam mengelola proses pembelajaran dan interaksi dengan dosen. *Learning Management System* dalam penggunaannya menyimpan banyak rekaman data dengan jumlah besar dan akan selalu bertambah penyimpanan setiap harinya yang akan tercatat dalam bentuk *event log*. *Process mining* bertujuan untuk mengekstrak informasi yang berguna dari *event log* dan menggunakannya sebagai panduan bagi organisasi untuk mengevaluasi proses bisnis mereka sendiri [1].

Penelitian ini bertujuan untuk mengenali proses belajar mahasiswa yang lulus dan tidak lulus menggunakan *Learning Management System* pada mata kuliah Algoritma Pemrograman tahun ajaran 2021/2022 semester genap. Mengenali proses belajar mahasiswa bertujuan untuk mempermudah dosen menyediakan lingkungan yang mendukung, mempermudah mahasiswa menyerap informasi secara maksimal dan digunakan sebagai tolak ukur mahasiswa meningkatkan kualitas pembelajaran. Data yang digunakan adalah data *Event Log* dari mata kuliah Algoritma Pemrograman tahun ajaran 2021/2022 semester genap menggunakan data tersebut karena mata kuliah algoritma pemrograman termasuk mata kuliah yang tingkat ketidaksilesaiannya tinggi dan menggunakan tahun ajaran 2021/2022 semester genap karena sudah menggunakan kurikulum terbaru yang mana semua kegiatan belajar mengajar sudah sepenuhnya menggunakan LMS. Data tersebut akan digunakan untuk menemukan pola dan menganalisis penggunaan mahasiswa terhadap *Learning Management System* (LMS) dengan menggunakan metode *process mining* sebagai pengolahan data. *Process mining* dilakukan menggunakan Algoritma *heuristic miner* untuk mendapatkan suatu model proses karena Algoritma *heuristic miner* dapat merepresentasikan suatu model dengan efektif yang ada di *event log* dengan *noise*, dan dapat menampilkan main behavior dari proses bisnis yang ada [2]. Penerapan algoritma *heuristic miner* menggunakan Process Mining for Python (PM4Py). Untuk melakukan penemuan proses model dan mengevaluasi model Algoritma *Heuristic Miner* menggunakan *process mining conformance checking* (nilai *fitness*, nilai *precision* dan nilai *generalization*) yang ada di PM4PY.

II. KAJIAN TEORI

Bagian ini memberikan gambaran tentang studi terkait yang akan membantu dalam penelitian ini untuk menjadi referensi. Penelitian [3] ini

dilakukan untuk melakukan implementasi *heuristic miner* pada *learning management system*. Data yang digunakan dalam penelitian ini adalah *event log* dari mata kuliah Pengembangan Aplikasi Website dan Sistem Enterprise. Metode *process mining* digunakan untuk melihat pola belajar mahasiswa selama perkuliahan dengan melihat pola mahasiswa yang digambarkan dalam suatu model proses. *Process mining* dilakukan menggunakan ProM 5.2 dan plugin *Heuristic Miner*. Setelah mendapatkan model proses, dilakukan *conformance checking* dengan hasil nilai *fitness*, mata kuliah Pengembangan Aplikasi Website menunjukkan nilai 0,887, dan mata kuliah Sistem Enterprise menunjukkan nilai 0,847. Nilai ini menunjukkan bahwa model proses dapat memodelkan *event log* dengan baik. Pada nilai *precision* mata kuliah Pengembangan Aplikasi Website, nilai *advanced behavioral appropriateness* sebesar 0,594, dan nilai *degree of model flexibility* sebesar 0,181. Sedangkan nilai *precision* mata kuliah Sistem Enterprise menunjukkan nilai *advanced behavioral appropriateness* sebesar 0,411 dan nilai *degree of model flexibility* sebesar 0,115. Pada nilai struktur dihasilkan nilai 1,0 pada kedua mata kuliah dalam penelitian ini.

Penelitian [4], *Heuristic Miner* adalah algoritma *process mining* yang stabil yang terbukti diimplementasikan dengan baik dalam banyak kasus. *Process mining* adalah studi yang relatif baru yang mencoba mengekstrak informasi dari *event log* yang secara otomatis direkam oleh sistem informasi. Penelitian ini melaporkan analisis peneliti setelah menerapkan *heuristic miner* untuk berbagai jenis *event log*. *Heuristic miner* diimplementasikan untuk tugas *process mining*, khususnya tugas penemuan. Ini bermanfaat bagi peneliti dengan studi kasus lain untuk menentukan apakah *heuristic miner* akan menjadi yang terbaik untuk menganalisis log peristiwa mereka.

Penelitian terkait implementasi *heuristic miner* pada *event log* juga sudah dilakukan pada penelitian [5]. *Process mining* dapat meningkatkan analisis proses bisnis pada perusahaan dengan lebih akurat. Ide dasarnya adalah dengan memanfaatkan informasi berupa *event log* yang terekam dalam sistem informasi perusahaan. Pada penelitian ini *process mining* dilakukan terhadap *event log* dari Rabobank Group ICT, sebuah bank yang sedang meningkatkan pelayanannya dengan menelusuri aktifitas yang sudah berlangsung dan memberikan rekomendasi berdasarkan hasil penelusuran tersebut. *Process mining* dilakukan dengan menggunakan *framework process mining*, yaitu PROM 5.2 dan plugins *Heuristic Miner*. Algoritma *heuristic miner* dipilih karena kemampuannya

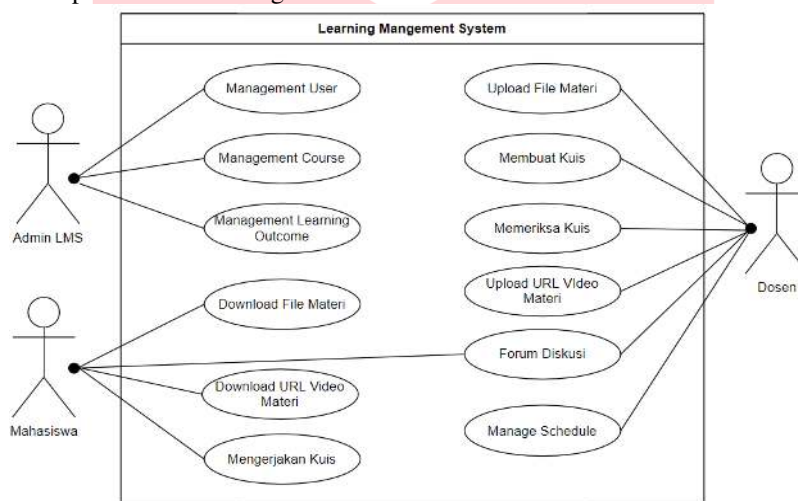
untuk menangani event logs dengan noise, dan dapat menampilkan main behavior dari proses bisnis yang ada [2].

Penulis pada studi [6], menjelaskan Algoritma *Heuristic Miner* adalah algoritma penemuan model proses yang menangani permasalahan *spaghetti processes* dengan berfokus pada jumlah kemunculan relasi antar aktivitas satu dengan lain pada *event log* dalam membentuk model proses. *Spaghetti processes* merupakan *event log* yang memiliki relasi terlalu banyak dan rumit sehingga sulit untuk dibaca [7]. Relasi antar aktivitas dengan jumlah sedikit tidak akan dimunculkan pada model proses yang dibentuk oleh algoritma *Heuristic Miner*. Langkah-langkah yang digunakan dalam pembentukan model proses oleh Algoritma

Heuristic Miner yang pertama adalah Membuat matrik *dependency graph* untuk menyimpan jumlah relasi kebergantungan antar dua aktivitas, menentukan *Dependency threshold*, *And threshold*, *Loop Two threshold* sebagai dasar pemilihan relasi kebergantungan yang akan dimunculkan dalam model proses, memeriksa jika terdapat short loop (length-one-loop atau length-two-loop), menentukan relasi parallel (*XOR* atau *AND*) antar aktivitas dan akhirnya model proses dapat terbentuk.

III. METODE

Gambaran umum dari use case diagram *Learning Management System* Telkom University



GAMBAR 1
USE CASE DIAGRAM LMS

Seperti yang terlihat pada gambar 1 use case diagram *Learning Management System* admin bertugas untuk management user, management course, management Learning Outcome. Sedangkan untuk dosen bertugas untuk upload file materi, membuat kuis, memeriksa kuis, upload URL video materi, forum diskusi yang bisa dilakukan bersama mahasiswa, management schedule. Dan untuk mahasiswa bisa melakukan download file materi, forum diskusi yang bisa dilakukan bersama dosen, download URL video materi, mengerjakan kuis.

A. Pengambilan Data

Dalam penelitian ini objek data yang akan dikumpulkan dan digunakan adalah data dari *event log* LMS Universitas Telkom yang di dapatkan dari *CeLOe*. Didalam *event log* tersebut memuat data berupa rekaman kegiatan yang tercatat dalam mata kuliah. Data yang digunakan untuk penelitian ini adalah data *eventlog* dari mata kuliah Algoritma Pemrograman tahun ajaran 2021/2022 semester

genap jurusan Informatika. Dataset yang gunakan yaitu kelas IF-45-02, IF-45-05, IF-42-07, IF-45-08 dan IF-45-10. Dataset tersebut digunakan karena data nilai kelulusan mahasiswa yang disediakan dari dosen hanya 5 kelas. Didalam *event log* tersebut memiliki data mentah sebanyak 124.316 rows. Data eventlog dipisahkan menjadi 2 bagian antara mahasiswa yang lulus dan tidak lulus.

TABEL 1
JUMLAH *EVENTLOG* LMS

No	Kelas	Jumlah Log (Rows)
1	IF-45-02	43.864
2	IF-45-05	14.331
3	IF-45-07	10.498
4	IF-45-08	16.647
5	IF-45-10	39.273
	Total	124.316

B. Preprocessing

Pada penelitian ini, *preprocessing* yang dilakukan pada data mata kuliah Algoritma

No	Nama Atribut/Kolom Data Mentah	Atribut/Kolom Data Mentah
1	Time	Waktu dimana user akses LMS
2	User full name	Nama user yang melakukan akses di LMS
3	Affected user	User lain yang terlibat dalam aktivitas user pertama
4	Event context	Aktivitas dimana user melakukan akses terhadap menu didalam LMS
5	Component	Label aktivitas dimana user melakukan akses terhadap menu didalam LMS
6	Event name	Action status dari user ketika melakukan akses pada menu yang ada didalam LMS
7	Description	Detail aktivitas dari user saat menjalankan menu yang ada didalam LMS
8	Origin	Media yang digunakan user untuk mengakses LMS
9	IP address	Alamat IP dari user saat mengakses LMS

Pemrograman dengan menggunakan Jupyter Notebook yang bertujuan untuk mengurangi ukuran data, menghapus row yang dilakukan selain mahasiswa, menambah atribut atau kolom baru yang digunakan untuk *process mining* [8]. Dalam tahap *preprocessing* juga melakukan perubahan pada *timestamp* tanggal menjadi DD/MM/YY dan untuk waktu HH/MM. Kemudian Case ID yang ditentukan berdasarkan User Id. Tabel 2 merupakan deskripsi dari data *eventlogs* yang

didapatkan.

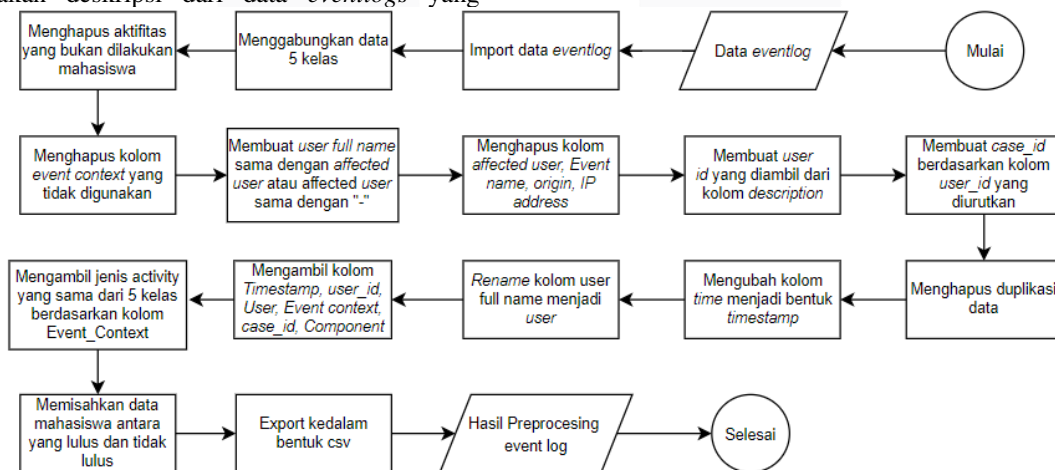
TABEL 2
DESKRIPSI KOLOM *EVENT LOG*

Berikut ini pada tabel 3 adalah contoh data mentah *eventlog Learning Management System Universitas Telkom*.

TABEL 3
CONTOH *EVENT LOG* ALGORITMA PEMOGGRAMAN

No	Nama Atribut/Kolom Data Mentah	Contoh Isi Dari Data Mentah
1	Time	14/02/22, 09:19
2	User full name	FARREL MUHAMMAD AL-FALAH
3	Affected user	-
4	Event context	File: Slide materi
5	Component	File
6	Event name	Course module viewed
7	Description	The user with id '54333' viewed the 'resource' activity with course module id '2060142'.
8	Origin	web
9	IP address	114.122.107.190

Data mentah tersebut tidak memenuhi kriteria untuk pemodelan dalam *Process Mining* dan memerlukan *preprocessing*. Berikut pada gambar 2 merupakan *flowchart* untuk melakukan *preprocessing*.



GAMBAR 2
FLOWCHART *PREPROCESSING*

Pada penelitian ini *preprocessing* dilakukan menggunakan Jupyter Notebook. Tahap pertama yang dilakukan adalah *data integration* yaitu menggabungkan data dari 5 kelas yang berbeda setelah *data integration* selesai selanjutnya melakukan *data cleaning* dari data *eventlog* diolah dengan menghapus aktivitas yang bukan dilakukan oleh mahasiswa. Setelah itu hapus beberapa kolom yang tidak dibutuhkan seperti kolom *affected user*, *Event name*, *origin*, *IP address*. Membuat *user id*

berdasarkan kolom *description* yang dilakukan oleh mahasiswa dalam menggunakan LMS, setelah mendapatkan *user id*, membuat *case id* berdasarkan kolom *user id* yang diurutkan. Selanjutnya menghapus duplikasi data. Mengubah *format time* menjadi *timestamp* DD/MM/YY untuk tanggal dan untuk waktu HH:MM. Mengambil kolom yang dibutuhkan untuk tahap *process mining* seperti kolom *Timestamp*, *user_id*, *User*, *Event context*, *case_id*, *Component*, selanjutnya

mengambil jenis aktivitas yang sama dari 5 kelas yang dilakukan oleh mahasiswa berdasarkan kolom *Event context*, untuk tahap terkahir dari *preprocessing* adalah memisahkan data mahasiswa yang lulus dan tidak lulus. Data yang telah melewati *preprocessing* merupakan kumpulan dataset untuk mata kuliah Algoritma Pemrograman, seperti terlihat pada tabel 4 dan untuk jumlah rows dari hasil *preprocessing* terlihat pada tabel 5.

TABEL 4
CONTOH ISI DATASET SETELAH *PREPROCESSING*

No	Nama Atribut atau Kolom	Contoh Isi Dataset Setelah <i>Preprocessing</i>
1	Timestamp	1/3/2022 12:11:00 AM
2	user_id	'55396'
3	User	FATHAN ASKAR
4	Event context	Quiz: Latihan 01 - IO dan Tipe

		Data
5	case_id	145
6	Component	Quiz

Dataset setelah *preprocessing* memiliki 6 atribut atau kolom dan telah memenuhi kriteria untuk melakukan tahap *process mining*.

TABEL 5
JUMLAH ROWS *EVENTLOG* SETELAH *PREPROCESSING*

No	Kelas	Jumlah Rows	
		Lulus	Tidak Lulus
Total		23.311	2.904

Jumlah *rows event log* setelah tahap *preprocessing* untuk mahasiswa yang lulus memiliki 23.311 *rows* dan untuk mahasiswa yang tidak lulus memiliki 2.904 *rows*.

TABEL 6
AKTIVITAS MAHASISWA SETELAH *PREPROCESSING*

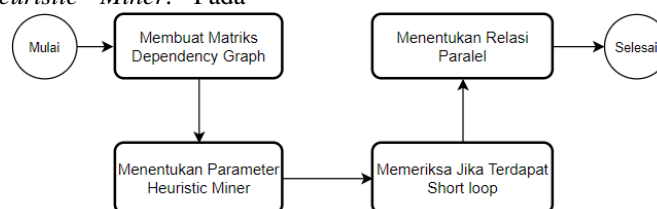
No	Aktivitas
1	Forum: Forum Diskusi
2	Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN
3	Quiz: 3. SESI ESAI
4	Quiz: A. CLO 2 – PILIHAN GANDA
5	Quiz: C. CLO 3 – TRACING
6	URL: Materi Perkuliahan
7	Quiz: B. CLO 2 – ESAI
8	Quiz: D. CLO 3 – ESAI
9	URL: SLIDE PERKULIAHAN (UPDATE BERKALA)
10	File: Materi Perkuliahan
11	URL: Video Rekaman Perkuliahan
12	URL: Link Video Conference Perkuliahan
13	URL: RPS Algoritma Pemograman URL
14	URL: INFO PRAKTIKUM – FIFLAB
15	URL: Link Perkuliahan
16	Forum: Announcements

Seperti yang terlihat pada tabel 6, aktivitas menu LMS mahasiswa memiliki 16 aktivitas.

C. *Process Mining (Discovery)*

Process discovery merupakan sebuah proses ekstraksi data *eventlog* untuk mendapatkan model proses [1] menggunakan PM4PY dengan menerapkan algoritma *Heuristic Miner*. Pada

tahap *process mining (discovery)*, setelah melakukan tahap *preprocessing* terhadap *event log* dan menghasilkan file CSV. Selanjutnya pada tahap *process mining (discovery)* dilakukan penerapan algoritma *heuristic miner* menggunakan *Process Mining for Python (PM4Py)*. Gambar 3 merupakan *flowchart* untuk melakukan *process mining (Discovery)*.



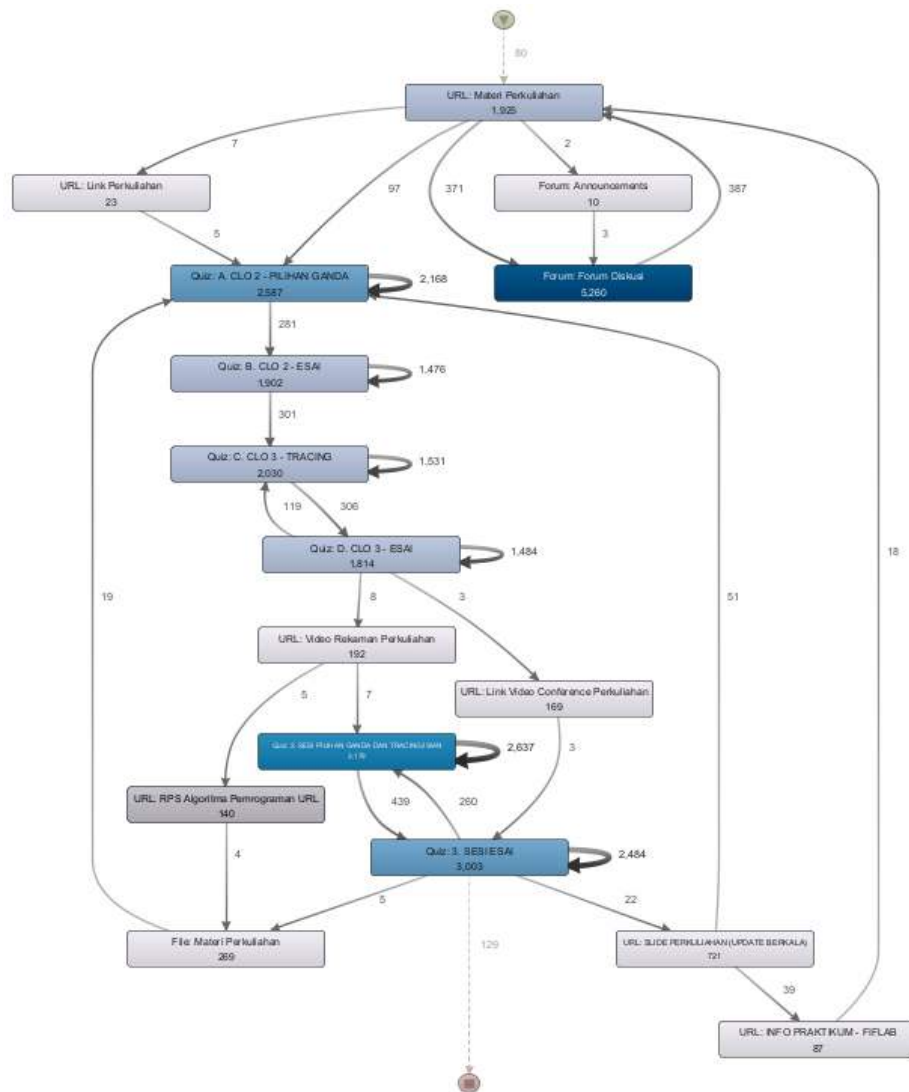
GAMBAR 3.
FLOWCHART *PROCESS DISCOVERY*

Tahapan pertama yang dilakukan pada *process discovery* dengan menerapkan algoritma

Heuristic Miner adalah membuat matriks *dependency graph* untuk menyimpan jumlah relasi ketergantungan antar dua *event*. Selanjutnya menentukan parameter *heuristic miner* seperti *Dependency threshold*, *And threshold* dan *Loop Two threshold* sebagai dasar pemilihan relasi ketergantungan yang akan ditampilkan dalam model proses, setelah itu memeriksa jika terdapat *short loop* (*length-one-loop* atau *length-two-loop*), Menentukan relasi parallel (*XOR* atau *AND*) antar

event/aktivitas. Setelah semua tahap dilakukan model proses algoritma *heuristic miner* dapat terbentuk.

Process *discovery* membutuhkan data input event log untuk membuat model proses yang mewakili proses aktual yang sedang berjalan. Gambar 4 menunjukkan model proses belajar sederhana mahasiswa yang lulus dan gambar 5 menunjukkan model proses sederhana mahasiswa yang tidak lulus menggunakan DISCO tools.



GAMBAR 4.
MODEL PROSES BELAJAR MAHASISWA YANG LULUS DALAM MENGGUNAKAN LMS

Dari gambar 4 dapat dijelaskan bahwa mahasiswa yang lulus mengakses aktivitas pertama kali adalah URL: Materi Perkuliahan setelah mengakses aktivitas tersebut mahasiswa mengakses Forum: Forum Diskusi, Quiz: A CLO 2-PILIHAN GANDA, URL: Link Perkuliahan dan Forum: Announcement. Untuk mahasiswa yang mengakses Forum: Forum Diskusi kembali lagi ke

URL: Materi Perkuliahan. Mahasiswa setelah mengakses Quiz: A CLO 2-PILIHAN GANDA selanjutnya mengakses Quiz: B. CLO 2-ESAI kemudian mengakses Quiz: C. CLO 3-TRACING selanjutnya mengakses Quiz: D. CLO 3-ESAI. Setelah mahasiswa mengakses Quiz: D. CLO 3-ESAI kemudian mahasiswa ada yang mengakses URL: Video Rekaman Perkuliahan dan URL: Link

Video Conference Perkuliahan. Setelah mengakses URL: Video Rekaman Perkuliahan selanjutnya mahasiswa mengakses Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN dan mengakses URL: RPS Algoritma Pemrograman URL. Mahasiswa setelah mengakses Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN dilanjutkan mengakses Quiz: 3 SESI ESAI sama seperti yang dilakukan oleh mahasiswa yang mengakses URL: Link Video Conference

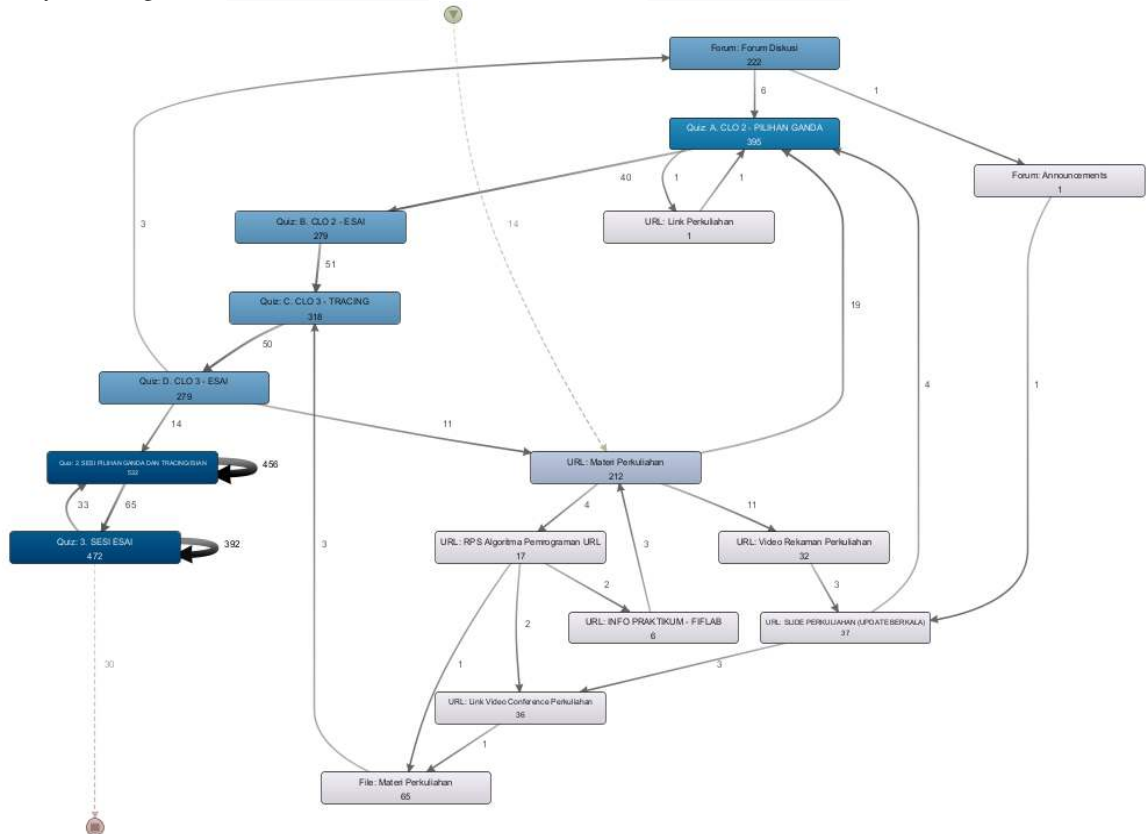
Perkuliahan. Setelah itu mahasiswa ada yang mengakses URL: SLIDE PERKULIAHAN (UPDATE BERKALA) dan File: Materi Perkuliahan kemudian mahasiswa mengakses kembali Quiz: A. CLO 2-PILIHAN GANDA. Setelah mengakses URL: SLIDE PERKULIAHAN (UPDATE BERKALA) mahasiswa mengakses URL: INFO PRAKTIKUM – FIFLAB kemudian kembali lagi mengakses URL: Materi Perkuliahan.

TABEL 7
FREKUENSI AKTIVITAS MAHASISWA YANG LULUS

Aktivitas	Frekuensi
Forum: Forum Diskusi	22,56%
Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN	13,64%
Quiz: 3. SESI ESAI	12,88%
Quiz: A. CLO 2 – PILIHAN GANDA	11,1%
Quiz: C. CLO 3 – TRACING	8,71%
URL: Materi Perkuliahan	8,26%
Quiz: B. CLO 2 – ESAI	8,16%
Quiz: D. CLO 3 – ESAI	7,78%
URL: SLIDE PERKULIAHAN (UPDATE BERKALA)	3,09%
File: Materi Perkuliahan	1,15%
URL: Video Rekaman Perkuliahan	0,82%
URL: Link Video Conference Perkuliahan	0,72%
URL: RPS Algoritma Pemrograman URL	0,6%
URL: INFO PRAKTIKUM – FIFLAB	0,37%
URL: Link Perkuliahan	0,1%
Forum: Announcements	0,04%

Dari model proses belajar mahasiswa yang lulus dapat diketahui bahwa mahasiswa yang lulus lebih banyak mengakses Forum: Forum Diskusi

dibandingkan dengan aktivitas lainnya. Tabel 7 menunjukkan frekuensi seluruh aktivitas mahasiswa yang lulus saat mengakses LMS.



GAMBAR 5
MODEL PROSES BELAJAR MAHASISWA YANG TIDAK LULUS DALAM MENGGUNAKAN LMS

Dari gambar 5 dapat dijelaskan bahwa mahasiswa yang tidak lulus mengakses aktivitas pertama kali adalah URL: Materi Perkuliahan kemudian setelah mengakses aktivitas tersebut mahasiswa mengakses Quiz: A. CLO 2 - PILIHAN GANDA, URL: Video Rekaman Perkuliahan, URL: RPS Algoritma Pemograman URL. Mahasiswa yang telah mengakses Quiz: A. CLO 2 - PILIHAN GANDA kemudian mengakses Quiz: B. CLO 2 - ESAI. Mahasiswa yang telah mengakses URL: Video Rekaman Perkuliahan kemudian mengakses URL: SLIDE PERKULIAHAN (UPDATE BERKALA). Mahasiswa yang telah mengakses URL: RPS Algoritma Pemograman URL kemudian mengakses File: Materi Perkuliahan, URL: Link Video Conference Perkuliahan dan URL: INFO PRAKTIKUM – FIFLAB. Mahasiswa yang telah mengakses URL: SLIDE PERKULIAHAN (UPDATE BERKALA) kemudian mengakses Quiz: A. CLO 2 - PILIHAN GANDA. Mahasiswa yang telah mengakses File: Materi Perkuliahan dan Quiz: B. CLO 2 - ESAI kemudian mengakses Quiz: C. CLO 3 – TRACING. Setelah mengakses aktivitas tersebut mahasiswa ada yang mengakses Quiz: 2 SESI PILIHAN GANDA DAN TRACING/ISIAN dan Forum: Forum Diskusi. Mahasiswa yang telah mengakses Forum: Forum Diskusi kemudian mengakses Forum: Announcements. Mahasiswa yang telah mengakses Quiz: 2 SESI PILIHAN GANDA DAN TRACING/ISIAN kemudian mengakses Quiz: 3. SESI ESAI.

TABEL 8
FREKUENSI AKTIVITAS MAHASISWA YANG TIDAK LULUS

Aktivitas	Frekuensi
Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN	18,32%
Quiz: 3. SESI ESAI	16,25%
Quiz: A. CLO 2 – PILIHAN GANDA	13,6%
Quiz: C. CLO 3 – TRACING	10,95%
Quiz: B. CLO 2 – ESAI	9,61%
Quiz: D. CLO 3 – ESAI	9,61%
Forum: Forum Diskusi	7,64%
URL: Materi Perkuliahan	7,3%
File: Materi Perkuliahan	2,24%
URL: SLIDE PERKULIAHAN (UPDATE BERKALA)	1,27%
URL: Link Video Conference Perkuliahan	1,24%
URL: Video Rekaman Perkuliahan	1,1%
URL: RPS Algoritma Pemograman URL	0,59%
URL: INFO PRAKTIKUM – FIFLAB	0,21%
URL: Link Perkuliahan	0,03%
Forum: Announcements	0,03%

Dari model proses belajar mahasiswa yang

tidak lulus dapat diketahui bahwa mahasiswa yang tidak lulus lebih banyak mengakses Quiz: 2. SESI PILIHAN GANDA DAN TRACING/ISIAN dibandingkan dengan aktivitas lainnya. Tabel 8 menunjukkan frekuensi seluruh aktivitas mahasiswa yang tidak lulus saat mengakses LMS.

Mahasiswa yang lulus sebelum mengakses Quiz: A. CLO 2 – PILIHAN GANDA ada yang terlebih dahulu mengakses URL: Link Perkuliahan sedangkan mahasiswa yang tidak lulus tidak mengakses URL: Link Perkuliahan terlebih dahulu langsung mengakses Quiz: A. CLO 2 – PILIHAN GANDA. Analisis model proses belajar mahasiswa tidak bisa dikaitkan dengan frekuensinya, karena frekuensi hanya untuk mengetahui seberapa banyak aktivitas yang dilakukan oleh mahasiswa.

D. Conformance Checking

Setelah proses *discovery* dilakukan dan hasil dari model proses telah didapatkan, tahap selanjutnya adalah melakukan *conformance checking*. *Conformance checking* dilakukan untuk memverifikasi apakah model proses yang dihasilkan sebelumnya apakah sesuai dengan *eventlog* yang ada, dan sebaliknya [1]. Pada tahap ini dilakukan transformasi dari model proses berbentuk *heuristic net* menjadi *petri net* menggunakan PM4PY. Proses yang sudah berbentuk *petri net* dianalisis dengan *conformance checker*. Hasilnya berupa nilai *fitness*, *precision* dan *generalization* yang menunjukkan tingkat kecocokan antara *eventlog* dan proses model dan juga sebaliknya.

IV. HASIL DAN PEMBAHASAN

Bab ini menganalisis hasil dari bab sebelumnya. Menjelaskan mengenai implementasi *process mining* pada *eventlog* dengan memanfaatkan algoritma *Heuristic Miner*.

A. Conformance Checking

Untuk mendapatkan hasil terbaik dari model proses dalam *process discovery*, penulis melakukan beberapa pengujian *conformance*. Model proses diuji dengan menetapkan beberapa nilai pada parameter untuk menemukan nilai *fitness* yang terbaik. Parameter yang digunakan pada algoritma *heuristic miner* di PM4PY adalah *dependency threshold*, *and threshold*, dan *loop two threshold*. Nilai ketiga parameter tersebut yang ditetapkan pada PM4PY untuk algoritma *heuristic miner* adalah 0,5 untuk *dependency Threshold*, 0,65 untuk *and threshold*, dan 0,5 untuk *loop two threshold*. Nilai ketiga parameter ini ditetapkan berdasarkan nilai default pada PM4PY. Pada

conformance ini salah satu nilai parameter yang digunakan akan diganti secara acak dan nilai parameter lainnya tetap *default*, akan bergantian yang akan diubah. Hasil dari gabungan parameter nantinya akan ditampilkan menggunakan tabel dan dianalisis pengaruh ketiga parameter tersebut terhadap nilai *fitness* yang dihasilkan terhadap model proses.

B. Pengujian *Conformance Checking*

Pengujian digunakan untuk menentukan model proses yang terbaik menggunakan algoritma *Heuristic Miner*, dengan melakukan perbandingan perubahan hasil nilai *fitness* terhadap parameter *dependency threshold*(DT), *and threshold* (AT), dan *loop two threshold*(LTT). Pengujian *conformance checking* menggunakan PM4PY.

1. Analisis Parameter *Dependency Threshold*(DT)

Pengujian terhadap parameter *Dependency Threshold*(DT) berguna untuk menilai kesesuaian dari model proses yang dihasilkan dengan *event log*. *Dependency Threshold*(DT) yaitu parameter yang mengacu pada nilai dari *dependency measure*/frekuensi antar aktivitas. Pengujian ini khusus dilakukan untuk mengetahui pengaruh perubahan nilai DT terhadap nilai *fitness* pada suatu model proses.

TABEL 9
HASIL PENGUJIAN TERHADAP PARAMETER
DEPENDENCY THRESHOLD

Kelulusan	DT	AT	LTT	<i>Fitness</i>
Lulus	0,1	0,65	0,5	0,96370
	0,2	0,65	0,5	0,96917
	0,3	0,65	0,5	0,97654
	0,4	0,65	0,5	0,97800
	0,5	0,65	0,5	0,98461
Tidak Lulus	0,1	0,65	0,5	0,95714
	0,2	0,65	0,5	0,97199
	0,3	0,65	0,5	0,97489
	0,4	0,65	0,5	0,97876
	0,5	0,65	0,5	0,98535

Berdasarkan hasil pengujian seperti yang ditunjukkan pada Tabel 9, mahasiswa yang lulus memiliki nilai *fitness* 0,96370-0,98461, untuk mahasiswa yang tidak lulus memiliki nilai *fitness* 0,95714-0,98535. Yang menunjukkan bahwa nilai *fitness* yang dihasilkan cukup baik untuk menggambarkan *event log* pada model proses. Sehingga dapat dikatakan bahwa semakin tinggi nilai parameter *dependency threshold* maka semakin tinggi nilai *fitness* yang dihasilkan

2. Analisis Parameter *And Threshold*(AT)

Pengujian terhadap parameter *And Threshold*(AT) berguna untuk menilai kesesuaian dari model proses yang dihasilkan dengan *event log*. *And threshold* adalah parameter yang mengacu

pada nilai *frequency*, yang membantu menentukan hubungan antara dua aktivitas tertentu. Pengujian ini khusus dilakukan untuk mengetahui pengaruh perubahan nilai AT terhadap nilai *fitness* pada suatu model proses.

TABEL 10
HASIL PENGUJIAN TERHADAP PARAMETER *AND THRESHOLD*

Kelulusan	DT	AT	LTT	<i>Fitness</i>
Lulus	0,5	0,1	0,5	0,96567
	0,5	0,2	0,5	0,98205
	0,5	0,3	0,5	0,98281
	0,5	0,4	0,5	0,98355
	0,5	0,5	0,5	0,98384
Tidak Lulus	0,5	0,1	0,5	0,95790
	0,5	0,2	0,5	0,98242
	0,5	0,3	0,5	0,98460
	0,5	0,4	0,5	0,98535
	0,5	0,5	0,5	0,98535

Berdasarkan hasil pengujian seperti yang ditunjukkan pada Tabel 10, mahasiswa yang lulus memiliki nilai *fitness* 0,96567-0,98384, untuk mahasiswa yang tidak lulus memiliki nilai *fitness* 0,95790-0,98535. Hal ini menunjukkan bahwa nilai *fitness* yang dihasilkan cukup baik untuk menggambarkan *eventlog* pada model proses. Bahwa seluruh aktifitas yang memiliki *frequency* rendah menyebabkan nilai *fitness* yang dihasilkan menjadi rendah. Sebaliknya jika nilai *and threshold* tinggi akan menyebabkan meningkatnya nilai *fitness*.

3. Analisis Parameter *Loop Two threshold* (LTT)

Pengujian terhadap parameter *Loop two threshold*(LTT) berguna untuk menilai kesesuaian dari model proses yang dihasilkan dengan *eventlog*. *Loop two threshold* (LTT) yaitu parameter batas minimum suatu *dependency* antar dua aktivitas yang terjadi berulang dalam sebuah *case* atau *trace*. Pengujian ini khusus dilakukan untuk mengetahui pengaruh perubahan nilai LTT terhadap nilai *fitness* pada suatu model proses.

TABEL 11
HASIL PENGUJIAN TERHADAP PARAMETER *LOOP TWO THRESHOLD*

Kelulusan	DT	AT	LTT	<i>Fitness</i>
Lulus	0,5	0,65	0,5	0,98461
	0,5	0,65	0,6	0,98442
	0,5	0,65	0,7	0,98392
	0,5	0,65	0,8	0,98388
	0,5	0,65	0,9	0,98136
Tidak Lulus	0,5	0,65	0,5	0,98535
	0,5	0,65	0,6	0,98472
	0,5	0,65	0,7	0,98240
	0,5	0,65	0,8	0,97587
	0,5	0,65	0,9	0,97188

Berdasarkan hasil pengujian seperti yang ditunjukkan pada Tabel 11, mahasiswa yang lulus

memiliki nilai *fitness* 0.98136-0.98461, untuk mahasiswa yang tidak lulus memiliki nilai *fitness* 0.97188-0.98535. Yang menunjukkan bahwa nilai *fitness* yang dihasilkan cukup baik untuk menggambarkan *event log* pada model proses. Sehingga dapat dikatakan bahwa semakin tinggi nilai parameter *loop two threshold* maka semakin rendah nilai *fitness* yang dihasilkan.

C. Evaluasi Hasil Pengujian *Conformance Checking*

Conformance checking melakukan analisis hubungan antara aktivitas yang digambarkan dalam model proses dan *event log* [9]. Penelitian ini menguji kesesuaian model proses belajar mahasiswa dengan *event log* yang ada di LMS dan menganalisis kinerja algoritma *heuristic miner* berdasarkan nilai *fitness*, nilai *precision* dan nilai *generalization*.

1. Hasil pengujian *Conformance Checking*

Pengujian yang telah dilakukan pada perubahan parameter DT, AT dan LTT dari pengujian tersebut menghasilkan model proses yang memiliki nilai *fitness* yang cukup baik. Pada Tabel 12 menunjukkan hasil keseluruhan untuk parameter DT, AT, dan LTT dengan nilai *fitness* terbaik dari mahasiswa yang lulus dan tidak lulus.

TABEL 12
HASIL PENGUJIAN DARI NILAI *FITNESS*

Kelulusan	DT	AT	LTT	<i>Fitness</i>
Lulus	0,5	0,65	0,5	0.98461
Tidak Lulus	0,5	0,5	0,5	0.98535

Dari hasil pengujian nilai *fitness* terbaik untuk mahasiswa yang lulus mendapatkan nilai *fitness* 0,98461 dan mahasiswa yang tidak lulus mendapatkan nilai *fitness* 0,98535.

2. Evaluasi Hasil *Conformance Checking*

Hasil pengujian dari *conformance checking* terdiri dari tiga bagian yaitu, nilai *fitness*, *precision* dan *generalization* dengan menggunakan PM4PY seperti yang ditampilkan seperti pada Tabel 13 berikut ini:

TABEL 13
HASIL PENGUJIAN DARI *CONFORMANCE CHECKING*

Kelulusan	<i>Fitness</i>	<i>Precision</i>	<i>Generalization</i>
Lulus	0.98461	0.18111	0.70013
Tidak Lulus	0.98535	0.27291	0.57563

Evaluasi nilai *fitness* yang terdapat pada Tabel 13 mahasiswa yang lulus mendapatkan nilai *fitness* 0,98461 dan mahasiswa yang tidak lulus mendapatkan nilai *fitness* 0,98535. Hal ini menunjukkan bahwa proses model dapat memodelkan *event log* dengan sangat baik, karena nilai sempurna yang dihasilkan jika seluruh log

secara benar dan sempurna adalah 1. Nilai *fitness* yang cukup baik memiliki arti bahwa model proses yang dibuat sangat baik dan menunjukkan bahwa semua *trace* atau *case* pada *event log* dapat diwakili dengan model prosesnya.

Evaluasi nilai *precision* yang terdapat pada Tabel 13, mahasiswa yang lulus mendapatkan nilai *precision* 0,18111 dan mahasiswa yang tidak lulus mendapatkan nilai *precision* 0,27291. Hal ini menunjukkan bahwa hasilnya belum sempurna atau *underfitting*, karena model yang dihasilkan terlalu menggeneralisasi contoh behavior di *event log*. Model memungkinkan menampilkan behavior yang sangat berbeda dari apa yang terlihat di *event log*.

Evaluasi nilai *generalization* yang terdapat pada Tabel 13, mahasiswa yang lulus mendapatkan nilai *generalization* 0,70013 dan mahasiswa yang tidak lulus mendapatkan nilai *generalization* 0,57563. Ini menunjukkan bahwa hasilnya cukup baik. *Generalization* dikatakan baik jika model yang dihasilkan tidak *overfitting*. Model dikatakan *overfitting* menunjukkan bahwa model tidak cukup menggeneralisasi suatu proses, karena model proses yang dihasilkan terlalu spesifik dan mengikuti contoh pada *event log*.

V. KESIMPULAN

Berdasarkan penelitian ini dapat disimpulkan bahwa pola belajar mahasiswa pada mata kuliah algoritma pemrograman tahun ajaran 2021/2022 semester genap berdasarkan menu LMS mahasiswa yang lulus dan tidak lulus memiliki jumlah aktivitas yang sama yaitu 16 aktivitas. Mahasiswa yang lulus mengakses aktivitas pertama kali adalah URL: Materi Perkuliahan setelah mengakses aktivitas tersebut mahasiswa mengakses Forum: Forum Diskusi, Quiz: A CLO 2-PILIHAN GANDA, URL: Link Perkuliahan dan Forum: Announcement, sedangkan untuk mahasiswa yang tidak lulus mengakses aktivitas pertama kali adalah URL: Materi Perkuliahan kemudian setelah mengakses aktivitas tersebut mahasiswa mengakses Quiz: A. CLO 2 - PILIHAN GANDA, URL: Video Rekaman Perkuliahan, URL: RPS Algoritma Pemrograman URL.

Pada penelitian ini, penerapan algoritma *heuristic miner* dapat memodelkan *event log* pada model proses dengan baik, berdasarkan nilai *fitness* yang didapatkan. Hasil nilai *fitness* yang dihasilkan memiliki nilai yang baik, mahasiswa yang lulus mendapatkan nilai *fitness* 0,98461 dan untuk mahasiswa yang tidak lulus mendapatkan nilai *fitness* 0,98535. Hal ini menunjukkan bahwa proses model dapat memodelkan *event log* dengan cukup baik. Nilai *fitness* yang cukup baik memiliki arti bahwa model proses yang dibuat sangat baik dan menunjukkan bahwa semua *trace* atau *case*

pada *event log* dapat diwakili dengan model prosesnya. Meskipun nilai *precision* yang dihasilkan tidak cukup baik yaitu, mahasiswa yang lulus mendapatkan nilai *precision* 0,18111 dan untuk mahasiswa yang tidak lulus mendapatkan nilai *precision* 0,27291. Hal ini menunjukkan bahwa hasilnya belum sempurna atau *underfitting*, *behavior* dari model proses yang dihasilkan tidak menggambarkan *behavior* yang ada pada *event log* secara umum dengan baik karena model proses yang memiliki cukup banyak jalur untuk merepresentasikan urutan *task* yang berlangsung sesuai dengan *event log* nya. Kemudian untuk nilai *generalization* yang dihasilkan menunjukkan nilai yang cukup baik. Mahasiswa yang lulus mendapatkan nilai *generalization* 0,70013 dan untuk mahasiswa yang tidak lulus mendapatkan nilai *generalization* 0,57563. Hal ini menunjukkan bahwa hasilnya cukup baik.

Saran untuk penelitian adalah selanjutnya dapat dilakukan penelitian *process mining* dengan menerapkan algoritma ataupun metode lain yang berbeda agar model proses yang dihasilkan dapat dijadikan perbandingan dengan algoritma heuristic miner menggunakan PM4PY. Penerapan *process mining* berdasarkan *event log* LMS Universitas Telkom dapat ditingkatkan di penelitian selanjutnya dengan seluruh mata kuliah yang ada di Universitas Telkom untuk meningkatkan kualitas pembelajaran.

REFERENSI

- [1] W. van der Aalst, "Process Mining - Data Science in Action, Second," in *springer*, 2016.
- [2] Alejandro Bogarín, Rebeca Cerezo, Cristóbal Romero, "A survey on educational process mining," *Wires Data Mining and Knowledge Discovery*, 2017.
- [3] Sabila Chanifah, Rachmadita Andreswari, Rokhman Fauzi, "Process Mining," *PENERAPAN PROCESS MINING TERHADAP EVENT LOG UNTUK MENGETAHUI POLA BELAJAR MAHASISWA DENGAN MENGGUNAKAN ALGORITMA HEURISTIC MINER (Studi Kasus : Learning Management System (LMS) Universitas Telkom)*, 2021.
- [4] Angelina Prima Kurniati, Guntur Prabawa Kusuma, Gede Agung Ary Wisudiawan, "Menerapkan Penambang Heuristik untuk Berbagai Jenis Log Peristiwa," *Jurnal Internasional Riset Teknik Terapan*, vol. 11, 2016.
- [5] Rendy Setiadi Mangunsong, Angelina Prima Kurniati, Mira Kania Sabariah, "Analisis dan Implementasi Process Mining dengan Algoritma Heuristic Miner," *e-Proceeding of Engineering*, vol. 2, p. 2016, 1681.
- [6] R. Sarno, Y. A. Effendi, and F. Haryadita, "Modified Time-Based Heuristics Miner for Parallel Business," *IRECOS (International Rev. Comput. Software)*, vol. 11, no. 3, p. 249–260, 2016.
- [7] A. K. A. De Medeiros et al, "Process mining based on clustering: A quest for precision," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4928, p. 17–29, 2018.
- [8] Dicky Nofriansyah, Kamil Erwansyah, Mukhlis Ramadhan, "Penerapan Data Mining dengan Algoritma Naive Bayes Clasifier untuk Mengetahui Minat Beli Pelanggan terhadap Kartu Internet XL (Studi Kasus di CV. Sumber Utama Telekomunikasi)," *Jurnal Ilmiah Saintikom*, pp. 1978-6603, 2016.
- [9] J. Carmona, B. v. Dongen, A. Solti dan M. Weidlich, "“Relating Processes and Models” dalam Conformance Checking , Springer," vol. XIV, no. 270, 2018.