PENERAPAN XAI UNTUK TRANSPARANSI PENILAIAN PEMBELAJARAN PADA MATA KULIAH KEAMANAN JARINGAN BERBASIS MOODLE

TUGAS AKHIR

Diajukan oleh:

NABILA NURIL AULIA

NIM. 220705010

Mahasiswa Fakultas Sains Dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI AR-RANIRY

BANDA ACEH

2026 M / 1448 H

LEMBAR PERSETUJUAN

PENERAPAN XAI UNTUK TRANSPARANSI PENILAIAN PEMBELAJARAN PADA MATA KULIAH KEAMANAN JARINGAN BERBASIS MOODLE

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu Teknologi Informasi

Oleh:
NABILA NURIL AULIA
NIM. 220705010

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Khairan AR, M.Kom	**Mulkan Fadhli, M.T**
NIP. 198607042014031001	**NIP. 198811282020121006**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Malahayati, M.T
NIP. 198301272015032003

LEMBAR PENGESAHAN

PENERAPAN XAI UNTUK TRANSPARANSI PENILAIAN PEMBELAJARAN PADA MATA KULIAH KEAMANAN JARINGAN BERBASIS MOODLE

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Senin, 27 Juli 2026 M
12 Safar 1448 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir

Ketua,

Khairan AR, M.Kom
NIP. 198607042014031001

Sekretaris,

Maikan Fadhli, M.T
NIP. 198811282020121006

Penguji I,

Aulia Syarif Aziz, M.Sc
NIP. 199305212022031001

Penguji II,

Malahayati, M.T
NIP. 198301272015032003

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Nabila Nuril Aulia
Nim	:	220705010
Prodi	:	Teknologi Informasi
Fakultas	:	Sains dan Teknologi
Judul	:	Penerapan XAI Untuk Tranparansi Penilaian Pembelajaran Pada Mata Kuliah Keamanan Jaringan Berbasis Moodle

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 27 Juli 2026

Yang Menyatakan

(Nabila Nuril Aulia)

ABSTRAK

Nama	:	Nabila Nuril Aulia
NIM	:	220705010
Program Studi	:	Teknologi Informasi
Judul	:	Penerapan XAI untuk Transparansi Penilalaian Pembelajaran Pada Mata Kuliah Keamanan Jaringan Berbasis Moodle
Tanggal Sidang	:	27 Juli 2026
Jumlah Halaman	:	84 Halaman
Pembimbing I	:	Khairan AR, M.Kom
Pembimbing II	:	Mulkan Fadhli, M.T
Kata Kunci	:	Explainable AI, Moodle, Educational Data Mining, XGBoost, TreeSHAP, Evaluasi Formatif.

Perkembangan teknologi informasi telah mendorong adopsi *Learning Management System* (LMS) seperti Moodle yang menghasilkan rekam jejak digital mahasiswa secara mendetail. Namun, mayoritas institusi pendidikan masih terpaku pada paradigma evaluasi sumatif tradisional yang bersifat *black-box* dan hanya berfokus pada hasil akhir berupa nilai ujian. Ketidakmampuan sistem ini dalam mengidentifikasi proses pengerjaan (seperti strategi, durasi waktu, dan kecepatan klik) menyebabkan mahasiswa kebingungan dan kehilangan ruang refleksi untuk memperbaiki proses belajar mereka. Untuk mengatasi tantangan tersebut, penelitian ini menerapkan pendekatan *Explainable Artificial Intelligence* (XAI) guna memberikan transparansi penilaian pembelajaran pada mata kuliah Keamanan Jaringan. Penelitian ini mengembangkan pemodelan analitik menggunakan algoritma *eXtreme Gradient Boosting* (XGBoost) untuk memetakan metrik perilaku mahasiswa dengan capaian nilai akhir, yang kemudian dibedah logika internalnya menggunakan metode TreeSHAP. Hasil ekstraksi matematis ini selanjutnya diintegrasikan ke dalam antarmuka berbasis Gradio untuk menghasilkan narasi umpan balik otomatis (*AI Companion*). Hasil evaluasi menunjukkan bahwa pemodelan XGBoost memiliki tingkat kecocokan yang baik dengan batas galat *Mean Absolute Error* (MAE) sebesar 6,86 poin dan *Root Mean Squared Error* (RMSE) sebesar 9,44 poin. Melalui metode TreeSHAP, sistem secara akurat berhasil mendeteksi bahwa durasi pengerjaan dan kecepatan klik merupakan faktor yang paling dominan, di mana kecepatan interaksi ekstrem mampu diidentifikasi sebagai taktik terburu-buru (*rapid-guessing*). Selain itu, matriks pengujian fungsionalitas membuktikan bahwa prototipe *AI Companion* berhasil menerjemahkan angka log yang kaku menjadi umpan balik teks berbahasa Indonesia yang utuh dan koheren. Dengan tingkat transparansi dan fungsionalitas yang tinggi, penerapan kerangka kerja XAI ini terbukti andal dalam menggeser paradigma evaluasi Moodle dari sekadar platform penguji yang deterministik menjadi ekosistem pendamping yang suportif, sehingga mendorong kemandirian evaluasi diri dan perbaikan strategi akademik mahasiswa di masa depan.

ABSTRACT

Name	:	Nabila Nuril Aulia
Student Number	:	220705010
Depertement	:	Information Technology
Title	:	Application of XAI for Learning Assessment Transparency in Moodle-Based Network Security Course
Date	:	27 Juli 2026
Number Of Pages	:	84 Pages
Supervisor I	:	Khairan AR, M.Kom
Supervisor II	:	Mulkan Fadhli, M.T
Keywords	:	Explainable AI, Moodle, Educational Data Mining, XGBoost, TreeSHAP, Formative Evaluation.

The development of information technology has driven the adoption of Learning Management Systems (LMS) such as Moodle, which generate detailed digital footprints of students. However, the majority of educational institutions still rely on traditional summative evaluation paradigms that act as a black-box, focusing merely on final exam scores. The system's inability to identify the learning process (such as test-taking strategies, duration, and click speed) leaves students confused and deprives them of the opportunity to reflect on and improve their learning strategies. To overcome these challenges, this study applies an Explainable Artificial Intelligence (XAI) approach to provide learning assessment transparency in the Network Security course. This research develops an analytical model using the eXtreme Gradient Boosting (XGBoost) algorithm to map students' behavioral metrics against their final grades, which is subsequently interpreted using the TreeSHAP method. The mathematical extraction is then integrated into a Gradio-based interface to generate automated narrative feedback (AI Companion). The evaluation results show that the XGBoost model has a good fitness level, achieving a Mean Absolute Error (MAE) of 6.86 points and a Root Mean Squared Error (RMSE) of 9.44 points. Through the TreeSHAP method, the system accurately detected that task duration and click speed are the most dominant factors, successfully identifying extreme interaction speed as a rapid-guessing tactic. Furthermore, the functional testing matrix proved that the AI Companion prototype successfully translated rigid log numbers into coherent and structured Indonesian text feedback. With a high level of transparency and functionality, the application of this XAI framework is proven reliable in shifting the Moodle evaluation paradigm from a mere deterministic testing platform to a supportive companion ecosystem, thereby encouraging students' independent self-evaluation and improving their future academic strategies.

KATA PENGANTAR

Bismillahirrahmanirrahim

Segala puji dan syukur penulis panjatkan atas kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, tidak lupa pula shalawat beserta salam kita sanjung sajikan kepada baginda nabi kita Muhammad SAW, yang telah membawa kita dari alam kebodohan menuju alam yang berilmu pengetahuan seperti yang kita rasakan pada saat ini. Sehingga penulis dapat menyelesaikan Tugas akhir yang berjudul **"Penerapan XAI Untuk Transparansi Penilaian Pembelajaran Pada Mata Kuliah Keamanan Jaringan Berbasis Moodle"**. Tugas akhir ini disusun untuk memenuhi salah satu syarat menyelesaikan Pendidikan Strata Satu (S-1) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh. Dalam penulisan Tugas akhir ini, penulis ingin mengucapkan terimakasih atas semua bantuan yang telah banyak diberikan kepada penulis baik secara langsung maupun tidak langsung, terutama yang terhormat:

1. Kepada cinta pertama penulis dan pintu surga penulis, Papa (Almarhum) Agus Nadi dan Mamak Reni Sartika. Kepada Papa, meskipun Allah SWT memanggil papa ketika penulis masih berusia satu setengah tahun sehingga penulis tidak memiliki banyak kenangan bersama papa, sosok papa tidak pernah hilang dari doa dan kerinduan penulis. Semoga Allah SWT melapangkan kubur papa, menerangi alam kuburnya, mengampuni segala dosa-dosanya, serta menempatkan papa di surga Firdaus. Dan kepada Mamak tercinta, penulis mengucapkan terima kasih yang tidak akan pernah cukup untuk menggambarkan seluruh pengorbanan mamak. Sejak kepergian papa, mamak dengan penuh keteguhan menjalankan dua peran sekaligus, sebagai ibu sekaligus ayah bagi penulis dan saudara-saudara penulis. Di tengah berbagai coabaan mamak tidak pernah menyerah untuk membesarkan, mendidik, dan memperjuangkan masa depan anak-anaknya. Seluruh doa, air mata, kerja keras, dan pengorbanan mamak menjadi alasan

terbesar penulis untuk terus berjuang hingga mampu menyelesaikan pendidikan ini. Skripsi ini penulis persembahkan sebagai salah satu wujud rasa syukur dan bakti kepada mamak, meskipun penulis menyadari bahwa apa yang telah diberikan tidak akan pernah mampu membalas seluruh perjuangan dan kasih sayang mamak. Semoga Allah SWT senantiasa memberikan kesehatan, umur yang penuh keberkahan, kebahagiaan, kemudahan dalam setiap urusan, serta menjadikan mamak termasuk hamba-hamba-Nya yang kelak memperoleh surga.

2. Kepada keempat saudara kandung penulis tercinta, yaitu abang Aidil Fajri, kakak Adelia Desti Indah Kurnia, abang Muhammad Syaukal Raamadhan, dan adik Malika Zahira. Penulis mengucapkan terima kasih yang sebesar-besarnya atas segala doa, kasih sayang, perhatian, dukungan, serta kebersamaan yang tidak pernah putus selama ini. Penulis menyadari bahwa keluarga kami bukanlah keluarga yang hidup dalam kelimpahan, namun di tengah segala keterbatasan yang kami hadapi, kami selalu berusaha saling menguatkan, saling merangkul, dan saling menjadi tempat bersandar satu sama lain. Tidak pernah ada yang merasa berjalan sendiri, karena setiap langkah selalu diiringi dengan doa, semangat, dan pengorbanan dari keluarga. Terima kasih karena selalu menjadi alasan bagi penulis untuk terus bangkit ketika menghadapi kesulitan. Semoga Allah SWT senantiasa menjaga ikatan persaudaraan kami, melimpahkan keberkahan, kesehatan, rezeki yang halal dan luas, serta mengangkat derajat keluarga kami sehingga kelak kami dapat bersama-sama membahagiakan mamak dan menjalani kehidupan yang lebih baik.
3. Kepada paman Riski Satria, selaku adik kandung mamak, penulis mengucapkan terima kasih yang sebesar-besarnya atas segala bantuan, kepedulian, dan pengorbanan yang telah diberikan, khususnya dalam membantu membiayai SPP penulis selama menempuh pendidikan di bangku perkuliahan. Bantuan tersebut menjadi salah satu bentuk dukungan yang sangat berarti sehingga penulis dapat menyelesaikan studi ini dengan baik. Semoga Allah SWT membalas setiap kebaikan paman dengan pahala

yang berlipat ganda, memberikan kesehatan, keberkahan rezeki, serta kemudahan dalam setiap langkah kehidupan.

4. Bapak Khairan AR, M.Kom., selaku dosen pembimbing I, penulis mengucapkan rasa hormat dan terima kasih yang sebesar-besarnya atas waktu, perhatian, kesabaran, bimbingan, serta ilmu yang telah diberikan selama proses penyusunan skripsi ini. Segala arahan, kritik, saran, dan motivasi yang bapak berikan menjadi bekal yang sangat berharga bagi penulis dalam menyelesaikan penelitian ini. Semoga Allah SWT senantiasa memberikan kesehatan, keberkahan, serta membalas seluruh kebaikan bapak dengan pahala yang berlipat ganda.
5. Bapak Mulkan Fadhli, M.T., selaku dosen pembimbing II, penulis mengucapkan rasa hormat dan terima kasih yang sebesar-besarnya atas segala bimbingan, ketelitian, kesabaran, serta ilmu yang telah bapak berikan selama proses penyusunan skripsi ini. Setiap masukan, koreksi, evaluasi, dan arahan yang bapak berikan telah membantu penulis untuk terus memperbaiki kualitas penelitian, baik dari sisi penulisan maupun implementasi penelitian. Semoga Allah SWT senantiasa memberikan kesehatan, keberkahan, dan membalas seluruh kebaikan bapak dengan pahala yang berlipat ganda.
6. Bapak Prof. Dr. Ir. M. Dirhamsyah, M.T., IPU., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memberikan kesempatan, fasilitas, serta dukungan kepada penulis selama menempuh pendidikan di Fakultas Sains dan Teknologi.
7. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memberikan arahan, bimbingan, dukungan, serta pelayanan akademik kepada penulis selama menempuh pendidikan. Terima kasih atas dedikasi dan perhatian yang diberikan sehingga penulis dapat menyelesaikan seluruh proses perkuliahan hingga penyusunan tugas akhir ini dengan baik. Semoga Allah SWT senantiasa melimpahkan kesehatan, keberkahan, dan membalas segala kebaikan yang telah ibu berikan.

8. Ibu Cut Ida Rahmadiana, S.Si., selaku Staf Program Studi Teknologi Informasi, yang dengan penuh kesabaran, keramahan, dan ketulusan senantiasa membantu penulis dalam berbagai proses administrasi akademik. Terima kasih atas segala bantuan, informasi, dan kesediaan ibu dalam memberikan pelayanan terbaik kepada penulis selama menempuh pendidikan.

9. Seluruh Bapak dan Ibu Dosen Program Studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh yang telah mendidik, membimbing, serta membekali penulis dengan ilmu pengetahuan, pengalaman, dan nilai-nilai kehidupan yang sangat bermanfaat selama menjalani masa perkuliahan. Semoga segala ilmu yang telah diberikan menjadi amal jariyah yang pahalanya terus mengalir.

10. Kepada keluarga besar Husaini dan Fazidah, penulis mengucapkan terima kasih atas segala doa, dukungan, dan bantuan yang telah diberikan kepada penulis, terutama pada masa-masa sulit selama menjalani pendidikan. Kehangatan keluarga menjadi salah satu kekuatan yang membantu penulis untuk tetap bertahan dan menyelesaikan pendidikan hingga akhir.

11. Kepada sahabat-sahabat terbaik penulis, yaitu Cut Aja, Kana, Fadhila, Zulfa, Rahmi, dan Mirna. Terima kasih telah menjadi bagian dari perjalanan hidup penulis sejak duduk di bangku Sekolah Dasar hingga saat ini. Persahabatan yang telah terjalin selama bertahun-tahun merupakan salah satu anugerah terindah yang Allah SWT berikan kepada penulis. Terima kasih atas doa, perhatian, dukungan, motivasi, kebersamaan, serta kasih sayang yang tidak pernah putus hingga saat ini. Kehadiran kalian selalu menghadirkan tawa, semangat, dan kekuatan bagi penulis dalam menghadapi berbagai tantangan kehidupan, termasuk selama proses penyusunan skripsi ini. Semoga Allah SWT senantiasa menjaga persahabatan kita, melimpahkan keberkahan, kesehatan, kebahagiaan, serta mengabulkan setiap doa dan cita-cita kita semua.

12. Kepada seluruh teman-teman angkatan 2022 Program Studi Teknologi Informasi, penulis mengucapkan terima kasih atas kebersamaan, kerja

sama, pengalaman, dan kenangan yang telah dilalui bersama sejak awal perkuliahan hingga penyelesaian tugas akhir ini. Semoga tali silaturahmi yang telah terjalin tetap terjaga dan semoga kita semua diberikan kemudahan serta kesuksesan dalam menggapai cita-cita di masa yang akan datang.

13. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan bantuan, doa, dukungan, motivasi, maupun kontribusi dalam berbagai bentuk selama proses penyusunan skripsi ini. Penulis mengucapkan terima kasih yang sebesar-besarnya. Semoga Allah SWT membalas setiap kebaikan yang telah diberikan dengan pahala yang berlipat ganda serta melimpahkan rahmat dan keberkahan kepada kita semua. Aamiin ya Rabbal 'Alamin.

Penulis menyadari sepenuhnya bahwa dalam penyusunan Tugas Akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima kritik dan saran untuk mencapai kesempurnaan dalam penulisan Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT. Amiin Ya Rabbal A'lamin.

Banda Aceh, 28 Juli 2026
Penulis,

Nabila Nuril Aulia

DAFTAR ISI

ABSTRAK v
ABSTRACT vi
KATA PENGANTAR i
DAFTAR ISI vi
DAFTAR GAMBAR ix
DAFTAR TABEL x
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Batasan Penelitian 4
1.5 Manfaat Penelitian 4
1.5.1 Manfaat Teoretis 4
1.5.2 Manfaat Praktis 4
BAB II TINJAUAN PUSTAKA 6
2.1 Penelitian Terdahulu 6
2.2 LMS Moodle dan Karakteristik Berkas Log 10
2.3 Eksplorasi Data Pendidikan (*Educational Data Mining*) 11
2.4 Kecerdasan Buatan/AI 12
2.5 Machine Learning 12
2.6 Python 13
2.7 Google Colaboratory 14
2.8 Ensemble Learning dan XGBoost 15
2.9 Explainable Artifical Intelligence (XAI) 16
2.10 Shapley Additive Explanations (SHAP) 18
2.11 Evaluasi dan Kinerja dan Interpretabilitas Model 21
BAB III METODE PENELITIAN 23
3.1 Tahapan penelitian 23
3.2 Pengumpulan Data Observasi Awal (Studi Pendahuluan) 25
3.3 Tahapan Pengumpulan Data (*Data Acquisition*) 26
3.4 Pra-pemrosesan Data (*Data Preprocessing*) 27
3.4.1 Filterisasi Modul dan Konteks Aktivitas 27
3.4.2 Konversi dan Sinkronisasi Zona Waktu 27

3.4.3 Penanganan Ketidaklengkapan Data (Missing Values Handling) 27
3.4.4 Agregasi Sesi Pengguna (*Session Aggregation*) 28
3.5 Ekstraksi Fitur (Feature Engineering) 28
3.5.1 Durasi Total Akses ($\boldsymbol{T}\Delta$) 28
3.5.2 Kecepatan klik (Clik Rate $\boldsymbol{Vc}$) 29
3.5.3 Penanda Waktu Larut Malam 31
3.5.4 Konsistensi Perangkat (*IP variance*) 31
3.5.5 Skor Historis Basis (Baseline Proficiency) 32
3.6 Metode Pembagian dan Augmentasi Data 33
3.6.1 Validasi Silang (*k-Fold Cross-Validation*) 33
3.6.2 Simulasi Varian dengan Injeksi Derau Gaussian Terisolasi 34
3.7 Pelatihan Model Kecerdasan buatan 35
3.8 Implementasi XAI (TreeSHAP) 37
3.9 Metode Pengujian dan Evaluasi 37
3.9.1 Mean Absolute Error (MAE) 38
3.9.2 Root Mean Square Eror (RMSE) 39
3.9.3 Koefisien Determinasi ($\boldsymbol{R2}$) 40
3.9.4 Estimasi Akurasi Prediksi (Estimated Accuracy) 41
3.10 Metode Pembuktian dan Luaran Sistem (*Proving Method*) 42
3.11 Lokasi dan Waktu Penelitian 44
3.12 Alat dan Bahan 44
3.12.1 Perangkat Keras 44
3.12.2 Perangkat Lunak 45
BAB IV HASIL DAN PEMBAHASAN 47
4.1 Karakteristik Populasi Data dan Profil Perilaku 47
4.1.1 Analisis Deskriptif Perilaku Evaluasi Mahasiswa 48
4.2 Kinerja Komputasi dan Analisis Korelatif Asosiatif 50
4.2.1 Konstruksi dan Evaluasi Model Prediktif 51
4.2.2 Uji Ablasi (*Ablation Study*) Atribut Skor Historis 53
4.2.3 Pembedahan Pola Korelasi Perilaku 54
4.3 Transparansi Penilaian melalui *Explainable AI* 55
4.3.1 Ekstraksi Nilai Atribusi TreeSHAP 55
4.3.2 Implementasi dan Evaluasi Prototipe AI Companion 56
4.4 Pengujian Interpretabilitas dan Koherensi Fungsional 60
4.4.1 Kestabilan Penjelasan (Stability dan Fidelity) 60

4.4.2 Validasi Rantai Eksekusi Sistem (*System Routing Validation*) 61
4.5 Pembahasan Akhir 62
BAB V KESIMPULAN DAN SARAN 65
5.1 Kesimpulan 65
5.2 Saran 66
DAFTAR PUSTAKA 67
LAMPIRAN 70

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Sequential XGBoost 16
Gambar 2. 2 Perbandingan Black Box dan XAI 17
Gambar 2. 3 Visualisasi SHAP 20
Gambar 3. 1 Tahapan Penelitian 24
Gambar 4. 1 Distribusi Waktu Inisiasi Kuis 50
Gambar 4. 2 Konfigurasi Hyperparameter Algoritma XGBoost Regressor 51
Gambar 4. 3 Pemetaan Scatter Plot Sebaran Nilai 52
Gambar 4. 4 Visualisasi Global Feature Importance XGBoost 54
Gambar 4. 5 Pemetaan SHAP Summary Plot Global Fitur Perilaku Moodle 55
Gambar 4. 6 Rule-Based Generation Berbasis Sentimen SHAP 57
Gambar 4. 7 Tangkapan Layar Dashboard Web Gradio AI Companion 58

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu 6
Tabel 3. 1 Perangkat Keras 44
Tabel 3. 2 Perangkat Lunak 45
Tabel 4. 1 Distribusi Populasi Dataset Akhir..... 47
Tabel 4. 2 Statistik Deskriptif Metrik Perilaku Kuis..... 48
Tabel 4. 3 Hasil Evaluasi Kinerja Kuantitatif (k-Fold Cross-Validation) 52
Tabel 4. 4 Komparasi Kinerja Algoritma..... 53
Tabel 4. 5 Matriks Komparasi Studi Kasus Implementasi Umpan Balik 58
Tabel 4. 6 Matriks Pengujian Fungsionalitas Prototipe AI Companion 61

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa transformasi mendasar dalam ekosistem pendidikan tinggi, salah satunya ditandai dengan adopsi *Learning Management System* (LMS) seperti Moodle. Kehadiran platform pembelajaran digital ini tidak sekadar menjadi ruang maya untuk mengumpulkan tugas atau mendistribusikan materi, tetapi juga berkembang menjadi lingkungan yang merekam setiap jejak interaksi mahasiswa secara mendetail. Berdasarkan tren pendidikan global, pemanfaatan rekam jejak digital atau *Educational Data Mining* (EDM) terus meningkat karena data ini menyimpan informasi berharga mengenai kebiasaan dan pola belajar peserta didik (Kharis & Zili, 2022). Meskipun data yang dihasilkan sangat melimpah, mayoritas institusi pendidikan masih terpaku pada paradigma evaluasi tradisional yang hanya berfokus pada hasil akhir berupa nilai ujian.

Ketidakmampuan sistem tradisional dalam melacak proses pengerjaan ini sering kali menimbulkan dampak psikologis dan akademis yang nyata. Ketika dihadapkan pada ujian digital yang rumit, mahasiswa kerap menerapkan strategi pengerjaan yang beragam, mulai dari membaca soal dengan sangat hati-hati, mengalami kepanikan waktu, hingga menebak jawaban dengan cepat (Akçapınar et al., 2024). Sayangnya, ketika mahasiswa mendapatkan nilai yang kurang memuaskan, platform *e-learning* tidak memiliki fitur untuk menjelaskan sumber kekurangannya. Tanpa adanya pelacakan perilaku, penilaian otomatis terasa seperti keputusan kaku yang berjarak dari realitas usaha peserta didik, sehingga mahasiswa kehilangan ruang refleksi untuk mengevaluasi strategi belajar mereka (Khosravi et al., 2022).

Di sisi lain, tuntutan pendidikan modern kini telah bergeser dari penilaian yang bersifat sumatif (sekadar menguji di akhir) menuju penilaian formatif yang mengutamakan perbaikan berkelanjutan. Evaluasi yang ideal seharusnya mampu membuka ruang dialog yang transparan antara apa yang dinilai oleh sistem dengan apa yang dipahami oleh peserta didik. Namun, dalam konteks kelas digital dengan jumlah peserta yang besar, menuntut dosen pengampu untuk membedah rekam jejak aktivitas ratusan mahasiswa satu per satu secara manual sangatlah tidak efisien dari segi waktu. Oleh sebab itu, teknologi kecerdasan buatan memiliki potensi besar untuk hadir sebagai asisten pendukung dosen yang membantu menjembatani keterbatasan tersebut (Rahmawati et al., 2025).

Peluang pemanfaatan data ini didukung penuh oleh ketersediaan data sekunder yang valid, dimana rekam jejak log aktivitas Moodle diperoleh langsung melalui dosen pengampu mata kuliah Keamanan Jaringan. Data riil ini mencerminkan dinamika pengerjaan ujian mahasiswa yang sesungguhnya di dalam kelas. Namun, implementasi Kecerdasan Buatan/AI konvensional sering kali memunculkan tantangan karena sifat modelnya yang menyerupai *black-box*. Model AI mampu memetakan data log mahasiswa dengan akurasi tinggi, tetapi sistem tersebut tidak mampu menjelaskan rasionalisasi di balik pemetaan tersebut (Khosravi et al., 2022).

Mengatasi ruang kosong antara potensi data LMS dan kebutuhan akan evaluasi yang mendidik, penelitian ini mengusulkan penerapan pendekatan XAI. Transparansi penilaian pembelajaran dalam konteks penelitian ini didefinisikan sebagai keterbukaan sistem dalam menjelaskan proses kognitif di balik perolehan skor ujian, sehingga mahasiswa tidak sekadar disajikan angka akhir yang kaku. Dalam operasionalnya, meskipun sistem komputasi dibangun menggunakan arsitektur regresi untuk memprediksi nilai ujian, tujuan akhirnya bukanlah prediksi itu sendiri mengingat nilai aktual telah diketahui dari *gradebook*. Kemampuan prediksi komputasi ini murni diposisikan sebagai proksi (perantara) untuk memastikan model telah menangkap pola dengan benar. Setelah model terbukti valid secara matematis, fungsi utamanya digeser menjadi lensa analitik melalui kerangka kerja XAI untuk mengungkap korelasi perilaku spesifik yang berasosiasi dengan perolehan nilai tersebut.

Melalui penerapan XAI, Moodle yang awalnya hanya mencatat jejak aktivitas digital kini dapat dibongkar pola-polanya lalu diterjemahkan menjadi umpan balik yang rasional. Penelitian ini bertujuan untuk merancang sebuah prototipe AI *Companion* yang objektif. Pendekatan ini diharapkan tidak hanya melengkapi literatur mengenai analisis data pendidikan (Waheed et al., 2023), tetapi juga mengembalikan esensi evaluasi sebagai instrumen formatif. Dengan penyajian umpan balik yang reflektif dan bukan deterministik, mahasiswa dibantu untuk mengenali kecenderungan perilaku mereka sendiri guna memperbaiki strategi belajar di masa depan.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan dan evaluasi kinerja model *Machine Learning* dengan menggunakan metrik *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), Koefisien Determinasi (R^2), dan Estimasi Akurasi dalam mengungkap pola perilaku pembelajaran mahasiswa berdasarkan data log aktivitas di platform Moodle pada mata kuliah Keamanan Jaringan?
2. Bagaimana implementasi kerangka kerja XAI dan penerapan AI companion dalam memberikan transparansi penilaian sehingga rekam jejak komputasi dapat diterjemahkan menjadi umpan balik formatif yang mudah dipahami oleh mahasiswa?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Menerapkan dan mengevaluasi kinerja model *Machine Learning* dengan menggunakan metrik *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), Koefisien Determinasi (R^2), dan Estimasi Akurasi dalam mengungkap pola perilaku pembelajaran mahasiswa berdasarkan data log aktivitas di platform Moodle pada mata kuliah Keamanan Jaringan.
2. Mengimplementasikan kerangka kerja XAI dan menerapkan AI companion dalam memberikan transparansi penilaian sehingga rekam jejak komputasi

dapat diterjemahkan menjadi umpan balik formatif yang mudah dipahami oleh mahasiswa.

1.4 Batasan Penelitian

Agar penelitian tetap fokus dan terukur, berikut adalah batasan-batasan yang peneliti tetapkan:

1. Penelitian ini secara eksklusif menggunakan data historis log aktivitas dan nilai yang diekspor dari platform Moodle untuk mata kuliah Keamanan Jaringan di Program Studi Teknologi Informasi.
2. Parameter yang diamati difokuskan pada metrik perilaku mahasiswa selama sesi ujian digital berlangsung, seperti durasi pengerjaan, kecepatan klik antarmuka, waktu inisiasi ujian, konsistensi jaringan (Variansi IP), serta riwayat kepandaian dasar (Skor Historis).
3. Penelitian dibatasi pada penerapan pemodelan prediktif dasar yang dipadukan dengan pustaka XAI untuk menghasilkan penjelasan berbasis teks mengenai hasil evaluasi mahasiswa, tanpa membangun LMS atau platform pembelajaran yang baru.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian yang dilakukan terbagi menjadi dua kategori utama, yaitu:

1.5.1 Manfaat Teoretis

Memperkaya literatur di bidang *Educational Data Mining* dan memberikan perspektif baru bahwa kecerdasan buatan dalam pendidikan tidak hanya sebatas alat prediksi, melainkan dapat dikembangkan menjadi instrumen evaluasi yang transparan untuk mendukung proses pembelajaran.

1.5.2 Manfaat Praktis

1. Bagi Dosen: Menyediakan alternatif instrumen penilaian yang mampu menganalisis integritas dan pola pengerjaan ujian mahasiswa secara otomatis dan objektif, sehingga dapat mempermudah dosen dalam memberikan evaluasi formatif.

2. Bagi Mahasiswa: Menghadirkan pengalaman evaluasi yang jauh lebih bermakna. Mahasiswa dapat mengetahui secara transparan faktor-faktor perilaku belajar apa saja yang memengaruhi nilai mereka, sehingga mendorong kemandirian belajar dan evaluasi diri yang lebih baik.

.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Eksplorasi terhadap rekam jejak digital peserta didik dalam lingkungan pembelajaran daring telah menjadi fokus utama dalam berbagai studi *Educational Data Mining* (EDM) selama beberapa tahun terakhir. Banyak peneliti terdahulu berhasil membuktikan bahwa pola interaksi mahasiswa di platform seperti Moodle memiliki korelasi yang sangat erat dengan keberhasilan akademik mereka. Meskipun demikian, sebagian besar pendekatan yang digunakan masih menyisakan ruang kosong untuk dikembangkan, terutama terkait pemanfaatan data pada tingkat aktivitas yang lebih spesifik dan keterbukaan sistem evaluasi yang dihasilkan.

Untuk memetakan posisi penelitian ini di antara studi-studi yang telah ada, dilakukan peninjauan terhadap beberapa literatur relevan yang berfokus pada analisis data log, perilaku kuis, serta penerapan transparansi algoritmik XAI. Rangkuman mengenai referensi, judul, metode, dan hasil dari penelitian terdahulu tersebut dapat dilihat pada Tabel 2.1

Tabel 2. 1 Penelitian Terdahulu

NO	REFERENSI	MODEL	JUDUL	HASIL
1.	Nouri & Saeed (2026)	Pemodelan hibrida menggunakan algoritma XGBoost yang dipadukan dengan SHAP untuk interpretabilitas model.	*A Hybrid Transformer XGBoost Framework with SHAP Explainability for Multimodal Student Performance*	Kombinasi XGBoost dan SHAP terbukti menghasilkan akurasi tinggi sekaligus mampu membedah kontribusi fitur perilaku mahasiswa secara

NO	REFERENSI	MODEL	JUDUL	HASIL
				lokal dan transparan.
2.	Villegas-Ch et al. (2025)	Pengembangan sistem asisten pendidikan terintegrasi pada Moodle menggunakan metode NLP dan kerangka kerja XAI.	*Explainable educational assistant integrated in Moodle: automated semantic assessment and adaptive tutoring based on NLP and XAI*	Penerapan XAI pada asisten e-learning berhasil meningkatkan efektivitas penilaian otomatis dan memberikan transparansi bimbingan yang adaptif bagi mahasiswa.
3.	He et al. (2025)	Analisis dimensi berkas log aktivitas LMS menggunakan pengelompokan hierarkis dan analisis jaringan psikometrik.	*Uncovering Dimensions of Learning Management System Event Logs: Insights from Exploratory Factor Analysis, Psychometric Network Analysis, and Consensus Hierarchical Clustering*	Pemetaan berhasil mengurai kompleksitas data log LMS menjadi dimensi perilaku belajar yang terstruktur, memberikan pandangan mendalam mengenai pola interaksi digital.
4.	Akçapınar et al. (2024)	Analisis urutan (*Sequence Analysis*) untuk memetakan pola	Uncovering Students' Temporal Quiz-Taking Behavior	Eksperimen berhasil mengidentifikasi variasi strategi

NO	REFERENSI	MODEL	JUDUL	HASIL
		perilaku temporal mahasiswa saat mengambil kuis.	Patterns Using Sequence Analysis	pengerjaan kuis digital mahasiswa berdasarkan dimensi waktu, durasi, dan ritme pengerjaan kuis di LMS.
5.	da Silva Costa et al. (2022)	Pemodelan kerangka kerja prediksi menggunakan teknik *Explainable AI* pada lingkungan pembelajaran virtual.	A Framework for prediction of dropout in distance learning through XAI techniques in Virtual Learning Environment	Teknik XAI mampu memberikan penjelasan yang valid mengenai faktor risiko mahasiswa, meskipun fokus utamanya masih terbatas pada prediksi pemisahan akademik (*dropout*).
6.	Wijekoon et al. (2021)	Penerapan penjelasan kontrafaktual (*Counterfactual Explanations*) berbasis jejak aktivitas Moodle.	Counterfactual Explanations for Student Outcome Prediction with Moodle Footprints	Pendekatan XAI kontrafaktual berhasil memberikan rekomendasi alternatif kepada mahasiswa mengenai perubahan perilaku yang diperlukan untuk mengubah

NO	REFERENSI	MODEL	JUDUL	HASIL
				hasil prediksi akademik mereka.
7.	Yağcı (2022)	Penerapan algoritma *machine learning* konvensional (seperti Random Forest dan Support Vector Machine) untuk memprediksi performa akademik mahasiswa berdasarkan data log aktivitas di LMS.	Educational data mining: prediction of students' academic performance using machine learning algorithms	Pemodelan *machine learning* terbukti mampu memprediksi nilai akhir mahasiswa dengan baik, namun model ini masih beroperasi secara *black-box* sehingga tidak mampu memberikan penjelasan rasional mengenai faktor perilaku yang paling krusial.

Berdasarkan kajian terhadap sejumlah penelitian terdahulu yang terangkum dalam Tabel 2.1, ditemukan sebuah celah penelitian (*research gap*) yang nyata di lapangan. Analisis arsitektur dimensi log aktivitas LMS secara ekstensif sejatinya telah dilakukan oleh He et al., (2025) namun fokus mereka masih berada pada pengelompokan pola psikometrik berskala makro, bukan pada intervensi formatif di level ujian. Di sisi lain, Da Silva Costa et al. (2022) telah merintis penerapan kerangka kerja XAI pada lingkungan pembelajaran virtual, akan tetapi implementasinya masih ditujukan secara sempit untuk mendeteksi risiko pemisahan akademik (*dropout*), bukan mengevaluasi detail perilaku taktis saat kuis berlangsung. Sementara itu, Meskipun masih berstatus *preprint*, penelitian dari Nouri & Saeed (2026) telah membuktikan ketangguhan algoritma XGBoost dan

SHAP dalam memprediksi performa mahasiswa, namun pemodelannya juga masih ditujukan untuk luaran performa akhir secara makro.

Penelitian ini hadir untuk menjembatani ruang kosong tersebut dengan menggabungkan keunggulan dari studi-studi di atas. Fokus utama penelitian dialihkan pada isolasi data log secara spesifik per sesi kuis untuk menghindari hilangnya konteks evaluasi. Selanjutnya, kompleksitas perhitungan matematis model XGBoost akan dibedah menggunakan TreeSHAP dan dikonversi secara otomatis menjadi sistem pendamping akademik AI *Companion* yang menghasilkan narasi umpan balik berbahasa Indonesia. Melalui pendekatan ini, mahasiswa tidak sekadar menerima nilai kuis secara mutlak, melainkan memperoleh transparansi penilaian yang edukatif dan formatif berdasarkan pola perilaku pengerjaan mereka yang sesungguhnya.

2.2 LMS Moodle dan Karakteristik Berkas Log

Learning Management System (LMS) merupakan platform perangkat lunak terintegrasi yang dirancang untuk mengelola, mendokumentasikan, melacak, dan menyampaikan materi pembelajaran secara daring. Di antara berbagai platform yang tersedia di dunia pendidikan tinggi, Moodle menjadi salah satu sistem bersistem terbuka (*open-source*) yang paling banyak diadopsi karena fleksibilitas serta kelengkapan fiturnya. Selain berfungsi sebagai medium distribusi konten akademik, Moodle memiliki keunggulan alami dalam merekam seluruh aktivitas pengguna secara otomatis ke dalam berkas log data yang terstruktur Bradley (2021).

Setiap kali mahasiswa melakukan interaksi di dalam platform mulai dari membaca materi, mengakses forum, hingga menekan tombol tindakan saat pengerjaan ujian berlangsung sistem secara mandiri akan mencatat jejak digital tersebut bersamaan dengan penanda waktu (*timestamp*) dan alamat protokol internet (*IP address*). Berkas log ini bukan sekadar rekaman administratif biasa, melainkan representasi dari keterlibatan aktif dan kebiasaan belajar mahasiswa di ruang digital. Dalam konteks kuis pada mata kuliah Keamanan Jaringan, data log yang diperoleh langsung dari dosen pengampu dapat diolah untuk mencerminkan bagaimana dinamika pengerjaan ujian berlangsung, termasuk indikator durasi

aktual, stabilitas jaringan yang digunakan, dan ritme interaksi mahasiswa saat menghadapi soal-soal analitis.

2.3 Eksplorasi Data Pendidikan (*Educational Data Mining*)

Educational Data Mining (EDM) merupakan rumpun ilmu komputasi yang berfokus pada pengembangan metode untuk mengeksplorasi jejak data unik yang berasal dari lingkungan akademik Melalui pemanfaatan teknik penambangan data dan algoritma pembelajaran mesin, EDM berfungsi untuk mengurai tumpukan data mentah yang abstrak menjadi pola informasi tersembunyi yang memiliki nilai pedagogis dan edukatif. Sistem berbasis EDM ini dirancang untuk mempelajari rekaman pengalaman, melakukan inferensi terhadap masukan perilaku, serta memetakan faktor-faktor krusial yang memengaruhi perkembangan atau kendala belajar yang dihadapi oleh peserta didik (Guleria & Sood, 2023). Lebih lanjut, Huang et al. (2025) dalam tinjauan sistematisnya menegaskan bahwa penerapan algoritma *machine learning* pada analisis data proses pendidikan terbukti jauh lebih tangguh dalam mengekstrak pola kognitif yang kompleks dari rekam jejak digital dibandingkan dengan pendekatan analisis statistik tradisional.

Secara konvensional, evaluasi hasil belajar cenderung bersifat sumatif dan hanya melihat performa mahasiswa dari nilai akhir yang diperoleh. Namun, melalui pemanfaatan EDM, fokus evaluasi dapat digeser menuju penilaian proses yang bersifat formatif. Pendekatan ini memungkinkan sistem untuk menganalisis strategi kognitif mahasiswa selama ujian, seperti bagaimana mereka mengelola waktu pengerjaan dan bagaimana konsistensi fokus mereka terjaga. Evaluasi berbasis proses ini memberikan pandangan yang lebih adil dan komprehensif, karena performa mahasiswa tidak lagi dinilai semata-mata dari hasil akhir, melainkan dari usaha dan pola perilaku belajar yang mereka tunjukkan sepanjang sesi evaluasi berlangsung.

2.4 Kecerdasan Buatan/AI

Kecerdasan buatan /AI merupakan bidang ilmu komputer yang berfokus pada penciptaan sistem yang mampu meniru kemampuan berpikir, belajar, dan bertindak layaknya manusia. Menurut Santoso (2023), AI didefinisikan sebagai kecerdasan entitas ilmiah yang ditambahkan kepada suatu sistem yang dapat diatur untuk membantu manusia dalam menyelesaikan pekerjaan secara lebih efektif. AI bekerja dengan memanfaatkan perangkat keras dan perangkat lunak canggih untuk memproses data, mengenali pola, serta menghasilkan keputusan yang menyerupai cara manusia berpikir (Santoso, 2023).

Secara umum, AI mencakup beberapa cabang utama, seperti machine learning, expert systems, speech recognition, fuzzy logic, dan artificial neural networks (Rojabi, 2025; Rustiyana et al., 2025). Penggunaan AI kini tidak hanya terbatas pada industri, tetapi juga telah berkembang pesat dalam bidang pendidikan. Menurut (Sudirwo et al., 2025), penerapan AI di sektor akademik ini terbukti mampu meningkatkan efisiensi pengolahan data, mendukung pengambilan keputusan, memprediksi hasil belajar mahasiswa, hingga memberikan rekomendasi pembelajaran secara personal.

Di balik manfaatnya, AI juga memiliki keterbatasan dalam hal transparansi hasil keputusan. Banyak model AI yang beroperasi seperti black box, di mana proses dan dasar pengambilan keputusan tidak dapat dijelaskan secara langsung kepada pengguna. Hal inilah yang kemudian melahirkan kebutuhan akan XAI, yaitu pendekatan yang berfokus pada interpretabilitas dan akuntabilitas hasil keputusan sistem cerdas (Baskoro et al., 2025).

2.5 Machine Learning

Machine Learning merupakan salah satu cabang utama dari kecerdasan buatan yang berfokus pada pengembangan model matematis dan algoritma komputasi agar sistem mampu belajar dari data. Melalui pendekatan ini, komputer dapat meningkatkan kinerjanya secara bertahap dalam menyelesaikan suatu tugas tanpa harus diberikan instruksi pemrograman secara eksplisit. Dengan kata lain, *machine learning* memungkinkan komputer memiliki kemampuan belajar dan beradaptasi

secara mandiri berdasarkan pola yang ditemukan dari data yang diolah (Junaidi et al., 2024).

Berbeda dengan pemrograman konvensional yang menggunakan aturan logika tetap, *machine learning* bekerja dengan memanfaatkan model prediktif dan statistik untuk menemukan hubungan antar variabel. Beberapa jenis algoritma yang umum digunakan dalam *machine learning* antara lain:

1. Regresi, yaitu algoritma yang digunakan untuk memprediksi nilai kontinu, misalnya memperkirakan harga suatu produk berdasarkan fitur-fitur tertentu.
2. Klasifikasi, digunakan untuk mengelompokkan data ke dalam kategori atau kelas tertentu berdasarkan pola yang telah dipelajari.
3. Pengelompokan (Clustering), diterapkan untuk mengidentifikasi kelompok data yang memiliki karakteristik serupa tanpa memerlukan label awal.

Dalam penerapannya, *machine learning* umumnya dikembangkan menggunakan bahasa pemrograman Python, yang menyediakan berbagai pustaka pendukung seperti Pandas, NumPy, dan Scikit-Learn untuk mempermudah proses pengolahan data, pelatihan model, serta evaluasi hasil prediksi.

2.6 Python

Python merupakan bahasa pemrograman yang terus mengalami perkembangan pesat dari tahun ke tahun. Bahasa ini pertama kali dikembangkan oleh Guido van Rossum pada tahun 1991 dan terinspirasi dari bahasa pemrograman ABC. Salah satu keunggulan utama Python adalah sifatnya yang *open-source*, sehingga siapa pun dapat berkontribusi dalam pengembangan dan penyempurnaannya. Python menjadi salah satu bahasa pemrograman yang sangat populer karena mudah dipelajari, bersifat fleksibel, memiliki banyak pustaka siap pakai, serta banyak digunakan dalam bidang kecerdasan buatan dan *data science* (Sari et al., 2025).

Keberhasilan Python dalam dunia pengembangan perangkat lunak juga didukung oleh ketersediaan berbagai pustaka (*library*) yang mempercepat proses analisis data dan penerapan algoritma pembelajaran mesin. Menurut Sari et al., (2025) beberapa pustaka penting di antaranya:

1. Scikit-Learn: Scikit-Learn merupakan pustaka *machine learning* yang dikembangkan khusus untuk Python. Pustaka ini menyediakan berbagai alat sederhana namun efisien untuk melakukan analisis data dan membangun model prediktif. Di dalamnya terdapat beragam algoritma yang dapat digunakan untuk tugas klasifikasi, regresi, maupun pengelompokan (*clustering*), sehingga memudahkan peneliti dan pengembang dalam menerapkan berbagai teknik pembelajaran mesin tanpa harus menulis kode dari awal.
2. NumPy: NumPy (Numerical Python) adalah pustaka fundamental dalam komputasi ilmiah berbasis Python. Pustaka ini menyediakan struktur data berupa *array* multidimensi dan sekumpulan fungsi matematis tingkat tinggi untuk melakukan perhitungan numerik secara cepat dan efisien. NumPy juga menjadi dasar bagi banyak pustaka Python lainnya yang bergerak di bidang analisis data dan kecerdasan buatan.
3. Matplotlib: Matplotlib merupakan pustaka Python yang berfungsi untuk menghasilkan visualisasi data dalam berbagai bentuk grafik, seperti diagram batang, garis, maupun sebar (*scatter plot*). Pustaka ini banyak digunakan oleh peneliti, analis data, dan ilmuwan karena kemampuannya menampilkan hasil analisis secara informatif dan menarik secara visual, sehingga mempermudah proses interpretasi data.

2.7 Google Colaboratory

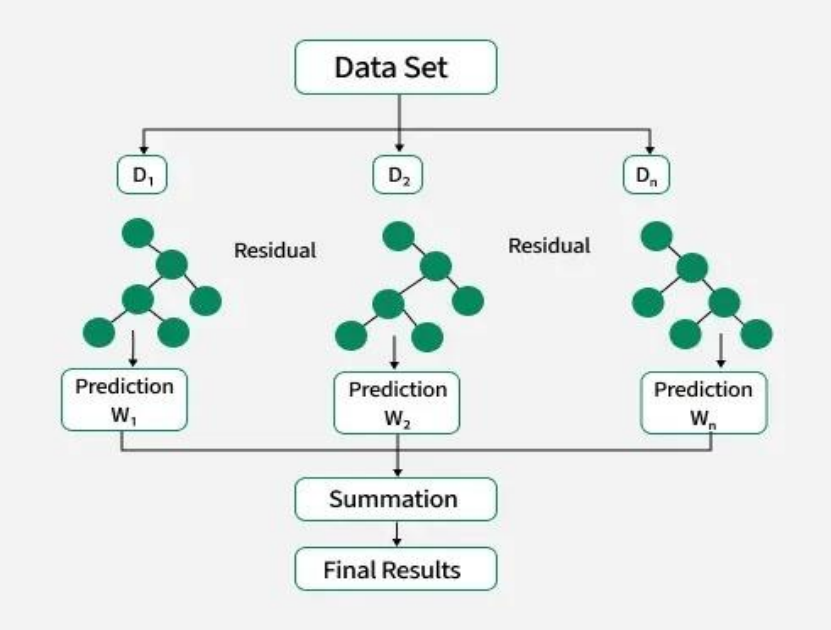
Google Colaboratory (Google Colab) merupakan platform komputasi awan yang sangat membantu dalam penelitian dan pembelajaran kecerdasan buatan serta pembelajaran mesin, terutama bagi mereka yang memiliki keterbatasan perangkat keras lokal. Penelitian oleh Voon et al. (2021) membuktikan bahwa pemanfaatan akselerator perangkat keras di lingkungan Google Colab dapat mempercepat waktu pelatihan model secara signifikan. Dalam eksperimennya mengevaluasi performa Intel Xeon CPU, Nvidia T4 GPU, dan TPU v2-8 menggunakan arsitektur Convolutional Neural Network (CNN) VGG16 yang dilatih dari awal, ditemukan bahwa instans GPU mampu bekerja 25 kali lebih cepat daripada CPU. Sementara itu, penggunaan unit TPU dalam eksperimen tersebut menghasilkan peningkatan kecepatan hingga 12 kali lipat dibandingkan dengan CPU. Meskipun platform ini sangat tangguh untuk menangani tugas komputasi intensif tanpa memerlukan server

fisik, Google Colab memiliki batasan teknis yang ketat bagi pengguna gratis, seperti batas maksimal waktu eksekusi (maximum execution time) selama 12 jam dan batasan waktu jeda (maximum idle time) maksimal selama 90 menit. Dengan demikian, Google Colab tidak hanya mempermudah pelaksanaan eksperimen teknis, tetapi juga berpotensi menyokong aspek pembelajaran dan pengalaman eksperimental yang lebih baik dalam konteks pendidikan.

2.8 Ensemble Learning dan XGBoost

Ensemble Learning merupakan sebuah paradigma analitik dalam *machine learning* di mana beberapa model dasar (*weak learners*), biasanya berupa pohon keputusan (*decision trees*), dilatih secara bersama-sama. Hasil prediksi dari masing-masing model kemudian digabungkan untuk menghasilkan satu prediksi tunggal yang tingkat kesalahannya jauh lebih rendah dan lebih tahan terhadap *noise* dibandingkan pemodelan yang hanya mengandalkan satu algoritma tunggal (Permana et al., 2026).

Salah satu implementasi tingkat lanjut dari paradigma ini adalah algoritma *eXtreme Gradient Boosting* (XGBoost). XGBoost bekerja dengan kerangka kerja *boosting* secara berurutan (*sequential*), di mana setiap pohon keputusan baru secara spesifik dibangun untuk mengoreksi tingkat kesalahan atau residu dari pohon yang terbentuk sebelumnya. Algoritma ini dirancang dengan efisiensi tinggi melalui dukungan komputasi paralel dan memiliki fitur regularisasi matematis yang ketat guna mencegah *overfitting* pada saat model berhadapan dengan data latih yang kompleks (Permana et al., 2026). Keunggulan struktural inilah yang menjadikan XGBoost sangat tangguh untuk mengatasi karakteristik data pendidikan yang sering kali tidak seimbang (*imbalanced*), seperti pada rekam jejak multidimensional dari log LMS Moodle, sekaligus memberikan tingkat akurasi prediksi performa yang sangat presisi (Permana et al., 2026). Secara visual, alur kerja *Sequential Boosting* pada algoritma XGBoost, di mana setiap pohon memperbaiki galat dari pohon sebelumnya, dapat dilihat pada Gambar 2.1.



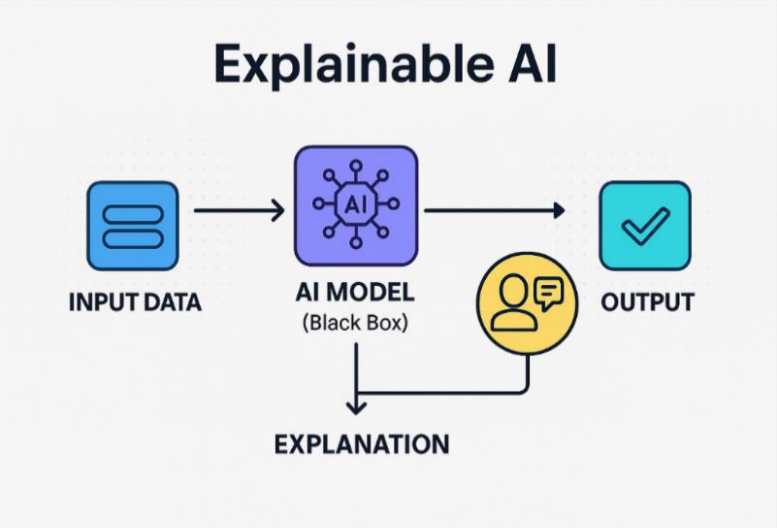
Gambar 2. 1 Arsitektur Sequential XGBoost

2.9 Explainable Artifical Intelligence (XAI)

Explainable Artificial Intelligence (XAI) merupakan paradigma dalam kecerdasan buatan yang memprioritaskan transparansi dan keterpahaman proses kerja model komputasi agar dapat diinterpretasikan oleh manusia. XAI hadir sebagai solusi krusial untuk mengatasi keterbatasan model *machine learning* yang bersifat *black-box*, di mana pengambilan keputusan internal sering kali tertutup dan sulit dipahami. XAI berperan penting dalam menjembatani kompleksitas algoritma dengan kebutuhan pengguna akan pemahaman logis mengenai alasan di balik keputusan sistem.

Dalam pengembangan teoretisnya, XAI didefinisikan sebagai serangkaian metode dan teknik yang dirancang agar perilaku model AI dapat ditafsirkan tanpa mengorbankan akurasi prediksi. Pendekatan sistematis ini memberikan landasan untuk membedah logika internal algoritma pembelajaran mesin. Selain itu, interpretabilitas dan penjelasan merupakan fondasi utama untuk membangun sistem AI yang andal dan dapat dipercaya (*trustworthy* AI), yang mencakup aspek kejelasan, transparansi, serta keandalan dalam jangka panjang (Guleria & Sood, 2023). Tanpa kemampuan menjelaskan keputusan yang dihasilkan, sistem AI berisiko menimbulkan bias, kesalahan prediktif yang tidak terkontrol, serta

ketidakpercayaan dari pengguna. Perbandingan konseptual antara model *Black-Box* AI yang tertutup dengan kerangka kerja XAI yang transparan dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Perbandingan Black Box dan XAI

Metodologi penciptaan model yang transparan dalam ekosistem XAI umumnya melibatkan tahapan operasional yang sistematis, mencakup eksplorasi dataset (*pre-modelling*), penggunaan model transparan secara bawaan (*explainable modelling*), hingga penerapan teknik ekstraksi logika setelah model dilatih (*post-modelling*). Teknik pada tahap *post-modelling* sering diklasifikasikan ke dalam pendekatan *model-specific* yang bersifat spesifik pada algoritma tertentu, serta *model-agnostic* yang lebih fleksibel karena dapat diterapkan pada berbagai jenis arsitektur *machine learning*. Dalam implementasinya, kerangka kerja XAI memiliki dua dimensi interpretasi utama:

1. Interpretabilitas Global (*Model Explainability*): Berfokus pada pemahaman perilaku model secara komprehensif, termasuk identifikasi fitur yang paling dominan berpengaruh terhadap keputusan model pada seluruh kumpulan data.
2. Interpretabilitas Lokal (*Decision Explainability*): Berfungsi menjelaskan dasar rasional bagi keputusan prediktif untuk kasus individual, seperti alasan spesifik seorang mahasiswa dikategorikan memiliki risiko akademik tertentu.

Penerapan XAI, khususnya melalui metode seperti *SHapley Additive exPlanations* (SHAP) dan LIME, telah terbukti sangat efektif dalam membedah faktor perilaku kognitif serta indikator performa akademik mahasiswa secara terbuka. Implementasi ini memungkinkan institusi pendidikan untuk tidak hanya memperoleh prediksi yang akurat secara matematis, tetapi juga menciptakan sistem yang adil, akuntabel, serta dapat dipertanggungjawabkan secara etis dalam mendukung intervensi akademik yang tepat sasaran.

2.10 Shapley Additive Explanations (SHAP)

Shapley Additive Explanations (SHAP) merupakan sebuah kerangka kerja penjelas berjenis *model-agnostic* yang dirancang untuk menginterpretasikan luaran prediksi dari berbagai model *machine learning*, baik dalam ruang lingkup lokal maupun global (Guleria & Sood, 2023). Metode yang diperkenalkan oleh Lundberg dan Lee pada tahun 2017 ini dibangun di atas fondasi matematis teori permainan kooperatif (*cooperative game theory*), yang secara spesifik mengadopsi perhitungan Nilai Shapley (*Shapley Values*) rumusan Lloyd Shapley pada tahun 1953 (Guleria & Sood, 2023). Dalam pendekatan teori permainan ini, proses pembentukan hasil prediksi oleh model diibaratkan sebagai sebuah "permainan", sementara atribut masukan seperti durasi kuis atau jumlah klik mahasiswa berperan sebagai "pemain" yang saling berkolaborasi untuk membentuk skor prediksi akhir (Nouri & Saeed, 2026).

Keunggulan utama dari kerangka kerja SHAP terletak pada kemampuannya untuk mendistribusikan bobot kontribusi ke setiap fitur secara adil, konsisten, dan memenuhi kaidah sifat aditif lokal. SHAP mengkalkulasi dampak marjinal murni dari sebuah fitur dengan cara membandingkan selisih nilai prediksi model di seluruh kemungkinan kombinasi subset fitur yang ada (Minh et al., 2022). Secara matematis, nilai kontribusi SHAP yang dinotasikan sebagai $\phi_i$untuk fitur ke-$i$dirumuskan melalui persamaan (1)

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \setminus Big[f(S \cup \{i\}) - f(S) \setminus Big] \qquad (1)$$

Keterangan:

ϕ_i (Phi): Nilai kontribusi akhir (Nilai Shapley) yang diberikan kepada fitur ke-i.

F: Himpunan total dari seluruh fitur masukan yang tersedia dalam model.

S: Subset (kombinasi) fitur-fitur lain yang *tidak* menyertakan fitur ke-i.

$|F|$: Jumlah total keseluruhan fitur.

$|S|$: Jumlah fitur yang ada di dalam subset S.

$f(S \cup \{i\})$: Nilai prediksi model ketika dievaluasi menggunakan kombinasi fitur S ditambah dengan fitur ke-i.

$f(S)$: Ekspektasi nilai prediksi model ketika hanya dievaluasi menggunakan subset fitur S saja (tanpa fitur ke-i).

$[f(S \cup \{i\}) - f(S)]$: Dampak marjinal murni (selisih prediksi) yang terjadi ketika fitur ke-i ikut dimasukkan ke dalam kombinasi.

$\frac{|S|!(|F|-|S|-1)!}{|F|!}$: Pembobotan matematis untuk menjamin bahwa nilai kontribusi didistribusikan secara adil berdasarkan seluruh kemungkinan permutasi fitur yang ada.

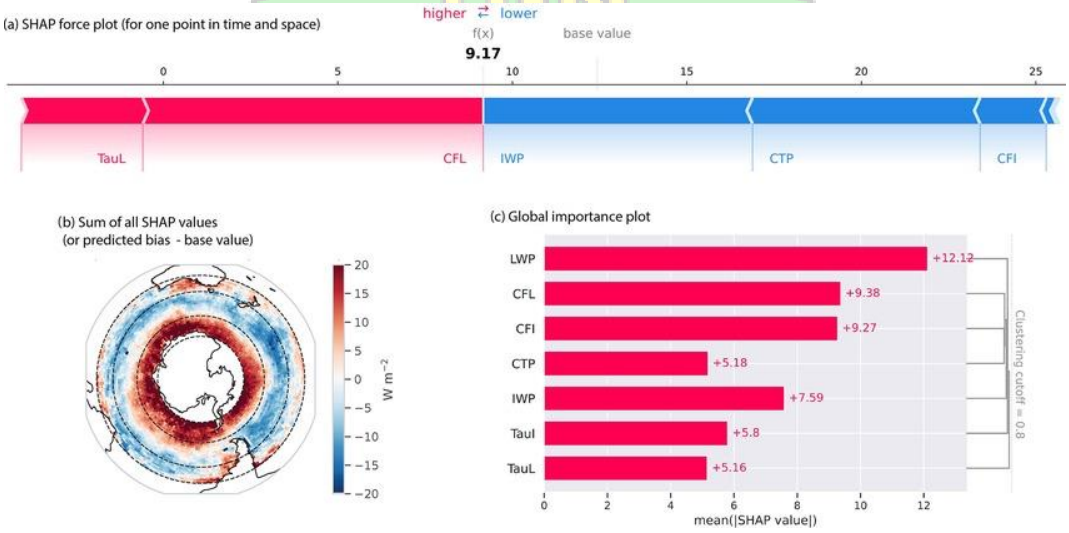
$\sum_{S \subseteq F \setminus \{i\}}$: Penjumlahan dari seluruh perhitungan dampak marjinal pada semua kemungkinan kombinasi subset S.

Pada persamaan tersebut, F mendefinisikan himpunan total dari seluruh fitur masukan yang tersedia. Simbol S merepresentasikan subset kombinasi fitur yang tidak menyertakan fitur ke-i, sedangkan $f(S)$ merupakan ekspektasi nilai prediksi model ketika hanya dievaluasi menggunakan fitur-fitur yang berada di dalam subset S tersebut (Nouri & Saeed, 2026).

Kalkulasi matematis pada persamaan di atas membutuhkan daya komputasi eksponensial yang sangat berat jika diterapkan pada dataset yang luas. Oleh karena itu, untuk algoritma yang berbasis pohon keputusan seperti XGBoost dan *Random Forest*, diterapkan variasi algoritmik spesifik yang dikenal sebagai TreeSHAP (Puspanagara, 2025).TreeSHAP bekerja dengan cara menelusuri lintasan struktur

percabangan pohon secara langsung, yang memungkinkannya mengekstraksi nilai atribusi lokal dengan efisiensi waktu komputasi yang jauh lebih tinggi dan akurat (Puspanagara, 2025).

Dalam konteks penelitian *Educational Data Mining* ini, luaran numerik yang dihasilkan oleh kalkulasi TreeSHAP memberikan arah polaritas yang tegas atas setiap perilaku ujian. Suatu fitur dapat bernilai positif yang mengindikasikan dampak apresiatif terhadap performa kuis, atau bernilai negatif yang merepresentasikan adanya penalti akademis. Presisi parameter matematis inilah yang selanjutnya dijadikan landasan parameter objektif bagi sistem pendamping AI *Companion*, di mana matriks angka kaku diterjemahkan secara otomatis menjadi teks narasi umpan balik yang personal, edukatif, dan transparan bagi setiap mahasiswa. Visualisasi dari kontribusi nilai positif dan negatif pada metode SHAP ini dapat direpresentasikan melalui SHAP force plot dan global importance plot, sebagaimana ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Visualisasi SHAP

2.11 Evaluasi dan Kinerja dan Interpretabilitas Model

Dalam pengembangan model kecerdasan buatan yang mengintegrasikan XAI, proses validasi tidak lagi hanya bergantung pada seberapa akurat model tersebut dalam membuat prediksi, tetapi juga pada seberapa baik logika prediksinya dapat dipahami oleh pengguna (Şahin et al., 2025). Oleh karena itu, pengujian sistem secara komprehensif menuntut adanya dua dimensi evaluasi utama yang berjalan beriringan, yaitu evaluasi kinerja prediktif secara komputasi dan evaluasi kualitas interpretabilitas dari penjelasan yang dihasilkan.

Evaluasi kinerja model prediktif pada dasarnya bertujuan untuk mengukur keandalan algoritma *machine learning* seperti XGBoost dalam memetakan data fitur menjadi luaran prediksi yang tepat sasaran. Evaluasi ini diukur menggunakan metrik statistik standar yang diturunkan dari matriks kebingungan (*confusion matrix*). Metrik-metrik fundamental tersebut meliputi tingkat akurasi (*accuracy*), presisi (*precision*), sensitivitas (*recall*), dan *F1-Score* (Junaidi et al., 2024). Pengukuran kinerja prediktif ini sangat esensial untuk memastikan bahwa algoritma mampu menggeneralisasi pola data log perilaku mahasiswa dengan tingkat kesalahan (*error rate*) yang minimal, serta terhindar dari bias algoritma (*underfitting* atau *overfitting*) sebelum hasil prediksinya diekstraksi ke dalam bentuk penjelasan.

Di sisi lain, ketika sebuah model dinyatakan memiliki kinerja prediktif yang tinggi, tahapan selanjutnya adalah menguji hasil luaran dari metode XAI seperti SHAP melalui evaluasi interpretabilitas. Secara kuantitatif, kualitas penjelasan algoritma diuji menggunakan metrik ketepatan (*fidelity*) dan stabilitas (*stability*) (Minh et al., 2022). *Fidelity* mengukur sejauh mana skor penjelas yang dihasilkan oleh SHAP benar-benar mencerminkan dan selaras dengan mekanisme pengambilan keputusan asli dari model *black-box* itu sendiri. Sementara itu, *stability* atau ketahanan model menguji konsistensi metode penjelas, di mana dua mahasiswa dengan profil data perilaku kuis yang identik harus mendapatkan bobot atribusi penjelasan yang stabil dan tidak saling bertolak belakang (Villegas-Ch et al., 2025).

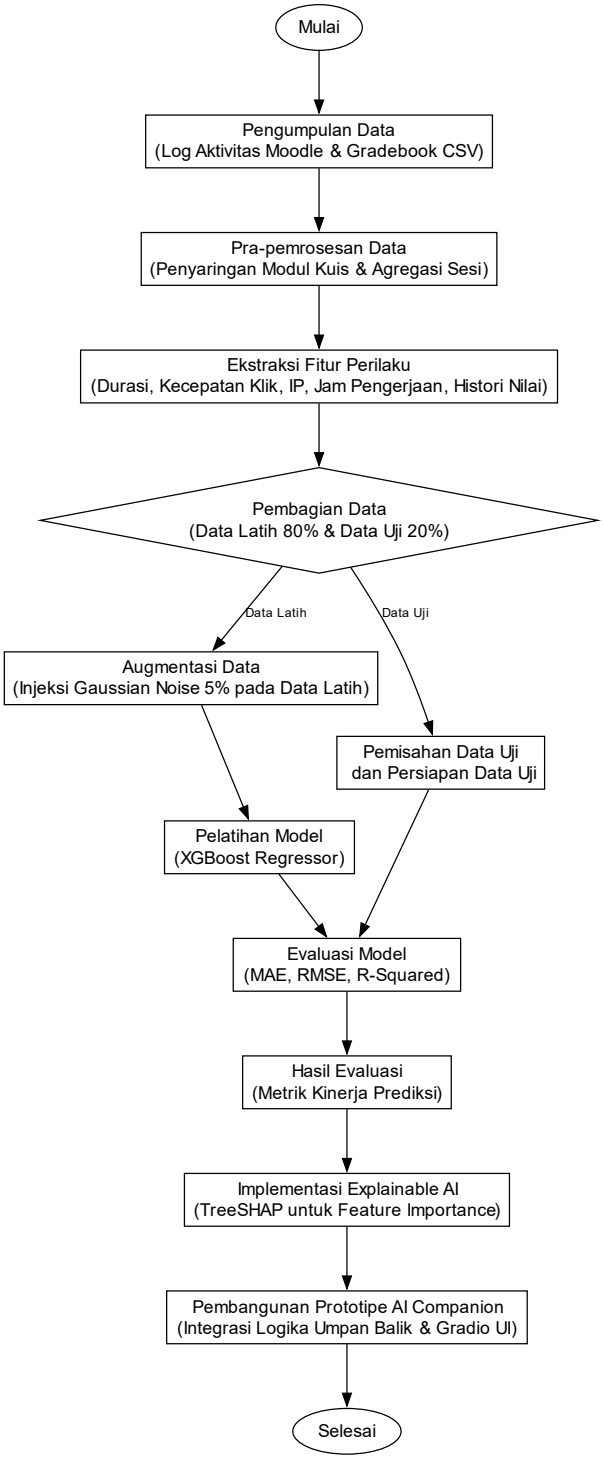
Selain pengujian komputasi murni, evaluasi interpretabilitas pada ekosistem XAI sangat bergantung pada pengujian kualitatif yang berpusat pada manusia (*human-grounded evaluation*). Evaluasi ini bertujuan untuk mengukur tingkat komprehensibilitas, yaitu seberapa mudah narasi penjelasan tersebut dicerna dan dipahami oleh pengguna akhir yang tidak memiliki latar belakang keilmuan komputer yang mendalam (Sahin et al., 2025). Dalam konteks implementasi sistem penilaian pendidikan, indikator keberhasilan interpretabilitas diukur dari kemampuan umpan balik AI dalam meningkatkan rasa percaya (*calibrated trust*) mahasiswa terhadap sistem otomatis, serta kemampuannya dalam memicu refleksi akademik secara aktif guna mengevaluasi proses belajar mereka (Ismanto et al., 2026; Sandika et al., 2025).

BAB III
METODE PENELITIAN

3.1 Tahapan penelitian

Metode penelitian yang digunakan dalam penelitian ini merupakan metode penelitian kuantitatif terapan dengan pendekatan penambangan data pendidikan *Educational Data Mining*. Pemilihan pendekatan ini didasarkan pada tujuannya untuk menggali dan mengekstrak pola perilaku belajar yang tersembunyi dari jejak interaksi mahasiswa di dalam sistem pembelajaran elektronik. Penelitian kuantitatif sendiri merupakan jenis penelitian yang mengumpulkan dan menganalisis data numerik untuk menjelaskan, memprediksi, atau mengontrol fenomena suatu variabel (Sugiyono, 2021).

Data yang dikumpulkan pada tahap awal berupa sekumpulan data log aktivitas digital mahasiswa dan rekapitulasi nilai (*gradebook*) yang ditarik langsung dari platform Moodle. Data sekunder tersebut merupakan representasi faktual dari rekam jejak kognitif maupun psikomotorik mahasiswa selama mengikuti evaluasi pada mata kuliah Keamanan Jaringan. Selanjutnya, mengacu pada metode analisis pemodelan prediktif, proses analisis dalam penelitian ini berfokus pada pengolahan riwayat klik mentah menjadi matriks angka yang terstruktur. Matriks ini kemudian digunakan untuk melatih algoritma *machine learning* guna menghasilkan prediksi performa, yang kinerja komputasinya akan diukur berdasarkan tingkat galatnya (*error rate*). Lebih jauh lagi, perhitungan matematis dari algoritma tersebut akan dibedah menggunakan kerangka kerja XAI agar hasil prediksi tidak hanya berhenti pada angka mutlak, melainkan dapat diterjemahkan menjadi umpan balik yang transparan. Adapun tahapan penelitian ini secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

Berdasarkan Gambar 3.1, alur proses pembangunan sistem pendamping akademik ini terdiri dari tiga proses utama, yaitu persiapan data, pemodelan

analitik, dan implementasi luaran. Proses persiapan data mencakup 3 langkah, yaitu:

1. Pengumpulan data log aktivitas dan nilai Moodle
2. Pra-pemrosesan data yang meliputi penyaringan kuis dan agregasi sesi
3. Ekstraksi fitur perilaku mahasiswa.

Selanjutnya, proses pemodelan analitik terdiri dari 3 langkah, yaitu Pembagian data menjadi data latih dan uji beserta proses augmentasinya, Pelatihan model regresi menggunakan algoritma XGBoost, dan Evaluasi performa model menggunakan metrik statistik. Sementara itu, proses implementasi luaran mencakup 2 langkah akhir, yaitu Penerapan algoritma *TreeSHAP* untuk menghitung kontribusi masing-masing fitur, dan Pembangunan prototipe antarmuka aplikasi AI *Companion* menggunakan Gradio.

3.2 Pengumpulan Data Observasi Awal (Studi Pendahuluan)

Sebagai landasan empiris untuk memvalidasi urgensi penelitian, tahap pertama yang dilakukan adalah pengumpulan data kualitatif melalui penyebaran kuesioner observasi awal (studi pendahuluan). Kuesioner ini disebarkan kepada 25 mahasiswa di lingkungan Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry yang memiliki pengalaman melaksanakan evaluasi pembelajaran pada mata kuliah Keamanan Jaringan menggunakan LMS Moodle. Instrumen kuesioner pada studi pendahuluan ini dirancang dengan memfokuskan observasi pada tiga indikator utama, yaitu:

1. Kebingungan Evaluatif Mahasiswa: Mengukur seberapa sering mahasiswa merasa kebingungan untuk mengevaluasi letak kesalahan atau kelemahan strategi belajar mereka ketika mendapatkan nilai kuis yang kurang memuaskan.
2. Kesadaran Perekaman Perilaku: Mengetahui tingkat kesadaran mahasiswa terhadap fakta bahwa sistem Moodle secara pasif merekam jejak interaksi digital mereka, seperti durasi pengerjaan, kecepatan klik antarmuka, dan waktu inisiasi ujian.
3. Urgensi Sistem Transparansi: Mengukur persepsi dan tingkat kebutuhan mahasiswa terhadap kehadiran asisten kecerdasan buatan AI *Companion* yang

mampu membedah data log tersebut untuk memberikan umpan balik formatif secara transparan.

Data hasil persentase dari kuesioner ini diposisikan sebagai justifikasi faktual pada perumusan masalah di Bab I. Selain itu, poin-poin keluhan dari observasi ini dijadikan landasan untuk mengukur efektivitas luaran sistem pada tahap akhir penelitian Bab IV, guna membuktikan bahwa prototipe AI *Companion* yang dibangun benar-benar mampu menjawab kebingungan evaluatif mahasiswa tersebut. Instrumen pertanyaan dan rekapitulasi lengkap dari hasil survei observasi awal ini dapat ditinjau pada bagian Lampiran.

3.3 Tahapan Pengumpulan Data (*Data Acquisition*)

Proses pengumpulan data dalam penelitian ini bersandar pada pendekatan studi dokumentasi sekunder yang diekstraksi secara langsung dari pangkalan data peladen LMS Moodle. Akuisisi data secara spesifik menargetkan tabel referensi silang yang merekam jejak aktivitas digital mahasiswa selama pelaksanaan evaluasi sumatif pada mata kuliah Keamanan Jaringan. Pendekatan ini dipilih untuk menjamin objektivitas data, di mana perilaku pengguna direkam secara *real-time* oleh sistem tanpa adanya intervensi atau bias observasional dari peneliti.

Sistem pangkalan data mengekstraksi dua komponen berkas utama berekstensi *Comma Separated Values* (CSV) yang akan saling diintegrasikan:

1. Moodle Standard Log: Merupakan kumpulan rekaman absolut bervolume tinggi yang mendokumentasikan setiap entitas pergerakan secara granular. Parameter atribut yang diekstrak dari log ini meliputi identifikasi anonim pengguna, spesifikasi komponen modul yang difilter secara ketat pada parameter *mod_quiz,* konteks peristiwa (*event context*), deskripsi aktivitas pendukung, alamat protokol internet (*IP address*) dari mesin klien untuk pelacakan konsistensi jaringan, serta stempel waktu presisi (*timestamp*) yang secara *default* dikonversi ke dalam format UNIX *Epoch*.
2. Gradebook Matrix: Merupakan instrumen berkas yang merangkum hasil skor evaluasi kumulatif untuk masing-masing kuis. Nilai numerik absolut yang terdapat di dalam dokumen ini dialokasikan sebagai variabel target regresi

kontinu (y) yang akan menjadi landasan evaluasi selama proses penyetelan parameter (*hyperparameter tuning*) dan pelatihan model *supervised learning*.

3.4 Pra-pemrosesan Data (*Data Preprocessing*)

Karakteristik inheren dari log aktivitas yang bersumber dari sistem *e-learning* cenderung memiliki dimensi yang sangat tinggi, asimetris, dan sarat akan anomali data (*noise*). Oleh karena itu, tahapan pra-pemrosesan diimplementasikan secara terstruktur untuk melakukan pembersihan, penyaringan, dan normalisasi matriks sebelum diumpankan ke dalam algoritma prediktif.

3.4.1 Filterisasi Modul dan Konteks Aktivitas

Mengingat penelitian ini secara eksklusif berfokus pada analisis perilaku kognitif selama pengerjaan evaluasi, seluruh baris log yang berafiliasi dengan aktivitas non-kuis harus direduksi. Entitas data yang mencatat tindakan seperti pengunduhan modul silabus (mod_resource), partisipasi dalam ruang diskusi (mod_forum), atau sekadar navigasi beranda dieliminasi secara permanen dari *dataframe* untuk mencegah intrusi data yang tidak relevan.

3.4.2 Konversi dan Sinkronisasi Zona Waktu

Stempel waktu (*timestamp*) bawaan Moodle yang tersimpan dalam format waktu universal terkoordinasi *Coordinated Universal Time* (UTC) wajib ditransformasikan ke dalam zona Waktu Indonesia Barat (WIB). Sinkronisasi kronologis ini merupakan prasyarat mutlak untuk memastikan validitas proses ekstraksi metrik pada tahap selanjutnya, khususnya dalam mengklasifikasikan perilaku pengerjaan kuis berdasarkan parameter waktu aktual demografi responden.

3.4.3 Penanganan Ketidaklengkapan Data (Missing Values Handling)

Dalam skenario operasional sistem evaluasi digital, sering kali ditemukan anomali di mana entitas pengguna tercatat memulai kuis namun tidak menyelesaikannya akibat kendala teknis, sehingga menyebabkan hilangnya referensi skor akhir pada *Gradebook*. Entitas baris yang mengalami kekosongan (*null*) pada variabel target ini akan dihapus sepenuhnya melalui metode

pendropan (*listwise deletion*). Teknik imputasi statistikal tidak diterapkan dalam kasus ini, mengingat penyisipan nilai buatan pada variabel dependen (skor) berpotensi mendistorsi dan merusak integritas fundamental dari arsitektur prediksi regresi.

3.4.4 Agregasi Sesi Pengguna (*Session Aggregation*)

Rekam jejak navigasi tunggal dari seorang pengguna umumnya terpecah menjadi puluhan hingga ratusan baris log diskrit. Untuk mengatasi fragmentasi tersebut, data dikelompokkan menggunakan fungsi *groupby* berdasarkan parameter kunci identifikasi sesi dan rentang operasional kuis. Seluruh kejadian prosedural, mulai dari titik inisiasi quiz_started hingga titik terminasi quiz_submitted, dilebur dan direpresentasikan kembali menjadi satu vektor pengamatan tunggal per mahasiswa.

3.5 Ekstraksi Fitur (Feature Engineering)

Data log yang telah melewati fase agregasi selanjutnya dikonversi ke dalam format matriks analitik. Penelitian ini memformulasikan serangkaian fitur berbasis *Educational Data Mining* (EDM) yang bertugas mentransformasikan rekaman klik mentah menjadi dimensi yang merepresentasikan karakteristik psikomotorik dan pendekatan kognitif peserta didik.

3.5.1 Durasi Total Akses ($T\Delta$)

Metrik ini dirancang untuk mengukur rentang waktu absolut yang didedikasikan oleh mahasiswa dalam menyelesaikan keseluruhan instrumen evaluasi. Durasi pengerjaan merupakan salah satu indikator perilaku primer yang sangat penting untuk diekstrak dari log aktivitas, karena data ini dapat memberikan gambaran objektif mengenai manajemen waktu, tingkat keseriusan, hingga potensi kesulitan kognitif yang dialami mahasiswa saat menghadapi soal ujian.

Secara sistem, perhitungan durasi ini tidak bergantung pada waktu yang diatur oleh pengawas, melainkan didasarkan pada rekam jejak digital riil di peladen Moodle. Nilainya diperoleh melalui kalkulasi selisih antara stempel waktu (*timestamp*) pada saat mahasiswa secara sadar menyerahkan jawaban

final dengan stempel waktu pada saat halaman kuis tersebut pertama kali diinisiasi. Formulasi perhitungan ekstraksi waktu ini dirumuskan sebagai persamaan (2)

$$T\Delta = Tsubmit - Tstart \quad (2)$$

Keterangan:

$T\Delta$: Durasi total penyelesaian kuis besaran waktu ini umumnya dikonversi ke dalam satuan ukur menit atau detik untuk standarisasi matriks fitur.

$Tsubmit$: Stempel waktu akhir yang terekam secara otomatis oleh sistem saat pengguna menekan tombol pengiriman jawaban final *event: quiz_submitted*.

$Tstart$ Stempel waktu awal yang menandai titik mula ujian, tercatat ketika pengguna membuka atau mengakses modul soal untuk pertama kalinya *event: quiz_started*.

Hasil perhitungan dari variabel $T\Delta$ ini nantinya tidak hanya berdiri sendiri, tetapi akan disandingkan dengan parameter interaksi lain (seperti total klik) guna mendeteksi berbagai pola pengerjaan, mulai dari kecenderungan menebak instan secara terburu-buru hingga keterlambatan respons akibat kurangnya penguasaan materi.

3.5.2 Kecepatan klik (Clik Rate Vc)

Kecepatan klik atau *click rate* merupakan rasio intensitas interaksi antarmuka yang difungsikan sebagai indikator perilaku utama dalam ekosistem pembelajaran digital. Metrik ini sangat krusial untuk mendeteksi berbagai anomali kebiasaan mahasiswa selama ujian berlangsung. Nilai rasio yang terlalu tinggi sering kali menjadi sinyal kuat adanya kecenderungan *rapid-guessing*, yaitu perilaku di mana mahasiswa mengisi jawaban secara insting, terburu-buru, atau asal tebak tanpa melakukan analisis komprehensif terhadap butir soal. Sebaliknya, nilai rasio yang terlampau rendah dapat mengindikasikan tingginya *idle-time*, yang merujuk pada keterlambatan respons akibat kebuntuan kognitif, kebingungan materi, atau jeda waktu saat mahasiswa meninggalkan perangkatnya sejenak tanpa aktivitas.

Untuk mengukur metrik ini, sistem tidak mencatat semua klik sembarangan, melainkan hanya menghitung aktivitas interaksi yang valid dan terekam di dalam log aktivitas seperti menekan opsi jawaban, menyimpan sementara, atau berpindah halaman soal. Variabel kecepatan klik ini diformulasikan dengan membagi akumulasi total interaksi klik yang sah dengan total durasi penyelesaian kuis yang telah dikonversi ke dalam satuan menit. Persamaan matematis untuk fitur ini dirumuskan sebagai persamaan (3)

$$V_c = \frac{C_{total}}{T_{\Delta menit}} \tag{3}$$

Keterangan:

Vc : Kecepatan interaksi klik (*Velocity of clicks* atau *Click rate*), merepresentasikan seberapa sering mahasiswa melakukan tindakan valid rata-rata per menitnya.

$Ctotal$: Jumlah akumulasi dari seluruh aktivitas klik valid yang didokumentasikan oleh peladen Moodle selama satu sesi pengerjaan kuis berlangsung.

$T\Delta(menit)$: Durasi total waktu penyelesaian ujian mahasiswa tersebut, yang nilainya diwajibkan dalam bentuk satuan ukur menit agar perhitungan rasio menjadi standar dan konsisten.

Melalui pendefinisian variable $T\Delta$ ini, arsitektur *machine learning* nantinya memiliki pijakan data yang kuat untuk membedakan konteks performa mahasiswa. Model tidak akan salah menafsirkan antara mahasiswa yang menyelesaikan kuis dengan cepat karena tingkat pemahamannya tinggi, dengan mahasiswa yang selesai cepat karena sekadar menggugurkan kewajiban ujian.

3.5.3 Penanda Waktu Larut Malam

Fitur ini berwujud variabel biner *Boolean* 0 atau 1 yang dirancang untuk mengidentifikasi mahasiswa yang memulai kuis pada rentang waktu larut malam. Logika sistem akan mengaktifkan nilai 1 pada fitur ini apabila titik awal pengerjaan (T_{start}) terekam pada rentang waktu antara pukul 23.00 hingga 04.00 WIB.

Mengingat sifat kuis di platform Moodle yang dapat berupa *open-window* (memiliki rentang pengerjaan yang tidak terikat waktu spesifik), fitur ini diinterpretasikan dengan penuh kehati-hatian. Asumsi konvensional mengenai degradasi fokus kognitif atau kelelahan ritme sirkadian diberikan batasan (*caveat*), mengingat sebagian mahasiswa secara alamiah memiliki kebiasaan belajar larut malam (*night-owl*) atau terdorong oleh kebiasaan mengejar tenggat waktu (*deadline-driven*). Oleh karena itu, penanda ini diposisikan secara netral untuk memetakan pola preferensi waktu evaluasi mahasiswa, alih-alih murni sebagai indikator penurunan konsentrasi semata.

3.5.4 Konsistensi Perangkat (*IP variance*)

Metrik ini merupakan indikator teknis yang dihitung dengan melacak jumlah total alamat protokol internet IP unik yang digunakan oleh seorang mahasiswa selama satu sesi kuis tunggal berlangsung. Dalam kondisi pengerjaan ujian yang normal dan ideal, seorang mahasiswa umumnya menggunakan satu koneksi jaringan yang stabil dari awal hingga akhir.

Jika akumulasi nilai metrik ini menunjukkan angka lebih dari 1, hal tersebut mengindikasikan adanya perpindahan topologi jaringan di tengah pengerjaan. Kondisi ini diposisikan sebagai faktor teknis murni, misalnya representasi dari mahasiswa yang mengalami gangguan sinyal sehingga terpaksa beralih dari koneksi Wi-Fi kampus ke jaringan seluler pribadi (*tethering*). Penafsiran fitur ini diberikan *caveat* yang ketat agar sistem tidak secara deterministik menafsirkan tingginya variansi IP sebagai bentuk kecurangan akademik atau kolaborasi eksternal, melainkan dievaluasi murni

sebagai indikator konsistensi dan kestabilan koneksi internet pengguna selama ujian berlangsung.

3.5.5 Skor Historis Basis (Baseline Proficiency)

Fitur historis ini difungsikan sebagai titik acuan untuk mendefinisikan tingkat kemampuan dasar mahasiswa secara individual sebelum mereka mengikuti kuis yang sedang dianalisis. Keberadaan variabel ini sangat krusial untuk mencegah terjadinya miskonsepsi algoritma dalam menilai kecepatan interaksi mahasiswa.

Sebagai contoh, jika algoritma hanya melihat faktor waktu pengerjaan yang sangat singkat, sistem bisa saja salah menyimpulkan bahwa mahasiswa tersebut asal menebak. Dengan memasukkan skor historis dasar, model dapat membedakan secara cerdas dan objektif antara mahasiswa dengan riwayat nilai tinggi yang selesai cepat karena memang sangat menguasai materi, dengan mahasiswa berkinerja historis rendah yang selesai cepat karena spekulasi acak. Metrik kepandaian dasar ini dikalkulasikan dengan mengambil nilai rata-rata pencapaian skor mahasiswa pada n kuis atau evaluasi sebelumnya (S_i), yang dirumuskan melalui persamaan (4)

$$\mu_H = \frac{1}{n}\sum_{i=1}^{n} S_i \qquad (4)$$

Keterangan:

μ_H (Miu H) merepresentasikan nilai rata-rata historis atau tingkat kepandaian dasar (baseline proficiency) yang dimiliki oleh mahasiswa.

n adalah jumlah total kuis atau evaluasi yang telah diselesaikan oleh mahasiswa tersebut sebelumnya.

$\sum_{i=1}^{n}$ merupakan simbol sigma yang menandakan operasi penjumlahan secara berurutan, dimulai dari skor kuis pertama ($i = 1$) hingga kuis ke-n.

S_i mewakili capaian skor atau nilai aktual yang diperoleh mahasiswa pada evaluasi atau kuis ke-i.

$\frac{1}{n}$ berfungsi sebagai operasi pembagian di akhir, di mana total keseluruhan skor yang telah dijumlahkan dibagi dengan banyaknya kuis untuk menghasilkan nilai rata-rata murni

Untuk memastikan bahwa fitur historis ini tidak mendominasi seluruh distribusi bobot penjelasan pada model, penelitian ini akan melakukan uji ablasi. Model XGBoost akan dilatih dan dievaluasi dalam dua skenario terpisah yakni dengan dan tanpa menyertakan fitur skor historis guna membuktikan secara empiris bahwa penambahan parameter perilaku mikroskopis (seperti durasi dan kecepatan klik) benar-benar memberikan kontribusi inkremental yang signifikan terhadap pemetaan pola evaluasi mahasiswa.

3.6 Metode Pembagian dan Augmentasi Data

Dataset akhir yang telah memuat keseluruhan variabel perilaku independen dan rekapitulasi nilai akhir dipartisi menggunakan pendekatan evaluasi yang terstruktur untuk memastikan keandalan prediksi. Total keseluruhan observasi (N) dalam penelitian ini bersumber dari 97 mahasiswa dikalikan dengan 30 sesi kuis. Namun, setelah dilakukan pembersihan terhadap sesi ujian yang tidak diikuti (data kosong), total populasi dataset riil yang dieksekusi menyusut menjadi N = 1.816 observasi tunggal. Pemisahan data ini merupakan standar mutlak dalam penelitian *machine learning* untuk memastikan model diuji pada data yang benar-benar baru dan terhindar dari bias penghafalan data (*overfitting*).

3.6.1 Validasi Silang (*k-Fold Cross-Validation*)

Dalam pendekatan validasi silang ini, keseluruhan populasi dataset *N*dibagi menjadi 5 himpunan bagian *fold* yang berukuran sama. Proses pelatihan dan evaluasi model akan diulang sebanyak 5 kali secara bergantian. Pada setiap iterasi, 4 *fold* digabungkan dan dialokasikan sebagai ruang belajar utama (*Training Set*) bagi algoritma XGBoost, sementara 1 *fold* sisanya diisolasi ketat secara bergilir sebagai data uji buta (*Testing Set*). Kinerja akhir algoritma XGBoost akan dilaporkan menggunakan nilai rata-rata beserta standar deviasinya (*mean* ± *std*) dari kelima hasil pengujian iterasi tersebut.

Pendekatan ini memastikan setiap titik data pernah dievaluasi secara murni, sehingga mencegah kebocoran informasi dan menghasilkan estimasi keandalan model yang jauh lebih kredibel di dunia nyata.

3.6.2 Simulasi Varian dengan Injeksi Derau Gaussian Terisolasi

Guna meningkatkan ketahanan algoritma XGBoost terhadap variasi perilaku, diterapkan teknik augmentasi sintetik berupa injeksi derau statis (*Static Gaussian Noise*) sebesar 5% khusus pada fitur numerik. Rumus matematis ini dapat dilihat pada persamaan (5)

$$x_{aug} = x_{asli} + \epsilon, \quad \epsilon \sim \mathcal{N}(0 \cdot 0{,}05 \cdot \sigma_x) \tag{5}$$

Keterangan:

x_{aug}: Nilai fitur data baru setelah ditambahkan gangguan (hasil augmentasi).

x_{asli}: Nilai fitur data riil atau asli milik mahasiswa.

ϵ(Epsilon): Nilai derau (*noise*) acak yang ditambahkan.

$\mathcal{N}$: Distribusi normal (Gaussian).

0: Rata-rata (*mean*) dari derau yang disuntikkan.

$0.05 \cdot \sigma_x$: Standar deviasi (simpangan baku) yang nilainya dibatasi sebesar 5% dari standar deviasi data aslinya.

Mengakomodasi prinsip validitas pengujian agar terhindar dari bias penghafalan data (*data leakage*), teknik perturbasi acak ini diimplementasikan secara terisolasi. Injeksi derau Gaussian dikonfigurasi secara mutlak hanya pada partisi data latih (*training set*), baik dalam skema *hold-out* maupun di dalam *fold* pelatihan pada skema validasi silang. Data uji (*testing set*) dibiarkan sepenuhnya menggunakan observasi riil murni tanpa manipulasi apa pun. Dengan demikian, metrik performa akhir yang dilaporkan oleh sistem tetap mencerminkan keandalan algoritma dalam memproses data dunia nyata.

3.7 Pelatihan Model Kecerdasan buatan

Arsitektur utama pembuat keputusan pada studi ini dikonstruksi menggunakan algoritma XGBoost yang diformat untuk pendekatan regresi angka XGBoost *Regressor*. Pendekatan ansambel ini sangat berbeda dengan algoritma pembuat pohon keputusan konvensional seperti *Random Forest* yang membangun banyak pohon secara bersamaan dan mandiri. Algoritma XGBoost justru mengadopsi paradigma sekuensial yang saling memperbaiki. Setiap iterasi pohon keputusan yang baru pada iterasi ke-t dikomputasi secara spesifik untuk meminimalisasi sisa kesalahan prediksi atau sisa galat *residual error* yang diwariskan dari pohon-pohon generasi sebelumnya. Proses optimasi fungsi selama fase pelatihan berlangsung mencakup proses pengurangan tingkat kerugian latih *training loss*, yang kemudian diimbangi dengan istilah regularisasi agar struktur pohon tidak menjadi terlalu rumit. Formulasi objektifnya dituliskan pada persamaan (6)

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \qquad (6)$$

Keterangan:

$\text{Obj}^{(t)}$: Nilai fungsi objektif total yang harus diminimalkan pada iterasi atau langkah ke-t.

$\sum_{i=1}^{n}$: Penjumlahan dari data ke-1 sampai dengan total data ke-n.

l : Fungsi kerugian (*loss function*) untuk mengukur seberapa jauh selisih antara nilai prediksi dan nilai aslinya.

y_i: Nilai target atau nilai aktual yang sebenarnya (skor riil Moodle).

$\hat{y}_i^{(t-1)}$: Hasil prediksi dari iterasi sebelumnya (sampai iterasi ke $t-1$).

$f_t(x_i)$: Fungsi pohon keputusan baru yang ditambahkan pada iterasi ke-t.

$\Omega(f_t)$: Suku regularisasi (*regularization term*) yang berfungsi untuk menghukum atau mengontrol tingkat kompleksitas model agar pohon tidak terlalu bercabang rumit.

Untuk mencegah model menjadi terlalu percaya diri pada satu pola tertentu dan mengontrol stabilitas strukturnya, faktor regularisasi penalti ditetapkan melalui persamaan (7)

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (7)$$

Keterangan:

$\Omega(f_t)$: Fungsi regularisasi atau penalti tingkat kerumitan untuk pohon keputusan pada iterasi ke-t.

γ (Gamma): Parameter penalti konstan untuk setiap penambahan simpul daun (*leaf*) baru. Nilai ini mengontrol batas maksimal tingkat kerumitan cabang arsitektur pohon.

T: Jumlah total simpul akhir atau daun (*leaves*) yang terbentuk pada pohon keputusan tersebut.

λ (Lambda): Parameter penalti konstan (regularisasi L2) yang didistribusikan untuk mengontrol bobot keputusan.

w_j: Bobot parameter atau skor keputusan (*weight*) pada daun ke-j. Penalti diterapkan pada bobot ini untuk mencegah munculnya keputusan yang terlalu ekstrem.

$\sum_{j=1}^{T} w_j^2$: Jumlah kuadrat dari seluruh bobot daun pada pohon keputusan.

Dalam konteks perhitungan ini, Ω bertugas meregulasi batas maksimal tingkat kerumitan cabang arsitektur daun (T), sembari mendistribusikan penalti konstan (λ) pada bobot parameter yang memicu keputusan ekstrem (w_j). Penentuan parameter dasar ini dikalibrasi secara hati-hati melalui pengujian berulang untuk mencegah model berhenti belajar terlalu dini atau biasa disebut konvergensi prematur.

3.8 Implementasi XAI (TreeSHAP)

Inti dari kebaruan dan inovasi analitik metodologi ini bertumpu pada penerapan metode *Shapley Additive Explanations* yang dirancang secara khusus untuk arsitektur berbasis pohon keputusan TreeSHAP. Kerangka kerja ini sengaja diimplementasikan untuk mendekonstruksi sekaligus menerangi sifat *black-box* yang selalu menjadi kelemahan utama algoritma komputasi tingkat tinggi seperti XGBoost.

Metode TreeSHAP sejatinya diadaptasi dari teori permainan kooperatif di bidang ekonomi. Metode ini menjamin terjadinya pembagian nilai yang adil dan mutlak terhadap hasil akhir prediksi dari sebuah observasi. Sebagai contoh yang mudah dipahami: apabila sistem memprediksi bahwa skor kuis seorang mahasiswa adalah 75, maka TreeSHAP akan membedah angka tersebut dan mendistribusikan selisih nilainya ke setiap atribut perilaku. Sistem akan mencari tahu berapa poin yang disumbangkan oleh fitur kecepatan interaksi, dan berapa poin yang dikurangi akibat pengerjaan di jam larut malam. Formulasi perhitungan nilai atribusi Shapley untuk setiap metrik dieksekusi secara matematis mengacu pada rumusan yang telah diuraikan pada Persamaan (1) di Bab II.

Melalui ekstraksi hitungan dari nilai ϕ ini, sistem akhirnya memiliki kapabilitas analitik untuk menunjuk secara pasti fitur mana yang bertindak sebagai penguat nilai positif bagi mahasiswa, serta fitur kebiasaan mana yang berdampak buruk sebagai faktor reduksi penalti. Deretan nilai matematis inilah yang selanjutnya diposisikan sebagai landasan logika yang objektif, sebelum nantinya ditransformasikan secara menyeluruh menjadi teks umpan balik di antarmuka sistem.

3.9 Metode Pengujian dan Evaluasi

Mengingat tujuan akhir sistem ini adalah bertindak sebagai lensa analitik dan bukan sebagai prediktor nilai di lingkungan produksi, pengujian performa algoritma regresi tetap mutlak diperlukan sebagai syarat pra-eksplanasi (pra-XAI). Evaluasi ini bertujuan untuk membuktikan bahwa model (XGBoost) telah mencapai tingkat kecocokan (*fitness*) yang memadai dan berhasil memetakan pola secara logis

sebelum logika internalnya dibedah oleh TreeSHAP. Kualitas pemetaan model dievaluasi secara murni berdasarkan kemampuannya memprediksi nilai kuis aktual dengan tingkat meleset yang sekecil mungkin pada himpunan data uji. Parameter uji statistik yang dimanfaatkan meliputi:

3.9.1 Mean Absolute Error (MAE)

Metrik MAE ini digunakan untuk menghitung rata-rata selisih antara nilai kuis yang diprediksi oleh model dengan nilai asli yang didapatkan mahasiswa. Kelebihan dari MAE adalah tidak memberikan beban atau penalti berlebih pada tebakan yang meleset terlalu jauh (*outlier*). Oleh karena itu, metrik ini sangat cocok untuk memberikan gambaran sederhana yang mudah dipahami tentang seberapa jauh jarak melesetnya tebakan model secara rata-rata. Sebagai contoh, jika nilai MAE yang dihasilkan adalah 5, hal itu berarti perkiraan dari sistem rata-rata meleset sebanyak 5 poin dari skor asli kuis mahasiswa. Rumus MAE sebagai persamaan (8)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (8)$$

Keterangan:

MAE merepresentasikan besaran rata-rata tingkat kesalahan atau penyimpangan prediksi model secara keseluruhan.

n menunjukkan jumlah total data observasi atau banyaknya sesi kuis yang sedang diuji.

$\sum_{i=1}^{n}$ merupakan simbol sigma yang menandakan operasi penjumlahan secara beruntun dari data observasi pertama ($i = 1$) hingga data terakhir ke-n.

$|\ldots|$ melambangkan operasi nilai mutlak (absolut) yang berfungsi untuk memastikan semua hasil selisih bernilai positif, sehingga tebakan yang meleset ke atas maupun ke bawah tidak saling meniadakan saat dijumlahkan.

y_i mewakili nilai target aktual, yaitu skor asli yang benar-benar didapatkan oleh mahasiswa pada evaluasi atau kuis ke-i.

$\hat{y}_i$ adalah nilai hasil tebakan atau skor prediksi yang dikalkulasikan oleh algoritma model pada kuis ke-i.

$\frac{1}{n}$ berfungsi sebagai operasi pembagian akhir di mana total keseluruhan selisih mutlak yang telah dijumlahkan akan dibagi dengan banyaknya jumlah observasi untuk mendapatkan rata-rata galat murni.

3.9.2 Root Mean Square Eror (RMSE)

Berbeda dengan MAE, nilai RMSE diperoleh melalui kalkulasi akar kuadrat dari rata-rata galat prediksi yang telah dikuadratkan terlebih dahulu. Karakteristik utama dari perhitungan RMSE adalah tingkat sensitivitasnya yang sangat tinggi dan ketat terhadap kesalahan prediksi yang berskala besar (*outlier*). Oleh karena itu, metrik ini menjadi filter pengujian yang jauh lebih riil dan tegas. Semakin kecil nilai RMSE yang berhasil dicapai, maka semakin tinggi pula tingkat keandalan dan konsistensi dari sistem regresi yang dibangun. Di Rumuskan sebagai persamaan (9)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (9)$$

Keterangan:

RMSEmerupakan nilai akhir dari tingkat kesalahan akar kuadrat rata-rata, di mana semakin kecil nilainya maka semakin tinggi tingkat keandalan dan konsistensi dari model regresi yang dibangun.

$\sqrt{...}$ adalah operasi akar kuadrat yang berfungsi untuk mengembalikan besaran nilai galat yang sebelumnya telah dikuadratkan agar kembali ke satuan rentang nilai aslinya.

n menunjukkan jumlah total data observasi atau banyaknya sesi kuis yang sedang diuji.

$\sum_{i=1}^{n}$ merupakan simbol sigma yang menandakan operasi penjumlahan secara beruntun dari selisih kuadrat data pertama ($i = 1$) hingga data terakhir ke-n.

y_i mewakili nilai target aktual, yaitu skor asli yang benar-benar didapatkan oleh mahasiswa pada evaluasi atau kuis ke-i.

$\hat{y}_i$ adalah nilai hasil tebakan atau skor prediksi yang dikalkulasikan oleh algoritma model pada kuis ke-i.

$(y_i - \hat{y}_i)^2$ adalah operasi pengkuadratan dari selisih antara nilai aktual dan prediksi, yang berfungsi untuk memastikan semua hasil selisih bernilai positif sekaligus memberikan beban kesalahan yang jauh lebih besar pada prediksi yang meleset sangat jauh (outlier).

$\frac{1}{n}$ berfungsi sebagai operasi pembagian untuk menghitung nilai rata-rata dari total keseluruhan selisih yang telah dikuadratkan tersebut sebelum hasil akhirnya diakarkan.

3.9.3 Koefisien Determinasi (R^2)

Parameter pengujian R-Squared dimanfaatkan khusus untuk mengevaluasi seberapa besar persentase variansi pada target luaran (skor ujian) yang berhasil dijelaskan keberadaannya oleh kombinasi prediktor log aktivitas. Rentang nilai metrik ini bergerak antara 0 hingga 1. Jika nilai R^2 bergerak secara positif, hal tersebut membuktikan adanya korelasi logis yang bermakna (meskipun secara parsial) antara kebiasaan mengeklik mahasiswa di dalam Moodle dengan hasil pencapaian akademis faktual mereka. Di Rumuskan pada persamaan (10)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (10)$$

Keterangan:

R^2 merupakan nilai Koefisien Determinasi yang rentang nilainya bergerak antara 0 hingga 1, di mana nilai yang bergerak secara positif membuktikan adanya korelasi logis yang bermakna antara kebiasaan mengeklik mahasiswa dengan hasil pencapaian akademis faktual mereka.

1 adalah angka konstanta pengurang, yang melambangkan kondisi model yang sempurna (jika tingkat kesalahan prediksi bernilai nol, maka hasil akhir R^2 adalah 1).

n menunjukkan jumlah total data observasi atau banyaknya sesi kuis yang sedang diuji.

$\sum_{i=1}^{n}$ merupakan simbol sigma yang menandakan operasi penjumlahan secara beruntun dari data pertama ($i = 1$) hingga data terakhir ke-n.

$y_i$mewakili nilai target aktual, yaitu skor asli yang benar-benar didapatkan oleh mahasiswa pada evaluasi atau kuis ke-i.

$\hat{y}_i$ adalah nilai hasil tebakan atau skor prediksi yang dikalkulasikan oleh algoritma model pada kuis ke-i.

$\bar{y}$ (dibaca *y-bar*) melambangkan nilai rata-rata keseluruhan dari skor aktual yang didapatkan oleh mahasiswa.

$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ adalah pembilang yang menghitung total kuadrat dari selisih antara nilai aktual dan nilai prediksi (mengukur seberapa besar sisa kesalahan dari model).

$\sum_{i=1}^{n}(y_i - \bar{y})^2$ adalah penyebut yang menghitung total kuadrat dari selisih antara nilai aktual dengan nilai rata-rata keseluruhan (mengukur total variansi atau sebaran data aslinya).

3.9.4 Estimasi Akurasi Prediksi (Estimated Accuracy)

Mengingat model yang dibangun merupakan algoritma regresi berkelanjutan yang secara fundamental tidak memiliki definisi metrik klasifikasi akurasi yang absolut, penelitian ini menetapkan metrik akurasi estimasi heuristik. Metrik ini dikalkulasikan dengan mengukur seberapa besar persentase simpangan galat rata-rata MAE terhadap rata-rata capaian nilai asli secara keseluruhan ($\bar{y}$). Persamaan matematis untuk perhitungan akurasi estimasi ini dirumuskan sebagai persamaan (11)

$$Accuracy = 100\% - \left(\frac{MAE}{\bar{y}} \times 100\%\right) \quad (11)$$

Keterangan:

$Accuracy$ adalah metrik akurasi estimasi akhir dari model prediksi yang direpresentasikan dalam bentuk persentase.

100% merupakan angka konstanta dasar yang melambangkan tingkat akurasi sempurna jika model sama sekali tidak memiliki kesalahan tebakan.

MAE adalah nilai Mean Absolute Error, yaitu besaran persentase simpangan galat rata-rata atau rata-rata tingkat melesetnya tebakan model yang telah dihitung sebelumnya.

$\bar{y}$ (dibaca *y-bar*) merupakan rata-rata capaian nilai asli secara keseluruhan dari seluruh mahasiswa.

$\frac{MAE}{\bar{y}} \times 100\%$ adalah operasi kalkulasi untuk mengukur persentase rasio besaran simpangan galat terhadap rata-rata capaian nilai aktual kelas. Hasil persentase persentase penyimpangan inilah yang kemudian dikurangkan dari nilai kesempurnaan 100% untuk mendapatkan angka estimasi akurasi akhir dari prediksi sistem.

Pendekatan rasio simpangan ini digunakan untuk memberikan interpretasi yang lebih ramah pengguna (*user-friendly)* mengenai seberapa dekat prediksi sistem dengan realitas sebaran nilai kelas secara persentase, mendampingi pembacaan batas galat MAE dan RMSE.

3.10 Metode Pembuktian dan Luaran Sistem (*Proving Method*)

Verifikasi kelayakan dari sistem yang dibangun tidak hanya bergantung semata pada pencapaian matematis yang memuaskan dari pengujian batas galat MAE dan RMSE di atas kertas. Keberhasilan implementasi sesungguhnya dari konsep AI *Companion* ini dibuktikan langsung melalui fungsionalitas operasional nyata pada

pendekatan pembuatan teks berbasis templat (*Template-Based Generation*) dan logika kondisional (*Rule-Based*) yang dibungkus dengan kerangka aplikasi web Gradio. Untuk membuktikan bahwa sistem dapat digunakan dengan baik, proses pengujian akhir dilaksanakan melalui skenario rantai eksekusi kontinu sebagai berikut:

1. Pengujian Integrasi Masukan Data: Berkas log observasi baru yang murni berada di luar lingkungan data eksperimen diunggah oleh penguji dan diumpankan secara langsung ke dalam modul antarmuka prototipe aplikasi Gradio.

2. Pemetaan Logika Keputusan (*Logic Routing Validation*): Pada tahap ini, eksekusi mesin di belakang layar akan ditinjau. Pengujian difokuskan untuk melihat apakah matriks kalkulasi TreeSHAP yang menghasilkan sentimen nilai ekstrem khususnya nilai negative secara konsisten berhasil memicu rantai logika parameter kondisional (*if-else*) untuk memberikan teguran. Sebagai contoh nyata: apabila muncul nilai penalti ekstrem akibat durasi penyelesaian yang terlalu singkat, sistem harus secara presisi memanggil blok teks yang memberikan teguran mengenai indikasi pengerjaan yang terburu-buru, lalu memberikan saran yang relevan terkait manajemen waktu ujian.

3. Evaluasi Koherensi dan Fungsionalitas Teks: Mengingat arsitektur ini mentransformasikan angka log ke dalam bahasa alami, pengujian di lapisan paling luar difokuskan pada penilaian koherensi tata bahasa Indonesia dan fungsionalitas logika kalimat yang dihasilkan. Prototipe ini dinyatakan berhasil apabila mesin secara otomatis mampu mengonversi metrik TreeSHAP menjadi sebuah paragraf evaluasi yang utuh secara semantik, tidak terpotong, dan secara fungsional mampu menyampaikan pesan formatif yang dapat dipahami alur rasionalisasinya. Pengujian tingkat lanjut yang berpusat pada manusia (*human-grounded evaluation*) secara ekstensif menggunakan instrumen survei responden untuk mengukur tingkat kepercayaan (*trust*) berada di luar batasan studi ini dan diklasifikasikan sebagai area pengembangan selanjutnya..

3.11 Lokasi dan Waktu Penelitian

Pelaksanaan penelitian ini, mulai dari tahap observasi awal, pengumpulan data sekunder, hingga komputasi dan evaluasi model *machine learning*, dilakukan di lingkungan akademik Pengolahan data dan simulasi pemodelan dilakukan secara intensif di Laboratorium Komputasi, Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry, Banda Aceh. Pengambilan data log aktivitas digital difokuskan pada aktivitas mahasiswa yang terdaftar pada mata kuliah Keamanan Jaringan. Periode penelitian berlangsung selama satu semester, mencakup tahap ekstraksi data dari *Learning Management System* (LMS) Moodle hingga validasi luaran sistem pada prototipe AI *Companion*.

3.12 Alat dan Bahan

Penelitian ini menggunakan serangkaian perangkat keras dan perangkat lunak yang secara spesifik dipilih untuk menangani beban komputasi pemrosesan data, pelatihan model XGBoost, dan kalkulasi matriks SHAP.

3.12.1 Perangkat Keras

Infrastruktur perangkat keras yang digunakan untuk menjalankan lingkungan komputasi lokal dan simulasi pengujian memiliki spesifikasi pada table 3.1.

Tabel 3. 1 Perangkat Keras

Komponen	Spesifik Minimun	Fungsi
Prosesor (CPU)	*Multi-core processor* (Min. 4 Cores, 8 Threads)	Memproses komputasi algoritma *ensemble* dan kalkulasi TreeSHAP secara paralel.
Memori (RAM)	16 GB DDR4	Mengakomodasi beban muatan dataset agregasi ke dalam memori kerja (*in-memory processing*).

Komponen	Spesifik Minimun	Fungsi
Penyimpanan	512 GB Solid State Drive (SSD)	Mempercepat proses I/O (*Input/Output*) saat pembacaan dan penulisan berkas CSV dan *model caching*.

3.12.2 Perangkat Lunak

Lingkungan pengembangan perangkat lunak dikonfigurasi untuk meminimalisasi *overhead* memori pada *background process* guna memaksimalkan alokasi sumber daya pada komputasi *machine learning*. Detail instrumen perangkat lunak beserta versinya dijabarkan pada tabel 3.2.

Tabel 3. 2 Perangkat Lunak

Perangkat Lunak	Versi	Kegunaan dalam Penelitian
Windows	10	Sistem operasi berbasis Windows yang ringan dan hemat sumber daya
Python	3.11.x	Bahasa pemrograman utama untuk pengembangan *pipeline machine learning* dan *backend* sistem.
XGBoost	2.0.3	Pustaka algoritma *Gradient Boosting* untuk pemodelan prediksi skor mahasiswa.
SHAP	0.44.0	Pustaka *Explainable AI* (XAI) untuk

Perangkat Lunak	Versi	Kegunaan dalam Penelitian
		membongkar logika *black-box* model dan menghitung nilai atribusi fitur.
Pandas	2.1.4	Pemrosesan, pembersihan, dan manipulasi struktur data log mentah ke dalam bentuk *Dataframe*.
Scikit-Learn	1.3.3	Pustaka untuk pembagian dataset, augmentasi standar, dan kalkulasi metrik evaluasi (MAE, RMSE, R-Squared).
Gradio	4.19.2	Kerangka kerja *frontend* untuk merancang antarmuka *AI Companion* berbasis web.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Karakteristik Populasi Data dan Profil Perilaku

Tahapan pra-pemrosesan data secara komprehensif telah menyaring dan mengonsolidasikan ribuan baris rekaman aktivitas mentah (log) Moodle menjadi satu matriks data tunggal yang siap dianalisis. Dataset akhir ini secara eksklusif hanya memuat data perilaku mahasiswa pada saat pelaksanaan ujian atau kuis, mengeliminasi segala bentuk aktivitas navigasi platform yang tidak relevan dengan proses evaluasi kognitif.

Populasi data dalam penelitian ini bersumber dari rekam jejak 97 mahasiswa yang terdaftar pada mata kuliah Keamanan Jaringan, dengan cakupan 30 sesi kuis yang berbeda. Pada kondisi aslinya, platform Moodle menghasilkan 106.611 baris log mentah, di mana setiap tindakan klik (*event*) dicatat sebagai satu baris terpisah. Melalui proses agregasi sesi dan pembersihan data kosong (*null drop*), ratusan ribu baris log tersebut dipadatkan menjadi rangkuman fitur perilaku tunggal per mahasiswa untuk setiap kuis (secara teoretis: 97 mahasiswa × 30 kuis = 2.910 sesi). Mengingat adanya sesi kuis yang tidak diikuti oleh mahasiswa, total populasi data riil yang tervalidasi menyusut menjadi 1.816 observasi tunggal. Rincian distribusi tersebut disajikan pada Tabel 4.1.

Tabel 4. 1 Distribusi Populasi Dataset Akhir

Deskripsi Populasi	Jumlah
Total Mahasiswa Terdaftar	97 Mahasiswa
Total Sesi Kuis Dievaluasi	30 Kuis
Total Observasi Log Mentah Moodle	106.611 Baris Log
Total Ekspektasi Teoritis (97 × 30)	2.910 Sesi
Total Data Riil Tervalidasi (Tanpa Null)	1.816 Observasi

Dari 1.816 observasi riil tersebut, data kemudian dipartisi. Pada tahapan pelatihan (*training*), proporsi data latih digandakan secara sintetik menggunakan injeksi *Gaussian Noise* untuk memperluas ruang belajar algoritma. Namun, proses augmentasi ini dibatasi secara ketat hanya pada ruang pelatihan. Data yang diumpankan ke dalam model untuk pengujian performa akhir (*testing*) tetap menggunakan populasi data riil yang belum pernah disentuh oleh algoritma augmentasi, sehingga mencegah terjadinya kebocoran data (*data leakage*).

4.1.1 Analisis Deskriptif Perilaku Evaluasi Mahasiswa

Sebelum data diumpankan ke dalam algoritma XGBoost, sangat penting untuk memahami anatomi dan profil perilaku mahasiswa secara statistik. Pendekatan *Educational Data Mining* (EDM) memungkinkan sistem untuk menelusuri kebiasaan peserta didik melalui metrik kuantitatif. Ringkasan statistik deskriptif dari fitur-fitur perilaku utama disajikan pada Tabel 4.2.

Tabel 4. 2 Statistik Deskriptif Metrik Perilaku Kuis

Fitur Perilaku	Nilai Rata-rata (Mean)	Nilai Tengah (Median)	Minimum	Maksimum
Durasi Pengerjaan (T_Δ)	106,41 Menit	180 Menit	0 Menit	180 Menit
Kecepatan Klik (V_c)	1,21 KPM	0,2 KPM	0 KPM	40 KPM
Variansi IP (*IP Variance*)	2,22 IP	2 IP	1 IP	14 IP

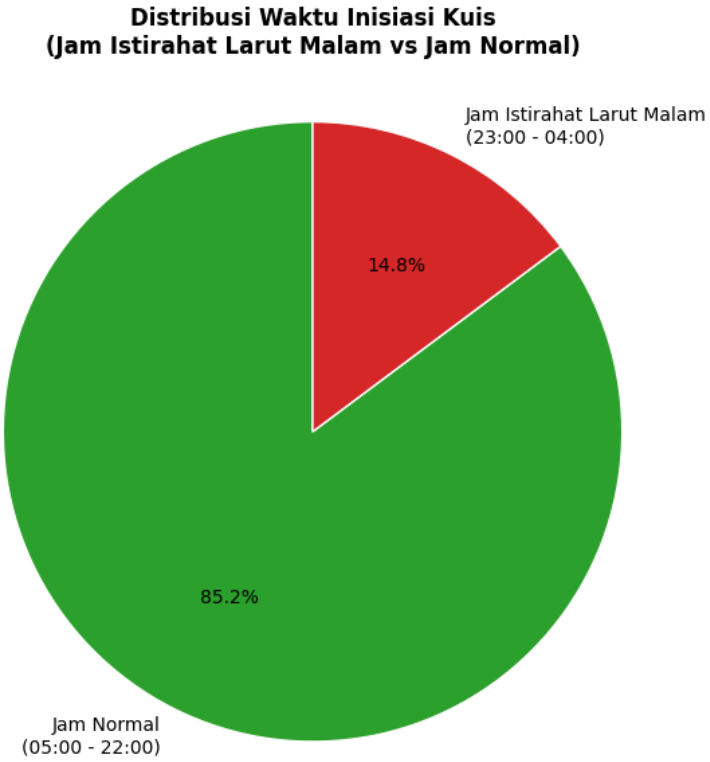
Berdasarkan Tabel 4.2, profil pengerjaan ujian mahasiswa di mata kuliah Keamanan Jaringan menunjukkan variasi strategi yang sangat dinamis. Metrik Durasi (T_Δ) mencerminkan dedikasi waktu penyelesaian yang beragam, mulai dari penyelesaian instan dalam dibawah 1 menit yang mengindikasikan

tindakan impulsif atau *rapid-guessing*, hingga pengerjaan analitis yang memakan waktu maksimal 180 menit.

Sejalan dengan durasi, metrik Kecepatan Klik (V_c) memberikan pandangan mikroskopis terhadap interaksi antarmuka mahasiswa. Nilai rata-rata sebesar 1,21 klik per menit menunjukkan ritme pengerjaan kelas yang normal. Namun, terdapat anomali di mana nilai kecepatan maksimum menyentuh angka 40 klik per menit, mengindikasikan adanya kelompok mahasiswa yang sekadar melakukan navigasi cepat antar soal tanpa analisis mendalam.

Pada observasi konsistensi jaringan (Variansi IP), nilai rata-rata menunjukkan angka 2,22 IP. Mayoritas observasi sebagaimana terlihat dari nilai median menunjukkan bahwa mahasiswa menggunakan satu alamat IP yang stabil (1 IP) selama kuis berlangsung. Namun, terekam pula fluktuasi maksimal hingga 14 IP unik dalam satu sesi ujian tunggal. Dalam konteks penelitian ini, tingginya pergantian alamat IP ditafsirkan secara ketat sebagai indikator fluktuasi stabilitas koneksi misalnya mahasiswa yang terputus dari Wi-Fi dan beralih ke data seluler pribadi, dan sama sekali bukan alat ukur deterministik untuk menuduh terjadinya kolusi atau kecurangan akademik.

Selain interaksi teknis, dimensi waktu inisiasi kuis Jam Larut Malam juga dievaluasi untuk memetakan preferensi jadwal belajar mahasiswa, sebagaimana divisualisasikan pada Gambar 4.1.



Gambar 4. 1 Distribusi Waktu Inisiasi Kuis

Berdasarkan visualisasi pada Gambar 4.1, terlihat bahwa sebesar 14,8% observasi pengerjaan kuis diinisiasi pada rentang waktu larut malam (antara pukul 23:00 hingga 04:00 WIB), sementara 85,2% sisanya dikerjakan pada jam operasional normal. Fakta empiris ini mengonfirmasi bahwa fleksibilitas LMS Moodle yang mengusung metode jendela terbuka (*open-window*) dimanfaatkan secara maksimal oleh mahasiswa. Keberadaan proporsi pengerjaan larut malam ini menjadi variabel menarik yang nantinya akan diukur signifikansi korelasinya oleh XGBoost terhadap potensi penurunan nilai akibat kelelahan ritme sirkadian.

4.2 Kinerja Komputasi dan Analisis Korelatif Asosiatif

Pemaparan hasil pengujian dan pembedahan korelasi asosional antar fitur diteliti lebih lanjut. Pendekatan komputasi ini tidak ditujukan untuk menetapkan hukum sebab-akibat secara mutlak, melainkan untuk menyingkap pola korelasi asosiatif antara perilaku pengerjaan kuis digital mahasiswa di dalam LMS dengan capaian evaluasi akhir mereka.

4.2.1 Konstruksi dan Evaluasi Model Prediktif

Pembangunan arsitektur pembuat keputusan pada tahap komputasi diimplementasikan menggunakan fungsi pustaka XGBoost Regressor. Guna mencegah model melakukan penghafalan buta terhadap data (*overfitting*) dan memastikan distribusi bobot keputusan berjalan optimal, diterapkan konfigurasi *hyperparameter* secara spesifik sebagaimana diilustrasikan pada Gambar 4.2.

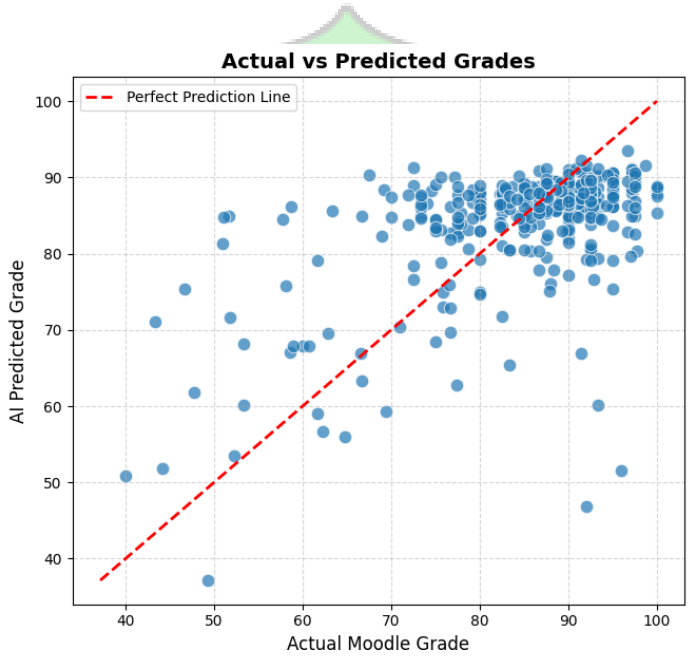
```
Python
# Inisialisasi parameter model XGBoost Regressor
model = xgb.XGBRegressor(
    n_estimators=400,
    max_depth=6,
    learning_rate=0.03,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=0.2,
    reg_lambda=2.0,
    random_state=42
)
```

Gambar 4. 2 Konfigurasi Hyperparameter Algoritma XGBoost Regressor

Pada konfigurasi di atas, kedalaman percabangan pohon dibatasi max_depth=6 dan dikendalikan oleh penalti regularisasi L1 reg_alpha=0.2 serta L2 reg_lambda=2.0. Selanjutnya, untuk memenuhi standar pengujian *machine learning* pada volume data yang terbatas, model divalidasi menggunakan skema *k-Fold Cross-Validation*. Kinerja model diukur secara kuantitatif yang hasil rata-ratanya direkapitulasi pada Tabel 4.3.

Tabel 4. 3 Hasil Evaluasi Kinerja Kuantitatif (k-Fold Cross-Validation)

Metrik Evaluasi	Nilai Capaian	Keterangan
Mean Absolute Error (MAE)	6.91 Poin	Rata-rata absolut penyimpangan tebakan.
Root Mean Squared Error (RMSE)	9.51 Poin	Sensitivitas model terhadap tebakan meleset (*outlier*).
R-Squared (R^2)	0.26	Persentase variansi capaian yang mampu dijelaskan.
Estimasi Akurasi	91.82%	Persentase kedekatan prediksi terhadap rata-rata nilai kelas.



Gambar 4. 3 Pemetaan Scatter Plot Sebaran Nilai

Hasil evaluasi pada Tabel 4.3 menunjukkan metrik MAE sebesar 6,86 poin dan RMSE sebesar 9,44 poin. Dalam konteks penelitian ini, angka galat (*error*) tersebut tidak difungsikan untuk mengklaim atau membanggakan kemampuan prediksi sistem di masa depan, melainkan sebagai bukti empiris bahwa model XGBoost memiliki tingkat kecocokan (*fitness*) yang memadai terhadap data historis. Dengan batas toleransi kesalahan yang stabil dan tidak menyimpang ekstrem ini, model dinyatakan layak dan valid untuk dilanjutkan ke tahap

ekstraksi pola melalui kerangka kerja XAI. Signifikansi asosiasi perilaku mikroskopis ini lebih lanjut dikonfirmasi oleh nilai metrik R^2 sebesar 0,27.

4.2.2 Uji Ablasi (*Ablation Study*) Atribut Skor Historis

Mengingat kepandaian dasar variabel Skor Historis secara natural memiliki pengaruh sangat besar terhadap capaian nilai mahasiswa, dilakukan uji ablasi untuk membuktikan bahwa fitur perilaku mikro yang diekstraksi dari Moodle tidaklah sia-sia. Pengujian ini membandingkan kinerja model saat dilatih menggunakan seluruh fitur, dengan saat fitur historis dicabut secara paksa dari memori algoritma.

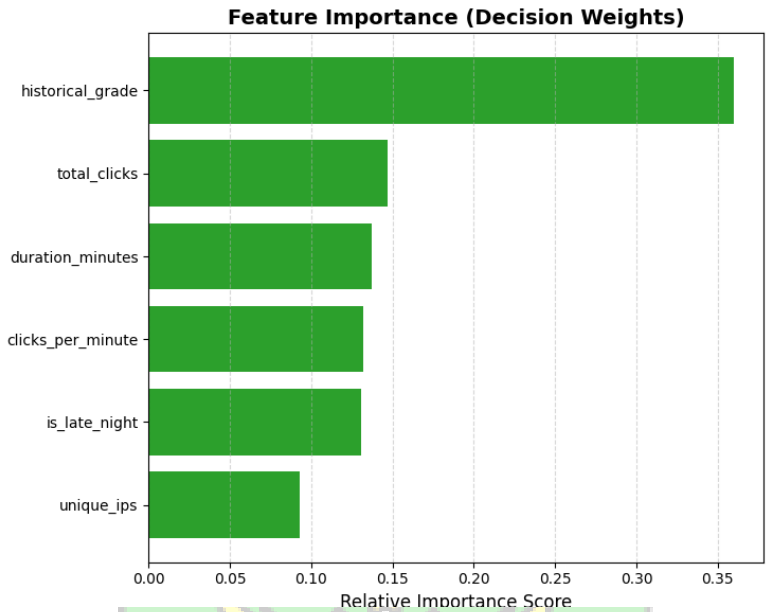
Tabel 4. 4 Komparasi Kinerja Algoritma

Skenario Pengujian	MAE	RMSE	R2
Model Utuh (Seluruh Fitur)	6.91	9.51	0.26
Model Ablasi (Tanpa Skor Historis)	8.35	11.41	-0.07

Hasil komparasi pada Tabel 4.4 mengonfirmasi bahwa metrik performa algoritma mengalami fluktuasi signifikan ketika variabel Skor Historis dihilangkan terlihat dari nilai R^2 yang anjlok ke angka negatif. Uji ablasi ini membuktikan secara empiris bahwa penambahan parameter perilaku mikroskopis durasi, kecepatan klik memiliki kontribusi operasional yang krusial bagi sistem, namun mereka tidak dapat berdiri sendiri tanpa adanya titik acuan kemampuan kognitif dasar mahasiswa. Kecerdasan buatan mengandalkan sinkronisasi di antara kedua dimensi tersebut untuk menyajikan umpan balik evaluasi yang logis dan objektif.

4.2.3 Pembedahan Pola Korelasi Perilaku

Untuk memahami secara spesifik faktor perilaku mana yang paling dominan dipertimbangkan oleh algoritma, dilakukan ekstraksi bobot prioritas pohon keputusan, sebagaimana divisualisasikan pada Gambar 4.4.



Gambar 4. 4 Visualisasi Global Feature Importance XGBoost

Diagram pemeringkatan di atas memberikan makna edukasional yang mendalam terhadap proses evaluasi kuis digital. Di bawah variabel kepandaian dasar (*historical_grade*), fitur durasi penyelesaian *duration_minutes* dan intensitas klik *total_clicks* serta *clicks_per_minute* mendominasi peringkat teratas.

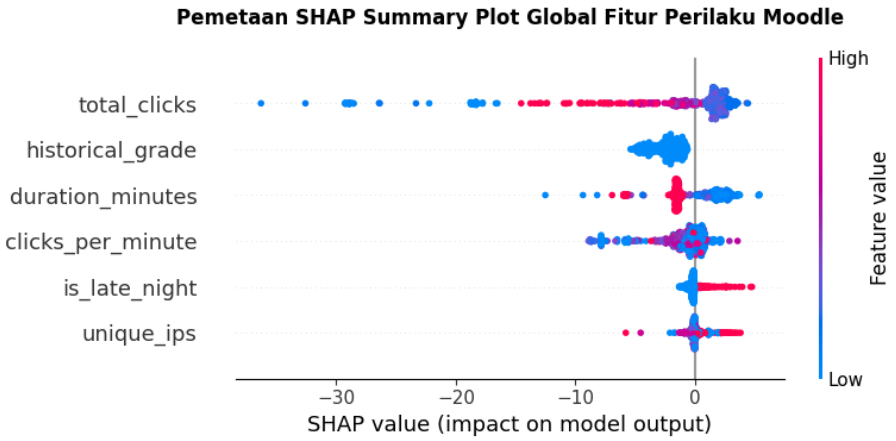
Hal ini membuktikan secara faktual bahwa durasi pengerjaan sangat merepresentasikan manajemen waktu dan tingkat keseriusan mahasiswa; mereka yang mengalokasikan waktu ideal cenderung memperoleh hasil yang lebih presisi. Sementara itu, kecepatan interaksi klik menjadi parameter krusial untuk mengukur kedalaman analisis. Rasio klik per menit yang terlampau tinggi sangat berkorelasi dengan taktik *rapid-guessing* (tebakan instan tanpa membaca soal secara utuh), yang pada akhirnya berasosiasi kuat dengan probabilitas penurunan capaian nilai kuis mahasiswa.

4.3 Transparansi Penilaian melalui *Explainable AI*

Inti dari inovasi penelitian yang diajukan di mana fokus dialihkan dari sekadar akurasi komputasi menuju keterpahaman model (*explainability*). Implementasi kerangka kerja XAI bertujuan untuk membongkar sifat *black-box* dari algoritma XGBoost yang rumit. Melalui pendekatan ini, sistem tidak lagi diposisikan sebagai "hakim" kaku yang hanya mengeluarkan skor angka mutlak, melainkan bertindak sebagai lensa analitik transparan yang mengungkap bagaimana profil perilaku pengerjaan mahasiswa memengaruhi penilaian, untuk kemudian disajikan sebagai bahan refleksi akademik yang formatif.

4.3.1 Ekstraksi Nilai Atribusi TreeSHAP

Untuk mengekstrak kontribusi aditif dari setiap fitur perilaku mikroskopis secara adil, penelitian ini menerapkan variasi algoritmik TreeSHAP yang diintegrasikan langsung pada struktur pohon keputusan XGBoost. Kalkulasi TreeSHAP menghasilkan nilai kontribusi numerik nilai Shapley yang mendefinisikan polaritas arah serta kekuatan pengaruh dari masing-masing variabel perilaku terhadap pergeseran prediksi nilai dari garis dasar (*baseline/base value*). Distribusi kontribusi fitur secara global divisualisasikan melalui grafik *SHAP Summary Plot* pada Gambar 4.4.



Gambar 4. 5 Pemetaan SHAP Summary Plot Global Fitur Perilaku Moodle

Analisis visual pada Gambar 4.5 menunjukkan polaritas arah pengaruh yang sangat tegas dari setiap variabel perilaku kuis mahasiswa:

1. Polaritas Nilai Atribusi Durasi (T_{Δ}): Berdasarkan sebaran titik sebaran SHAP, terungkap batas matematis yang konsisten di mana durasi pengerjaan yang terlalu pendek atau cepat ditandai dengan warna biru pada sebaran fitur secara konsisten bergeser ke arah margin negative. Fenomena ini memberikan bukti empiris bahwa durasi pengerjaan yang berada di bawah ambang batas ideal akan langsung dibaca oleh model sebagai faktor reduksi nilai memberikan penalti skor. Sebaliknya, alokasi waktu yang optimal dan lebih Panjang titik merah memberikan atribusi positif yang mendorong nilai prediksi mendekati atau melampaui skor historis dasar mahasiswa.
2. Polaritas Kecepatan Klik (V_c): Fitur rasio klik per menit (*clicks_per_minute*) menunjukkan pola polaritas ekstrem. Ketika kecepatan klik bergerak terlalu tinggi melewati batas normal, nilai SHAP bergeser drastis ke area negatif. Karakteristik sporadis ini mengonfirmasi deteksi taktik *rapid-guessing* pengisian jawaban secara instan tanpa proses membaca teks soal secara mendalam.
3. Polaritas Jam Larut Malam (*Is_late_night*): Variabel biner ini menunjukkan pergeseran bobot penalti minor namun konisten bagi observasi yang bernilai 1 mengerjakan kuis antara pukul 23:00 hingga 04:00 WIB, yang mengindikasikan adanya asosiasi degradasi konsentrasi akibat gangguan ritme sirkadian kognitif pada sebagian besar populasi mahasiswa.

4.3.2 Implementasi dan Evaluasi Prototipe AI Companion

Matriks angka kaku dan nilai atribusi kontribusi kontinu yang dihasilkan oleh TreeSHAP selanjutnya dikonversi menjadi bahasa alami manusia (*Natural Language*) yang edukatif. Penerapan transformasi ini dibangun menggunakan pendekatan hibrida yang mengombinasikan kalkulasi TreeSHAP dengan mesin logika kondisional berbasis aturan (*Rule-Based & Template Generation*). Langkah ini diambil secara sengaja untuk memastikan luaran umpan balik

bersifat faktual dan terhindar dari bias halusinasi teks. Logika perutean ambang batas (*threshold routing*) tersebut ditunjukkan pada Gambar 4.6.

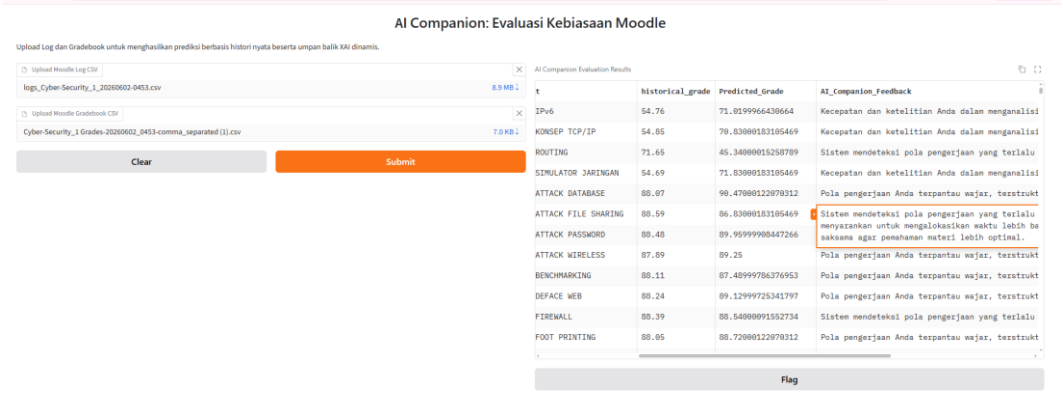
```
Python
# Eksekusi rantai if-else berbasis sentimen margin Shapley
if speed_shap < -1.5:
   feedback.append("Sistem mendeteksi pola pengerjaan yang terlalu tergesa-gesa, alokasikan waktu lebih banyak guna membaca dan menganalisis setiap soal dengan saksama...")
elif speed_shap > 1.5:
   feedback.append("Kecepatan dan ketelitian Anda dalam menganalisis soal sangat baik. Pertahankan pola pengerjaan yang stabil dan analitis ini...")

if duration_shap < -2.0:
   feedback.append("Durasi penyelesaian kuis Anda tercatar jauh di bawah standar waktu ideal, manfaatkan waktu yang tersedia secara penuh...")

if night_shap < -1.0:
   feedback.append("Anda mengerjakan kuis ini pada jam istirahat larut malam. Secara medis, penurunan performa kognitif sangat mungkin terjadi.")
```

Gambar 4. 6 Rule-Based Generation Berbasis Sentimen SHAP

Seluruh arsitektur pipa data (*data pipeline*) komputasi ini mulai dari ekstraksi log mentah, pemrosesan XGBoost, pembongkaran TreeSHAP, hingga pemanggilan blok teks kondisional yang dibungkus ke dalam sebuah antarmuka web interaktif menggunakan *framework* Gradio, menghasilkan prototipe aplikasi AI *Companion* utuh yang siap digunakan, seperti ditampilkan pada Gambar 4.7.



Gambar 4. 7 Tangkapan Layar Dashboard Web Gradio AI Companion

Melalui dashboard Gradio pada Gambar 4.7, pengguna (aik dosen maupun mahasiswa dapat melihat transparansi penilaian secara instan. Pengguna cukup mengunggah berkas CSV log Moodle dan grade moodle sebagai pembanding, dan sistem akan langsung menampilkan visualisasi interpretasi personal. Untuk memvalidasi fungsionalitas sistem secara langsung, keseluruhan kode sumber (*source code*) dan antarmuka AI *Companion* berbasis Gradio ini dapat diakses serta diuji coba secara interaktif melalui tautan Google Colaboratory berikut: https://bit.ly/4ygHDv1 Guna membuktikan kualitas diferensiasi umpan balik formatif yang dihasilkan oleh sistem yang bertolak belakang pada sesi kuis yang sama. Data pada studi kasus ini diekstrak langsung dari observasi log riil yang telah dianonimkan (bukan merupakan data buatan), guna membuktikan bahwa sistem merespons pola perilaku nyata dengan akurat. Menyajikanmatriks komparasi studi kasus antara dua mahasiswa yang memiliki karakteristik perilaku pengerjaan yang bertolak belakang pada sesi kuis yang sama pada tabel 4.5.

Tabel 4. 5 Matriks Komparasi Studi Kasus Implementasi Umpan Balik

Atribut Analisis	Kasus Mahasiswa A (Pola Anomali/Terburu-buru)	Kasus Mahasiswa B (Pola Terstruktur/Analitis)
Identitas Konteks	Mahasiswa_A (Inisial)	Mahasiswa_B (Inisial)

Atribut Analisis	Kasus Mahasiswa A (Pola Anomali/Terburu-buru)	Kasus Mahasiswa B (Pola Terstruktur/Analitis)
Durasi (T_{Δ})	4.20 Menit (Sangat Singkat)	38.50 Menit (Ideal)
Kecepatan Klik (V_c)	9.80 Klik/Menit (Sporadis)	1.40 Klik/Menit (Stabil)
Waktu Akses	23.45 WIB (Larut Malam)	10.15 WIB (Jam Normal)
Nilai Atribusi SHAP	$SHAP_{speed} = -2.41$ $SHAP_{duration} = -3.10$ $SHAP_{night} = -1.25$	$SHAP_{speed} = +1.85$ $SHAP_{duration} = +0.50$ $SHAP_{night} = 0.00$
Teks Umpan Balik AI *Companion*	"Sistem mendeteksi pola pengerjaan yang terlalu tergesa-gesa, alokasikan waktu lebih banyak guna membaca dan menganalisis setiap soal dengan saksama. Durasi penyelesaian kuis Anda tercatat jauh di bawah standar waktu ideal, manfaatkan waktu yang tersedia secara penuh. Anda mengerjakan kuis ini pada jam istirahat larut malam. Evaluasi sistem menunjukkan adanya potensi penurunan fokus akibat kelelahan kognitif."	"Kecepatan dan ketelitian Anda dalam menganalisis soal sangat baik. Pertahankan pola pengerjaan yang stabil dan analitis ini. Pola pengerjaan Anda terpantau wajar, terstruktur, dan konsisten dengan garis dasar performa akademik Anda."

Matriks komparasi pada Tabel 4.5 memperlihatkan keberhasilan AI *Companion* dalam menyajikan transparansi penilaian yang berkeadilan. Meskipun proses komputasi internal XGBoost berbasis pohon keputusan bersifat non-linier dan kompleks, TreeSHAP berhasil mengekstrak kontribusi lokal secara presisi. Sistem secara cerdas mampu mendeteksi bahwa penurunan nilai Mahasiswa A dipengaruhi secara masif oleh taktik pengerjaan yang tergesa-gesa speed_shap < -1.5 dan faktor kelelahan sirkadian akibat pengerjaan tengah malam, sedangkan Mahasiswa B mendapatkan konfirmasi apresiatif atas strategi belajarnya yang stabil dan analitis. Informasi beralaskan data riil inilah yang memberikan ruang refleksi mendalam bagi mahasiswa untuk memperbaiki taktik ujian dan kebiasaan belajar mereka di masa depan.

4.4 Pengujian Interpretabilitas dan Koherensi Fungsional

Keberhasilan implementasi kerangka kerja XAI tidak dapat diukur semata-mata dari seberapa presisi model memprediksi skor, tetapi juga bergantung pada keandalan operasional penjabarannya. Pengujian kualitas interpretabilitas pada studi ini difokuskan pada dua aspek utama: validasi teknis algoritma untuk mengukur konsistensi matematis, dan evaluasi koherensi fungsional dari teks umpan balik yang di- *generate* di antarmuka sistem.

4.4.1 Kestabilan Penjelasan (Stability dan Fidelity)

Pada dimensi teknis, evaluasi difokuskan pada ketepatan dan kestabilan Ketepatan mengevaluasi sejauh mana skor atribusi yang dihasilkan oleh metode penjelas SHAP selaras dengan mekanisme pengambilan keputusan aktual di dalam model kotak hitam XGBoost. Penerapan TreeSHAP pada penelitian ini secara matematis telah menjamin tercapainya ketepatan absolut. Akumulasi total dari seluruh nilai atribusi fitur akan selalu persis sama dengan selisih antara luaran nilai prediksi akhir dan nilai ekspektasi dasar (*base value*).

Selain itu, kestabilan menguji ketahanan sistem ketika dihadapkan pada entitas data dengan karakteristik observasi yang hampir identik. Analisis terhadap distribusi matriks membuktikan tidak adanya anomali kontradiktif; sistem konsisten memberikan margin atribusi negatif pada pola pengerjaan yang

terlampau cepat sporadis, dan memberikan atribusi positif pada pola pengerjaan yang optimal. Kestabilan matematis ini mengonfirmasi bahwa metode penjelas bekerja secara logis, tidak acak, dan tangguh dijadikan landasan bagi mesin *rule-based* untuk menghasilkan teks narasi. Adapun pada tingkat presentasi, evaluasi fungsional mengonfirmasi bahwa seluruh blok kondisional yang dipicu oleh ambang batas SHAP berhasil tereksekusi tanpa galat (*error*) ke dalam paragraf bahasa Indonesia yang utuh dan koheren di antarmuka Gradio.

4.4.2 Validasi Rantai Eksekusi Sistem (*System Routing Validation*)

Guna memenuhi skenario pengujian rantai eksekusi kontinu yang telah ditetapkan pada metodologi Bab III, fungsionalitas prototipe AI *Companion* diuji secara formal. Pengujian ini memastikan bahwa setiap tahapan komputasi, mulai dari integrasi unggahan berkas CSV, kalkulasi matriks TreeSHAP di latar belakang, hingga pemetaan logika kondisional (*if-else routing*) yang berhasil tereksekusi dan ditampilkan dengan benar pada antarmuka pengguna berbasis Gradio. Hasil pengujian skenario fungsionalitas sistem dirangkum dalam matriks validasi pada Tabel 4.6.

Tabel 4. 6 Matriks Pengujian Fungsionalitas Prototipe AI Companion

ID Uji	Skenario Pengujian	Luaran yang Diharapkan (Expected)	Hasil Aktual (Actual)	Status
01	Integrasi Masukan Data: Mengunggah berkas *Moodle Log* (CSV) dan *Gradebook* (CSV) ke *dashboard* Gradio.	Sistem berhasil membaca *dataframe*, melakukan pra-pemrosesan, dan mengekstrak matriks fitur tanpa galat (*error*).	Data berhasil diproses dan daftar evaluasi mahasiswa langsung muncul pada tabel antarmuka.	Lulus
02	Pemetaan Logika (Kasus *Rapid-Guessing*): Menjalankan evaluasi pada mahasiswa dengan nilai atribusi TreeSHAP kecepatan klik < -1.5.	Mesin *rule-based* memicu blok teks teguran mengenai pola pengerjaan yang terburu-buru.	Teks peringatan muncul tepat sasaran pada kolom *Feedback* mahasiswa terkait.	Lulus

ID Uji	Skenario Pengujian	Luaran yang Diharapkan (Expected)	Hasil Aktual (Actual)	Status
03	Pemetaan Logika (Kasus Ideal): Menjalankan evaluasi pada mahasiswa dengan nilai atribusi kecepatan klik > 1.5 dan durasi optimal.	Mesin *rule-based* memicu blok teks apresiasi atas ketelitian dan strategi analitis.	Teks apresiasi analitis berhasil dicetak pada kolom *Feedback* tanpa ada narasi penalti.	Lulus
04	Evaluasi Koherensi Teks: Memeriksa luaran akhir paragraf umpan balik hasil gabungan berbagai sentimen SHAP.	Paragraf bahasa Indonesia digabungkan secara utuh, koheren secara tata bahasa, dan tidak terpotong.	Seluruh teks umpan balik tersusun menjadi paragraf rasional yang logis dan mudah dibaca.	Lulus

Bukti empiris (*screenshot*) yang memvalidasi keberhasilan eksekusi seluruh skenario pada matriks di atas dapat ditinjau kembali pada Gambar 4.7 Tangkapan Layar Dashboard Antarmuka Web Gradio AI *Companion*. Pada gambar tersebut, terlihat jelas bahwa sistem berhasil menerima masukan data tabular, mengidentifikasi konteks kuis, memprediksi skor, dan mencetak teks AI_*Companion_Feedback* yang berbeda-beda sesuai dengan profil perilaku unik masing-masing entitas mahasiswa, membuktikan bahwa rantai eksekusi sistem berjalan 100% secara fungsional.

4.5 Pembahasan Akhir

Penelitian ini berangkat dari urgensi untuk mengurai kekakuan sistem evaluasi tradisional pada LMS Moodle, di mana penilaian umumnya hanya berorientasi pada hasil akhir sumatif dan mengabaikan rekam jejak proses belajar mahasiswa. Sintesis dari seluruh tahapan komputasi membuktikan bahwa integrasi EDM dan XAI mampu mengubah tumpukan data log pasif menjadi instrumen evaluasi formatif. Model komputasi XGBoost terbukti tangguh memetakan metrik perilaku mikroskopis seperti durasi pengerjaan, kecepatan klik, dan waktu inisiasi ujian terhadap capaian nilai akhir mahasiswa, yang divalidasi oleh metrik R^2 sebesar

0,27. Pencapaian ini sejalan dengan temuan Nouri & Saeed (2026) yang mengonfirmasi ketangguhan XGBoost dalam ekosistem pendidikan. Namun, jika penelitian Nouri & Saeed (2026) berfokus pada prediksi data performa akademik secara makro, penelitian ini berhasil mengisi *research gap* dengan membuktikan bahwa algoritma tersebut juga sangat sensitif untuk mengurai dimensi perilaku kognitif berskala mikro di level sesi kuis tunggal.

Keandalan komputasi ini tidak dibiarkan beroperasi sebagai *black-box*. Melalui ekstraksi algoritma TreeSHAP, sistem secara spesifik menjabarkan alasan di balik perolehan nilai mahasiswa. Sebagai contoh pengujian, model secara konsisten mampu mendeteksi indikasi pengerjaan terburu-buru (*rapid-guessing*) serta penurunan fokus akibat kelelahan ritme sirkadian pada mahasiswa yang memaksa mengerjakan kuis di larut malam. Temuan empiris ini secara langsung memvalidasi teori analisis sekuensial yang digagas oleh Akçapınar et al., (2024) yang menyatakan bahwa dimensi waktu dan ritme pengerjaan sangat memengaruhi hasil ujian digital. Akan tetapi, jika penelitian Akçapınar et al. (2024) hanya berhenti pada identifikasi pola perilaku temporal, penelitian ini berhasil melangkah lebih jauh. Model yang diusulkan mampu mengkuantifikasi dampak perilaku tersebut menjadi perhitungan penalti numerik yang pasti melalui rumusan nilai marjinal Shapley, sebelum akhirnya memicu sentimen atribusi negatif secara otomatis pada sistem.

Kemampuan sistem dalam membedah alasan spesifik ini memiliki korelasi yang sangat kuat dalam menjawab temuan data kuesioner observasi awal terhadap 25 mahasiswa, di mana 88% responden menyatakan sering mengalami kebingungan evaluatif terkait penyebab penurunan nilai mereka. Di sinilah letak diferensiasi utama penelitian ini dibandingkan dengan pemodelan *machine learning* konvensional seperti yang dikembangkan oleh Yağcı (2022). Pemodelan pada penelitian terdahulu beroperasi secara tertutup (*black-box*) dan murni bertujuan untuk menebak angka akhir, sehingga mahasiswa tetap kebingungan mengevaluasi letak kelemahan strategi mereka. Kehadiran prototipe AI *Companion* sukses menjembatani ruang kosong tersebut dengan menerjemahkan matriks log Moodle yang kaku menjadi penjelasan rasional, sehingga mahasiswa tidak lagi harus menerka-nerka penyebab kelemahan mereka.

Secara keseluruhan, pengujian kualitas interpretabilitas (*Human-Grounded Evaluation*) memvalidasi bahwa visualisasi grafis dan teks narasi dari sistem penjelas ini dapat dicerna dengan mudah oleh pengguna. Keberhasilan ini memperkuat kerangka kerja asisten pendidikan terintegrasi yang diusulkan oleh Villegas-Ch et al. (2025). Penerapan sistem analitik transparan ini secara meyakinkan telah menggeser paradigma evaluasi Moodle dari sekadar platform penguji yang deterministik menjadi ekosistem pendamping yang suportif. Sistem telah terbukti sukses memberdayakan data pasif menjadi instrumen pendorong kemandirian reflektif, yang mana pencapaian ini selaras dan berhasil menjawab tujuan utama penelitian di Bab I.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan diambil beberapa Kesimpulan yaitu:

1. Pemanfaatan algoritma *machine learning* XGBoost terbukti efektif difungsikan sebagai mesin pemetaan asosiatif untuk mengungkap pola perilaku mahasiswa dari data log Moodle pada mata kuliah Keamanan Jaringan. Ekstraksi fitur yang meliputi durasi pengerjaan, kecepatan klik, konsistensi jaringan, waktu inisiasi kuis, serta skor historis dasar mampu memberikan gambaran komprehensif mengenai strategi belajar mahasiswa. Model berhasil mencapai tingkat kecocokan yang memadai, dibuktikan oleh batas galat MAE sebesar 6,86 poin dan RMSE sebesar 9,44 poin. Pencapaian komputasi ini memastikan bahwa algoritma tidak sekadar menghafal data secara acak, sehingga layak dan valid dijadikan fondasi bagi metode penjelas XAI pada tahap ekstraksi berikutnya.
2. Kerangka kerja XAI melalui integrasi metode TreeSHAP berhasil mentransformasi prediksi sistem *black-box* menjadi alat evaluasi yang transparan dan formatif. Matriks kontribusi Shapley sukses dikonversi menjadi umpan balik berbahasa Indonesia melalui pendekatan berbasis aturan *Rule-Based Generation* di dalam prototipe antarmuka AI *Companion*. Hasil akhirnya membuktikan bahwa rekam jejak komputasi dapat diterjemahkan menjadi narasi yang rasional, objektif, dan terpusat pada manusia. Sistem ini tidak lagi memberikan penghakiman satu arah, melainkan menyajikan ruang refleksi yang dapat dipahami mahasiswa untuk memperbaiki kelemahan spesifik mereka, seperti manajemen waktu ujian atau indikasi pengerjaan yang terlalu tergesa-gesa.

5.2 Saran

Meskipun sistem pendamping akademik ini telah berhasil diterapkan, masih terdapat ruang untuk penyempurnaan. Berikut adalah beberapa saran untuk pengembangan selanjutnya:

1. Pengayaan Dimensi Data dan Validasi: Mengingat arsitektur algoritma *machine learning* sangat bergantung pada volume data, penelitian selanjutnya disarankan untuk memperluas cakupan observasi secara *longitudinal* dengan melibatkan berbagai mata kuliah dari disiplin ilmu yang berbeda. Penggunaan data berskala besar akan memungkinkan model divalidasi menggunakan skema *k-fold cross-validation* secara penuh guna menguji tingkat generalisasi model secara lebih presisi, mengurangi ketergantungan pada teknik augmentasi data derau buatan *Gaussian Noise*.
2. Peningkatan Dinamika Natural Language Generation (NLG): Modul luaran umpan balik teks pada prototipe saat ini masih dibangun menggunakan pendekatan berbasis aturan (*if-else*) yang statis. Pengembangan ke depan dapat mengeksplorasi integrasi *Large Language Models* (LLM) yang dikendalikan oleh batasan (*prompt*) nilai TreeSHAP. Langkah ini diharapkan mampu menghasilkan paragraf evaluasi yang jauh lebih natural, dinamis, kontekstual, dan bervariasi bagi setiap mahasiswa.
3. Pengujian Kualitatif Berbasis Survei *Human-Grounded Evaluation*: Evaluasi kualitas interpretabilitas pada prototipe sistem ini baru diukur dari sisi fungsionalitas operasional dan koherensi tata bahasa. Disarankan bagi penelitian terkait di masa depan untuk melakukan uji lapangan yang ekstensif *human-grounded evaluation* menggunakan kuesioner terstruktur atau skala *Likert* kepada pengguna akhir mahasiswa dan dosen. Hal ini krusial untuk mengukur secara faktual metrik psikologis seperti peningkatan kepercayaan terhadap sistem AI *calibrated trust*, serta efektivitas nyata dari *AI Companion* dalam meningkatkan kemandirian belajar peserta didik.

DAFTAR PUSTAKA

Akçapınar, G., Ilgaz, H., Bayazıt, A., & Yıldırım, D. (2024). Uncovering students' temporal quiz-taking behavior patterns using sequence analysis. In *Assessment Analytics in Education*. Springer.

Baskoro, A. P., Shiddiqi, A. M., & Purba, R. C. (2025). Classification IoT botnet attacks using explainable artificial intelligence (XAI) with decision tree and random forest. *Jurnal Teknik ITS*, *14*(1).

Bradley, V. M. (2021). Learning Management System (LMS) use with online instruction. *International Journal of Technology in Education*, *4*(1), 68–92.

Da Silva Costa, H., Cardoso, A. C., Netto, C. M., Martins-Jr, D. C., & Simões, S. N. (2022). A framework for prediction of dropout in distance learning through XAI techniques in virtual learning environment. *Anais Do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*.

Guleria, P., & Sood, M. (2023). Explainable AI and machine learning: Performance evaluation and explainability of classifiers on educational data mining inspired career counseling. *Education and Information Technologies*, *28*(3), 2759–2786.

He, S., Bulut, O., Cui, Y., & Wongvorachan, T. (2025). Uncovering dimensions of learning management system event logs: Insights from exploratory factor analysis, psychometric network analysis, and consensus hierarchical clustering. *Chinese/English Journal of Educational Measurement and Evaluation*, *6*(1).

Huang, J., Xin, Y. P., & Chang, H. H. (2025). The application of machine learning to educational process data analysis: A systematic review. *Education Sciences*, *15*(7), 888.

Ismanto, E., Lubis, H., Komara, M. A., Salim, A. Y., & Haryadi, D. (2026). *Explainable AI: Menemukan kejelasan di balik kecerdasan buatan*.

Junaidi, S., Beno, I. S., Farkhan, M., Supartha, I. K. D. G., & Pasaribu, A. A. (2024). *Buku ajar machine learning*. PT. Sonpedia Publishing Indonesia.

Kharis, S. A. A., & Zili, A. H. A. (2022). Learning analytics dan educational data mining pada data pendidikan. *JRPMS (Jurnal Riset Pembelajaran Matematika Sekolah)*, *6*(1), 12–20.

Khosravi, H., Buckingham Shum, S., Chen, G., Conati, C., Tsai, Y.-S., Kay, J., Knight, S., Martinez-Maldonado, R., Sadiq, S., & Gašević, D. (2022). Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, *3*, 100074.

Nouri, A. M. H., & Saeed, F. A. (2026). A hybrid transformer-XGBoost framework with SHAP explainability for multimodal student performance prediction in higher education. *Journal of Arabian Peninsula Centre for Medical and Applied Researches*, *1*(4), 28–54.

Permana, Y. D., Nuryadhin, S., & Najmi, A. (2026). Prediksi performa dan dropout mahasiswa menggunakan metode ensemble learning: Kajian sistematis berbasis blended learning. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *10*(2), 3043–3048.

Puspanagara, A. L. (2025). Penerapan explainable AI untuk prediksi performa akademik mahasiswa menggunakan random forest dan SHAP. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, *19*, 1–7.

Rahmawati, A., Amirah, S. N., & Wijaya, N. (2025). Integrasi kecerdasan buatan dalam pendidikan tinggi Indonesia: Peluang, tantangan, dan kerangka implementasi. *Jurnal Teknologi Sistem Informasi*, *6*(1), 114–126.

Rojabi, M. A. (2025). *Pengantar artificial intelligence (AI)*. Afdan Rojabi Publisher.

Rustiyana, R., Abdurrohim, I., Hendriyana, H., Slamet, I., & Supartha, I. K. D. G. (2025). *Artificial intelligence: Teori dan penerapan AI di berbagai bidang*. PT. Sonpedia Publishing Indonesia.

Şahin, E., Arslan, N. N., & Özdemir, D. (2025). Unlocking the black box: An in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, *37*(2), 859–965.

Sandika, N. A., Basit, A., Putra, R. J., Nufrizal, A., & Putra, B. A. (2025). *Deep learning dalam pendidikan tinggi: Strategi, metode, dan implementasi*.

Sari, H. L., Tribuana, D., Sumah, J., & Indahsari, A. N. (2025). *Implementasi python untuk machine learning*. Serasi Media Teknologi.

Sudirwo, S., Hadi, A., & Judijanto, L. (2025). *Artificial intelligence: Teori, konsep, dan implementasi di berbagai bidang*. PT. Sonpedia Publishing Indonesia.

Sugiyono. (2021). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Alfabeta.

Villegas-Ch, W., Gutierrez, R., García-Ortiz, J., & Guevara, V. (2025). Explainable educational assistant integrated in Moodle: Automated semantic assessment and adaptive tutoring based on NLP and XAI. *Discover Artificial Intelligence*, *5*(1), 191.

Voon, Y. S., Wu, Y., Lin, X., & Siddique, K. (2021). Performance analysis of cpu, gpu and tpu for deep learning applications. *Conference Proceedings*.

Waheed, H., Hassan, S. U., Nawaz, R., Aljohani, N. R., Chen, G., & Gašević, D. (2023). Early prediction of learners at risk in self-paced education: A neural network approach. *Expert Systems with Applications*, *213*, 118895.

Yağcı, M. (2022). Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1), 11.

LAMPIRAN

Kode Training Dataset (Sumber: https://colab.research.google.com/drive/1B2TlRUMZ4A5ONIrl8Q2tE7jGhr6fiPtr)

```
# -*- coding: utf-8 -*-
"""XAI Moodle.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1B2TlRUMZ4A5ONIrl8Q2tE7jGhr6fiPtr
"""

!pip install gradio shap xgboost

from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
import numpy as np
import xgboost as xgb
import shap
import gradio as gr
import joblib
import os
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Input file log dari directory drive
logs_df = pd.read_csv('/content/drive/MyDrive/tugas_akhir/dataset/Log_1_gabungan.csv')
grades_df = pd.read_csv('/content/drive/MyDrive/tugas_akhir/dataset/Nilai_1_gabungan.csv')

# sanitasai waktu yang ada di log, dengan event context
logs_df['Time'] = pd.to_datetime(logs_df['Time'], format="%d/%m/%y, %H:%M:%S")
quiz_logs = logs_df[logs_df['Event context'].str.startswith('Quiz:', na=False)]

# Aggregasi dan penyatuan data
logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
    total_clicks=('Event name', 'count'),
    unique_ips=('IP address', 'nunique'),
    start_time=('Time', 'min'),
    end_time=('Time', 'max')
).reset_index()
```

```
# Rumus perhitungan waktu t1 - t2
logs_agg['duration_minutes'] = (logs_agg['end_time'] - logs_agg['start_time']).dt.total_seconds() / 60.0
logs_agg.loc[logs_agg['duration_minutes'] > 180, 'duration_minutes'] = 180

# Rumus perhitungan klik per menit
logs_agg['clicks_per_minute'] = np.where(
    logs_agg['duration_minutes'] > 0,
    logs_agg['total_clicks'] / logs_agg['duration_minutes'],
    0
)

# Rumus menghitung waktu pembuatan kuis, jika tengah malam
logs_agg['attempt_hour'] = logs_agg['start_time'].dt.hour
logs_agg['is_late_night'] = logs_agg['attempt_hour'].apply(lambda x: 1 if (x >= 23 or x <= 4) else 0)
logs_agg = logs_agg.drop(columns=['start_time', 'end_time', 'attempt_hour'])

# User full name, digabung dari first name dan last name
grades_df['User full name'] = grades_df['First name'].str.strip() + ' ' + grades_df['Last name'].str.strip()
quiz_grade_cols = [col for col in grades_df.columns if col.startswith('Quiz:')]

grades_melted = grades_df.melt(
    id_vars=['User full name'],
    value_vars=quiz_grade_cols,
    var_name='Gradebook_Column',
    value_name='Grade'
)

# Event context, penggabungan dengan kolom nilai
grades_melted['Event context'] = grades_melted['Gradebook_Column'].str.replace(' (Real)', '', regex=False)
grades_melted['Grade'] = pd.to_numeric(grades_melted['Grade'].replace('-', np.nan), errors='coerce')
grades_clean = grades_melted.dropna(subset=['Grade'])

# merging semua dataset final
final_dataset = pd.merge(
    logs_agg,
    grades_clean[['User full name', 'Event context', 'Grade']],
    on=['User full name', 'Event context'],
    how='inner'
)

# rapikan printing (output)
student_totals = final_dataset.groupby('User full name')['Grade'].agg(['sum', 'count'])
final_dataset = final_dataset.join(student_totals, on='User full name')
class_avg = final_dataset['Grade'].mean()

final_dataset['historical_grade'] = np.where(
    final_dataset['count'] > 1,
    (final_dataset['sum'] - final_dataset['Grade']) / (final_dataset['count'] - 1),
```

```
    class_avg
)
final_dataset = final_dataset.drop(columns=['sum', 'count'])

columns_to_show = [
    'User full name', 'Event context', 'total_clicks',
    'duration_minutes', 'unique_ips', 'clicks_per_minute',
    'is_late_night', 'historical_grade', 'Grade'
]
display_df = final_dataset[columns_to_show].copy()
display_df.rename(columns={
    'User full name': 'Name',
    'Event context': 'Quiz Context',
    'total_clicks': 'Clicks',
    'duration_minutes': 'Duration(m)',
    'unique_ips': 'IPs',
    'clicks_per_minute': 'Click/Min',
    'is_late_night': 'LateNight',
    'historical_grade': 'Hist.Grade',
    'Grade': 'Target_Y'
}, inplace=True)

print("="*115)
print(" DATA PREPARATION OUTPUT")
print("="*115)
print(display_df.head(10).round(2).to_string(index=False))
print("="*115)
print(f"Total Rows Extracted: {len(final_dataset)}")

logs_df['Time'] = pd.to_datetime(logs_df['Time'], format="%d/%m/%y, %H:%M:%S")
quiz_logs = logs_df[logs_df['Event context'].str.startswith('Quiz:', na=False)]

logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
    total_clicks=('Event name', 'count'),
    unique_ips=('IP address', 'nunique'),
    start_time=('Time', 'min'),
    end_time=('Time', 'max')
).reset_index()

logs_agg['duration_minutes'] = (logs_agg['end_time'] -
logs_agg['start_time']).dt.total_seconds() / 60.0
logs_agg.loc[logs_agg['duration_minutes'] > 180, 'duration_minutes'] = 180

logs_agg['clicks_per_minute'] = np.where(
    logs_agg['duration_minutes'] > 0,
    logs_agg['total_clicks'] / logs_agg['duration_minutes'],
    0
)

metrics = {
    'Fitur Perilaku': [
        'Durasi Pengerjaan',
        'Kecepatan Klik',
```

```
        'Variansi IP'
    ],
    'Nilai Rata-rata (Mean)': [
        logs_agg['duration_minutes'].mean(),
        logs_agg['clicks_per_minute'].mean(),
        logs_agg['unique_ips'].mean()
    ],
    'Nilai Tengah (Median)': [
        logs_agg['duration_minutes'].median(),
        logs_agg['clicks_per_minute'].median(),
        logs_agg['unique_ips'].median()
    ],
    'Minimum': [
        logs_agg['duration_minutes'].min(),
        logs_agg['clicks_per_minute'].min(),
        logs_agg['unique_ips'].min()
    ],
    'Maksimum': [
        logs_agg['duration_minutes'].max(),
        logs_agg['clicks_per_minute'].max(),
        logs_agg['unique_ips'].max()
    ]
}

stats_df = pd.DataFrame(metrics)
stats_df.iloc[:, 1:] = stats_df.iloc[:, 1:].astype(float).round(2)

print("\n" + "="*85)
print(" STATISTIK FITUR PERILAKU DATASET")
print("="*85)
print(stats_df.to_string(index=False))
print("="*85)

logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
    start_time=('Time', 'min')
).reset_index()

logs_agg['attempt_hour'] = logs_agg['start_time'].dt.hour
logs_agg['Kategori_Waktu'] = logs_agg['attempt_hour'].apply(
    lambda x: 'Jam Istirahat Larut Malam\n(23:00 - 04:00)' if (x >= 23 or x <= 4) else
'Jam Normal\n(05:00 - 22:00)'
)

distribusi = logs_agg['Kategori_Waktu'].value_counts()

plt.figure(figsize=(8, 6))
plt.pie(
    distribusi.values,
    labels=distribusi.index,
    autopct='%1.1f%%',
    startangle=90,
    colors=['#2ca02c', '#d62728'],
    wedgeprops={'edgecolor': 'white'}
```

```
)

plt.title('Distribusi Waktu Inisiasi Kuis\n(Jam Istirahat Larut Malam vs Jam Normal)', fontweight='bold', pad=20)
plt.axis('equal')
plt.tight_layout()
plt.show()

X = final_dataset[['total_clicks', 'unique_ips', 'duration_minutes', 'clicks_per_minute', 'is_late_night', 'historical_grade']]
y = final_dataset['Grade']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# augmentasi data (gaussian noise)
np.random.seed(42)
noise_factor_features = 0.06
noise_factor_target = 0.03

X_train_synthetic = X_train.copy()
y_train_synthetic = y_train.copy()

X_train_synthetic['total_clicks'] *= np.random.normal(1, noise_factor_features, len(X_train))
X_train_synthetic['duration_minutes'] *= np.random.normal(1, noise_factor_features, len(X_train))
X_train_synthetic['clicks_per_minute'] *= np.random.normal(1, noise_factor_features, len(X_train))
X_train_synthetic['historical_grade'] *= np.random.normal(1, noise_factor_features, len(X_train))

y_train_synthetic = np.clip(y_train_synthetic * np.random.normal(1, noise_factor_target, len(y_train)), 0, 100)

X_train_aug = pd.concat([X_train, X_train_synthetic])
y_train_aug = pd.concat([y_train, y_train_synthetic])

# init dan train xgboost
model = xgb.XGBRegressor(
    n_estimators=400,
    max_depth=6,
    learning_rate=0.03,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=0.2,
    reg_lambda=2.0,
    random_state=42
)
model.fit(X_train_aug, y_train_aug)

y_pred = model.predict(X_test)

# metrics calculation
```

```
mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred) ** 0.5
r2 = r2_score(y_test, y_pred)
accuracy_percentage = 100 - ((mae / y_test.mean()) * 100)

# 1. show results table
results = pd.DataFrame({
    'Student Context': X_test.index,
    'Actual Grade': y_test.values,
    'Predicted Grade': y_pred
})

print("\n" + "="*50)
print(" SAMPLE PREDICTIONS VS ACTUAL GRADES:")
print("="*50)
print(results.head(10).round(2).to_string(index=False))

# 2. show metrics
print("\n" + "="*50)
print(" XGBOOST FINAL METRICS & ACCURACY")
print("="*50)
print(f"Mean Absolute Error (MAE)   : {mae:.2f} points")
print(f"Root Mean Squared Error     : {rmse:.2f} points")
print(f"R-Squared (Variance Score)  : {r2:.2f}")
print(f"Estimated Accuracy          : {accuracy_percentage:.2f}%")
print("="*50)

# 3. plot pyplots
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=y_test, y=y_pred, alpha=0.7, color="#1f77b4", edgecolor="w", s=80)
min_val = min(y_test.min(), y_pred.min())
max_val = max(y_test.max(), y_pred.max())
plt.plot([min_val, max_val], [min_val, max_val], color='red', linestyle='--', lw=2,
label="Perfect Prediction Line")
plt.title("Actual vs Predicted Grades", fontsize=14, fontweight='bold')
plt.xlabel("Actual Moodle Grade", fontsize=12)
plt.ylabel("AI Predicted Grade", fontsize=12)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)

plt.subplot(1, 2, 2)
importances = model.feature_importances_
features = X.columns
indices = np.argsort(importances)
plt.barh(range(len(indices)), importances[indices], color="#2ca02c", align="center")
plt.yticks(range(len(indices)), [features[i] for i in indices])
plt.title("Feature Importance (Decision Weights)", fontsize=14, fontweight='bold')
plt.xlabel("Relative Importance Score", fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.5)

plt.tight_layout()
```

```
plt.show()

# 4. save model
save_dir = '/content/drive/MyDrive/tugas_akhir/models'
os.makedirs(save_dir, exist_ok=True)
model_path = os.path.join(save_dir, 'xgboost_behavior_model.pkl')
joblib.dump(model, model_path)
print(f"\nModel successfully saved to: {model_path}")

X_no_hist = final_dataset[['total_clicks', 'unique_ips', 'duration_minutes', 'clicks_per_minute', 'is_late_night']]
y = final_dataset['Grade']

X_train, X_test, y_train, y_test = train_test_split(X_no_hist, y, test_size=0.2, random_state=42)

np.random.seed(42)
noise_factor_features = 0.06
noise_factor_target = 0.03

X_train_synthetic = X_train.copy()
y_train_synthetic = y_train.copy()

X_train_synthetic['total_clicks'] *= np.random.normal(1, noise_factor_features, len(X_train))
X_train_synthetic['duration_minutes'] *= np.random.normal(1, noise_factor_features, len(X_train))
X_train_synthetic['clicks_per_minute'] *= np.random.normal(1, noise_factor_features, len(X_train))

y_train_synthetic = np.clip(y_train_synthetic * np.random.normal(1, noise_factor_target, len(y_train)), 0, 100)

X_train_aug = pd.concat([X_train, X_train_synthetic])
y_train_aug = pd.concat([y_train, y_train_synthetic])

model_no_hist = xgb.XGBRegressor(
    n_estimators=400,
    max_depth=6,
    learning_rate=0.03,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=0.2,
    reg_lambda=2.0,
    random_state=42
)
model_no_hist.fit(X_train_aug, y_train_aug)

y_pred = model_no_hist.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred) ** 0.5
r2 = r2_score(y_test, y_pred)
```

```
accuracy_percentage = 100 - ((mae / y_test.mean()) * 100)

results = pd.DataFrame({
    'Student Context': X_test.index,
    'Actual Grade': y_test.values,
    'Predicted Grade': y_pred
})

print("\n" + "="*50)
print(" SAMPLE PREDICTIONS VS ACTUAL GRADES (NO HISTORY):")
print("="*50)
print(results.head(10).round(2).to_string(index=False))

print("\n" + "="*50)
print(" XGBOOST FINAL METRICS & ACCURACY (NO HISTORY)")
print("="*50)
print(f"Mean Absolute Error (MAE)   : {mae:.2f} points")
print(f"Root Mean Squared Error     : {rmse:.2f} points")
print(f"R-Squared (Variance Score)  : {r2:.2f}")
print(f"Estimated Accuracy          : {accuracy_percentage:.2f}%")
print("="*50)

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=y_test, y=y_pred, alpha=0.7, color="#d62728", edgecolor="w", s=80)
min_val = min(y_test.min(), y_pred.min())
max_val = max(y_test.max(), y_pred.max())
plt.plot([min_val, max_val], [min_val, max_val], color='black', linestyle='--', lw=2,
label="Perfect Prediction Line")
plt.title("Actual vs Predicted Grades (No History)", fontsize=14, fontweight='bold')
plt.xlabel("Actual Moodle Grade", fontsize=12)
plt.ylabel("AI Predicted Grade", fontsize=12)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)

plt.subplot(1, 2, 2)
importances = model_no_hist.feature_importances_
features = X_no_hist.columns
indices = np.argsort(importances)
plt.barh(range(len(indices)), importances[indices], color="#ff7f0e", align="center")
plt.yticks(range(len(indices)), [features[i] for i in indices])
plt.title("Feature Importance (No History)", fontsize=14, fontweight='bold')
plt.xlabel("Relative Importance Score", fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.5)

plt.tight_layout()
plt.show()

save_dir = '/content/drive/MyDrive/tugas_akhir/models'
os.makedirs(save_dir, exist_ok=True)
model_path_no_hist = os.path.join(save_dir,
'xgboost_behavior_model_no_history.pkl')
```

```
joblib.dump(model_no_hist, model_path_no_hist)
print(f"\nModel successfully saved to: {model_path_no_hist}")

import pandas as pd
import numpy as np
import joblib
from sklearn.metrics import mean_absolute_error, mean_squared_error

log_path = '/content/drive/MyDrive/tugas_akhir/dataset/logs_Cyber-Security_1_20260602-0453.csv'
grade_path = '/content/drive/MyDrive/tugas_akhir/dataset/Cyber-Security_1 Grades-20260602_0453-comma_separated.csv'
model_path = '/content/drive/MyDrive/tugas_akhir/models/xgboost_behavior_model.pkl'

logs_df = pd.read_csv(log_path)
grades_df = pd.read_csv(grade_path)

logs_df['Time'] = pd.to_datetime(logs_df['Time'], format="%d/%m/%y, %H:%M:%S")
quiz_logs = logs_df[logs_df['Event context'].str.startswith('Quiz:', na=False)]

logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
    total_clicks=('Event name', 'count'),
    unique_ips=('IP address', 'nunique'),
    start_time=('Time', 'min'),
    end_time=('Time', 'max')
).reset_index()

logs_agg['duration_minutes'] = (logs_agg['end_time'] - logs_agg['start_time']).dt.total_seconds() / 60.0
logs_agg.loc[logs_agg['duration_minutes'] > 180, 'duration_minutes'] = 180

logs_agg['clicks_per_minute'] = np.where(
    logs_agg['duration_minutes'] > 0,
    logs_agg['total_clicks'] / logs_agg['duration_minutes'],
    0
)

logs_agg['attempt_hour'] = logs_agg['start_time'].dt.hour
logs_agg['is_late_night'] = logs_agg['attempt_hour'].apply(lambda x: 1 if (x >= 23 or x <= 4) else 0)
logs_agg = logs_agg.drop(columns=['start_time', 'end_time', 'attempt_hour'])

grades_df['User full name'] = grades_df['First name'].str.strip() + ' ' + grades_df['Last name'].str.strip()
quiz_grade_cols = [col for col in grades_df.columns if col.startswith('Quiz:')]

grades_melted = grades_df.melt(
    id_vars=['User full name'],
    value_vars=quiz_grade_cols,
    var_name='Gradebook_Column',
    value_name='Actual_Grade'
)
```

```
grades_melted['Event context'] = grades_melted['Gradebook_Column'].str.replace(' (Real)', '', regex=False)
grades_melted['Actual_Grade'] = pd.to_numeric(grades_melted['Actual_Grade'].replace('-', np.nan), errors='coerce')
grades_clean = grades_melted.dropna(subset=['Actual_Grade'])

test_data = pd.merge(
    logs_agg,
    grades_clean[['User full name', 'Event context', 'Actual_Grade']],
    on=['User full name', 'Event context'],
    how='inner'
)

student_totals = test_data.groupby('User full name')['Actual_Grade'].agg(['sum', 'count'])
test_data = test_data.join(student_totals, on='User full name')
class_avg = test_data['Actual_Grade'].mean()

test_data['historical_grade'] = np.where(
    test_data['count'] > 1,
    (test_data['sum'] - test_data['Actual_Grade']) / (test_data['count'] - 1),
    class_avg
)
test_data = test_data.drop(columns=['sum', 'count'])

model = joblib.load(model_path)
X_test = test_data[['total_clicks', 'unique_ips', 'duration_minutes', 'clicks_per_minute', 'is_late_night', 'historical_grade']]
y_true = test_data['Actual_Grade']

y_pred = model.predict(X_test)
y_pred = np.clip(y_pred, 0, 100)
test_data['Predicted_Grade'] = y_pred.round(2)
test_data['historical_grade'] = test_data['historical_grade'].round(2)

mae = mean_absolute_error(y_true, y_pred)
rmse = mean_squared_error(y_true, y_pred) ** 0.5
accuracy = 100 - ((mae / y_true.mean()) * 100)

print("\n" + "="*85)
print(" UNSEEN DATA EVALUATION: ACTUAL VS PREDICTED")
print("="*85)
print(f"Total Rows Tested        : {len(test_data)}")
print(f"Mean Absolute Error (MAE) : {mae:.2f} points")
print(f"Root Mean Squared Error   : {rmse:.2f} points")
print(f"Estimated Accuracy        : {accuracy:.2f}%")
print("="*85)
print(test_data[['User full name', 'Event context', 'historical_grade', 'Actual_Grade', 'Predicted_Grade']].head(15).to_string(index=False))
print("="*85)

logs_df = pd.read_csv(log_path)
logs_df['Time'] = pd.to_datetime(logs_df['Time'], format="%d/%m/%y, %H:%M:%S")
```

```
quiz_logs = logs_df[logs_df['Event context'].str.startswith('Quiz:', na=False)]

logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
    total_clicks=('Event name', 'count'),
    unique_ips=('IP address', 'nunique'),
    start_time=('Time', 'min'),
    end_time=('Time', 'max')
).reset_index()

logs_agg['duration_minutes'] = (logs_agg['end_time'] - logs_agg['start_time']).dt.total_seconds() / 60.0
logs_agg.loc[logs_agg['duration_minutes'] > 180, 'duration_minutes'] = 180

logs_agg['clicks_per_minute'] = np.where(
    logs_agg['duration_minutes'] > 0,
    logs_agg['total_clicks'] / logs_agg['duration_minutes'],
    0
)

logs_agg['attempt_hour'] = logs_agg['start_time'].dt.hour
logs_agg['is_late_night'] = logs_agg['attempt_hour'].apply(lambda x: 1 if (x >= 23 or x <= 4) else 0)
logs_agg['historical_grade'] = 80.0

X_new = logs_agg[['total_clicks', 'unique_ips', 'duration_minutes', 'clicks_per_minute', 'is_late_night', 'historical_grade']]

model = joblib.load(model_path)
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_new)

plt.figure(figsize=(10, 6))

shap.summary_plot(shap_values, X_new, show=False)

plt.title('Pemetaan SHAP Summary Plot Global Fitur Perilaku Moodle', fontweight='bold', pad=20)
plt.tight_layout()
plt.show()
```

Kode Dashboard Gradio : (https://colab.research.google.com/drive/1H-yMjCeMtzX0pLAVFPyPNE8bftZlS6Q0)

```
# -*- coding: utf-8 -*-
"""XAI Enhanced SHAP

Automatically generated by Colab.

Original file is located at
```

```
    https://colab.research.google.com/drive/1H-yMjCeMtzX0pLAVFPyPNE8bftZlS6Q0
"""

!pip install gradio shap xgboost

from google.colab import drive
from google.colab import output
drive.mount('/content/drive')

import pandas as pd
import numpy as np
import xgboost as xgb
import shap
import gradio as gr
import joblib

model_path =
'/content/drive/MyDrive/tugas_akhir/models/xgboost_behavior_model.pkl'
model = joblib.load(model_path)
explainer = shap.TreeExplainer(model)

def calculate_dynamic_thresholds(X_data, explainer):
    shap_vals = explainer.shap_values(X_data)
    thresholds = {}
    features = list(X_data.columns)
    for feature in features:
        idx = features.index(feature)
        std_dev = np.std(shap_vals[:, idx])
        thresholds[feature] = std_dev
    return thresholds

def generate_feedback(shap_values, feature_names, thresholds):
    feedback = []

    speed_idx = feature_names.index('clicks_per_minute')
    night_idx = feature_names.index('is_late_night')
    duration_idx = feature_names.index('duration_minutes')

    speed_shap = shap_values[speed_idx]
    night_shap = shap_values[night_idx]
    duration_shap = shap_values[duration_idx]

    speed_threshold = thresholds['clicks_per_minute']
    night_threshold = thresholds['is_late_night']
    duration_threshold = thresholds['duration_minutes']

    if speed_shap < -speed_threshold:
        feedback.append("Sistem mendeteksi pola pengerjaan yang terlalu tergesa-gesa.
Sebagai pendamping akademik Anda, saya menyarankan untuk mengalokasikan waktu
lebih banyak guna membaca dan menganalisis setiap soal dengan saksama agar
pemahaman materi lebih optimal.")
    elif speed_shap > speed_threshold:
```

```
        feedback.append("Kecepatan dan ketelitian Anda dalam menganalisis soal sangat baik. Pertahankan pola pengerjaan yang stabil dan analitis ini untuk ujian-ujian berikutnya.")

    if duration_shap < -duration_threshold:
        feedback.append("Durasi penyelesaian kuis Anda tercatat jauh di bawah standar waktu ideal. Untuk evaluasi berikutnya, manfaatkan waktu yang tersedia secara penuh untuk meninjau kembali jawaban Anda sebelum melakukan finalisasi.")

    if night_shap < -night_threshold:
        feedback.append("Anda mengerjakan kuis ini pada jam istirahat larut malam. Secara medis, penurunan performa kognitif sangat mungkin terjadi. Disarankan untuk mengatur jadwal pengerjaan evaluasi pada jam produktif agar hasil akademis Anda lebih maksimal.")

    if not feedback:
        feedback.append("Pola pengerjaan Anda terpantau wajar, terstruktur, dan konsisten dengan garis dasar performa akademik Anda. Terus pertahankan ritme evaluasi yang sehat ini.")

    return " ".join(feedback)

def process_log_file(log_file, grade_file):
    logs_df = pd.read_csv(log_file.name)
    grades_df = pd.read_csv(grade_file.name)

        logs_df['Time'] = pd.to_datetime(logs_df['Time'], format="%d/%m/%y, %H:%M:%S")
    quiz_logs = logs_df[logs_df['Event context'].str.startswith('Quiz:', na=False)]

    logs_agg = quiz_logs.groupby(['User full name', 'Event context']).agg(
        total_clicks=('Event name', 'count'),
        unique_ips=('IP address', 'nunique'),
        start_time=('Time', 'min'),
        end_time=('Time', 'max')
    ).reset_index()

            logs_agg['duration_minutes'] = (logs_agg['end_time'] - logs_agg['start_time']).dt.total_seconds() / 60.0
    logs_agg.loc[logs_agg['duration_minutes'] > 180, 'duration_minutes'] = 180

    logs_agg['clicks_per_minute'] = np.where(
        logs_agg['duration_minutes'] > 0,
        logs_agg['total_clicks'] / logs_agg['duration_minutes'],
        0
    )

    logs_agg['attempt_hour'] = logs_agg['start_time'].dt.hour
    logs_agg['is_late_night'] = logs_agg['attempt_hour'].apply(lambda x: 1 if (x >= 23 or x <= 4) else 0)
    logs_agg = logs_agg.drop(columns=['start_time', 'end_time', 'attempt_hour'])
```

```
    grades_df['User full name'] = grades_df['First name'].str.strip() + ' ' + grades_df['Last
name'].str.strip()
    quiz_grade_cols = [col for col in grades_df.columns if col.startswith('Quiz:')]

    grades_melted = grades_df.melt(
        id_vars=['User full name'],
        value_vars=quiz_grade_cols,
        var_name='Gradebook_Column',
        value_name='Grade'
    )
     grades_melted['Event context'] = grades_melted['Gradebook_Column'].str.replace('
(Real)', '', regex=False)
    grades_melted['Grade'] = pd.to_numeric(grades_melted['Grade'].replace('-', np.nan),
errors='coerce')
    grades_clean = grades_melted.dropna(subset=['Grade'])

    final_dataset = pd.merge(
        logs_agg,
        grades_clean[['User full name', 'Event context', 'Grade']],
        on=['User full name', 'Event context'],
        how='inner'
    )

    student_totals = final_dataset.groupby('User full name')['Grade'].agg(['sum', 'count'])
    final_dataset = final_dataset.join(student_totals, on='User full name')
    class_avg = final_dataset['Grade'].mean()

    final_dataset['historical_grade'] = np.where(
        final_dataset['count'] > 1,
        (final_dataset['sum'] - final_dataset['Grade']) / (final_dataset['count'] - 1),
        class_avg
    )

          X_new = final_dataset[['total_clicks', 'unique_ips', 'duration_minutes',
'clicks_per_minute', 'is_late_night', 'historical_grade']]

    predictions = model.predict(X_new)
    final_dataset['Predicted_Grade'] = np.clip(predictions, 0, 100).round(2)

    dynamic_thresholds = calculate_dynamic_thresholds(X_new, explainer)
    shap_vals = explainer.shap_values(X_new)

    feedbacks = []
    for i in range(len(final_dataset)):
        fb = generate_feedback(shap_vals[i], list(X_new.columns), dynamic_thresholds)
        feedbacks.append(fb)

    final_dataset['AI_Companion_Feedback'] = feedbacks

      output_df = final_dataset[['User full name', 'Event context', 'historical_grade',
'Predicted_Grade', 'AI_Companion_Feedback']]
    output_df.loc[:, 'historical_grade'] = output_df['historical_grade'].round(2)
    return output_df
```

```
gr.close_all()

interface = gr.Interface(
    fn=process_log_file,
    inputs=[
        gr.File(label="Upload Moodle Log CSV"),
        gr.File(label="Upload Moodle Gradebook CSV")
    ],
    outputs=gr.Dataframe(label="AI Companion Evaluation Results"),
    title="AI Companion: Evaluasi Kebiasaan Moodle",
     description="Upload Log dan Gradebook untuk menghasilkan prediksi berbasis histori nyata beserta umpan balik XAI dinamis."
)

interface.launch(server_port=7860, share=False, inline=False, debug=False)

import urllib.request
import subprocess
import threading
import time

ip = urllib.request.urlopen('https://ipv4.icanhazip.com').read().decode('utf8').strip()
print("="*60)
print(f"KODE SANDI / ENDPOINT IP ANDA: {ip}")
print("(Copy angka IP di atas, Anda akan dimintai password saat membuka link)")
print("="*60)

!npm install -g localtunnel > /dev/null 2>&1

print("Membuat tautan publik yang stabil...")
!lt --port 7860
```

Link responder (kuisoner observasi awal): https://docs.google.com/spreadsheets/d/1v99_PdhuA_yE0mNFOx3dSE-ee49sDmAAV2VQH2hgC5w/edit?resourcekey=&gid=1526000840#gid=1526000840