



UNIVERSITAS  
GADJAH MADA

**Pengembangan Prototipe Plugin Learning Analytics Pada Moodle 3.9**

Muhamad Gesit Sumunar, Dr. Sri Suning Kusumawardani, S.T., M.T. ; Ir. Lukito Edi Nugroho, M.Sc., Ph.D.

Universitas Gadjah Mada, 2023 | Diunduh dari <http://etd.repository.ugm.ac.id/>

# **PENGEMBANGAN PROTOTIPE PLUGIN LEARNING ANALYTICS PADA MOODLE 3.9**

**DOKUMEN C-501**



**Disusun oleh:**

**Muhamad Gesit Sumunar**

**18/429074/TK/47576**

**DOKUMENTASI SKRIPSI *CAPSTONE PROJECT*  
PROGRAM STUDI TEKNOLOGI INFORMASI  
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA  
2023**



## HALAMAN PENGESAHAN

1. USULAN JUDUL SKRIPSI	: Pengembangan Prototipe Plugin Learning Analytics Pada Moodle 3.9 : <i>Prototype Plugin Learning Analytics Development on Moodle 3.9</i>
2. JENIS DOKUMEN	: IMPLEMENTASI SOLUSI
3. KODE DOKUMEN	: C-501
4. NOMOR DOKUMEN	: C-501-SSKLEN_21_A13CF
5. NOMOR REVISI	: 00
6. TANGGAL PENERBITAN	: 15 Februari 2023
7. DATA MAHASISWA	Tanda Tangan :
a. Nama lengkap	: M Gesit Sumunar
b. NIM	: 18/429074/TK/47576
c. Prodi	: Teknologi Informasi
d. Email	: gesitsumunar@mail.ugm.ac.id
8. DOSEN PEMBIMBING I	Tanda Tangan :
a. Nama lengkap	: Dr. Sri Suning Kusumawardani, S.T., M.T.
b. NIP	: 196911221995122001
9. DOSEN PEMBIMBING II	Tanda Tangan :
a. Nama lengkap	: Ir. Lukito Edi Nugroho, M.Sc., Ph.D.
b. NIP	: 96603271991031002
10. TEMPAT PELAKSANAAN	: Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik
11. JUMLAH HALAMAN	: 56



## HALAMAN PENGESAHAN

### PENGEMBANGAN PROTOTIPE PLUGIN LEARNING ANALYTICS PADA MOODLE 3.9

#### SKRIPSI (CAPSTONE DESIGN PROJECT)

Diajukan sebagai Salah Satu Syarat untuk Memperoleh

Gelar Sarjana Teknik

pada Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik

Universitas Gadjah Mada

Disusun oleh :

**Muhamad Gesit Sumunar**

**18/429074/TK/47576**

**Zahra Rizki Annisa**

**18/431412/TK/48005**

**Leon Davin**

**18/429072/TK/47574**

Telah disetujui dan disahkan

pada tanggal, 01 Maret 2023

Dosen Pembimbing I

Dr. Sri Suning Kusumawardani, S.T., M.T.  
NIP. 196911221995122001

Dosen Pembimbing II

Prof. Ir. Lukito Edi Nugroho, M.Sc., Ph.D.  
NIP. 196603271991031002





**SKRIPSI**

*(Capstone Design Project)*

**PENGEMBANGAN PROTOTIPE PLUGIN LEARNING ANALYTICS PADA MOODLE  
3.9**

Dipersiapkan dan disusun oleh

**Muhamad Gesit Sumunar - 18/429074/TK/47576**

**Zahra Rizki Annisa - 18/431412/TK/48005**

**Leon Davin - 18/429072/TK/47574**

Telah dipertahankan di depan dewan penguji  
pada tanggal : 01 Maret 2023

**Susunan Dewan Penguji**

**Pembimbing Utama**

**Pembimbing Pendamping**

  
**Dr. Sri Suning Kusumawardani, S.T., M.T.**

  
**Prof. Ir. Lukito Edi Nugroho, M.Sc., Ph.D.**

**Anggota Dewan Penguji Lain**

  
**Dr. Bimo Sunarfri Hantono, S.T., M.Eng.**

  
**Ir. Agus Bejo, S.T., M.Eng., D.Eng., IPM.**

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Teknik


Tanggal: 09 Mei 2023

Pengelola Program Studi: Sarjana Teknologi Informasi

  
**Ir. Agus Bejo, S.T., M.Eng., D.Eng., IPM.**  
**NIP 198001012015041002**

Mengetahui,  
**Ketua Departemen Teknik Elektro dan Teknologi Informasi**



  
**Ir. Hanung Adi Nugroho, S.T., M.E., Ph.D., IPM.**  
**NIP 197802242002121001**



## BUKTI BEBAS PLAGIASI

### PENGEMBANGAN PROTOTYPE PLUGIN LEARNING ANALYTICS PADA MOODLE 3.9

#### ORIGINALITY REPORT

**13%**

SIMILARITY INDEX

**11%**

INTERNET SOURCES

**2%**

PUBLICATIONS

**5%**

STUDENT PAPERS

#### PRIMARY SOURCES

<b>1</b>	Submitted to Clarkston Community Schools Student Paper	<b>1%</b>
<b>2</b>	<a href="http://it.rsudsekayu.mubakab.go.id">it.rsudsekayu.mubakab.go.id</a> Internet Source	<b>1%</b>
<b>3</b>	<a href="http://informatika.untag-sby.ac.id">informatika.untag-sby.ac.id</a> Internet Source	<b>1%</b>
<b>4</b>	Submitted to SDM Universitas Gadjah Mada Student Paper	<b>1%</b>
<b>5</b>	<a href="http://wakool.id">wakool.id</a> Internet Source	<b>1%</b>
<b>6</b>	<a href="http://anyflip.com">anyflip.com</a> Internet Source	<b>1%</b>
<b>7</b>	<a href="http://publikasiilmiah.ums.ac.id">publikasiilmiah.ums.ac.id</a> Internet Source	<b>1%</b>
<b>8</b>	<a href="http://repository.ub.ac.id">repository.ub.ac.id</a> Internet Source	<b>&lt;1%</b>
<b>9</b>	<a href="http://repository.dinamika.ac.id">repository.dinamika.ac.id</a> Internet Source	<b>&lt;1%</b>



## **PERNYATAAN BEBAS PLAGIASI**

Saya yang bertanda tangan di bawah ini:

Nama : M Gesit Sumunar  
NIM : 18/429074/TK/47576  
Tahun terdaftar : 2018  
Program Studi : Teknologi Informasi  
Fakultas : Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 21 Juli 2023



10000  
METERAI  
TEMPEL  
B22DOAKX256329924

M Gesit Sumunar  
18429074/TK/47576

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
BUKTI BEBAS PLAGIASI.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	x
CATATAN REVISI DOKUMEN .....	xi
INTISARI.....	xii
RINGKASAN EKSEKUTIF.....	xiii
BAB 1    PENGANTAR .....	1
1.1    Rumusan Masalah.....	1
1.2    Tujuan.....	2
1.3    Manfaat.....	2
1.4    Struktur Dokumen .....	2
BAB 2    DASAR TEORI PENDUKUNG .....	3
1.1    LMS.....	3
1.1.1    Moodle .....	3
1.1.2    Schoology .....	4
1.1.3    Blackboard Learn.....	4
2.1    Arsitektur Plug-in.....	5
2.1.1    Core System.....	6
2.1.2    Plug-in .....	6
2.1.3    Hubungan Core System dan Plug-in .....	7
2.2    Bahasa Pemrograman .....	8
2.2.1    PHP.....	9
2.3.1.1.    Laravel .....	10
2.3.1.2.    CodeIgniter .....	11
2.2.2    Python.....	11
2.3.2.1.    Django .....	12
2.3.2.2.    Flask.....	12
2.3    Software Development Life Cycle .....	13
2.4    Learning Analytics .....	14
BAB 3    ANALISIS STUDI PUSTAKA KUNCI DAN PEMILIHAN METODE.....	17



3.1	Analisis Jenis LMS.....	17
3.1.1	Moodle.....	17
3.1.2	Blackboard Learn LMS .....	17
3.1.3	Schoology .....	17
3.1.4	Perbandingan Moodle, Blackboard Learn, dan Schoology .....	18
3.2	Pemilihan LMS .....	18
BAB 4	DETAIL IMPLEMENTASI .....	19
4.1	Luaran <i>Capstone Project</i> Beserta Spesifikasinya .....	19
4.2	Batasan Masalah.....	20
4.3	Detail Rancangan .....	20
4.3.1	Learning Analytics Component.....	21
4.3.2	Data Flow Diagram .....	23
4.3.3	Integrasi Plug-in Dengan Moodle.....	24
BAB 5	PENGUJIAN DAN PEMBAHASAN .....	27
5.1	Pengujian dan Pembahasan .....	27
5.1.1	Skenario Pengujian 1 dan Analisis .....	27
5.1.2	Skenario Pengujian 2 dan Analisis .....	32
5.2	<i>Improvement</i> .....	36
BAB 6	ANALISIS MENGENAI PENGARUH SOLUSI <i>ENGINEERING DESIGN</i> .....	38
6.1	Konteks Global.....	38
6.2	Konteks Ekonomi.....	38
6.3	Konteks Lingkungan .....	39
6.4	Konteks Sosial.....	39
BAB 7	KESIMPULAN DAN SARAN .....	40
7.1	Kesimpulan.....	40
7.2	Saran.....	40
REFERENSI.....		41



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Plug-in</i> [5] .....	5
Gambar 2.2 Software Development Life Cycle [10] .....	13
Gambar 4.1 <i>Learning analytics component</i> .....	21
Gambar 4.2 <i>Data flow diagram</i> .....	23
Gambar 5.1 Halaman utama <i>plug-in</i> .....	27
Gambar 5.2 Partisipan <i>course</i> forensik digital .....	28
Gambar 5.3 <i>Grading</i> Forensik Digital pada tanggal 9 Februari 2023 .....	28
Gambar 5.4 Evaluasi model data pertama.....	29
Gambar 5.5 Hasil prediksi <i>course</i> Forensik Digital .....	29
Gambar 5.6 Prediksi Lark Marcus secara detail .....	30
Gambar 5.7 Prediksi Master Chief secara detail .....	30
Gambar 5.8 Prediksi Arthur Leywin secara detail .....	31
Gambar 5.9 Contoh komunikasi pada peserta didik .....	31
Gambar 5.10 Partisipan <i>course</i> Testing 2 .....	32
Gambar 5.11 Bobot penilaian <i>course</i> Testing 2.....	33
Gambar 5.12 <i>Grading</i> peserta didik <i>course</i> Testing 2 .....	33
Gambar 5.13 Evaluasi model data kedua .....	34
Gambar 5.14 Hasil prediksi peserta didik <i>course</i> Testing 2 .....	34
Gambar 5.15 Detail prediksi Venom Snake pada <i>course</i> Testing 2.....	35
Gambar 5.16 Detail prediksi Geralt Of Rivia pada <i>course</i> Testing 2 .....	35
Gambar 5.17 Detail prediksi Cloud Strife pada <i>course</i> Testing 2.....	36



## DAFTAR TABEL

Tabel 3.1 Perbandingan Moodle, Blackboard Learn, dan Schoology .....	18
Tabel 3.2 Luaran Proyek <i>Capstone</i> .....	19
Tabel 3.3 Spesifikasi Luaran Proyek .....	19



UNIVERSITAS  
GADJAH MADA

## Pengembangan Prototipe Plugin Learning Analytics Pada Moodle 3.9

Muhamad Gesit Sumunar, Dr. Sri Suning Kusumawardani, S.T., M.T. ; Ir. Lukito Edi Nugroho, M.Sc., Ph.D.

Universitas Gadjah Mada, 2023 | Diunduh dari <http://etd.repository.ugm.ac.id/>

### CATATAN REVISI DOKUMEN

VERSI	TANGGAL	PERBAIKAN
V.0		
V.1.1		
V.2.1		
V.x.y		
V.x.y		
V.x.y		
V.x.y		

## INTISARI

Munculnya revolusi industri 4.0 membuat beberapa kegiatan yang sebelumnya dilakukan secara konvensional mulai bertransformasi menjadi dalam bentuk digital. Salah satu yang terkena imbas adalah kegiatan belajar mengajar. Kegiatan belajar mengajar secara *online* sekarang sudah banyak ditemui pada platform *Learning Management System* (LMS). Kemunculan LMS beserta turunannya sangatlah membantu kegiatan belajar mengajar, terutama di masa pandemi Covid-19 yang mengharuskan kita untuk tetap di rumah dan mengurangi mobilitas. Meskipun demikian, tanpa pertemuan tatap muka antara pengajar dan pengajar membuat penilaian kualitatif peserta didik sulit dilakukan.

Proyek *capstone* ini memiliki tema untuk membantu tenaga pengajar untuk membuat penilaian kualitatif kepada peserta didik dengan cara melihat bagaimana *progress* belajar peserta didik di dalam suatu *course*. Tim *capstone* mengembangkan sebuah mesin pintar dengan menggunakan teknik *machine learning* dan *deep learning*. Dari teknik tersebut, terciptalah sebuah mesin pintar yang dapat digunakan untuk tujuan tersebut. Cara kerja dari mesin tersebut adalah dengan mengambil *data* yang dimiliki oleh *course* yang sudah selesai kemudian menggunakan algoritma tertentu untuk mendapatkan keluaran yang diinginkan.

Untuk penerapannya, mesin tersebut diintegrasikan ke dalam *plug-in* yang penulis kembangkan dengan menggunakan bahasa PHP dengan *framework* yang sudah tersedia pada Moodle. Kemudian *plug-in* tersebut dipasang pada Moodle 3.9.

## RINGKASAN EKSEKUTIF

Seiring bertambahnya waktu serta keadaan pandemi global Covid-19 yang baru saja terjadi, memaksa pertumbuhan teknologi yang semakin berkembang pesat. Dengan berkembangnya teknologi, kegiatan-kegiatan yang sebelumnya hanya dapat dilakukan dengan tatap muka, sekarang dapat dilakukan dengan bentuk digital, tak terkecuali dengan kegiatan pembelajaran. Saat ini, kegiatan pembelajaran *online* sudah semakin banyak menjamur dan banyak digunakan terutama ketika pandemi Covid-19 yang membuat masyarakat membatasi ruang lingkungannya.

LMS atau (*Learning Management System*) merupakan sistem pembelajaran *online* yang sering digunakan, tak terkecuali oleh Universitas Gadjah Mada. LMS yang dimiliki oleh UGM bernama eLOK dan memiliki basis Moodle. Dengan adanya eLOK, UGM berkontribusi menjadi bagian dari usaha mencerdaskan kehidupan bangsa dengan memanfaatkan teknologi. Kegiatan pembelajaran di eLOK dibuat semirip mungkin dengan pembelajaran tatap muka. Akan tetapi, terdapat keterbatasan dalam eLOK ketika dibandingkan dengan pembelajaran tatap muka, yakni tenaga pengajar yang mengalami kesulitan dalam menilai peserta didik secara kualitatif karena interaksi antara peserta didik dan tenaga pengajar yang terbatas jika dibandingkan dengan kegiatan pembelajaran konvensional.

Dalam usaha yang dilakukan untuk mengatasi keterbatasan di atas, penulis mengangkat tema *capstone* ini. Penulis mengembangkan sebuah *plug-in learning analytics* yang di dalamnya terdapat sebuah mesin pintar yang dapat melakukan prediksi hasil akhir yang akan dicapai oleh peserta didik. Indikator yang digunakan oleh *plug-in* tersebut dalam membuat prediksi tidak hanya terdiri dari nilai yang didapatkan ketika tugas dan ujian, namun juga keaktifan peserta didik dalam LMS juga diperhitungkan untuk membuat prediksi akhir. Oleh karena itu, tenaga pengajar dapat dengan mudah melakukan penilaian kualitatif terhadap peserta didik.

*Plug-in* yang penulis kembangkan menggunakan model data dari *course* yang sudah selesai sebelumnya. Semakin banyak *course* yang sudah selesai pada sebuah LMS, maka dataset yang tersedia semakin banyak sehingga menyebabkan tingkat akurasi mesin pintar di dalam *plug-in* menjadi lebih tinggi. Untuk proses pengujian, *plug-in* tersebut penulis *install* pada Moodle 3.9 yang merupakan basis dari eLOK. Penulis melakukan pengujian pada dua buah *course* dalam Moodle. Dikarenakan pengujian pada *course* kedua dilakukan setelah *course* pertama sudah selesai sehingga dataset yang dimiliki pada pengujian kedua menjadi lebih banyak dibandingkan dengan dataset pada pengujian pertama dan tingkat akurasi pada pengujian kedua menjadi lebih tinggi.



Apabila proyek *capstone* ini dapat diimplementasikan secara luas, maka terdapat beberapa dampak positif dalam yang didapat dalam pembelajaran *online* dikarenakan tenaga pengajar dapat melakukan penilaian kualitatif pada peserta didik. Dampak pertama yang terjadi adalah meningkatnya kualitas pembelajaran online sehingga pembelajaran online untuk ke depannya semakin banyak digunakan. Kemudian, dikarenakan semakin banyak pembelajaran *online*, maka kebutuhan gedung yang digunakan untuk ruang kelas dapat berkurang dan gedung yang sebelumnya digunakan untuk ruang kelas dapat dialokasikan untuk kegiatan yang lain seperti laboratorium penelitian. Pembelajaran *online* dapat dilakukan dimanapun tanpa terikat tempat, asalkan terdapat fasilitas internet. Oleh karena itu, peserta didik dan tenaga pengajar tidak perlu datang ke tempat tertentu untuk melakukan kegiatan pembelajaran sehingga dapat memangkas biaya transportasi dan polusi yang ditimbulkan dari transportasi. Selain itu, *plug-in* dari proyek ini juga dapat menambah interaksi antara tenaga pengajar dengan peserta didik karena terdapat *shortcut* yang dapat digunakan tenaga pengajar untuk menghubungi peserta didik, terutama kepada peserta didik yang dianggap oleh *plug-in* memiliki risiko gagal dalam suatu course.



## BAB 1

### PENGANTAR

eLOK adalah sebuah LMS (*Learning Management System*) yang digunakan untuk implementasi MOOC (*Massive Online Open Course*) dari Universitas Gadjah Mada, yang berarti eLOK memberikan konten pembelajaran secara *online* dan *virtual* kepada siapapun, bahkan kepada masyarakat umum. eLOK berbasis pada Moodle, lebih tepatnya Moodle versi 3.9. Seperti kebanyakan LMS, eLOK dapat membantu baik mahasiswa maupun pengajar dalam proses belajar mengajar.

Saat ini penggunaan eLOK beserta LMS yang lain telah digunakan secara masif, terutama saat pandemi Covid-19 dimana proses kegiatan belajar yang dijalankan oleh institusi pendidikan harus dilaksanakan secara *online* dan *virtual*. Sekalipun pandemi Covid-19 telah usai, penggunaan LMS diprediksikan tidak akan mengalami penurunan yang signifikan dan transformasi pembelajaran akan terus berlanjut. Hal tersebut terjadi agar penyelenggaraan pendidikan dapat relevan dengan revolusi industri 4.0.

Meskipun sudah memiliki banyak *user*, LMS juga memiliki beberapa kekurangan untuk menunjang industri 4.0. Salah satunya adalah kurangnya sarana dan cara untuk memunculkan hal-hal yang terjadi seperti bagaimana menunjukkan performa belajar, bagaimana menunjukkan tingkat motivasi mahasiswa, bahkan bagaimana sistem dapat menginformasikan prediksi terhadap permasalahan dan keberhasilan mahasiswa. Sarana ini bisa difasilitasi dalam *Learning Analytics*.

*Learning Analytics* merupakan salah satu metode yang dapat digunakan untuk mengolah data yang diperoleh dalam pembelajaran. *Learning analytics* ini memiliki hubungan yang erat kaitannya dengan statistika, data mining, dan sistem informasi. *Learning analytics* menggunakan data-data yang diperoleh pada saat peserta didik menggunakan sistem *e-learning*. Selanjutnya, data yang sudah diperoleh kemudian dianalisis untuk tujuan pengembangan pembelajaran itu sendiri dalam kerangka menciptakan lingkungan pembelajaran berbasis keunikan individu (*personalised learning*).

#### 1.1 Rumusan Masalah

eLOK merupakan sebuah LMS yang memiliki fitur yang terbilang cukup banyak untuk sebuah LMS. Akan tetapi, eLOK tidak memiliki sistem yang dapat memprediksi bagaimana hasil atau nilai yang akan dicapai dalam sebuah mata kuliah di akhir berdasarkan keaktifan mahasiswa



di mata kuliah tersebut. Padahal fitur tersebut sangatlah penting, terutama untuk melacak bagaimana *progress* peserta didik dalam suatu mata kuliah.

## 1.2 Tujuan

Tujuan yang hendak dicapai dengan dilakukannya proyek *capstone* ini adalah untuk mengetahui seberapa jauh *progress* mahasiswa dalam mengikuti sebuah mata kuliah secara online. Hal tersebut dilakukan dengan cara membuat sistem prediksi performa pembelajaran sehingga peserta didik dapat mengetahui bagaimana perkiraan nilai yang akan didapatkan di akhir semester.

## 1.3 Manfaat

*Plug-in* yang penulis kembangkan dapat memberikan manfaat kepada kedua belah pihak pada suatu *course* di dalam LMS, yakni untuk tenaga pengajar dan peserta didik. Untuk tenaga pengajar, *plug-in* dapat digunakan untuk memantau bagaimana perkembangan peserta didik dalam suatu mata kuliah sehingga dapat membantu mengembangkan pembelajaran di masa depan. Sedangkan untuk peserta didik, *plug-in* dapat digunakan untuk memprediksi kemampuan peserta didik.

## 1.4 Struktur Dokumen

Dokumen terdiri dari 7 bab, yaitu :

1. Pengantar : berisi penjelasan umum mengenai topik yang menjadi bahasan pada dokumen C251.
2. Dasar Teori Pendukung : berisi dasar teori pendukung dalam proses pengerjaan proyek *capstone* ini.
3. Analisis Studi Pustaka : berisi pemaparan isi dari referensi yang dijadikan sebagai rujukan utama dalam menentukan metode pengembangan proyek.
4. Detail Implementasi : berisi permodelan yang digunakan untuk menyelesaikan permasalahan yang menjadi dasar pengerjaan proyek.
5. Pengujian dan Pembahasan : berisi pengujian yang dilakukan pada proyek *capstone* ini.
6. Analisis Mengenai Pengaruh Solusi *Engineering Design* : berisi tentang dampak yang didapatkan apabila proyek ini diimplementasikan secara luas.
7. Kesimpulan dan Saran : berisi tentang kesimpulan dan saran untuk ke depannya.

## BAB 2

### DASAR TEORI PENDUKUNG

#### 1.1 LMS

LMS atau *Learning Management System* adalah sebuah *software* atau teknologi berbasis web yang digunakan untuk merencanakan, mengimplementasikan, dan menilai sebuah proses belajar mengajar. Sehingga bisa dikatakan bahwa proses belajar mengajar yang menggunakan LMS disebut sebagai eLearning. LMS secara umum terdiri dari dua elemen, yakni sebuah server yang menjalankan fungsi basis dan user interface yang dapat dioperasikan oleh *instructor* atau tenaga pengajar, *student* atau peserta didik, dan administrator.

Secara umum, LMS menyediakan role *instructor* agar dapat membuat sebuah materi kemudian membagikannya, memantau partisipasi *student*, dan menilai kinerja *student*. Sedangkan untuk role *student*, LMS memungkinkan untuk menggunakan fitur interaktif seperti *threaded discussions*, *video conference*, maupun forum diskusi.

LMS sering digunakan oleh segala ukuran bisnis, mulai dari ukuran bisnis yang paling kecil maupun sampai yang berukuran raksasa. Selain itu, LMS juga sering digunakan oleh lembaga pemerintah nasional dan daerah dan juga oleh lembaga pendidikan, baik yang konvensional maupun yang sudah berbasis eLearning. Hal tersebut karena LMS dapat meningkatkan metode pendidikan terutama yang masih tradisional, sekaligus menghemat waktu dan uang organisasi. Sistem yang efektif akan memungkinkan *instructor* dan administrator mengelola elemen secara efisien seperti pendaftaran pengguna, konten, kalender, akses pengguna, komunikasi, sertifikasi, dan pemberitahuan.

##### 1.1.1 Moodle

Moodle (*Modular Object-Oriented Dynamic Learning Environment*) adalah sebuah platform pembelajaran yang dirancang untuk memberikan lingkungan belajar yang dipersonalisasi bagi para pendidik, administrator, dan pelajar dengan satu sistem yang kuat, aman, dan terintegrasi. Per 6 Juni 2020, Moodle telah memberikan basis kepada ratusan ribu lingkungan pembelajaran secara global dengan *user* lebih dari 213 juta [2]. Salah satu sistem yang berbasis Moodle adalah eLOK milik Universitas Gadjah Mada.

Moodle menyediakan *tool-set* yang fleksibel untuk mendukung pembelajaran secara *blended* maupun 100% secara *online*. Moodle dapat dikonfigurasi untuk mengaktifkan atau menonaktifkan



fitur ini dan dapat dengan mudah mengintegrasikan semua yang diperlukan untuk pembelajaran *online* dengan menggunakan rangkaian fitur bawaan.

Karena bersifat *Open Source*, Moodle dapat dikustomisasikan sesuai dengan kebutuhan individu. Pengaturan modular dan desain yang dapat dioperasikan memungkinkan pengembang untuk membuat *plug-in* dan mengintegrasikan dengan aplikasi eksternal untuk mencapai fungsionalitas tertentu. Moodle dapat diperluas cakupannya dengan menggunakan *plug-in* dan *add-on* yang sudah disediakan dengan kemungkinan yang tak terbatas.

Moodle berbasis *web* sehingga dapat diakses dimanapun. Dengan *default mobile-compatible interface* dan kompatibilitas lintas-browser, konten dari Moodle dapat dengan mudah diakses dan tetap konsisten di berbagai *web browser* dan perangkat.

### **1.1.2 Schoology**

Schoology adalah sebuah sistem manajemen *online course* yang memungkinkan tenaga pengajar untuk menciptakan sekaligus mengelola kursus akademik untuk peserta didik yang dimiliki. Schoology memberi tenaga pengajar metode untuk mengelola pelajaran, melibatkan siswa, dan berbagi konten. Schoology menyediakan cara yang aman dan terjamin, mudah digunakan, bagi guru, siswa, orang tua, dan administrator untuk berkomunikasi dan berkolaborasi secara lancar guna meningkatkan pembelajaran dan kesuksesan siswa.

### **1.1.3 Blackboard Learn**

Blackboard Learn adalah sebuah LMS dan lingkungan pembelajaran *virtual* berbasis web yang dikembangkan oleh Blackboard Inc. LMS ini memiliki fitur *course management*, arsitektur terbuka yang dapat dikustomisasi, dan desain yang *scalable* yang memungkinkan Blackboard Learn untuk berintegrasi dengan sistem informasi peserta didik dan protokol autentikasi. Blackboard Learn dapat diinstal pada server lokal, maupun dihosting oleh Blackboard ASP Solutions, atau Amazon Web Services.

Blackboard Learn dapat disesuaikan baik sebagai LMS yang menggantikan kelas tatap muka maupun melengkapi kelas tersebut untuk kebutuhan pembelajaran. Banyak fungsi Blackboard Learn yang sudah disamakan dengan fungsi kelas konvensional yang masih tatap muka karena Blackboard Learn memungkinkan siswa dan guru untuk meniru hampir setiap elemen pengalaman ruang kelas fisik.

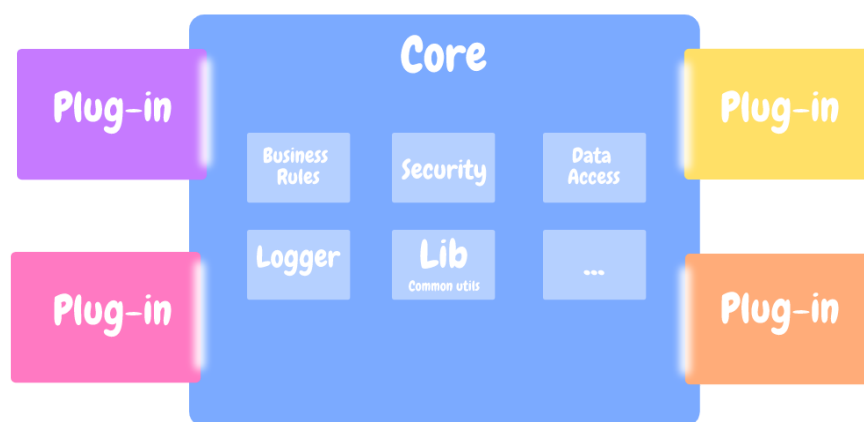
Apabila digunakan untuk pembelajaran yang sepenuhnya online, guru dapat menugaskan semua materi kepada siswa secara digital, berkomunikasi di luar kelas menggunakan alat



komunikasi yang disediakan, dan meminta siswa berdiskusi dan berkolaborasi secara online. Namun dalam model hybrid atau gabungan online dan offline, kelas mungkin bertemu langsung hanya beberapa kali dalam seminggu dan menggunakan aktivitas “Blackboard” yang dapat diselesaikan siswa di luar waktu pembelajaran. Oleh karena itu, memungkinkan untuk berlangsung secara tatap muka, tetapi guru dapat memposting materi pendukung seperti silabus, tugas, atau diskusi opsional secara online.

## 2.1 Arsitektur Plug-in

Arsitektur *Plug-in* terdiri dari dua komponen, yaitu *core system* atau sistem inti dan *plug-in modules*. Arsitektur ini memungkinkan untuk penambahan, pengurangan, maupun pengeditan fitur atau *plug-in* ke aplikasi inti yang sudah ada dengan sedikit atau bahkan tanpa efek kepada aplikasi inti maupun modul *plug-in* yang lain. Kelebihan lain dari arsitektur ini adalah dengan memberikan keleluasan terhadap *plug-in* yang akan ditambahkan, fleksibilitas yang tinggi, dan isolasi antara fitur aplikasi dan logika pemrosesan aplikasi [5].



Gambar 2.1 Arsitektur *Plug-in* [5]

Dengan adanya kedua komponen tersebut, *plug-in* memiliki beberapa kelebihan[6], antara lain:

- membuat cakupan fungsi dari *website* menjadi lebih luas dan dapat dikustomisasi sesuai kebutuhan *user*,



- meminimalkan dampak dari *upgrade*,
- menawarkan transparansi kode,
- mempermudah dalam penanganan kustomisasi.

### 2.1.1 Core System

Pada tingkat yang tinggi, *core system* mendefinisikan bagaimana sistem bekerja dan dasar dari *business logic* yang dimiliki. Pada tingkat ini, *core system* memiliki bentuk abstrak, tidak memiliki implementasi khusus, dan tidak memungkinkan kustomisasi. Sebagai contoh *core system* pada level ini adalah alur kerja umum yang mendefinisikan bagaimana aliran data dalam aplikasi. Akan tetapi, langkah-langkah dalam alur kerja tersebut ditentukan oleh *plug-in* dan semua ekstensi dari *plug-in* akan mengikuti alur kerja umum dari implementasi yang sudah disesuaikan.

Pada tingkat yang lebih rendah, *core system* menangani kasus khusus, menerapkan aturan khusus, dan memproses apabila keadaan menjadi kompleks. Selain itu, pada tingkat ini, *core system* juga berisi kode umum yang digunakan (atau harus digunakan) oleh beberapa *plug-in* sebagai cara untuk menghilangkan duplikasi dan *boilerplate* kode. Contohnya apabila dua plugin mencatat transaksi dan kemudian terjadi kegagalan, *core system* harus menyediakan pencatat fitur sebagai bagian darinya. Belum lagi hal-hal seperti keamanan, versi, komponen UI, akses *database*, caching, dll

### 2.1.2 Plug-in

*Plug-in* atau *plugin* adalah sebuah program komputer yang menambah fitur pada sebuah program tanpa melakukan perubahan pada program tersebut [3]. Program utama biasanya memberikan semacam antarmuka agar *plug-in* dapat berinteraksi dengan program utama. Sebagai contoh, *plug-in* biasanya digunakan pada web browser untuk memutar video, memindai virus, dan menampilkan jenis *file* baru. Contoh *plug-in* yang dikenal adalah *plug-in* untuk Adobe Flash Player dan Quick Time.

Dengan *plug-in*, *user* mendapatkan solusi terbaik ketika menambahkan fitur pada sebuah *website* dikarenakan *user* tak perlu mengakses *source code* utama dari *website* tersebut ketika menambahkan fitur. *User* hanya perlu mengunduh dan memasang *plugin* yang tersedia.



Menggunakan *plug-in* berarti *user*, atau dalam topik ini adalah mahasiswa peserta mata kuliah memasukkan “*plug*” kode tambahan untuk memodifikasi kode original. Apabila suatu saat perubahan yang dilakukan tidak lagi digunakan, maka *user* hanya perlu “*unplug*” kode dengan cara menonaktifkan *plug-in*.

Salah satu jenis *plug-in* yang dapat dikembangkan adalah *plug-in machine learning*. *Plug-in* ini memiliki *machine learning backend* yang dapat memproses *dataset* yang dihasilkan dari *indicator* dan *target* yang telah dihitung oleh *Analytics API*. *Target* merupakan hal yang akan kita prediksi sedangkan *indicator* adalah beberapa hal yang menjadi dasar dalam memprediksi. *Analytics API* digunakan untuk pelatihan *Machine Learning*, memprediksi kemungkinan yang akan terjadi, dan mengevaluasi model yang telah dibuat.

Komunikasi antara *Machine Learning Backend* dan Moodle adalah dengan melalui *file* karena kode yang akan memproses *dataset* dapat ditulis dengan menggunakan bahasa PHP, Python, atau dalam bahasa lain, dan dapat juga dituliskan dengan menggunakan layanan *cloud* [4]. Komunikasi tersebut harus terukur agar dapat mengatur *file* dengan ukuran yang besar atau bahkan dapat melatih algoritma dalam membaca kumpulan *file input*.

*Plug-in* secara umum berupa komponen yang independen dan berisi *process* yang spesifik, tambahan fitur, dan kustomisasi kode yang berupa peningkatan atau perluasan dari *core system* sebagai upaya untuk meningkatkan kapabilitas *core system*. Meskipun demikian, beberapa *plug-in* membutuhkan komunikasi dengan *plug-in* lain agar dapat bekerja dengan maksimal. Sekalipun demikian, komunikasi dan juga dependensi antar-*plug-in* harus berada diminimalkan sebisa mungkin.

### 2.1.3 Hubungan Core System dan Plug-in

*Core system* harus mengetahui ekstensi dari *plug-in module* sekaligus cara untuk mendapatkannya. Kemudian *core system* akan mendeklarasikan *extension points* agar *plug-in* dapat terhubung ke *core system*. *Extension point* ini dapat merepresentasikan *life cycle* dari *core system*. Bagaimana cara *core system* terhubung dengan semua *plug-in* ditentukan dengan tipe pembangunan aplikasi-nya (*small product* atau *large business application*) beserta kebutuhan spesifiknya (*single deploy* atau *distributed deploy*).

Setiap *plug-in* akan mendaftarkan dirinya sendiri ke *core system*, saling bertukar informasi seperti nama, protokol komunikasi, input/output *data handler*, *data format*, dan mengaitkan dirinya sendiri ke *extension point* yang sudah dideklarasikan oleh *core system*.



## **2.2 Bahasa Pemrograman**

Bahasa pemrograman merupakan sekumpulan instruksi yang diberikan kepada komputer untuk dapat melaksanakan tugas-tugas tertentu dengan tujuan untuk menyelesaikan suatu permasalahan. Bahasa pemrograman berfungsi untuk memberikan perintah kepada komputer agar dapat mengolah data sesuai dengan langkah-langkah penyelesaian yang telah ditentukan oleh programmer. Bahasa ini memungkinkan seorang programmer untuk menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa yang akan diambil dalam berbagai situasi secara persis. Perkembangan bahasa pemrograman ini tentu menciptakan berbagai inovasi teknologi yang dapat membantu masyarakat dalam kehidupan sehari-hari. Contohnya program lampu lalu lintas di jalan, mesin kasir, aplikasi game, anti-virus, *web*, media sosial, dan lain sebagainya.

Bahasa pemrograman memiliki 3 tingkatan yang dikelompokkan berdasarkan seberapa mirip bahasa tersebut dengan bahasa manusia. 3 Tingkatan tersebut antara lain :

- Bahasa tingkat rendah

Bahasa ini merupakan bahasa yang paling tidak mirip dengan bahasa manusia sekaligus bahasa yang digunakan oleh komputer. Bahasa program tingkat rendah berisi instruksi-instruksi yang ditujukan kepada komputer dengan menggunakan kode biner (0 dan 1). Kode-kode tersebut akan langsung diterjemahkan oleh komputer tanpa harus melalui proses kompilasi. Oleh karena itu, bahasa pada tingkat ini memerlukan memori yang kecil serta lebih cepat untuk proses eksekusi.

- Bahasa tingkat menengah

Bahasa program tingkat menengah terdiri dari instruksi-instruksi yang berupa kode mnemonic seperti ADD, SUB, DIV, STO, LOD, JMP dan lainnya. Namun, bahasa dalam tingkat menengah ini harus diterjemahkan terlebih dahulu ke dalam bahasa mesin, karena komputer hanya mengerti penggunaan bahasa mesin.

- Bahasa tingkat tinggi

Bahasa ini merupakan bahasa pemrograman yang paling mendekati dengan bahasa manusia. Bahasa pemrograman tingkat tinggi ini berisi instruksi-instruksi yang menggunakan bahasa alamiah yang dapat dengan mudah dimengerti oleh manusia, seperti bahasa Inggris atau matematika. Tingkatan ini diciptakan untuk mengatasi kekurangan yang dimiliki tingkat menengah dan tingkat rendah.



Untuk mempermudah dalam mengembangkan *software*, bahasa pemrograman menyediakan sebuah fitur yang bernama *framework*. *Framework* merupakan kerangka kerja yang dapat digunakan untuk mengembangkan aplikasi berbasis desktop atau aplikasi berbasis website [17]. Dengan adanya *framework*, pengembang akan lebih mudah dalam membuat aplikasi atau *website*. Hal tersebut terjadi karena pengembang hanya perlu menyusun komponen-komponen pemrograman yang sudah jadi. Atau dengan kata lain, pengembang tidak perlu membuat berbagai fitur dari awal lagi. *Framework* diciptakan untuk membantu pengembang dalam mengembangkan aplikasi atau website lebih cepat serta tersusun dan terstruktur. Dengan begitu, pengembang dan pemrogram tidak perlu melakukan koding program yang diulang-ulang.

Berikut adalah beberapa manfaat dari *framework*

- program menjadi lebih terstruktur dan rapi,
- praktis dalam pengembangan,
- keamanan yang lebih unggul.

Pada saat ini berbagai macam bahasa pemrograman yang tersedia untuk membantu para pengembang *software* untuk mengembangkan *software* yang sesuai dengan kebutuhan. Dari sekian banyak bahasa pemrograman, terdapat 2 bahasa pemrograman yang cukup banyak digunakan dalam pengembangan *website*. 2 bahasa pemrograman tersebut adalah Python dan PHP.

### 2.2.1 PHP

PHP atau *Hypertext Preprocessor* merupakan bahasa pemrograman dengan sisi *server* yang dirancang untuk pengembangan *web* [7]. Bahasa pemrograman PHP termasuk kedalam *open source* sehingga dapat diunduh dan digunakan dengan gratis. PHP termasuk bahasa pemrograman yang mudah untuk dipelajari karena PHP sudah diinterpretasikan dan tidak membutuhkan kompiler. Bahasa yang pertama kali diinspirasi oleh Rasmus Ledorf ini memiliki ekstensi *.php*

Berikut merupakan karakteristik dari bahasa pemrograman PHP:

- kode PHP dieksekusi di *server*,
- PHP dapat diintegrasikan dengan banyak *database*, seperti Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, maupun Informix,
- sangat kuat dalam memegang konten dari *system management* seperti WordPress dan dapat digunakan untuk mengontrol akses *user*,

PHP dapat dengan mudah ditambahkan di *file* HTML dan kode HTML juga dapat dituliskan di *file* PHP,

PHP banyak digunakan karena PHP dapat melakukan apapun yang berhubungan dengan *scripting* pada sisi *server* atau lebih dikenal dengan *backend* dari sebuah *website*. Sebagai contoh, PHP dapat menerima data dari formulir yang sudah diterima, membuat konten dari *website* lebih dinamis dan menarik, dapat bekerja dengan *database*, mengirim dan menerima *cookies*, mengirim *email*, dan seterusnya. Dalam PHP juga terdapat banyak fungsi *hash* untuk meng-enkripsi *data user* sehingga PHP menjadi lebih aman dan reliabel.

Bahasa Pemrograman PHP memiliki banyak sekali framework. Bahkan terdapat framework bawaan dari Moodle yang berbahasa PHP. Berikut merupakan dua contoh dari framework PHP, yakni Laravel dan CodeIgniter

### 2.3.1.1.Laravel

Laravel merupakan salah satu framework dari bahasa pemrograman PHP yang paling banyak digunakan. Framework ini dikenal karena memiliki sintaks yang mudah dipahami dan digunakan. Dengan Laravel, pengembangan *website* dan aplikasi dapat diselesaikan dalam waktu yang lebih singkat. Laravel menyediakan fungsi, seperti otentikasi *user*, manajemen sesi, dan *caching* sehingga pekerjaan-pekerjaan dasar tersebut tidak perlu dilakukan lagi. Singkatnya, Laravel menawarkan semua fungsionalitas yang dibutuhkan untuk membuat serta mengembangkan aplikasi atau *website* PHP modern [16].

Dari segi performa, Laravel memiliki *core* yang dapat diandalkan serta dikembangkan dengan menggunakan berbagai *add-ons*. Di sisi lain, Laravel juga terintegrasi dengan *library* dan *platform* pihak ketiga, seperti Amazon Web Services (AWS). Untuk *task* yang membutuhkan waktu penyelesaian yang lama (sekitar lebih dari 30 detik), pengembang bisa mengurutkannya agar dapat dijalankan secara *asynchronously* di background sehingga performa dapat ditingkatkan menjadi lebih baik.

Laravel memiliki komunitas yang besar dan aktif. Adanya komunitas ini membantu setiap memudahkan pengembang dalam mengembangkan aplikasi atau *website* dengan menggunakan Laravel.



## 2.3.1.2. CodeIgniter

CodeIgniter merupakan framework PHP yang menggunakan arsitektur Model View Controller (MVC) sehingga CodeIgniter menggunakan komponen yang berbeda-beda untuk mengelola *task development* yang spesifik [16]. Metode yang ditawarkan CodeIgniter lebih disukai oleh para pengembang karena mereka bisa membuat serta mengembangkan aplikasi web yang mudah diskalakan tanpa menyita banyak ruang.

Kemudahan dalam pembelajaran dan penggunaan membuat CodeIgniter menjadi salah satu framework PHP terbaik bagi pemula. Kelebihan lainnya adalah informasi bantuan atau tutorial yang lengkap. Pengembang yang baru saja mengenal CodeIgniter pun dapat memahami hal-hal dasar terkait penggunaan CodeIgniter dengan lebih mudah. Performa dari *website* dan aplikasi yang dikembangkan dengan menggunakan CodeIgniter juga dapat diandalkan, terlebih jika itu merupakan aplikasi yang ‘ringan’ yang dijalankan di server sederhana.

Di samping kelebihan, juga ada kekurangan. Sebagai contoh, tidak ada jadwal yang tetap dan teratur mengenai update CodeIgniter. Jadi PHP framework ini tidak disarankan untuk digunakan bilamana aplikasi yang dikembangkan memiliki standar keamanan tinggi.

## 2.2.2 Python

Python merupakan bahasa pemrograman tingkat tinggi yang sangat populer. Bahasa pemrograman Python dapat digunakan pada pengembangan *web*, penerapan dari *machine learning*, serta akrab dengan teknologi paling mutakhir dari industri perangkat lunak [8]. Python sangat cocok digunakan oleh pemula, maupun oleh orang yang sudah berpengalaman pada bahasa pemrograman lain seperti C++ dan Java.

Berikut merupakan beberapa fakta dari bahasa pemrograman Python:

- menjadi bahasa pemrograman tingkat tinggi yang paling banyak digunakan sehingga memudahkan proses *troubleshoot*,
- Python memungkinkan untuk pemrograman berorientasi objek dan paradigma prosedural,
- program Python relatif kecil jika dibandingkan dengan bahasa pemrograman lain seperti Java.



bahasa Pemrograman Python digunakan oleh hampir semua perusahaan raksasa yang bergerak dibidang teknologi, seperti Google, Amazon, Facebook, Instagram, Dropbox, Uber, dan sebagainya,

- keunggulan terbesar dari Python adalah banyaknya *library* standar yang dapat digunakan pada *machine learning*, Aplikasi GUI, *Web Frameworks*, *Image Processing*, *Web Scrapping*, dan sebagainya.

Bahasa pemrograman Python diciptakan oleh Guido van Rossum pada tahun 1991 dan kemudian dikembangkan oleh Python Software Foundation. Python didesain dengan menekankan pada kemampuan membaca kode. Dan sintaks yang dimiliki oleh Python memungkinkan pengembang mengekspresikan konsep yang dimiliki dengan kode yang lebih singkat. Jadi bahasa Python merupakan bahasa pemrograman yang dapat mempercepat pekerjaan dan mengintegrasikan sistem menjadi lebih efisien.

Python juga memiliki framework yang beragam, walau tidak sebanyak PHP. Dan Framework yang penulis bandingkan untuk digunakan dalam mengembangkan *plug-in* adalah Django dan Flask.

### **2.3.2.1.Django**

Django merupakan framework Python yang *open-source* dan dapat digunakan dengan gratis. Django memungkinkan pengembang mengembangkan kode dan aplikasi kompleks dengan cepat. Framework Django membantu dalam mengembangkan aplikasi web yang berkualitas. Django merupakan salah satu framework python terbaik yang digunakan untuk pengembangan API dan aplikasi web dengan cepat.

Framework tingkat tinggi ini menyederhanakan pengembangan aplikasi web dengan memberikan fitur-fitur canggih yang berbeda. Django juga memiliki berbagai macam *library* dan mengutamakan efektivitas, menyederhanakan pengkodean, dan komponen yang dapat digunakan kembali.

### **2.3.2.2.Flask**

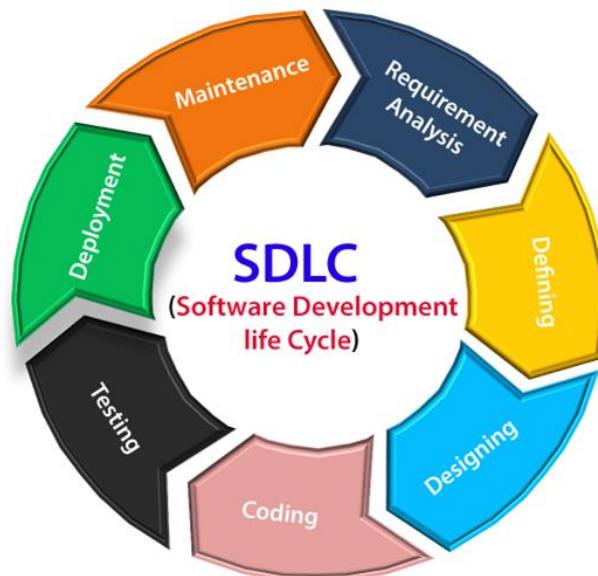
Flask merupakan framework Python yang dapat diakses di bawah lisensi BSD. Flask terinspirasi oleh framework Sinatra Ruby. Flask bergantung pada toolbox Werkzeug WSGI dan template Jinja2. Tujuan utamanya adalah untuk membantu mengembangkan basis aplikasi web yang kuat.



Pengembang dapat mengembangkan framework Python dengan cara apa pun yang dibutuhkan. Akan tetapi Flask dirancang untuk aplikasi yang terbuka. Flask telah digunakan oleh perusahaan besar, seperti LinkedIn dan Pinterest. Dibandingkan dengan Django, Flask paling cocok untuk proyek kecil dan mudah. Dengan demikian, pengembang dapat mengharapkan pengembangan server web, dukungan untuk Google App Engine serta pengujian unit bawaan.

### 2.3 Software Development Life Cycle

Dalam *Software Engineering*, SDLC atau *Software Development Life Cycle* merupakan langkah langkah yang dilakukan dalam mengembangkan perangkat lunak [10]. Tujuan dari SDLC adalah untuk menghasilkan sistem terbaik dengan biaya dan usaha serendah-rendahnya, memenuhi permintaan dari klien, serta pengembangan sesuai dengan jadwal yang sudah ditentukan.



Gambar 2.2 Software Development Life Cycle [10]

Secara umum, SDLC terdiri dari 7 langkah atau fase, antara lain : *understanding the problem* (pemahaman terhadap masalah), *planning* (perencanaan), *feasibility analysis* (analisis kelayakan), *design and prototyping* (pembuatan desain dan prototipe), *coding* (penulisan kode), *software testing* (pengujian perangkat lunak), *maintenance and support* (perawatan dan dukungan).

*Understanding the problem* bertujuan untuk mengembangkan fungsionalitas penuh dari solusi perangkat lunak. Fase ini bagi banyak pengembang perangkat lunak dianggap sebagai fase yang paling penting karena sangat penting bagi pengembang perangkat lunak untuk memahami



dengan penuh penyelesaian dari permasalahan yang dihadapi beserta dengan persyaratan tambahan apapun yang mungkin dimiliki oleh organisasi.

*Planning* merupakan fase yang dilakukan tim pengembang untuk mengorganisir dan menyusun rencana untuk mengembangkan perangkat lunak. Sebagai bagian dari fase *planning*, tim pengembang akan memperkirakan biaya dan sumber daya yang dibutuhkan. Fase ini seringkali digabungkan dengan fase analisis kelayakan.

*Feasibility analysis* merupakan fase yang dilakukan oleh tim pengembang untuk menentukan cara tercepat untuk mengimplementasikan proyek yang sedang dikerjakan dengan risiko dan biaya yang paling rendah. Fase ini juga memberikan perincian tentang risiko yang ada dan menyediakan rencana yang dilakukan untuk meringankan risiko tersebut. Menentukan kelunakan juga membantu memastikan tim pengembang tidak berinvestasi dalam proyek yang tidak dapat diselesaikan.

*Design and prototyping* bertujuan untuk membuat desain yang *user-friendly*. Hasilnya *user* dapat dengan mudah mencapai hasil yang terbaik ketika menggunakan perangkat lunak. Tim pengembang seringkali membangun prototipe pada fase ini agar lebih meyakinkan para *stakeholder* dari proyek. Prototipe terkadang dapat menjadi versi pertama dari produk yang kemudian dapat dikembangkan lebih lanjut setelah proses perilisan.

*Coding* dilakukan setelah desain produk selesai dilakukan. Apabila pada fase-fase sebelumnya berjalan dengan baik dan lancar, maka kemungkinan besar pada fase ini akan berjalan dengan baik dan lancar pula.

*Software testing* dilakukan untuk menjamin kualitas dari perangkat lunak yang sudah dibuat sebelum proses perilisan. Pada fase ini, tim penjamin kualitas akan mencari kecacatan dari perangkat lunak kemudian dikirimkan ke tim pengembang yang bertanggung jawab.

*Maintenance and support* merupakan fase yang dilakukan setelah proses perilisan perangkat lunak. Sebagian tim pengembang akan menentukan periode waktu yang dilakukan untuk fase ini. Apabila kecacatan perangkat lunak ditemukan, maka tim *support* akan memperbaikinya.

## **2.4 Learning Analytics**

*Learning analytics* adalah proses mengumpulkan dan menganalisis data-data yang dimiliki oleh peserta didik dan lingkungannya untuk tujuan memahami dan meningkatkan hasil belajar. *Learning analytics* berisi informasi apa pun yang dapat membantu semua *user* LMS, yang terdiri



dan peserta didik, tenaga pengajar, administrator, maupun pengambil keputusan. Kebutuhan utama untuk *learning analytics* adalah untuk mempromosikan perkembangan peserta didik dengan mengidentifikasi status siswa (berisiko, normal, unggul), berdasarkan aktivitas mereka dalam proses belajar mengajar.

Selain yang telah disebutkan di atas, *learning analytics* juga dapat digunakan untuk :

- indikator alat ukur performa peserta didik,
- mendukung perkembangan peserta didik,
- memahami dan meningkatkan tingkat efektivitas praktik mengajar,
- menginformasikan hasil dari keputusan dan strategi institusi.

The Horizon Report: 2019 Higher Education Edition yang diproduksi oleh EDUCASE Learning Initiative mengidentifikasi *learning analytics* sebagai salah satu strategi dan teknologi digital yang diharapkan dapat digunakan secara umum dalam waktu dekat. Dengan demikian, *Data Analyst* yang terampil diperlukan untuk mendukung gerakan ini dan mewujudkan kesuksesan institusional.

“Salah satu kekuatan pendorong di balik pertumbuhan bidang *learning analytics* adalah institusi pendidikan saat ini menghadapi tantangan sumber daya yang terbatas dan akuntabilitas yang meningkat, sehingga mendorong kreativitas dan penemuan,” kata Melisa LaCroix, administrator *database* dan alumni Northeastern University. “Program analitik yang dulu disediakan untuk bisnis besar sekarang banyak digunakan di pendidikan tinggi dan lembaga K-12 untuk mengukur pertumbuhan siswa, menginformasikan keputusan kurikulum, dan mengidentifikasi siswa yang berisiko gagal dalam kursus atau program.”

“*Learning analytics* bukanlah solusi ajaib untuk tantangan pendidikan,” catat LaCroix. Oleh karena itu, *learning analytics* harus dilengkapi dengan komunikasi yang kuat antara analyst dan tenaga pendidik agar berhasil memanfaatkan wawasan yang diperoleh dari upaya tersebut. “Untuk membangun *learning analytics* yang berhasil dalam pendidikan, penting untuk melibatkan dan menginformasikan para pemimpin sekolah, mendengarkan kebutuhan tenaga pengajar dan peserta didik, serta mengedukasi *user* tentang cara menggunakan dan bertindak berdasarkan data yang tersedia. Tenaga pengajar dan administrator juga hanya akan berkomitmen pada pengambilan keputusan berdasarkan data.”

Sebuah *learning analytics* yang baik, harus memiliki 3 kriteria, yakni :



- *Predictive* : menggabungkan data-data yang tersedia untuk menunjukkan status peserta didik saat ini dan membuat prediksi status peserta didik di masa depan.
- *Proactive* : memberikan pemberitahuan kepada *user* tentang apa yang harus dilakukan agar status peserta didik menjadi lebih baik.
- *Verifiable* : hasil prediksi dengan hasil akhir dapat dibandingkan dan hasil perbandingan tersebut hampir mirip / akurat.

## **BAB 3**

### **ANALISIS STUDI PUSTAKA KUNCI DAN PEMILIHAN METODE**

#### **3.1 Analisis Jenis LMS**

LMS memiliki banyak sekali jenis. Di antara jenis-jenis LMS tersebut ada yang cocok digunakan oleh institusi pendidikan, lembaga pemerintahan, maupun perusahaan. Dari sekian banyak LMS, penulis membandingkan 3 LMS yang paling cocok digunakan oleh institusi pendidikan, khususnya perguruan tinggi karena *Learning Analytics* yang penulis kembangkan memiliki paling cocok digunakan pada *environment* tersebut. 3 LMS tersebut adalah Moodle, Blackboard Learn, dan Schoology. Berikut merupakan karakteristik dari ketiga LMS tersebut.

##### **3.1.1 Moodle**

- Gratis
- Proses instalasi yang mudah
- Open source

##### **3.1.2 Blackboard Learn LMS**

- Terdapat collaboration tools yang memudahkan pengguna
- Beintegrasi dengan zoom
- Kurang intuitif
- UI yang disediakan tidak terlalu bagus

##### **3.1.3 Schoology**

- Sering terjadi *lag*
- Kurangnya kustomisasi
- Tampilan yang user friendly
- Pembelajaran berbasis *project*



### 3.1.4 Perbandingan Moodle, Blackboard Learn, dan Schoology

Perbandingan lebih jelas antara ketiga LMS tersebut dapat dilihat pada tabel di bawah ini.

Tabel 3.1 Perbandingan Moodle, Blackboard Learn, dan Schoology

Parameter	Moodle	Blackboard Learn	Schoology
Jumlah pengguna	Lebih dari 200 juta yang tersebar di 242 negara	Lebih dari 100 juta pengguna yang berkonsentrasi di Amerika Utara	Belasan juta pengguna yang berkonsentrasi di Amerika Serikat
Deployment	Web, Android, iOS, Windows, Mac	Web, Android, iOS	Web, Android, iOS
Biaya	Gratis	Gratis	Gratis
Open Source	Ya	Tidak	Tidak
Support plugin	Ya	Ya	Ya

## 3.2 Pemilihan LMS

Setelah membandingkan ketiga LMS yang sudah disebutkan di atas, penulis memutuskan untuk menggunakan Moodle yang menjadi *environment* untuk mengembangkan *Learning Analytics*. Ada beberapa alasan yang mendasari keputusan penulis. Pertama, Moodle memiliki pengguna yang paling banyak dibandingkan dua LMS yang lain serta lebih *diverse* sehingga *Learning Analytics* yang penulis kembangkan dapat lebih menjangkau lebih banyak pengguna. Alasan kedua, Moodle termasuk ke dalam open source sehingga memudahkan proses pengembangan *Learning Analytics* dari penulis.



## BAB 4

### DETAIL IMPLEMENTASI

#### 4.1 Luaran *Capstone Project* Beserta Spesifikasinya

Proyek berjudul “Pengembangan Prototipe Plugin Learning Analytics Pada Moodle 3.9” yang dilakukan oleh penulis memiliki luaran dalam tiga bentuk. Bentuk pertama adalah sebuah *plug-in* Moodle yang digunakan untuk menghitung tingkat keberhasilan peserta didik dalam suatu mata kuliah. Luaran tersebut merupakan intisari dari seluruh proyek *Capstone* penulis. Luaran kedua memiliki jenis sistem informasi yang merupakan tahap lanjutan dari luaran pertama, yaitu sistem yang digunakan untuk memberikan informasi mengenai tingkat keberhasilan peserta didik. Bentuk luaran ketiga dari proyek *Capstone* penulis adalah terdapatnya dokumen yang berisikan proses perencanaan model, dokumentasi model, serta analisis dari model yang terbentuk. Luaran ketiga disajikan dalam bentuk dokumen C-501 yang ditulis penulis dimana terdapat dokumentasi yang lengkap mengenai proyek.

Tabel 3.2 Luaran Proyek *Capstone*

Jenis Luaran	Contoh
<i>Software</i> di PC	<i>Plug-in</i> untuk menghitung tingkat keberhasilan peserta didik
Sistem Informasi	Sistem informasi pemberitahuan tingkat keberhasilan peserta didik
Dokumen (Kebijakan, SOP, lingkungan, Ekonomi, dan lain-lain)	SOP teknis perencanaan model, dan dokumentasi model yang terbentuk dan analisisnya

Spesifikasi yang penulis gunakan untuk proyek *capstone* ini hanya berjumlah satu, yakni tingkat akurasi. Akurasi merupakan parameter yang menjelaskan seberapa banyak hasil prediksi yang sesuai dengan data nilai riil-nya. Setiap model data, akan menghasilkan tingkat akurasi yang berbeda pula.

Tabel 3.3 Spesifikasi Luaran Proyek

No	Spesifikasi	Satuan	Standar	Keterangan
1	Akurasi	Persentase (%)	$\geq 80\%$	Tercantum



## 4.2 Batasan Masalah

Penulis mengembangkan sebuah *plug-in* yang berbasis Moodle yang mampu memberikan keluaran dari sistem yang digunakan untuk memprediksi pencapaian akhir dari peserta didik di mata kuliah yang diikutinya. Kemudian *plug-in* juga dapat memberikan beberapa saran kepada peserta didik berdasarkan prediksi yang dihasilkan. Akan tetapi, penulis tidak mampu memilih bagaimana sistem yang paling optimal untuk digunakan karena hal tersebut sudah dilakukan oleh rekan dari penulis.

Bentuk data yang diolah juga merupakan batasan lain dari penulis. Data yang dapat diolah adalah data *course* yang sudah selesai. Semakin banyak data yang diambil, maka semakin tinggi tingkat akurasi dari sistem *machine learning*. Hal tersebut juga berlaku sebaliknya, yakni jika data yang digunakan sedikit, maka tingkat akurasi yang dihasilkan tidak terlalu tinggi. Dikarenakan waktu yang terbatas, data yang diolah oleh penulis hanyalah data *dummy* yang penulis buat khusus untuk proyek ini yang terdiri dari satu buah mata kuliah. Meskipun demikian, data *dummy* yang penulis gunakan memiliki basis pada data yang dihasilkan pada mata kuliah Transformasi Digital pada tahun 2021 semester 1. Terdapat juga batasan dimana data pada mata kuliah tersebut tidak menyimpan data personal mahasiswa, hanya data interaksi. Dengan begitu, data yang digunakan untuk melatih model menjadi spesifik khusus untuk kasus mata kuliah tersebut saja sehingga model tidak dapat berfungsi secara optimal dalam memprediksi performa mahasiswa dalam mata kuliah lainnya.

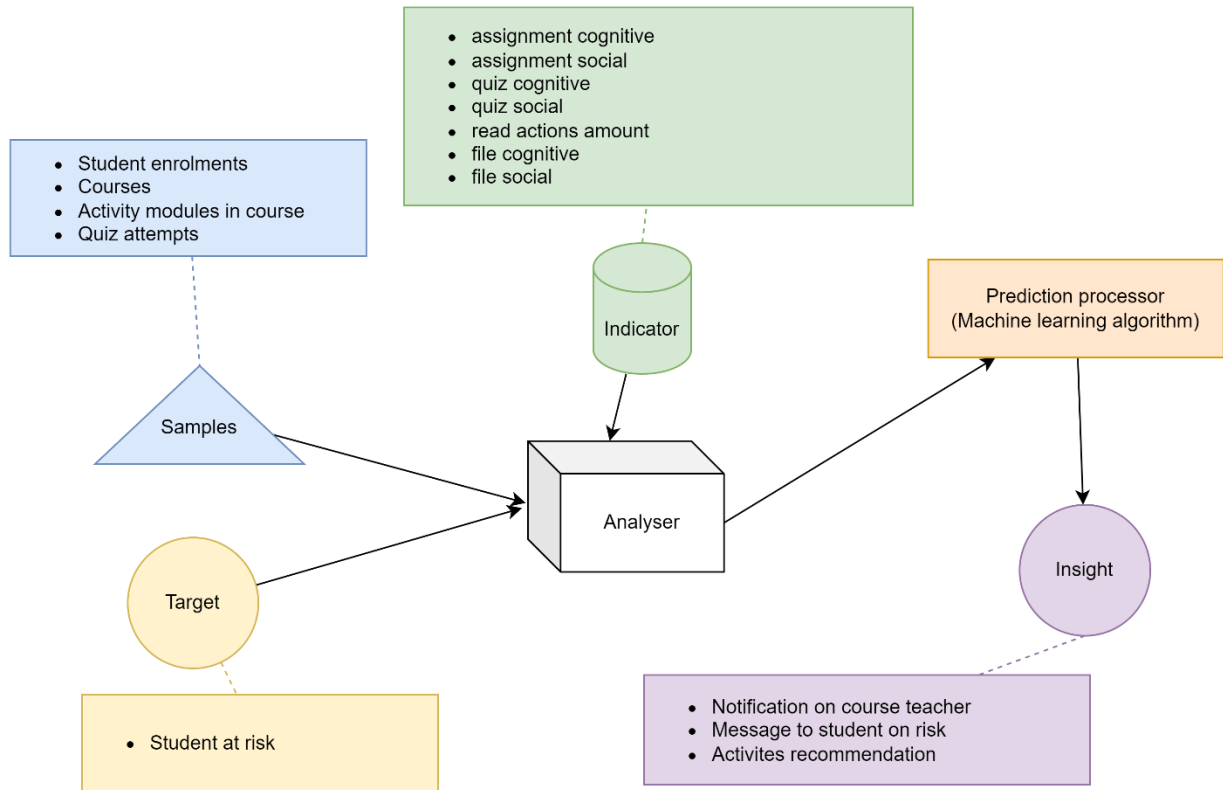
Batasan terakhir dari proyek *capstone* ini adalah hasil akhir atau *plug-in* dari proyek ini yang masih berupa prototipe. Dikarenakan masih berupa prototipe, *plug-in* yang penulis kembangkan bukanlah suatu produk akhir yang dapat dipasangkan pada sebuah LMS, seperti eLOK. Akan tetapi, penulis telah berhasil melakukan pengujian pada Moodle 3.9 yang merupakan basis dari eLOK dan didapatkan bahwa fitur dan fungsi dari *plug-in* tersebut sesuai dengan kebutuhan.

## 4.3 Detail Rancangan

Produk keluaran dari proyek *capstone* ini berupa *plug-in* yang dapat melakukan prediksi terhadap hasil belajar peserta didik. *Plug-in* tersebut lalu dipasangkan pada Moodle 3.9 yang merupakan basis dari eLOK. Proses perancangan sistem dari *plug-in* digambarkan dengan diagram-diagram di bawah ini dan juga bagaimana integrasi antara *plug-in* dengan Moodle. Diagram tersebut berupa komponen-komponen dari *plug-in* dan *data flow diagram*. Perlu digaris bawahi bahwa *plug-in* hanya digunakan untuk menampung sistem untuk memprediksi pencapaian hasil akhir mahasiswa sehingga metode untuk memprediksi tersebut tidak terlalu dijelaskan pada dokumen ini.

#### 4.3.1 Learning Analytics Component

Terdapat beberapa komponen-komponen penting yang terdapat pada *plug-in* yang penulis kembangkan untuk melakukan fungsi *learning analytics*. Komponen-komponen tersebut akan dijelaskan pada gambar 4.1 di bawah ini.



Gambar 4.1 *Learning analytics component*

Komponen yang pertama bernama *Samples*. *Samples* merupakan kumpulan data pada Moodle yang akan digunakan untuk membuat suatu perhitungan. Kumpulan data tersebut tidak terkait dengan pengujian data maupun PHPunit data. *Samples* juga diidentifikasi oleh *id* yang sesuai dengan elemen data yang menjadi perhitungan. *Id* tersebut dapat berupa *id* entitas dari Moodle, seperti *id course*, *id user*, *id enrolment*, maupun *id* entitas yang lain. Setiap tipe entitas Moodle yang digunakan sebagai *sample* membantu mengembangkan prediksi yang melibatkan entitas tersebut. Misalnya, *sample* berisi *quiz attempts* akan membantu mengembangkan wawasan potensial yang mungkin ditawarkan analitik yang terkait dengan upaya *quiz* oleh sekelompok siswa tertentu.

Komponen kedua bernama *target*. *Target* merupakan representasi dari tujuan yang diprediksi oleh model data. *Target* juga menentukan tindakan apa yang harus dilakukan setelah



prediksi tersebut diterima. Pada *plug-in* yang penulis kembangkan, tujuan yang menjadi *target* adalah peserta didik yang memiliki risiko mendapat nilai rendah atau gagal pada *course*.

Komponen selanjutnya adalah *indicator*. *Indicator* merupakan *predictor value* atau variabel independen pada sebuah *course* yang sedang diawasi. *Indicator* menentukan kumpulan data minimum yang dibutuhkan untuk melakukan perhitungan. *Indicator* dapat dengan mudah digunakan kembali pada model data yang berbeda sehingga *indicator* tersebut harus dapat bekerja pada analisis yang berbeda. Contohnya adalah *indicator* keaktifan peserta didik di forum pada suatu *course* dapat digunakan untuk mencari peserta didik yang sudah tidak aktif di dalam *course* tersebut ataupun mencari peserta didik yang pemalu. Dalam mengerjakan proyek ini, penulis menggunakan 7 *indicator* sesuai dengan model data yang disediakan, yakni *assignment cognitive* atau nilai yang dihasilkan dari tugas, *assignment social* atau interaksi dengan tugas, *quiz cognitive* atau nilai *quiz*, *quiz social* atau interaksi dengan *quiz*, *read actions amount* atau jumlah materi yang dibaca di Moodle, *file cognitive* atau jumlah file yang diunduh dalam *course*, dan *file social* atau interaksi dengan *file*.

*Analyzer* merupakan komponen selanjutnya dari *plug-in* yang penulis kembangkan. *Analyzer* bertugas untuk membuat *dataset file* yang akan dikirimkan kepada *machine learning processors*. *Dataset file* tersebut dikodekan sebagai kelas PHP. Selain itu, *analyzer* juga memiliki tugas lain, seperti mendefinisikan isi dari prediksi, membuang data yang tidak valid, menyaring *samples* yang sudah di-*training*, meneruskan perhitungan ke *indicator* dan *target*, merekam semua perhitungan, dan merekam semua *sample id* yang sudah di-analisis dalam *database*.

Komponen selanjutnya bernama *prediction processors*. *Prediction processors* merupakan *machine learning backends* yang memproses *dataset* yang dihasilkan dari *indicator* dan *target*. *Prediction processors* termasuk kedalam sub tipe *tool\_inspire plug-in* sehingga memiliki dapat untuk mengevaluasi model yang disediakan, melatih algoritma *machine learning*, memprediksi *target* berdasarkan algoritma yang sudah dilatih, dan berkomunikasi dengan Moodle melalui *file*.

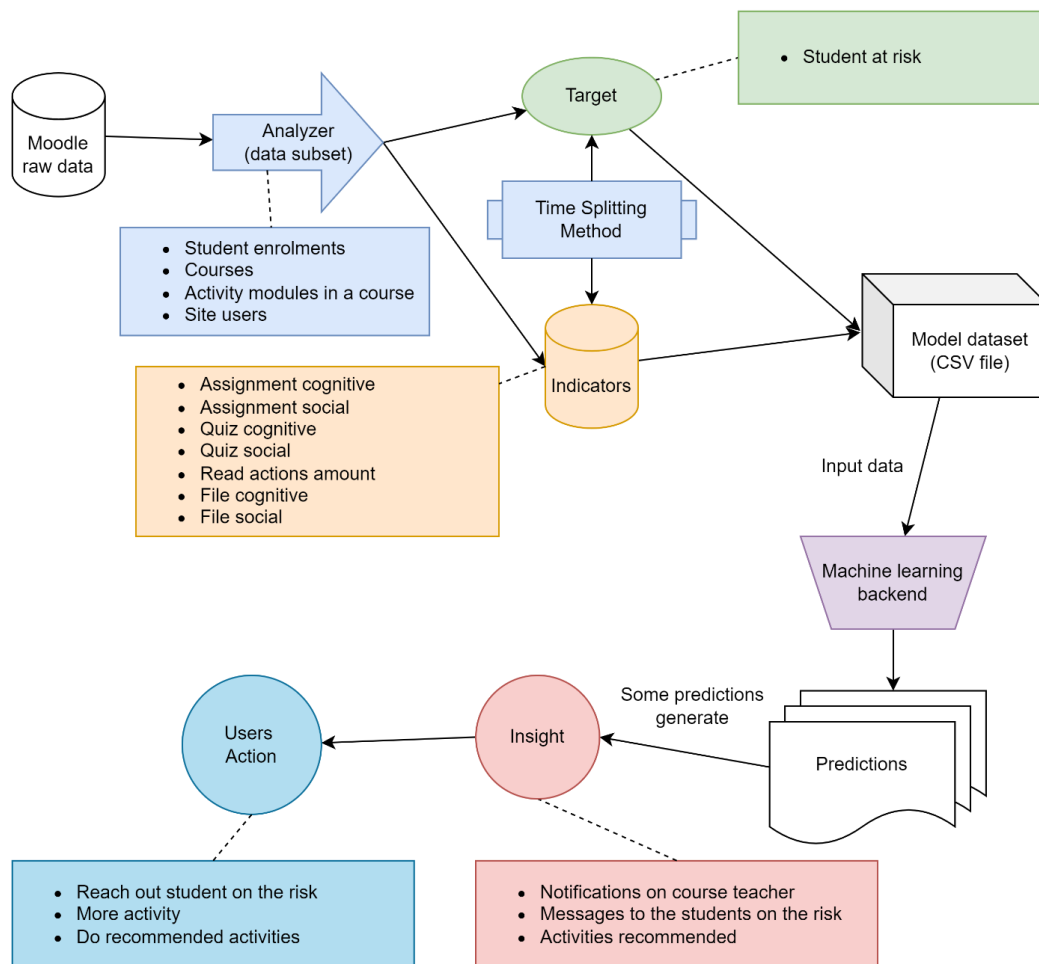
Komponen terakhir dari *plug-in* yang penulis kembangkan adalah *insight*. *Insight* adalah representasi dari hasil prediksi yang dibuat tentang elemen tertentu dari *sample* di dalam suatu model *analyzer*. *Insight* digunakan untuk memberikan pemberitahuan kepada *user* tentang hasil prediksi yang sudah dilakukan sebelumnya. Selain itu, *insight* juga dapat memberi saran kepada *user* tentang aktivitas apa yang diperlukan pada setiap prediksi. Misalnya, *insight* memberikan pemberitahuan tentang seorang peserta didik yang terancam gagal pada suatu *course* karena tidak



pernah membaca materi yang dibagikan, sehingga *insight* juga akan memberi saran kepada peserta didik dan tenaga pengajar agar peserta didik lebih banyak membaca materi.

#### 4.3.2 Data Flow Diagram

Agar dapat berfungsi dengan baik, suatu *plug-in* harus memiliki cara kerja yang jelas. Dan *plug-in* yang penulis kembangkan memiliki cara kerja seperti yang dijelaskan pada gambar 4.2



Gambar 4.2 Data flow diiagram

Pada awalnya data yang dimiliki oleh Moodle diambil semuanya oleh *Analyzer*. Setelah itu, *analyzer* mengelompokkan data mentah tersebut menjadi dua, yakni *target* dan *indicator*. *Target* merupakan suatu tujuan yang ingin dicapai sedangkan *indicator* merupakan variabel data yang digunakan. Setelah didapatkan *indicator* dan *target*, keduanya lalu digabungkan agar membuat sebuah model dataset. Dimana model dataset tersebut akan digunakan untuk proses prediksi.

Sebuah *course* memiliki tanggal dimulainya *course* dan tanggal selesai. Oleh karena itu, dalam melakukan pengelompokan data menjadi *target* dan *indicator* harus memiliki rentang waktu yang sesuai. Apabila tidak memiliki rentan waktu, hasil prediksi yang dihasilkan akan selalu



inkonsisten dan menjadikannya sangat kompleks. Misalnya, ketika *course* sudah selesai dan ada seorang peserta didik yang kemudian menjadi lebih rajin seperti mengumpulkan tugas atau mengikuti *quiz* yang tertinggal. Seharusnya kegiatan yang dilakukan peserta didik tersebut tidak akan berpengaruh pada hasil akhir karena sudah melewati batas waktu dari *course*. Oleh karena itu, digunakanlah *time splitting methods* agar perhitungan dilakukan pada rentang waktu tertentu. Dari sekian banyak rentang waktu yang dapat digunakan, penulis memilih rentang waktu tiap 4 bulan dikarenakan waktu aktif *course* di eLOK kebanyakan hanya 4 bulan. Rentang waktu yang digunakan juga bisa dipilih dari dimulainya *course* (akumulatif) atau rentang waktu tertentu.

Dataset yang telah dibuat dari gabungan *target* dan *indicator* akan dikirimkan ke dalam *machine learning backends*. Di sini dataset tersebut akan digunakan untuk *machine learning training*, prediksi, dan *model evaluation*. *Machine learning backend* dapat dibuat dengan menggunakan bahasa PHP maupun Python. Penulis menggunakan bahasa PHP karena sudah menjadi *machine learning backend* bawaan dari moodle sehingga tidak membutuhkan dependensi eksternal.

Hasil prediksi dari *machine learning backend* yang sesuai dengan *analyzer* akan menjadi *insight*. *Insight* yang penulis gunakan adalah memberikan pemberitahuan kepada tenaga pengajar pada *course*, mengirim pesan kepada peserta didik yang memiliki risiko gagal dalam *course*, dan merekomendasikan kegiatan untuk mengurangi tingkat risiko gagal pada *course*. Setelah mendapatkan *insight*, *user* dapat melakukan kegiatan yang disarankan oleh *insight*.

#### 4.3.3 Integrasi Plug-in Dengan Moodle

Seperti yang sudah disebutkan sebelumnya, *plug-in* merupakan suatu program tambahan yang berisi fungsi atau fitur spesifik yang dapat ditambahkan pada suatu *website*. Dikarenakan berbeda dengan *website* itu sendiri, diperlukan suatu perintah di dalam *plug-in* agar dapat berintegrasi dengan Moodle sehingga *plug-in* yang dikembangkan dapat berfungsi dengan baik. Dalam hal ini, Moodle menyediakan banyak sekali API yang dapat digunakan *plug-in* untuk berinteraksi dengan *core system* dari Moodle.

Ketika membuat *plug-in*, tidak bisa dipungkiri bahwa akan terdapat halaman tambahan dalam suatu *website*, tak terkecuali pada Moodle. Moodle menyediakan sebuah API yang bernama Page API yang dapat digunakan untuk mengatur halaman, menambahkan JavaScript, serta mengatur *interface* yang akan ditampilkan pada *user*. Berikut beberapa sintaks dalam Page API yang terdapat pada *plug-in* pada proyek *capstone* ini.



```
$PAGE->Set_context($context);

// Mengatur plug-in agar sesuai dengan context atau course dalam Moodle

$PAGE->set_url($url);

// Mengatur letak url plug-in di dalam Moodle

$PAGE->set_pagelayout('report');

// Mengatur tata letak interface plug-in agar sesuai dengan report

$PAGE->set_title($context->get_context_name());

// Mengatur judul plug-in

$PAGE->set_heading($context->get_context_name());

// Mengatur heading plug-in
```

Setelah itu, langkah yang perlu dilakukan dalam melakukan integrasi adalah dengan menyamakan database yang digunakan oleh *plug-in* dan Moodle. Moodle mengatur koneksi ke dalam *database* berdasarkan *value* yang sudah diatur pada *file* config.php. Ada sebuah API dari Moodle yang dapat digunakan agar *plug-in* dapat menerima atau mengubah isi dari suatu database di Moodle. Fungsi tersebut bernama Data Manipulation API. Data Manipulation API akan mengakses objek *database* melalui *public method*. Sintaks yang digunakan untuk mengakses database di dalam Moodle adalah sebagai berikut

```
<?php
require(__DIR__ . '/config.php');
```

Setelah mempunyai akses yang sama dengan *database* dari Moodle, langkah selanjutnya yang dilakukan adalah menentukan bagaimana *plug-in* akan dipasangkan dalam Moodle. Ketika mengembangkan *plug-in* Moodle dengan menggunakan framework yang sudah disediakan oleh Moodle, maka terdapat sebuah *file* yang bernama install.php yang terletak pada folder db. *File* tersebut mengijinkan pengembang untuk mendefinisikan bagaimana proses instalasi pada *plug-in*. Isi pada *file* install.php yang penulis gunakan adalah sebagai berikut :

```
defined('MOODLE_INTERNAL') || die();
// Mendefinisikan bahwa software yang dikembangkan termasuk ke dalam plug-in Moodle

function xmldb_tool_inspire_install() {
```



```
$target =  
\tool_inspire\manager::get_target('\tool_inspire\local\target\course_dropou  
t');  
$indicators = \tool_inspire\manager::get_all_indicators();  
  
$model = \tool_inspire\model::create($target, array());  
  
$model->update(0, $model->get_potential_indicators());  
}  
  
// Mendefinsikan variabel-variabel yang akan digunakan pada plug-in
```



## BAB 5

### PENGUJIAN DAN PEMBAHASAN

#### 5.1 Pengujian dan Pembahasan

##### Models list

Target	Enabled	Indicators	Time splitting	View model predictions	Actions
Students at risk of dropping out	✓	<ul style="list-style-type: none"> <li>• Assignment cognitive</li> <li>• Assignment social</li> <li>• Quiz cognitive</li> <li>• Quiz social</li> <li>• Read actions amount</li> <li>• File cognitive</li> <li>• File social</li> </ul>	Quarters accumulative	Choose... ▾	Edit ▾

Gambar 5.1 Halaman utama *plug-in*

Dalam pengujian pada proyek *capstone* ini, penulis membuat 2 skenario yang berbeda. Kedua skenario tersebut adalah dua buah mata kuliah atau *course* dalam Moodle yang dibuat khusus oleh penulis khusus untuk pengujian *capstone* kali ini. Tujuan yang hendak dicapai dari kedua skenario tersebut adalah untuk membuktikan apakah *plug-in* yang sudah penulis kembangkan bekerja dengan baik sekaligus membuktikan bahwa dengan bertambahnya *dataset*, maka tingkat akurasi yang dihasilkan akan meningkat. Apabila terdapat suatu kesamaan dalam data yang disajikan merupakan murni suatu ketidaksengajaan. Untuk lebih jelasnya, akan di jelaskan di bawah ini :

##### 5.1.1 Skenario Pengujian 1 dan Analisis

Penulis melakukan pengujian yang pertama ini bertujuan untuk mengetahui apakah *plug-in* yang penulis kembangkan dapat bekerja dengan baik. Pada skenario pertama ini, terdapat sebuah *course* di Moodle yang bernama Forensik Digital. Forensik Digital di Moodle dimulai pada tanggal 8 Oktober 2022 sampai dengan 12 Februari 2023. *Course* tersebut memiliki 4 partisipan, yang terdiri dari 1 tenaga pengajar dan 3 peserta didik. Ketiga peserta didik tersebut antara lain :

- Master Chief, seorang peserta didik yang sering absen ketika di kelas
- Lark Marcus, seorang peserta didik yang sakit-sakitan
- Arthur Leywin, seorang peserta didik yang rajin dalam mengikuti mata kuliah

## Forensik Digital

[Dashboard](#) / [My courses](#) / [FD](#) / [Participants](#)

### Participants



Enrol users

Match

+ Add condition

4 participants found

First name

Surname

<input type="checkbox"/>	First name / Surname	Email address	Roles	Groups	Last access to course	Status
<input type="checkbox"/>	Master Chief	chief@halo.com	Student	No groups	31 days 21 hours	Active
<input type="checkbox"/>	Arthur Leywin	arhur@tbate.com	Student	No groups	180 days 1 hour	Active
<input type="checkbox"/>	Lark Marcus	lark@magus.com	Student	No groups	31 days 21 hours	Active
<input type="checkbox"/>	Admin User	admin@noreply.com	Teacher	No groups	2 secs	Active

Gambar 5.2 Partisipan *course* forensik digital

Ada 4 aspek yang menjadi penilaian pada *course* Forensik Digital, yakni LO 01, LO 02, LO 03, dan Tugas Pengganti UTS. Bobot dari masing-masing LO menyumbangkan 20% dari nilai akhir dan sisanya disumbangkan oleh Tugas Pengganti UTS. Keempat aspek tersebut memiliki bentuk pengumpulan tugas. Untuk mendapatkan nilai masing-masing dari peserta didik harus mengumpulkan file berisi tugas pada rentang waktu yang ditentukan. Apabila sudah terlewat, maka peserta didik mendapat nilai 0.

Ketika tanggal 9 Februari 2023, *course* ini telah melaksanakan 3 penilaian berdasarkan LO (*Learning Outcome*), yakni untuk LO 01, LO 02 dan LO 03 dalam bentuk pengumpulan tugas. Di LO 01 semua peserta didik selesai mengumpulkan tugas. Sedangkan di LO 02 dan LO 03 masing-masing terdapat satu peserta didik yang gagal mengumpulkan tugas tepat waktu. Untuk lebih jelas mengenai penilaiannya bisa di lihat pada gambar 5.3

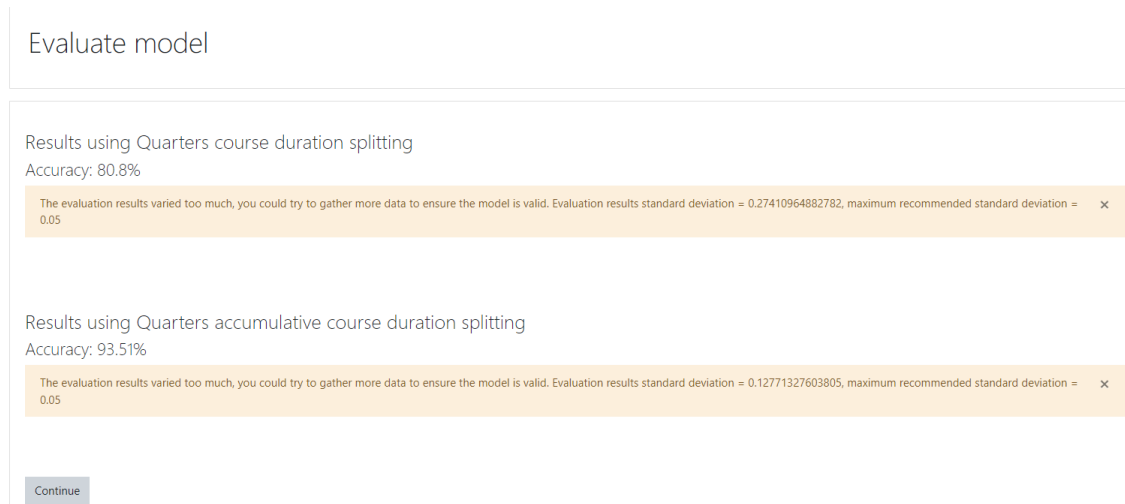
Forensik Digital		LO 01	LO 02	LO 03	[Deletion in progress] UTS	Tugas Pengganti UTS	Course total
First name / Surname	Email address						
Master Chief	chief@halo.com	90.00	0.00	92.00	-	-	182.00
Arthur Leywin	arhur@tbate.com	91.67	85.58	90.00	-	-	267.25
Lark Marcus	lark@magus.com	85.00	83.35	0.00	-	-	168.35
Overall average	Overall average	88.89	56.31	60.67	-	-	205.87

Gambar 5.3 *Grading* Forensik Digital pada tanggal 9 Februari 2023

Pada hari itu juga, tenaga pengajar melakukan pengecekan bagaimana tingkat keberhasilan mahasiswa pada *course* Forensik Digital. Untuk mempermudah pengecekan tersebut, tenaga

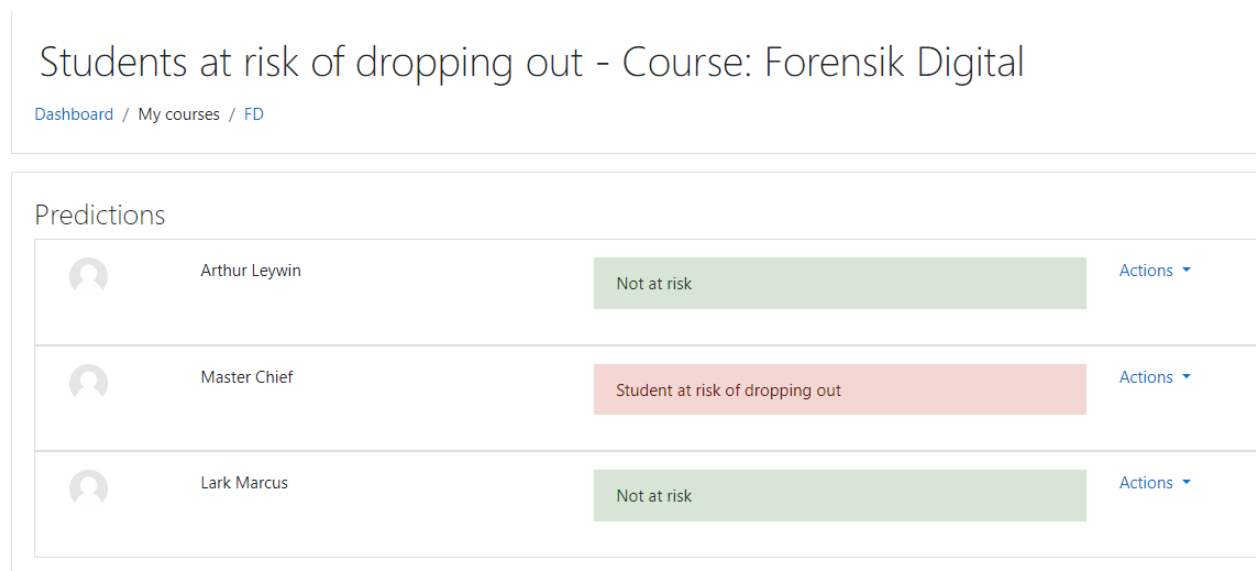


pengajar dapat meng-*install plug-in* yang dikembangkan pada C501. Setelah di-*install*, *plug-in* tersebut kemudian mencoba untuk membuat model data berdasarkan dengan *courses* lain yang sudah selesai. Semakin banyak data dari *course* lain, maka semakin tinggi persentase dari keakuratan model tersebut. Untuk hasil evaluasi model, dapat dilihat pada gambar 5.4



Gambar 5.4 Evaluasi model data pertama

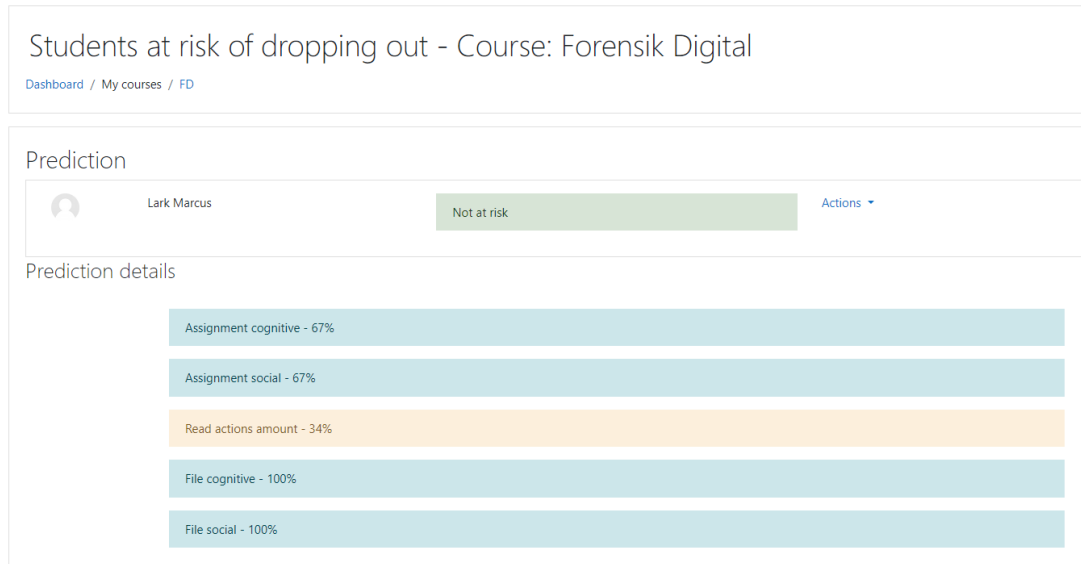
Di sini, tenaga pengajar menggunakan hasil *modelling* yang *quarter accumulative*, yang berarti *modelling* menggunakan data yang dihasilkan oleh *course* dalam jangka waktu 4 bulan sejak dimulai-nya *course*. Yang berarti model data yang digunakan memiliki tingkat akurasi sebesar 93,51%. Setelah itu, model dapat dieksekusi sehingga menghasilkan prediksi pada *course* Forensik Digital. Hasil prediksi tersebut dapat dilihat pada gambar 5.5



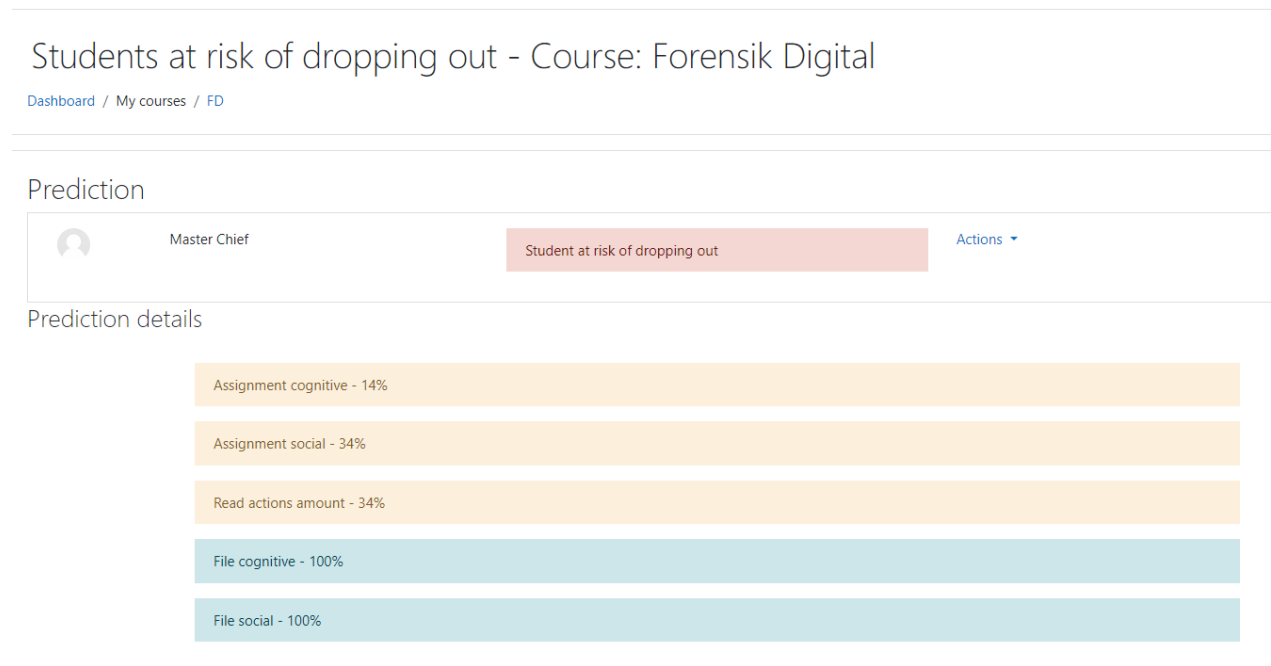
Gambar 5.5 Hasil prediksi *course* Forensik Digital



Dari hasil prediksi, dapat dikatakan bahwa 1 dari 3 peserta didik berisiko gagal atau memiliki nilai yang buruk pada *course* Forensik Digital. Tenaga pengajar ingin mengecek bagaimana hasil prediksi Master Chief dan Lark Marcus berbeda, padahal keduanya sama-sama satu kali tidak mengikuti LO. Sehingga langkah yang dilakukan adalah melihat detail prediksi yang terdapat pada tombol action. Dan detail tersebut dapat dilihat pada gambar 5.6 sampai dengan gambar 5.8



Gambar 5.6 Prediksi Lark Marcus secara detail



Gambar 5.7 Prediksi Master Chief secara detail



## Prediction

 Arthur Leywin

Not at risk


Actions ▾


## Prediction details


Assignment cognitive - 80%
Assignment social - 100%
Read actions amount - 34%
File cognitive - 100%
File social - 100%


Gambar 5.8 Prediksi Arthur Leywin secara detail

Dari kedua gambar di atas, dapat disimpulkan bahwa Lark Marcus lebih rajin dalam membuka materi dan tugas yang ada di Moodle dibandingkan dengan Master Chief. Sehingga sistem menyimpulkan bahwa Lark Marcus tidak berisiko, sedangkan Master Chief berisiko. Tombol action di sebelah kanan, selain dapat melihat detail prediksi, juga bisa digunakan sebagai jalan pintas untuk melihat laporan garis besar dari peserta didik maupun memberi pesan kepada peserta didik. Sehingga memudahkan tenaga pengajar untuk memberikan motivasi yang sesuai dengan keadaan peserta didik. Seperti yang digambarkan pada gambar 5.9

 Master Chief  
Offline

 Admin User

 Group (0)

 Private (1)

9 February

halo master chief, apa kabarnya

Write a message...

Gambar 5.9 Contoh komunikasi pada peserta didik



### 5.1.2 Skenario Pengujian 2 dan Analisis

Pengujian kedua yang penulis lakukan bertujuan untuk membuktikan apakah dengan bertambahnya *dataset* yang dimiliki, maka tingkat akurasi yang dihasilkan akan meningkat. Untuk *testing* kedua, penulis juga membuat sebuah *dummy course*. *Course* yang bernama “Testing 2” ini dimulai pada tanggal 1 Januari 2023 dan berakhir pada tanggal 1 April 2023. *Course* ini memiliki 3 peserta didik *dummy*, yakni Geralt Of Rivia, Cloud Strife, dan Venom Snake dan satu orang tenaga pengajar. Alasan *course* ini dan sebelumnya hanya memiliki 3 peserta didik adalah karena ketiga *dummy* tersebut sudah dapat merepresentasikan garis besar peserta didik. Dimulai dari peserta didik yang sangat rajin, yang agak rajin, dan yang kurang rajin.

#### Testing 2

[Dashboard](#) / [My courses](#) / [T 2](#) / [Participants](#)

#### Participants



Enrol users

Match

Any

Select

+ Add condition

Clear filters

Apply filters

4 participants found

First name

All

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Surname

All

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U





















V

W

X

Y

Z

	First name / Surname	Email address	Roles	Groups	Last access to course	Status
<input type="checkbox"/>						
<input type="checkbox"/>	 Geralt Of Rivia	geralt@witcher.com	Student 	No groups	14 days 11 hours	Active   
<input type="checkbox"/>	 Venom Snake	snake@mgs.com	Student 	No groups	14 days 11 hours	Active   
<input type="checkbox"/>	 Cloud Strife	cloud@ff7.com	Student 	No groups	14 days 11 hours	Active   
<input type="checkbox"/>	 Admin User	admin@noreply.com	Teacher 	No groups	14 days 11 hours	Active   

Gambar 5.10 Partisipan *course* Testing 2

Di dalam *course* Testing 2, terdapat 3 metode penilaian yang digunakan, yakni tugas, *quiz*, dan UAS. Tugas di *course* ini ada 4 dan masing-masing tugas memiliki bobot penilaian 7,5%. Untuk *quiz*, *course* ini mengadakan sebanyak 2 kali dengan masing-masing bobot penilaian sebesar 15%. Dan untuk sisa penilaian atau sebesar 40% disumbangkan oleh UAS. Untuk lebih jelas mengenai bobot penilaian, dapat dilihat pada gambar 5.11



Name	Weights	Max grade	Actions
Testing 2	-	-	Edit
Tugas 1	7.5	100.00	Edit
Tugas 2	7.5	100.00	Edit
Quiz 1	15.0	100.00	Edit
Tugas 3	7.5	100.00	Edit
Tugas 4	7.5	100.00	Edit
Quiz 2	15.0	100.00	Edit
UAS	40.0	100.00	Edit
Course total		700.00	Edit

Gambar 5.11 Bobot penilaian *course* Testing 2

Per tanggal 14 Februari 2023, *course* Testing 2 telah melaksanakan 3 buah penilaian, yakni penilaian untuk Tugas 1, Tugas 2, dan Quiz 3. Semua peserta didik telah berhasil menyelesaikan penilaian tersebut sesuai dengan waktu yang telah ditentukan. Masing-masing nilai yang dihasilkan peserta didik dapat dilihat pada gambar 5.12

Grader report										
View Setup Scales Letters Import Export										
Grader report Grade history Outcomes report Overview report Single view User report										
All participants: 3/3										
First name All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z										
Surname All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z										
Testing 2										
First name / Surname	Email address	Tugas 1	Tugas 2	Quiz 1	Tugas 3	Tugas 4	Quiz 2	UAS	Course total	
Geralt Of Rivia	geralt@witcher.com	82.00	92.00	95.00	-	-	-	-	273.00	
Venom Snake	snake@mgsv.com	86.00	90.00	62.00	-	-	-	-	225.00	
Cloud Strife	cloud@ff7.com	72.00	85.00	92.00	-	-	-	-	255.75	
Overall average	Overall average	80.00	89.00	83.00	-	-	-	-	251.25	

Gambar 5.12 *Grading* peserta didik *course* Testing 2

Pada tanggal itu juga tenaga pengajar berencana melakukan penilaian mengenai tingkat keberhasilan peserta didik. Oleh karena itu, tenaga pengajar menggunakan *plug-in learning analytics*. Hal pertama yang dilakukan oleh tenaga pengajar adalah melakukan *training data*. Dibandingkan pengujian sebelumnya yang hanya menggunakan dataset dari 1 *course*, pengujian kali ini menggunakan dataset dari 2 *course* karena *course* pada pengujian sebelumnya sudah selesai sehingga dapat dijadikan sebagai dataset. Berikut merupakan hasil *modelling* tersebut.



## Evaluate model

Results using Quarters course duration splitting

Accuracy: 86.4%

The evaluation results varied too much, you could try to gather more data to ensure the model is valid. Evaluation results standard deviation = 0.18675704124819, maximum recommended standard deviation = 0.05

Results using Quarters accumulative course duration splitting

Accuracy: 95.2%

The evaluation results varied too much, you could try to gather more data to ensure the model is valid. Evaluation results standard deviation = 0.11268634711244, maximum recommended standard deviation = 0.05

Gambar 5.13 Evaluasi model data kedua

Berdasarkan gambar 5.13 didapatkan tingkat akurasi untuk metode *quarter* adalah 86,4% dan untuk metode *quarter accumulative* adalah 95,2%. Dikarenakan metode *quarter accumulative* memulai perhitungan 4 bulan sejak *course* dimulai dibandingkan metode *quarter* yang memulai perhitungan 4 bulan sejak awal tahun, maka metode *quarter accumulative* memiliki tingkat akurasi yang lebih tinggi. Selain itu, akurasi yang didapatkan pada pengujian kali ini lebih tinggi dibandingkan dengan tingkat akurasi pada pengujian sebelumnya. Hal tersebut membuktikan bahwa semakin banyak dataset, maka tingkat akurasi yang dihasilkan menjadi semakin tinggi. Setelah didapatkan model data yang diinginkan, model data tersebut dapat dieksekusi dengan data yang tersedia pada *course* Testing 2. Hasil prediksi tersebut dapat dilihat pada gambar 5.14.

## Students at risk of dropping out - Course: Testing 2

[Dashboard](#) / [My courses](#) / [T 2](#)

### Predictions

	Geralt Of Rivia	Not at risk	Actions ▾
	Venom Snake	Not at risk	Actions ▾
	Cloud Strife	Not at risk	Actions ▾

Gambar 5.14 Hasil prediksi peserta didik *course* Testing 2



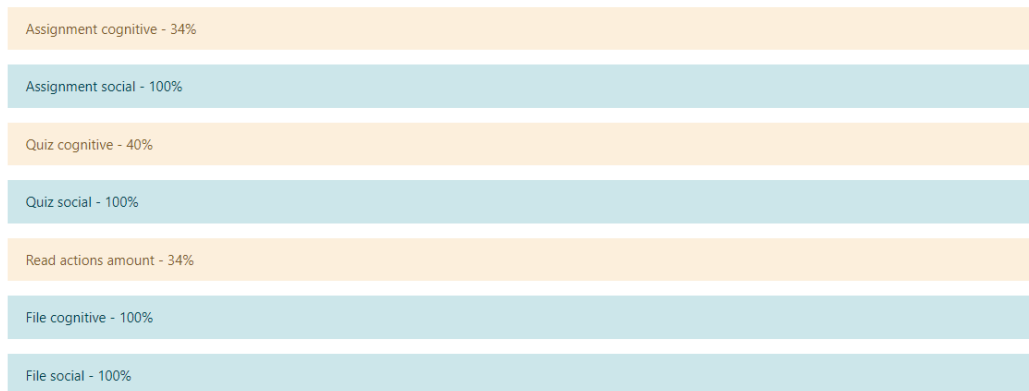


Dari gambar 5.14, dapat dikatakan bahwa ketiga peserta didik yang mengikuti *course* tidak berisiko gagal ketika mengikuti *course*. Untuk lebih jelas mengenai detail tiap peserta didik, dapat dilihat pada gambar 5.15 sampai dengan gambar 5.16.

#### Prediction

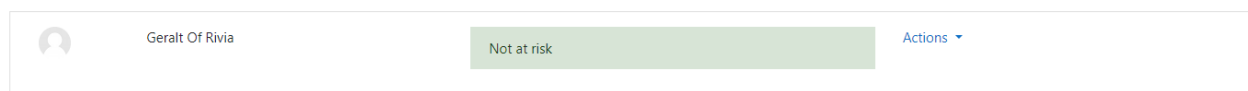


#### Prediction details

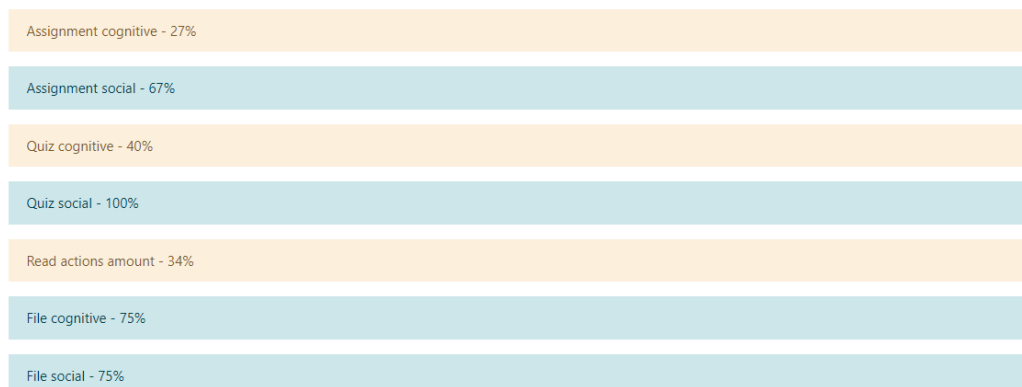


Gambar 5.15 Detail prediksi Venom Snake pada *course* Testing 2

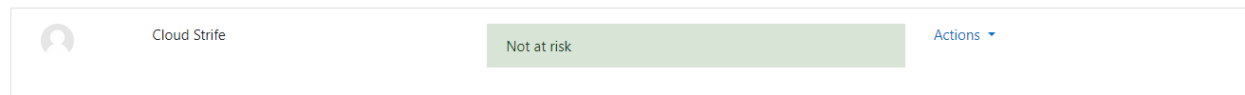
#### Prediction



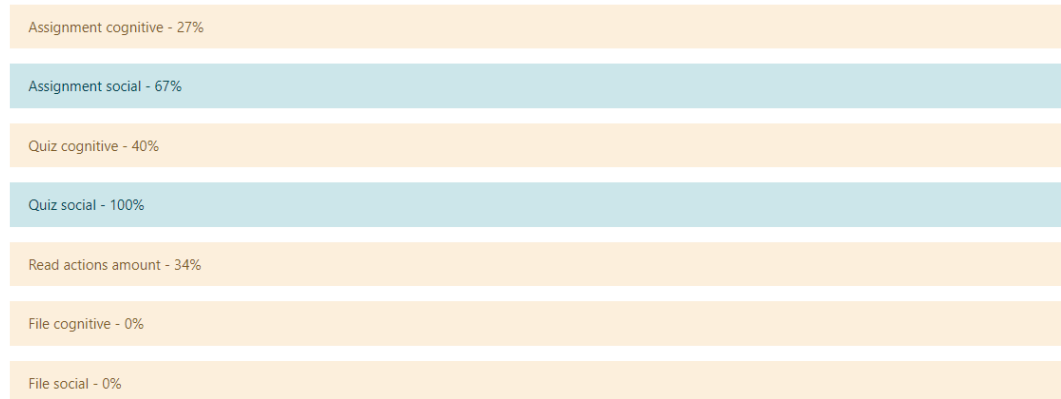
#### Prediction details



Gambar 5.16 Detail prediksi Geralt Of Rivia pada *course* Testing 2



Prediction details



Gambar 5.17 Detail prediksi Cloud Strife pada course Testing 2

Dari ketiga gambar tersebut dapat ditarik beberapa kesimpulan. Yang pertama, Venom Snake termasuk ke dalam peserta didik yang rajin karena dia tercatat sudah mengerjakan semua tugas dan *quiz*, serta sudah mengunduh semua materi yang sudah disediakan sampai akhir *course*. Yang kedua, Geralt Of Rivia juga sama dengan Venom Snake, hanya saja materi yang sudah diunduh belum sebanyak Venom Snake. Yang terakhir, Cloud Strife terlihat sudah mengerjakan semua tugas dan *quiz*, namun belum pernah melihat materi yang diberikan pada *course*. Akan tetapi, karena masih awal pembelajaran, sistem tidak menganggap Cloud Strife berisiko gagal dalam *course*. Meskipun demikian, peran tenaga pengajar dalam memotivasi peserta didik, terutama Cloud Strife masih sangat dibutuhkan agar prediksi yang dihasilkan juga tidak melenceng.

## 5.2 Improvement

*Improvement* yang dapat dilakukan pada proyek Capstone ini adalah dari segi jumlah data dan jenis data. Dataset yang digunakan pada proyek ini masih terlalu terbatas dan berdasarkan satu dan dua *course* saja sehingga model yang terbentuk masih tidak dapat digunakan secara lebih luas. Selain itu, kekurangan dataset juga menjadi limitasi pada beberapa algoritma seperti algoritma *deep learning*. Cara paling mudah untuk meningkatkan dataset yang digunakan adalah dengan semakin lama *plug-in* ini digunakan. Jika semakin lama *plug-in* ini digunakan, maka semakin banyak dataset yang dimiliki mengingat bahwa semakin banyak juga *course* yang sudah selesai sehingga dapat digunakan sebagai dataset. Jika dataset yang sudah digunakan banyak, maka dapat



dipastikan bahwa hasil prediksi yang dihasilkan menjadi semakin akurat karena *machine learning* sudah menjadi lebih ahli dalam melihat pola-pola yang dihasilkan peserta didik dalam suatu *course*.

Selain itu, hasil prediksi yang dihasilkan juga tidak maksimal karena data yang diambil hanya pada Moodle. Sedangkan para peserta didik ketika belajar tidak wajib menggunakan Moodle sehingga waktu belajar mereka tidak sepenuhnya terekam. Untuk mengatasi hal tersebut dapat pula dilakukan *tracking* pada aplikasi yang digunakan untuk membuka dokumen atau melihat tayangan video. Akan tetapi, hal tersebut sangatlah mengganggu privasi peserta didik. Peserta didik mungkin merasa selalu diawasi setiap saat. Belum lagi dengan adanya risiko kebocoran data.

Terakhir, *plug-in* yang penulis kembangkan hanyalah sebuah alat. Sehingga untuk meningkatkan performa pembelajaran tidak cukup hanya menggunakan *plug-in* ini. Perlu adanya kerja sama antara tenaga pengajar dan peserta didik. Peserta didik selalu berusaha semaksimal mungkin dalam mengikuti mata kuliah dan tenaga pengajar membimbing peserta didik dengan tekun, terutama kepada peserta didik yang mengalami masalah.

## BAB 6

### ANALISIS MENGENAI PENGARUH SOLUSI *ENGINEERING DESIGN*

Berdasarkan hasil dari proyek *capstone* yang dilakukan penulis, penulis dapat menganalisis dampak apa saja yang akan diperoleh apabila proyek *capstone* ini sudah diimplementasikan secara luas. Dampak tersebut dapat dikelompokkan menjadi 4 konteks, yakni konteks global, konteks ekonomis, konteks lingkungan, dan konteks sosial.

#### 6.1 Konteks Global

Salah satu kekurangan dari pembelajaran *online* dibandingkan dengan pembelajaran *offline* adalah interaksi antara tenaga pengajar dan peserta didik yang lebih sedikit. Tenaga pengajar tidak dapat memantau secara langsung mengenai *progress* dari peserta didik sehingga penilaian kualitatif menjadi salah satu hambatan ketika pembelajaran *online*. Akan tetapi, proyek *capstone* yang sudah dilakukan oleh penulis membuktikan bahwa penilaian kualitatif dapat dilakukan kepada peserta didik pada lingkungan LMS atau Learning Management System. Beberapa penilaian kualitatif tersebut seperti seberapa sering peserta didik mengakses materi dalam *course*, maupun seberapa banyak tingkat pemahaman peserta didik dalam suatu *course*.

Meski penggunaan pembelajaran *online* yang semakin menurun dikarenakan pandemi Covid-19 yang sudah membaik, namun pembelajaran *online* masih menjadi salah satu pilihan metode pendidikan yang masih dipilih. Dan dengan adanya *plug-in* yang penulis kembangkan, maka keberhasilan pembelajaran *online* juga akan semakin meningkat karena penilaian tidak hanya dapat dilakukan ketika ujian atau saat tugas saja, namun juga *progress* peserta didik juga dapat dinilai. Dan dalam dunia pendidikan nilai memang dianggap penting, tetapi proses untuk mencapai nilai tersebut seharusnya juga sama atau bahkan lebih penting dari nilai tersebut.

#### 6.2 Konteks Ekonomi

*Plug-in* yang penulis kembangkan dapat digunakan untuk menilai bagaimana kesiapan peserta didik dalam suatu pembelajaran. Secara konvensional, menilai bagaimana kesiapan peserta didik dalam pembelajaran membutuhkan waktu, tenaga, dan biaya yang tidak sedikit. Akan tetapi dengan adanya *plug-in* yang penulis kembangkan waktu, tenaga, dan biaya yang diperlukan untuk melakukan analisis bagaimana tingkat kesiapan peserta didik dapat dipangkas sehingga sumber daya tersebut dapat dialokasikan ke hal lain yang lebih penting, seperti pembangunan gedung,



memperbaiki infrastruktur, atau penambahan sarana dan prasarana. Selain itu, dengan menggunakan *plug-in* ini, peserta tidak diperlukan untuk melakukan presensi manual ketika pembelajaran *online* karena terdapat log yang menyimpan kegiatan peserta didik dalam LMS sehingga mengurangi beban administrasi yang harus mencatat kehadiran setiap peserta didik dalam suatu pembelajaran *online*.

### **6.3 Konteks Lingkungan**

Dengan adanya *plug-in* yang dikembangkan oleh penulis, membuat pembelajaran *online* menjadi lebih dinamis karena bertambahnya interaksi antara peserta didik dan tenaga pengajar. Sehingga pembelajaran *online* untuk kedepannya masih dapat dilakukan dengan meminimalisir perbedaan dengan pembelajaran *offline*. Dengan semakin banyak pembelajaran *online* yang dilakukan, maka kebutuhan gedung sekolah, universitas, maupun gedung untuk tempat belajar yang lain dapat berkurang sehingga lahan-lahan yang tersedia dapat dialokasikan untuk ruang hijau. Selain itu, pembelajaran *online* juga memangkas transportasi yang dibutuhkan untuk melakukan kegiatan belajar mengajar sehingga polusi-polusi yang disebabkan oleh kendaraan dapat berkurang.

### **6.4 Konteks Sosial**

*Plug-in* yang penulis kembangkan dapat meningkatkan interaksi antara peserta didik dengan tenaga pengajar dalam suatu pembelajaran *online*. Dengan menggunakan *plug-in* ini, tenaga pengajar tidak hanya memberikan penjelasan tentang materi, memberikan tugas dan ujian lalu menilainya, namun juga dapat menilai tentang *progress* peserta didik dalam suatu pembelajaran *online*. Tenaga pengajar dapat memberikan perhatian lebih kepada peserta didik yang menurut *plug-in* mengalami permasalahan dalam mengikuti pembelajaran *online* sehingga meningkatkan interaksi dan kedekatan antara peserta didik dan tenaga pengajar.

## BAB 7

### KESIMPULAN DAN SARAN

#### 7.1 Kesimpulan

Tema yang diangkat oleh penulis pada proyek *capstone* ini adalah pengembangan prototipe *plug-in learning analytics* yang digunakan untuk memprediksi hasil akhir dari mahasiswa pada suatu LMS, yakni Moodle. *Plug-in* yang penulis kembangkan sudah dapat memprediksi hal tersebut dengan tingkat akurasi 93.51% apabila hanya menggunakan dataset berasal dari Transformasi Digital semester 1 tahun 2021 dan 95,3% apabila datasetnya ditambah dengan data *dummy* pengujian 1 .

#### 7.2 Saran

Prototipe *plug-in* yang penulis kembangkan sudah dapat berjalan baik fungsi dan fiturnya. Oleh karena itu, untuk ke depannya *plug-in* tersebut dapat dikembangkan lebih lanjut agar dapat benar-benar terpasang pada LMS yang sebenarnya, seperti eLOK. Selain itu, tingkat akurasi dari *plug-in* yang penulis kembangkan dalam proyek *capstone* ini dapat ditingkatkan dengan berbagai cara, antara lain :

- memperbanyak jumlah dataset
- memperbanyak jumlah *course* dalam LMS
- memperbanyak jam terbang *plug-in*
- mengintegrasikan LMS dengan aplikasi yang digunakan dalam pembelajaran
- melakukan eksplorasi teknik optimasi lebih dalam.

## REFERENSI

- [1] V. Esichaikul, "Student Modelling in Adaptive E-Learning Systems," in *Knowledge Management & E-Learning : An International Journal*, Vol 3, No 3, Bangkok, Thailand. pp. 342-355
- [2] G. Vaseko, "About Moodle," 31 August 2020. [Online]. Available: [https://docs.moodle.org/311/en/About\\_Moodle](https://docs.moodle.org/311/en/About_Moodle). [Diakses 14 November 2021].
- [3] A. Faradilla, "Apa Itu Plugin? Pengertian dan Fungsi Plugin di WordPress," 19 August 2021. [Online]. Available: <https://idcloudhost.com/mengenal-apa-itu-plugin-in-defenisi-keunggulan-dan-cara-kerja/>. [Diakses 14 November 2021].
- [4] S. Martin, "Machine Learning backends," 30 June 2020. [Online]. Available: [https://docs.moodle.org/dev/Machine\\_learning\\_backends](https://docs.moodle.org/dev/Machine_learning_backends). [Diakses 15 November 2021].
- [5] O. Elgabry, "Plug-in Architecture and the story of the data pipeline framework," 1 May 2019. [Online]. Available: <https://medium.com/omarelgabrys-blog/plugin-architecture-dec207291800>. [Diakses 15 November 2021].
- [6] "Advantages of the Plug-in Architecture – Microstrategy," Microstrategy, 2021. [Online]. Available: [https://www2.microstrategy.com/producthelp/Current/WebSDK/Content/topics/webarch/Advantages\\_of\\_the\\_Plug-in\\_Architecture.htm](https://www2.microstrategy.com/producthelp/Current/WebSDK/Content/topics/webarch/Advantages_of_the_Plug-in_Architecture.htm). [Diakses 19 November 2021].
- [7] "PHP | Introduction," GeeksforGeeks, 2021. [Online]. Available: <https://www.geeksforgeeks.org/php-introduction/>. [Diakses 16 November 2021].
- [8] "Python Programming Language," GeeksforGeeks, 2021. [Online]. Available: <https://www.geeksforgeeks.org/python-programming-language/?ref=shm>. [Diakses 16 November 2021].
- [9] "Waterfall Methodology," Workfront, 2021. [Online]. Available: <https://www.workfront.com/en-gb/project-management/methodologies/waterfall>. [Diakses 18 November 2021].
- [10] "Software Development Life Cycle," javapoint. [Online]. Available: <https://www.javatpoint.com/software-engineering-software-development-life-cycle>. [Diakses 20 November 2021]
- [11] "Agile Guide," Hygger, 2021. Available: <https://hygger.io/guides/agile/> [Diakses 18 November 2021]
- [12] "Intermediate Algebra Tutorial : Introduction to Problem Solving," West Texas A&M University, 2011. [Online]. Available:



- [https://www.wtamu.edu/academic/anns/mps/math/mathlab/int\\_algebra/int\\_alg\\_tut8\\_probsol.htm](https://www.wtamu.edu/academic/anns/mps/math/mathlab/int_algebra/int_alg_tut8_probsol.htm) [Diakses 22 November 2021]
- [13] D. Landa-Silva. 2006. "Large Scale Systems Design," [Online]. Available: <http://www.cs.nott.ac.uk/~pszjds/teaching/archive/G52LSSLecture2.pdf>
- [14] N. Tomasevic, "An Overviwe and Comparison of Supervised Data Mining Techniques for Student Exam Performance Prediction," in *Computer & Education*, Belgrade, Serbia, 2019.
- [15] A. N. Fatah, "Pengembangan Plugin Penjadwalan Seminar Pada Web Jurusan Ilmu Komputer Berbasis Wordpress," in *Jurnal Komputasi*, vol. 4, no. 1, Lampung, Indonesia, 2016. pp. 51-64.
- [16] D. Kurniawan, "Apa Itu Framework? Yuk Kenali Penertian dan Fungsinya!," Niagahoster, 13 May 2020. [Online]. Available: <https://www.niagahoster.co.id/blog/apa-itu-framework/>
- [17] M. Garg, "Top 11 Python Frameworks for Web Development In 2023," Net Solutions, 4 January 2023. [Online]. Available : <https://www.netsolutions.com/insights/top-10-python-frameworks-for-web-development-in-2019/>
- [18] Ariata C., "8 Framework PHP Terbaik Untuk Developer," Hostinger, 18 January 2023. [Online] . Available : <https://www.hostinger.co.id/tutorial/framework-php>
- [19] Y.S. Tsai, "What Is Learning Analytics?" SoLAR, 2017. [Online]. Available : <https://www.solaresearch.org/about/what-is-learning-analytics/>
- [20] K. Miller, "What Is Learning Analytics & How Can It Be Used?," Northeastern University Graduate Programs, 18 February 2020. [Online]. Available : <https://www.northeastern.edu/graduate/blog/learning-analytics/>
- [21] A. Nicols, "Analytics API," Moodle, 11 June 2022. [Online]. Available : <https://moodledev.io/docs/apis/subsystems/analytics>
- [22] K. Brush, "learning management system (LMS)," TechTarget, December 2019. [Online]. Available : <https://www.techtarget.com/searchcio/definition/learning-management-system>