

BAB II

LANDASAN TEORI

Bab ini menyajikan landasan teori dan kajian penelitian terdahulu yang menjadi dasar dalam pengembangan sistem automasi rekomendasi penilaian dokumen akademik. Pembahasan diawali dengan analisis penelitian terdahulu untuk mengidentifikasi perkembangan penelitian, menemukan kesenjangan penelitian (*research gap*), serta memperjelas posisi penelitian ini dibandingkan penelitian sebelumnya. Selanjutnya, bab ini menguraikan teori-teori yang mendukung penelitian, meliputi konsep *workflow automation*, perkembangan platform *workflow automation*, *Large Language Model* (LLM), GPT, *prompt engineering*, rubrik penilaian, JavaScript, serta berbagai tools yang digunakan dalam implementasi sistem. Selain itu, bab ini juga menjelaskan kerangka kerja yang digunakan dalam proses pengembangan sistem sebagai acuan pelaksanaan penelitian. Landasan teori tersebut menjadi dasar dalam perancangan, implementasi, dan evaluasi sistem yang dikembangkan.

2.1 Penelitian Terdahulu

Dalam beberapa tahun terakhir, penelitian mengenai pemanfaatan *workflow automation* dan *Large Language Model* (LLM) mengalami perkembangan yang cukup pesat di berbagai bidang, termasuk pendidikan tinggi. *Workflow automation* banyak dimanfaatkan untuk mengautomasi proses yang bersifat berulang sehingga mampu meningkatkan efisiensi, mengurangi kesalahan administratif, dan mempercepat penyelesaian pekerjaan. Di sisi lain, perkembangan LLM memungkinkan analisis dokumen dilakukan secara lebih cepat dengan kemampuan memahami konteks, menghasilkan ringkasan, serta memberikan rekomendasi berdasarkan instruksi yang diberikan. Kombinasi kedua teknologi tersebut membuka peluang untuk membangun sistem yang tidak hanya mampu mengautomasi alur kerja, tetapi juga mendukung proses analisis dokumen secara lebih terstruktur.

Penelitian terdahulu menunjukkan bahwa *workflow automation* telah diterapkan pada berbagai kebutuhan, seperti integrasi sistem, otomasi proses bisnis,

dan pengelolaan dokumen, sedangkan LLM banyak dimanfaatkan untuk menghasilkan umpan balik otomatis, membantu penilaian berbasis rubrik, dan mendukung proses pembelajaran. Meskipun demikian, sebagian besar penelitian masih mengembangkan kedua pendekatan tersebut secara terpisah. Oleh karena itu, penelitian ini mengkaji berbagai penelitian terdahulu yang relevan sebagai dasar dalam merancang sistem automasi rekomendasi penilaian dokumen akademik berbasis *workflow automation* dan LLM. Ringkasan penelitian terdahulu yang menjadi acuan dalam penelitian ini disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
1	<i>Generative AI vs. instructor vs. peer assessments: a comparison of grading and feedback in higher education;</i> Usher, M; [13]	<i>Assessment & Evaluation in Higher Education</i>	<i>Comparative Analysis</i>	Menemukan bahwa penilaian manual dosen memiliki variansi subjektivitas yang tinggi, sementara AI memberikan konsistensi yang lebih stabil dalam penilaian dokumen skala besar.
2	<i>Job stress and burnout among lecturers: a systematic literature review and meta analysis;</i> Yingying, S; Omar, M; Ismail, N; [14]	<i>Frontiers in Psychology</i>	<i>Meta-Analysis</i>	Mengonfirmasi bahwa aktivitas repetitif seperti koreksi manual dokumen merupakan kontributor utama stres kerja dosen yang menurunkan kualitas edukasi.
3	<i>Exploring the impact of feedback timing on student performance in online testing;</i> Ding, G; Li, M; Li, S; Wu, H; [15]	<i>Asia Pacific Education Review</i>	<i>Experimental Design</i>	Hasil penelitian menunjukkan keterlambatan penilaian manual menghambat progres belajar, yang dapat diatasi dengan sistem automasi umpan balik.
4	<i>n8n: An Open-Source Workflow Automation Platform for Enterprise Integration and AI-Driven Orchestration;</i> Venkiteela, P; [16]	<i>International Journal of Computer Applications</i>	<i>Architectural Analysis</i>	n8n terbukti mampu menangani volume data besar dengan biaya operasional 0 (nol) karena bersifat <i>self-hosted</i> . Penelitian mencatat n8n memiliki keunggulan pada

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
				Code <i>Node</i> yang memungkinkan manipulasi JSON kompleks hingga 80% kustomisasi dibanding Zapier yang bersifat kaku.
5	<i>Implementing Workflow Automation with N8N to Enhance Operational Efficiency and Performance in the Sharia Cooperative of Bank Indonesia, Aceh Province;</i> Wali, M; Nasir, N; Iqbal, T; [17]	<i>Journal Digital Technology Trend</i>	<i>Research & Development Framework, N8N, PostgreSQL</i>	Automasi menggunakan n8n berhasil memangkas waktu proses administrasi dari 15-20 menit per dokumen menjadi kurang dari 2 menit. Implementasi ini secara riil meningkatkan efisiensi operasional sebesar 70-85% pada tugas yang bersifat repetitif.
6	<i>Harnessing Generative AI (GenAI) for Automated Feedback in Higher Education: A Systematic Review;</i> Lee, S; Moore, R; [18]	<i>Online Learning Journal</i>	PRISMA	Penggunaan GenAI dalam memberikan umpan balik meningkatkan kepuasan mahasiswa karena kecepatan respon. Data menunjukkan 80% mahasiswa merasa umpan balik AI lebih detail dan objektif dibandingkan umpan balik manual yang seringkali hanya berupa nilai angka tanpa penjelasan.
7	<i>Reliability of ChatGPT-5.0 as an Automated Essay Scoring Tool: What Matters?</i> Trinh, T; [19]	<i>International Journal of English Language and Pedagogy</i>	ChatGPT, <i>Qualitative Feedback</i>	GPT-4 memiliki reliabilitas yang cukup tinggi untuk penilaian berbasis rubrik esai mahasiswa dibandingkan metode manual.
8	<i>Not a programmer? You can mobilize data from biodiversity informatics APIs, too!</i> Ower, Geoffrey, D; [20]	<i>Biodiversity Information Science and Standards</i>	<i>No-code/Low-code Workflow Automation</i> (menggunakan platform n8n.io)	Penggunaan n8n.io memungkinkan pengguna <i>non-programmer</i> untuk membangun alur kerja pemrosesan data yang kompleks melalui antarmuka grafis tanpa menulis

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
				kode. Sistem ini berhasil mengintegrasikan berbagai API informatika keanekaragaman hayati (seperti Wikidata dan <i>Catalogue of Life</i>) untuk pencarian ID spesies secara otomatis, pengiriman notifikasi <i>email</i> , dan sinkronisasi data dengan Google Sheets secara terjadwal.
9	<i>The future of grading programming assignments in education: The role of ChatGPT in automating the assessment and feedback process;</i> Jukiewicz, M; [21]	<i>Thinking Skills and Creativity</i>	Eksperimental & Komparatif. Menggunakan API ChatGPT (gpt-3.5-turbo) untuk menilai 1.579 tugas Python dari 67 mahasiswa selama 15 minggu.	Menemukan korelasi positif yang kuat ($r = 0,81$) antara nilai dosen dan ChatGPT. Namun, ChatGPT cenderung lebih "pelit" (nilai lebih rendah dibanding dosen). Pengulangan penilaian oleh AI menunjukkan hasil yang sangat konsisten (<i>negligible differences</i>).
10	<i>A Practical Evaluation of Self-Hosted n8n for Secure and Scalable Workflow Automation;</i> Pawar, S; Pattewar, S; Shelke, M; [7]	<i>International Journal of Scientific Research in Engineering and Management</i>	<i>Comparative Benchmarking Approach.</i> Peneliti membandingkan kinerja n8n yang di-self-host (menggunakan Docker pada VPS) dengan GitHub Actions. Pengujian dilakukan dengan menjalankan alur kerja (<i>workflow</i>) <i>build-test-deploy</i> sebanyak 20 kali untuk proyek berbasis Python dan Node.js.	Penelitian ini menunjukkan bahwa implementasi n8n secara <i>self-hosted</i> memberikan keunggulan performa yang signifikan yang dibandingkan dengan platform <i>cloud</i> seperti GitHub Actions. Dari sisi kecepatan, n8n mencatat rata-rata latensi eksekusi hanya sebesar 9,3 detik, yang menunjukkan reduksi waktu lebih dari 37% dibandingkan GitHub Actions yang memerlukan

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
				waktu 14,8 detik. Efisiensi ini juga didukung oleh penggunaan sumber daya sistem yang tergolong ringan, di mana n8n hanya mengonsumsi rata-rata CPU sebesar 41% dan RAM sebesar 520 MB selama proses berlangsung.
11	<i>A Python-Based Automation Script to Mark Computer-Aided Design Assessments;</i> <i>Tierney, G; [22]</i>	<i>Applied Sciences</i>	<i>Experimental Study</i>	Mengembangkan sistem automasi penilaian berbasis Python yang mampu mengurangi waktu penilaian hingga sekitar 89,7% dengan tingkat konsistensi yang tinggi pada penilaian CAD.
12	<i>Optimizing Project Management Using No-Code Automation via Microsoft Power Automate: A Systematic Review Team Case Study;</i> <i>Jones, E; Sorelle, C; [23]</i>	<i>Medical Reference Services Quarterly</i>	<i>Systematic Review & Case Study</i>	Microsoft Power Automate berhasil mengautomasi <i>workflow</i> manajemen proyek, meningkatkan koordinasi tim, mengurangi pekerjaan administratif, dan mempercepat penyelesaian proses operasional.
13	<i>Enhancing Curriculum Quality and Academic Readiness through Power BI and Automation;</i> <i>Kinley, K; Cosgrove K; Bourke, J; Booten, S; [24]</i>	<i>ASCILITE Publications</i>	<i>Case Study</i>	Integrasi Microsoft Power Automate dan Power BI berhasil mengautomasi pengumpulan data akademik serta mempercepat proses evaluasi kurikulum melalui <i>dashboard</i> otomatis.
14	<i>FlowMind: Automatic Workflow Generation with Large Language Models;</i> <i>Zeng, Z; Watson, W; Cho, N; Rahimi, S; Reynolds, S; Balch, T;</i>	<i>Proceedings of the Fourth ACM International Conference on AI in Finance</i>	<i>Framework Development</i>	Mengembangkan sistem yang memanfaatkan LLM untuk menghasilkan <i>workflow</i> secara otomatis berdasarkan kebutuhan pengguna

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
	<i>Veloso, M; [25]</i>			sehingga mempercepat proses perancangan <i>workflow</i> .
15	<i>Robotic Process Automation Using Process Mining: A Systematic Literature Review;</i> <i>El-Gharib, N; Amyot, D; [26]</i>	<i>Data & Knowledge Engineering</i>	<i>Systematic Literature Review</i>	Menunjukkan bahwa <i>workflow automation</i> berbasis RPA dan <i>process mining</i> mampu meningkatkan efisiensi proses bisnis, namun implementasinya masih didominasi pada domain non-pendidikan.

Berdasarkan pemetaan penelitian terdahulu pada Tabel 2.1, penelitian yang relevan dapat dikelompokkan ke dalam dua fokus utama, yaitu pemanfaatan *Large Language Model* (LLM) dalam penilaian akademik serta penerapan *workflow automation* untuk meningkatkan efisiensi proses kerja.

Kelompok pertama merupakan penelitian yang membahas pemanfaatan LLM dalam proses penilaian akademik dan pemberian umpan balik kepada mahasiswa. Penelitian yang dilakukan oleh Usher (2023) [13], Trinh (2025) [19], Jukiewicz (2024) [21], serta Lee dan Moore (2024) [18] menunjukkan bahwa LLM mampu meningkatkan konsistensi penilaian, mempercepat proses evaluasi, serta menghasilkan umpan balik yang lebih cepat dibandingkan metode manual. Selain itu, Tierney et al. (2025) [22] menunjukkan bahwa automasi penilaian berbasis Python pada tugas *Computer-Aided Design* mampu mengurangi waktu penilaian secara signifikan dengan tetap mempertahankan konsistensi hasil. Temuan tersebut menunjukkan bahwa automasi penilaian memiliki potensi untuk meningkatkan efisiensi proses evaluasi akademik. Namun demikian, sebagian besar penelitian tersebut masih memanfaatkan LLM maupun sistem automasi sebagai alat penilaian yang berdiri sendiri (*standalone assessment tool*) tanpa mengintegrasikannya ke dalam alur kerja penilaian yang terstruktur dan terdokumentasi secara menyeluruh.

Kelompok kedua merupakan penelitian yang berfokus pada penerapan *workflow automation* untuk meningkatkan efisiensi proses operasional melalui

berbagai platform dan pendekatan automasi. Penelitian oleh Venkiteela (2025) [16], Pawar et al. (2025) [7], Ower (2022) [20], serta *Journal Digital Technology Trend* [17] menunjukkan bahwa implementasi n8n mampu mengurangi pekerjaan yang bersifat repetitif, meningkatkan konsistensi proses, serta mempermudah integrasi antar aplikasi. Penelitian lain juga memperlihatkan perkembangan *workflow automation* pada platform yang berbeda. Jones et al. (2025) [23] memanfaatkan Microsoft Power Automate untuk mengautomasi proses manajemen proyek melalui pendekatan *no-code*, sedangkan Kinley et al. (2025) [24] mengintegrasikan Microsoft Power Automate dengan Power BI untuk mendukung evaluasi kurikulum dan pengolahan data akademik secara otomatis. Selain itu, Zeng et al. (2024) [25] melalui FlowMind menunjukkan bahwa LLM dapat dimanfaatkan untuk menghasilkan *workflow* secara otomatis, sedangkan El-Gharib dan Amyot (2022) [26] menjelaskan perkembangan *Robotic Process Automation* (RPA) dan *process mining* sebagai fondasi *workflow automation modern*. Meskipun demikian, sebagian besar penelitian tersebut masih berfokus pada automasi proses bisnis atau administrasi dan belum mengintegrasikan kemampuan LLM untuk menghasilkan rekomendasi penilaian terhadap dokumen akademik.

Berdasarkan kedua kelompok penelitian tersebut, masih terdapat peluang untuk mengintegrasikan *workflow automation* dan LLM ke dalam satu sistem yang mendukung proses penilaian dokumen akademik secara menyeluruh. Penelitian ini menggabungkan *workflow automation* berbasis n8n dengan LLM berbasis GPT dalam satu alur kerja yang mencakup pengambilan dokumen, ekstraksi isi dokumen, pemberian rekomendasi penilaian berdasarkan rubrik, hingga penyimpanan hasil secara otomatis. Dengan pendekatan tersebut, sistem yang dikembangkan tidak hanya meningkatkan efisiensi proses penilaian, tetapi juga tetap mempertahankan dosen sebagai pengambil keputusan akhir melalui mekanisme *human-in-the-loop*.

2.1.1 Sintesis dan Kontribusi Penelitian Terdahulu

Berdasarkan penelitian terdahulu yang telah dipaparkan pada Tabel 2.1, dapat disimpulkan bahwa perkembangan penelitian dalam bidang ini mengarah pada dua pendekatan utama, yaitu pemanfaatan *Large Language Model* (LLM) untuk

mendukung proses penilaian akademik dan penerapan *workflow automation* untuk meningkatkan efisiensi proses operasional. Penelitian yang memanfaatkan LLM umumnya berfokus pada peningkatan konsistensi penilaian, percepatan pemberian umpan balik, serta pengurangan beban kerja dosen dalam melakukan evaluasi terhadap dokumen mahasiswa. Di sisi lain, penelitian mengenai *workflow automation* lebih menitikberatkan pada automasi proses administratif, integrasi antar aplikasi, serta penyederhanaan alur kerja melalui pendekatan *low-code* atau *no-code*.

Meskipun kedua kelompok penelitian tersebut menunjukkan hasil yang positif, masih terdapat beberapa keterbatasan yang menjadi peluang bagi penelitian lebih lanjut. Penelitian yang memanfaatkan LLM umumnya menggunakan model bahasa sebagai alat penilaian yang berdiri sendiri (*standalone assessment tool*) tanpa mengintegrasikannya ke dalam alur kerja penilaian secara menyeluruh. Sebaliknya, penelitian mengenai *workflow automation*, baik yang menggunakan n8n maupun Microsoft Power Automate, lebih banyak berfokus pada peningkatan efisiensi proses bisnis dan administrasi tanpa memanfaatkan kemampuan LLM untuk melakukan analisis isi dokumen atau menghasilkan rekomendasi penilaian. Selain itu, penelitian terbaru seperti FlowMind menunjukkan bahwa LLM mulai dimanfaatkan dalam pembangunan *workflow* secara otomatis, namun penerapannya masih berorientasi pada automasi proses umum dan belum secara khusus menyasar proses penilaian dokumen akademik.

Berdasarkan sintesis tersebut, masih terdapat kesenjangan penelitian pada integrasi antara *workflow automation* dan LLM dalam mendukung proses penilaian dokumen akademik secara end-to-end. Penelitian ini berupaya mengisi kesenjangan tersebut dengan mengintegrasikan *workflow automation* berbasis n8n dan LLM berbasis GPT ke dalam satu alur kerja yang mencakup pengambilan dokumen, ekstraksi informasi, pemberian rekomendasi penilaian berdasarkan rubrik, hingga penyimpanan hasil penilaian secara otomatis. Pendekatan ini diharapkan mampu meningkatkan efisiensi proses penilaian, menjaga konsistensi rekomendasi yang dihasilkan, serta tetap mempertahankan dosen sebagai pengambil keputusan akhir melalui mekanisme *human-in-the-loop*.

2.1.2 Posisi Penelitian

Berdasarkan penelitian terdahulu, pemanfaatan *workflow automation* dan LLM telah banyak diterapkan untuk meningkatkan efisiensi proses kerja maupun mendukung penilaian akademik. Namun, sebagian besar penelitian masih mengembangkan kedua teknologi tersebut secara terpisah. Penelitian ini menempati posisi yang berbeda dengan mengintegrasikan *workflow automation* berbasis n8n dan LLM berbasis GPT ke dalam satu alur penilaian dokumen akademik yang terstruktur. Sistem yang dikembangkan tidak bertujuan menggantikan peran dosen, melainkan membantu menghasilkan rekomendasi penilaian awal berdasarkan rubrik sehingga proses penilaian menjadi lebih efisien, terdokumentasi, dan tetap berada di bawah kendali dosen sebagai pengambil keputusan akhir.

2.2 Teori yang berkaitan

2.2.1 Workflow Automation

Workflow automation merupakan pendekatan dalam mengotomasi alur kerja yang melibatkan tugas, dokumen, serta pertukaran informasi antar aktivitas kerja agar dapat berjalan secara mandiri sesuai dengan aturan bisnis yang telah ditetapkan [27]. *Workflow automation* memanfaatkan integrasi antar sistem sehingga proses yang sebelumnya dilakukan secara manual dapat dijalankan secara lebih efisien, konsisten, dan terdokumentasi dengan baik [28]. Dalam konteks organisasi, aturan bisnis yang digunakan dalam otomasi dapat disesuaikan dengan kebutuhan dan kebijakan internal yang berlaku [29].

Penerapan *workflow automation* bertujuan untuk mengurangi pekerjaan yang bersifat repetitif, meminimalkan kesalahan manusia (*human error*), meningkatkan kecepatan penyelesaian proses, serta memperbaiki kualitas pengelolaan informasi. Selain itu, *workflow automation* mendukung proses *monitoring*, audit, dan evaluasi karena setiap aktivitas yang dijalankan dapat direkam secara otomatis sehingga mempermudah pelacakan terhadap setiap tahapan proses [24].

Seiring berkembangnya teknologi informasi, *workflow automation* berkembang melalui berbagai platform berbasis *low-code* dan *no-code* yang

memungkinkan pengguna membangun alur kerja secara visual tanpa memerlukan proses pemrograman yang kompleks. Pendekatan ini memungkinkan pengguna membangun alur kerja secara visual dengan memanfaatkan komponen yang telah tersedia sehingga waktu pengembangan menjadi lebih singkat dan kebutuhan pemrograman dapat diminimalkan [25]. Beberapa platform workflow automation yang banyak digunakan saat ini antara lain Zapier, Make, Microsoft Power Automate, dan n8n. Masing-masing platform memiliki karakteristik yang berbeda dalam hal fleksibilitas integrasi, kemampuan kustomisasi, model implementasi, serta dukungan terhadap berbagai layanan berbasis *Application Programming Interface* (API) [7] [25].

2.2.2 Artificial Intelligence (AI)

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu meniru kemampuan kognitif manusia, seperti belajar, bernalar, mengenali pola, memahami bahasa, serta menyelesaikan permasalahan secara otomatis [30]. Kemampuan tersebut diperoleh melalui berbagai teknik komputasi, termasuk *machine learning*, *deep learning*, dan *natural language processing* (NLP), sehingga sistem dapat melakukan pengambilan keputusan maupun menghasilkan prediksi berdasarkan data yang diproses [31].

Perkembangan AI telah mendorong penerapannya pada berbagai bidang, termasuk pendidikan tinggi. AI dimanfaatkan untuk mendukung proses pembelajaran, analisis data akademik, personalisasi materi pembelajaran, serta automasi berbagai aktivitas yang sebelumnya dilakukan secara manual. Pemanfaatan AI membantu meningkatkan efisiensi proses kerja sekaligus memberikan dukungan dalam pengolahan informasi yang melibatkan data dalam jumlah besar [32].

Salah satu perkembangan AI yang saat ini banyak dimanfaatkan adalah *Generative Artificial Intelligence* (*Generative AI*), yaitu teknologi yang mampu menghasilkan berbagai jenis konten, seperti teks, gambar, audio, maupun kode program berdasarkan data dan instruksi yang diberikan pengguna. Perkembangan

Generative AI kemudian melahirkan *Large Language Model (LLM)*, yaitu model bahasa berskala besar yang dirancang untuk memahami konteks bahasa alami serta menghasilkan respons yang relevan berdasarkan masukan (*prompt*) yang diberikan [33].

2.2.3 Large Language Model (LLM)

Large Language Model (LLM) merupakan model pembelajaran mendalam (*deep learning*) yang dirancang untuk memahami, memproses, dan menghasilkan teks dalam bahasa alami berdasarkan data pelatihan berskala besar [34]. LLM dibangun menggunakan arsitektur *Transformer* yang memungkinkan model mempelajari hubungan antar kata maupun konteks dalam suatu kalimat secara lebih efektif dibandingkan pendekatan pemrosesan bahasa alami (*Natural Language Processing/NLP*) konvensional [35]. Model ini dilatih menggunakan miliaran hingga triliunan parameter sehingga mampu menghasilkan respons yang koheren, memahami konteks percakapan, melakukan penalaran, meringkas informasi, menerjemahkan bahasa, hingga menghasilkan berbagai bentuk konten tekstual [34].

Kemampuan utama LLM terletak pada pemahaman konteks bahasa secara menyeluruh (*contextual understanding*). Melalui mekanisme tersebut, LLM mampu menganalisis dokumen yang panjang, mengidentifikasi informasi penting, memahami hubungan semantik antar kalimat, serta menghasilkan keluaran yang relevan berdasarkan instruksi (*prompt*) yang diberikan pengguna [35]. Perkembangan kemampuan tersebut menjadikan LLM banyak dimanfaatkan pada berbagai bidang, seperti layanan pelanggan, pengembangan perangkat lunak, analisis dokumen, pembuatan konten, serta pendidikan [32].

Dalam bidang pendidikan, LLM mulai dimanfaatkan untuk mendukung berbagai aktivitas pembelajaran, seperti pemberian umpan balik otomatis, analisis dokumen akademik, penyusunan materi pembelajaran, serta asistensi dalam proses evaluasi. Berbagai penelitian menunjukkan bahwa pemanfaatan LLM mampu meningkatkan efisiensi proses evaluasi sekaligus membantu menghasilkan umpan balik yang lebih cepat dan konsisten dibandingkan proses manual [13] [18] [19].

Saat ini telah dikembangkan berbagai model LLM dengan karakteristik yang berbeda, di antaranya GPT yang dikembangkan oleh OpenAI, Gemini oleh Google, Claude oleh Anthropic, Llama oleh Meta, dan Mistral oleh Mistral AI. Masing-masing model memiliki kemampuan, arsitektur, serta karakteristik yang berbeda sesuai dengan tujuan pengembangannya. Di antara berbagai model tersebut, GPT menjadi salah satu LLM yang paling banyak digunakan karena memiliki kemampuan memahami konteks, menghasilkan teks yang koheren, serta mendukung berbagai tugas berbasis bahasa alami. Oleh karena itu, pembahasan mengenai GPT dijelaskan lebih lanjut pada subbab berikutnya.

2.2.4 Generative Pre-trained Transformer (GPT)

Generative Pre-trained Transformer (GPT) merupakan keluarga *Large Language Model* (LLM) yang dikembangkan oleh OpenAI untuk memahami serta menghasilkan teks dalam bahasa alami berdasarkan konteks yang diberikan pengguna. GPT dibangun menggunakan arsitektur *Transformer* dan memanfaatkan proses *pre-training* pada korpus data berskala besar sehingga mampu memahami hubungan antarkata, menangkap konteks, serta menghasilkan respons yang koheren pada berbagai tugas pemrosesan bahasa alami (*Natural Language Processing/NLP*) [36] [37]. Kemampuan tersebut memungkinkan GPT digunakan untuk menjawab pertanyaan, melakukan peringkasan dokumen, menerjemahkan bahasa, menghasilkan konten, hingga membantu analisis informasi secara otomatis [36].

Sejak diperkenalkan, GPT mengalami perkembangan yang signifikan dari sisi jumlah parameter, kemampuan penalaran, serta kualitas respons yang dihasilkan. GPT-3 memperkenalkan model bahasa berskala besar dengan kemampuan menghasilkan teks yang lebih alami dibandingkan generasi sebelumnya. Selanjutnya, GPT-3.5 meningkatkan kemampuan percakapan (*conversational AI*) sehingga lebih baik dalam memahami instruksi pengguna. Perkembangan berikutnya menghadirkan GPT-4 dengan peningkatan kemampuan penalaran (*reasoning*), pemahaman konteks, dan akurasi pada berbagai tugas akademik maupun profesional. Selain itu, GPT-4o (*omni*) dikembangkan sebagai model multimodal yang mampu memproses teks, gambar, dan suara secara terpadu dengan waktu respons yang lebih cepat serta efisiensi komputasi yang lebih baik [37].

Selain model utama GPT-4o, tersedia pula GPT-4o mini yang dirancang untuk memberikan keseimbangan antara performa, kecepatan respons, dan efisiensi komputasi. Model ini tetap mempertahankan kemampuan memahami konteks, mengikuti instruksi, serta menghasilkan teks yang koheren sehingga sesuai digunakan pada berbagai aplikasi yang memerlukan pemrosesan dokumen secara otomatis dalam jumlah besar [38].

Perkembangan GPT mendorong pemanfaatannya pada berbagai bidang, termasuk pendidikan. Model ini dimanfaatkan untuk menghasilkan umpan balik otomatis, membantu penyusunan materi pembelajaran, menganalisis dokumen akademik, serta mendukung proses evaluasi berbasis rubrik. Berbagai penelitian menunjukkan bahwa GPT mampu meningkatkan efisiensi proses evaluasi dan menghasilkan respons yang konsisten ketika diberikan instruksi yang terstruktur melalui *prompt* [13] [18] [19]. Oleh karena itu, kualitas keluaran GPT sangat dipengaruhi oleh cara penyusunan instruksi yang diberikan, sehingga teknik *prompt engineering* menjadi aspek penting dalam pemanfaatan LLM dan dibahas lebih lanjut pada subbab berikutnya.

2.2.5 Prompt Engineering

Prompt engineering merupakan teknik penyusunan instruksi (*prompt*) secara sistematis untuk mengarahkan *Large Language Model* (LLM) agar menghasilkan keluaran yang akurat, relevan, dan sesuai dengan tujuan yang diinginkan tanpa mengubah parameter internal model [39]. *Prompt* dapat berupa pertanyaan, perintah, pemberian konteks, contoh, maupun batasan tertentu yang disusun menggunakan bahasa alami sehingga model mampu memahami tugas yang diberikan secara lebih tepat [39]. Seiring berkembangnya pemanfaatan LLM, *prompt engineering* menjadi salah satu faktor penting yang menentukan kualitas respons yang dihasilkan oleh model pada berbagai bidang, termasuk pendidikan, analisis dokumen, dan pengembangan perangkat lunak [39].

Berdasarkan cara penyusunannya, *prompt engineering* dapat dibedakan menjadi beberapa pendekatan, yaitu *zero-shot prompting*, *one-shot prompting*, *few-shot prompting*, dan *structured prompting*. *Zero-shot prompting* memberikan

instruksi tanpa disertai contoh, sedangkan *one-shot prompting* dan *few-shot prompting* menyertakan satu atau beberapa contoh sebagai acuan bagi model dalam menghasilkan respons. Sementara itu, *structured prompting* menyusun instruksi secara lebih sistematis dengan memberikan konteks, tujuan, aturan, maupun format keluaran yang diharapkan sehingga respons yang dihasilkan menjadi lebih konsisten [39] [40].

Salah satu tantangan dalam pemanfaatan LLM adalah risiko *hallucination*, yaitu kondisi ketika model menghasilkan informasi yang tidak sesuai dengan fakta, tidak didukung oleh sumber yang valid, atau menyimpulkan informasi yang sebenarnya tidak terdapat pada data masukan [40]. Risiko tersebut dapat diminimalkan melalui penyusunan *prompt* yang jelas, spesifik, serta dilengkapi dengan konteks yang memadai sehingga model memiliki batasan yang lebih baik dalam menghasilkan respons [40]. Oleh karena itu, *prompt engineering* tidak hanya berfungsi untuk meningkatkan kualitas keluaran LLM, tetapi juga membantu meningkatkan konsistensi, reliabilitas, dan relevansi informasi yang dihasilkan.

2.2.6 Human-in-the-Loop (HITL)

Human-in-the-Loop (HITL) merupakan pendekatan dalam pengembangan sistem kecerdasan buatan yang melibatkan manusia secara aktif dalam proses pengambilan keputusan maupun evaluasi hasil yang dihasilkan oleh sistem AI. Pendekatan ini menempatkan manusia sebagai pengawas (*human oversight*) yang berperan untuk memvalidasi, mengoreksi, maupun menyempurnakan keluaran AI sehingga keputusan akhir tetap berada di bawah kendali manusia [41]. Dengan demikian, AI berfungsi sebagai alat bantu yang meningkatkan efisiensi proses, bukan sebagai pengganti pertimbangan manusia [42] [41].

Sebagai pendekatan kolaboratif, HITL menggabungkan kemampuan AI dalam memproses data secara cepat dengan kemampuan manusia dalam memberikan penilaian kontekstual, pertimbangan etis, serta pengambilan keputusan yang memerlukan pengalaman dan pengetahuan khusus. Keterlibatan manusia pada berbagai tahapan, mulai dari proses validasi hingga evaluasi hasil, membantu

meningkatkan kualitas keluaran AI sekaligus mengurangi risiko kesalahan maupun bias yang mungkin muncul selama proses inferensi [41].

Dalam bidang pendidikan, pendekatan Human-in-the-Loop (HITL) menjadi salah satu prinsip penting dalam pemanfaatan *Large Language Model* (LLM). Kehadiran tenaga pendidik sebagai pengawas memungkinkan keluaran yang dihasilkan AI ditinjau kembali sebelum digunakan sebagai dasar pengambilan keputusan akademik. Pendekatan ini mendukung transparansi, akuntabilitas, serta penggunaan AI secara bertanggung jawab (*responsible AI*), khususnya pada aktivitas yang memerlukan pertimbangan profesional, seperti evaluasi dokumen akademik dan pemberian umpan balik kepada mahasiswa [32]. Oleh karena itu, AI diposisikan sebagai alat bantu yang mendukung proses penilaian, sementara keputusan akhir tetap berada pada tenaga pendidik sebagai pihak yang memiliki otoritas akademik [32].

2.2.7 Penilaian Dokumen Berbasis Rubrik

Penilaian dokumen berbasis rubrik merupakan metode evaluasi yang menggunakan seperangkat kriteria dan tingkat pencapaian yang telah ditentukan untuk menilai kualitas suatu hasil pekerjaan atau dokumen akademik mahasiswa [43]. Rubrik memberikan pedoman yang jelas mengenai aspek-aspek yang dinilai beserta indikator pencapaiannya, sehingga proses penilaian dapat dilakukan secara lebih objektif, konsisten, dan transparan. Selain membantu dosen dalam melakukan evaluasi, rubrik juga memberikan informasi kepada mahasiswa mengenai standar kualitas yang diharapkan pada setiap tugas atau dokumen yang dikerjakan [43].

Penerapan rubrik dalam proses penilaian memberikan berbagai manfaat, antara lain meningkatkan konsistensi penilaian antar penilai, mempermudah proses evaluasi, mengurangi subjektivitas, serta memberikan umpan balik yang lebih terarah kepada mahasiswa. Dengan adanya kriteria yang terstruktur, setiap aspek penilaian dapat dievaluasi secara sistematis sesuai dengan bobot yang telah ditentukan sehingga hasil penilaian menjadi lebih terukur dan dapat dipertanggungjawabkan [44].

Secara umum, rubrik penilaian dapat dibedakan menjadi beberapa jenis, yaitu *holistic rubric*, *analytic rubric*, *checklist rubric*, dan *rating scale rubric*. *Holistic rubric* memberikan satu nilai secara keseluruhan berdasarkan kualitas pekerjaan, sedangkan *analytic rubric* mengevaluasi setiap kriteria secara terpisah sehingga menghasilkan penilaian yang lebih rinci. *Checklist rubric* digunakan untuk memeriksa keberadaan komponen tertentu pada suatu pekerjaan, sementara *rating scale rubric* memberikan penilaian berdasarkan tingkat pencapaian pada setiap indikator yang telah ditentukan [45].

Dalam pendidikan tinggi, *analytic rubric* merupakan jenis rubrik yang paling banyak digunakan pada penilaian dokumen akademik, seperti jawaban esai, laporan praktikum, studi kasus, maupun tugas proyek. Jenis rubrik ini memungkinkan setiap aspek penilaian dievaluasi secara terpisah sehingga memberikan hasil yang lebih rinci, konsisten, dan mudah dipahami baik oleh dosen maupun mahasiswa [44].

Dalam pendidikan tinggi, rubrik digunakan pada berbagai bentuk penilaian dokumen, seperti jawaban esai pada ujian tengah semester (UTS) dan ujian akhir semester (UAS), laporan praktikum, studi kasus, proposal, laporan penelitian, maupun tugas proyek. Masing-masing bentuk penilaian memerlukan indikator yang berbeda sesuai dengan capaian pembelajaran mata kuliah sehingga rubrik dapat disesuaikan dengan karakteristik dokumen yang dinilai [44].

2.2.8 Metrik Evaluasi Kinerja Sistem Penilaian

Dalam mengevaluasi kinerja sistem rekomendasi penilaian berbasis *Large Language Model* (LLM), diperlukan metrik statistik untuk mengukur tingkat kesesuaian antara hasil rekomendasi sistem dengan penilaian yang diberikan oleh dosen. Pengukuran tersebut bertujuan untuk mengetahui tingkat akurasi, besarnya kesalahan prediksi, serta hubungan antara hasil rekomendasi sistem dan penilaian manusia. Penelitian ini menggunakan tiga metrik evaluasi, yaitu *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan Koefisien Korelasi Pearson.

2.2.9.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah metrik evaluasi yang digunakan untuk mengukur rata-rata besaran kesalahan (*error*) secara mutlak antara nilai prediksi

yang dihasilkan sistem dengan nilai aktual [46]. Metrik ini memberikan gambaran langsung mengenai seberapa jauh simpangan model tanpa mempertimbangkan arah kesalahan (apakah prediksi lebih tinggi atau lebih rendah). Persamaan matematis MAE adalah sebagai berikut:

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

Sumber: [46]

Keterangan:

n = Jumlah sampel data

y_i = Nilai aktual yang diberikan oleh dosen pada sampel ke- i

$\hat{y}_i$ = Nilai prediksi dari sistem (skor AI) pada sampel ke- i

2.2.9.2 Root Mean Square Error (RMSE)

Berbeda dengan MAE yang menghitung rata-rata kesalahan absolut, *Root Mean Square Error* (RMSE) mengukur akar dari rata-rata kuadrat kesalahan. Karakteristik utama dari RMSE adalah memberikan bobot penalti yang lebih besar pada kesalahan (deviasi) yang bernilai besar, karena selisih nilai akan dikuadratkan terlebih dahulu sebelum dirata-ratakan. Persamaan matematis RMSE dirumuskan sebagai berikut:

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

Sumber: [46]

Metrik ini sangat berguna untuk mendeteksi apakah sistem sering melakukan kesalahan prediksi yang ekstrem secara signifikan. Sama seperti MAE, nilai RMSE yang rendah menunjukkan performa sistem yang lebih baik.

2.2.9.3 Koefisien Korelasi Pearson (r)

Koefisien Korelasi Pearson digunakan untuk mengukur kekuatan dan arah hubungan linear antara dua variabel numerik, dalam hal ini adalah skor aktual dan skor prediksi. Nilai koefisien korelasi (r) berada pada rentang -1 hingga 1.

$$r = \frac{n(\sum y\hat{y}) - (\sum y)(\sum \hat{y})}{\sqrt{[n\sum y^2 - (\sum y)^2][n\sum \hat{y}^2 - (\sum \hat{y})^2]}} \quad (3)$$

Sumber: [47]

Keterangan:

r = Koefisien korelasi Pearson

y = Skor aktual penilai manusia (dosen)

$\hat{y}$ = Skor prediksi sistem AI

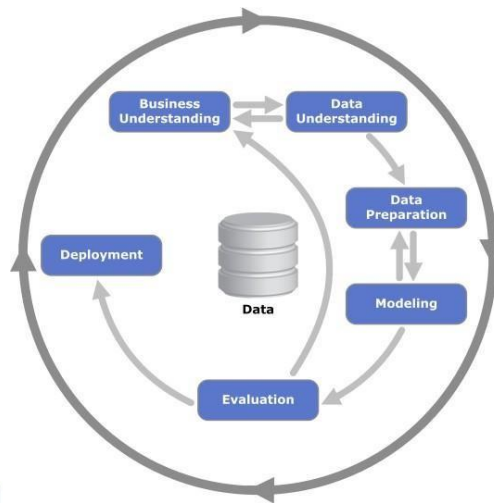
n = Jumlah sampel

Nilai r yang mendekati 1 menunjukkan adanya hubungan linear positif yang kuat (jika dosen memberi nilai tinggi, AI juga memberi nilai tinggi), sedangkan nilai mendekati 0 menunjukkan tidak adanya hubungan linear antar kedua penilaian.

2.3 Framework Pengembangan Sistem

2.3.1 CRISP-DM

Penelitian ini menggunakan CRISP-DM (Cross-Industry Standard Process for Data Mining) sebagai framework pengembangan sistem untuk memberikan alur kerja yang sistematis dalam proses pengembangan sistem rekomendasi penilaian dokumen berbasis workflow automation dan *Large Language Model* (LLM). CRISP-DM digunakan sebagai acuan dalam memahami permasalahan, menyiapkan data, merancang solusi, mengevaluasi hasil, hingga implementasi sistem. Pada penelitian ini, CRISP-DM digunakan sebagai framework pengembangan sistem yang memberikan alur kerja pengembangan secara sistematis, bukan sebagai metode penelitian [48]. CRISP-DM pertama kali diperkenalkan pada tahun 1996 sebagai kerangka kerja standar untuk proyek berbasis data dan hingga saat ini masih banyak digunakan karena memiliki tahapan yang sistematis dan bersifat iteratif [49].



Gambar 2.1 Diagram Tahapan CRISP-DM [50]

Gambar 2.1 menunjukkan diagram tahapan utama dalam kerangka kerja CRISP-DM. Diagram ini menggambarkan hubungan iteratif dan siklus yang mencakup enam tahapan, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. Hubungan iteratif ini menunjukkan bahwa proses pengembangan sistem berbasis data dan kecerdasan buatan tidak selalu berjalan secara linear, sehingga revisi atau pengulangan pada tahapan tertentu dapat dilakukan untuk mencapai hasil yang optimal. Kerangka kerja CRISP-DM bersifat fleksibel dan dapat diterapkan pada berbagai jenis proyek berbasis data, termasuk pengembangan sistem automasi penilaian dokumen menggunakan *workflow automation* dan kecerdasan buatan seperti pada penelitian ini. Berikut penjelasan masing-masing tahapan pada CRISP-DM:

1. *Business Understanding*

Tahapan ini bertujuan untuk memahami permasalahan dan kebutuhan proses penilaian dokumen secara menyeluruh. Pada fase ini dilakukan identifikasi permasalahan utama, yaitu proses penilaian yang masih dilakukan secara manual dan memakan waktu lama. Selanjutnya ditentukan tujuan penelitian, yaitu membangun sistem automasi penilaian dokumen berbasis *workflow automation* dan Large Language Model (LLM). Selain itu, ditetapkan pula kriteria keberhasilan sistem, seperti peningkatan efisiensi waktu, konsistensi hasil penilaian, serta kemudahan integrasi dengan alur kerja yang

telah ada. Rencana penelitian disusun secara terstruktur sebagai panduan pelaksanaan pengembangan sistem.

2. *Data Understanding*

Pada fase ini dilakukan identifikasi dan pengumpulan data yang digunakan dalam sistem, berupa dokumen UTS atau laporan mahasiswa beserta informasi pendukungnya. Selanjutnya dilakukan pemahaman terhadap struktur, format, dan karakteristik dokumen yang akan diproses oleh sistem. Analisis awal dilakukan untuk memastikan data yang digunakan relevan, memiliki kualitas yang memadai, serta dapat diproses secara otomatis oleh *workflow automation* dan model AI. Tahap ini juga mencakup identifikasi karakteristik dokumen akademik, struktur rubrik penilaian, serta kebutuhan informasi yang akan diproses oleh sistem.

3. *Data Preparation*

Fase *Data Preparation* melibatkan proses penyiapan dokumen sebelum dianalisis lebih lanjut. Tahapan ini mencakup validasi format *file*, pembersihan teks, serta ekstraksi konten utama dari dokumen. Proses ini bertujuan untuk memastikan bahwa *input* yang diberikan kepada sistem Large Language Model (LLM) berada dalam format yang konsisten dan mudah dianalisis. Pendekatan yang digunakan bersifat fleksibel dan disesuaikan dengan kebutuhan penilaian serta karakteristik dokumen.

4. *Modeling*

Tahap *modeling* dalam penelitian ini tidak berfokus pada pelatihan model *data mining* secara konvensional, melainkan pada perancangan logika analisis dan integrasi *Large Language Model* berbasis GPT ke dalam *workflow automation*. Pada fase ini dilakukan perancangan *prompt* penilaian berbasis rubrik, pengaturan alur pemanggilan API LLM berbasis GPT, serta pengolahan hasil analisis agar sesuai dengan format penilaian yang diinginkan. Model AI digunakan sebagai alat bantu analisis dokumen untuk menghasilkan rekomendasi penilaian awal secara otomatis.

5. Evaluation

Pada fase evaluasi, hasil penilaian yang dihasilkan oleh sistem automasi dibandingkan dengan tujuan penelitian yang telah ditetapkan pada tahap *Business Understanding*. Evaluasi dilakukan untuk menilai efektivitas sistem dalam meningkatkan efisiensi waktu penilaian, menjaga konsistensi hasil, serta membantu dosen dalam proses evaluasi dokumen. Evaluasi dilakukan menggunakan metrik MAE, RMSE, dan Koefisien Korelasi Pearson untuk mengukur tingkat kesesuaian antara hasil rekomendasi sistem dan hasil penilaian dosen. Hasil evaluasi ini digunakan sebagai dasar untuk melakukan perbaikan dan penyempurnaan sistem.

6. Deployment

Fase *deployment* merupakan tahap penerapan sistem automasi penilaian ke dalam lingkungan operasional. Pada tahap ini, *workflow automation* dijalankan menggunakan platform n8n, termasuk proses pemantauan dan pemeliharaan sistem. *Deployment* bertujuan untuk memastikan sistem dapat digunakan secara stabil, berkelanjutan, dan sesuai dengan kebutuhan pengguna.

Pemilihan CRISP-DM didasarkan pada karakteristik penelitian yang berorientasi pada pemrosesan data dan analisis dokumen menggunakan LLM. Berbeda dengan pendekatan pengembangan perangkat lunak yang bersifat linear, CRISP-DM menyediakan proses yang bersifat iteratif sehingga memungkinkan evaluasi dan penyempurnaan dilakukan pada setiap tahapan apabila ditemukan ketidaksesuaian hasil. Karakteristik tersebut menjadikan CRISP-DM sesuai digunakan sebagai framework pengembangan sistem pada penelitian ini [51].

Implementasi setiap tahapan CRISP-DM pada penelitian ini dijelaskan lebih lanjut pada Bab III. Tahap *Business Understanding* digunakan untuk menganalisis kebutuhan sistem, *Data Understanding* dan *Data Preparation* diterapkan pada proses identifikasi serta pengolahan dokumen, *Modeling* diwujudkan melalui integrasi workflow automation dan LLM berbasis GPT, sedangkan *Evaluation* dan *Deployment* dilakukan melalui pengujian kinerja serta implementasi sistem menggunakan platform n8n.

2.3.2 Mekanisme Sistem

Penelitian ini mengembangkan sistem automasi penilaian dokumen dengan mengintegrasikan workflow automation, *Large Language Model* (LLM), dan rubrik penilaian ke dalam satu alur kerja yang saling terhubung. Mekanisme sistem dirancang untuk mengautomasi proses penilaian dokumen mulai dari penerimaan dokumen, pemrosesan isi dokumen, pemberian rekomendasi penilaian berdasarkan rubrik, hingga penyimpanan hasil secara otomatis.

Secara umum, mekanisme sistem pada penelitian ini terdiri atas tiga komponen utama, yaitu orkestrasi workflow (*workflow orchestration*), analisis dokumen menggunakan *Large Language Model* (LLM), dan penilaian dokumen berbasis rubrik (*rubric-based evaluation*). Ketiga komponen tersebut bekerja secara terpadu untuk menghasilkan proses penilaian yang lebih efisien, konsisten, dan terdokumentasi dengan baik.

2.3.2.1 Orkestrasi Workflow

Orkestrasi *workflow* pada penelitian ini diimplementasikan menggunakan platform n8n yang menerapkan mekanisme *rule-based workflow*. Mekanisme ini mengatur urutan eksekusi setiap proses berdasarkan aturan logis (*if-then rules*) sehingga seluruh tahapan penilaian dokumen dapat berjalan secara otomatis mulai dari penerimaan dokumen, validasi berkas, ekstraksi teks, pemanggilan API LLM, hingga penyimpanan hasil penilaian [52]. Mekanisme ini memungkinkan setiap proses dijalankan secara berurutan maupun berdasarkan kondisi tertentu (*conditional workflow*), sehingga alur kerja menjadi lebih efisien, terdokumentasi, mudah dipantau, serta memudahkan proses pelacakan (*traceability*) apabila terjadi kesalahan pada salah satu tahapan proses.

2.3.2.2 Analisis Dokumen Menggunakan Large Language Model (LLM)

Dari sisi mekanisme pemrosesan, LLM memanfaatkan arsitektur *Transformer* dengan mekanisme *self-attention* untuk memahami hubungan antarbagian dalam dokumen sehingga mampu menghasilkan analisis yang mempertimbangkan konteks secara menyeluruh [53]. LLM menerima teks sebagai masukan (*input*) untuk dianalisis berdasarkan konteks yang diberikan melalui *prompt*. Hasil analisis

berupa rekomendasi penilaian dan umpan balik dihasilkan sesuai dengan kriteria yang telah ditentukan. Oleh karena itu, pada penelitian ini LLM berperan sebagai alat bantu analisis yang menghasilkan rekomendasi penilaian, sedangkan keputusan akhir tetap berada pada dosen sebagai penilai melalui pendekatan *Human-in-the-Loop*.

2.3.2.3 Penilaian Dokumen Berbasis Rubrik (Rubric-Based Evaluation)

Mekanisme penilaian pada penelitian ini menggunakan pendekatan *rubric-based evaluation*. Rubrik penilaian disusun dalam bentuk kriteria dan indikator penilaian yang kemudian diterjemahkan ke dalam *prompt* sehingga LLM dapat mengevaluasi setiap aspek dokumen sesuai dengan rubrik yang telah ditetapkan [54]. Nilai dari setiap aspek kemudian digabungkan untuk menghasilkan rekomendasi nilai beserta umpan balik yang sesuai dengan rubrik penilaian. Pendekatan ini membantu menjaga konsistensi, transparansi, dan objektivitas penilaian serta mengurangi subjektivitas yang sering muncul pada proses penilaian manual. Selain itu, penggunaan rubrik memungkinkan setiap hasil rekomendasi memiliki dasar penilaian yang jelas dan sesuai dengan capaian pembelajaran yang telah ditetapkan [54].

2.4 Tools/software yang digunakan

Dalam penelitian ini digunakan beberapa *tools* dan perangkat lunak untuk mendukung proses pengembangan, implementasi, serta pengujian sistem. Pemilihan *tools* didasarkan pada kemampuan integrasi, fleksibilitas pengembangan, serta kesesuaiannya dengan kebutuhan *workflow automation* dan pemrosesan dokumen akademik. *Tools* yang digunakan meliputi platform n8n, JavaScript, OpenAI API, Google Workspace (Google Drive, Google Docs, Google Sheets, dan Gmail), serta lingkungan *self-hosting* yang mendukung implementasi *workflow* secara mandiri.

2.4.1 N8n

n8n merupakan platform automasi *workflow* berbasis *open-source* yang bersifat *low-code/no-code*, dirancang untuk mempermudah proses automasi alur kerja dengan menghubungkan berbagai aplikasi dan layanan tanpa memerlukan

keahlian pemrograman yang kompleks [55]. Platform ini memungkinkan pengguna untuk membangun *workflow* secara visual melalui antarmuka berbasis *node*, sehingga proses automasi dapat dirancang secara intuitif dan efisien.

Dalam praktiknya, n8n banyak digunakan untuk mengautomasi pemrosesan data, pengelolaan dokumen, integrasi API, pengiriman notifikasi, serta sinkronisasi data antar aplikasi [7]. Kemampuan tersebut menjadikan n8n sesuai digunakan pada sistem yang memerlukan integrasi beberapa layanan secara otomatis, termasuk pada proses penilaian dokumen akademik.

n8n bekerja dengan konsep *node-based workflow*, di mana setiap *node* merepresentasikan satu fungsi atau layanan tertentu. Salah satu keunggulan n8n adalah tersedianya *Code Node*, yang memungkinkan pengguna menambahkan kode JavaScript untuk mengimplementasikan logika bisnis khusus atau melakukan transformasi data yang tidak dapat dilakukan secara *default* [17]. Fleksibilitas ini menjadikan n8n sangat adaptif terhadap kebutuhan organisasi yang memiliki aturan atau proses bisnis yang spesifik.

Pemilihan n8n dibandingkan dengan platform automasi komersial lainnya seperti Zapier atau Make didasarkan pada arsitekturnya yang mendukung opsi *self-hosting*. Fitur ini memberikan kontrol maksimal atas kedaulatan data (*data sovereignty*) dan privasi infrastruktur, yang seringkali menjadi limitasi pada platform berbayar berbasis *cloud-only* [56]. Selain itu, evaluasi komparatif menunjukkan bahwa n8n menawarkan efisiensi biaya (*cost efficiency*) yang lebih baik karena membebaskan pengguna dari ketergantungan skema harga per tugas tunggal (*task-based pricing*), serta memiliki skalabilitas dan kemampuan orkestrasi tingkat lanjut untuk mengintegrasikan berbagai ekosistem secara mandiri [16].

Berdasarkan karakteristik tersebut, penelitian ini menggunakan n8n sebagai platform utama *workflow automation* yang mengorkestrasi seluruh proses penilaian dokumen, mulai dari pengambilan data, pemrosesan dokumen, pemanggilan LLM, hingga penyimpanan hasil secara otomatis.

Untuk memberikan gambaran konseptual mengenai komponen dasar yang membangun alur kerja tersebut, Tabel 2.2 menyajikan klasifikasi fitur dan *node*

utama pada platform n8n. Klasifikasi ini disusun berdasarkan peran fungsional masing-masing komponen dalam mendukung pembentukan *workflow automation*, serta dilengkapi dengan contoh implementasinya dalam sistem yang dikembangkan pada penelitian ini.

Tabel 2.2 Klasifikasi Fitur dan *Node* pada Platform n8n

Kategori Fungsional	Deskripsi Konseptual	Contoh <i>Node</i> dalam Sistem
<i>Trigger Nodes</i> (Pemicu)	Komponen yang berfungsi sebagai titik awal eksekusi <i>workflow</i> dengan merespons suatu kejadian (<i>event</i>) tertentu dari sistem eksternal atau berdasarkan jadwal waktu.	<i>Google Sheets Trigger</i> (memulai <i>workflow</i> secara otomatis saat terdapat data mahasiswa baru dari Google Form).
<i>Integration / Action Nodes</i> (Aksi)	Komponen yang menjalankan aksi spesifik melalui integrasi dengan layanan pihak ketiga menggunakan <i>Application Programming Interface</i> (API).	Google Drive (unduh/unggah dokumen), Google Docs (pembuatan laporan), Gmail (pengiriman <i>email</i>), HTTP Request (akses API model AI).
<i>Data Transformation & Logic Nodes</i> (Pemrosesan Data)	Komponen yang digunakan untuk memproses, membersihkan, dan mentransformasi data mentah menjadi format yang sesuai untuk analisis lebih lanjut.	<i>Code</i> (JavaScript untuk logika bisnis), <i>Extract from File</i> (ekstraksi teks dari PDF), <i>Set</i> (penyimpanan dan manipulasi variabel data).
<i>Flow Control Nodes</i> (Pengendali Alur)	Komponen yang mengatur alur eksekusi <i>workflow</i> , termasuk percabangan logika, pengulangan proses, dan pengelolaan <i>batch</i> data.	<i>Split In Batches</i> (memproses data mahasiswa secara bertahap dalam loop).

Klasifikasi tersebut menunjukkan bahwa platform n8n tidak hanya berfungsi sebagai alat automasi, tetapi juga sebagai orkestrator proses yang mampu mengintegrasikan pengambilan data, pemrosesan dokumen, analisis menggunakan *Large Language Model* (LLM), serta distribusi hasil dalam satu alur kerja yang terstruktur. Secara operasional, setiap kategori *node* bekerja secara berkesinambungan membentuk sebuah *workflow pipeline*, dimulai dari *Trigger Node* yang menerima data, dilanjutkan dengan proses transformasi data melalui *Processing Nodes*, pemanggilan layanan eksternal menggunakan *Integration Nodes*, hingga pengendalian alur menggunakan *Flow Control Nodes*.

Dalam penelitian ini, mekanisme tersebut memungkinkan setiap dokumen mahasiswa diproses secara otomatis dan terpisah melalui proses *batching*, sehingga menjaga konsistensi pemrosesan, meminimalkan risiko kesalahan, serta memudahkan proses pelacakan terhadap setiap tahapan *workflow*. Dengan demikian, n8n berperan sebagai platform *workflow automation* yang mengintegrasikan seluruh proses penilaian dokumen secara *end-to-end*, sementara keputusan akhir terhadap hasil penilaian tetap berada pada dosen melalui pendekatan *Human-in-the-Loop*.

2.4.2 JavaScript

JavaScript merupakan bahasa pemrograman yang banyak digunakan untuk membangun aplikasi web maupun mengembangkan proses automasi melalui manipulasi data, pengolahan logika program, serta integrasi dengan berbagai *Application Programming Interface* (API) [57]. Seiring berkembangnya teknologi, JavaScript tidak hanya digunakan pada sisi klien (*client-side*), tetapi juga dapat dijalankan pada sisi server (*server-side*) melalui lingkungan eksekusi seperti Node.js, sehingga menjadi salah satu bahasa pemrograman yang fleksibel dalam pengembangan aplikasi modern.

Pada platform *workflow automation* seperti n8n, JavaScript digunakan untuk menjalankan logika pemrosesan data yang tidak dapat diselesaikan hanya dengan konfigurasi node standar. JavaScript memungkinkan proses validasi data, manipulasi teks, transformasi struktur data, penerapan ekspresi reguler (*regular expression*), serta pengolahan hasil dari berbagai layanan yang terintegrasi melalui API [58].

Dalam penelitian ini, JavaScript digunakan sebagai bahasa pemrograman pendukung untuk mengimplementasikan logika pemrosesan data pada *workflow automation*. JavaScript dimanfaatkan untuk melakukan ekstraksi dan pembersihan teks dokumen, transformasi data, penyusunan *prompt* untuk LLM, pengolahan hasil rekomendasi penilaian, serta konversi data ke dalam format yang digunakan pada proses penyimpanan dan pelaporan. Dengan demikian, JavaScript berperan sebagai

komponen pendukung integrasi antarproses dalam *workflow automation* yang dikembangkan.

2.4.3 Self-Hosting

Self-hosting merupakan pendekatan di mana individu atau organisasi menjalankan dan mengelola perangkat lunak atau layanan digital pada infrastruktur milik sendiri, baik berupa server fisik maupun virtual, tanpa bergantung sepenuhnya pada penyedia layanan pihak ketiga [59]. Pendekatan ini memberikan kontrol penuh terhadap sistem, data, serta konfigurasi lingkungan kerja.

Keunggulan utama dari *self-hosting* terletak pada aspek keamanan data, privasi, dan fleksibilitas kustomisasi [59]. Melalui eksekusi sistem secara mandiri, organisasi dapat memastikan bahwa data sensitif tidak dikelola oleh pihak eksternal, serta dapat menyesuaikan konfigurasi sistem sesuai kebutuhan internal. Namun demikian, *self-hosting* juga menuntut kemampuan teknis yang memadai, karena tanggung jawab pemeliharaan, pembaruan sistem, serta pengamanan infrastruktur berada sepenuhnya pada pengelola.

Dalam penelitian ini, n8n diimplementasikan menggunakan pendekatan *self-hosting* dengan memanfaatkan teknologi virtualisasi dan *container* seperti *Docker*. Implementasi ini memungkinkan *workflow automation* dijalankan pada server yang dikelola secara mandiri sehingga seluruh konfigurasi sistem, data, dan lingkungan eksekusi berada di bawah kendali pengelola.

Self-hosted n8n juga mengurangi ketergantungan terhadap layanan eksternal dan meningkatkan reliabilitas sistem, terutama untuk kebutuhan penelitian yang memerlukan kestabilan dan konsistensi lingkungan pengujian [60]. Oleh karena itu, pendekatan *self-hosting* dipilih sebagai solusi yang sesuai untuk mendukung *workflow automation* yang sesuai dengan kebutuhan penelitian yang melibatkan pemrosesan dokumen akademik dan integrasi berbagai layanan melalui API.

2.4.4 Google Workspace Services

Google Workspace merupakan platform produktivitas berbasis cloud yang menyediakan berbagai layanan untuk mendukung pengelolaan dokumen, penyimpanan data, serta kolaborasi secara daring [61]. Dalam penelitian ini,

layanan Google Workspace yang digunakan meliputi Google Drive, Google Docs, Google Sheets, dan Gmail. Keempat layanan tersebut dipilih karena menyediakan *Application Programming Interface* (API) yang memungkinkan integrasi dengan platform *workflow automation* n8n sehingga proses pengelolaan dokumen dan pertukaran data dapat dilakukan secara otomatis [62].

Google Drive digunakan sebagai media penyimpanan dokumen mahasiswa dan hasil penilaian, Google Docs dimanfaatkan untuk menghasilkan laporan penilaian secara otomatis, Google Sheets digunakan sebagai media penyimpanan data penilaian dan monitoring hasil evaluasi, sedangkan Gmail digunakan untuk mengirimkan laporan penilaian kepada dosen maupun mahasiswa melalui *workflow automation*. Integrasi seluruh layanan tersebut dilakukan menggunakan Google Workspace API sehingga setiap proses dapat dijalankan secara otomatis dalam satu alur kerja yang terintegrasi.

Pemanfaatan layanan Google Workspace pada penelitian ini mendukung proses penyimpanan, pengolahan, distribusi, serta pelaporan hasil penilaian secara otomatis. Integrasi dengan n8n membantu mengurangi pekerjaan yang bersifat repetitif, meningkatkan konsistensi proses, serta mempercepat pengelolaan dokumen akademik tanpa mengubah alur kerja yang telah digunakan oleh pengguna.

