

## Analisis dan Penerapan *Process Mining* Pada Data Perkuliahan *Online Studi Kasus Virtual Class Universitas Lampung*

<sup>1</sup>Suci Hasanah Bertha, <sup>2</sup>Astria Hijriani, <sup>\*3</sup>Yunda Heningtyas, dan <sup>4</sup>Wartariyus

<sup>1,2,3,4</sup>Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung

e-mail: <sup>1</sup>[suci.hasanah1003@students.unila.ac.id](mailto:suci.hasanah1003@students.unila.ac.id), <sup>2</sup>[astria.hijriani@fmipa.unila.ac.id](mailto:astria.hijriani@fmipa.unila.ac.id), <sup>\*3</sup>[yunda.heningtyas@fmipa.unila.ac.id](mailto:yunda.heningtyas@fmipa.unila.ac.id), <sup>4</sup>[wartariyus@fmipa.unila.ac.id](mailto:wartariyus@fmipa.unila.ac.id)

---

**Abstract** — Increasing data innovation collectively affects all regions of the association. The development of the Corona virus episode towards the beginning of 2020 has made all forms of movement carried out online. Lampung University is one of the institutions affected by the Covid-19 outbreak. The rampant transmission of this virus requires that all the academics of the Lampung campus complete internet learning until things return to normal. All learning exercises are diverted by utilizing an online-based framework called Virtual Class University of Lampung. This research uses process mining. Process mining is a strategy that applies unique calculations to record information. Process mining is a valuation strategy between cycle models and event or event log information contained in a data frame. Event Logs can be easily incorporated into process models using heuristic mining algorithms. The best cycle model is obtained by utilizing three constraints, namely Relative-to-best Threshold (RT), Positive Observations Threshold (PT), and Dependency Threshold (DT). This limit is used to find the best health value. Health values are used to demonstrate model cycles by logging true or false events. If they match, the bottleneck found is valid or actually occurred. In addition, for this situation study, the soundness value obtained indicates suitability so that the obstacles that occur can be found. Each subject has a different frequency for activities completed in the Virtual Class. Process mining is the act of creating a cyclical model from the log of events that occurred, planning to work in the most common way of identifying and solving a problem. For this case the event log used is the event log from Virtual Class University of Lampung and handle mining can help distinguish bottlenecks that occur in the selected course.

**Keywords:** Process Mining; Virtual Class; Online Lecture Data.

---

### 1. PENDAHULUAN

Peningkatan inovasi data mempengaruhi seluruh bidang. Berkembangnya virus Corona pada awal tahun 2020 membuat segala bentuk kegiatan yang seharusnya dilakukan secara langsung, harus dilakukan secara tidak langsung. Universitas Lampung merupakan salah satu institusi atau organisasi yang terdampak oleh wabah Covid-19. Maraknya penularan ini menuntut seluruh aktivitas akademika Universitas Lampung menyelesaikan pembelajaran internet sampai keadaan kembali normal seperti biasa. Dengan kebijakan yang baru, tenaga pengajar diharapkan untuk tetap dapat memberikan pengajaran kepada semua mahasiswa secara *online*. Hal ini membuat seluruh lingkungan kampus, baik tenaga pengajar maupun mahasiswa menggunakan sebuah *framework* yang dapat membantu pembelajaran jarak jauh. Seluruh pembelajaran dialihkan dengan memanfaatkan *framework* berbasis *online* bernama *Virtual Class* Universitas Lampung yang menggunakan metode *e-learning* seperti daftar kehadiran *online*, soal berbasis *online*, dan penyediaan materi pembelajaran. Semenjak terjadinya pandemi hingga masa *new normal*, *VClass* semakin sering digunakan karena sangat membantu pembelajaran dalam menghadapi situasi yang tidak memungkinkan secara *offline* serta lebih fleksibel.

Data adalah realitas mengenai objek. Data juga dapat diartikan sebagai fakta mendasar yang dirumuskan dalam suatu kelompok atau lembaga tertentu untuk menunjukkan aktivitas, tindakan, dan peristiwa yang tidak memiliki arti dan tidak berpengaruh ketika digunakan [1]. Informasi yang belum penting kemudian dapat diolah menjadi data yang signifikan. Seperti kerangka pembelajaran pada umumnya, *VClass* menyimpan banyak catatan data dan akan terus bertambah seiring bertambahnya jumlah siswa setiap tahun. Catatan informasi tersebut akan menyimpan latihan siswa yang berbeda secara konsisten sebagai *event log*. *Event log* adalah catatan aktivitas atau transaksi pengguna pada suatu sistem dan berisi informasi tentang

---

aktivitas saat ini [2]. Proses bisnis lembaga atau organisasi adalah salah satu jenis informasi yang dapat diperoleh dari aktivitas log peristiwa.

*Business Process Management* (BPM) digunakan untuk menganalisis, mengidentifikasi, dan memantau proses untuk memaksimalkan kinerja [3]. Teknik ini sering digunakan oleh organisasi atau perusahaan untuk terus meningkatkan kinerja dalam jangka panjang. BPM akan membantu menangani seluruh perkembangan kinerja untuk pengambilan keputusan terbaik sehingga dapat mengurangi peluang kesalahan dan kerugian bagi institusi. Untuk mendapatkan data dan informasi dari *event log*, diperlukan pengolahan data. Salah satu cara untuk menangani informasi tersebut adalah dengan memanfaatkan *process mining*. *Process mining* adalah cara untuk mendapatkan informasi dari proses transaksi yang terjadi. *Process Mining* terdiri dari tiga fase: *Process Discovery*, *Process Conformance Checking*, dan *Process Enhancement* [4]. Ketiga tahap ini digunakan untuk memproses dan mengubah *log* peristiwa yang ada menjadi informasi yang dapat diambil kesimpulannya. Kesimpulan ini akan dijadikan pertimbangan untuk membantu pengambilan keputusan.

Aktivitas pembelajaran yang dilakukan oleh siswa seringkali berbeda, sehingga menyebabkan perubahan nilai akhir. Aktivitas ini dapat dianalisis untuk mendapatkan data yang mempengaruhi para siswa tersebut sehingga mereka mendapatkan nilai yang berbeda satu sama lain. Jika seorang mahasiswa tidak lulus suatu mata kuliah, maka mahasiswa harus mengulang mata kuliah tersebut yang menimbulkan kemunduran waktu kelulusan. Aktivitas para mahasiswa tersebut dapat berpengaruh untuk jangka panjang. Solusi atas permasalahan tersebut adalah dengan menganalisis aktivitas pembelajaran mahasiswa Universitas Lampung pada VClass dengan memanfaatkan *process mining*. Analisis ini dilakukan dengan tujuan untuk terus meningkatkan proses bisnis di Universitas Lampung dan mencari tahu hambatan apa yang terjadi selama proses pembelajaran.

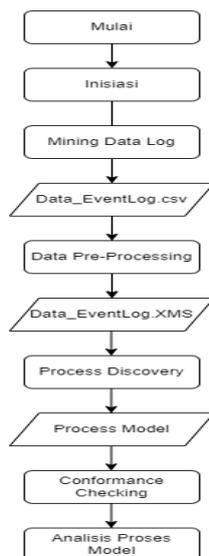
Penelitian ini didasari oleh belum adanya penelitian mengenai aktivitas pada sistem belajar di Jurusan Ilmu Komputer, Universitas Lampung. Informasi yang digunakan adalah *event log* dari mata kuliah Pemrograman Terstruktur dan Aplikasi Android dengan *Flutter* tahun akademik 2021/2022. Dua mata kuliah ini dipilih setelah penyaringan dari beberapa mata kuliah yang ada, dimana kedua mata kuliah tersebut memiliki kriteria VClass yang paling ideal. Data *event log* untuk mata kuliah ini akan digunakan untuk membandingkan aktivitas menggunakan metode *process mining* berdasarkan aktivitas mahasiswa pada VClass. Teknik ini akan mengolah *event log* yang merupakan representasi dari proses bisnis. *Event log* kemudian akan ditampilkan menjadi proses model untuk dievaluasi dan dipertimbangkan. *Process mining* dilakukan dengan menggunakan algoritma *heuristic miner*. Untuk dapat memodelkan algoritma tersebut, diperlukan perangkat atau aplikasi, yaitu ProM. ProM adalah aplikasi yang dapat digunakan untuk melakukan *process mining* mulai dari informasi dasar yang diolah menjadi *event log* yang sesuai, dan kemudian dimodelkan berdasarkan algoritma yang dipilih. Model proses yang dihasilkan akan memudahkan analisis proses bisnis. Dari penelitian ini diharapkan latihan pembelajaran mahasiswa di VClass Unila dapat dijadikan tolak ukur dasar untuk perbaikan strategi pembelajaran selanjutnya sehingga mahasiswa dapat memperoleh nilai tinggi secara keseluruhan.

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan *process mining*, yaitu strategi yang menerapkan perhitungan unik untuk mencatat informasi. *Process mining* merupakan strategi penilaian antara model siklus dan informasi peristiwa atau *event log* yang terkandung dalam kerangka data [5].

### 2.1. Tahapan Penelitian

Langkah-langkah yang akan dilakukan peneliti untuk menyelesaikan penelitian disebut sebagai tahapan penelitian. Untuk melakukan *process mining*, diperlukan sejumlah proses dengan sejumlah fase yaitu fase inisiasi, penambahan *log* data (*mining data log*), pra-pemrosesan data (*pre-processing*), penemuan proses (*process discovery*), pemeriksaan kesesuaian (*conformance checking*), dan analisis hasil proses model. Gambar 1 menunjukkan alur tahapan untuk menyelesaikan penelitian.



Gambar 1. Tahapan penelitian.

### 2.1.1. Fase Inisiasi

Tahapan utama adalah survei untuk menemukan permasalahan yang terjadi dan memutuskan perpanjangan batas penelitian. Masalah yang diangkat adalah mengkaji aktivitas pembelajaran di VClass Universitas Lampung. Setelah perincian masalah dibuat, studi literatur dan pendalaman materi dimulai untuk menemukan solusi yang akan dilakukan terhadap masalah yang terjadi. Solusi yang dipilih untuk penelitian ini adalah dengan memanfaatkan *Process Mining*.

### 2.1.2. Penambangan Log Data (*Mining Data Log*)

Tahapan ini merupakan tahapan yang dilakukan setelah siklus awal selesai. Pada tahap ini ditargetkan untuk menemukan dan mengelola catatan peristiwa atau latihan mahasiswa pada mata pelajaran tertentu untuk digunakan sebagai analisis melalui VClass. Informasi yang didapat masih berupa informasi mentah yang masih harus diolah lagi untuk mendapatkan *log* sesuai kebutuhan penelitian.

### 2.1.3. Pra-Pemrosesan Data (*Pre-Processing*)

Tahap *pre-processing* dilakukan setelah tahap penambangan informasi *log* selesai. Tahap ini merupakan tahap untuk mengidentifikasi informasi yang penting dan menghilangkan informasi yang tidak berguna [6]. Siklus *preprocessing* diselesaikan dengan bantuan aplikasi Microsoft Excel. Langkah pertama dalam proses *preprocessing* adalah mengidentifikasi kolom yang akan digunakan dan membuang kolom yang tidak sesuai atau tidak diperlukan. Dalam penelitian ini informasi yang digunakan adalah informasi waktu akses, *username*, *event name*, dan *event context*. Setelah informasi dibersihkan maka dapat digunakan pada tahap selanjutnya.

### 2.1.4. Penemuan Proses (*Process Discovery*)

Setelah tahap *preprocessing* selesai dan informasi telah dibersihkan sesuai kebutuhan, tahap selanjutnya adalah proses *discovery*. Data yang telah siap digunakan diubah pada tahap ini menjadi proses model menggunakan aplikasi ProM, yaitu aplikasi yang dapat memvisualisasikan hasil proses model. Algoritma yang digunakan untuk mengubah *event log* menjadi proses model pada penelitian ini adalah algoritma *heuristic miner*. Dalam proses transformasi ini, data awal berupa *event log* diubah menjadi proses model berupa *heuristic net* [7]. Hubungan atau koneksi *event log* antara proses dan aktivitas dijelaskan oleh algoritma *heuristic miner*.

### 2.1.5. Pemeriksaan Kesesuaian (*Conformance Checking*)

Tahap selanjutnya adalah tahap *conformance checking*. Tahapan ini dilakukan setelah *heuristic net* terbentuk sesuai dengan kebutuhan [8]. Dalam tahapan ini, *heuristic net* diubah menjadi *petri net*. Setelah itu, *petri net* divalidasi untuk memeriksa apakah hasil proses model *heuristic* cocok dengan

parameter proses model awal. Hasil dari *conformance checking* ini adalah nilai *fitness* yang menunjukkan tingkat kesamaan antara proses model dan *event log* atau sebaliknya. Setelah tahap ini selesai, hasil proses model dapat dianalisis atau dilanjutkan ke tahap terakhir.

**2.1.6. Analisis Hasil Proses Model**

Tahapan selanjutnya dalam penelitian ini adalah analisi proses model. Proses model yang telah dibuat dianalisis sesuai dengan perincian masalah yang telah dibuat. Pada tahap ini diharapkan dapat mendapatkan hasil analisis berupa aktivitas belajar mahasiswa yang terbentuk dari *event log* yang telah ditransformasi menjadi proses model. Analisis dilakukan untuk mendapatkan jawaban dari rumusan masalah yang telah diajukan. Hasil dari proses model dapat ditarik kesimpulan setelah melalui proses analisis kesesuaian dalam proses model dan hasilnya sesuai dengan *event log* yang ada.

**3. PEMBAHASAN**

**3.1 Implementasi**

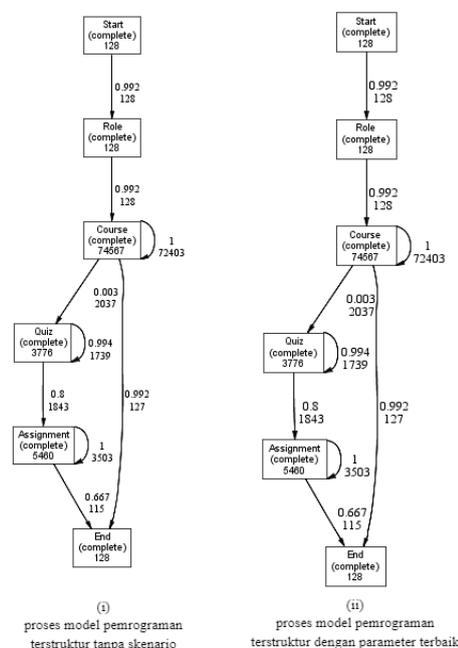
**3.1.1 Hasil Analisis Mata Kuliah Pemrograman Terstruktur menggunakan ProM dan Disco**

**a) Pengujian Parameter RT, PT, dan DT**

Pengujian ini dilakukan dengan mengelompokkan *event log* ke dalam tiga parameter dengan batasan yang berbeda yaitu *Relative to-best Threshold* (RT), *Positive Observation Threshold* (PT), dan *Dependency Threshold* (DT). Pengujian ini dilakukan dengan menggunakan algoritma *heuristic miner* untuk menemukan nilai *fitness* tertinggi. Nilai tersebut diuji dengan mengubah nilai RT, PT, dan DT.

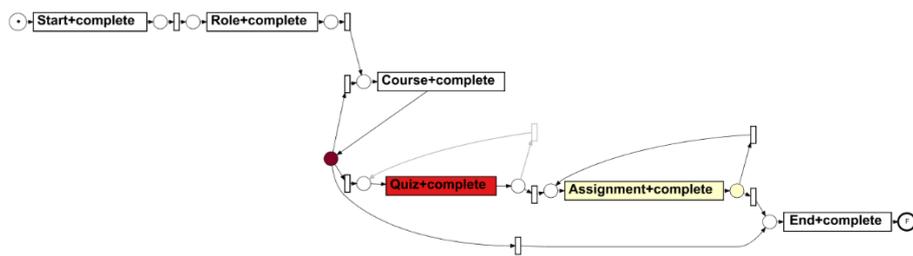
**b) Pengujian Proses Model dan Conformance Checking**

Setelah mendapatkan nilai *fitness* terbaik dari parameter yang ada, didapatkan hasil perbandingan antara penggunaan setelan parameter terbaik dan tanpa penggunaan skenario. Parameter terbaik yang didapatkan yaitu dengan nilai RT 0,05, PT 10, dan DT 0,3. Berdasarkan hasil pengujian parameter, proses model ini merupakan model terbaik. Hasil yang didapat adalah tidak adanya perbedaan yang terjadi pada kedua model siklus tersebut. Hal ini menunjukkan bahwa tanpa adanya setelan parameter, nilai seluruh aktivitas telah memenuhi aturan untuk digunakan sebagai proses model.



Gambar 2. Perbandingan proses model mata kuliah Pemrograman Terstruktur dengan nilai parameter terbaik.

Langkah selanjutnya adalah melakukan *conformance checking* untuk menemukan nilai *fitness*, *precision*, dan *structure*. *Conformance checking* juga dilakukan untuk menemukan *bottleneck* pada *event log*. Gambar 3 menunjukkan proses model yang telah dilakukan *conformance checking*.



Gambar 3. Hasil *conformance checking* pada mata kuliah Pemrograman Terstruktur.

Tanda merah dan kekuningan pada bagian *Quiz* and *Assignment* berdasarkan hasil *conformance checking* menunjukkan bahwa kegiatan tersebut terdapat *bottleneck*. Warna ini menunjukkan adanya proses yang terhambat dan akan ditunjukkan dengan menganalisis aktivitas menggunakan aplikasi Disco. Hasil dari *conformance checking* yang didapat dari Gambar 3 ditampilkan pada Tabel 1.

Tabel 1. Hasil *conformance checking* pada mata kuliah Pemrograman Terstruktur.

Nilai	Hasil
<i>Fitness</i>	0.9878
<i>Advance behavioral appropriateness</i>	0.8295
<i>Degree of model flexibility</i>	0.2500
<i>Structure</i>	1.0000

Tabel 1 menunjukkan nilai *fitness*, *precision*, dan *structure*. Nilai *fitness* menunjukkan angka 0.9878 dan nilai tertinggi untuk parameter *fitness* adalah 1. Hal ini menunjukkan bahwa proses model dapat memodelkan *event log* secara akurat dan hampir sempurna. Nilai *precision* menunjukkan angka 0.8295 pada *advanced behavioral appropriateness*, dan 0.25 pada *degree of model flexibility*. Hal ini menunjukkan bahwa proses model sangat fleksibel sehingga memungkinkan banyak variasi yang terjadi pada *event log*. Nilai *structure* menunjukkan angka 1. Hal ini menunjukkan bahwa tidak terdapat *duplikasi task* atau *task* yang berulang selama *event log* terjadi.

### c) Analisis menggunakan Aplikasi Disco

Pada tahap ini dilakukan *cross check* pada aplikasi Disco untuk menunjukkan apakah terdapat *bottleneck* pada aktivitas berdasarkan hasil dari *conformance checking* sebelumnya. Hasil analisis aktivitas pada mata kuliah Pemrograman Terstruktur ditampilkan pada Tabel 2. Aktivitas ini diperoleh dengan mengakumulasikan jumlah frekuensi aktivitas per kategori nilai akhir.

Tabel 2. Hasil analisis aktivitas mahasiswa pada mata kuliah Pemrograman Terstruktur.

Nilai	Aktivitas
A	600+ aktivitas
B+	400-600 aktivitas
B	300-400 aktivitas
C+	100-300 aktivitas
C	40-100 aktivitas
D dan E	<40 aktivitas

Dari hasil pada Tabel 2, semakin banyak frekuensi aktivitas yang dilakukan mahasiswa pada mata kuliah Pemrograman Terstruktur di VClass membuat nilai yang didapat mahasiswa tersebut semakin tinggi. Begitupun sebaliknya, semakin sedikit frekuensi aktivitas, maka nilai yang didapat semakin rendah. Selain banyaknya frekuensi aktivitas, frekuensi akses *Quiz* dan *Assignment* juga dapat mempengaruhi nilai akhir mahasiswa. Tabel 3 menunjukkan daftar frekuensi akses *Quiz* dan *Assignment* pada mata kuliah pemrograman terstruktur.

Tabel 3. Frekuensi akses *Quiz* dan *Assignment* pada mata kuliah Pemrograman Terstruktur.

Nilai	Quiz	Assignment
A	37.0%	55.4%
B+	30.8%	42.0%
B	28.2%	34.0%
C+	14.3%	16.3%
C	2.0%	3.5%
D dan E	0.0%	0.2%

Mahasiswa yang mendapatkan nilai A cenderung mengakses lebih banyak aktivitas *Quiz* serta *Assignment* dan sebaliknya, mahasiswa yang mendapat nilai rendah cenderung jarang dan hampir tidak pernah mengakses aktivitas *Quiz* serta *Assignment*. Kesimpulan yang diperoleh berdasarkan Tabel 3 adalah frekuensi akses *Quiz* dan *Assignment* dapat mempengaruhi nilai dari para mahasiswa. Dari hasil tersebut dapat dibuktikan bahwa benar terdapat *bottleneck* atau hambatan pada aktivitas *Quiz* dan *Assignment* yang mempengaruhi variasi nilai akhir para mahasiswa sesuai dengan hasil *conformance checking* pada aplikasi ProM. Durasi bergabung pada suatu kelas juga dapat menjadi salah satu pengaruh dalam mendapatkan nilai. Gambar 4 menunjukkan contoh waktu durasi pada mahasiswa yang mendapatkan nilai rendah.

Case ID	Events	Started	Finished	Duration
80-D	20	22.03.2021 07:18:00	13.08.2021 16:00:00	144 days, 8 hours
117-D	18	30.06.2021 13:48:00	13.08.2021 16:00:00	44 days, 2 hours
112-D	16	15.06.2021 18:18:00	13.08.2021 16:00:00	58 days, 21 hours
104-D	13	23.04.2021 22:35:00	13.08.2021 16:00:00	111 days, 17 hours
65-D	11	04.06.2021 14:56:00	13.08.2021 16:00:00	70 days, 1 hour
93-E	11	28.04.2021 07:41:00	13.08.2021 16:00:00	107 days, 8 hours
127-E	11	07.06.2021 08:25:00	13.08.2021 16:00:00	67 days, 7 hours
102-E	9	10.06.2021 07:22:00	13.08.2021 16:00:00	64 days, 8 hours
96-E	7	16.06.2021 14:50:00	13.08.2021 16:00:00	58 days, 1 hour
121-E	7	09.06.2021 09:41:00	13.08.2021 16:00:00	65 days, 6 hours
38-E	6	16.05.2021 10:16:00	13.08.2021 15:05:00	89 days, 4 hours

Gambar 4. Jangka waktu mahasiswa bergabung pada mata kuliah Pemrograman Terstruktur.

Gambar 4 menunjukkan hasil rata-rata mahasiswa yang mendapat nilai D dan E pada mata kuliah Pemrograman Terstruktur memiliki durasi *join* lebih pendek dibandingkan dengan waktu *join* seharusnya yakni 144 hari atau satu semester pembelajaran. Durasi pengerjaan *quiz*/ujian dapat dianalisis untuk melihat apakah waktu pengerjaan dapat berpengaruh pada nilai akhir yang didapat. Gambar 5 menunjukkan durasi waktu pengerjaan ujian tengah semester pada mata kuliah Pemrograman Terstruktur.

Case ID	Events	Variant	Started	Finished	Duration
12-A	2	Variant 1	16.05.2021 09:20:00	16.05.2021 12:38:00	3 hours, 18 mins
90-A	2	Variant 1	16.05.2021 10:10:00	16.05.2021 13:25:00	3 hours, 15 mins
107-A	2	Variant 1	16.05.2021 09:16:00	16.05.2021 12:29:00	3 hours, 13 mins
66-B	2	Variant 1	16.05.2021 10:10:00	16.05.2021 13:16:00	3 hours, 6 mins
124-A	2	Variant 1	16.05.2021 10:23:00	16.05.2021 13:28:00	3 hours, 5 mins
57-A	2	Variant 1	16.05.2021 09:16:00	16.05.2021 12:16:00	3 hours
82-B	2	Variant 1	16.05.2021 10:21:00	16.05.2021 13:13:00	2 hours, 52 mins
7-B	2	Variant 1	16.05.2021 09:18:00	16.05.2021 12:03:00	2 hours, 45 mins
13-B+	2	Variant 1	16.05.2021 09:21:00	16.05.2021 11:52:00	2 hours, 31 mins
79-C+	2	Variant 1	16.05.2021 10:27:00	16.05.2021 12:52:00	2 hours, 25 mins
74-C+	2	Variant 1	16.05.2021 11:10:00	16.05.2021 13:30:00	2 hours, 20 mins
30-C+	2	Variant 1	16.05.2021 11:02:00	16.05.2021 13:18:00	2 hours, 16 mins
62-A	2	Variant 1	16.05.2021 09:15:00	16.05.2021 11:22:00	2 hours, 7 mins

Gambar 5. Durasi waktu pengerjaan UTS Pemrograman Terstruktur.

Gambar 5 menunjukkan bahwa durasi waktu pengerjaan *quiz*/ujian yang diakses mahasiswa adalah rata-rata sama. Hal ini karena waktu pengerjaan yang telah ditetapkan sebelumnya, sehingga mahasiswa serentak memulai *quiz*/ujian di waktu bersamaan, begitu pula dengan waktu pengerjaannya. Mahasiswa rata-rata mengakhiri pengerjaan *quiz*/ujian di waktu yang sama karena waktu pengerjaan telah ditetapkan. Mahasiswa yang mendapatkan nilai C+ memiliki durasi waktu pengerjaan hampir sama dengan mahasiswa yang mendapatkan nilai A, B, dan B+. Hal ini menunjukkan durasi pengerjaan *quiz*/ujian tidak berpengaruh pada nilai akhir yang didapatkan oleh mahasiswa. Selain itu, hal lain yang dapat dianalisis adalah frekuensi akses per aktivitas. Tabel 4 menunjukkan frekuensi yang terjadi pada tiap aktivitas.

Tabel 4. Tabel frekuensi seluruh aktivitas pada mata kuliah Pemrograman Terstruktur.

Aktivitas	Total Frekuensi
Course	74.543
Assignment	3.776
Quiz	5.460

Berdasarkan Tabel 4, hasil frekuensi paling banyak terjadi pada aktivitas *course* dengan akses 74.543 kali. Aktivitas tersebut menunjukkan aktivitas yang paling sering digunakan oleh rata-rata mahasiswa pada mata kuliah pemrograman terstruktur, diikuti dengan akses *Quiz* dan *Assignment*. Terjadinya perbedaan frekuensi aktivitas pada mata kuliah Pemrograman Terstruktur disebabkan oleh waktu akses yang berbeda-beda. Contohnya seperti mahasiswa satu mengakses *course* ketika mata kuliah sedang berlangsung dan membuka kembali *course* tersebut setelah mata kuliah selesai, namun mahasiswa lainnya hanya mengakses *course* pada saat mata kuliah sedang berlangsung. Hal tersebut dapat mempengaruhi perbedaan frekuensi pada tiap mahasiswa.

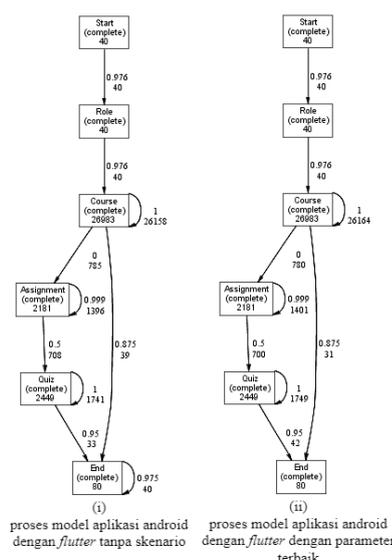
### 3.1.2 Hasil Analisis Mata Kuliah Aplikasi Android dengan Flutter menggunakan ProM dan Disco

#### a) Pengujian Parameter RT, PT, dan DT

Pengujian ini dilakukan dengan mengelompokkan *event log* ke dalam tiga parameter dengan batasan yang berbeda yaitu *Relative to-best Threshold* (RT), *Positive Observation Threshold* (PT), dan *Dependency Threshold* (DT). Pengujian ini dilakukan dengan menggunakan algoritma *heuristic miner* untuk menemukan nilai *fitness* tertinggi. Nilai tersebut diuji dengan mengubah nilai RT, PT, dan DT.

#### b) Pengujian Proses Model dan Conformance Checking

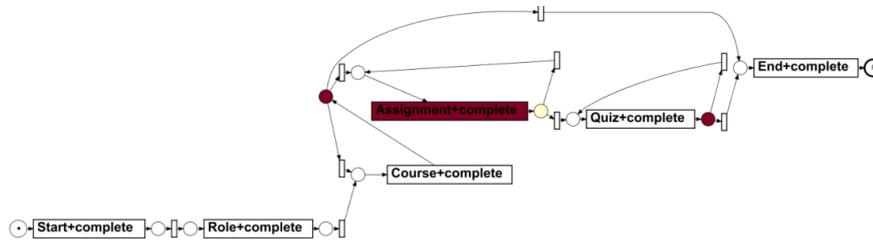
Setelah mendapatkan nilai *fitness* terbaik dari parameter yang ada, didapatkan hasil perbandingan antara penggunaan setelan parameter terbaik dan tanpa penggunaan skenario. Parameter terbaik yang didapatkan yaitu dengan nilai RT 0.05, PT 50, dan DT 0.9. Berdasarkan hasil pengujian parameter, proses model ini merupakan proses model terbaik. Hasil yang didapat adalah tidak adanya perbedaan yang terjadi pada kedua model siklus tersebut. Hal ini menunjukkan bahwa tanpa adanya setelan parameter, nilai seluruh aktivitas telah memenuhi aturan untuk digunakan sebagai proses model. Gambar 6 merupakan hasil perbandingan dari penerapan algoritma *heuristic miner* tanpa skenario dan penerapan algoritma *heuristic miner* menggunakan setelan parameter terbaik.



Gambar 6. Perbandingan proses model mata kuliah Aplikasi Android dengan Flutter dengan nilai parameter terbaik.

Langkah selanjutnya adalah melakukan *conformance checking* untuk menemukan nilai *fitness*,

*precision*, dan *structure*. *Conformance checking* juga dilakukan untuk menemukan *bottleneck* pada *event log*. Gambar 7 menunjukkan proses model yang telah dilakukan *conformance checking*.



Gambar 7. Hasil *conformance checking* pada mata kuliah Aplikasi Android dengan *Flutter*.

Tanda merah dan kekuningan pada bagian *Quiz* and *Assignment* berdasarkan hasil *conformance checking* menunjukkan bahwa kegiatan tersebut terdapat *bottleneck*. Warna ini menunjukkan adanya proses yang terhambat dan akan ditunjukkan dengan menganalisis aktivitas menggunakan aplikasi *Disco*. Hasil dari *conformance checking* yang didapat dari Gambar 3 ditampilkan pada Tabel 5.

Tabel 5. Hasil *conformance checking* pada mata kuliah Aplikasi Android dengan *Flutter*.

Nilai	Hasil
<i>Fitness</i>	0.9882
<i>Advance behavioral appropriateness</i>	0.5902
<i>Degree of model flexibility</i>	0.3095
<i>Structure</i>	1.0000

Tabel 5 menunjukkan nilai *fitness*, *precision*, dan *structure*. Nilai *fitness* menunjukkan angka 0.9882 dan nilai tertinggi untuk parameter *fitness* adalah 1. Hal ini menunjukkan bahwa proses model dapat memodelkan *event log* secara akurat dan hampir sempurna. Nilai *precision* menunjukkan angka 0.5902 pada *advanced behavioral appropriateness*, dan 0.3095 pada *degree of model flexibility*. Hal ini menunjukkan bahwa proses model sangat fleksibel sehingga memungkinkan banyak variasi yang terjadi pada *event log*. Nilai *structure* menunjukkan angka 1. Hal ini menunjukkan bahwa tidak terdapat *duplikasi task* atau *task* yang berulang selama *event log* terjadi.

**c) Analisis menggunakan Aplikasi Disco**

Pada tahap ini dilakukan *cross-check* pada aplikasi *Disco* untuk menunjukkan apakah terdapat *bottleneck* pada aktivitas berdasarkan hasil dari *conformance checking* sebelumnya. Hasil analisis aktivitas pada mata kuliah Aplikasi Android dengan *Flutter* ditampilkan pada Tabel 6. Aktivitas ini diperoleh dengan mengakumulasikan jumlah frekuensi aktivitas per-kategori nilai akhir.

Tabel 6. Hasil analisis aktivitas mahasiswa pada mata kuliah Aplikasi Android dengan *Flutter*.

Nilai	Aktivitas
A	750+ aktivitas
B+	700-750 aktivitas
B	500-700 aktivitas
C	300-500 aktivitas
D	100-300 aktivitas
E	<100 aktivitas

Tabel 6 menunjukkan bahwa semakin banyak frekuensi aktivitas yang dilakukan mahasiswa pada mata kuliah Aplikasi Android dengan *Flutter* di *VClass* membuat nilai yang didapat mahasiswa tersebut semakin tinggi. Begitupun sebaliknya, semakin sedikit frekuensi aktivitas, maka nilai yang

didapat semakin rendah. Selain banyaknya frekuensi aktivitas, frekuensi akses *Quiz* dan *Assignment* juga dapat mempengaruhi nilai akhir mahasiswa. Tabel 7 menunjukkan daftar frekuensi akses *Quiz* dan *Assignment* pada mata kuliah Aplikasi Android dengan *Flutter*.

Tabel 7. Frekuensi akses *quiz* dan *assignment* pada mata kuliah Aplikasi Android dengan *Flutter*.

Nilai	Quiz	Assignment
A	77,4%	71,4%
B+	86,0%	56,7%
B	45,7%	40,0%
C	36,0%	23,5%
D	0,0%	5,0%
E	0,0%	1,3%

Mahasiswa yang mendapatkan nilai A cenderung mengakses lebih banyak aktivitas *Quiz* serta *Assignment*, dan sebaliknya mahasiswa yang mendapat nilai rendah cenderung jarang dan hampir tidak pernah mengakses aktivitas *Quiz* serta *Assignment*. Kesimpulan yang diperoleh berdasarkan Tabel 7 adalah frekuensi akses *Quiz* dan *Assignment* dapat mempengaruhi nilai dari para mahasiswa. Dari hasil tersebut dapat dibuktikan bahwa benar terdapat *bottleneck* atau hambatan pada aktivitas *Quiz* dan *Assignment* yang mempengaruhi variasi nilai akhir para mahasiswa sesuai dengan hasil *conformance checking* pada aplikasi ProM. Durasi bergabung pada suatu kelas juga dapat menjadi salah satu pengaruh dalam mendapatkan nilai. Gambar 8 menunjukkan contoh waktu durasi pada mahasiswa yang mendapatkan nilai rendah.

Case ID	Events	Variant	Started	Finished	Duration
23-B	563	Variant 23	04.08.2021 08:42:00	31.12.2021 19:27:00	149 days, 10 hours
13-B	518	Variant 13	04.08.2021 08:54:00	31.12.2021 19:02:00	149 days, 10 hours
12-B	503	Variant 12	04.08.2021 08:51:00	31.12.2021 19:13:00	149 days, 10 hours
25-C	397	Variant 25	04.08.2021 08:25:00	31.12.2021 19:30:00	149 days, 11 hours
40-C	362	Variant 40	04.08.2021 08:41:00	31.12.2021 19:24:00	149 days, 10 hours
2-D	226	Variant 2	04.08.2021 08:15:00	31.12.2021 19:28:00	149 days, 11 hours
27-D	152	Variant 27	04.08.2021 08:10:00	31.12.2021 19:44:00	149 days, 11 hours
32-D	115	Variant 32	04.08.2021 08:35:00	31.12.2021 19:08:00	149 days, 10 hours
7-E	69	Variant 7	04.08.2021 08:37:00	31.12.2021 19:20:00	149 days, 10 hours
37-E	13	Variant 37	04.08.2021 08:40:00	31.12.2021 19:18:00	149 days, 10 hours
28-E	9	Variant 28	04.08.2021 08:38:00	31.12.2021 19:43:00	149 days, 11 hours

Gambar 8. Jangka waktu mahasiswa bergabung pada mata kuliah Aplikasi Android dengan *Flutter*.

Gambar 8 menunjukkan durasi bergabung seluruh mahasiswa pada mata kuliah Aplikasi Android dengan *Flutter*. Berbeda dengan mata kuliah Pemrograman Terstruktur, durasi gabung pada mata kuliah ini tidak mempengaruhi nilai akhir yang didapat. Durasi pengerjaan *quiz*/ujian dapat dianalisis untuk melihat apakah waktu pengerjaan dapat berpengaruh pada nilai akhir yang didapat. Gambar 9 menunjukkan durasi waktu pengerjaan ujian tengah semester pada mata kuliah Aplikasi Android dengan *Flutter*.

Case ID	Events	Variant	Started	Finished	Duration
18-A	204	Variant 18	04.08.2021 10:02:00	07.12.2021 04:07:00	124 days, 16 hours
29-A	129	Variant 25	04.08.2021 10:09:00	07.12.2021 04:07:00	124 days, 17 hours
31-A	92	Variant 27	04.08.2021 10:15:00	07.12.2021 04:07:00	124 days, 17 hours
38-B+	158	Variant 32	04.08.2021 10:17:00	07.12.2021 04:07:00	124 days, 17 hours
35-A	48	Variant 30	04.08.2021 10:49:00	07.12.2021 04:07:00	124 days, 17 hours
36-A	131	Variant 31	04.08.2021 10:54:00	07.12.2021 04:07:00	124 days, 17 hours
33-B	46	Variant 28	04.08.2021 11:44:00	07.12.2021 04:07:00	124 days, 16 hours
20-A	130	Variant 20	04.08.2021 12:35:00	07.12.2021 04:07:00	124 days, 15 hours
3-A	119	Variant 5	04.08.2021 13:28:00	07.12.2021 04:07:00	124 days, 14 hours
40-C	45	Variant 2	04.08.2021 16:19:00	07.12.2021 04:07:00	124 days, 11 hours
34-A	54	Variant 29	04.08.2021 20:31:00	07.12.2021 04:07:00	124 days, 7 hours

Gambar 9. Durasi waktu pengerjaan UTS Aplikasi Android dengan *Flutter*.

Gambar 9 menunjukkan bahwa durasi waktu pengerjaan *quiz*/ujian yang diakses mahasiswa adalah rata-rata sama. Hal ini disebabkan karena waktu pengerjaan yang telah ditetapkan sebelumnya, sehingga mahasiswa serentak memulai *quiz*/ujian di waktu bersamaan, begitu pula dengan waktu pengerjaannya. Mahasiswa rata-rata mengakhiri pengerjaan *quiz*/ujian di waktu yang sama karena waktu pengerjaan telah ditetapkan. Mahasiswa yang mendapatkan nilai C+ memiliki durasi waktu pengerjaan hampir sama dengan mahasiswa yang mendapatkan nilai A, B, dan B+. Ini berarti durasi pengerjaan *quiz*/ujian tidak berpengaruh pada nilai akhir yang didapatkan oleh mahasiswa. Selain itu,

hal lain yang dapat dianalisis adalah frekuensi akses per aktivitas. Tabel 8 menunjukkan frekuensi yang terjadi pada tiap aktivitas.

Tabel 8. Tabel frekuensi seluruh aktivitas pada mata kuliah Aplikasi Android dengan *Flutter*.

Aktivitas	Total Frekuensi
<i>Course</i>	26.983
<i>Assignment</i>	2.181
<i>Quiz</i>	2.449

Berdasarkan Tabel 4, hasil frekuensi paling banyak terjadi pada aktivitas *course* dengan akses 74.543 kali. Aktivitas tersebut menunjukkan aktivitas yang paling sering digunakan oleh rata-rata mahasiswa pada mata kuliah Aplikasi Android dengan *Flutter*, diikuti dengan akses *Quiz* dan *Assignment*. Terjadinya perbedaan frekuensi aktivitas pada mata kuliah Aplikasi Android dengan *Flutter* disebabkan oleh waktu akses yang berbeda-beda. Contohnya seperti mahasiswa satu mengakses *course* ketika mata kuliah sedang berlangsung dan membuka kembali *course* tersebut setelah mata kuliah selesai, namun mahasiswa lainnya hanya mengakses *course* pada saat mata kuliah sedang berlangsung. Hal tersebut dapat mempengaruhi perbedaan frekuensi pada tiap mahasiswa.

#### 4. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- Proses model terbaik didapatkan dengan menggunakan tiga parameter yakni *Relative-to-best Threshold* (RT), *Positive Observations Threshold* (PT), dan *Dependency Threshold* (DT). Parameter tersebut digunakan untuk mencari nilai *fitness* terbaik. Nilai *fitness* digunakan untuk membuktikan proses model dengan *event log* sesuai atau tidak sesuai. Apabila sesuai, maka *bottleneck* yang telah berhasil ditemukan bernilai valid atau benar terjadi. Pada studi kasus ini, nilai *fitness* yang didapatkan menunjukkan kesesuaian.
- Algoritma *heuristic miner* dapat memodelkan *event log* ke dalam proses model dengan baik. Dengan nilai *fitness* 0.9878 untuk mata kuliah pemrograman terstruktur, dan nilai *fitness* 0.9882 untuk mata kuliah aplikasi android dengan *flutter*. Dengan nilai mendekati angka 1, membuktikan bahwa *bottleneck* yang terjadi pada kedua mata kuliah valid. *Bottleneck* yang terjadi pada kedua mata kuliah adalah kurangnya akses mahasiswa pada aktivitas *Assignment* dan *Quiz*, yang menyebabkan perbedaan nilai yang signifikan.
- Frekuensi aktivitas yang didapatkan di VClass Universitas Lampung bervariasi. Pada kedua mata kuliah, aktivitas yang paling sering diakses adalah pada kategori *course*, kemudian *Quiz*, dan terakhir yakni *Assignment*. Perbedaan intensitas aktivitas yang dilakukan oleh setiap mahasiswa menyebabkan terjadinya perbedaan nilai. Mahasiswa yang melakukan lebih banyak aktivitas cenderung mendapatkan nilai yang lebih tinggi dibandingkan mahasiswa yang melakukan aktivitas rendah.
- Pada mata kuliah Pemrograman Terstruktur, durasi bergabung mempengaruhi nilai akhir yang didapat para mahasiswa, sedangkan pada mata kuliah Aplikasi Android dengan *Flutter*, durasi bergabung tidak berpengaruh pada nilai akhir yang didapatkan mahasiswa.
- Durasi pengerjaan *quiz/ujian* pada kedua mata kuliah tidak mempengaruhi nilai akhir yang didapatkan karena waktu pengerjaan telah ditetapkan durasinya.

#### 5. DAFTAR PUSTAKA

- M. Abdurahman, Sistem Informasi Data Pegawai Berbasis Web Pada Kementerian Kelautan dan Perikanan Kota Ternate, *Jurnal Ilmiah Ilkominfo*, vol. 1, no. 2, 2018, <https://doi.org/10.47324/ilkominfo.v1i2.10>.
- R. Adhim, M. A. Shiddiq, F. Ghizbunaza, & M. A. Yaqin, Process Discovery pada Event Log Permainan Hay Day menggunakan Algoritma Inductive Miner, *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri 2019*, vol. 5, 2019.

- [3] M. Dumas, M. L. Rosa, J. Mendling, & H. A. Reijers, *Fundamentals of Business Process Management*, New York: Springer, 2018.
- [4] S. Huda, Penentuan Attribute Value Untuk Menentukan Bobot Fraud Dalam Transaksi Online, *Proceeding Seminar Nasional Sistem Informasi dan Teknologi Informasi (SENSITEK)*, 2018, <http://dx.doi.org/10.30700/pss.v1i1.310>.
- [5] L. Adhiguna, *Analisis Proses Bisnis Penjualan Dengan Menggunakan Metode Process Mining Pada PT. XYZ Indonesia*, Skripsi, Institut Teknologi Sepuluh Nopember, 2018.
- [6] F. A. Muttaqin, *Implementasi Teks Mining pada Aplikasi Pengawasan Penggunaan Internet Anak 'Dodo Kids Browser'*, Skripsi, Universitas Komputer Indonesia, 2015.
- [7] W. M. P. v. d. Aalst & S. Dustdar, Process mining put into context, *IEEE Internet Computing*, vol. 16, no. 1, pp. 82-86, 2012, <https://doi.org/10.1109/MIC.2012.12>.
- [8] J. Carmona, B. v. Dongen, A. Solti, & M. Weidlich, *Conformance Checking: Relating Processes and Models*, Switzerland: Springer, pp. 1-277, 2018, <https://doi.org/10.1007/978-3-319-99414-7>.