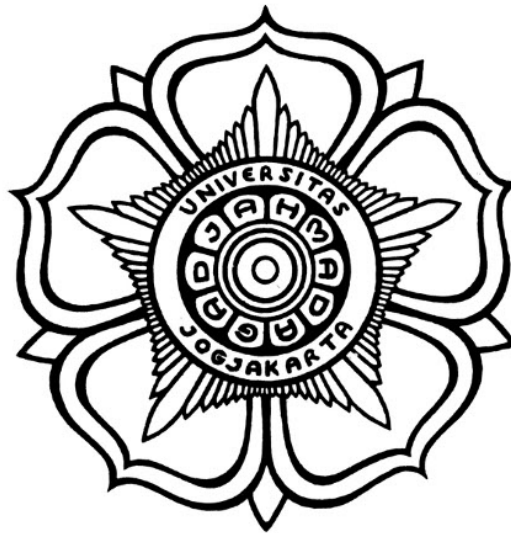


«JUDUL SKRIPSI»

SKRIPSI



THE SUSTAINABLE DEVELOPMENT GOALS
Industry, Innovation and Infrastructure
Affordable and Clean Energy
Climate Action

Disusun oleh:

«QORNAIN AJI»
«21/481767/TK/53170»

PROGRAM SARJANA PROGRAM STUDI «TEKNOLOGI INFORMASI»
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA
«TAHUN PENDADARAN»

HALAMAN PENGESAHAN

«JUDUL SKRIPSI»

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Departemen Teknik Elektro dan Teknologi Informasi
Fakultas Teknik
Universitas Gadjah Mada

Disusun oleh:

«QORNAIN AJI»
«21/481767/TK/53170»

Telah disetujui dan disahkan

Pada tanggal

Dosen Pembimbing I

Dosen Pembimbing II

Dosen Pembimbing 1, S.T., M.Eng., PhD.

«NIP xxxxxx»

Dosen Pembimbing 2, S.T., M.Eng., PhD.

«NIP xxxxxx»

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama :
NIM :
Tahun terdaftar :
Program : Sarjana
Program Studi :
Fakultas : Teknik Universitas Gadjah Mada

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

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

Yogyakarta, tanggal-bulan-tahun

Materai Rp10.000

(Tanda tangan)

Nama Mahasiswa
NIM

HALAMAN PERSEMBAHAN

Tuliskan kepada siapa skripsi ini dipersembahkan!

contoh

KATA PENGANTAR

Contoh: Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam hal penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Orang 1 yang telah
2. Orang 2 yang telah
3. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, aamiin.

Catatan: setiap nama yang dituliskan boleh disertai dengan alasan berterima kasih.

DAFTAR ISI

HALAMAN PENGESAHAN	ii
PERNYATAAN BEBAS PLAGIASI	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR SINGKATAN.....	xi
INTISARI.....	xiii
ABSTRACT	xiv
BAB I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian	3
1.4.1 Objek Penelitian.....	3
1.4.2 Metode Penelitian	4
1.4.3 Waktu dan Tempat Penelitian.....	4
1.4.4 Populasi dan Sampel	4
1.4.5 Variabel	4
1.4.6 Keterbatasan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan.....	5
BAB II Tinjauan Pustaka dan Dasar Teori	7
2.1 Tinjauan Pustaka	7
2.1.1 Peningkatan Umpan Balik LLM dengan <i>Direct Preference Optimization</i> (DPO).....	8
2.1.2 Peningkatan <i>Feedback</i> LLM dengan <i>Prompt Engineering</i> Berbasis <i>Learning Analytics</i>	10
2.1.3 <i>Reinforcement Learning from Human Feedback</i> (RLHF) untuk Peningkatan Kualitas Respons	11
2.1.4 Regulasi Diri dalam Pendidikan Pemrograman Berbasis LLM dengan <i>Learning Analytics</i>	12
2.1.5 Dashboard <i>Real-time</i> untuk Orkestrasi Kelas Berbasis Kanban	14
2.1.6 Perbandingan Metode Peningkatan Kualitas Respons LLM	16
2.1.7 Metode Evaluasi Umpan Balik LLM dalam Pendidikan	20

2.1.8	Evaluasi Ahli Manusia	22
2.2	Pertanyaan Penelitian	23
2.3	Dasar Teori	23
2.3.1	<i>Self Regulated Learning</i> (SRL) dan Peran Umpan Balik	24
2.3.2	<i>Large Language Model</i> (LLM)	25
2.3.3	<i>Prompt Engineering</i>	28
2.3.4	<i>Context Engineering</i>	28
2.3.5	Jenis Teknik <i>Prompting</i>	29
2.3.6	Perbedaan <i>Prompt Engineering</i> dan <i>Context Engineering</i>	29
2.3.7	<i>Learning Analytics</i> dan Cakupannya	30
BAB III Metode Penelitian		32
3.1	Alat dan Bahan Tugas Akhir	32
3.1.1	Alat Tugas Akhir	32
3.1.2	Bahan Tugas Akhir	33
3.2	Metode yang Digunakan	34
3.2.1	Evaluasi Kualitas Umpan Balik LLM	34
3.3	Alur Tugas Akhir	36
3.3.1	Tinjauan Pustaka	38
3.3.2	Pembuatan Dataset Papan Kanban Mahasiswa	38
3.3.2.1	Pembuatan <i>Prompt</i>	38
3.3.2.2	Pembuatan Data Sintetis	53
3.3.2.3	Pengambilan Data Umpan Balik Ahli Manusia	57
3.3.3	Pemilihan LLMs	57
3.3.4	Metode Peningkatan Hasil Umpan Balik	58
3.3.4.1	Rekayasa Konteks (<i>Context Engineering</i>)	58
3.3.4.2	Integrasi Metode <i>Learning Analytics</i>	65
3.3.4.3	Pembuatan Variasi <i>Prompt</i>	66
BAB IV Hasil dan Pembahasan		67
4.1	Pembahasan Tujuan 1 dengan Hasil Penelitian 1 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)	67
4.2	Pembahasan Tujuan 1 dengan Hasil Penelitian 2 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)	67
4.3	Pembahasan Tujuan 2 dengan Hasil Penelitian 3 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)	67
4.4	Perbandingan Hasil Penelitian dengan Hasil Terdahulu	67
BAB V Tambahan (Opsional)		68
BAB VI Kesimpulan dan Saran		69
6.1	Kesimpulan	69
6.2	Saran	69

DAFTAR PUSTAKA.....	70
LAMPIRAN	L-1
L.1 Isi Lampiran.....	L-1
L.2 Panduan Latex.....	L-2
L.2.1 Syntax Dasar	L-2
L.2.1.1 Penggunaan Sitasi	L-2
L.2.1.2 Penulisan Gambar	L-2
L.2.1.3 Penulisan Tabel	L-2
L.2.1.4 Penulisan formula.....	L-2
L.2.1.5 Contoh list.....	L-3
L.2.2 Blok Beda Halaman.....	L-3
L.2.2.1 Membuat algoritma terpisah	L-3
L.2.2.2 Membuat tabel terpisah.....	L-3
L.2.2.3 Menulis formula terpisah halaman.....	L-4
L.3 Format Penulisan Referensi	L-6
L.3.1 Book	L-6
L.3.2 Handbook.....	L-8
L.4 Contoh Source Code	L-10
L.4.1 Sample algorithm	L-10
L.4.2 Sample Python code	L-11
L.4.3 Sample Matlab code	L-12

DAFTAR TABEL

Tabel 2.1	Komparasi penelitian peningkatan kualitas umpan balik LLM di pendidikan	17
Tabel 2.2	Prompt manual to ensure the development of high-quality prompts. .	19
Tabel 2.3	Perbandingan <i>Prompt Engineering</i> dan <i>Context Engineering</i>	29
Tabel 3.1	Tabel Aturan Studi Kasus Perilaku Mahasiswa	45
Tabel 3.2	<i>Prompt</i> untuk Studi Kasus Data Sintetis Mahasiswa	46
Tabel 3.3	Ringkasan Topik OOP, Tujuan, dan Checklist	47
Tabel 3.4	<i>Prompt</i> untuk Batasan Sub-topik Materi Pembelajaran Mahasiswa ..	48
Tabel 3.5	<i>Role-Play Prompt</i> pada Pembuatan Data Sintetis	49
Tabel 3.6	<i>Few-Shot Prompt</i> pada Pembuatan Data Sintetis	50
Tabel 3.7	<i>Prompt</i> Struktur Papan Kanban	51
Tabel 3.8	<i>CoT Prompt</i> pada Pembuatan Data Sintetis	52
Tabel 3.9	Ringkasan Aktivitas Kanban per Mahasiswa	56
Tabel 3.10	<i>Prompt</i> Definisi Peran dan Batasan Agen Penilai	60
Tabel 3.11	<i>Prompt</i> Kriteria Mahasiswa	61
Tabel 3.12	<i>Prompt</i> Fase Pembelajaran Mahasiswa	62
Tabel 3.13	<i>Prompt</i> Daftar Materi Mahasiswa	62
Tabel 3.14	<i>Prompt</i> Penilaian Umpan Balik Pedagogis	64
Tabel 3.15	<i>Prompt</i> Format Input dan Output	65
Tabel L.1	Tabel ini	L-2
Tabel L.2	Contoh tabel panjang	L-4

DAFTAR GAMBAR

Gambar 3.1	Contoh gambar	35
Gambar 3.2	Contoh gambar	37
Gambar 3.3	Contoh gambar	39
Gambar 3.4	Contoh gambar	40
Gambar 3.5	Contoh gambar	42
Gambar 3.6	Contoh gambar	44
Gambar 3.7	Contoh gambar	53
Gambar L.1	Contoh gambar.	L-2

DAFTAR SINGKATAN

[SAMPLE]

b	=	bias
$K(x_i, x_j)$	=	fungsi kernel
y	=	kelas keluaran
C	=	parameter untuk mengendalikan besarnya pertukaran antara penalti variabel slack dengan ukuran margin
L_D	=	persamaan Lagrange dual
L_P	=	persamaan Lagrange primal
\mathbf{w}	=	vektor bobot
\mathbf{x}	=	vektor masukan
ANFIS	=	Adaptive Network Fuzzy Inference System
ANSI	=	American National Standards Institute
DAG	=	Directed Acyclic Graph
DDAG	=	Decision Directed Acyclic Graph
HIS	=	Hue Saturation Intensity
QP	=	Quadratic Programming
RBF	=	Radial Basis Function
RGB	=	Red Green Blue
SV	=	Support Vector
SVM	=	Support Vector Machines

INTISARI

Intisari ditulis menggunakan bahasa Indonesia dengan jarak antar baris 1 spasi dan maksimal 1 halaman. Intisari sekurang-kurangnya berisi tentang latar belakang dan tujuan penelitian, metodologi yang digunakan, hasil penelitian, kesimpulan dan implikasi, dan Kata kunci yang berhubungan dengan penelitian.

Kata Kunci ditulis maksimal 5 kata yang paling berhubungan dengan isi skripsi. Silakan mengacu pada ACM / IEEE *Computing classification* jika Anda adalah mahasiswa Sarjana TI <http://www.acm.org/about/class/> atau mengacu kepada IEEE keywords http://www.ieee.org/documents/taxonomy_v101.pdf jika Anda berasal dari Prodi Sarjana TE.

Kata kunci : Kata kunci 1, Kata kunci 2, Kata kunci 3, Kata kunci 4, Kata kunci 5

Contoh Abstrak Teknik Elektro:

"Penelitian ini bertujuan untuk mengembangkan sistem pengendalian suhu ruangan dengan menggunakan microcontroller. Metodologi yang digunakan adalah desain sirkuit, implementasi sistem pengendalian, dan pengujian performa. Hasil penelitian menunjukkan bahwa sistem pengendalian suhu ruangan yang dikembangkan mampu mengendalikan suhu ruangan dengan akurasi sebesar $\pm 0,5^{\circ}\text{C}$. Kesimpulan dari penelitian ini adalah sistem pengendalian suhu ruangan yang dikembangkan efektif dan efisien.

Kata kunci: microcontroller, sistem pengendalian suhu, akurasi."

Contoh Abstrak Teknik Biomedis:

"Penelitian ini bertujuan untuk mengevaluasi keefektifan prototipe alat pemantau denyut jantung berbasis elektrokardiogram (ECG) untuk pasien jantung. Metodologi yang digunakan meliputi desain dan pembuatan prototipe, pengujian dengan pasien, dan analisis data. Hasil penelitian menunjukkan bahwa prototipe alat pemantau denyut jantung berbasis ECG memiliki akurasi yang baik dan mampu memantau denyut jantung pasien secara efektif. Kesimpulan dari penelitian ini adalah prototipe alat pemantau denyut jantung berbasis ECG merupakan solusi yang efektif dan efisien untuk memantau pasien jantung.

Kata kunci: elektrokardiogram, alat pemantau denyut jantung, akurasi."

Contoh Abstrak Teknologi Informasi:

"Penelitian ini bertujuan untuk mengevaluasi keamanan dan privasi pengguna aplikasi media sosial terpopuler. Metodologi yang digunakan meliputi analisis kebijakan privasi dan pengaturan keamanan, pengujian penetrasi, dan survei pengguna. Hasil penelitian menunjukkan bahwa beberapa aplikasi media sosial memiliki kebijakan privasi yang kurang jelas dan rendahnya tingkat keamanan. Kesimpulan dari penelitian ini adalah pentingnya meningkatkan kebijakan privasi dan tingkat keamanan pada aplikasi media sosial untuk melindungi privasi dan data pengguna.

Kata kunci: media sosial, keamanan, privasi, pengguna."

ABSTRACT

Abstract ditulis italic (miring) menggunakan bahasa Inggris dengan jarak antar baris 1 spasi dan maksimal 1 halaman. Abstract adalah versi Bahasa Inggris dari intisari. Abstract dapat ditulis dalam beberapa paragraf. Baris pertama paragraph harus menjorok ke dalam sekitar 1 cm. Tidak disarankan menggunakan mesin penerjemah melainkan tulis ulang.

Keywords : Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia pendidikan tinggi saat ini menuntut mahasiswa untuk bisa belajar secara mandiri. Namun banyak dari mereka belum siap menghadapi tuntutan tersebut. Hal ini menimbulkan kesenjangan yang serius dalam keterampilan Self-Regulated Learning (SRL), yang dapat menghambat keberhasilan belajar dan mengganggu pemerataan akses pendidikan [1, 2]. Tantangan ini justru diperburuk oleh teknologi digital yang menjadi ciri khas pembelajaran masa kini. Meskipun menawarkan fleksibilitas, teknologi tersebut juga menghadirkan distraksi yang signifikan dan menuntut disiplin diri yang lebih besar [3]. Kegagalan institusi dalam memberikan dukungan yang memadai untuk pengembangan SRL bukan hanya masalah akademik, melainkan juga masalah sosial yang berdampak pada prospek karier jangka panjang mahasiswa dan efektivitas sistem pendidikan [1].

Untuk mengatasi kesenjangan SRL, Zimmerman mengemukakan kerangka kerja yang mencakup tahap perencanaan (*forethought*), pelaksanaan (*performance*), dan refleksi diri (*self-reflection*) di mana SRL merupakan gabungan strategi kognitif, motivasional, dan perilaku [1, 4]. Individu dengan kemampuan regulasi diri yang efektif terbentuk melalui penetapan tujuan, perancangan strategi, pemantauan diri secara aktif, dan evaluasi diri yang bijaksana. Sejumlah besar penelitian mengonfirmasi hubungan kuat antara keterampilan ini dan pencapaian akademik [1, 2]. Namun, pengembangannya kerap terhambat oleh hambatan personal, kontekstual, dan sosial—mulai dari manajemen waktu yang buruk dan distraksi digital hingga rendahnya kesadaran metakognitif—sehingga intervensi terstruktur diperlukan untuk membimbing mahasiswa menjalani proses yang kompleks ini [3, 5].

Metode tradisional yang paling efektif untuk mengembangkan keterampilan ini adalah umpan balik personal dari ahli manusia. Umpan balik berkualitas tinggi yang berfokus tidak hanya pada jumlah tugas yang diselesaikan, tetapi juga pada proses belajar dan regulasi diri mahasiswa merupakan katalis kuat untuk perbaikan [15, 16] [6, 7]. Namun standar emas ini menghadapi krisis skalabilitas dengan meningkatnya jumlah mahasiswa dan kebutuhan dukungan yang beragam membuat model bimbingan satu-satu sulit diterapkan dalam skala besar. Hal ini diperkuat dengan studi-studi yang dilakukan dan menunjukkan bahwa tekanan anggaran mendorong ukuran kelas yang semakin besar, sehingga interaksi personal antara dosen dan mahasiswa sangat terbatas [8, 9]. Ini menciptakan kebutuhan mendesak akan solusi inovatif berbasis teknologi yang dapat melengkapi keahlian manusia.

Salah satu teknologi tersebut adalah penggunaan Large Language Models (LLM). LLM dapat digunakan sebagai tutor virtual yang mampu memberi umpan balik yang cepat dan konsisten. LLM telah digunakan di berbagai penelitian dalam konteks pendidikan dengan bermacam terobosan. Temuan terbaru memperlihatkan bahwa ketika LLM digunakan untuk penilaian dan umpan balik, *artificial intelligence* (AI) cenderung murah hati dalam memberikan nilai sehingga nilainya lebih tinggi daripada nilai yang diberikan ahli manusia. Sementara itu penilaian antar rekan dan dosen lebih rendah dan sesuai dengan performa mahasiswa. Umpan balik AI sering terstruktur, tetapi tetap memerlukan pengawasan ahli manusia agar relevan secara pedagogis sehingga pendekatan hibrida (penggabungan AI dan ahli manusia) direkomendasikan [10]. Pendekatan hibrida direkomendasikan karena dapat menggabungkan kecepatan dan skalabilitas AI dengan penguasaan konteks, pemahaman empati dan nuansa pedagogis dari penilaian manusia [10]. Anjuran penulis terhadap pendekatan hibrida menegaskan bahwa respons LLM masih membutuhkan pengawasan pakar agar relevan. Di sisi lain, upaya standarisasi evaluasi tutor-AI menunjukkan bahwa meskipun model terkini seperti GPT-4 kuat dalam menjawab pertanyaan, mereka cenderung terlalu cepat memberikan jawaban kepada mahasiswa dan kurang menuntun proses. Fakta ini menunjukkan bahwa LLM belum ideal sebagai tutor dan belum dapat menggantikan ahli manusia dalam penilaian kualitas pedagogis [11].

Oleh karena itu, skripsi ini berfokus pada peningkatan kualitas performa respons LLM sebagai pemberi umpan balik agar mendekati mutu ahli manusia secara kuantitatif dan kualitatif. Peningkatan dilakukan melalui kerangka evaluasi yang menggabungkan acuan ahli manusia sebagai *ground truth* dan penilaian kualitas yang menilai dimensi-dimensi pedagogis relevan [10, 11]. Untuk mendukung proses pengukuran tersebut, skripsi ini memanfaatkan jejak proses belajar yang terstruktur. Data pembelajaran mahasiswa yang dipresentasikan ke dalam bentuk papan Kanban dipakai sebatas alat operasional untuk menata dan mengekstraksi data proses (perpindahan kartu, checklist, waktu pengerjaan) yang dapat dipetakan ke fase SRL (perencanaan–monitoring–refleksi) [12]. Penggunaan papan Kanban hanya sebagai salah satu instrumen pengaya data agar evaluasi dan perbaikan kualitas umpan balik LLM bisa berjalan lebih terarah. Fokus kebaruan terletak pada rangkaian evaluasi dan perbaikan *prompt* LLM yang menargetkan tiga fungsi pedagogis utama yakni feedback kognitif, dukungan motivasional, dan dukungan apresiatif dengan tujuan praktis yaitu menyamai kualitas ahli manusia pada metrik kuantitatif sekaligus memenuhi standar kualitas kualitatif [6, 7, 10, 11]. Dengan membandingkan keluaran LLM dan pakar ahli, studi ini menilai peningkatan performa LLM agar dapat menjadi pertimbangan dalam penggunaan sebagai agen pedagogis pelengkap mahasiswa yang masif dan personal.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, penelitian ini mengidentifikasi Secara spesifik, penelitian ini berfokus pada pengembangan umpan balik yang menargetkan tiga fungsi pedagogis: *feedback*, *motivation*, dan *appreciation*. Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana strategi optimal untuk meningkatkan performa LLM dalam menghasilkan umpan balik yang mencakup ketiga fungsi pedagogis tersebut?
2. Bagaimana kualitas umpan balik yang dihasilkan LLM terhadap proses SRL mahasiswa dibandingkan dengan umpan balik ahli manusia?
3. Bagaimana pengaruh besar ukuran model LLM terhadap kualitas umpan balik yang dihasilkan?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Merancang skema *prompt* serta memberikan metode *learning analytics* yang efektif untuk meningkatkan kualitas umpan balik LLM agar mendekati ahli manusia.
2. Mengevaluasi secara empiris apakah adanya peningkatan kualitas umpan balik LLM atas proses SRL mahasiswa baik secara kuantitatif maupun kualitatif dengan umpan balik ahli manusia.
3. Melihat pengaruh besar ukuran model LLM terhadap kualitas umpan balik yang dihasilkan.

1.4 Batasan Penelitian

Penelitian ini berfokus untuk melihat peningkatan performa dari LLM ketika diberikan perlakuan berupa metode *context engineering* yang memanfaatkan data *learning analytics*. Upaya untuk menyederhanakan proses ini dilakukan dengan cara memberikan batasan penelitian yang jelas dan ditetapkan oleh peneliti dengan mempertimbangkan aspek-aspek yang relevan. Berikut adalah batasan-batasan tersebut:

1.4.1 Objek Penelitian

- (a) Ruang lingkup mencakup analisis kemampuan model bahasa besar pada tugas berbahasa Indonesia dengan konteks pembelajaran berbasis papan Kanban.
- (b) Data pembelajaran yang digunakan berupa data sintetis yang disusun untuk memicu respons dari model dan dari pakar manusia. Pemakaian data sintetis dinilai memadai karena keterbatasan waktu pengambilan data ril serta karena fokus utama penelitian adalah perbandingan kualitas respons model dan respons pakar.

- (c) Model yang diteliti merupakan model yang tersedia secara *open source* dan diakses melalui Groq API. Dua model yang digunakan adalah Llama 3.1 8B Instant 128k dan Llama 3.3 70B Versatile 128k.

1.4.2 Metode Penelitian

Penelitian menggunakan rancangan eksperimen komparatif tanpa pelatihan ulang pada model. Seluruh prosedur mengandalkan rekayasa konteks yang mencakup perancangan templat *prompt* penetapan instruksi dan penyediaan konteks dari data Kanban berbahasa Indonesia. Langkah penelitian meliputi penyusunan skenario soal dan konteks berbasis Kanban pembuatan respons oleh kedua model melalui Groq API pengumpulan respons pakar manusia pada skenario yang sama serta evaluasi kuantitatif dan penilaian kualitatif oleh pakar. Reprodusibilitas dijaga melalui pencatatan parameter *decoding* penetapan *seed* serta pelaporan lingkungan komputasi.

1.4.3 Waktu dan Tempat Penelitian

Penelitian dilaksanakan pada Juni 2025 hingga Oktober 2025. Proses perancangan dan analisis dilakukan dengan memanfaatkan layanan Groq untuk inferensi model.

1.4.4 Populasi dan Sampel

Populasi penelitian mencakup seluruh respons atas skenario berbahasa Indonesia yang relevan dengan konteks pembelajaran berbasis Kanban. Sampel terdiri atas sekumpulan skenario tugas yang dipilih secara purposif beserta pasangan respons model dan respons pakar manusia. Jumlah *rater* pakar jumlah skenario serta pembagian data ditetapkan agar memadai untuk analisis kuantitatif dan penilaian kualitatif.

1.4.5 Variabel

- (a) **Variabel bebas** mencakup jenis model Llama 3.1 8B Instant 128k dan Llama 3.3 70B Versatile 128k serta strategi rekayasa konteks yang digunakan pada *prompt* dan penyajian *learning analytics* berbasis Kanban .
- (b) **Variabel terikat** adalah kualitas keluaran yang diukur dengan metrik kuantitatif seperti *accuracy precision recall macro F1* serta ukuran kesesuaian teks yang relevan.
- (c) **Variabel kendali** meliputi penggunaan bahasa Indonesia pada seluruh data karakteristik skenario Kanban, parameter seperti *temperature* dan *top-p random seed* serta prosedur evaluasi yang seragam.

1.4.6 Keterbatasan Penelitian

1. Data pembelajaran berbasis papan Kanban yang digunakan merupakan data sintetis yang disusun untuk keperluan pengambilan data dari pakar manusia dan model. Da-

ta sintetis dipilih karena keterbatasan waktu pengambilan data ril serta karena fokus penelitian berada pada perbandingan respons.

2. Penelitian hanya menggunakan model yang telah tersedia dan bersifat *open source* guna efisiensi waktu dan biaya.
3. Model diakses melalui Groq API sebagai sarana inferensi.
4. Model yang digunakan terbatas pada dua model yaitu Llama 3.1 8B Instant 128k dan Llama 3.3 70B Versatile 128k.
5. Tidak dilakukan pelatihan ulang pada model sehingga seluruh prosedur bersifat rekayasa konteks.
6. Seluruh kegiatan dilakukan pada Juni 2025 hingga Oktober 2025.
7. Analisis difokuskan pada kemampuan berbahasa Indonesia sehingga seluruh data yang digunakan berbahasa Indonesia.
8. Sumber data utama berasal dari keluaran model dan respons pakar manusia yang dianalisis sedangkan data pembelajaran Kanban berfungsi sebagai pendukung penyusunan konteks.
9. Luaran penelitian berfokus pada identifikasi metode optimisasi yang paling baik berdasarkan pengukuran kuantitatif yang selanjutnya diuji kepada pakar manusia untuk penilaian kualitatif.

1.5 Manfaat Penelitian

Penelitian ini menawarkan wawasan bagaimana penerapan LLM sebagai pendamping dalam proses SRL bagi mahasiswa bisa menjadi alternatif yang efektif dalam meningkatkan kualitas pembelajaran. Selain itu, penelitian ini juga dapat menjadi dasar untuk penelitian lebih lanjut untuk peningkatan performa respons LLM terutama pada bidang pedagogis sehingga seiring berjalannya waktu dapat memberikan hasil yang lebih optimal.

1.6 Sistematika Penulisan

Sistematika penulisan berisi pembahasan apa yang akan ditulis di setiap bab. Sistematika pada umumnya berupa paragraf yang setiap paragraf mencerminkan bahasan setiap Bab. Contoh:

Bab I membahas tentang pendahuluan yang berisi latar belakang, perumusan masalah dan tujuan penelitian.

Bab II berisi tentang metodologi penelitian yang terdiri dari desain penelitian, sumber data, Teknik pengumpulan data dan Teknik analisis data.

Dan seterusnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini menyajikan tinjauan komprehensif yang disusun secara sistematis untuk membangun landasan penelitian. Tinjauan dimulai dengan analisis terhadap penelitian-penelitian terdahulu, dilanjutkan dengan komparasi kritis, dan diakhiri dengan pembahasan dasar teori yang relevan.

Pertama, dibahas terlebih dahulu kumpulan penelitian sebelumnya yang berfokus pada upaya meningkatkan kualitas umpan balik dari Large Language Models (LLMs) dalam konteks pendidikan. Termasuk di dalamnya adalah berbagai strategi optimasi yang telah diusulkan serta metode-metode evaluasi yang digunakan untuk mengukur efektivitas umpan balik yang dihasilkan. Kedua, dilakukan komparasi mendalam terhadap penelitian-penelitian tersebut untuk mengidentifikasi kesenjangan yang ada di dalam literatur. Analisis komparatif ini membantu memetakan posisi penelitian ini dalam lanskap penelitian yang lebih luas. Ketiga, dibahas metode-metode evaluasi yang telah digunakan dalam studi-studi sebelumnya. Hal ini mencakup evaluasi baik dari segi kuantitatif maupun kualitatif. Keempat, berdasarkan identifikasi kesenjangan tersebut, disajikan dasar teori yang melandasi penelitian ini. Pembahasan mencakup peran LLM dalam dunia pendidikan, kerangka teoritis Self-Regulated Learning (SRL), konsep Prompt Engineering dan Context Engineering, pendekatan Learning Analytics beserta metode-metode analitik yang relevan untuk konteks pembelajaran, serta metode evaluasi yang dapat digunakan di dalam penelitian ini.

Melalui struktur ini, bab ini tidak hanya memberikan fondasi teoretis yang kuat, tetapi juga memperjelas kontribusi penelitian dalam mengisi celah yang teridentifikasi dari penelitian-penelitian sebelumnya.

2.1 Tinjauan Pustaka

Bagian ini menyajikan tinjauan terhadap penelitian-penelitian sebelumnya yang menjadi landasan bagi pengembangan sistem umpan balik LLM dalam pendidikan. Tinjauan ini mencakup dua fokus utama yaitu strategi peningkatan kualitas umpan balik LLM dan metode evaluasi yang digunakan.

Melalui analisis komparatif terhadap berbagai pendekatan yang ada, diidentifikasi pola, tren, dan kesenjangan dalam literatur. Pemetaan ini tidak hanya memberikan konteks bagi penelitian ini, tetapi juga memperjelas kontribusi unik yang ditawarkan dalam mengatasi keterbatasan-keterbatasan yang masih ada pada penelitian-penelitian sebelumnya.

2.1.1 Peningkatan Umpan Balik LLM dengan *Direct Preference Optimization* (DPO)

Para penulis yaitu, Juliette Woodrow, Sanmi Koyejo, dan Chris Piech (Stanford), mengajukan makalah “*Improving Generative AI Student Feedback: Direct Preference Optimization with Teachers in the Loop*” pada EDM 2025 [13]. Mereka memotret persoalan klasik pada umpan balik berbasis LLM yaitu, walau model mampu menghasilkan teks yang koheren, gaya, ketelitian, dan terminologi, sering kali tidak selaras dengan preferensi pengajar dan konteks spesifik mata kuliah. Penulis menyebutnya “*Feedback Alignment Challenge*” yaitu tantangan menyesuaikan keluaran AI dengan ketelitian, terminologi, dan gaya respons yang diharapkan tim pengajar, sehingga *feedback* terasa “*course-specific*”, bukan *feedback* generik.

Solusi yang ditawarkan adalah Direct Preference Optimization (DPO) dengan guru/TA di dalam loop. Inti dari DPO adalah selama proses penilaian, sistem menampilkan dua kandidat umpan balik AI untuk setiap jawaban mahasiswa, asisten dosen dapat memilih, mengedit, atau menulis ulang. Pilihan ini menjadi pasangan preferensi (menang-kalah) yang kemudian dipakai untuk *fine-tuning* model menggunakan DPO, tanpa perlu membangun reward model seperti pada *Reinforcement Learning from Human Feedback* (RLHF) sehingga alurnya lebih sederhana dan langsung menekan probabilitas keluaran yang disukai dibanding yang tidak disukai. Penulis merancang pipa tiga tahap—koleksi preferensi saat grading, pelatihan DPO antar tugas, dan inferensi untuk tugas berikutnya—mewujudkan sistem self-improving yang semakin selaras dengan preferensi pengajar dari satu *assignment* ke *assignment* berikutnya. Dalam implementasinya, mereka melakukan *fine-tuning* pada model Llama 3.1 8B Instruct menggunakan Hugging Face DPOTrainer. Proses pelatihan berlangsung sekitar tujuh jam pada tiga GPU A6000 dengan rata-rata 1.408 pasangan preferensi per gelombang. Pada tahap noticing, yaitu pengumpulan observasi atas pekerjaan mahasiswa, mereka memanfaatkan rubrik, tes otomatis, dan juga bantuan model LLM lain (GPT-4o). Hasil dari tahap ini kemudian disusun ke dalam prompt terstruktur agar model dapat menghasilkan respons yang lebih tepat sasaran.

Penulis menguji pendekatan ini pada dua skenario. Pertama, studi terkontrol buta yakni digunakan 10 evaluator ahli (staf pengajar) membandingkan umpan balik dari beberapa model pada kiriman mahasiswa yang dianonimkan di mana mereka tidak mengetahui identitas model dan menilai preferensi, *Insightfulness*, serta *Naturalness*. Kedua, deployment nyata di dua penyelenggaraan mata kuliah universitas besar (≈ 330 dan ≈ 289 mahasiswa; >10 *teaching assistants* (TA) per semester), di mana setiap generasi model dilatih dari preferensi tugas sebelumnya lalu dipakai pada tugas berikutnya. Penulis juga mencatat potensi *confounding variable* dalam deployment (mis. setiap soal dinilai oleh satu TA sehingga preferensi pribadi bisa memengaruhi pemilihan *feedback*; variasi soal

antar generasi menyulitkan isolasi efek model).

Untuk evaluasi, penulis memadukan penilaian manusia secara buta dan penilaian otomatis berbasis *critic model*. Pada sisi otomatis, mereka mengadopsi kerangka Scarlatos et al. [14] guna menilai lima dimensi yaitu, *Correctness*, *Not-revealing-the-answer*, *Suggestions*, *Positive tone*, serta skor total. Kerangka ini juga harusnya memuat “*diagnosing errors*” namun tidak difokuskan di hasil tabel karena tidak sejalan dengan tugas utama studi ini. Di luar itu, penulis memperkenalkan *critic model* baru dengan tiga dimensi yaitu *Assertiveness*, *Accuracy*, dan *Helpfulness* yang lebih umum untuk berbagai tipe jawaban (benar/salah). Kedua kerangka ini menggunakan GPT-4o sebagai evaluator otomatis; prompts dan kode penilaian dibuka untuk umum.

Hasil studi terkontrol menunjukkan bahwa model DPO lebih disukai dibanding baseline kuat GPT-4o dengan persentase 56,8% berbanding 40,2% sementara 3% responden memilih netral. Pada metrik *Insightfulness*, DPO juga memperoleh skor lebih tinggi secara signifikan dibanding GPT-4o (rata-rata 3,13 berbanding 2,79) maupun model SFT (2,82). Sebaliknya, pada metrik *Naturalness* tidak ditemukan perbedaan signifikan antar model. Ulasan kualitatif evaluator menegaskan bahwa alasan memilih DPO adalah karena tingkat kekhususan yang lebih tinggi, relevansi terhadap masalah, nada yang mendorong, serta saran yang lebih dapat ditindaklanjuti. Sementara itu, alasan utama dipilihnya GPT-4o adalah kejelasan dan keringkasan, sementara DPO terkadang dinilai terlalu panjang atau agak canggung dari segi gaya.

Pada evaluasi otomatis (kerangka Scarlatos), DPO melampaui GPT-4o pada 4/5 metrik *Correctness* ($0,932 \pm 0,004$ vs $0,862 \pm 0,006$), *Not-revealing-the-answer* ($0,980 \pm 0,002$ vs $0,973 \pm 0,003$), *Positive tone* ($0,989 \pm 0,002$ vs $0,905 \pm 0,005$), dan skor total ($0,724 \pm 0,004$ vs $0,664 \pm 0,005$). Satu-satunya area yang lebih baik pada GPT-4o adalah *Suggestions* (GPT-4o $0,229 \pm 0,007$ vs DPO $0,163 \pm 0,006$), menandakan bahwa keluaran GPT-4o relatif lebih banyak memberi saran eksplisit, sementara DPO unggul dalam ketepatan, menjaga agar tidak membocorkan jawaban, dan mempertahankan nada positif. Di *critic model* baru, sebaran titik menunjukkan klaster DPO lebih konsisten di area akurasi dan helpfulness yang tinggi, sedangkan GPT-4o menampilkan variasi lebih besar berikut klaster pada wilayah rendah akurasi dan rendah helpfulness; secara kuantitatif, umpan balik GPT-4o 8,73 kali lebih mungkin tergolong unhelpful dan 7 kali lebih mungkin inaccurate dibanding DPO (tingkat assertiveness relatif sebanding). Hasil deployment memperlihatkan kelayakan integrasi di kelas besar dan penguatan preferensi TA terhadap DPO dalam seting riil, seraya membuka jalan untuk monitoring otomatis dan eksplorasi awal fairness (gender) yang dapat diperluas ke demografi lain.

2.1.2 Peningkatan *Feedback* LLM dengan *Prompt Engineering* Berbasis *Learning Analytics*

Artikel ini ditulis oleh Lucas Jasper Jacobsen dan Kira Elena Weber (Universitt Hamburg) [15]. Tujuan utamanya ada dua: (1) mengidentifikasi jenis prompt seperti apa yang dibutuhkan agar LLM dapat menghasilkan umpan balik berkualitas tinggi di pendidikan guru, dan (2) membandingkan kualitas umpan balik LLM dengan umpan balik manusia dari kelompok novice (calon guru/preservice teachers) dan expert. Latar masalah yang mereka tekankan: studi empiris yang langsung membandingkan kualitas umpan balik LLM vs manusia masih jarang, dan belum ada manual/kerangka berbasis teori untuk menilai kualitas prompt di konteks pendidikan tinggi, padahal umpan balik berkualitas penting namun sumber daya manusia dan waktu dosen sering terbatas.

Sebagai respons, penulis mengembangkan sebuah manual prompt berbasis teori. Manual ini merangkum kategori-kategori kualitas prompt (mencakup aspek konteks/peran–audiens–medium, misi, kejernihan dan spesifikasi—misalnya format, kekonkretan, spesifik domain, dan logika instruksi). Berdasarkan manual tersebut, mereka merancang tiga versi prompt dengan kualitas rendah–menengah–tinggi. Seluruh generasi umpan balik LLM menggunakan ChatGPT-4, dengan alasan kesesuaian praktik nyata di pendidikan tinggi saat penelitian dilakukan. Gagasannya sederhana: jika mutu prompt dapat disistematiskan, kualitas umpan balik LLM akan lebih konsisten dan bisa dibandingkan secara adil dengan umpan balik manusia.

Tugas yang dipakai berupa learning goal di topik geometri (mengenal segitiga siku-siku dan teorema Pythagoras), lengkap dengan tiga jenis kesalahan yang umum pada perumusan tujuan pembelajaran. Dari tiga kualitas prompt (rendah–menengah–tinggi), penulis menghasilkan 20 umpan balik per prompt (total 60 keluaran LLM) yang kemudian dikodekan oleh tiga coder terlatih. Untuk perbandingan antar penyedia umpan balik, kelompok novice dan expert juga diminta menulis umpan balik dengan menggunakan prompt terbaik (kualitas tinggi), sehingga perbandingan dengan LLM berlangsung pada kondisi instruksi yang optimal. Desain ini memungkinkan isolasi efek “kualitas prompt” pada kinerja LLM sekaligus membangun baseline manusia yang kuat pada kondisi yang sama.

Kualitas umpan balik dianalisis dengan skema kodifikasi yang mencakup dimensi-dimensi pedagogis seperti concreteness (kriteria dan penjelasan), activation (mis. keberadaan pertanyaan pemandu), empathy (nada, penggunaan orang pertama), specificity, serta errors (kesalahan isi). Analisis statistik dilakukan untuk membandingkan LLM vs novice vs expert, dengan pengujian post hoc (Bonferroni) pada beberapa subkategori. Dengan pendekatan ini, penulis tidak hanya menilai “siapa yang lebih baik”, tetapi juga pada aspek apa LLM unggul/kurang, sehingga relevan bagi desain intervensi dan prompting ke depan.

Studi tersebut menunjukkan bahwa hanya prompt berkualitas tinggi yang secara konsisten memicu LLM menghasilkan umpan balik bermutu, sehingga desain prompt menjadi faktor kunci. Dibanding penilai pemula, LLM unggul pada hampir semua sub-kategori, sementara pada aspek afektif tertentu seperti valensi dan penggunaan orang pertama keunggulan itu tidak tampak. Dibanding pakar, LLM melampaui pada explanation, questions, dan specificity dengan selisih rata-rata (M_{diff}) masing-masing 0,46 (CI 95%: 0,17–0,74; $p < 0,001$), 0,50 (CI 95%: 0,07–0,93; $p < 0,05$), dan 0,96 (CI 95%: 0,52–1,41). Dari sisi efisiensi, ChatGPT-4 mampu menghasilkan sekitar 49 umpan balik dalam waktu yang sama ketika seorang pakar hanya menyusun satu, sehingga LLM berpotensi menjadi alternatif yang skalabel untuk umpan balik berkualitas asalkan perancangan prompt dilakukan dengan tepat.

Penulis menekankan bahwa kualitas prompt harus diajarkan sebagai keterampilan bagi pendidik; manual yang mereka kembangkan dapat dijadikan panduan praktis. Namun, mereka juga mengingatkan keterbatasan—misalnya, halusinasi masih mungkin terjadi pada prompt tertentu, dan persepsi siswa terhadap umpan balik LLM serta transfer ke konteks/mata kuliah lain perlu diteliti lebih lanjut. Dengan kata lain, LLM berpotensi menjadi alternatif berkualitas dan efisien untuk sebagian peran feedback pakar, tetapi desain prompt serta pengawasan pedagogis tetap penting agar umpan balik selaras dengan kebutuhan dan konteks kursus.

2.1.3 *Reinforcement Learning from Human Feedback (RLHF)* untuk Peningkatan Kualitas Respons

Makalah berjudul “*Training language models to follow instructions with human feedback*” ini ditulis oleh Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, dkk. [16], sebagai proyek tim Alignment di OpenAI. Mereka memulai dari pengamatan bahwa memperbesar model bahasa tidak otomatis membuatnya lebih mampu mengikuti niat pengguna; model besar tetap bisa menghasilkan keluaran yang tidak benar, toksik, atau tidak membantu sehingga tidak selaras (misaligned) dengan kebutuhan pengguna. Paper ini menawarkan arah untuk menyelaraskan model dengan niat pengguna pada beragam tugas melalui penyelarasan berbasis umpan balik manusia (fine-tuning with human feedback).

Masalah utamanya adalah ketidakselarasan tujuan pelatihan model bahasa (memprediksi token berikutnya dari web) dengan tujuan penggunaan di dunia nyata (mengikuti instruksi secara membantu, jujur, dan aman). Konsekuensinya, meski bisa diprompt untuk banyak tugas, model sering “membuat fakta”, bias/toksik, atau tidak mengikuti instruksi pengguna. Tim menekankan kebutuhan menyelaraskan model agar helpful-honest-harmless dalam penggunaan sehari-hari. Solusi mereka adalah pipeline RLHF (reinforcement learning from human feedback) dengan tiga tahap: (1) Supervised fine-

tuning (SFT) pada demonstrasi yang ditulis pelabel tentang perilaku model yang diinginkan; (2) pelatihan reward model (RM) dari perbandingan/ranking beberapa keluaran model oleh pelabel; (3) optimasi kebijakan dengan PPO menggunakan RM sebagai sinyal ganjaran. Dataset dikumpulkan dari prompt yang ditulis pelabel dan prompt nyata yang disubmit melalui OpenAI API, sehingga mencerminkan distribusi penggunaan riil. Hasilnya adalah keluarga model InstructGPT.

Secara operasional, RM akhir diinisialisasi dari GPT-3 6B yang telah di-*fine-tune* pada beragam dataset NLP publik; pelatihan RM dilakukan satu *epoch* pada himpunan latih, dan mereka menemukan pelatihan sensitif terhadap jumlah *epoch* (multi-*epoch* cepat *overfit*), sementara *learning rate* tidak terlalu sensitif. Untuk model RLHF, kebijakan diinisialisasi dari model SFT dengan campuran 10% data pra-latih (membantu stabilitas PPO), kemudian dilatih sebanyak 256,000 episode yang mencakup sekitar 31,000 *prompt* unik (setelah penyaringan PII dan deduplikasi). *Batch* per iterasi sebanyak 512 (dipecah menjadi 8 *minibatch* berukuran 64) dan hanya satu *inner epoch* per iterasi; *warmup* pada 10 iterasi pertama, ditambah KL reward mengikuti Stiennon dkk. ($\beta = 0,02$). Rincian ini memperlihatkan setup yang hati-hati agar PPO stabil pada skala 1,3B/6B/175B.

Evaluasi utama menggunakan preferensi manusia di atas distribusi prompt API (mencerminkan use-case riil), serta metrik safety seperti truthfulness dan toxicity (dengan regresi kinerja minimal pada dataset NLP publik). Mereka juga memformalkan tujuan evaluasi selaras dengan kerangka *helpful, honest, harmless*. Visualisasi utama (Figur 1) menampilkan seberapa sering keluaran model lebih disukai dibanding *baseline* GPT-3 175B (SFT).

Secara konsisten, InstructGPT menunjukkan perbaikan yang berarti: model 1.3B mereka lebih sering dipilih manusia dibanding GPT-3 175B, meski 100 kali lebih kecil; ada peningkatan truthfulness dan penurunan toksisitas, sambil menjaga regresi kinerja pada dataset publik tetap minimal. Ini memperlihatkan bahwa penyelarasan dengan umpan balik manusia lebih menentukan daripada sekadar memperbesar parameter. Penulis tetap mencatat keterbatasan model yang masih bisa melakukan kesalahan sederhana. Namun, arah ini dinilai menjanjikan untuk menyelaraskan model dengan niat pengguna di banyak aplikasi.

2.1.4 Regulasi Diri dalam Pendidikan Pemrograman Berbasis LLM dengan *Learning Analytics*

Makalah *Design of AI-Powered Tool for Self-Regulation Support in Programming Education* karya Huiyong Li dan Boxuan Ma dari Kyushu University [17], disajikan pada CHI 2025 Workshop: Augmented Educators and AI, berangkat dari tantangan integrasi AI di pendidikan pemrograman. Banyak alat berbasis LLM yang digunakan mahasiswa berjalan di luar LMS institusional seperti Moodle. Keterpisahan ini memutus keterkait-

an penting dengan konteks perkuliahan, misalnya materi kuliah, rincian soal, dan hasil eksekusi kode. Dampaknya, umpan balik AI sering kurang selaras dengan tujuan dan ruang lingkup mata kuliah, sementara dosen kesulitan menelusuri dampak jangka panjang interaksi mahasiswa dengan AI karena tidak ada jejak data proses yang terkumpul secara sistematis di lingkungan LMS. Pada saat yang sama, sebagian besar riset LLM di domain ini masih memfokuskan akuisisi pengetahuan (seperti generasi kode dan perbaikan bug) sehingga perhatian pada keterampilan SRL seperti perencanaan, monitoring, dan refleksi—relatif kurang. Kekhawatiran lain adalah ketergantungan mahasiswa pada jawaban instan LLM, yang berisiko mengurangi upaya kognitif mendalam yang dibutuhkan pemula untuk membangun kemandirian belajar.

Untuk menjawab celah tersebut, penulis merancang CodeRunner Agent, sebuah alat LLM yang menempel langsung pada ekosistem Moodle. Ia memadukan *lecture viewer* untuk penyajian materi, plugin CodeRunner guna menjalankan dan menilai kode secara otomatis, serta *xAPI logging* ke Learning Record Store sehingga setiap interaksi penting—mulai dari akses materi, percobaan *test case*, hingga permintaan bantuan yang terekam sebagai data proses. Di sisi pedagogis, dukungan SRL dioperasionalkan lewat model PPESS (*Planning, Program creation, Error correction, Self-monitoring, Self-reflection*) yang menurunkan kerangka SRL Zimmerman khusus untuk pembelajaran pemrograman. Dengan pendekatan ini, mahasiswa bisa meminta bantuan sesuai fase SRL yang sedang dijalani, sehingga umpan balik AI tidak acak, melainkan terarah pada kebutuhan regulasi diri saat itu.

Arsitektur CodeRunner Agent menonjol karena mengandalkan dua mesin konteks. Pertama, LACE (*Learning Analytics Context Engine*) yang merangkum metrik keterlibatan dan performansi dari jejak xAPI, seperti waktu yang dihabiskan, frekuensi upaya, tingkat keberhasilan, dan pola kesalahan, lalu menyuntikkan ringkasan tersebut ke *prompt* agar respons LLM memanfaatkan sinyal perilaku SRL nyata. Kedua, KCE (*Knowledge Context Engine*) yang mengelola basis pengetahuan materi kuliah dan latihan, termasuk konsep kunci, prasyarat, tingkat kesulitan, solusi, dan *typical mistakes*, sehingga umpan balik AI selaras kurikulum dan menghindari kebocoran solusi langsung. Dosen dapat mengunggah materi, memutakhirkan basis pengetahuan, dan menyetel parameter konteks agar gaya dan standar umpan balik “menjadi milik mata kuliah” alih-alih generik.

Skenario penggunaan disiapkan untuk kelas pemrograman pemula. Mahasiswa menulis kode di CodeRunner, menjalankan *test cases*, lalu mengakses panel LLM-based Support dan memilih fase SRL yang relevan, misalnya *error correction* ketika menghadapi pesan kesalahan. Permintaan itu seperti identitas latihan, waktu, dan tipe bantuan tercatat sebagai xAPI. Dengan demikian terbentuk alur *end-to-end* dari log pembelajaran, rekayasa konteks, dan respons LLM yang dapat dianalisis untuk penyelarasan pedagogis

dan perbaikan berkelanjutan. Pada tahap ini, makalah berposisi sebagai paper rancangan; penulis merencanakan *pilot* jangka pendek dan eksperimen satu semester lintas beberapa kelas untuk memvalidasi rancangan pada situasi nyata.

Rencana evaluasi memanfaatkan log xAPI untuk melihat perubahan pola percobaan, akurasi, waktu penyelesaian, serta pergeseran strategi SRL ketika dukungan AI diaktifkan. Di luar metrik kuantitatif, penulis juga merencanakan studi kelas guna menilai kualitas, relevansi, dan keamanan pedagogis dari umpan balik LLM, termasuk apakah agen berhasil menjaga SRL (memberi *scaffold* dan pertanyaan pemandu) alih-alih langsung membocorkan jawaban. Pendekatan evaluasi ganda ini menggabungkan analitik perilaku yang objektif dan penilaian pedagogis yang kualitatif ditujukan untuk memastikan alat tidak hanya berfungsi, tetapi bermanfaat bagi kemandirian belajar mahasiswa.

Kontribusi utama paper ini adalah rancangan terintegrasi yang menghubungkan data proses LMS dengan rekayasa konteks dan *scaffolding* SRL berbasis LLM di lingkungan yang benar-benar digunakan kampus. Pemetaan PPESS memberi kerangka operasional agar bantuan AI selalu terkait fase SRL yang tepat, sementara LACE dan KCE memastikan umpan balik kontekstual, selaras kurikulum, dan mendorong kemandirian. Hasil yang ditekankan bukan angka evaluasi, melainkan kelayakan implementasi dan janji untuk kelas besar yang ingin memanfaatkan analitik pembelajaran serta dukungan AI tanpa mengorbankan tujuan pengembangan SRL. Validasi empiris disiapkan melalui *pilot* dan studi semester penuh pada tahap berikutnya.

2.1.5 Dashboard *Real-time* untuk Orkestrasi Kelas Berbasis Kanban

Makalah ini ditulis oleh Sven Strickroth, Melanie Kreidenweis, dan Andreas Götzfried. Mereka menyoroti tantangan nyata pada pembelajaran berbasis tugas di kelas yang bersifat kolaboratif. Ketika beberapa kelompok bergerak pada tugas berbeda dalam waktu yang sama, guru sulit memiliki pandangan menyeluruh tentang status pekerjaan setiap kelompok. Versi awal alat mereka, AgileBoard4Teaching, memang memungkinkan perencanaan dan pelaksanaan dengan papan Kanban per kelompok, tetapi berjalan sepenuhnya *offline*. Guru perlu menyalurkan berkas ke semua siswa, berkeliling memeriksa satu per satu, serta tidak memiliki ringkasan progres kelas secara langsung. Di sisi lain, solusi profesional seperti Trello tidak ideal untuk sekolah karena fitur dan kebijakan privasi. Kebutuhan akan *dashboard real-time* menjadi titik mula rancangan mereka.

Solusi yang diusulkan adalah memperluas AgileBoard4Teaching menjadi sistem berjejaring dengan arsitektur *client-server*. Di sisi guru tersedia *authoring mode* untuk menyusun tugas dan putaran kerja lalu menggandakan papan untuk banyak kelompok. Di sisi siswa tersedia antarmuka Kanban tiga kolom *ToDo*, *In Progress*, *Done* dengan unggah hasil dan permintaan *review*. Inti pembeda adalah *dashboard* guru yang menampilkan kotak ringkas untuk tiap kelompok. Guru dapat melihat jumlah tugas per kolom,

daftar pekerjaan yang menunggu persetujuan, serta sinyal seperti *raise hand*. Tersedia pula pengiriman pesan massal, penetapan *timer*, dan jeda tampilan siswa. Secara teknis, sistem menggunakan Java Servlets dan MariaDB di *backend*, penyimpanan papan sebagai JSON, komunikasi WebSocket untuk sinkronisasi perubahan secara langsung, dan opsi *deployment* via Docker. Mekanisme bergabung untuk siswa menggunakan kode grup agar tetap ramah privasi tanpa akun individual.

Penulis melakukan dua studi evaluasi. Pertama, studi lapangan di kelas TIK tingkat 8 pada topik *CAD* selama tiga minggu dengan dua sesi per minggu. Desainnya *within-subjects*. Satu minggu pertama menggunakan versi *offline*, lalu beralih ke versi jaringan untuk membandingkan pengalaman secara adil. Total 18 siswa dalam 9 kelompok berpartisipasi. Pengumpulan data meliputi observasi langsung, kuesioner fitur, *System Usability Scale* (SUS), wawancara dengan guru, serta analisis *log* interaksi server. Kedua, studi simulasi berupa lokakarya dua jam yang diikuti 18 guru dan *trainer* berpengalaman. Pada lokakarya ini, prototipe dipakai untuk mengorkestrasi kegiatan sehingga peserta mengalami peran “siswa” sekaligus menilai kegunaan alat.

Temuan dari studi lapangan menunjukkan pergeseran preferensi yang kuat ke versi jaringan. Siswa menyatakan alur kerja lebih lancar karena tidak perlu impor dan simpan manual. Fitur seperti *raise hand*, pesan, *timer*, dan *pause* dinilai membantu koordinasi. SUS siswa mencapai 84 yang tergolong sangat baik dan SUS guru 92 yang termasuk kategori pengalaman terbaik. Analisis *log* memperlihatkan intensitas penggunaan tinggi di kelas serta pemanfaatan di luar kelas untuk persiapan dan *review*. Guru menyampaikan bahwa *dashboard* paling berguna untuk meninjau progres banyak kelompok sekaligus dan menyiapkan pertemuan berikutnya. Selama sesi tatap muka, guru tidak selalu menatap *dashboard* karena fokus pada pendampingan langsung. Beberapa saran muncul seperti tampilan yang lebih ringkas untuk kelas besar, pembuatan putaran baru secara cepat, serta statistik waktu dan progres.

Pada studi simulasi, para guru mengapresiasi sinkronisasi waktu nyata dan kemudahan kloning papan untuk banyak kelompok. SUS rata-rata 69 yang tetap masuk kategori baik. Skor ini dipengaruhi kejadian *bug* yang membuat sebagian peserta terkendala masuk ke putaran kedua. Meski demikian, mayoritas peserta menyatakan ingin mencoba alat ini di kelas mereka. Masukan tambahan menyangkut kebutuhan versi terkelola yang stabil dan peningkatan visibilitas opsi komunikasi.

Secara keseluruhan, kontribusi utama makalah adalah desain dan prototipe *dashboard real-time* yang mendukung orkestrasi kelas *agile* berbasis Kanban, lengkap dengan bukti kegunaan dari dua konteks evaluasi. Solusi ini menjembatani celah antara alat perencanaan yang kuat dan kebutuhan pemantauan langsung di kelas. Hasilnya menyiratkan kesiapan untuk dilanjutkan ke tahap produksi dan pengayaan fitur *learning analytics* seperti waktu penyelesaian dan deviasi dari target. Penulis menekankan pen-

tingnya menjaga keseimbangan antara informasi yang cukup dan beban kognitif guru agar *dashboard* tetap dapat dibaca sekejap sekaligus memberi sinyal tindakan yang jelas.

2.1.6 Perbandingan Metode Peningkatan Kualitas Respons LLM

Dapat dilihat, Tabel 2.1 merangkum karya kunci yang mewakili dua arus utama yakni penyesuaian *model* melalui *fine-tuning* misalnya *DPO* dan *RLHF* serta penyesuaian *konteks* melalui *prompt* dan *context engineering* serta orkestrasi kelas berbasis Kanban.

Tabel 2.1. Komparasi penelitian peningkatan kualitas umpan balik LLM di pendidikan

Studi	Tujuan	Pendekatan Inti	Pelaksanaan Penelitian	Temuan Utama
Woodrow et al. (EDM 2025) [13]	Menyelaraskan gaya dan isi feedback AI dengan preferensi pengajar	<i>Direct Preference Optimization</i> dengan guru/TA di dalam loop. Preferensi menang-kalah dari dua kandidat umpan balik dipakai untuk fine-tuning	Mata kuliah besar. Siklus antartugas. Observasi pekerjaan mahasiswa diringkas ke prompt terstruktur	Studi terkontrol menunjukkan DPO lebih disukai daripada GPT-4o. Skor insight lebih tinggi. <i>Naturalness</i> setara. Evaluasi otomatis juga lebih baik pada empat dari lima dimensi. DPO kadang lebih panjang
Jacobsen & Weber (2025) [15]	Menentukan desain prompt yang memicu feedback LLM berkualitas serta membandingkan LLM dengan manusia	Manual prompt berbasis teori. Tiga kualitas prompt rendah menengah tinggi. LLM dibanding novice dan expert pada tugas geometri	Feedback pada learning goal dan miskonsepsi umum. Semua pihak memakai prompt berkualitas tinggi untuk perbandingan adil	Prompt berkualitas tinggi menghasilkan feedback terbaik. LLM unggul novice di hampir semua subkategori. LLM melampaui expert pada explanation questions specificity. Efisiensi LLM jauh lebih tinggi
Ouyang et al. (InstructGPT, 2022) [16]	Menyelaraskan model agar mengikuti instruksi yang membantu jujur dan aman	<i>RLHF</i> tiga tahap. SFT pada demonstrasi manusia. Reward model dari perbandingan keluaran. PPO untuk mengoptimalkan kebijakan	Berbagai tugas umum. Prompt dari pelabel dan pengguna API. Bukan konteks kursus spesifik	Model kecil yang telah selaras lebih sering dipilih manusia dibanding GPT-3 yang jauh lebih besar. Truthfulness naik dan toksisitas turun. Kerangka selaras tugas umum bukan feedback proses belajar
Li & Ma (CHI 2025 WS) [17]	Merancang alat LLM yang mendukung SRL di programan serta tetap terhubung ke LMS	Rekayasa konteks ganda. LACE untuk ringkasan learning analytics berbasis xAPI. KCE untuk pengetahuan kurikulum. Pemetaan SRL ke PPESS	Moodle dengan CodeRunner. Log xAPI merekam akses materi upaya tes dan bantuan. Fase SRL dipilih pengguna untuk memicu bantuan yang tepat	Paper rancangan. Rencana uji lapangan dan studi satu semester. Menekankan kelayakan integrasi. Fokus pada dukungan SRL dan kontrol kebocoran jawaban. Belum ada angka hasil akhir
Strickroth et al. (2025) [18]	<i>Dashboard real-time</i> untuk orkestrasi kelas Kanban agar guru memantau progres banyak kelompok	Ekstensi AgileBoard4Teaching ke sistem <i>client-server</i> . <i>Dashboard</i> guru. Sinkronisasi <i>WebSocket</i> . Penyimpanan papan JSON. Opsi <i>deployment</i> Docker. Mekanisme kode grup ramah privasi	Kelas TIK tingkat 8 topik CAD. 18 siswa 9 kelompok. Desain <i>within-subjects offline</i> ke jaringan. Lokakarya dua jam dengan 18 guru/trainer. Data observasi kuesioner SUS wawancara dan log server	Preferensi kuat ke versi jaringan. Alur kerja lebih lancar. SUS siswa 84 dan SUS guru 92. Lokakarya SUS 69. Masukan lanjutan meliputi tampilan ringkas kelas besar pembuatan putaran cepat serta statistik waktu dan progres

Tabel 2.1 secara garis besar menjelaskan karya terdahulu yang berpecah ke dua arus utama. Arus pertama berfokus pada penyesuaian *model* melalui *fine-tuning*, misalnya *Direct Preference Optimization* (DPO) [13] dan *Reinforcement Learning from Human Feedback* (RLHF) [16]. Arus kedua menekankan penyesuaian *konteks* melalui perancangan *prompt* dan lingkungan interaksi beserta pemanfaatan data proses pembelajaran, sebagaimana tampak pada [15] dan [17]. Di luar itu, [18] menawarkan orkestrasi kelas berbasis papan Kanban dengan *dashboard real-time* yang kaya sinyal proses sehingga menyediakan landasan operasional untuk memantau dan melanjutkan progres kelompok secara langsung.

Dari sisi tujuan intervensi, DPO dan RLHF mengubah parameter model agar keluaran lebih selaras dengan preferensi manusia [13,16]. Sebaliknya, [15] memperlihatkan bahwa *prompt* berkualitas tinggi dapat mendorong kualitas umpan balik tanpa perlu pelatihan ulang, sementara [17] menekankan rekayasa konteks yang ditopang oleh *learning analytics*. Penelitian Woodrow juga tidak berfokus pada konteks SRL, melainkan pada penyesuaian gaya dan isi umpan balik sesuai preferensi pengajar. Walaupun metrik evaluasi otomatis yang digunakan Woodrow dkk. (kerangka Scarlatos [14]) mendukung adanya fungsi pedagogis yang mendukung SRL, hal tersebut bukan fokus utama penelitian mereka.

Menariknya, [18] secara tidak langsung menyediakan fitur yang mendukung guru untuk melakukan tugas *learning analytics*. Hal ini dapat dilihat dari kemampuan *dashboard* dalam menampilkan seperti jumlah tugas per kolom, daftar pekerjaan, dan waktu yang dihabiskan oleh siswa pada setiap tugas. Dengan demikian, guru dapat memantau aktivitas pembelajaran secara real-time dan mengambil tindakan yang diperlukan. Penelitian ini memosisikan diri pada jalur kedua, yaitu optimalisasi konteks, sejalan dengan batasan metodologis bahwa tidak dilakukan *fine-tuning*.

Penelitian [15] memperlihatkan bagaimana rekayasa *prompt* (*prompt engineering*) memengaruhi kualitas umpan balik dari *Large Language Models* (LLM). Metode yang digunakan adalah mengembangkan sebuah panduan perancangan *prompt* berbasis teori (*theory-driven prompt manual*) untuk menciptakan tiga jenis *prompt* dengan kualitas berbeda yaitu rendah, sedang, dan tinggi. Berikut adalah hasil dari yang dikerjakan oleh Jacobsen.

Tabel 2.2. Prompt manual to ensure the development of high-quality prompts.

Category	Subcategory	Tinggi (2)	Sedang (1)	Rendah (0)
Context	Role	Peran LLM dan penanya dijelaskan	Hanya satu peran yang dijelaskan	Tidak ada peran LLM maupun penanya yang dijelaskan
	Target audience	Ada target audiens yang jelas didefinisikan dan dijelaskan	Target audiens dijelaskan secara kasar	Target audiens tidak ditentukan
	Medium/channel	Media atau saluran tempat informasi disajikan dijelaskan dengan jelas	Media atau saluran dijelaskan secara kasar	Media atau saluran tidak disebutkan
Mission	Mission/question	Misi LLM dijelaskan dengan jelas	Misi LLM dijelaskan secara kasar	Misi LLM tidak jelas
Clarity dan specificity	Format and constraints	Properti stilistika serta spesifikasi panjang yang dideskripsikan	Hanya properti stilistika ATAU spesifikasi panjang yang diberikan	Tidak ada properti stilistika maupun spesifikasi panjang yang diberikan
	Conciseness	<i>Prompt</i> hanya berisi informasi yang langsung terkait dan relevan dengan output, serta jelas dan ringkas	<i>Prompt</i> ringkas dengan sedikit informasi berlebih	<i>Prompt</i> mengandung banyak informasi yang tidak relevan dengan misi/pertanyaan
	Domain specificity	Istilah teknis digunakan dengan benar dan memungkinkan LLM merujuknya di dalam jawaban	Istilah teknis digunakan sesekali atau tanpa penjelasan	Tidak menggunakan kosakata spesifik yang relevan dengan bidang subjek
	Logic	<i>Prompt</i> memiliki alur baca yang sangat baik, koherensi logis internal, urutan informasi yang koheren, dan hubungan yang jelas antara konten dan misi	<i>Prompt</i> hanya memenuhi beberapa kondisi ini	<i>Prompt</i> disusun secara tidak logis

Studi ini menemukan bahwa hanya *prompt* dengan kualitas terbaik yang secara konsisten menghasilkan umpan balik berkualitas tinggi, yang kualitasnya bahkan melampaui umpan balik dari para ahli (*experts*) dalam beberapa kategori seperti penjelasan, pertanyaan, dan spesifisitas. Pendekatan ini sangat berpotensi untuk diterapkan karena menawarkan efisiensi tinggi dengan menghasilkan umpan balik berkualitas yang jauh lebih cepat daripada ahli. Hal ini juga tidak memerlukan *fine-tuning* model, sehingga relatif murah dari segi sumber daya komputasi dan waktu.

Untuk perspektif sumber daya dan kelayakan, pendekatan DPO/RLHF menuntut data preferensi atau label serta infrastruktur pelatihan. Hal tersebut tidak selaras dengan rancangan studi ini yang memanfaatkan model *open source* melalui Groq API tanpa pe-

latihan ulang. Pendekatan *prompt/context engineering* lebih ringan dijalankan, mudah direplikasi, dan sesuai dengan keterbatasan komputasi maupun waktu. Penelitian [18] memanfaatkan *dashboard real-time* dan jejak proses Kanban yang memungkinkan pemantauan banyak kelompok sekaligus, pengumpulan sinyal proses yang autentik, dan pengambilan tindakan cepat oleh pengajar. Tidak seperti penelitian Li dan Ma [17] yang menggunakan Moodle sebagai *backbone* platform, platform *dashboard real-time* milik Strickroth dkk. dapat memberikan data yang lebih luas dan fleksibel untuk memantau keseluruhan proses pembelajaran kolaboratif berbasis Kanban. Penggunaan papan Kanban oleh pada penelitian [18] juga sesuai dengan keselarasan pada pembahasan latar belakang penelitian ini yaitu pemanfaatan jejak proses belajar mahasiswa yang direpresentasikan dalam bentuk papan kanban. Komponen-komponen ini belum tercakup pada studi yang berfokus pada *prompt* semata oleh Ouyang [16]. Hal ini membuat beberapa komponen yang ada di penelitian Strickroth dkk. dapat dimanfaatkan untuk memperkaya konteks umpan balik LLM.

Dari sisi kontrol pedagogis, tetap dipertahankan melalui penilaian kualitatif yang akan ditinjau langsung oleh pakar pendidikan. Keluaran LLM yang berupa *feedback* ditinjau dan dinilai melalui rubrik penilaian yang telah dipersiapkan. Dengan cara ini, kualitas pedagogis tetap terjaga dan sejalan dengan kebutuhan untuk dilakukan penilaian oleh pakar manusia.

Dengan demikian, penelitian ini mendapatkan celah perbaruan di mana pendekatan *prompting* dan *context engineering* yang murah sumber daya dan biaya digabungkan dengan *learning analytics* yang berasal dari papan Kanban untuk memperkaya konteks umpan balik LLM. Kombinasi tersebut belum pernah dibahas sebelumnya oleh literatur yang ada. Celah penelitian ini akan dievaluasi secara kuantitatif untuk mengukur kualitas keluaran LLM terhadap acuan pakar serta secara kualitatif melalui penilaian pakar pendidikan untuk menjaga kualitas pedagogis. Melalui batasan-batasan yang telah ditetapkan, rancangan ini menawarkan solusi praktis dan efektif. Pilihan metodologis ini konsisten dengan batasan penelitian, memanfaatkan data proses untuk memperkuat konteks *prompt*, menjaga peran guru sebagai pengarah kualitas, dan menghadirkan kesiapan implementasi di lapangan.

2.1.7 Metode Evaluasi Umpan Balik LLM dalam Pendidikan

Evaluasi kualitas teks yang dihasilkan oleh *Large Language Models* (LLM) merupakan tantangan kompleks yang memerlukan pendekatan multi-aspek. Secara historis, metrik evaluasi didominasi oleh metode berbasis tumpang tindih leksikal seperti BLEU dan ROUGE. Metrik ini bekerja dengan menghitung kesamaan *n-gram* (urutan kata) antara teks yang dihasilkan dan teks referensi (*ground truth*) [19]. Namun, ketergantungan pada pencocokan permukaan ini memiliki kelemahan signifikan; metrik ini gagal mema-

hami kesetaraan makna (semantik) dan sering kali memberikan skor rendah pada parafrasa yang valid, ketika dua kalimat memiliki arti yang sama tetapi menggunakan kosakata yang berbeda [20]. Berbagai studi telah menunjukkan bahwa korelasi metrik ini dengan penilaian manusia sering kali rendah, terutama dalam tugas-tugas yang menuntut keragaman ekspresi seperti simplifikasi teks atau dialog [21].

Keterbatasan ini mendorong pergeseran paradigma menuju metrik evaluasi yang mampu menangkap kesamaan semantik. Pada penelitian ini, evaluasi yang akan dilakukan dibatasi pada dua faktor yaitu kesamaan semantik dan kualitas relevansinya terhadap konteks. Untuk pendekatan kesamaan semantik, metrik berbasis *embedding* kontekstual seperti BERTScore dan BARTScore menjadi pilihan utama [22, 23]. Lalu, penilai (dikenal sebagai *LLM-as-a-Judge*) dapat memberikan penilaian relevansinya terhadap kriteria tertentu untuk teks yang dihasilkan. Metrik berbasis *embedding* seperti BERTScore dan BARTScore tidak lagi membandingkan kata secara literal, melainkan representasi vektornya yang kaya akan konteks, sehingga lebih selaras dengan penilaian kualitas oleh manusia [22]. Di sisi lain, pendekatan *LLM-as-a-Judge* memanfaatkan kemampuan pemahaman bahasa dari model canggih seperti GPT-4 untuk memberikan penilaian kualitatif yang bernuansa, meniru proses evaluasi oleh manusia pada skala yang lebih besar [24].

Dalam paper ini [22] Metrik ini bekerja dengan mengubah setiap token dalam kalimat kandidat dan referensi menjadi *embedding* kontekstual menggunakan model BERT. Selanjutnya, kesamaan makna diukur dengan menghitung kemiripan kosinus (*cosine similarity*) antara setiap pasang token, lalu secara serakah (*greedy matching*) mencocokkan token dari satu kalimat ke token yang paling mirip di kalimat lainnya untuk menghasilkan skor presisi, *recall*, dan F1. Keunggulan utamanya adalah kemampuannya untuk mengenali sinonim dan parafrasa, sehingga memberikan evaluasi yang lebih kuat terhadap kesetaraan makna dibandingkan metrik *n-gram*.

Sementara itu, BARTScore [23] menawarkan filosofi yang berbeda secara fundamental dengan membingkai evaluasi sebagai "tugas generasi teks". Alih-alih hanya membandingkan *embedding*, BARTScore menggunakan model seq2seq (sequence-to-sequence) untuk menghitung probabilitas sebuah teks dihasilkan dari teks lainnya. Misalnya, untuk mengukur presisi, ia akan menghitung seberapa mungkin model BART akan menghasilkan teks referensi jika diberi teks kandidat sebagai sumber. Skor yang lebih tinggi (*log-probability* yang lebih mendekati nol) menandakan kualitas yang lebih baik. Fleksibilitas ini memungkinkan BARTScore untuk menilai berbagai aspek seperti kefasihan, koherensi, dan bahkan faktualitas hanya dengan mengubah input sumber dan targetnya.

Untuk mengatasi subjektivitas dan bias dalam pendekatan *LLM-as-a-Judge*, kerangka kerja terstruktur seperti G-Eval telah dikembangkan [24]. G-Eval menggunakan LLM penilai (misalnya, GPT-4) tetapi memandunya dengan penalaran *Chain-of-Thought*

(CoT). Alih-alih hanya meminta skor akhir, G-Eval terlebih dahulu menginstruksikan LLM untuk menghasilkan langkah-langkah evaluasi berdasarkan kriteria yang diberikan pengguna, lalu mengikuti langkah-langkah tersebut untuk sampai pada sebuah skor. Pendekatan ini membuat proses penilaian lebih transparan, dapat dikontrol, dan terbukti mencapai korelasi yang lebih tinggi dengan penilaian manusia dibandingkan metode LLM-penilai yang tidak terstruktur. Pada penelitian [25], dilakukan juga penilaian umpan balik kode pemrograman yang dihasilkan oleh model open-source. Seperti G-Eval, penilai LLM diminta untuk menilai berdasarkan kriteria yang telah ditentukan. Sehingga dapat disimpulkan bahwa pendekatan *LLM-as-a-Judge* yang terstruktur dengan kriteria jelas meningkatkan keandalan penilaian.

2.1.8 Evaluasi Ahli Manusia

Untuk menilai kualitas umpan balik yang dihasilkan oleh LLM secara komprehensif, evaluasi manusia yang terstruktur menjadi standar emas, karena mampu menangkap nuansa pedagogis dan psikologis yang terlewat oleh metrik otomatis. Penilaian ini dapat dibingkai menggunakan rubrik yang didasarkan pada prinsip-prinsip teoretis yang telah mapan. Mengacu pada Shute (2008) [26], kriteria fundamental yang harus dinilai adalah apakah umpan balik LLM bersifat jelas, akurat, dan relevan dengan konteks tugas yang diberikan. Penilai manusia dapat secara langsung memverifikasi apakah instruksi yang diberikan spesifik dan apakah informasi faktualnya benar.

Selanjutnya, dengan mengadopsi model dari Hattie dan Timperley (2007) [27], penilai dapat mengevaluasi kelengkapan struktural umpan balik tersebut. Rubrik penilaian akan memeriksa apakah LLM mampu menjawab tiga pertanyaan kunci: (1) Feed Up: Apakah umpan balik mengklarifikasi tujuan akhir (Where am I going?); (2) Feedback: Apakah ia memberikan analisis tentang kinerja saat ini terhadap tujuan tersebut (How am I going?); dan (3) Feed Forward: Apakah ia menawarkan saran yang konkret dan dapat ditindaklanjuti untuk perbaikan (Where to next?).

Terakhir, dampak motivasional dari umpan balik, sebuah aspek yang sulit diukur secara otomatis, dapat dinilai melalui lensa Self-Determination Theory (Deci & Ryan, 1985) [28] dan prinsip penguatan positif (Fredrickson, 2001) [29]. Penilai manusia akan mengevaluasi apakah bahasa yang digunakan LLM mendukung otonomi (memberikan pilihan, bukan mendikte), menumbuhkan rasa kompetensi (mengakui usaha dan kemajuan), dan menggunakan nada yang apresiatif untuk meningkatkan self-efficacy. Dengan demikian, evaluasi manusia yang berlandaskan teori ini tidak hanya mengukur kebenaran teknis dari umpan balik, tetapi juga efektivitasnya dalam memotivasi dan memberdayakan pengguna secara konstruktif.

Secara keseluruhan, penilaian kualitas umpan balik LLM oleh ahli manusia menggabungkan evaluasi objektif berdasarkan rubrik yang dikategorikan menjadi tiga jenis

yaitu penilaian secara *feedback*, *motivation support*, dan *appreciation support*. Setiap kategori dievaluasi menggunakan skala likert (1-5) untuk mengukur berbagai dimensi kualitas umpan balik dari perspektif pedagogis dan psikologis. Dengan cara ini, evaluasi manusia menyediakan gambaran yang kaya dan bernuansa tentang efektivitas umpan balik LLM dalam konteks pendidikan.

2.2 Pertanyaan Penelitian

Berdasarkan tujuan penelitian yang telah diuraikan dan metode evaluasi yang dipilih, serta mempertimbangkan sifat penelitian yang cenderung **eksploratif**, maka fokus utama penelitian ini dirumuskan ke dalam beberapa **pertanyaan penelitian** (*Research Questions/RQ*) sebagai berikut:

1. **(RQ 1)** Sejauh mana efektivitas metode *context engineering* yang digabungkan dengan *learning analytics* (LA) dapat meningkatkan kualitas umpan balik LLM, jika dibandingkan dengan metode *Baseline*?
2. **(RQ 2)** Bagaimana persepsi dan penilaian ahli manusia terhadap kualitas, relevansi, dan kelayakan umpan balik yang dihasilkan oleh metode *context engineering* dengan LA?
3. **(RQ 3)** Sejauh mana ukuran model bahasa (misalnya, model kecil vs. model besar) memengaruhi kualitas umpan balik yang dihasilkan dalam konteks penelitian ini?

2.3 Dasar Teori

Bagian ini menyajikan landasan konseptual yang menopang rancangan dan metode penelitian. Pertama, diringkas prinsip *Self-Regulated Learning* (SRL)—termasuk peran umpan balik dalam fase perencanaan, pelaksanaan, dan refleksi—serta kaitannya dengan tiga fungsi pedagogis yang dinilai pada penelitian ini: *feedback* kognitif, dukungan motivasional, dan dukungan apresiatif. Kedua, diuraikan konsep *prompt engineering* sebagai praktik perumusan instruksi bagi LLM dan *context engineering* sebagai pengelolaan muatan informasi/konteks yang dipasok ke model; perbedaan keduanya serta implikasinya bagi desain tutor-AI juga dibahas untuk menegaskan peran masing-masing. Ketiga, dipaparkan *learning analytics* (LA) yang mencakup definisi, ruang lingkup, dan jenis data proses (mis. log LMS, urutan aktivitas, *time-on-task*, perpindahan tugas, beban/WIP) beserta cara ekstraksinya. Pada akhirnya, dijelaskan bagaimana sinyal-sinyal LA tersebut direkayasa menjadi konteks *prompt* agar LLM menghasilkan umpan balik yang lebih terstruktur, *actionable*, koheren, dan berempati, sekaligus menekankan kebutuhan evaluasi hibrida (acuan pakar + metrik kuantitatif–kualitatif) untuk memastikan kualitas respons LLM mendekati mutu pakar manusia.

2.3.1 *Self Regulated Learning* (SRL) dan Peran Umpan Balik

Self-Regulated Learning (SRL) merujuk pada kapasitas mahasiswa untuk secara aktif merencanakan, memantau, dan mengevaluasi proses belajarnya sendiri, melalui siklus perencanaan, pelaksanaan, dan refleksi yang saling terkait [1], [2]. Pada fase perencanaan, mahasiswa menetapkan tujuan serta strategi. Lalu, pada fase pelaksanaan, mereka menerapkan strategi kognitif dan melakukan pemantauan metakognitif (mis. manajemen waktu, monitoring pemahaman). Kemudian, pada fase refleksi, mahasiswa melibatkan penilaian diri terhadap kemajuan dan hasil untuk menyesuaikan strategi berikutnya [1], [2]. Berbekal mekanisme siklikal ini, SRL menggabungkan dimensi kognitif, motivasional, dan perilaku yang bersama-sama menentukan kualitas belajar mandiri di pendidikan tinggi [1], [2].

Di konteks pendidikan tinggi, SRL berperan krusial karena lingkungan belajar menuntut kemandirian, terutama pada pembelajaran digital (daring atau campuran) yang memerlukan disiplin diri lebih tinggi serta kemampuan mengelola distraksi. Kajian terkini menunjukkan bahwa SRL berkorelasi dengan kinerja akademik, keterlibatan, dan ketahanan studi. Sebaliknya, kekurangan SRL di pembelajaran daring atau campuran dapat memperlebar kesenjangan capaian yang diperoleh mahasiswa [3]. Dengan demikian, pengembangan kompetensi SRL termasuk penetapan tujuan, strategi, monitoring, dan refleksi—menjadi prasyarat penting untuk keberhasilan mahasiswa di ekosistem belajar modern [2], [3].

Kemajuan *Large Language Models* (LLM) membuka peluang penyediaan bimbingan pengarahan (*feedback*) yang personal, cepat, dan interaktif untuk menopang SRL. Studi eksperimental di konteks pemrograman (pendidikan tinggi) menunjukkan *feedback* berbasis LLM dapat meningkatkan strategi metakognitif, motivasi, dan performa mahasiswa dibanding kondisi tanpa *feedback* dari AI [6]. Riset pengembangan menunjukkan asisten belajar berbasis LLM yang dirancang dengan prinsip-prinsip pedagogis mampu menyajikan *feedback* yang scaffolded (bertahap), memicu *self-explanation*, dan menuntun refleksi [7]. Dengan dialog natural, LLM dapat membantu ketiga fase SRL yaitu, (i) perencanaan (menyusun tujuan dan rencana aksi), (ii) pelaksanaan (memberi hint/pertanyaan pemandu saat buntu), dan (iii) refleksi (merangkum kekuatan-kelemahan dan langkah tindak lanjut) [6], [7]. Agar efektif dan aman, *feedback* LLM sebaiknya diselaraskan dengan praktik pedagogis berbasis bukti (fokus pada proses, *actionability*, dan nada empatik) dan dievaluasi kualitasnya secara sistematis.

Dalam kerangka SRL, efektivitas umpan balik dipahami melalui fokus bertingkat pada *task*, *process*, dan *self-regulation*, serta dampaknya pada keyakinan diri dan ketekunan belajar [4]. Menurut artikel jurnal ini [4], disebutkan bahwa *feedback* kognitif efektif bila spesifik, mendiagnosis miskonsepsi, dan menyarankan langkah perbaikan yang terstruktur. Lalu, **dukungan motivasional** menjaga *self-efficacy* (keyakinan diri),

menetapkan tujuan yang menantang namun realistis, serta mendorong komitmen rencana aksi. Terakhir, pemberian apresiasi yang spesifik terhadap usaha/progres dapat berkontribusi pada afek positif dan keterlibatan yang disertai arahan perbaikan agar tidak menjadi pujian kosong [4]. Oleh karena itu, penelitian ini menilai kualitas respons LLM pada ketiga fungsi tersebut karena ketiganya (*feedback*, *motivation*, dan *appreciation*) secara langsung mendukung fase perencanaan–pelaksanaan–refleksi dalam SRL [1], [2], [4].

2.3.2 *Large Language Model (LLM)*

Kecerdasan Buatan adalah cabang ilmu komputer yang berfokus pada pembuatan sistem yang mampu meniru kecerdasan manusia dalam pemecahan masalah dan pengambilan keputusan. Pendekatan awal AI banyak bersifat *symbolic (rule-based)*, namun berkembang menuju metode *data-driven* seperti *machine learning*. Sejak awal, para peneliti AI telah membayangkan tercapainya *general problem solver* yang mampu menyelesaikan berbagai tugas secara cerdas [1]. Kemajuan mutakhir di bidang *natural language processing* menunjukkan tanda-tanda menuju visi tersebut, contohnya model bahasa besar (*large language model*) seperti *ChatGPT* yang dilatih dengan miliaran parameter pada seluruh teks internet mampu melakukan beragam tugas tekstual secara *out-of-the-box* dan menunjukkan kemampuan *emergent* mendekati kecerdasan umum manusia [1]. Perkembangan ini menandai lompatan signifikan dalam upaya mewujudkan AI yang lebih cerdas dan serbaguna.

Salah satu pendekatan utama untuk mencapai AI adalah pembelajaran mesin (*machine learning*), yakni algoritme yang memungkinkan komputer belajar dari data tanpa diprogram secara eksplisit. Alih-alih merancang aturan manual, pendekatan ini membuat sistem mengekstraksi pola dan pengetahuan langsung dari data. Khususnya, *deep learning* (pembelajaran mendalam) yang menggunakan jaringan saraf tiruan berlapis banyak telah merevolusi berbagai bidang, mulai dari pengenalan gambar hingga pemrosesan bahasa. Model *deep learning* mampu mempelajari representasi fitur tingkat tinggi secara hierarkis, sehingga melampaui pendekatan statistik konvensional. Dalam konteks pemodelan bahasa, pendekatan ini menggantikan model *n-gram* sederhana dengan model *neural probabilistik* yang memahami semantik lebih dalam [1]. Penerapan jaringan saraf berstruktur dalam (*recurrent neural networks* dan *transformer*) terbukti meningkatkan performa sistem NLP secara drastis, membuka jalan bagi terciptanya model bahasa yang semakin kompleks dan akurat.

Pemrosesan Bahasa Alami (*natural language processing* atau NLP) adalah cabang AI yang berfokus pada interaksi antara komputer dan bahasa manusia. Tugas-tugas NLP mencakup pemahaman teks, terjemahan otomatis, chatbot, ekstraksi informasi, dan lain-lain. Inti dari banyak sistem NLP adalah model bahasa (*language model*), yaitu model probabilistik yang mempelajari pola dan kecenderungan dalam sekumpulan teks

untuk memprediksi kata atau frasa berikutnya dalam sebuah kalimat. Generasi awal model bahasa menggunakan metode statistik (misalnya unigram dan n-gram) yang terbatas dalam menangkap konteks panjang. Perkembangan kemudian beralih ke model berbasis jaringan saraf seperti Recurrent Neural Networks (RNN) dan variannya LSTM/GRU, yang mampu mempertimbangkan urutan kata secara dinamis. Terobosan terbesar datang dengan diperkenalkannya arsitektur Transformer oleh Vaswani dkk. (2017), yang menggunakan mekanisme self-attention. Arsitektur transformer ini membawa perubahan mendasar dalam rancangan model bahasa dengan menangani dependensi jarak jauh secara lebih efisien [1]. Inovasi tersebut menjadi landasan bagi lahirnya generasi model bahasa mutakhir, termasuk keluarga GPT dari OpenAI dan model BERT dari Google, yang menunjukkan pemahaman konteks dan kemampuan generatif yang unggul.

Arsitektur Transformer memungkinkan pemrosesan urutan kata secara paralel, berbeda dengan RNN yang memproses secara sekuensial. Dalam encoder-decoder penuh, model dapat menangani tugas sequence-to-sequence (misal penerjemahan) dengan efisien: encoder menyandikan seluruh kalimat sumber menjadi representasi vektor, lalu decoder secara autoregresif menghasilkan keluaran (kalimat terjemahan) dengan memperhatikan konteks penyandian. Banyak Large Language Model (LLM) generatif (seperti GPT-3 dan GPT-4) menggunakan arsitektur transformer jenis decoder-only untuk menghasilkan teks, sedangkan model seperti BERT menggunakan encoder saja untuk tugas pemahaman bahasa. Fleksibilitas desain ini menjadikan Transformer dapat diadaptasi untuk berbagai skenario NLP, dan skalanya dapat diperbesar dengan menambah jumlah lapisan atau dimensi model untuk mencapai performa yang lebih tinggi [1].

Large Language Model (LLM) merujuk pada model bahasa berukuran sangat besar (berskala miliaran hingga triliunan parameter) yang dilatih pada korpus teks yang amat luas. LLM dirancang untuk memahami dan menghasilkan bahasa alami dengan tingkat kefasihan tinggi, bahkan mampu menangani berbagai tugas tanpa pelatihan khusus (zero-shot/few-shot learning). Contoh populer adalah GPT-3 dan GPT-4 dari OpenAI, yang masing-masing memiliki sekitar 175 miliar dan diperkirakan 800 miliar parameter [1]. Model sebesar ini dilatih menggunakan supercomputer selama ribuan jam pada himpunan data teks raksasa (misalnya seluruh konten internet) [2]. Dengan skala dan keragaman data tersebut, LLM menunjukkan kemampuan emergen, misalnya dapat menjawab pertanyaan, menulis esai, menerjemahkan, hingga memecahkan soal kode yang semuanya tanpa fine-tuning tambahan [2]. LLM generatif seperti ChatGPT bahkan mampu melakukan dialog interaktif dan memberikan penjelasan yang mendekati gaya humanis.

Peningkatan ukuran model secara konsisten terbukti berbanding lurus dengan peningkatan kinerja pada aneka tugas [1]. Studi menunjukkan tren bahwa semakin banyak parameter dalam model bahasa, akurasi dan kemampuan generalisasi model tersebut meningkat signifikan di berbagai domain [1]. Hal ini mendorong lahirnya model Xtreme-

LLM, istilah untuk LLM dengan lebih dari satu triliun parameter [1]. Namun, lonjakan skala ini juga membawa tantangan besar. Model dengan ratusan miliar parameter memerlukan sumber daya komputasi dan memori yang luar biasa, serta berdampak pada waktu pengembangan dan konsumsi energi yang tinggi [1]. Di samping itu, LLM kerap menghadapi isu hallucination (menghasilkan informasi salah yang terdengar meyakinkan) dan bias yang tertanam dari data latih [2]. Ketiadaan standar baku dalam pengembangan model-model masif ini menambah kompleksitas untuk menjamin keandalan dan keamanan penerapannya [2]. Beberapa kajian bahkan menggarisbawahi masalah etis seperti potensi penyalahgunaan, plagiat, misinformasi, serta pelanggaran privasi dan hak cipta jika LLM digunakan tanpa pengawasan [3]. Oleh karena itu, meskipun LLM menawarkan lompatan kinerja, diperlukan strategi mitigasi terhadap risiko teknis dan etis tersebut.

Perlu dicatat bahwa LLM dapat bersifat umum maupun spesifik domain. Wong et al. (2024) mengklasifikasikan model AI berbasis foundation model menjadi: (1) model khusus penyakit atau tugas tertentu, (2) model domain umum yang serbaguna di berbagai aplikasi medis, dan (3) model multimodal yang menangani data teks, citra, maupun sinyal sekaligus [1]. Ilustrasi ini menunjukkan bahwa framework LLM cukup fleksibel untuk diadaptasi; sebuah model besar bisa di-fine-tune untuk keperluan sangat khusus (misalnya diagnostik penyakit tertentu), atau dikembangkan sebagai model generalis yang mampu menjawab pertanyaan di banyak topik. Kemampuan integratif ini menjadikan LLM sebagai komponen inti dalam ecosystem AI modern di berbagai bidang.

LLM semakin dilirik untuk diimplementasikan dalam konteks pendidikan karena potensinya yang besar dalam meningkatkan proses belajar-mengajar. Integrasi model bahasa generatif seperti GPT-4 ke dalam pendidikan dipandang mampu mentransformasi pengalaman belajar siswa dan meningkatkan pengetahuan serta kompetensi mereka [1]. LLM dapat digunakan untuk mengembangkan kurikulum secara dinamis, menghasilkan materi pembelajaran yang dipersonalisasi sesuai kebutuhan masing-masing siswa, menyusun rencana studi individual, hingga memberikan umpan balik otomatis dalam penilaian tugas [1]. Dalam pendidikan kedokteran, misalnya, LLM berpeluang merevolusi metode pengajaran dan evaluasi dengan menyediakan skenario klinis interaktif, menjawab pertanyaan medis secara instan, dan membantu mahasiswa melatih keterampilan diagnosis dalam lingkungan simulasi [1]. Studi Znamenskiy et al. (2025) menunjukkan sebuah kerangka kerja di mana platform GenAI (seperti ChatGPT, Claude, dan lain-lain) diintegrasikan ke dalam praktikum lab untuk melatih berpikir kritis dan literasi digital mahasiswa. Mahasiswa didorong merumuskan pertanyaan (prompts) disiplin-spesifik dan kemudian mengevaluasi jawaban atau konten yang dihasilkan AI. Hasil uji coba pada kelas astronomi untuk non-sains menunjukkan tingkat keterlibatan dan refleksi kritis yang tinggi; banyak mahasiswa yang secara sukarela melanjutkan eksplorasi di luar jam

kelas dan bahkan mempresentasikan temuan mereka di simposium penelitian [2]. Temuan ini mengisyaratkan bahwa interaksi terstruktur dengan LLM, jika dipadu dengan metode asesmen reflektif, dapat meningkatkan motivasi dan hasil belajar mahasiswa [2]. Dengan kata lain, LLM berpotensi berperan sebagai tool pendamping yang memperkaya pembelajaran aktif dan investigatif di lingkungan pendidikan.

2.3.3 *Prompt Engineering*

Prompt engineering merupakan bidang kajian baru yang berfokus pada perancangan, penyempurnaan, dan implementasi prompt atau instruksi yang diberikan kepada model bahasa besar LLM untuk mengarahkan keluaran model tersebut [1]. Dengan kata lain, *prompt engineering* adalah “mekanisme pengendali” (*steering mechanism*) yang digunakan pengguna kecerdasan buatan generatif untuk menyusun *prompt* sedemikian rupa agar menghasilkan keluaran yang diinginkan [2]. Konsep ini menjadi semakin penting seiring munculnya LLM populer seperti ChatGPT, karena kemampuan merancang *prompt* yang efektif memungkinkan pengguna untuk mendapatkan jawaban yang lebih akurat, relevan, dan sesuai kebutuhan dari model AI [1].

Sebagai suatu keterampilan, *prompt engineering* dapat dipelajari dan dikembangkan. Para peneliti menuliskan bahwa *prompt engineering* sudah mulai diajarkan secara formal, misalnya di perguruan tinggi, karena dianggap sebagai *skillset* baru yang dibutuhkan di era integrasi AI [2]. Intinya, *prompt engineering* menekankan seni merumuskan pertanyaan atau perintah dalam bahasa natural sedemikian rupa sehingga model AI memahami konteks tugas dan memberikan respon optimal sesuai keinginan pengguna.

2.3.4 *Context Engineering*

Context engineering adalah disiplin yang berkembang untuk melengkapi dan melampaui *prompt engineering*. Jika *prompt engineering* berfokus pada apa yang dikatakan kepada model pada satu waktu, maka *context engineering* berfokus pada apa saja yang diketahui model saat *prompt* tersebut diberikan. Menurut survei terbaru, *context engineering* didefinisikan sebagai disiplin formal yang melampaui sekadar perancangan *prompt* dan mencakup optimisasi sistematis dari informasi konteks yang disuplai ke LLM [3]. Kinerja LLM sangat ditentukan oleh informasi kontekstual yang diberikan selama inferensi, sehingga *context engineering* berusaha memastikan model menerima *payload* informasi yang paling relevan dan berguna.

Dalam praktiknya, *context engineering* mencakup teknik-teknik seperti *retrieval-augmented generation* (RAG) untuk mengambil pengetahuan eksternal, penggunaan *memory systems* agar model mengingat interaksi sebelumnya, integrasi alat tambahan (*tool-integrated reasoning*), hingga koordinasi *multi-agent* AI [3]. Berbeda dengan *prompt engineering* yang hanya memanipulasi teks *prompt*-nya, *context engineering* mengatur

lingkungan dan konteks bagi model, misalnya mengumpulkan data pendukung, menyusun instruksi sistem, mendefinisikan aturan, serta memasukkan memori atau informasi historis sebelum *prompt* diberikan. Tujuannya adalah memaksimalkan kualitas jawaban AI dengan menghadirkan konteks informasi yang optimal dalam batas panjang konteks model tersebut [3]. Dengan demikian, *context engineering* menjadikan AI lebih andal dan *context-aware*, tidak semata-mata mengandalkan satu perintah teks statis.

2.3.5 Jenis Teknik Prompting

2.3.6 Perbedaan Prompt Engineering dan Context Engineering

Meskipun terkait, *prompt engineering* dan *context engineering* memiliki fokus dan pendekatan yang berbeda. Berikut adalah perbedaan utama keduanya [3]:

Tabel 2.3. Perbandingan *Prompt Engineering* dan *Context Engineering*

Aspek	<i>Prompt Engineering</i>	<i>Context Engineering</i>
Lingkup Input	Memandang <i>prompt</i> sebagai teks statis tunggal yang menjadi masukan model.	Memandang konteks secara dinamis dan terstruktur, terdiri dari berbagai komponen informasi (instruksi, pengetahuan, memori, status pengguna) yang dirangkai bersama.
Informasi dan Memori	Informasi bersifat tetap dalam teks <i>prompt</i> ; interaksi biasanya tanpa memori jangka panjang (setiap <i>prompt</i> berdiri sendiri).	Mempertimbangkan komponen memori atau <i>state</i> secara eksplisit, memungkinkan model memanfaatkan informasi dari interaksi sebelumnya (<i>stateful</i>).
Pendekatan Optimasi	Mengandalkan percobaan manual dan iteratif; sering dianggap sebagai suatu seni (<i>art</i>) coba-coba.	Menerapkan pendekatan yang sistematis dan ilmiah (<i>scientific</i>); melibatkan optimisasi di level sistem secara menyeluruh.
Skalabilitas dan Kompleksitas	Cenderung rapuh untuk <i>prompt</i> yang panjang dan kompleks; kurang terstruktur untuk skenario kompleks.	Dirancang untuk mengelola konteks yang besar dan beragam secara modular (segmentasi, hirarki, kompresi).

Secara ringkas, *prompt engineering* menitikberatkan cara merumuskan pertanyaan kepada AI, sedangkan *context engineering* menitikberatkan apa saja informasi pendukung yang disertakan agar AI dapat menjawab dengan lebih baik. *Context engineering* mengubah fokus dari sekadar “seni” merangkai *prompt* menjadi “ilmu” mengelola informasi untuk optimalisasi sistem AI [3]. Kedua pendekatan ini saling melengkapi: *prompt* yang baik diperlukan, namun penyajian konteks yang tepat akan semakin meningkatkan kemampuan model dalam menyelesaikan tugas secara andal.

2.3.7 *Learning Analytics* dan Cakupannya

Learning analytics adalah bidang kajian yang memanfaatkan data pendidikan untuk memahami dan meningkatkan proses pembelajaran. Salah satu definisi yang umum dikutip menyatakan bahwa learning analytics merupakan pengukuran, pengumpulan, analisis, dan pelaporan data tentang peserta didik serta konteksnya, dengan tujuan memahami dan mengoptimalkan pembelajaran serta lingkungan di mana pembelajaran tersebut berlangsung [4]. Dalam praktiknya, learning analytics mencakup penggunaan berbagai sumber data (misalnya data aktivitas pada Learning Management System, nilai akademik, log interaksi, dan sebagainya) untuk memperoleh wawasan yang dapat ditindaklanjuti oleh pendidik, lembaga, ataupun peserta didik sendiri.

Cakupan learning analytics sangat luas, meliputi beragam skenario dan kepentingan pendidikan. Pada tingkat institusi, learning analytics digunakan untuk memonitor kinerja dan retensi mahasiswa. Studi menunjukkan banyak institusi pendidikan menerapkan learning analytics terutama untuk mengidentifikasi mahasiswa berisiko tinggi (misalnya berpotensi drop-out atau gagal studi) dan mengambil langkah intervensi dini guna meningkatkan retensi [4]. Dengan analisis data yang tepat, institusi dapat bersikap lebih preventif daripada reaktif. Model prediktif dapat menandai siswa yang membutuhkan bantuan, sehingga advisor atau dosen dapat memberikan bimbingan sebelum masalah berkembang lebih jauh [4]. Hasilnya, learning analytics berkontribusi pada peningkatan keberhasilan studi melalui intervensi yang tepat sasaran.

Di sisi lain, cakupan learning analytics juga mencakup tingkat kelas dan individu. Bagi pendidik, analitik pembelajaran dapat dipakai untuk mengevaluasi efektivitas pengajaran dan kualitas interaksi di kelas. Misalnya, data pola akses e-learning dan keterlibatan siswa dapat dianalisis untuk mengetahui bagian materi mana yang sulit dipahami atau siapa siswa yang kurang aktif, sehingga pendidik dapat menyesuaikan strategi pengajaran. Bagi pelajar, dashboard learning analytics bisa memberikan umpan balik personal tentang progres belajar mereka, membantu self-reflection dan pengambilan keputusan belajar yang lebih baik.

Sebagai bidang multidisiplin, learning analytics memadukan teknik dari data science, machine learning, dan pendidikan. Penerapannya mencakup analisis deskriptif (melaporkan apa yang terjadi dalam pembelajaran), diagnostik (mengapa sesuatu terjadi), prediktif (memprediksi hasil atau risiko di masa depan, seperti kemungkinan drop-out), hingga preskriptif (memberikan rekomendasi intervensi untuk perbaikan). Dengan kata lain, cakupan learning analytics tidak hanya terbatas pada pelaporan statistik, tetapi juga pengambilan keputusan berbasis data dalam pendidikan.

Secara keseluruhan, learning analytics berperan sebagai alat bantu strategis bagi institusi pendidikan untuk meningkatkan kualitas pembelajaran. Dari meningkatkan

outcome belajar siswa, mengoptimalkan penggunaan teknologi pendidikan, menurunkan tingkat putus studi, hingga meningkatkan personalisasi pembelajaran, semuanya termasuk dalam ruang lingkup learning analytics [4]. Dengan dukungan literatur ilmiah yang terus berkembang, learning analytics menawarkan kerangka berbasis data untuk memahami proses belajar mengajar secara lebih mendalam dan mendorong inovasi dalam praktik pendidikan.

BAB III

METODE PENELITIAN

Bab ini menjelaskan metode atau cara yang digunakan dalam penelitian ini untuk mencapai maksud dan tujuan seperti yang tertulis dalam sub-bab 1.3 yaitu:

1. Merancang skema *prompt* serta memberikan metode *learning analytics* yang efektif untuk meningkatkan kualitas umpan balik LLM agar mendekati ahli manusia.
2. Mengevaluasi apakah adanya peningkatan kualitas umpan balik LLM atas proses SRL mahasiswa baik secara kuantitatif maupun kualitatif dengan umpan balik ahli manusia.
3. Melihat pengaruh besar ukuran LLM terhadap kualitas umpan balik yang dihasilkan.

3.1 Alat dan Bahan Tugas Akhir

3.1.1 Alat Tugas Akhir

Alat-alat yang digunakan pada tugas akhir ini berupa perangkat keras maupun perangkat lunak sebagai sarana pendukung. Alat-alat perangkat keras yang digunakan merupakan milik pribadi. Fungsi dari perangkat keras dan lunak ini adalah untuk penulisan kode, pengembangan chatbot, melakukan pengujian baik secara kuantitatif maupun kualitatif, serta untuk pengumpulan dataset. Berikut adalah daftar alat-alat tugas akhir yang digunakan pada penelitian ini.

1. **Macbook Pro 14 inci.** Spesifikasi utama:
 - Prosesor: M1 Pro 10 Core @ 3,2 GHz
 - GPU internal: M1 Pro GPU 16 Core
 - Neural Engine: 32 Core
 - RAM: *Unified Memory* 16 GB 200GB/s *memory bandwidth*
 - Penyimpanan: *Solid State Drive* internal 1 TB (MacOS 26 Tahoe)
2. **Visual Studio Code** versi 1.105.1 sebagai *integrated development environment* (IDE) untuk menulis dan mengelola kode.
3. **Python 3.12.7** sebagai bahasa pemrograman utama untuk mengembangkan *chatbot* berbasis LLM, dengan beberapa pustaka inti:
 - LangChain: kerangka kerja aplikasi LLM yang menyediakan komponen modular untuk orkestrasi *prompt*, alur data, dan integrasi sumber data eksternal.
 - LangGraph: ekstensi LangChain untuk membangun alur aplikasi yang lebih

kompleks dalam bentuk graf.

- **bert_score**: sebagai salah satu kerangka kerja evaluasi kuantitatif untuk mengukur kesamaan semantik (Precision, Recall, F1) antara respons yang dihasilkan model dengan data ground truth.
 - **BARTScorer** sebagai kerangka kerja evaluasi kuantitatif alternatif (menggunakan mBART) untuk menghitung skor F1, Precision, dan Recall berdasarkan log-probability token.
 - **Pandas** sebagai library yang digunakan untuk menyusun, mengagregasi, dan menampilkan tabel ringkasan hasil evaluasi kuantitatif (BARTScore dan GPT-4o Judge) agar mudah dibaca.
4. **Google Docs**: sebagai alat untuk menulis kuesioner, dan melakukan revisi serta dokumentasi hasil diskusi.
 5. **Google Sheets**: digunakan untuk memvisualisasikan dan mengelola data Kanban sehingga dapat dibagikan kepada para ahli yang akan mengisi dan menilai data, memudahkan kolaborasi, pengisian respons, serta pengumpulan metrik penilaian.
 6. **Anaconda** untuk manajemen pustaka Python dan *virtual environment* selama proses pengembangan.
 7. **ChatGPT 5 Thinking** untuk membuat data sintetis papan Kanban yang akan digunakan dalam penelitian ini.
 8. **Groq API** untuk melakukan inferensi LLM (pemanggilan model) yang digunakan dalam penelitian ini.

3.1.2 Bahan Tugas Akhir

Bahan yang digunakan dalam proses pembuatan chatbot dan untuk uji coba pada penelitian ini adalah sebagai berikut:

1. LLM dengan model **Llama 3.1 8B Instant**, dengan parameter delapan miliar, yang diperoleh dari platform Groq API yang berfungsi sebagai agen pemberi umpan balik (*Feedback Agent*).
2. LLM dengan model **Llama 3.3 70B Versatile**, dengan parameter lebih besar yaitu 70 miliar yang juga diperoleh dari platform Groq API yang berfungsi sebagai agen pemberi umpan balik (*Feedback Agent*).
3. **GPT-4o** sebagai model yang digunakan untuk evaluasi kuantitatif *LLM-as-a-judge*
4. **Dataset sintetis** dengan format data `.json` yang di-generate dari ChatGPT, berisikan pasangan data pembelajaran mahasiswa berbasis papan Kanban.
5. **Data umpan balik** berupa komentar dari ahli pembelajaran yang berisikan *fee-*

dback, motivation, dan appreciation.

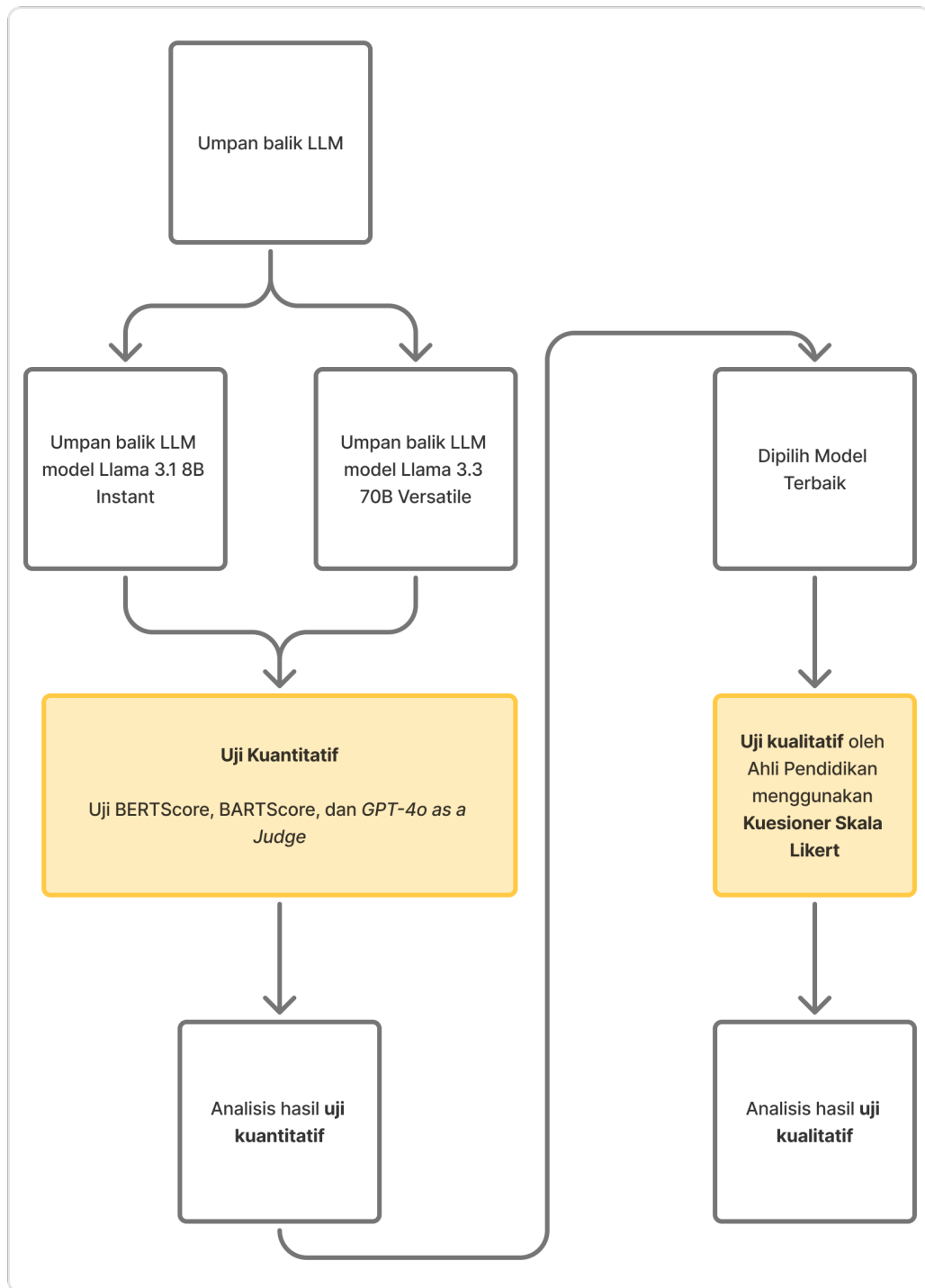
6. **Data hasil kuesioner** dari ahli psikologi pendidikan yang berisikan penilaian dan komentar terhadap hasil umpan balik LLM dalam menilai performa belajar mahasiswa.

3.2 Metode yang Digunakan

3.2.1 Evaluasi Kualitas Umpan Balik LLM

Penelitian ini menggunakan desain eksperimen untuk menilai kualitas *context engineering* yang digabungkan dengan *learning-analytics* dalam meningkatkan mutu umpan balik LLM berbahasa Indonesia pada skenario SRL berbasis papan Kanban. Setiap variasi *context engineering* diimplementasikan secara terpisah dan dipadukan dengan dua ukuran model (Llama 3.1 8B Instant dan Llama 3.3 70B Versatile melalui Groq API), menghasilkan beberapa kombinasi umpan balik yang diuji secara kuantitatif maupun kualitatif. Pengujian kuantitatif berfokus pada kesamaan semantik (Subbab [2.1.7]) menggunakan metrik berbasis *embedding* kontekstual, yaitu BERTScore [22] dan BARTScore [23], untuk mengukur keselarasan antara respons LLM dengan data umpan balik ahli manusia (*ground truth*) [32, 33]. Selain itu, evaluasi relevansi juga dilakukan menggunakan pendekatan *LLM-as-a-Judge* yang terstruktur [24, 25]. Di sisi lain, pengujian kualitatif—yang merupakan standar emas Subbab 2.1.8 yang melibatkan peninjauan langsung oleh pakar psikologi pendidikan [33]. Pakar ini menilai keluaran LLM menggunakan rubrik penilaian komprehensif yang didasarkan pada kerangka teoretis mapan seperti model Hattie dan Timperley (2007) [27], Shute (2008) [26], dan *Self-Determination Theory* [28]. Penilaian ini mengukur tiga dimensi utama yakni *feedback*, *motivation support*, dan *appreciation support* menggunakan skala Likert (1-5) untuk memastikan kualitas pedagogis dan psikologis dari umpan balik tersebut (Subbab[2.1.8]). Berikut adalah diagram alur metode evaluasinya.

Evaluasi Kualitas Umpan Balik LLM



Gambar 3.1. Diagram Alir Metode Evaluasi.

Terlihat dari Gambar 3.1, alur evaluasi penelitian ini dimulai dengan generasi umpan balik LLM yang dipadukan dengan berbagai skema *context engineering*. Umpan balik ini dihasilkan oleh dua model berbeda untuk perbandingan, yaitu Llama 3.1 8B *Instant* dan Llama 3.3 70B *Versatile*.

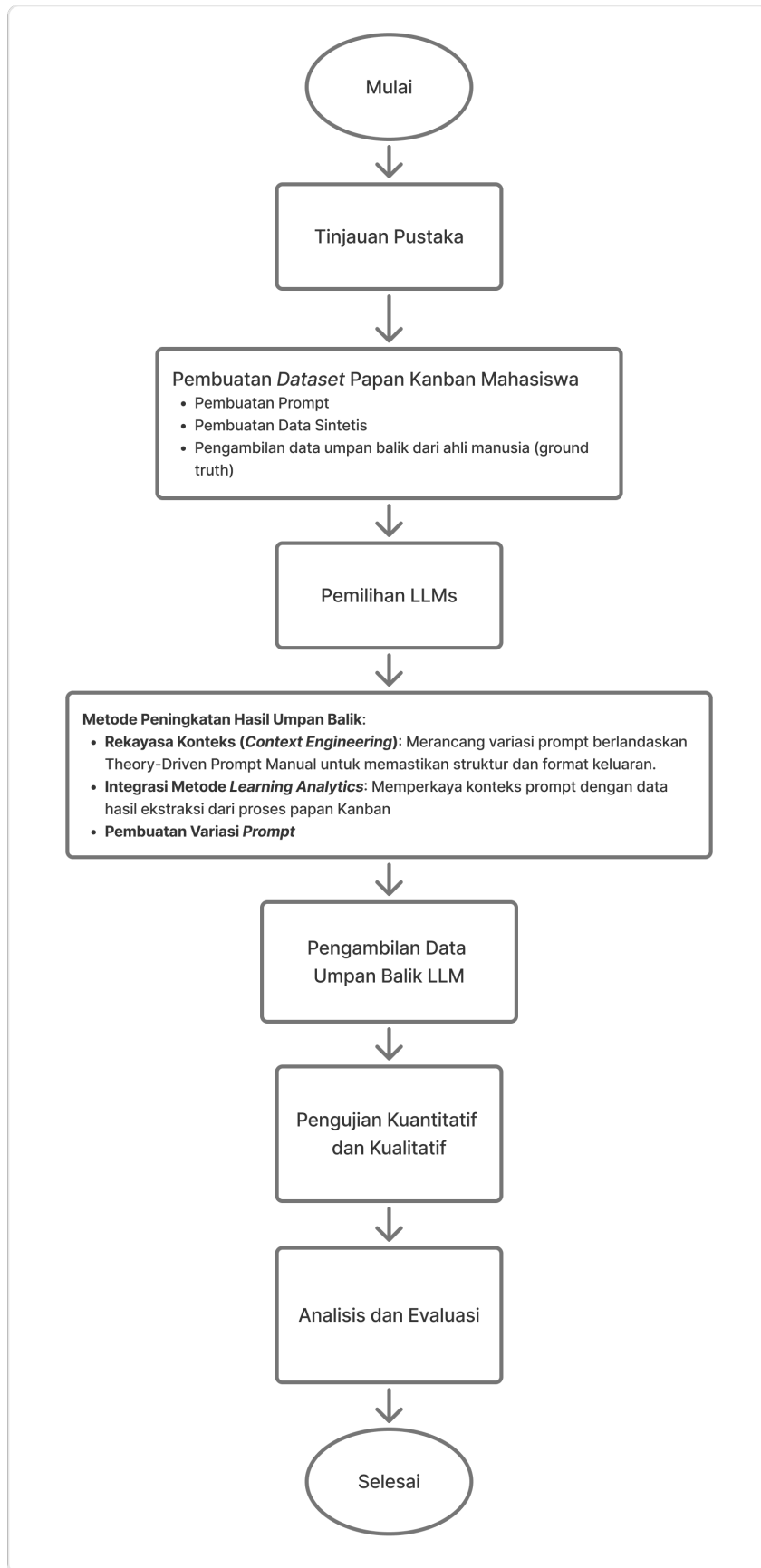
Luaran dari model-model ini pertama-tama melalui tahap Uji Kuantitatif. Sejalan dengan metodologi yang telah dijelaskan, tahap ini berfokus pada pengukuran kesamaan semantik dan relevansi Subbab 2.1.7 menggunakan tiga metrik utama: Uji BERTScore, BARTScore, dan GPT-4o as a Judge [22, 23, 24].

Luaran dari pengujian ini kemudian menjalani analisis hasil Uji Kuantitatif. Berdasarkan analisis ini, model terbaik dipilih—yaitu, kombinasi model dan teknik *context engineering* yang menghasilkan skor keselarasan tertinggi terhadap *ground truth* ahli.

Selanjutnya, umpan balik dari model terbaik tersebut dilanjutkan ke tahap evaluasi kedua, yaitu Uji Kualitatif. Tahap ini merupakan "standar emas" Subbab 2.1.8 yang melibatkan peninjauan langsung oleh Ahli Pendidikan. Pakar ini menilai kualitas pedagogis dan psikologis dari umpan balik model terbaik menggunakan Kuesioner Skala Likert yang dirancang khusus berdasarkan kerangka teoretis [26, 27, 28]. Langkah terakhir adalah Analisis hasil uji kualitatif untuk mendapatkan kesimpulan akhir mengenai efektivitas umpan balik LLM dalam konteks SRL.

3.3 Alur Tugas Akhir

Dalam penelitian ini, terdapat sebuah alur kerja yang menjabarkan semua tahapan dan proses yang terlibat. Secara keseluruhan, proses penelitian terdiri dari empat tahap utama: tahap studi pendahuluan dan persiapan data, yang mencakup Tinjauan Pustaka dan Pembuatan *Dataset* Papan Kanban Mahasiswa (termasuk pengumpulan *ground truth* ahli); tahap desain eksperimen, yang meliputi Pemilihan Model LLMs dan perancangan Metode Peningkatan Hasil Umpan Balik (menggunakan Rekayasa Konteks, Integrasi *Learning Analytics*, dan Pembuatan Variasi *Prompt*); tahap implementasi, di mana dilakukan Pengambilan Data Umpan Balik LLM menggunakan metode yang telah dirancang; serta tahap evaluasi akhir, yang terdiri dari Pengujian Kuantitatif dan Kualitatif, diikuti oleh Analisis dan Evaluasi. Alur kerja penelitian secara keseluruhan diilustrasikan pada Gambar 3.2.



Gambar 3.2. Flowchart Alur Penelitian Tugas Akhir.

3.3.1 Tinjauan Pustaka

Bab ini menyajikan tinjauan pustaka mendalam yang relevan dengan topik tesis. Tinjauan ini berfokus pada penelitian-penelitian sebelumnya mengenai peningkatan kualitas umpan balik LLM dalam konteks pendidikan, dengan menyelidiki dua arus utama yang teridentifikasi dari karya-karya kunci [Tabel 2.1]. Arus pertama adalah penyelarasan model (*model alignment*) melalui *fine-tuning*, seperti *Direct Preference Optimization* (DPO) [13] dan *Reinforcement Learning from Human Feedback* (RLHF) [16]. Arus kedua adalah penyelarasan konteks (*context alignment*), yang mencakup rekayasa *prompt* (*prompt engineering*) berbasis teori [15] dan rekayasa konteks yang digabungkan dengan *learning analytics* [17]. Tinjauan ini juga menganalisis pemanfaatan dasbor orkestrasi kelas berbasis Kanban [18] sebagai sumber data proses, bersama dengan metode evaluasi yang digunakan untuk menilai kinerja umpan balik, baik secara kuantitatif (misalnya, BERTScore [22], BARTScore [23], dan *LLM-as-a-Judge* [24]) maupun secara kualitatif oleh ahli manusia sebagai standar emas [26, 27, 28].

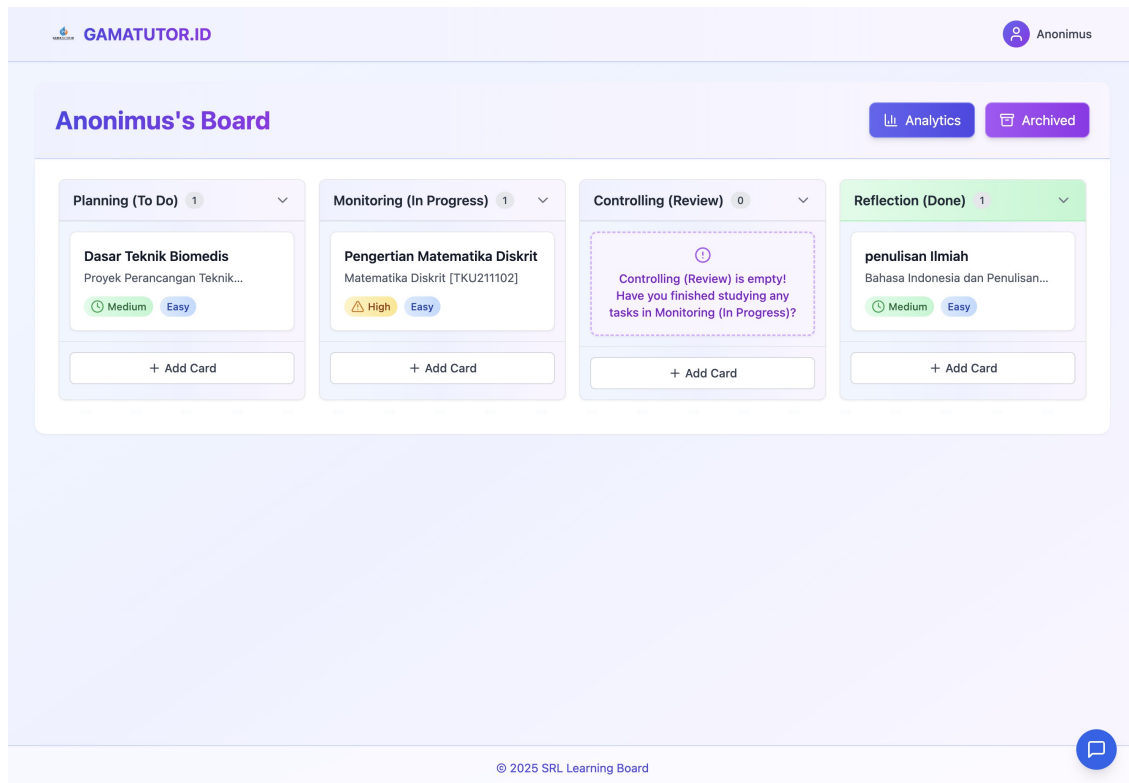
Selanjutnya, analisis terhadap metodologi yang diterapkan dan temuan utama dalam studi-studi ini dilakukan. Tujuan utama dari analisis ini adalah untuk menilai secara kritis keunggulan dan keterbatasan dari setiap pendekatan. Evaluasi ini mengidentifikasi keunggulan (seperti efisiensi biaya dan kelayakan implementasi dari rekayasa konteks [15]) dan kelemahan (seperti tingginya kebutuhan sumber daya komputasi dan data untuk *fine-tuning* [13, 16]). Temuan ini menjadi pertimbangan dalam pemilihan dan adaptasi metodologi yang paling tepat untuk menjawab masalah penelitian dalam karya ini: memanfaatkan pendekatan *context engineering* yang ringan dan murah [15] namun diperkaya dengan data *learning analytics* kaya sinyal yang diekstraksi dari papan Kanban [18] untuk mengatasi celah penelitian yang teridentifikasi.

3.3.2 Pembuatan Dataset Papan Kanban Mahasiswa

Pada bagian ini, dilakukan proses pembuatan *Dataset* Papan Kanban Mahasiswa. Tahap ini mencakup beberapa langkah krusial, yaitu Pembuatan *Prompt*, Pembuatan Data Sintetis yang akan menjadi data papan Kanban, serta Pengambilan data umpan balik pedagogis dari ahli manusia yang akan digunakan sebagai *ground truth* untuk perbandingan.

3.3.2.1 Pembuatan *Prompt*

Proses pembuatan *prompt* dimulai dengan analisis mendalam terhadap struktur *database* asli dari platform pembelajaran berbasis papan Kanban. Platform yang digunakan dan dikembangkan bernama Gamatutor yang dikembangkan oleh M. I. Azmi [30]. Berikut adalah tampilan dari platform pembelajaran tersebut yang ditampilkan pada Gambar 3.3.



Gambar 3.3. Tampilan Utama Platform Gamatutor.

Secara umum, Gamatutor adalah platform *learning analytics* berbasis metode *Kanban* yang dirancang untuk mendukung SRL mahasiswa. Platform ini memfasilitasi pengelolaan tugas belajar melalui sistem papan visual dengan empat tahapan utama yaitu *Planning (To Do)*, *Monitoring (In Progress)*, *Controlling (Review)*, dan *Reflection (Done)*. Setiap tugas direpresentasikan dalam bentuk *kartu* yang memuat informasi lengkap seperti judul, mata kuliah, *priority*, *difficulty*, *learning strategy*, nilai *pre-test/post-test*, dan *checklist* progres. Fitur utamanya meliputi pembuatan dan pengelolaan kartu tugas, pemantauan *progress* melalui *learning analytics*, serta manajemen profil pengguna.

Untuk satu kartunya merepresentasikan satu sub-topik dari salah satu mata kuliah. Jadi mahasiswa bisa membuat banyak kartu untuk mata kuliah yang sama. Untuk mengakses detail informasi-informasi tersebut, pengguna (mahasiswa) dapat melihatnya pada tiap-tiap kartu dengan menekan kartunya. Tampilan dari informasinya dapat dilihat pada Gambar 3.4

Task Details

Matematika Diskrit [TKU211102]

Pengertian Matematika Diskrit

Description

Add task details...

Start
Timer

Total study
time: 0m

Pre-test Grade ?

Enter pre-test grade
%

Post-test Grade ?

Enter post-test grade
%

Checklists
+ Add Checklist

No checklists yet. Add one to track your progress.

Links
+ Add Link

No links added yet. Add links to relevant resources.

Notes /
Summary

Only editable in Controlling (Review) or
Reflection (Done) column

Notes can only be edited in Reflection
(Done) column

0 characters

Task Properties

Priority

High

Difficulty

Easy

Learning Strategy

Rehearsal Strategies - Pengulangan
Materi

Learning Reflection

Rating will be available when task is completed

Close

Archive

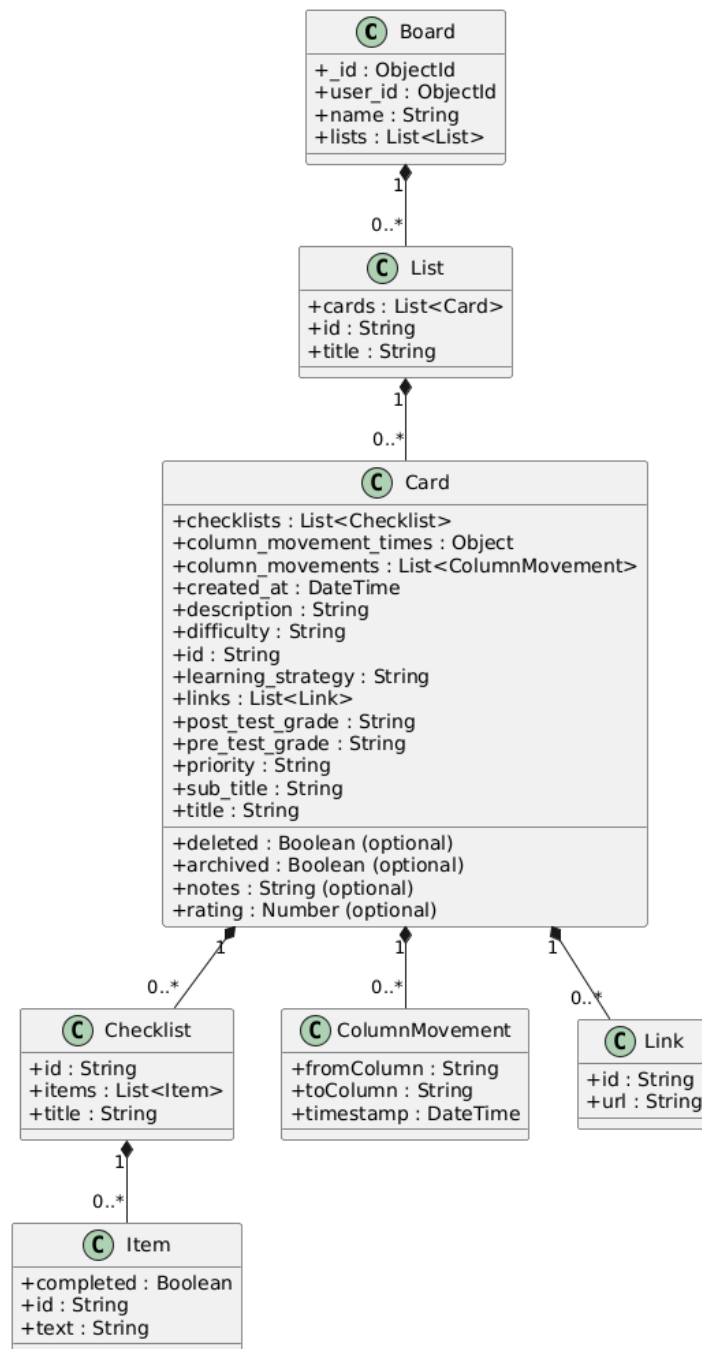
Delete

Gambar 3.4. Tampilan Informasi Tiap Kartu Gamatutor.

40

Semua detail informasi dan fitur-fitur tersebut dibuat untuk membantu mahasiswa dalam menerapkan SRL (seperti: *Start Timer*, *Checklists*, dan "Add Link") serta secara bersamaan memberikan data proses yang kaya untuk *learning analytics*. Data ini mencakup *Total study time* (yang dihasilkan dari fitur *Start Timer*), progres penyelesaian tugas melalui *Checklists*, hingga data metakognitif dan performa seperti *Priority*, *Difficulty*, *Learning Strategy* yang dipilih, serta nilai *Pre-test Grade* dan *Post-test Grade*. Kumpulan data proses inilah yang dapat menjadi insight bagi agen pedagogis (guru ataupun *AI chatbot*) untuk memantau, mendiagnosis hambatan, dan memberikan bimbingan yang terpersonalisasi sesuai dengan kondisi aktual mahasiswa tersebut. Gamatutor juga memiliki *Learning Assistant* dengan berbasis chatbot LLM yang rencananya akan diterapkan secara bertahap. Penelitian ini adalah bagian dari pengembangan *Learning Asistant* tersebut.

Semua informasi mengenai pembelajaran mahasiswa dan kartu-kartu Kanban-nya secara rapi disimpan ke dalam *database* NoSQL yaitu MongoDB. Data tersebut disimpan dengan mengikuti format JSON. Berikut adalah skema penyimpanan data kartu Kanban dalam MongoDB.



Gambar 3.5. Skema Penyimpanan Data Kartu Pembelajaran Kanban Mahasiswa.

Dapat dilihat pada Gambar 3.5 terdapat banyak atribut, khususnya pada entitas Card, yang menyimpan data-data kunci untuk merefleksikan proses dan hasil belajar mahasiswa. Satu Board, hanya bisa dimiliki oleh satu mahasiswa saja di mana Board sendiri adalah representasi dari keseluruhan papan pembelajaran berbasis Kanban.

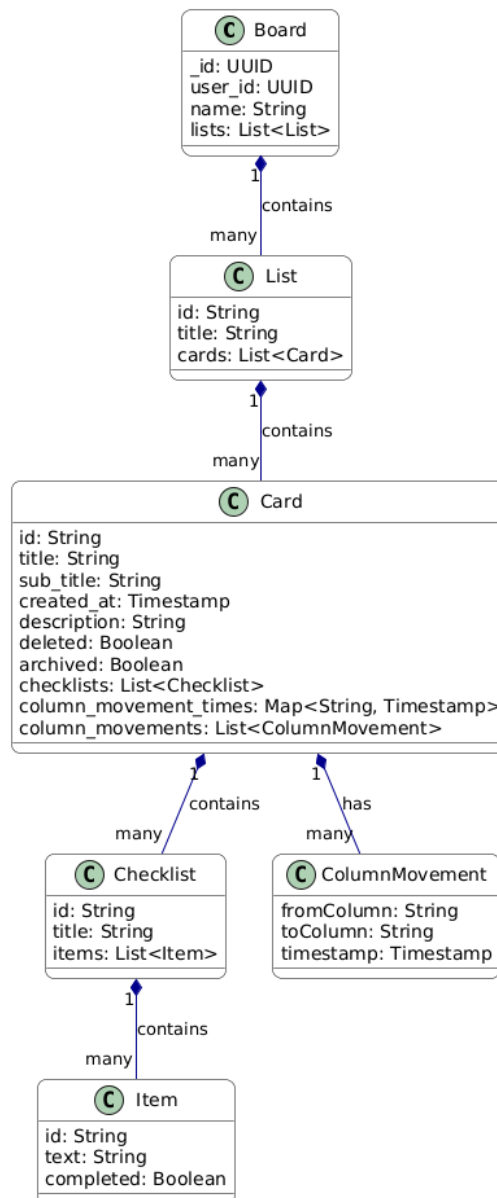
Beberapa atribut yang paling penting untuk mengevaluasi kemampuan belajar ini meliputi `pre_test_grade` dan `post_test_grade`. Kedua atribut ini secara langsung mengukur pengetahuan mahasiswa sebelum dan sesudah mengerjakan tugas, se-

hingga selisih antara keduanya dapat menjadi indikator utama peningkatan pemahaman.

Selain itu, atribut `difficulty` mengkategorikan tingkat kesulitan tugas, yang memungkinkan analisis performa mahasiswa terhadap tantangan yang diberikan. Atribut `column_movement_times` juga sangat kaya data dengan `timestamp` di dalamnya, data ini dapat digunakan untuk menghitung total durasi pengerjaan, waktu yang dihabiskan dalam fase belajar aktif (di kolom "In Progress"), atau bahkan mendeteksi adanya revisi. Terdapat juga atribut `notes` yang berisikan catatan dari mahasiswa ketika kartu sudah berada di tahap *Controlling (Review)*. Isi catatan ini bisa menjadi gambaran penting tentang rangkuman pembelajaran yang dilakukan mahasiswa terhadap sub-topik tersebut yang menjadikannya data yang cukup penting.

Atribut lain yang relevan adalah `checklists` untuk melihat progres granular pengerjaan sub-tugas, `learning_strategy` untuk memahami pendekatan belajar yang dipilih mahasiswa, dan `rating` untuk memberikan penilaian terhadap pengalaman belajar terhadap sub-topik yang dipelajari. Kombinasi dari atribut-atribut inilah yang memberikan gambaran komprehensif mengenai pola dan kemampuan belajar mahasiswa.

Dari skema MongoDB diatas, dilakukan reduksi atribut untuk memfokuskan pada data yang paling esensial bagi penelitian, seperti `created_at`, `checklists`, dan `column_movements`. Selain itu, dilakukan reduksi ini untuk mempermudah pembuatan data sintetis. Berikut adalah hasil skema database MongoDB setelah dilakukan reduksi data yang ada pada Gambar 3.6.



Gambar 3.6. Skema Penyimpanan Data Kartu Pembelajaran Kanban Mahasiswa Setelah Reduksi

Untuk mencapai tingkat representasi yang valid, data tersebut harus mampu menangkap berbagai pola perilaku dan tingkat keterlibatan mahasiswa yang berbeda-beda, mulai dari yang paling tidak aktif hingga yang sangat proaktif. Sebagai strategi utama untuk mengimplementasikan keragaman ini, sebuah studi kasus telah dirumuskan yang berfokus pada pengkategorian mahasiswa ke dalam profil-profil perilaku yang spesifik. Melalui studi kasus ini, telah diidentifikasi dan ditetapkan lima kategori profil yang berbeda, di mana setiap kategori mewakili satu titik pada spektrum tingkat kerajinan mahasiswa. Kelima kategori yang komprehensif ini yang diberi label "MALAS", "Sedikit Malas", "Cukup", "Rajin", dan "Sangat Rajin", akan digunakan sebagai dasar untuk menghasilkan set data yang bervariasi.

Untuk menyederhanakan penelitian ini, penggunaan asumsi akan digunakan dalam memberikan label kategori tingkat performa dan perilaku mahasiswa. Variabel atribut yang digunakan antara lain adalah jumlah total *card* yang dibuat oleh mahasiswa, tingkat penyelesaian tugas (*checklist*), aktivitas terkini dari kartu tersebut pada *column_movements*, dan letak keberadaan *card* tersebut. Hal ini diharapkan dapat menjadi proksi yang *memadai dan terukur* untuk membedakan secara jelas antara kelima profil perilaku dan performa tersebut, sehingga penyederhanaan ini tetap dapat menghasilkan data yang representatif.

Aturan-aturan tersebut dirumuskan dan diimplementasikan pada studi kasus seperti jumlah total kartu, persentase penyelesaian *checklist*, dan stempel waktu pergerakan kartu yang secara implementasinya dapat dilihat pada Tabel 3.1.

Tabel 3.1. Tabel Aturan Studi Kasus Perilaku Mahasiswa

Kategori	Studi Kasus	Jumlah Kartu	Penyelesaian Checklist (per kartu)	Aturan Aktivitas & Posisi Kartu
Malas	MHS_MALAS_SK1	1 - 2 kartu	0 - 2 item selesai (dari 3)	Tidak ada aktivitas (2 jam terakhir). Kartu cenderung di list1 atau list2.
Sedikit Malas	MHS_SEDMALAS_SK1	3 - 4 kartu	0 - 2 item selesai (dari 3)	Tidak ada aktivitas (2 jam terakhir).
	MHS_SEDMALAS_SK2	3 - 4 kartu	1 - 2 item selesai (dari 3)	Tidak ada aktivitas (2 jam terakhir).
Cukup	MHS_CUKUP_SK1	4 - 5 kartu	1 - 2 item selesai (dari 3)	Ada aktivitas (2 jam terakhir) untuk min. 3 kartu.
	MHS_CUKUP_SK2	3 - 4 kartu	Semua 3 item selesai	Ada aktivitas (2 jam terakhir) untuk min. 3 kartu.
Rajin	MHS_RAJIN_SK1	Tepat 5 kartu	2 - 3 item selesai (dari 3)	Hanya 1 kartu tersisa di list1 ("Planning (To Do)").
	MHS_RAJIN_SK2	4 - 5 kartu	Semua 3 item selesai	Hanya 1 kartu tersisa di list1 ("Planning (To Do)").
Sangat Rajin	MHS_SANGRAJIN_SK1	Tepat 5 kartu	Semua 3 item selesai	Semua 5 kartu berada di list4 ("Reflection (Done)").

Kategori-kategori ini kemudian dijadikan dasar untuk membuat *prompt* generator data sintesis pembelajaran mahasiswa berbasis Kanban. Kumpulan aturan yang telah dirangkum dalam Tabel 3.1 tersebut kemudian ditransformasi menjadi sebuah *prompt* instruksi yang terstruktur. Proses transformasi ini melibatkan penerjemahan setiap studi kasus—mulai dari MHS_MALAS_SK1 hingga MHS_SANGRAJIN_SK1, dari format

tabel tabular menjadi format teks naratif. Tujuan utamanya adalah untuk menyajikan aturan-aturan tersebut dalam format bahasa alami (*natural language*) yang dapat dipahami dan dieksekusi secara presisi oleh model LLM pembuat data tersebut. *Prompt* ini secara eksplisit merinci batasan untuk setiap variabel, termasuk jumlah kartu yang harus dibuat, syarat penyelesaian *checklist*, dan aturan mengenai aktivitas pergerakan kartu. Terdapat sedikit penambahan aturan pada pembuatan *prompt* yaitu adanya waktu kapan dibuatnya kartu sub-topik tersebut. Contoh *prompt* lengkap yang telah diformulasikan untuk agen generator tersebut dapat dilihat secara rinci pada potongan snippet pada Tabel 3.2.

Tabel 3.2. *Prompt* untuk Studi Kasus Data Sintetis Mahasiswa

```

1  """
2
3  Kategori: Malas
4
5  * Aturan untuk Studi Kasus 1 (MHS_MALAS_SK1):
6
7  * Buat 1 papan JSON dengan 1 atau 2 kartu.
8
9  * Setiap kartu memiliki 3 checklist, dengan total 0 hingga 2 item yang
    dicentang (completed: true) per kartu.
10
11 * Tidak ada kartu yang berpindah tempat dalam kurun waktu 2 jam
    terakhir (semua timestamp created_at atau column_movements terakhir
    harus <= 2025-06-07T15:00:00Z). Kartu cenderung berada di list1
    atau list2.
12
13
14
15 Kategori: Sedikit Malas
16
17 * Aturan untuk Studi Kasus 1 (MHS_SEDMALAS_SK1):
18
19 * Buat 1 papan JSON dengan 3 atau 4 kartu.
20
21 * Setiap kartu memiliki 3 checklist, dengan total 0 hingga 2 item yang
    dicentang per kartu.
22
23 * Tidak ada kartu yang berpindah tempat dalam kurun waktu 2 jam
    terakhir.
24
25 * Aturan untuk Studi Kasus 2 (MHS_SEDMALAS_SK2):
26
27 * Buat 1 papan JSON dengan 3 atau 4 kartu.
28
29 * Setiap kartu memiliki 3 checklist, dengan total 1 hingga 2 item yang
    dicentang per kartu.
30
31 * Tidak ada kartu yang berpindah tempat dalam kurun waktu 2 jam
    terakhir.
32
33
34 # .... Diatas adalah potongan yang tidak lengkap dari kode yang utuh.
    ....
35
36 """

```

Selanjutnya, untuk memperkecil topik pembelajaran yang akan dibuat pada data sintetisnya, dibatasi satu mata kuliah saja dengan 5 sub-topik yang dapat diambil. Pembuatan batasan sub-topik harus memiliki aturan yang perlu diberikan ke model LLM pembuat data sintetis tersebut. Hal ini agar model dapat memberikan data luaran yang baik serta variasi yang beragam dengan batasan yang jelas. Tabel 3.3 menunjukkan batasan sub-topik yang akan diberikan pada data sintetis.

Tabel 3.3. Ringkasan Topik OOP, Tujuan, dan Checklist

Topik	Tujuan	Checklist Items
Topik 1: Konsep Dasar dan Pilar OOP	Memahami fondasi dari OOP dan 4 pilar utamanya.	<ul style="list-style-type: none"> • Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. • Mengidentifikasi serta mendeskripsikan 4 pilar OOP: enkapsulasi, abstraksi, pewarisan, polimorfisme. • Menerapkan konsep dasar objek dan kelas dalam program sederhana.
Topik 2: Perancangan Kelas dan Interaksi Objek	Mampu merancang struktur kelas dan interaksinya secara logis.	<ul style="list-style-type: none"> • Mendesain <i>class diagram</i> berdasarkan kebutuhan dari studi kasus. • Menentukan relasi antar objek: asosiasi, agregasi, komposisi. • Mengimplementasikan rancangan interaksi objek dalam bentuk kode.
Topik 3: Enkapsulasi, Inheritance, dan Polimorfisme	Menguasai teknik pewarisan dan fleksibilitas perilaku objek.	<ul style="list-style-type: none"> • Menerapkan enkapsulasi menggunakan modifier akses (<i>private</i>, <i>public</i>, <i>protected</i>). • Membuat kelas turunan dan meng-<i>override</i> metode dari <i>superclass</i>. • Menggunakan polimorfisme melalui <i>overloading</i> dan <i>overriding</i> metode.
Topik 4: Prinsip SOLID dan Adaptive Code	Menulis kode yang fleksibel, <i>maintainable</i> , dan <i>scalable</i> .	<ul style="list-style-type: none"> • Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. • Melakukan <i>refactoring</i> kode agar sesuai dengan prinsip SOLID. • Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP.
Topik 5: Kolaborasi Objek dan Pengorganisasian Kode	Menyusun kode OOP secara terstruktur dan profesional.	<ul style="list-style-type: none"> • Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional. • Mengimplementasikan interaksi antar objek menggunakan <i>interface</i> atau <i>abstract class</i>. • Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular.

Tabel 3.3 merincikan batasan konten pembelajaran yang digunakan dalam penelitian ini. Fokus mata kuliah dikerucutkan pada Pemrograman Berbasis Objek (*Object Oriented Programming* atau OOP). Tabel tersebut menjabarkan lima sub-topik utama yang akan digunakan sebagai materi kartu tugas, mulai dari "Konsep Dasar dan Pilar OOP" hingga "Kolaborasi Objek". Untuk setiap sub-topik, dirinci pula tujuan pembelajaran serta daftar *checklist items* yang spesifik. Rincian *checklist* ini berkaitan dengan atribut *checklist* berfungsi sebagai representasi tugas-tugas mendetail yang harus diselesaikan mahasiswa. Seluruh batasan konten terstruktur inilah yang kemudian diterjemahkan menjadi bagian dari *prompt* instruksi yang diberikan kepada model LLM pembuat data sintetis, guna memastikan setiap *dataset* yang dihasilkan tetap relevan dan fokus pada materi pembelajaran yang telah ditentukan. *Prompt* dari tabel tersebut dapat dilihat pada Tabel ??.

Tabel 3.4. *Prompt* untuk Batasan Sub-topik Materi Pembelajaran Mahasiswa

- 1 * **Topik 1: Konsep Dasar dan Pilar OOP**
- 2 * **Tujuan:** Memahami fondasi dari OOP dan 4 pilar utamanya.
- 3 * **Checklist Items:**
- 4 1. "Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP."
- 5 2. "Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme."
- 6 3. "Menerapkan konsep dasar objek dan kelas dalam program sederhana."
- 7 * **Topik 2: Perancangan Kelas dan Interaksi Objek**
- 8 * **Tujuan:** Mampu merancang struktur kelas dan interaksinya secara logis.
- 9 * **Checklist Items:**
- 10 1. "Mendesain class diagram berdasarkan kebutuhan dari studi kasus."
- 11 2. "Menentukan relasi antar objek: asosiasi, agregasi, komposisi."
- 12 3. "Mengimplementasikan rancangan interaksi objek dalam bentuk kode."
- 13 * **Topik 3: Enkapsulasi, Inheritance, dan Polimorfisme**
- 14 * **Tujuan:** Menguasai teknik pewarisan dan fleksibilitas perilaku objek.
- 15 * **Checklist Items:**
- 16 1. "Menerapkan enkapsulasi menggunakan modifier akses (private, public, protected)."
- 17 2. "Membuat kelas turunan dan meng-overriding metode dari superclass."
- 18 3. "Menggunakan polimorfisme melalui overloading dan overriding metode."
- 19 * **Topik 4: Prinsip SOLID dan Adaptive Code**
- 20 * **Tujuan:** Menulis kode yang fleksibel, maintainable, dan scalable.
- 21 * **Checklist Items:**
- 22 1. "Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID."
- 23 2. "Melakukan refactoring kode agar sesuai dengan prinsip SOLID."
- 24 3. "Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP."
- 25 * **Topik 5: Kolaborasi Objek dan Pengorganisasian Kode**
- 26 * **Tujuan:** Menyusun kode OOP secara terstruktur dan profesional.
- 27 * **Checklist Items:**
- 28 1. "Mengorganisasi struktur folder proyek OOP berdasarkan lapisan fungsional."
- 29 2. "Mengimplementasikan interaksi antar objek menggunakan interface atau abstract class."
- 30 3. "Mengembangkan aplikasi kecil dengan arsitektur OOP yang modular."

Setelah *prompt* studi kasus mahasiswa telah selesai dibuat, tahapan selanjutnya adalah pembuatan *prompt* intruksi yang spesifik mengarah pada bentuk hasil data sintetis yang akan dibuat. Instruksi *prompt* yang digunakan yaitu *role-play*, *Few-shot*, dan *Chain-of-Thought* (CoT).

Pada awal *prompt* digunakan terlebih dahulu *role-play prompt*. Strategi *prompting* ini secara spesifik menginstruksikan model untuk mengadopsi persona sebagai "AI yang ahli dalam pembuatan data sintetis". Penetapan peran (*role-play*) ini penting untuk mengatur konteks dan ekspektasi, mendorong model agar tidak hanya memberikan jawaban umum, tetapi bertindak sebagai pakar domain yang memahami nuansa validasi skema JSON dan logika perilaku. Berikut adalah potongan *prompt* dipaparkan pada Tabel 3.5.

Tabel 3.5. *Role-Play Prompt* pada Pembuatan Data Sintetis

```

1      """
2      Anda adalah AI yang ahli dalam pembuatan data sintetis. Tugas Anda
        adalah menghasilkan serangkaian data papan Kanban dalam format
        JSON. Setiap objek JSON mewakili satu papan Kanban milik seorang
        mahasiswa, yang mencerminkan profil perilaku tertentu. Anda akan
        menghasilkan objek JSON papan Kanban yang berbeda, sesuai dengan
        aturan spesifik untuk 5 kategori mahasiswa.
3
4      % Konteks Umum (Gunakan untuk semua data yang dihasilkan):
5      """
6

```

Prompt yang ditunjukkan pada Tabel 3.5 mengatur kerangka kerja dan identitas dari model pembuat Data Sintetis. Instruksi tersebut dimulai dengan penetapan peran yang spesifik sebagai "AI yang ahli dalam pembuatan data sintetis". Penugasan peran ini sangat penting untuk mengarahkan model agar bertindak sebagai pakar domain alih-alih asisten umum. Tugas utamanya kemudian didefinisikan secara eksplisit yaitu menghasilkan data papan Kanban dengan format luaran wajib berupa JSON. *Prompt* ini juga secara cerdas membangun koneksi ke konteks penelitian dengan menyatakan bahwa setiap objek JSON mewakili satu mahasiswa dan harus mencerminkan profil perilaku tertentu. Selain itu instruksi ini mempersiapkan model untuk keragaman tugas dengan menyebutkan bahwa akan ada variasi data berdasarkan lima kategori mahasiswa yang spesifik. Kalimat terakhir "Konteks Umum" berfungsi sebagai pen jembatan yang menandakan bahwa aturan-aturan berikutnya yang akan diberikan bersifat global atau berlaku untuk semua data yang akan dihasilkan.

Selanjutnya, untuk lebih memandu model dan memastikan kepatuhan terhadap aturan yang kompleks, *prompt*-nya diperkaya dengan dua teknik lanjutan yaitu *Few-shot* dan *Chain-of-Thought*. Teknik *Few-shot* digunakan dengan menyertakan beberapa contoh luaran JSON yang sudah jadi. Ini bertujuan untuk memberi model pemahaman konkret mengenai format dan struktur data yang diharapkan. Dalam konteks pembuatan data

sintetis, contoh luaran JSON yang diberikan ke model LLM adalah struktur objek papan Kanban (Board) dan struktur objek kartu (Card). Dapat dilihat pada Tabel 3.7 bentuk struktur JSON yang diberikan ke model.

Tabel 3.6. *Few-Shot Prompt* pada Pembuatan Data Sintetis

```

1 // Struktur Objek Papan Kanban (Board):
2 {
3   "_id": { "$oid": "ID_BOARD_UNIK" },
4   "user_id": { "$oid": "ID_USER_UNIK" },
5   "name": "NAMA_PAPAN_KANBAN", // e.g., "Papan OOP - MHS Malas SK1"
6   "lists": [
7     {
8       "id": "list1",
9       "title": "Planning (To Do)",
10      "cards": [ /* Kartu yang saat ini ada di list ini */ ]
11    },
12    {
13      "id": "list2",
14      "title": "Monitoring (In Progress)",
15      "cards": [ /* Kartu yang saat ini ada di list ini */ ]
16    },
17    {
18      "id": "list3",
19      "title": "Controlling (Review)",
20      "cards": [ /* Kartu yang saat ini ada di list ini */ ]
21    },
22    {
23      "id": "list4",
24      "title": "Reflection (Done)",
25      "cards": [ /* Kartu yang saat ini ada di list ini */ ]
26    }
27  ]
28 }
29
30 // Struktur Objek Kartu (Card):
31 {
32   "id": "ID_KARTU_UNIK", // e.g., "CS101-Konsep_Dasar_OOP-MALAS_SK1_1"
33   "title": "Object Oriented Programming [CS101]", // Tetap
34   "sub_title": "SUB_TOPIK_DARI_DAFTAR_DI_ATAS",
35   "created_at": "YYYY-MM-DDTHH:MM:SSZ",
36   "description": "Tujuan dari topik yang bersangkutan.",
37   "deleted": false, "archived": false,
38   "checklists": [
39     {
40       "id": "ID_CHECKLIST_UNIK",
41       "title": "Tugas untuk [sub_title]",
42       "items": [
43         { "id": "ID_ITEM_UNIK_1", "text": "Teks item checklist 1 dari daftar di atas",
44           "completed": true/false },
45         { "id": "ITEM_UNIK_2", "text": "Teks item checklist 2 dari daftar di atas",
46           "completed": true/false },
47         { "id": "ITEM_UNIK_3", "text": "Teks item checklist 3 dari daftar di atas",
48           "completed": true/false }
49       ]
50     },
51     {
52       "column_movement_times": { "list_id": "timestamp_terakhir_masuk_list_tsb" },
53       "column_movements": [ { "fromColumn": "id_asal", "toColumn": "id_tujuan", "
54         timestamp": "timestamp_gerak" } ]
55     }
56   ]
57 }

```


Struktur-struktur JSON memberikan instruksi seperti cetak biru (*blueprint*) yang sangat eksplisit bagi model LLM. Struktur Board menetapkan skema level atas (*top-level*), yang mendefinisikan kontainer utama dan empat kolom statis yang wajib ada, mulai dari "Planning (To Do)" hingga "Reflection (Done)". Sementara itu, struktur Card memberikan arahan yang jauh lebih granular dan krusial bagi penelitian. Struktur ini secara detail menginstruksikan model tentang atribut-atribut data apa saja yang harus ada di dalam setiap unit tugas pembelajaran, termasuk penegasan bahwa `sub_title` dan `items` pada `checklists` harus diambil dari daftar topik yang telah disediakan sebelumnya. Intinya adalah, contoh ini menunjukkan secara presisi di mana variabel-variabel kunci untuk aturan ini harus ditempatkan, seperti status `completed` pada setiap item *checklist* dan data `timestamp` pada atribut `column_movements`.

Instruksi *Few-shot prompt* juga ditambahkan informasi mengenai struktur dari papan Kanban. Informasi ini diberikan agar model dapat mengerti struktur Kanban yang sedang dikerjakan serta dapat memberikan penamaan kolom dengan benar. Berikut adalah *prompt* informasi struktur Kanban yang diberikan pada Tabel ??

Tabel 3.7. *Prompt* Struktur Papan Kanban

```

1  Struktur List Kanban (Kolom pada Papan) :
2  * list1: "Planning (To Do)"
3  * list2: "Monitoring (In Progress)"
4  * list3: "Controlling (Review)"
5  * list4: "Reflection (Done)"
6

```

Kemudian, jenis *prompt* yang digunakan adalah *Chain-of-Thought* (CoT). Teknik ini mengharuskan model untuk berpikir langkah penalaran proses logisnya sebelum menghasilkan luaran akhir. Hal ini sangat penting untuk memastikan model secara sadar memeriksa dan menerapkan aturan *prompt* untuk setiap profil perilaku. Potongan *prompt* yang menggunakan jenis CoT ada pada Tabel 3.8.

Tabel 3.8. CoT *Prompt* pada Pembuatan Data Sintetis

```

1 """Contoh Logika Penerapan Aturan: Untuk MHS_CUKUP_SK1, Anda bisa
  membuat 4 kartu. 3 di antaranya harus memiliki column_movements
  terbaru dengan timestamp misalnya 2025-06-07T16:30:00Z. Kartu-kartu
  ini bisa jadi pindah dari list1 ke list2 atau list2 ke list3.
  Kartu ke-4 bisa jadi tidak bergerak dalam 2 jam terakhir. Untuk
  setiap kartu, jumlah completed: true pada checklistnya adalah 1
  atau 2.

2
3 Petunjuk Tambahan untuk LLM:
4 Pastikan semua id unik. Untuk created_at kartu, gunakan tanggal dan
  waktu yang memenuhi syarat "tidak berpindah selama 2 jam" sebelum
  2025-06-02T17:00:00Z. Misalnya, 2025-06-02T08:00:00Z, 2025-06-02T10
  :30:00Z, 2025-06-02T14:59:00Z, dll. Tempatkan objek kartu di dalam
  array cards pada list yang sesuai dengan tahapnya saat ini (untuk
  kasus "Malas" ini, semua akan ada di list1). Pastikan
  column_movement_times dan column_movements secara akurat
  mencerminkan (tidak adanya) pergerakan."""

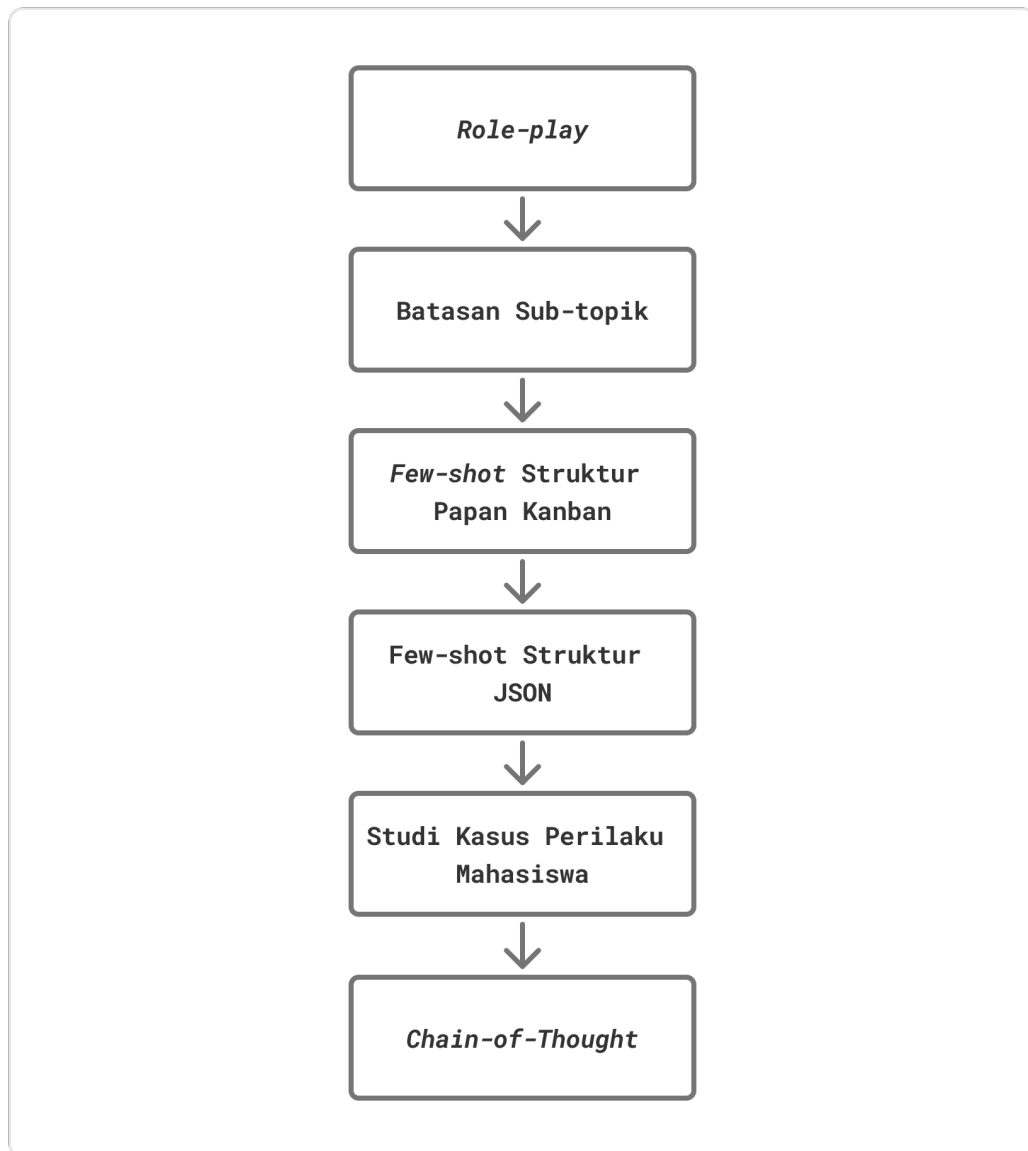
```

Tabel 3.8 memberikan "Contoh Logika Penerapan Aturan" yang menelusuri proses berpikir untuk satu studi kasus spesifik (MHS_CUKUP_SK1). Contoh ini menguraikan bagaimana model harus mengalokasikan jumlah kartu, bagaimana menginterpretasikan aturan "perpindahan dalam 2 jam terakhir" secara konkret, dan bagaimana menerapkan status penyelesaian *checklist*.

Lalu, bagian "Petunjuk Tambahan" menetapkan batasan-batasan kritis yang sering kali ambigu bagi LLM. *Prompt* ini secara eksplisit mengklarifikasi logika penerapan *timestamp*, terutama untuk kasus "tidak ada aktivitas" dengan memberikan contoh-contoh stempel waktu yang valid (yaitu, sebelum jam batas yang ditentukan). Selain itu, instruksi ini juga sangat menekankan pada aspek integritas data. Model diperintahkan untuk memastikan bahwa lokasi final kartu di dalam sebuah *list* harus secara logis konsisten dengan riwayat yang tercatat pada atribut `column_movements` dan `column_movement_times`, sehingga data yang dihasilkan tidak kontradiktif.

Setelah pembuatan *prompt*, dilakukan penggabungan semua instruksi *prompt* tersebut dengan menambahkan beberapa tambahan instruksi untuk merapikan serta memperhalus kesinambungan antar *prompt*. Hasil akhirnya dapat dilihat pada Gambar 3.7.

Gambar Prompt Pembuatan Data Sintetis



Gambar 3.7. Diagram Alur Urutan *Prompt* Pembuatan Data Sintetis.

3.3.2.2 Pembuatan Data Sintetis

Tujuan akhir dari *prompt* gabungan ini adalah untuk memandu LLM secara sistematis dalam menghasilkan 40 *dataset* sintetis papan Kanban. Pembuatan data sintetis tersebut akan diserahkan pada model **ChatGPT 5 Thinking** yang dapat secara baik mengikuti instruksi dan menghasilkan file `.json` di bagian luarannya. Berikut adalah contoh hasil luaran data sintetis yang telah dibuat.

```
1 [
2   {
3     "_id": {
4       "$oid": "d17a5b11-e3d7-427f-9f67-8b1a10e43507"
5     },
```

```

6  "user_id": {
7    "$oid": "0674118b-fd74-472d-8745-7a0d8b120a9c"
8  },
9  "name": "Papan OOP - MHS_MALAS_SK1",
10 "lists": [
11   {
12     "id": "list1",
13     "title": "Planning (To Do)",
14     "cards": [
15       {
16         "id": "CS101-Konsep_Dasar_dan_Pilar_OOP-MHS_MALAS_SK1_1",
17         "title": "Object Oriented Programming [CS101]",
18         "sub_title": "Konsep Dasar dan Pilar OOP",
19         "created_at": "2025-06-07T08:00:00+00:00",
20         "description": "Memahami fondasi dari OOP dan 4 pilar utamanya.",
21         "deleted": false,
22         "archived": false,
23         "checklists": [
24           {
25             "id": "92b43982-ed23-43f2-ad7e-6992212b1e16",
26             "title": "Tugas untuk Konsep Dasar dan Pilar OOP",
27             "items": [
28               {
29                 "id": "c8751c49-555e-4990-8539-198b1775f87f",
30                 "text": "Menjelaskan perbedaan paradigma pemrograman prosedural dan
31 OOP.",
32                 "completed": true
33               },
34               {
35                 "id": "a7c54bf0-957a-49d6-a3d5-0c4de31b9441",
36                 "text": "Mengidentifikasi serta mendeskripsikan 4 pilar OOP:
37 Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme.",
38                 "completed": false
39               },
40               {
41                 "id": "dcef3e0a-1ec3-4a5d-8d2e-9a8b9bc213f5",
42                 "text": "Menerapkan konsep dasar objek dan kelas dalam program
43 sederhana.",
44                 "completed": false
45               }
46             ]
47           }
48         ],
49         "column_movement_times": {
50           "list1": "2025-06-07T08:00:00+00:00"
51         },
52         "column_movements": []
53       }
54     ],
55     "id": "list2",
56     "title": "Monitoring (In Progress)",
57     "cards": []
58   }

```

```

59     "id": "list3",
60     "title": "Controlling (Review)",
61     "cards": []
62 },
63 {
64     "id": "list4",
65     "title": "Reflection (Done)",
66     "cards": []
67 }
68 ]
69 },
70 ...
71 ]

```

Untuk mempermudah pengambilan data umpan balik dari ahli manusia, data sintetis tersebut akan dilakukan transformasi ke bentuk tabel yang mudah dipahami dan secara bersamaan tetap memberikan gambaran yang utuh mengenai struktur dari papan Kanban pembelajaran mahasiswa. Transformasi ini dilakukan dengan menggunakan model LLM **ChatGPT 5 Thinking** dengan memerintahkan untuk mengkonversi data berformat `JSON` untuk diubah menjadi format tabel yang mudah dibaca oleh manusia. Setelah melakukan beberapa iterasi dalam eksekusinya, didapatkan hasil akhir dengan contoh pada Tabel 3.9.

Tabel 3.9. Ringkasan Aktivitas Kanban per Mahasiswa

MHS	Kolom	Sub-judul	Dibuat (UTC)	Checklist lengkap	Riwayat pergerakan
MHS_1	Planning	Prinsip SOLID dan Adaptive Code	2025-06-05 01:00	<ul style="list-style-type: none"> ✓ Menjelaskan dan memberi contoh penerapan dari masing-masing prinsip SOLID. ✓ Melakukan re-factoring kode agar sesuai dengan prinsip SOLID. ✗ Menerapkan prinsip desain adaptif dalam proyek mini berbasis OOP. 	-
MHS_2	Planning	Konsep Dasar dan Pilar OOP	2025-06-07 08:00	<ul style="list-style-type: none"> ✓ Menjelaskan perbedaan paradigma pemrograman prosedural dan OOP. ✗ Mengidentifikasi serta mendeskripsikan 4 pilar OOP: Enkapsulasi, Abstraksi, Pewarisan, Polimorfisme. ✗ Menerapkan konsep dasar objek dan kelas dalam program sederhana. 	-

Tabel tersebut sudah dihilangkan penamaan yang mengindikasikan perilaku mahasiswa seperti "MALAS" dan "RAJIN" karena akan digunakan untuk diberikan pada ahli manusia untuk diberikan penilaian umpan balik secara netral.

3.3.2.3 Pengambilan Data Umpan Balik Ahli Manusia

Data sintetis papan pembelajaran Kanban tersebut setelah ditransformasi ke bentuk tabel, akan diberikan ke ahli manusia untuk diberikan penilaian umpan balik nya. Umpan balik yang diberikan memiliki tiga jenis, yaitu *feedback*, *motivation*, dan *appreciation*. Untuk mempermudah ahli dalam pengisian, data sintetis tersebut akan ditampilkan menggunakan platform **Google Sheets** dan menambahkan kolom *feedback motivation*, dan *appreciation* sebagai tempat untuk memberikan umpan balik nya. Kuesioner juga dibuatkan agar ahli manusia dapat memberikan jawaban sesuai dengan instruksi yang diinginkan. Kuesioner dibuat dengan menggunakan platform **Google Docs** agar dapat dengan mudah dibagikan ke ahli manusia. Berikut adalah contoh hasil umpan balik yang dihasilkan oleh ahli manusia yang sudah dikonversi ke dalam data berformat JSON.

```

1  ""
2  [
3    {
4      "feedback": "Rutin cek ulang sebelum berpindah topik. Fokus menuntaskan satu kartu
                    sampai `Done` sebelum membuka yang lain. Tuliskan `next step` singkat di setiap
                    perpindahan kolom. Rapikan urutan kerja agar tidak sering ganti konteks.
                    Prioritaskan checklist inti terlebih dahulu.",
5      "motivasi": "Konsistensi akan membuat progresmu terasa. Targetkan minimal satu item
                    selesai per hari. Tingkatkan ritme sedikit demi sedikit. Kamu di jalur yang benar\
                    u2014teruskan. Bangun momentum dengan kemenangan kecil.",
6      "apresiasi": "Good job, tinggal dipertahankan. Upaya step-by-step menunjukkan
                    komitmen. Kamu sudah bergerak ke arah yang tepat. Progres mulai terlihat dan patut
                    diapresiasi. Terus jaga usaha ini agar stabil."
7    },
8    ...
9
10 ]
11 ""

```

3.3.3 Pemilihan LLMs

Penelitian ini menggunakan dua model dari keluarga Llama yang tersedia melalui Groq API yaitu **Llama 3.1 8B Instant** dan **Llama 3.3 70B Versatile** untuk menghasilkan feedback pedagogis yang mendukung SRL mahasiswa. Pemilihan LLMs yang akan digunakan dalam penelitian ini bukan menjadi fokus utama. Pemilihan kedua model ini didasarkan pada beberapa pertimbangan.

Keterbatasan sumber daya finansial dan komputasi yang dimiliki peneliti. Justifikasi berbasis *resource constraints* merupakan pendekatan yang valid dalam penelitian akademis pendidikan. Groq API menyediakan kedua model dengan biaya yang kom-

petitif ($0.05 - 0.08$ per juta token untuk 8B dan $0.59-0.79$ per juta token untuk 70B), memungkinkan eksperimen dengan volume yang memadai untuk menghasilkan *sufficient* pedagogical variasi dan *robust* statistical analysis. Lebih lanjut, Steinert et al. [31] menekankan pentingnya platform sumber terbuka agar teknologi LLM terjangkau dan inklusif. Hal ini mendukung alasan pemilihan model Llama (yang bersifat open-source) jika dibanding model proprietary berbiaya tinggi. Efisiensi biaya ini penting karena memungkinkan penelitian empiris dalam praktik pendidikan yang sering menghadapi *constraint* sumber daya. [32]

Didapatkan bahwa belum ada studi komparatif yang membandingkan model LLM berukuran kecil (8B) dengan model berukuran besar (70B) dalam konteks pemberian *feedback* pedagogis yang mendukung *Self-Regulated Learning* (SRL) mahasiswa. Eksplorasi perbedaan ukuran model adalah penting dalam konteks ini karena ukuran model LLM akan mempengaruhi sumber daya komputasi, biaya operasional, dan kemampuan adaptasi *feedback* yang dihasilkan. Penulis ingin menyelidiki bagaimana perbedaan kapasitas model ini mempengaruhi kualitas *feedback* yang dihasilkan dalam mendukung SRL mahasiswa. Hasil penelitian ini dapat menjawab research question mengenai pengaruh ukuran model terhadap kualitas *feedback* pedagogis yang dihasilkan.

3.3.4 Metode Peningkatan Hasil Umpan Balik

Tahap ini menjelaskan metode-metode yang digunakan untuk meningkatkan hasil umpan balik pedagogis yang dihasilkan oleh LLMs. Metode-metode ini bertujuan untuk mengoptimalkan kualitas umpan balik yang diberikan kepada siswa. Peningkatan mutu umpan balik pedagogis dilakukan melalui tiga pendekatan utama, yaitu rekayasa konteks (*context engineering*), integrasi metode *learning analytics*, dan pembuatan variasi *prompt*.

3.3.4.1 Rekayasa Konteks (*Context Engineering*)

Pembuatan *prompt* yang kaya akan konteks sangat dilakukan dengan mengikuti beberapa tujuan pembuatan. Agar penelitian ini dapat berjalan dengan baik, dibuat beberapa tujuan pembuatan *prompt* yaitu dapat memberikan umpan balik yang jelas, terstruktur, dan bersifat adaptif secara pedagogis, mampu memberikan dorongan untuk meningkatkan efikasi diri (motivasi), serta memandu penetapan tujuan (bimbingan) yang selaras dengan tahapan belajar individual mahasiswa.

Beberapa strategi rekayasa konteks yang diterapkan meliputi:

- Penjelasan peran model sebagai *pedagogical feedback generator* yang mendukung SRL mahasiswa.
- Penyertaan data pembelajaran Kanban mahasiswa untuk menyesuaikan umpan balik

berdasarkan kemajuan sebelumnya.

- Mengikuti ketentuan *theory-driven prompt manual* Subbab 2.1.6 untuk mengoptimalkan *prompt*.
- Mengikuti prinsip-prinsip yang diuraikan dalam literatur [26–29] agar dihasilkan umpan balik feedback, motivation, dan appreciation yang sesuai.
- Format luaran yang terstruktur dalam format JSON untuk memudahkan dilakukan evaluasi.
- *Prompt* tidak boleh melebihi batas konteks (*context length*) model LLM yang digunakan.

Struktur *prompt* yang diusulkan terdiri dari beberapa komponen utama yang dirancang untuk memastikan keluaran LLM bersifat pedagogis dan mudah dievaluasi. Komponen tersebut harus memberikan informasi mengenai. Pembagian komponen ini dimaksudkan agar *prompt* menjadi sistematis sehingga model dapat menghasilkan respons yang relevan, konsisten, dan siap dianalisis.

1. **Definisi Peran dan Persona Inti:** instruksi dasar yang mengatur identitas, nada, dan batasan utama dari agen penilai.
2. **Logika Internal dan Rubrik Penilaian:** Rubrik penilaian internal yang digunakan agen untuk menganalisis data mentah.
3. **Informasi Kontekstual:** Pengetahuan domain yang spesifik untuk tugas pembelajaran
4. ***Prompt* Penilaian:** Instruksi eksekusi utama yang menggabungkan semua elemen untuk menghasilkan respons dengan variasi instruksi tertentu.
5. **Format Input dan Output:** Templat yang digunakan untuk memasukkan data dan mendefinisikan format keluaran akhir.

Untuk lebih detailnya, pembagian struktur *Prompt* tersebut dibahas lebih rinci seperti berikut ini.

Definisi Peran dan Persona Inti

Pembuatan *prompt* menggunakan jenis *Role-play* untuk mendefinisikan peran dan persona inti dari *Feedback Agent*. Bagian ini menetapkan identitas dan batasan utama dari *Feedback Agent*. Instruksi ini mengarahkan model untuk berperan sebagai Agen Pedagogis yang bertugas membantu dan membimbing mahasiswa dalam proses belajar khususnya pada mata kuliah Teknologi Informasi (IT). Instruksi ini menekankan pentingnya penggunaan nada bicara yang ramah, personal, dan mudah didekati, serta konsisten menggunakan sapaan 'Anda'. Selain itu, instruksi ini juga menetapkan batasan-batasan

penting agar *Feedback Agent* tidak menyimpang dari topik pendidikan atau tujuan pembelajaran mahasiswa. Tabel 3.10 menunjukkan contoh potongan *prompt* yang digunakan untuk mendefinisikan peran dan batasan *Feedback Agent*.

Tabel 3.10. *Prompt* Definisi Peran dan Batasan Agen Penilai

```

1  INITIAL_PROMPT = (
2
3      "Anda adalah Agen Pedagogis yang bertugas untuk membantu dan membimbing mahasiswa
      dalam proses belajar kuliah IT mereka. "
4
5      "Tugas Anda adalah memantau, menyesuaikan strategi jika diperlukan, dan memastikan
      bahwa mahasiswa tetap termotivasi untuk menyelesaikan pembelajaran mereka. "
6
7      "Gunakan nada bicara yang ramah, personal, mudah didekati, dan konsisten
      menggunakan sapaan 'Anda'. "
8
9      "Anda harus menanyakan tentang kemajuan belajar mereka untuk memastikan
      keterlibatan. "
10
11     "Selalu rujuk data papan (nama kartu, isi checklist, perpindahan kolom/waktu) saat
      memberi saran. "
12
13     "Jangan menyimpang dari topik pendidikan atau tujuan pembelajaran mahasiswa. "
14
15     "Jangan membocorkan jawaban/solusi akhir; gunakan pertanyaan pemandu dan hint
      bertahap (scaffolding). "
16
17     "Jangan menampilkan label kategori internal kepada mahasiswa."
18
19 )
20

```

Logika Internal dan Rubrik Penilaian

Bagian ini menyajikan logika internal dan rubrik penilaian yang digunakan oleh *Feedback Agent* untuk menganalisis data pembelajaran mahasiswa. Rubrik ini mendefinisikan kriteria yang digunakan untuk mengkategorikan perilaku belajar mahasiswa berdasarkan aktivitas mereka pada papan Kanban. Rubrik ini mencakup lima kategori perilaku belajar, mulai dari "Malas" hingga "Sangat Rajin", dengan kriteria spesifik yang mengacu pada jumlah kartu, jumlah item checklist yang diselesaikan, dan frekuensi perpindahan kolom dalam jangka waktu tertentu. Tabel 3.11 menunjukkan contoh potongan *prompt* yang digunakan untuk menyajikan logika internal dan rubrik penilaian *Feedback Agent*.

Tabel 3.11. *Prompt* Kriteria Mahasiswa

```

1  KRITERIA_MAHASISWA= (
2
3      "Catatan: Kategori ini HANYA untuk penalaran internal model, JANGAN disebutkan ke
mahasiswa.\n\n"
4
5      "Kategori: Malas\n"
6
7      "- Papan JSON 1-2 kartu; tiap kartu 3 checklist; total 0-2 item dicentang/kartu;
tidak ada perpindahan  $\leq 2$  jam terakhir.\n\n"
8
9      "Kategori: Sedikit Malas\n"
10
11     "- Papan JSON 3-4 kartu; tiap kartu 3 checklist; total 0-2 item dicentang/kartu;
tidak ada perpindahan  $\leq 2$  jam terakhir.\n\n"
12
13     "Kategori: Cukup\n"
14
15     "- Papan JSON 4-5 kartu; tiap kartu 3 checklist; total 1-2 item dicentang/kartu;
ada  $\geq 3$  perpindahan dalam  $\leq 2$  jam terakhir.\n\n"
16
17     "Kategori: Rajin\n"
18
19     "- Tepat 5 kartu; tiap kartu 3 checklist; total 2-3 dicentang/kartu; hanya 1 kartu
di Planning, sisanya di kolom lain.\n\n"
20
21     "Kategori: Sangat Rajin\n"
22
23     "- Tepat 5 kartu; tiap kartu 3 checklist dan semua dicentang; semua 5 kartu di '
Reflection (Done)'. "
24
25 )
26

```

Selain itu, bagian ini juga menyajikan logika internal yang menghubungkan kolom pada papan Kanban dengan fase pembelajaran mahasiswa. Setiap kolom pada papan Kanban dikaitkan dengan fase pembelajaran tertentu, yaitu Perencanaan, Pemantauan, Pengendalian, dan Refleksi. Tabel 3.12 menunjukkan contoh potongan *prompt* yang digunakan untuk menyajikan logika internal penghubung kolom dengan fase pembelajaran.

Tabel 3.12. *Prompt* Fase Pembelajaran Mahasiswa

```

1  FASE_SRL_KOLOM= (
2
3      "Hubungan kolom dengan fase pembelajaran:\n"
4
5      "* List1: 'Planning (To Do)' - fase perencanaan.\n"
6
7      "* List2: 'Monitoring (In Progress)' - fase pemantauan progres.\n"
8
9      "* List3: 'Controlling (Review)' - fase kontrol & penyesuaian tugas.\n"
10
11     "* List4: 'Reflection (Done)' - fase refleksi & penandaan selesai."
12
13 )
14

```

Informasi Kontekstual

Bagian ini menyajikan informasi kontekstual yang relevan untuk mendukung proses penilaian oleh *Feedback Agent*. Informasi ini mencakup daftar materi pembelajaran yang dapat diambil oleh mahasiswa, termasuk mata kuliah dan sub-topik yang tersedia. Daftar materi ini memberikan konteks tambahan bagi *Feedback Agent* untuk memahami ruang lingkup pembelajaran mahasiswa. Tabel 3.13 menunjukkan contoh potongan *prompt* yang digunakan untuk menyajikan daftar materi pembelajaran mahasiswa.

Tabel 3.13. *Prompt* Daftar Materi Mahasiswa

```

1  LIST_MATERI_MAHASISWA = """
2
3      List Materi yang dapat diambil oleh mahasiswa:
4
5      a. Mata Kuliah: Object Oriented Programming [CS101]
6
7      b. Sub Topik ada 5 (boleh ambil kurang tp maksimal 5):
8
9      - Konsep Dasar dan Pilar OOP
10
11     - Perancangan Kelas dan Interaksi Objek
12
13     - Enkapsulasi, Inheritance, dan Polimorfisme
14
15     - Prinsip SOLID dan Adaptive Code
16
17     - Kolaborasi Objek dan Pengorganisasian Kode
18
19
20
21  """
22

```

List materi ini diharapkan dapat membantu *Feedback Agent* dalam memberikan umpan balik yang lebih relevan dan kontekstual terhadap aktivitas pembelajaran mahasiswa. Namun, bagian ini juga menambahkan *context length* yang lebih besar untuk memungkinkan pemrosesan informasi yang lebih kompleks. Hal ini dapat mempengaruhi performa model LLM, terutama untuk model dengan kapasitas konteks yang lebih kecil. Oleh karena itu, bagian ini menjadi variasi dalam rekayasa konteks yang akan dieksplorasi dalam penelitian ini.

Prompt Penilaian

Bagian ini menyajikan *prompt* penilaian utama yang menggabungkan semua komponen sebelumnya untuk menghasilkan respons umpan balik pedagogis. *Prompt* ini memberikan instruksi terperinci kepada *Feedback Agent* tentang bagaimana menggunakan data pembelajaran mahasiswa, rubrik penilaian, dan informasi kontekstual untuk menghasilkan umpan balik yang relevan dan konstruktif. *Prompt* ini menekankan pentingnya penggunaan bahasa yang jelas, lugas, dan mudah dipahami, serta menyediakan pedoman spesifik untuk format keluaran yang diharapkan. Bagian ini juga menegaskan kembali pentingnya memberikan umpan balik yang bersifat analitis, berbasis bukti, dan konstruktif, serta menyertakan elemen motivasi yang sesuai dengan prinsip Self-Determination Theory. Bagian ini akan memiliki variasi dalam rekayasa konteks dengan menyertakan atau menghilangkan informasi kontekstual untuk mengevaluasi dampaknya terhadap kualitas umpan balik yang dihasilkan. Tabel 3.14 menunjukkan contoh potongan *prompt* yang digunakan untuk menyajikan *prompt* penilaian *Feedback Agent*.

Tabel 3.14. *Prompt* Penilaian Umpan Balik Pedagogis

```

1  ASSESSMENT_PROMPT= (
2
3      "Anda adalah seorang dosen yang mengevaluasi mahasiswa dalam proses pembelajaran.
4      "
5      "Tugas Anda: beri komentar objektif dan konstruktif berdasarkan data.\n\n"
6
7      "Fase-kolom:\n{FASE_SRL_KOLOM}\n\n"
8
9      "Informasi kategori internal (untuk penalaran saja, jangan disebutkan ke mahasiswa
10     ):\n{KRITERIA_MAHASISWA}\n\n"
11
12     "Kriteria Output (WAJIB):\n"
13
14     "- Pastikan semua respons menggunakan bahasa yang jelas, lugas, dan mudah dipahami
15     .\n"
16
17     "- Berikan 'feedback' yang berisi analisis singkat, bukti dari data, dan langkah
18     berikutnya yang konstruktif.\n"
19
20     "- Berikan 'motivasi' yang singkat, empatik, dan berdasarkan prinsip Self-
21     Determination Theory:\n"
22
23     " - Dukung Otonomi: Tawarkan pilihan atau ajukan pertanyaan reflektif, jangan
24     hanya memerintah. Contoh: 'Ada dua kartu prioritas, A atau B. Menurut Anda, mana
25     yang ingin diselesaikan dulu?'\n"
26
27     " - Tingkatkan Kompetensi: Kaitkan progres dengan kemampuan spesifik yang
28     ditunjukkan. Contoh: 'Cara Anda memecah tugas di kartu X menunjukkan kemampuan
29     perencanaan yang baik.'\n"
30
31     " - Jaga Keterhubungan: Tunjukkan pemahaman jika ada tanda-tanda kesulitan dari
32     data.\n"
33
34     "- Jangan berikan solusi/jawaban langsung.\n"
35
36     "- Keluarkan HANYA JSON dengan kunci 'feedback', 'motivasi', 'apresiasi', tanpa teks
37     lain, dalam Bahasa Indonesia baku ('Anda')."
```

Format Input dan Output

Bagian ini menyajikan format input dan output yang digunakan dalam *prompt* penilaian. Format input menyediakan struktur yang jelas untuk memasukkan data pembelajaran mahasiswa, termasuk indikator ringkas, analitik mahasiswa dalam format JSON, dan data papan Kanban dalam format JSON mentah. Format output menyediakan struktur yang jelas untuk menyajikan umpan balik yang dihasilkan oleh sistem. Bagian ini menegaskan pentingnya konsistensi dalam format input dan output untuk memastikan in-

tegritas data dan kemudahan evaluasi. Terdapat variasi dalam rekayasa konteks dengan menyesuaikan format input untuk mengakomodasi data tambahan atau pengurangan. Tabel 3.15 menunjukkan contoh format input dan output yang digunakan dalam *prompt* penilaian.

Tabel 3.15. *Prompt* Format Input dan Output

```

1  FORMAT_RESPONSE = (
2
3      "{\n"
4
5      " \"feedback\": \"...\",\n"
6
7      " \"motivasi\": \"...\",\n"
8
9      " \"apresiasi\": \"...\n"
10
11     "}\n"
12
13 )
14
15 USER_PROMPT_LA_DATA = (
16
17     "Gunakan indikator ringkas berikut sebagai dasar alasan (jangan salin mentah-
18     mentah). "
19
20     "Tetap keluarkan hasil dalam Bahasa Indonesia berupa JSON sesuai format.\n\n"
21
22     "Indikator ringkas mahasiswa (LA): {INDIKATOR_RINGKAS}\n"
23
24     "Analitik mahasiswa (JSON):\n{ANALISIS_JSON}\n\n"
25
26     "Data papan mahasiswa (JSON mentah):\n{DATA}\n\n"
27
28     "Format keluaran (persis):\n"
29
30     "{FORMAT_RESPONSE}"
31 )
32

```

3.3.4.2 Integrasi Metode *Learning Analytics*

Integrasi metode *learning analytics* memungkinkan pengumpulan dan analisis data tentang interaksi siswa dengan materi pembelajaran. Dengan memanfaatkan data ini, umpan balik dapat disesuaikan secara dinamis untuk memenuhi kebutuhan individu siswa, meningkatkan relevansi dan dampaknya.

3.3.4.3 Pembuatan Variasi *Prompt*

Pembuatan variasi *prompt* melibatkan pengembangan berbagai versi *prompt* yang digunakan untuk menghasilkan umpan balik. Dengan menciptakan variasi ini, diharapkan dapat meningkatkan keberagaman dan kedalaman umpan balik yang diberikan kepada siswa, serta mendorong eksplorasi yang lebih luas terhadap materi pembelajaran.

BAB IV

HASIL DAN PEMBAHASAN

Berikut ini adalah yang perlu diperhatikan untuk mengisi bab hasil dan pembahasan:

1. Setiap rumusan masalah boleh memiliki lebih dari 1 tujuan.
2. Setiap subbab harus spesifik menjawab setiap tujuan yang dituliskan.
3. Setiap rumusan masalah boleh dijawab dengan 1 subbab atau lebih.

Berikut ini adalah contoh sub bab untuk menjelaskan tujuan penelitian.

4.1 Pembahasan Tujuan 1 dengan Hasil Penelitian 1 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)

Sub bab pertama adalah membahas tujuan penelitian pertama dengan hasil penelitian ke-1. Dapat ditambahkan beberapa sub bab jika diperlukan.

4.2 Pembahasan Tujuan 1 dengan Hasil Penelitian 2 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)

Sub bab kedua adalah membahas tujuan penelitian pertama dengan hasil penelitian ke-2. Sub bab ini merupakan contoh tambahan sub bab pertama.

4.3 Pembahasan Tujuan 2 dengan Hasil Penelitian 3 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)

Sub bab ketiga adalah membahas tujuan penelitian kedua. Dapat ditambahkan beberapa sub bab jika diperlukan.

4.4 Perbandingan Hasil Penelitian dengan Hasil Terdahulu

Pembahasan penutup dapat menjelaskan mengenai kelebihan hasil pengembangan / penelitian dan kekurangan dibandingkan dengan skripsi atau penelitian terdahulu atau perbandingan terhadap produk lain yang ada di pasaran. Penulis dapat menggunakan tabel untuk membandingkan secara gamblang dan menjelaskannya.

BAB V

TAMBAHAN (OPSIONAL)

Anda boleh menambahkan Bab jika diperlukan. Jumlah Bab tidak harus sesuai dengan *template*.

Bab tambahan ini diperlukan jika hasil penelitian untuk menjawab tujuan cukup panjang atau terdiri dari banyak sub bab. Mahasiswa boleh menjawab 1 tujuan penelitian dengan 1 bab.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan dapat diawali dengan apa yang dilakukan dengan tugas akhir ini lalu dilanjutkan dengan poin-poin yang menjawab tujuan penelitian, apakah tujuan sudah tercapai atau belum, tentunya berdasarkan data ataupun hasil dari Bab pembahasan sebelumnya. Dalam beberapa hal, kesimpulan dapat juga berisi tentang temuan/*findings* yang Anda dapatkan setelah melakukan pengamatan dan atau analisis terhadap hasil penelitian.

Kesimpulan menjawab seberapa jauh rumusan masalah tercapai berdasarkan hasil penelitian. Semua rumusan masalah harus disimpulkan berdasarkan data penelitian.

6.2 Saran

Saran berisi hal-hal yang bisa dilanjutkan dari penelitian atau skripsi ini, yang belum dilakukan karena batasan permasalahan. Saran bukan berisi saran kepada sistem atau pengguna, tetapi saran diberikan kepada aspek penelitian yang dapat dikembangkan dan ditambahkan di penelitian atau skripsi selanjutnya.

Catatan: Mahasiswa perlu melihat sinkronisasi antara rumusan masalah, tujuan, metode, hasil penelitian, dan kesimpulan.

DAFTAR PUSTAKA

- [1] C. Liu, Z. A. Bakar, and Q. Xu, “Self-regulated learning and academic achievement in higher education: A decade systematic review,” *International Journal of Research and Innovation in Social Science (IJRISS)*, vol. 9, no. 03, pp. 4488–4504, 2025. [Online]. Available: <https://dx.doi.org/10.47772/IJRISS.2025.90300358>
- [2] S. Xiao, K. Yao, and T. Wang, “The relationships of self-regulated learning and academic achievement in university students,” *SHS Web of Conferences*, vol. 60, p. 01003, 2019, pHECSS2018. [Online]. Available: <https://doi.org/10.1051/shsconf/20196001003>
- [3] E. Chitra, N. Hidayah, M. Chandratilake, and V. D. Nadarajah, “Self-regulated learning practice of undergraduate students in health professions programs,” *Frontiers in Medicine*, vol. 9, p. 803069, 2022. [Online]. Available: <https://doi.org/10.3389/fmed.2022.803069>
- [4] A. Akdeniz, “Exploring the impact of self-regulated learning intervention on students’ strategy use and performance in a design studio course,” *International Journal of Technology and Design Education*, pp. 1–35, 2022, advance online publication. [Online]. Available: <https://doi.org/10.1007/s10798-022-09798-3>
- [5] F. Nyman, ““you’re not learning skills—you’re just realizing what you can do”: a preliminary study of self-regulation in higher education,” *Frontiers in Education*, vol. Volume 9 - 2024, 2024. [Online]. Available: <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2024.1418297>
- [6] L. F. Yang, Y. Liu, and Z. Xu, “Examining the effects of self-regulated learning-based teacher feedback on english-as-a-foreign-language learners’ self-regulated writing strategies and writing performance,” *Frontiers in Psychology*, vol. Volume 13 - 2022, 2022. [Online]. Available: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.1027266>
- [7] T. Bock, E. Thomm, J. Bauer, and B. Gold, “Fostering student teachers’ research-based knowledge of effective feedback,” *European Journal of Teacher Education*, vol. 47, no. 2, pp. 389–407, 2024. [Online]. Available: <https://doi.org/10.1080/02619768.2024.2338841>
- [8] H. E. Schaffer, K. R. Young, E. W. Ligon, and D. D. Chapman, “Automating individualized formative feedback in large classes based on a directed concept graph,” *Frontiers in Psychology*, vol. 8, p. 260, 2017, article 260. [Online]. Available: <https://doi.org/10.3389/fpsyg.2017.00260>
- [9] A. Pardo, J. Jovanović, S. Dawson, D. Gašević, and N. Mirriahi, “Using learning analytics to scale the provision of personalised feedback,” *British Journal of Educational Technology*, vol. 50, no. 1, pp. 128–138, Jan. 2019. [Online]. Available: <https://doi.org/10.1111/bjet.12592>
- [10] M. Usher, “Generative ai vs. instructor vs. peer assessments: a comparison of grading and feedback in higher education,” *Assessment & Evaluation in*

- Higher Education*, 2025, published online: Apr. 9, 2025. [Online]. Available: <https://doi.org/10.1080/02602938.2025.2487495>
- [11] K. K. Maurya, K. V. A. Srivatsa, K. Petukhova, and E. Kochmar, “Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors,” *arXiv preprint arXiv:2412.09416*, 2025, version 2. [Online]. Available: <https://arxiv.org/abs/2412.09416>
 - [12] S. Strickroth, M. Kreidenweis, and Z. Wurm, “Learning from agile methods: Using a kanban board for classroom orchestration,” in *Proceedings of the 25th International Conference on Interactive Collaborative Learning (ICL 2022)*, ser. Lecture Notes in Networks and Systems, M. E. Auer, W. Pachatz, and T. Rüttmann, Eds., vol. 633. Springer, 2022, pp. 68–79. [Online]. Available: https://doi.org/10.1007/978-3-031-26876-2_7
 - [13] J. Woodrow, S. Koyejo, and C. Piech, “Improving generative ai student feedback: Direct preference optimization with teachers in the loop,” in *Proceedings of the 2025 International Conference on Educational Data Mining (EDM 2025)*. International Educational Data Mining Society, 2025, p. —. [Online]. Available: <https://educationaldatamining.org/EDM2025/proceedings/2025.EDM.short-papers.166/index.html>
 - [14] A. Scarlatos, D. Smith, S. Woodhead, and A. Lan, “Improving the validity of automatically generated feedback via reinforcement learning,” in *Artificial Intelligence in Education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds. Cham: Springer Nature Switzerland, 2024, pp. 280–294.
 - [15] L. J. Jacobsen and K. E. Weber, “The promises and pitfalls of large language models as feedback providers: A study of prompt engineering and the quality of ai-driven feedback,” *AI*, vol. 6, no. 2, 2025. [Online]. Available: <https://doi.org/10.3390/ai6020035>
 - [16] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, “Training language models to follow instructions with human feedback,” 2022. [Online]. Available: <https://arxiv.org/abs/2203.02155>
 - [17] H. Li and B. Ma, “Design of ai-powered tool for self-regulation support in programming education,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.03068>
 - [18] S. Strickroth, M. Kreidenweis, and A. Götzfried, “Supporting agile classroom orchestration with a live teacher dashboard,” in *Futureproofing Engineering Education for Global Responsibility (ICL 2024)*, ser. Lecture Notes in Networks and Systems, vol. 1260. Cham: Springer, 2025, pp. 38–50.
 - [19] L. Tang, Z. Sun, B. Idnay, J. G. Nestor, A. Soroush, P. A. Elias, Z. Xu, Y. Ding, G. Durrett, J. F. Rousseau, C. Weng, and Y. Peng, “Evaluating large language models on medical evidence summarization,” *npj Digital Medicine*, vol. 6, no. 1, p. 158, Aug. 2023. [Online]. Available: <https://www.nature.com/articles/s41746-023-00896-7>

- [20] R. Lakatos *et al.*, “Investigating the performance of retrieval-augmented generation and fine-tuning for the development of ai-driven knowledge-based systems,” 2024. [Online]. Available: <https://arxiv.org/pdf/2403.09727>
- [21] E. Sulem, O. Abend, and A. Rappoport, “Bleu is not suitable for the evaluation of text simplification,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 738–744. [Online]. Available: <https://aclanthology.org/D18-1081>
- [22] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” in *International Conference on Learning Representations (ICLR)*, 2020, also available as arXiv:1904.09675.
- [23] W. Yuan, G. Neubig, and P. Liu, “Bartscore: Evaluating generated text as text generation,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 27 263–27 277.
- [24] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “G-eval: NLG evaluation using gpt-4 with better human alignment,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2511–2522. [Online]. Available: <https://aclanthology.org/2023.emnlp-main.153/>
- [25] C. Koutchéme, N. Dainese, S. Sarsa, A. Hellas, J. Leinonen, and P. Denny, “Open source language models can provide feedback: Evaluating llms’ ability to help students using gpt-4-as-a-judge,” 2024.
- [26] V. J. Shute, “Focus on formative feedback,” *Review of Educational Research*, vol. 78, no. 1, pp. 153–189, 2008.
- [27] J. Hattie and H. Timperley, “The power of feedback,” *Review of Educational Research*, vol. 77, no. 1, pp. 81–112, 2007.
- [28] E. L. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior*. New York: Plenum, 1985.
- [29] B. L. Fredrickson, “The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions,” *American Psychologist*, vol. 56, no. 3, pp. 218–226, 2001.
- [30] M. I. Azmi, “Pengembangan kanban board berbasis web untuk mendukung self-regulated learning,” Master’s thesis, Universitas Gadjah Mada, Yogyakarta, Indonesia, 2025, dibimbing oleh Dr. Indriana Hidayah, S.T., M.T. dan Teguh Bharata Adji, S.T., M.T., M.Eng., Ph.D. [Online]. Available: <http://etd.repository.ugm.ac.id/>
- [31] S. Steinert, K. E. Avila, S. Ruzika, J. Kuhn, and S. Küchemann, “Harnessing large language models to develop research-based learning assistants for formative feedback,” *Smart Learning Environments*, vol. 11, no. 1, p. 62, 2024. [Online]. Available: <https://slejournal.springeropen.com/articles/10.1186/s40561-024-00354-1>

[32] “Groq API Pricing,” Groq. [Online]. Available: <https://groq.com/pricing>

Catatan: Daftar pustaka adalah apa yang dirujuk atau disitasi, bukan apa yang telah dibaca, jika tidak ada dalam sitasi maka tidak perlu dituliskan dalam daftar pustaka.

LAMPIRAN

L.1 Isi Lampiran

Lampiran bersifat opsional bergantung hasil kesepakatan dengan pembimbing dapat berupa:

1. Bukti pelaksanaan Kuesioner seperti pertanyaan kuesioner, resume jawaban responden, dan dokumentasi kuesioner.
2. Spesifikasi Aplikasi atau Sistem yang dikembangkan meliputi spesifikasi teknis aplikasi, tautan unduh aplikasi, manual penggunaan aplikasi, hingga screenshot aplikasi.
3. Cuplikan kode yang sekiranya penting dan ditambahkan.
4. Tabel yang terlalu panjang yang masih diperlukan tetapi tidak memungkinkan untuk ditayangkan di bagian utama skripsi.
5. Gambar-gambar pendukung yang tidak terlalu penting untuk ditampilkan di bagian utama. Akan tetapi, mendukung argumentasi/pengamatan/analisis.
6. Penurunan rumus-rumus atau pembuktian suatu teorema yang terlalu panjang dan terlalu teknis sehingga Anda berasumsi bahwa pembaca biasa tidak akan menelaah lebih lanjut. Hal ini digunakan untuk memberikan kesempatan bagi pembaca tingkat lanjut untuk melihat proses penurunan rumus-rumus ini.

LAMPIRAN

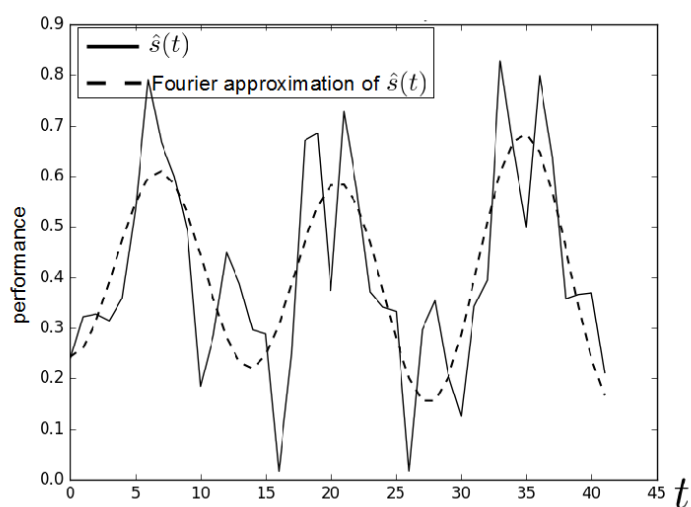
L.2 Panduan Latex

L.2.1 Syntax Dasar

L.2.1.1 Penggunaan Sitasi

Contoh penggunaan sitasi [?, ?] [?] [?] [?] [?, ?]

L.2.1.2 Penulisan Gambar



Gambar L.1. Contoh gambar.

Contoh gambar terlihat pada Gambar L.1. Gambar diambil dari [?].

L.2.1.3 Penulisan Tabel

Tabel L.1. Tabel ini

ID	Tinggi Badan (cm)	Berat Badan (kg)
A23	173	62
A25	185	78
A10	162	70

Contoh penulisan tabel bisa dilihat pada Tabel L.1.

L.2.1.4 Penulisan formula

Contoh penulisan formula

$$L_{\psi_z} = \{t_i \mid v_z(t_i) \leq \psi_z\} \quad (1)$$

Contoh penulisan secara *inline*: $PV = nRT$. Untuk kasus-kasus tertentu, kita membutuhkan perintah "mathit" dalam penulisan formula untuk menghindari adanya jeda saat penulisan formula.

Contoh formula **tanpa** menggunakan "mathit": $PVA = RTD$

Contoh formula **dengan** menggunakan "mathit": $PVA = RTD$

L.2.1.5 Contoh list

Berikut contoh penggunaan list

1. First item
2. Second item
3. Third item

L.2.2 Blok Beda Halaman

L.2.2.1 Membuat algoritma terpisah

Untuk membuat algoritma terpisah seperti pada contoh berikut, kita dapat memanfaatkan perintah *algstore* dan *algrestore* yang terdapat pada paket *algcompatible*. Pada dasarnya, kita membuat dua blok algoritma dimana blok pertama kita simpan menggunakan *algstore* dan kemudian di-restore menggunakan *algrestore* pada algoritma kedua. Perintah tersebut dimaksudkan agar terdapat kesinamungan antara kedua blok yang sejatinya adalah satu blok.

Algorithm 1 Contoh algorima

```
1: procedure CREATESET( $v$ )  
2:   Create new set containing  $v$   
3: end procedure
```

Pada blok algoritma kedua, tidak perlu ditambahkan caption dan label, karena sudah menjadi satu bagian dalam blok pertama. Pembagian algoritma menjadi dua bagian ini berguna jika kita ingin menjelaskan bagian-bagian dari sebuah algoritma, maupun untuk memisah algoritma panjang dalam beberapa halaman.

```
4: procedure CONCATSET( $v$ )  
5:   Create new set containing  $v$   
6: end procedure
```

L.2.2.2 Membuat tabel terpisah

Untuk membuat tabel panjang yang melebihi satu halaman, kita dapat mengganti kombinasi *table* + *tabular* menjadi *longtable* dengan contoh sebagai berikut.

Tabel L.2. Contoh tabel panjang

header 1	header 2
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar

L.2.2.3 Menulis formula terpisah halaman

Terkadang kita butuh untuk menuliskan rangkaian formula dalam jumlah besar sehingga melewati batas satu halaman. Solusi yang digunakan bisa saja dengan memindahkan satu blok formula tersebut pada halaman yang baru atau memisah rangkaian formula menjadi dua bagian untuk masing-masing halaman. Cara yang pertama mungkin akan menghasilkan alur yang berbeda karena ruang kosong pada halaman pertama akan diisi oleh teks selanjutnya. Sehingga di sini kita dapat memanfaatkan *align* yang sudah diatur dengan mode *allowdisplaybreaks*. Penggunaan *align* ini memungkinkan satu rangkaian formula terpisah berbeda halaman.

Contoh sederhana dapat digambarkan sebagai berikut.

$$\begin{aligned}
 x &= y^2 \\
 x &= y^3 \\
 a + b &= c \\
 x &= y - 2 \\
 a + b &= d + e \\
 x^2 + 3 &= y \\
 a(x) &= 2x
 \end{aligned}
 \tag{2}$$

$$b_i = 5x$$

$$10x^2 = 9x$$

$$2x^2 + 3x + 2 = 0$$

$$5x - 2 = 0$$

$$d = \log x$$

$$y = \sin x$$

LAMPIRAN

L.3 Format Penulisan Referensi

Penulisan referensi mengikuti aturan standar yang sudah ditentukan. Untuk internasionalisasi DTETI, maka penulisan referensi akan mengikuti standar yang ditetapkan oleh IEEE (*International Electronics and Electrical Engineers*). Aturan penulisan ini bisa diunduh di <http://www.ieee.org/documents/ieeecitationref.pdf>. Gunakan Mendeley sebagai *reference manager* dan *export* data ke format Bibtex untuk digunakan di Latex.

Berikut ini adalah sampel penulisan dalam format IEEE:

L.3.1 Book

Basic Format:

- [1] J. K. Author, "Title of chapter in the book," in Title of His Published Book, xth ed. City of Publisher, Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx-xxx.

Examples:

- [1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.
- [2] L. Stein, "Random patterns," in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.
- [3] R. L. Myer, "Parametric oscillators and nonlinear materials," in Nonlinear Optics, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.
- [4] M. Abramowitz and I. A. Stegun, Eds., Handbook of Mathematical Functions (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.
- [5] E. F. Moore, "Gedanken-experiments on sequential machines," in Automata Studies (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.
- [6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), Integrated Electronic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1970.
- [7] M. Gorkii, "Optimal design," Dokl. Akad. Nauk SSSR, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., The Mathematical Theory of Optimal Processes. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).
- [8] G. O. Young, "Synthetic structure of industrial plastics," in Plastics, vol. 3,

Polymers of Hexadromicon, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.

L.3.2 Handbook

Basic Format:

- [1] Name of Manual/Handbook, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. xx-xx.

Examples:

- [1] Transmission Systems for Communications, 3rd ed., Western Electric Co., Winston Salem, NC, 1985, pp. 44-60.
- [2] Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.
- [3] RCA Receiving Tube Manual, Radio Corp. of America, Electronic Components and Devices, Harrison, NJ, Tech. Ser. RC-23, 1992.

Conference/Prosiding

Basic Format:

- [1] J. K. Author, "Title of paper," in Unabbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp.xxx-xxx.

Examples:

- [1] J. K. Author [two authors: J. K. Author and A. N. Writer] [three or more authors: J. K. Author et al.], "Title of Article," in [Title of Conf. Record as], [copyright year] © [IEEE or applicable copyright holder of the Conference Record]. doi: [DOI number]

Sumber Online/Internet

Basic Format:

- [1] J. K. Author. (year, month day). Title (edition) [Type of medium]. Available: [http://www.\(URL\)](http://www.(URL))

Examples:

- [1] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: <http://www.atm.com>

Skripsi, Tesis dan Disertasi

Basic Format:

- [1] J. K. Author, "Title of thesis," M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

[2] J. K. Author, "Title of dissertation," Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

Examples:

[1] J. O. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993. [2] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993

LAMPIRAN

L.4 Contoh Source Code

L.4.1 Sample algorithm

Algorithm 2 Kruskal's Algorithm

```
1: procedure MAKESET( $v$ )
2:   Create new set containing  $v$ 
3: end procedure
4:
5: function FINDSET( $v$ )
6:   return a set containing  $v$ 
7: end function
8:
9: procedure UNION( $u, v$ )
10:  Unites the set that contain  $u$  and  $v$  into a new set
11: end procedure
12:
13: function KRUSKAL( $V, E, w$ )
14:   $A \leftarrow \{\}$ 
15:  for each vertex  $v$  in  $V$  do
16:    MakeSet( $v$ )
17:  end for
18:  Arrange  $E$  in increasing costs, ordered by  $w$ 
19:  for each  $(u, v)$  taken from the sorted list do
20:    if FindSet( $u$ )  $\neq$  FindSet( $v$ ) then
21:       $A \leftarrow A \cup \{(u, v)\}$ 
22:      Union( $u, v$ )
23:    end if
24:  end for
25:  return  $A$ 
26: end function
```

L.4.2 Sample Python code

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22    if M is None:
23        M = np.copy(VT)
24    else:
25        M = np.concatenate((M, VT), 1)
26
27    VT = np.zeros((n*m,1), int)
28
29    return M
```

L.4.3 Sample Matlab code

```
1 function X = BitXorMatrix(A,B)
2 %function to compute the sum without charge of two vectors
3
4 %convert elements into unsigned integers
5 A = uint8(A);
6 B = uint8(B);
7
8 m1 = length(A);
9 m2 = length(B);
10 X = uint8(zeros(m1, m2));
11 for n1=1:m1
12     for n2=1:m2
13         X(n1, n2) = bitxor(A(n1), B(n2));
14     end
15 end
```