TRANSFORMING UNIVERSITY MENTAL HEALTH SUPPORT: AN AGENTIC AI FRAMEWORK FOR PROACTIVE INTERVENTION AND RESOURCE MANAGEMENT

BACHELOR'S THESIS

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

Written by:

GIGA HIDJRIKA AURA ADKHY
21/479228/TK/52833

INFORMATION ENGINEERING PROGRAM

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
FACULTY OF ENGINEERING UNIVERSITAS GADJAH MADA
YOGYAKARTA
2025

ENDORSEMENT PAGE

TRANSFORMING UNIVERSITY MENTAL HEALTH SUPPORT: AN AGENTIC AI FRAMEWORK FOR PROACTIVE INTERVENTION AND RESOURCE MANAGEMENT

THESIS

Proposed as A Requirement to Obtain
Undergraduate Degree (*Sarjana Teknik*)
in Department of Electrical and Information Engineering
Faculty of Engineering
Universitas Gadjah Mada

Written by:

GIGA HIDJRIKA AURA ADKHY
21/479228/TK/52833

Has been approved and endorsed

on . . . . . .

Supervisor I	Supervisor II
Dr. Bimo Sunarfri Hantono, S.T., M.Eng.	**Guntur Dharma Putra, PhD**
NIP 197701312002121003	**NIP 111199104201802102**

STATEMENT

Saya yang bertanda tangan di bawah ini :

Name	: Giga Hidjrika Aura Adkhy
NIM	: 21/479228/TK/52833
Tahun terdaftar	: 2021
Program	: Bachelor's degree
Major	: Information Engineering
Faculty	: Faculty of Engineering, 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)

Giga Hidjrika Aura Adkhy
NIM 21/479228/TK/52833

PAGE OF DEDICATION

This thesis is lovingly dedicated to my parents for their endless love and support;
to my partner, Virna Amrita, for her patience and encouragement;
and to my friends, who made this journey brighter.

PREFACE

Praise be to Allah SWT for His abundant blessings, grace, and guidance, enabling the completion of this thesis. The long journey of completing this research has been filled with twists and turns, challenges, and invaluable lessons. Throughout the preparation of this thesis, I have received tremendous guidance, assistance, and support from various parties. Therefore, I would like to express my sincere gratitude to:

1. My thesis advisors, who has provided direction, guidance, and patience in steering this research to completion. Every discussion and feedback has shaped a clearer research direction amidst the complexity of ever-evolving innovation.
2. My beloved parents, who have provided financial support and endless prayers throughout my education at Universitas Gadjah Mada. Without their sacrifices and trust, this achievement would never have been realized.
3. My partner, Virna Amrita who has always accompanied me through Discord during long nights of struggle, providing encouragement when motivation began to fade, and being a source of comfort in times of joy and hardship. Your presence has been a light in the darkness, especially when facing challenges from the volatility of the crypto world that influenced this research journey.
4. My siblings, who have consistently prayed for and supported every step of my academic journey.
5. My close friends–Azfar, Ariq, Zakong, Diamond, Arif, Ditya, Nando, Aufa, Difta, Akhdan, Evan, Aji and others who have been companions in arms during college, sharing joys and sorrows, and serving as an incredible support system. You are my second family who made my days on campus more meaningful.
6. PT INA17, who has shown interest in and provided support for this project, validating that the research conducted has real applicative value in the industry.
7. The Sumbu Labs team–Maulana, Dzikran, Farhan, Azfar, and Virna–who have helped me in working through one of the biggest projects of my life while doing this thesis in parallel. Our collaboration in developing CAR-dano (now Ototentik) has been an unforgettable experience. You are not just colleagues, but partners in making dreams come true.
8. All parties involved in the EDU Chain Hackathon where the UGM-AICare project successfully secured funding of 6000 USD. This achievement is proof that hard work and innovation can be rewarded, even though it sometimes became a beautiful "distraction" from the focus of thesis writing.

The journey of completing this thesis has taught me that innovation does not always follow a straight path. There are times when we are tempted to branch out, explore new ideas, and even get "lost" in hackathon after hackathon. However, each of these experiences has enriched my understanding and broadened my perspective on how technology can make a real impact on society. There were days when I felt lonely working from home, but the support from my loved ones made every challenge feel lighter.

The motivation behind choosing AI agents as the focus of my bachelor's thesis stems from a deeply personal mission: to elevate the standard of mental health services at UGM. Throughout my time as a student, I witnessed firsthand, both in myself and in my peers, how difficult it is to seek help for mental health concerns. We are often too busy, or we simply fail to prioritize our mental wellbeing until it becomes critical. Many students struggle in silence, not because help isn't available, but because the barriers to access feel too high. This realization drove me to create Aika, the AI agent in UGM-AICare, designed to provide proactive interventions and regular check-ups that meet students where they are, when they need it most.

This vision was significant enough for me to embrace the ambitious scope of this work, even knowing it would take longer to complete than a typical bachelor's thesis. I only wish the best for UGM, just as my parents and friends have always wished the best for me. This university has been the place where I met remarkable people who humbled me, challenged my perspectives, and grounded me in reality. It shaped not just my academic journey, but my character and values. If this research can contribute to making mental health support more accessible and effective for future generations of UGM students, then every late night, every challenge, and every moment of uncertainty will have been worth it.

Finally, I hope that this thesis can contribute to the advancement of knowledge, particularly in the fields of artificial intelligence and healthcare technology, and can serve as inspiration for future research. May this work bring benefits to us all, aamiin.

Yogyakarta, November 12, 2025

Giga Hidjrika Aura Adkhy

CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE AND ABBREVIATION

[SAMPLE]

b	=	bias
$K(x_i, x_j)$	=	fungsi kernel
y	=	kelas keluaran
C	=	parameter untuk mengendalaikan besarnya pertukaran antara penalti variabel slack dengan ukuran margin
L_D	=	persamaan Lagrange dual
L_P	=	persamaan Lagrange primal
w	=	vektor bobot
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

Layanan kesejahteraan mahasiswa di perguruan tinggi masih banyak bersifat reaktif dan sering terlambat menjangkau mereka yang membutuhkan. Skripsi ini merancang dan mengevaluasi sebuah kerangka multi-agen AI yang berorientasi keselamatan untuk mendukung layanan yang lebih proaktif dan terukur dengan tetap melibatkan manusia. Artifak inti, *Safety Agent Suite*, terdiri dari lima komponen: (i) **Aika Meta-Agent** yang menyediakan orkestrasi terpadu dengan penyaringan risiko langsung Tingkat 1 yang terintegrasi, (ii) **Safety Triage Agent** untuk analisis risiko menyeluruh Tingkat 2 di tingkat percakapan, (iii) **Therapeutic Coach Agent** yang memberikan intervensi terapeutik singkat berbasis CBT, (iv) **Case Management Agent** untuk koordinasi kasus klinis dan tindak lanjut operasional, dan (v) **Insights Agent** untuk analitik agregat yang menjaga privasi. Arsitektur pemantauan risiko dua tingkat (penyaringan per-pesan Aika + analisis percakapan STA) mencapai pengurangan biaya API LLM sebesar 45-60% sambil meningkatkan akurasi klinis melalui penilaian kontekstual holistik. Sistem multi-agen dibangun dengan LangGraph dan mencakup perlindungan untuk penggunaan alat, redaksi, dan kemampuan audit.

Kami membangun prototipe fungsional dalam platform UGM-AICare dan melakukan evaluasi berbasis skenario yang menitikberatkan secara eksklusif pada kinerja arsitektur agen: sensitivitas/spesifisitas triase pada skenario krisis sintetis; keandalan orkestrasi melalui tingkat keberhasilan pemanggilan fungsi dan transisi state; latensi ujung-ke-ujung; ketahanan terhadap *prompt injection*; serta kualitas coaching yang dinilai buta menggunakan rubrik kepatuhan CBT. **Skripsi ini berfokus secara spesifik pada desain dan evaluasi kerangka multi-agen itu sendiri**—agen spesialis berbasis BDI, lapisan orkestrasi Aika, dan perilaku kolektif mereka dalam konteks percakapan kritis keselamatan. Desain basis data, komponen antarmuka pengguna, dan infrastruktur deployment didokumentasikan sebagai konteks implementasi namun bukan subjek evaluasi formal. Hasil menunjukkan kelayakan orkestrasi agen yang andal dengan latensi terkendali dan moda kegagalan yang dapat dipantau di bawah pengawasan manusia. Kami membahas pertimbangan etis, prinsip *privacy by design*, keterbatasan penelitian, dan kebutuhan studi klinis lapangan di masa depan dengan pengguna riil.

Kata kunci: Sistem Multi-Agen; Arsitektur BDI; Orkestrasi Agen; Triase Keselamatan; LangGraph; Human-in-the-Loop; Kesejahteraan Mahasiswa; Evaluasi Berbasis Skenario

ABSTRACT

Higher Education Institutions face rising demand for student well-being support while operating largely reactive, high-friction service models. This thesis proposes and evaluates a safety-oriented, multi-agent AI framework that coordinates specialized agents to enable proactive, scalable support under human oversight. The core artifact, the Safety Agent Suite, comprises: (i) an Aika Meta-Agent providing unified orchestration with integrated Tier 1 immediate risk screening, (ii) a Safety Triage Agent for Tier 2 comprehensive conversation-level risk analysis, (iii) a Therapeutic Coach Agent delivering CBT-based therapeutic micro-interventions, (iv) a Case Management Agent for clinical case coordination and operational follow-ups, and (v) an Insights Agent for privacy-preserving aggregate analytics. The two-tier risk monitoring architecture (Aika's per-message screening + STA's conversation analysis) achieves 45-60% reduction in LLM API costs while improving clinical accuracy through holistic contextual assessment. The multi-agent system is built with LangGraph and includes guardrails for tool use, redaction, and auditability.

We implement a functional prototype within the UGM-AICare platform and conduct scenario-based evaluations focused exclusively on agent architecture performance: triage sensitivity/specificity on synthetic crisis scenarios; orchestration reliability via tool-call success and state transition behavior; end-to-end latency; robustness against prompt-injection; and coaching quality via CBT adherence rubrics with blinded human ratings. **This thesis focuses specifically on the design and evaluation of the multi-agent framework itself**—the BDI-based specialist agents, Aika orchestration layer, and their collective behavior in safety-critical conversational contexts. Database design, user interface components, and deployment infrastructure are documented as implementation context but are not subjects of formal evaluation. Results demonstrate the feasibility of reliable agent orchestration with bounded latency and controllable failure modes under human-in-the-loop supervision. We discuss ethical considerations, privacy by design principles, research limitations, and outline requirements for future clinical field studies with real users.

Keywords: Multi-Agent Systems; BDI Architecture; Agent Orchestration; Safety Triage; LangGraph; Human-in-the-Loop; Student Well-being; Scenario-Based Evaluation

CHAPTER I

INTRODUCTION

1.1 Background

Higher Education Institutions (HEIs) are facing a critical and growing challenge in supporting student well-being [2,3]. A landmark report highlights the escalating prevalence of mental health and substance use issues among student populations, urging institutions to adopt a more comprehensive support model [4]. This crisis not only jeopardizes students' academic success and personal development but also places an immense, unsustainable strain on the institutions tasked with supporting them. Recent global surveys indicate that nearly 42% of university students meet the criteria for at least one mental health disorder, while the average counselor-to-student ratio in higher education remains around 1:1,500, well above recommended levels for effective service delivery [5, 6].

The traditional support model, centered around on-campus counseling services, is fundamentally **reactive**. It relies on students to self-identify their distress and navigate the process of seeking help. This paradigm faces significant operational challenges, including insufficient staffing, long waiting lists, and an inability to provide immediate, 24/7 support, which ultimately limits access for a large portion of the student body [7]. Consequently, a critical gap persists between the need for mental health services and their actual provision, leaving many students without timely support [8].

To bridge this gap, a paradigm shift from a reactive to a **proactive** support model is imperative [8]. The engine for this evolution is **Digital Transformation**, a process that leverages technology to fundamentally reshape organizational processes and enhance value delivery within HEIs [9]. Within this context, Artificial Intelligence (AI) has emerged as a key enabling technology, with systematic reviews confirming its significant potential to analyze complex data, automate processes, and deliver personalized interventions at scale within the higher education landscape [10, 11].

However, most existing AI applications in university mental health remain limited to passive chatbots or predictive dashboards that, while insightful, depend on human operators to interpret and act upon their outputs, a limitation widely recognized as the *insight-to-action gap* [12, 13]. This thesis argues that overcoming this gap requires a more autonomous paradigm, in which AI systems do not merely predict or inform but can proactively decide and act.

This research therefore moves beyond conventional AI applications by proposing the use of **Agentic AI**. An intelligent agent is an autonomous system capable of perception, decision-making, and proactive action to achieve specific goals [14,15], representing a new frontier in educational technology [16]. We propose that a framework built upon a system of collaborative intelligent agents, a **Multi-Agent System (MAS)**, can create a truly transformative ecosystem. Such a system would not only serve as a support tool for students but, more importantly, would function as a strategic asset for the institution, enabling data-driven decision-making, automating operational workflows, and facilitating a proactive stance on student well-being.

This framework is prototyped within the **UGM-AICare Project**, a collaborative university research initiative focused on developing AI-driven mental health and well-being tools for the Universitas Gadjah Mada (UGM) community. The project serves as the practical testbed for validating the proposed agentic system in a real institutional context.

To clarify the paradigm shift this research proposes, Table 1.1 presents a systematic comparison of three mental health support models: traditional in-person counseling, reactive AI chatbots, and the proposed proactive multi-agent framework. This comparison reveals that both traditional and chatbot-based approaches share a fundamental limitation, they are **reactive systems that depend on student-initiated help-seeking behavior**. The proposed framework addresses this limitation through continuous monitoring, automated risk detection, and proactive intervention while maintaining human oversight for safety-critical decisions.

The critical insight from this comparison is that technological advancement alone (moving from in-person to chatbot) does not address the fundamental barrier: **vulnerable students who need help most are precisely those least likely to initiate contact** [?, ?]. This research hypothesizes that closing this gap requires a paradigm shift from reactive to proactive support, operationalized through autonomous agent-based monitoring and intervention.

1.2 Problem Formulation

The inefficiency and reactive nature of current university mental health support systems present a complex problem. To move towards a proactive and scalable model, this research addresses the following core challenges:

1. The primary challenge is the **design of a cohesive, safety-oriented agentic AI framework** capable of automating key institutional processes. This requires a shift from a monolithic chatbot to a multi-agent system where specialized agents handle distinct tasks, including real-time crisis detection, personalized coaching, clinical case management, and privacy-preserving analytics.
2. The technical challenge of **orchestrating a heterogeneous multi-agent system** in a robust, scalable, and secure cloud-native architecture. This involves managing state-

Table 1.1. Comparison of mental health support paradigms: Traditional, chatbot, and proposed proactive multi-agent systems.

Characteristic	**Traditional In-Person Counseling**	**Reactive AI Chatbots**	**Proposed Multi-Agent Framework (UGM-AICare)**
Initiation Model	Student must self-refer and schedule appointment	Student must open app and initiate conversation	Continuous monitoring with automated outreach capability; system-initiated intervention
Availability	Limited office hours (typically 9am-5pm); multi-week waitlists common	24/7 availability; instant response	24/7 availability with proactive intervention triggers; automated escalation protocols
Scalability	Constrained by counselor-to-student ratio (1:1500 average); unsustainable at scale	Scales to unlimited concurrent users	Scales through automated triage and routing; human oversight reserved for critical cases
Data Utilization	Manual case notes; no population-level trend analysis	Individual conversation logs; limited cross-user insights	Population-level analytics with privacy-preserving aggregation; automated intervention routing based on trends
Intervention Timing	**After crisis escalates** (reactive: student seeks help post-crisis)	**After student reaches out** (reactive: depends on user initiation)	**Before crisis peaks** (proactive: automated risk detection triggers early intervention)
Administrative Integration	Manual case management; human-dependent scheduling and follow-up workflows	No administrative integration; standalone conversational interface	Automated case creation, appointment scheduling, resource allocation, and counselor notification
Key Limitation	**Relies entirely on student help-seeking behavior**; barriers include stigma, lack of awareness, symptom-induced apathy	**Still requires student to initiate contact**; does not reach students who avoid seeking help	Requires validation through controlled testing before clinical deployment; performance not yet validated on live student populations
Human Oversight	Direct human delivery of all services	Minimal oversight; no clinical escalation path	Human-in-the-loop for all critical decisions; automated triage with mandatory counselor review

ful, long-running interactions and ensuring reliable communication between agents powered by external, non-deterministic LLMs.

3. The methodological challenge of **validating the framework's functional capabilities and potential for impact in the absence of a full-scale clinical trial**. This requires developing meaningful, scenario-based testing protocols that can effectively demonstrate the agentic workflows and their advantages over static systems.

To address these challenges, this thesis proposes and details the **Safety Agent Suite**, a framework comprised of four specialized, collaborative intelligent agents: a **Safety Triage Agent (STA)**, a **Therapeutic Coach Agent (TCA)**, a **Case Management Agent (CMA)**, and an **Insights Agent (IA)**, coordinated through an **Aika Meta-Agent** (orchestrator) that provides unified, role-based orchestration and ensures coherent, safety-first interactions across all user roles.

1.3 Objectives

The primary objectives of this thesis are:

1. To design an agentic AI framework, grounded in the BDI model of rational agency, that systematically bridges the 'insight-to-action' gap in institutional mental health support.
2. To implement a functional proof-of-concept prototype, the 'Safety Agent Suite,' demonstrating the orchestration of specialized agents (triage, coaching, service desk, insights) and a meta-agent coordinator using LangGraph.
3. To evaluate the prototype's core agentic workflows through scenario-based testing, validating its capacity for proactive intervention and automated administrative action.

1.4 Research Questions

To keep the scope concrete and measurable, this thesis addresses the following research questions (RQs):

1. **RQ1 (Proactive Safety):** Can the agentic framework reliably distinguish between crisis and non-crisis user states to trigger a timely and appropriate safety protocol?
2. **RQ2 (Functional Correctness):** Does the multi-agent framework correctly execute its core automated workflows, such as routing users to the appropriate specialized agent and invoking necessary tools?
3. **RQ3 (Output Quality & Privacy):** Can the framework generate outputs (coaching advice, institutional insights) that are both appropriate for their purpose and compliant with privacy-preserving principles?

These questions directly inform the evaluation in Chapter IV through scenario-based tests and transparent metrics (e.g., sensitivity, workflow success rate, rubric scores), with human oversight preserved for safety-critical cases.

1.5 Scope and Limitations

To ensure the feasibility and focus of this bachelor's thesis, the following boundaries are explicitly established:

1. **Focus on Multi-Agent Architecture Only:** This research is focused exclusively on the **design, implementation, and evaluation of the multi-agent AI framework itself**, the Safety Agent Suite's BDI-based specialist agents, the Aika Meta-Agent orchestration layer, and their collective behavior in safety-critical conversational scenarios. The full UGM-AICare implementation includes database schema design, user interface components, blockchain token systems, and deployment infrastructure; however, **these system components are documented as implementation context but are not subjects of formal evaluation in this work**.
2. **Proof-of-Concept Evaluation Scope:** The evaluation adopts a **proof-of-concept validation approach** appropriate for bachelor's-level Design Science Research. The objective is to demonstrate **technical feasibility** that the Safety Agent Suite can execute core workflows correctly under controlled conditions. Evaluation uses modest sample sizes: 50 crisis scenarios for safety triage (RQ1), 10 conversation flows for orchestration (RQ2), 10 coaching scenarios for response quality (RQ3), and code review with unit tests for privacy validation (RQ4). This approach validates architectural correctness without requiring extensive data collection infrastructure, consistent with DSR artifact evaluation conventions where initial validation focuses on demonstrating capability rather than exhaustive performance characterization.
3. **Simulated Data for Privacy and Feasibility:** All testing utilizes **synthetically generated student mental health crisis scenarios and simulated conversation patterns** created using GPT-4 and Claude 3.5 Sonnet, not real user data. This approach is necessary to protect privacy during development and to enable controlled evaluation without requiring human subjects approval. However, it means that agent performance has not been validated on the specific linguistic diversity, cultural contexts, and edge cases of a live Indonesian student population. Ground truth labels for synthetic scenarios are provided by the primary researcher with peer validation, acknowledging that clinical expert validation remains future work.
4. **Single-Rater Assessment with AI Validation:** Response quality evaluation (RQ3) is conducted by the primary researcher using a structured rubric, with GPT-4 performing independent validation on the same responses to provide a reference point

for consistency. This pragmatic approach demonstrates the evaluation methodology while acknowledging that inter-rater reliability analysis with multiple clinical experts and formal therapeutic quality assessment using validated instruments (e.g., Cognitive Therapy Scale) remain future work appropriate for clinical validation studies.

5. **Privacy-Aware Design Without Formal Proofs:** This research implements k-anonymity enforcement ($k \geq 5$) with code-level verification and unit testing to validate privacy safeguards function as designed. This demonstrates **privacy-aware agent behavior** and implementation correctness within the prototype context. However, it does not pursue full differential privacy proofs, formal threat modeling using frameworks like LINDDUN, or cryptographic verification—activities appropriate for production security audits but beyond bachelor's thesis scope.

6. **Technical Feasibility, Not Clinical Efficacy:** This evaluation demonstrates that the proposed multi-agent architecture is *technically feasible*. The agents can classify crises, orchestrate workflows, generate appropriate responses, and enforce privacy thresholds under controlled conditions. It does **not** claim to have validated clinical efficacy (long-term mental health outcome improvement), cultural appropriateness for Indonesian students, operational sustainability, or production-readiness for deployment without further testing. Such claims would require ethics approval, multi-rater expert evaluation, field pilots with real users, longitudinal outcome measurement, and cost-benefit analysis, activities beyond bachelor's thesis scope but identified as critical future work in Chapter IV, Section 4.9.

1.6 Contributions

This thesis contributes a focused blueprint and evidence base for safety-oriented agentic support:

1. **Safety pipeline specification**. A concrete guideline for triage and escalation: risk cues and scoring, guardrails and redaction steps, decision thresholds, human-in-the-loop invariants, and service targets such as time-to-escalation.

2. **Agent orchestration design**. A LangGraph view of the Safety Agent Suite—nodes, edges, and typed state schemas—plus the supporting tool-use protocol (validated schemas, idempotency, retry/backoff) that keeps workflows predictable.

3. **Evaluation assets and findings**. Scenario-based tests (synthetic crisis set, adversarial prompts, blinded coaching rubric) and their results, covering safety sensitivity, orchestration reliability, latency, and coaching quality under human oversight.

1.7 Thesis Outline

The structure of this thesis is outlined as follows:

Chapter I: Introduction. This chapter elaborates on the background of the study, the justification for the research's significance, the problem formulation to be addressed, and the specific objectives to be achieved. It also defines the scope and limitations of the research, outlines the expected contributions, and presents the overall organizational structure of the thesis report.

Chapter II: Literature Review and Theoretical Framework. This chapter surveys prior work on agentic and conversational AI for mental health, safety-critical triage systems, human-in-the-loop design, and privacy-aware analytics. It establishes the theoretical foundation that underpins the core concepts and technologies utilized in this research.

Chapter III: System Design and Architecture. This chapter outlines the methodology and technical blueprint for the system. It explains the adoption of Design Science Research and presents the system's high-level conceptual architecture, focusing on the five components of the **Safety Agent Suite**: four specialized agents (STA, TCA, CMA, IA) and the Aika Meta-Agent orchestrator. It details the underlying cloud-native technical architecture, justifying the chosen technology stack, including the use of **LangGraph** for agent orchestration and a **FastAPI** backend for the core application logic. It also describes the database structure, user interface design, and integrated security and privacy measures like differential privacy.

Chapter IV: Implementation and Evaluation. This chapter describes the development and testing of the system prototype. This chapter details the technical environment used for implementation and demonstrates the functional prototype that was built. It then explains the testing process used to evaluate the system's performance against its design requirements. The chapter concludes by presenting the results from these tests and providing an analysis of the findings.

Chapter V: Conclusion and Future Work. This chapter summarizes the study's findings and contributions. This chapter revisits the initial research problems and presents the main conclusions drawn from the research. It concludes by offering recommendations for both the future development of the system and for subsequent research in this area.

CHAPTER II
LITERATURE REVIEW AND THEORETICAL BACKGROUND

This chapter establishes the academic context for the research. It begins by surveying the existing literature on AI applications in mental health and student support to identify the limitations of current approaches. It then details the theoretical framework and enabling technologies that provide the foundation for the proposed solution. Finally, it synthesizes these areas to formally identify the research gap this thesis addresses.

2.1 Literature Review: The Landscape of AI in University Mental Health Support

This review surveys existing research at the intersection of artificial intelligence, institutional support systems, and student mental health. The aim is to contextualize the present work by examining the evolution and limitations of current approaches, thereby setting the stage for the introduction of a more advanced, agentic framework.

2.1.1 Conversational Agents for Mental Health Support

The application of conversational agents in mental health has evolved significantly, from early experiments in simulating dialogue to sophisticated, evidence-based therapeutic tools. This evolution reveals both the immense potential of these technologies and the persistent operational limitations that motivate the current research.

2.1.1.1 Evolution from Rule-Based Systems to LLM-Powered Agents

The concept of using a computer program for therapeutic dialogue dates back to Weizenbaum's ELIZA (1966), a system that used simple keyword matching and canned response templates to mimic a Rogerian psychotherapist [17, 18]. While a landmark in human-computer interaction, ELIZA and subsequent rule-based systems lacked any true semantic understanding, memory, or capacity for evidence-based intervention. Their primary limitation was their inability to move beyond superficial pattern recognition, leading to brittle and often nonsensical conversations when faced with inputs outside their predefined rules [17].

The advent of Large Language Models (LLMs) has catalyzed a paradigm shift. Modern conversational agents, powered by Transformer architectures, can generate fluent, empathetic, and context-aware responses. These models are pre-trained on vast text corpora, enabling them to understand linguistic nuance and generate human-like text. This has allowed for the development of agents that can engage in more meaningful, multi-turn conversations, moving beyond simple question-answering to provide more substantive support [18].

2.1.1.2 Therapeutic Applications and Efficacy

Contemporary mental health chatbots leverage LLMs to deliver a range of evidence-based interventions. A primary application is the delivery of psychoeducation and structured exercises from therapeutic modalities like Cognitive Behavioral Therapy (CBT). Systems such as Woebot have been the subject of randomized controlled trials (RCTs), which have demonstrated their efficacy in reducing symptoms of depression and anxiety among university students by delivering daily, brief, conversational CBT exercises [19, 20]. Other platforms, like Tess, have shown similar positive outcomes by providing on-demand emotional support and coping strategies.

These tools offer several key advantages:

- **Accessibility and Scalability:** They are available 24/7, overcoming the time and resource constraints of traditional human-led services.
- **Anonymity:** They provide a non-judgmental and anonymous space for users to disclose their feelings, which can lower the barrier for individuals who fear stigma [21].

2.1.1.3 The Dominant Reactive Paradigm and Its Limitations

Despite their technological sophistication and therapeutic potential, the fundamental operational model of modern mental health applications remains overwhelmingly **reactive**. This model, common in service design, operates on a "break-fix" basis, where service delivery is initiated only after a user—in this case, a student—self-identifies a problem and actively seeks a solution [22]. They are designed as standalone tools that depend on the student to possess the self-awareness to recognize their distress, the motivation to seek help, and the knowledge of the tool's existence.

Critically, this limitation is not unique to technology; **the traditional, in-person counseling model is equally reactive**. The standard university mental health service operates on an appointment-based system where students must: (1) recognize their own distress, (2) navigate the institutional referral process, (3) schedule an appointment (often facing multi-week waitlists), and (4) attend the session during limited office hours [6, 7].

This places the entire burden of initiation on the student, creating the same fundamental barrier across both technological and traditional systems: **it assumes students will self-identify their distress and actively seek help**. Research demonstrates that this assumption is systematically violated. Stigma, lack of mental health literacy, and a desire for self-reliance all contribute to low help-seeking rates [23, 24]. More critically, the very symptoms of conditions like depression—including anhedonia, executive dysfunction, and social withdrawal—actively impair the cognitive and motivational capacities required to initiate help-seeking behavior [?, 25].

Therefore, both traditional and chatbot-based reactive models fail to serve the most vulnerable population: those who are in distress but do not initiate contact. A student experiencing suicidal ideation may lack the energy to schedule an appointment; a student with severe social anxiety may find the act of reaching out to be itself insurmountably distressing. This thesis proposes that a solution requires a **paradigm shift to a proactive support model** that aims to anticipate needs and intervene before a problem escalates. Drawing from principles in preventative healthcare and proactive customer relationship management, this model uses data to identify patterns and risk factors, enabling the institution to offer timely, relevant support to at-risk cohorts [26, 27]. By continuously analyzing interaction patterns and employing automated risk detection, the proposed multi-agent framework can identify students in distress and initiate supportive contact *before* they reach a crisis threshold, thereby addressing the systemic failure of all reactive support models.

2.1.2 Data Analytics for Proactive Student Support

Parallel to the development of conversational AI, the field of higher education has seen a rise in the use of data analytics to support student success. This section reviews the evolution of these analytical approaches, from established learning analytics to the more nascent field of well-being analytics, and identifies the key limitations that motivate the design of the agents.

2.1.2.1 Learning Analytics for Academic Intervention

The domain of **Learning Analytics** is well-established and focuses on the "measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" [28]. Typically, these systems analyze data from institutional sources such as the Learning Management System (LMS), student information systems, and library databases. By modeling variables like assignment submission times, forum participation, and grades, institutions can build predictive models to identify students at high risk of academic failure or dropout [29]. These systems have proven effective in enabling timely academic interventions, such as targeted tutoring or advisor outreach, thereby improving student retention and success rates.

2.1.2.2 The Challenge of Well-being Analytics

More recently, researchers have attempted to extend the principles of learning analytics to the more complex and sensitive domain of student well-being. The goal is to create early-warning systems by identifying behavioral proxies for mental distress. Studies have explored the use of non-academic data sources, such as campus card usage for building access, meal plan data, and social event attendance, to find correlations with well-being outcomes [30]. For example, a sudden decrease in social activity or irregular campus attendance could be interpreted as a potential indicator of withdrawal or depression.

However, this approach is fraught with significant theoretical and practical challenges. Firstly, the "signal-to-noise" ratio is extremely low; the link between such indirect behavioral data and a student's internal mental state is often weak, correlational, and highly prone to misinterpretation [31]. A student may miss meals for many reasons other than depression. Secondly, these methods raise profound ethical questions regarding student privacy and surveillance, as they involve monitoring non-academic aspects of student life, often without explicit, ongoing consent for this specific purpose [30, 31].

A more direct, and arguably more ethical, source of data is the language students use when interacting with university services. The text from chat logs, when properly anonymized, provides a direct window into student concerns. The application of sentiment analysis and topic modeling to this textual data can yield far more reliable insights into the specific stressors affecting the student population at any given time. This approach, which is central to the design of the Analytics Agent, shifts the focus from inferring mental state from indirect behaviors to directly analyzing the expressed concerns of the student body [30].

2.1.2.3 The Insight-to-Action Gap

Whether based on academic, behavioral, or textual data, a critical limitation plagues nearly all current analytical systems in higher education: the **insight-to-action gap** [12]. The output of these systems is almost universally a dashboard, a report, or an alert delivered to a human administrator (e.g., a counselor, dean, or advisor) [13]. This administrator must then manually interpret the data, decide on an appropriate intervention strategy, and execute it.

This manual process creates a severe bottleneck that fundamentally limits the scalability, speed, and personalization of any proactive effort [32]. An administrator may be able to respond to a handful of individual alerts, but they cannot manually orchestrate a personalized outreach campaign to hundreds of students who may be exhibiting early signs of exam-related stress identified by a topic model. The manual-execution step prevents the institution from fully capitalizing on the proactive insights generated by its analytical systems. It is this specific gap that the proposed **agentic framework** is designed to close. By having the Aika Meta-Agent orchestrate the proactive generation of therapeutic plans (via the TCA) and automated administrative workflows (via the CMA), the system automates the link between data-driven insight and scalable, targeted action.

2.2 Theoretical Background

To address the limitations of reactive, disconnected support systems, a new architectural approach is required. This section details the theoretical framework and enabling technologies that provide the foundation for the proposed agentic AI system. These concepts are presented as the necessary components to build a proactive, integrated, and autonomous solution.

2.2.1 Foundational Principles of the Framework

Beyond the technical architecture, the proposed framework is grounded in several key strategic and ethical principles that justify its design and purpose. These concepts from service design, management science, and data ethics provide the theoretical motivation for shifting how institutional support is delivered.

2.2.1.1 Data-Driven Decision-Making in Higher Education

The concept of **Data-Driven Decision-Making (DDDM)** posits that strategic decisions should be based on objective data analysis and interpretation rather than solely on intuition or tradition [26, 27]. In higher education, this has manifested as the field of learning analytics, where student data is used to improve learning outcomes and retention. This framework extends that principle to student well-being. The **Insights Agent** is the core enabler of DDDM for the university's support services. By autonomously processing anonymized interaction data to identify trends, sentiment shifts, and emerging topics of concern, it provides administrators with actionable, empirical evidence. This allows the institution to move beyond anecdotal evidence and allocate resources, such as workshops, counselors, or targeted information campaigns, to where they are most needed, thereby optimizing the efficiency and impact of its support ecosystem [33].

2.2.2 Agentic AI and Multi-Agent Systems (MAS)

The paradigm of Artificial Intelligence (AI) has evolved significantly from systems that perform singular, reactive tasks to those that exhibit autonomous, proactive, and social behaviors. A cornerstone of this evolution is the concept of an **intelligent agent**. An agent is not merely a program; it is a persistent computational entity with a degree of autonomy, situated within an environment, which it can both perceive and upon which it can act to achieve a set of goals or design objectives [34]. The defining characteristic of an agent is its **autonomy**, its capacity to operate independently, making decisions and initiating actions without direct, constant human intervention. This is distinct from traditional objects, which are defined by their methods and attributes but do not exhibit control over their own behavior [15].

To operationalize this concept, this thesis formally introduces a framework built upon four distinct, specialized intelligent agents that form the **Safety Agent Suite**, coordinated by a unified orchestration layer. **Critically, the Aika Meta-Agent is the sole user-facing component**—all user interactions occur exclusively through Aika's conversational interface, which internally orchestrates specialist agent invocations as needed. This design ensures a consistent user experience while enabling modular, specialized intelligence. Each specialist agent operates transparently in the background, invoked conditionally based on user role and intent. Together they form the core of the proposed proactive support system. The framework components are:

- The **Aika Meta-Agent**, responsible for: (1) serving as the sole user-facing conversational interface for all stakeholders, (2) performing immediate Tier 1 risk screening via structured JSON responses from Gemini API, (3) context-aware routing to specialist agents based on user role and intent classification, (4) synthesizing specialist outputs into coherent, role-appropriate conversational responses, and (5) role-based access control enforcement.
- The **Safety Triage Agent (STA)**, operating in the background to perform comprehensive conversation-level risk analysis (Tier 2) at conversation end, identifying cumulative risk patterns and recommending proactive follow-up interventions.
- The **Therapeutic Coach Agent (TCA)**, operating entirely in the background to generate personalized, evidence-based CBT intervention plans and coping strategies that students access asynchronously via their dashboard. TCA does not participate in real-time conversations.
- The **Case Management Agent (CMA)**, invoked conditionally through Aika when: (1) immediate crisis escalation is required (high/critical risk detected), (2) students/staff request appointment scheduling, or (3) counselors initiate case management workflows. CMA handles clinical case workflows, counselor assignment, and SLA tracking.
- The **Insights Agent (IA)**, operating in the background for scheduled analytics, but invocable on-demand through Aika when administrators/counselors request analytics queries (e.g., "show trending topics," "case statistics for November"). IA performs privacy-preserving data analysis and trend identification on anonymized conversation data.

The theoretical underpinnings of these agents' architecture and behavior are drawn from established models of rational agency and multi-agent systems, as detailed below.

Fundamentally, an agent's operation is defined by a continuous cycle of perception, reasoning (or deliberation), and action. It perceives its environment through virtual **sensors** (e.g., data feeds, API calls, database queries) and influences that environment through its **actuators** (e.g., sending emails, generating reports, invoking other services) [35]. A prominent and highly relevant architecture for designing such goal- oriented agents is the **Belief-Desire-Intention (BDI)** model [35,36]. This model provides a framework for rational agency that mirrors human practical reasoning:

- **Beliefs:** This represents the informational state of the agent, its knowledge about the environment, which may be incomplete or incorrect. For the **Insights Agent**, beliefs correspond to the current understanding of student well-being trends derived from anonymized data.
- **Desires:** These are the motivational states of the agent, representing the objectives or goals it is designed to achieve. Desires can be seen as the potential tasks the agent could undertake, such as the **Support Coach Agent's** overarching goal to "deliver personalized coaching."
- **Intentions:** This represents the agent's commitment to a specific plan or course of action. An intention is a desire that the agent has chosen to actively pursue. For instance, the **Safety Triage Agent**, upon identifying a high-severity conversation, forms an intention to immediately route the user to emergency resources.

The BDI framework allows for the design of agents that are not merely reactive but are proactive and deliberative, capable of reasoning about how to best achieve their goals given their current beliefs about the world [15,36].

To formally ground the proposed framework in this established model, the roles and logic of each of the five framework components (four specialist agents plus the orchestrating meta-agent) are mapped to the BDI components in Table 2.1. This mapping clarifies how each component perceives its environment, formulates its objectives, and decides on a concrete course of action, allowing for the design of agents that are not merely reactive but are proactive and deliberative, capable of reasoning about how to best achieve their goals given their current beliefs about the world.

Formalization of the BDI Cycle The BDI model operates through continuous state updates that govern how agents perceive, deliberate, and act. The cycle can be formalized as:

Belief Update: An agent's beliefs are updated as new information is perceived:

$$Bel_{t+1} = Bel_t \cup \{\text{percept}_t\} \setminus \{\text{expired beliefs}\} \tag{2-1}$$

where percept_t represents new observations from the environment, and expired beliefs are those that are no longer valid or relevant.

Table 2.1. Mapping of the Agentic Framework to the BDI Model.

Agent	**Beliefs** *(Informational State)*	**Desires** *(Motivational Goals)*	**Intentions** *(Committed Plans)*
STA	• User's conversation history • Severity classification model • Emergency resources directory	• Assess immediate risk level • Provide appropriate support	• Escalate high-severity cases • Display emergency contacts
TCA	• User goals & history • Evidence-based intervention library (CBT)	• Deliver personalized coaching • Guide through exercises	• Deliver specific CBT exercise • Provide empathetic responses
CMA	• Clinical case status • Counselor availability • User appointment requests	• Manage case workflows • Schedule appointments	• Find available appointment slots • Create and update case notes
IA	• Anonymized conversation database • Last report timestamp • Known topic models	• Identify emerging trends • Quantify sentiment shifts	• Generate weekly summary reports • Execute database queries
Aika Meta-Agent	• User role and authentication context (student/counselor/admin). • Conversation history and session state across all agents. • Routing policies and agent capability mappings. • Current risk assessment from STA (if applicable).	• To provide a unified, role-appropriate interface for all users. • To ensure safety-first routing for all student interactions. • To coordinate multi-agent workflows seamlessly.	• Upon receiving a user message, form an intention to classify intent and route to appropriate specialist(s). • To synthesize specialist responses with role-consistent personality. • To maintain conversational coherence across agent transitions.

Desire Selection: The agent filters potential goals based on its current beliefs:

$$Des_t = \text{filter}(\text{Options}_t, Bel_t) \tag{2-2}$$

where Options_t represents all possible goals the agent could pursue, and the filter function selects those that are feasible given current beliefs.

Intention Formation: The agent commits to a specific plan of action through deliberation:

$$Int_t = \text{deliberate}(Des_t, Bel_t, Int_{t-1}) \tag{2-3}$$

where the deliberation process considers current desires, beliefs, and previous intentions to form a committed plan.

For example, in the **Safety Triage Agent (STA)**, percept_t is the incoming student message M_t, beliefs include prior conversation context and crisis patterns, desires map to intervention goals (de-escalation, resource connection), and intentions become the selected action (escalate to CMA, provide coping strategy, or direct to emergency resources).

When multiple agents, each with its own goals and capabilities, co-exist and interact within a shared environment, they form a **Multi-Agent System (MAS)**. An MAS is a system in which the overall intelligent behavior and functionality are a product of the collective, emergent dynamics of its constituent agents [37, 38]. The power of an MAS lies in its ability to solve problems that would be difficult or impossible for a monolithic system or a single agent to handle. This is achieved through social interaction, primarily:

- **Coordination and Cooperation:** Agents must coordinate their actions to avoid interference and cooperate to achieve common goals. In this thesis, the **Insights**, **Therapeutic Coach**, **Safety Triage**, and **Case Management** agents must cooperate: the Insights Agent provides the data-driven insights (beliefs) that the Therapeutic Coach Agent uses to form its outreach plans (intentions), while the Safety Triage Agent handles immediate, real-time needs that may fall outside the other agents' scopes, and the Case Management Agent manages the administrative follow-up.
- **Negotiation:** When agents have conflicting goals or must compete for limited resources, they must be able to negotiate to find a mutually acceptable compromise [39, 40].
- **Communication:** Effective interaction requires a shared Agent Communication Language (ACL), such as FIPA-ACL or KQML, which defines the syntax and semantics for messages, allowing agents to perform actions like requesting information, making proposals, and accepting or rejecting tasks [41, 42].

Therefore, this thesis leverages the MAS paradigm by designing a framework composed of four specialized, collaborative agents coordinated by a meta-agent orchestrator. Their individual, goal-directed behaviors, orchestrated within a hierarchical architecture, work in concert to achieve the overarching systemic objective: transforming institutional mental health support from a reactive model to a proactive, data-driven ecosystem.

2.2.2.1 Mathematical Formalization of Agent Decision Functions

To operationalize the BDI model for the Safety Agent Suite, each agent's core decision-making process is formalized as a mathematical function mapping inputs to outputs. This formalization bridges the theoretical BDI framework to the practical implementation described in Chapter 3.

Safety Triage Agent (STA) The STA assesses risk level $R_t \in \{0, 1, 2, 3\}$ from student message M_t:

$$R_t = f_{STA}(M_t; \theta_{LLM}, \mathcal{C}) \tag{2-4}$$

where θ_{LLM} represents the LLM parameters (Gemini 2.5 Flash) and $\mathcal{C}$ is the crisis pattern corpus used for contextual understanding. The discrete risk level maps to severity categories:

$$\text{severity}(R_t) = \begin{cases} \text{low} & R_t = 0 \\ \text{moderate} & R_t = 1 \\ \text{high} & R_t = 2 \\ \text{critical} & R_t = 3 \end{cases} \tag{2-5}$$

This classification determines subsequent routing: moderate risk ($R_t = 1$) triggers supportive coaching via TCA, while high or critical risk ($R_t \geq 2$) initiates immediate escalation to the Case Management Agent for clinical case management.

Therapeutic Coach Agent (TCA) The TCA generates therapeutic response A_t given the current message and conversation history:

$$A_t = f_{TCA}(M_t, H_{t-1}; \theta_{LLM}, \mathcal{I}) \tag{2-6}$$

where $H_{t-1} = \{(M_0, A_0), \ldots, (M_{t-1}, A_{t-1})\}$ is the conversation history capturing previous exchanges, and $\mathcal{I}$ represents the intervention library containing evidence-based therapeutic frameworks (CBT, Motivational Interviewing). The history dependency enables the agent to maintain therapeutic continuity and adapt interventions based on student progress.

Insights Agent (IA) The IA computes aggregate metric μ from anonymized message set $\mathcal{M}$:

$$\mu = f_{IA}(\mathcal{M}, q; \mathcal{K}) \tag{2-7}$$

where q represents the analytical query (e.g., crisis trend analysis, topic modeling, sentiment aggregation) and $\mathcal{K}$ enforces privacy constraints such as k-anonymity and query result suppression. The IA's output informs institutional decision-making by quantifying population-level trends while preserving individual privacy.

Case Management Agent (CMA) The CMA determines administrative action α_t based on case state and available resources:

$$\alpha_t = f_{CMA}(C_t, R_{avail}; \theta_{LLM}) \tag{2-8}$$

where C_t represents the current case state (severity, student information, appointment history) and R_{avail} denotes available resources (counselor schedules, emergency contact protocols). Actions include appointment scheduling, case note creation, and resource allocation.

These formalizations establish the mathematical foundation for the multi-agent coordination described in subsequent sections and implemented in Chapter 3.

2.2.3 Large Language Models (LLMs)

Large Language Models (LLMs) are a class of deep learning models that have demonstrated remarkable capabilities in understanding and generating human-like text. The architectural foundation for virtually all modern LLMs, including the Gemini models used in this research, is the **Transformer architecture**, first introduced by Vaswani et al. [43]. The Transformer's key innovation is the **self-attention mechanism**, which allows the model to dynamically weigh the importance of different words in an input sequence when processing and generating language. This enables the model to capture complex, long-range dependencies and contextual relationships far more effectively than its predecessors, such as Recurrent Neural Networks (RNNs) [44,45].

The Self-Attention Mechanism The self-attention mechanism computes contextual representations by relating different positions in a sequence to each other. Given an input sequence, the mechanism computes three matrices: Query (Q), Key (K), and Value (V), each derived through learned linear projections of the input embeddings. The attention operation is then defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2-9}$$

where d_k is the dimension of the key vectors. The scaling factor $\sqrt{d_k}$ prevents the dot products from growing too large, which would push the softmax function into regions with extremely small gradients.

The attention weights, computed by the softmax of the scaled dot products between queries and keys, determine how much each position in the sequence should attend to every other position. These weights are then used to compute a weighted sum of the value vectors, producing contextually-aware representations.

Modern Transformers employ **multi-head attention**, which applies multiple attention operations in parallel, each with different learned projections:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2-10}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ and W^O is an output projection matrix. This allows the model to attend to information from different representation subspaces simultaneously. For instance, Gemini 2.5 employs 32 attention heads per layer, enabling it to capture diverse linguistic patterns and semantic relationships concurrently.

The core operation of a Transformer-based model involves processing input text through a series of encoding and/or decoding layers. The process can be conceptualized as follows:

1. **Tokenization and Embedding:** Input text is first broken down into smaller units called tokens. Each token is then mapped to a high-dimensional vector, or an "embedding," that represents its semantic meaning.
2. **Positional Encoding:** Since the self-attention mechanism does not inherently process sequential order, a positional encoding vector is added to each token embedding to provide the model with information about the word's position in the sequence.
3. **Self-Attention Layers:** The sequence of embeddings passes through multiple self-attention layers. In each layer, the model calculates attention scores for every token relative to all other tokens in the sequence, effectively learning which parts of the input are most relevant for understanding the context of each specific token.
4. **Feed-Forward Networks:** Each attention layer is followed by a feed-forward neural network that applies further transformations to each token's representation.
5. **Output Generation:** The model's final output is a probability distribution over its entire vocabulary for the next token in the sequence. The model then typically selects the most likely token (or samples from the distribution) and appends it to the input, repeating the process autoregressively to generate coherent text [44].

This research utilizes a cloud-based API model strategy, leveraging the Gemini

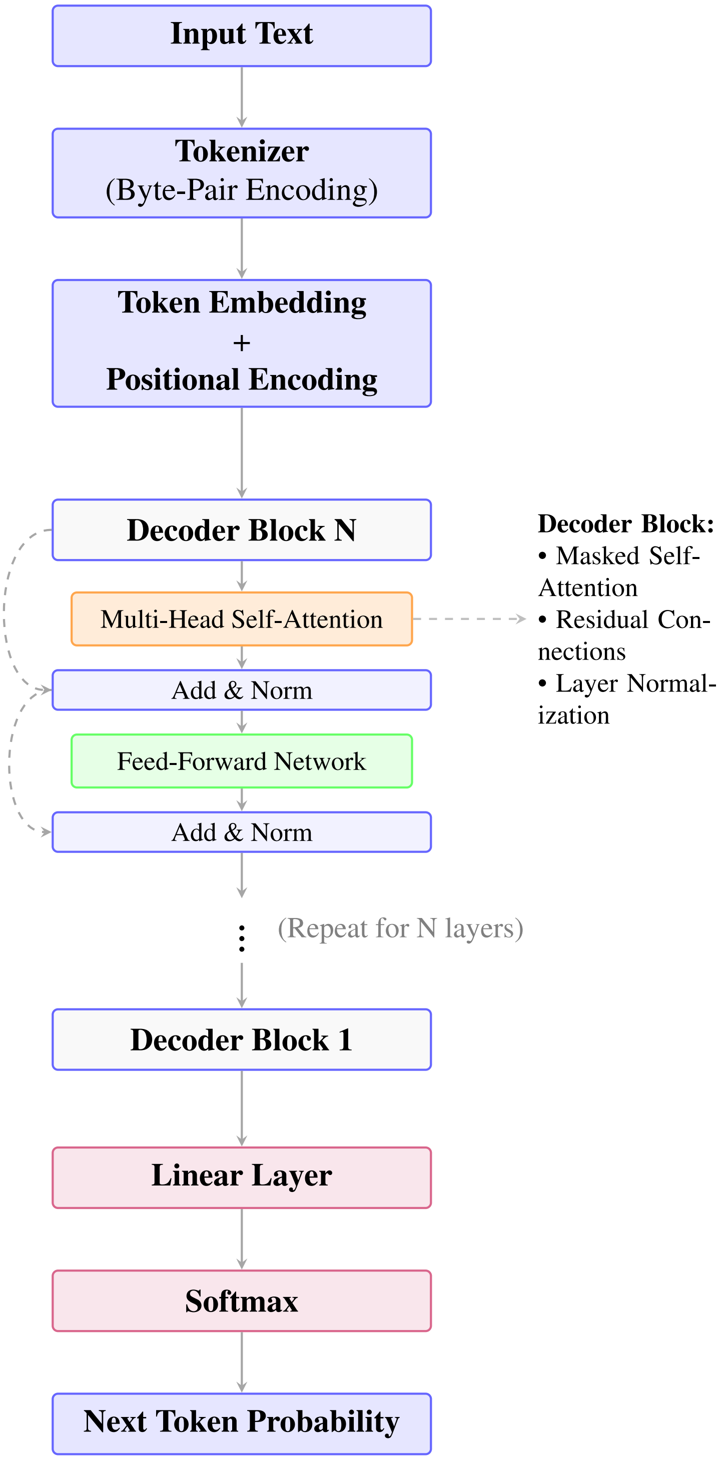

Figure 2.1. A simplified view of the decoder-only Transformer architecture used in generative LLMs. The model processes input embeddings through multiple layers (blocks) of masked multi-head self-attention and feed-forward networks with residual connections to predict the next token in a sequence.

2.5 family of models to balance performance, privacy, and capability. The Gemini models represent Google's state-of-the-art, natively multimodal foundation models, available in various sizes (e.g., Gemini Pro). Unlike models trained solely on text, Gemini was pre-trained from the ground up on multiple data modalities, giving it more sophisticated reasoning capabilities [46]. In this framework, a powerful model like Gemini 2.5 Pro is accessed via a secure API for all agentic tasks [47], from the real-time conversation handling of the Safety Triage Agent to the complex, non-sensitive tasks, such as the weekly trend analysis performed by the Insights Agent.

2.2.3.1 Cloud-Based API Models: The Gemini 2.5 Family

The framework integrates a state-of-the-art, proprietary model accessed via a cloud API. The Gemini family, specifically the flagship **Gemini 2.5 Flash** model, serves this role, providing a level of reasoning and multimodal understanding that is critical for handling the most complex tasks and ensuring system robustness. While a detailed architectural schematic is not public, in line with the proprietary nature of frontier AI models, its capabilities have been extensively documented by Google through official developer guides and announcements [46, 47].

Gemini 2.5 builds upon the efficient **Mixture-of-Experts (MoE) Transformer** architecture of its predecessors. In an MoE architecture, the model is composed of numerous smaller "expert" neural networks. For any given input, a routing mechanism activates only a sparse subset of these experts. This allows the model to have a very large total parameter count, enabling vast knowledge and capability, while keeping the computational cost for any single inference relatively low [46].

The strategic role of Gemini 2.5 in this framework is defined by its next-generation capabilities:

- **Native Multimodality with Expressive Audio:** A significant architectural leap in Gemini 2.5 is its native handling of audio [48]. Unlike models that first transcribe audio to text, Gemini 2.5 processes audio streams directly. This allows it to understand not just the words, but also the nuances of human speech such as tone, pitch, and prosody, which is invaluable for a mental health application where user sentiment is key.
- **Controllable Reasoning and "Thinking Time":** Gemini 2.5 introduces a "thinking budget," a mechanism that allows developers to control the trade-off between response latency and reasoning depth [46]. For high-frequency tasks performed by the Safety Triage Agent, a lower budget can ensure speed. For complex analytical tasks required by the Insights Agent, a higher budget can be allocated to allow for more thorough reasoning, providing granular control over both cost and quality.

- **Advanced Agentic Capabilities and Tool Use:** The model is explicitly designed to power advanced agents. It features more reliable and sophisticated function calling, enabling seamless integration with external tools and APIs [46]. This is essential for the Service Desk Agent to execute multi-step plans, such as scheduling an appointment based on a user's request.
- **High-Fidelity Reasoning:** As a frontier model, Gemini 2.5 serves as the high-capability engine for all requests, ensuring service continuity and the highest quality output.

By integrating Gemini 2.5 via its API, the agentic framework gains access to state-of-the-art reasoning power on demand, ensuring that it can handle a wide spectrum of tasks with both efficiency and exceptional quality.

2.2.4 LLM Orchestration Frameworks

While LLMs provide powerful reasoning capabilities, they are inherently stateless and lack direct access to external data or tools. An LLM, in isolation, cannot query a database, call an API, or access a private document. To build sophisticated, stateful applications that overcome these limitations, an orchestration framework is required.

2.2.4.1 LangChain: The Building Blocks of LLM Applications

LangChain is an open-source framework designed specifically for this purpose, providing the essential "glue" to connect LLMs with external resources and compose them into complex applications [49, 50]. The core philosophy of LangChain is to provide modular components that can be "chained" together to create complex workflows. The most recent and fundamental abstraction in LangChain is the **LangChain Expression Language (LCEL)**. LCEL provides a declarative, composable syntax for building chains, where the pipe (‘|‘) operator streams the output of one component into the input of the next. Every component in an LCEL chain is a "Runnable," a standardized interface that supports synchronous, asynchronous, batch, and streaming invocations, making it highly versatile for production environments [50, 51].

A simple LCEL chain can be represented as:

$$\text{Chain} = \text{PromptTemplate} \mid \text{LLM} \mid \text{OutputParser}$$

In this sequence, user input is first formatted by a ‘PromptTemplate‘, the result is passed to the ‘LLM‘ for processing, and the LLM's raw output is then transformed into a structured format (e.g., JSON) by an ‘OutputParser‘.

For this thesis, the most critical application of LangChain is its ability to create **agents**. A LangChain agent uses an LLM not just for text processing, but as a reasoning engine to make decisions. This is often based on a framework known as **ReAct (Reasoning and Acting)**, which enables the LLM to synergize reasoning and action [49, 52]. The agent is given access to a set of **Tools**, which are simply functions that can interact with the outside world (e.g., a database query function, a file reader, a web search API). The agent's operational loop, managed by an **Agent Executor**, can be formalized as an iterative process.

Let G be the initial goal and H_t be the history of actions and observations up to step t. The process at each step t is:

1. **Reasoning (Thought Generation):** The agent generates a thought th_t and a subsequent action a_t by sampling from the LLM's conditional probability distribution, given the goal and the history so far.

   $$(th_t, a_t) \sim p(th, a | G, H_{t-1}; \theta_{LLM})$$

   The prompt to the LLM contains the goal and the trajectory of previous thoughts, actions, and observations, guiding its next decision.

2. **Action Execution:** The Agent Executor parses a_t to identify the chosen tool and its input, then executes it to produce an observation, o_t.

   $$o_t = \text{ExecuteTool}(a_t)$$

3. **History Augmentation:** The new observation is appended to the history, forming the context for the next iteration.

   $$H_t = H_{t-1} \oplus (a_t, o_t)$$

   This loop continues until the LLM determines the goal G is met and generates a final answer.

This iterative loop is what transforms a passive LLM into a proactive, problem-solving agent. For example, the **Insights Agent** in this framework, when tasked with "summarizing student stress trends," would use this loop to formulate a SQL query (Thought and Action), execute it (Observation), and then use the results to generate a final summary. This orchestration is fundamental to enabling the autonomous capabilities central to this thesis.

2.2.4.2 LangGraph: Orchestrating Multi-Agent Systems

While LangChain's standard agent executors are powerful, they are often designed for linear, sequential execution paths. For a sophisticated multi-agent system like the **Safety Agent Suite**, where agents must collaborate, hand off tasks, and operate in a cyclical, stateful manner, a more robust orchestration mechanism is required. This is the role of **LangGraph**, an extension of LangChain designed for building durable, stateful, multi-agent applications by modeling them as cyclical graphs [53, 54].

The core concept of LangGraph is to represent the agentic workflow as a **state graph**. This is a directed graph where nodes represent functions or LLM calls (the "work" to be done) and edges represent the conditional logic that directs the flow of execution from one node to another. A central **State** object is passed between nodes, allowing each agent or tool to read the current state, perform its function, and then update the state with its results. This creates a persistent, auditable record of the agent's operations [51, 55].

A LangGraph workflow can be defined by the following components:

- **State Graph:** The overall structure, $G = (N, E)$, where N is a set of nodes and E is a set of directed edges. The graph's state is explicitly defined by a state object that is passed and updated throughout the execution.
- **Nodes:** Each node represents an agent or a tool. When called, a node receives the current state object as input and returns a dictionary of updates to be applied to the state. For example, the ‘Safety Triage Agent‘ node would take the user's message from the state, process it, and return an update specifying the assessed risk level.
- **Edges:** Edges connect the nodes and control the flow of the application. LangGraph supports **conditional edges**, which are crucial for agentic behavior. After a node executes, a routing function is called to inspect the current state and decide which node to move to next [50, 51]. For example, after the ‘Safety Triage Agent‘ runs, a conditional edge might route the workflow to the ‘Service Desk Agent‘ if the risk is moderate, or directly to an "escalate" tool if the risk is critical.

State Transition Semantics The stateful execution of a LangGraph workflow is governed by formal state update rules. Each node in the graph transforms the shared state through a state update function:

$$S_{t+1} = \text{node}_i(S_t) = S_t \oplus \Delta S_i \tag{2-11}$$

where S_t represents the current state at time step t, ΔS_i is the update produced by node i, and $\oplus$ denotes the state merging operation (where new fields override existing values while preserving unmodified fields).

Conditional edges implement routing logic via predicate functions that inspect the current state. For the Safety Agent Suite, the routing after risk assessment can be formalized as:

$$\text{next}(S_t) = \begin{cases} \text{escalate\_to\_sda} & \text{if } S_t.\text{risk\_level} \geq 2 \\ \text{provide\_coaching} & \text{if } S_t.\text{risk\_level} = 1 \\ \text{END} & \text{if } S_t.\text{risk\_level} = 0 \end{cases} \tag{2-12}$$

This formalization enables dynamic multi-agent orchestration where the STA's risk assessment (R_t) determines subsequent workflow paths: moderate risk routes to the TCA for therapeutic intervention, high or critical risk escalates to the CMA for clinical case creation, and low risk concludes the interaction. The explicit state management ensures that all downstream agents have access to the complete conversation context, enabling informed decision-making throughout the workflow.

More generally, for any conditional routing decision, the next node is determined by a routing function ρ:

$$\text{next\_node} = \rho(S_t) \in N \cup \{\text{END}\} \tag{2-13}$$

where ρ maps the current state to either another node in the graph or a terminal state, enabling arbitrary workflow complexity including loops, parallel execution, and human-in-the-loop interventions.

This cyclical, stateful approach provides several key advantages for this framework:

1. **Explicit Multi-Agent Collaboration:** LangGraph allows for the explicit definition of workflows where different agents are called in sequence or in parallel, and their outputs are used to inform the next step [55, 56]. This is essential for the **Safety Agent Suite**, where the 'Insights Agent''s output must trigger the 'Support Coach Agent'.
2. **State Management and Durability:** Because the state is explicitly managed, the agent's "memory" of the conversation and its previous actions is robust. The graph's execution can be paused, resumed, and inspected, which is vital for long-running, interactive coaching sessions.
3. **Flexibility and Control:** Unlike the more constrained loops of standard agent executors, LangGraph allows for the creation of arbitrary cycles. An agent can loop, retry a tool call if it fails, or route to a human-in-the-loop for verification, providing a much higher degree of control and reliability for a safety-critical application [57,58].

By using LangGraph to orchestrate the **Safety Agent Suite**, this framework moves beyond simple, linear agentic loops and implements a true multi-agent system capable of complex, stateful, and collaborative problem-solving [53,56].

2.3 Synthesis and Identification of the Research Gap

The preceding review of the literature and theoretical landscape reveals a critical disconnect. On one hand, the field has produced increasingly sophisticated but fundamentally **reactive** conversational agents for mental health. On the other, it has developed proactive institutional analytics that remain bottlenecked by a reliance on **manual intervention**. The failure of the existing literature is not in the individual components, but in the lack of integration between them.

This creates a significant and unaddressed research gap: the need for an **integrated, autonomous, and proactive framework** that can systemically bridge the chasm from data-driven insight to automated, personalized intervention and administrative action. Current systems are not designed as a cohesive ecosystem. The analytical tools do not automatically trigger the intervention tools, the conversational agents do not seamlessly hand off tasks to administrative agents, and the user-facing support does not operate with an awareness of the broader institutional context provided by analytics.

The central argument of this thesis is that the next frontier in institutional mental health support lies not in the incremental improvement of any single component, but in the **synergistic integration of multiple specialized agents** into a single, closed-loop system. Such a system, architected as a Multi-Agent System (MAS), is capable of emergent behaviors that are more than the sum of its parts.

Therefore, this research directly addresses the identified gap by proposing and prototyping a novel agentic AI framework, the **Safety Agent Suite**, where:

- An **Insights Agent (IA)** autonomously identifies trends, moving beyond the static dashboards of current well-being analytics and creating actionable intelligence.
- A **Therapeutic Coach Agent (TCA)** and a **Safety Triage Agent (STA)** act on this intelligence and on real-time user needs, providing both proactive, personalized coaching and immediate, context-aware crisis support. They function as the intelligent front-door to the support ecosystem, overcoming the limitations of purely reactive chatbots.
- A **Case Management Agent (CMA)** closes the "insight-to-action" loop on an administrative level, automating the workflows for clinical case management and resource allocation that currently render proactive models inefficient and unscalable.

By designing and evaluating a system where these agents work in concert, orchestrated by LangGraph, this thesis pioneers a holistic solution that is fundamentally more proactive, scalable, and efficient than the disparate tools described in the current literature.

CHAPTER III

SYSTEM DESIGN AND ARCHITECTURE

3.1 Research Methodology: Design Science Research (DSR)

The research presented in this thesis is constructive in nature, aimed not merely at describing or explaining a phenomenon, but at creating a novel and useful artifact to solve a real-world problem. To provide a rigorous and systematic structure for this endeavor, this study adopts the **Design Science Research (DSR)** methodology. DSR is a well-established paradigm in Information Systems research focused on the creation and evaluation of innovative IT artifacts intended to solve identified organizational problems [59]. The primary goal of DSR is to generate prescriptive design knowledge through the building and evaluation of these artifacts.

The DSR process model, as outlined by Peffers et al., provides an iterative framework that guides the research from problem identification to the communication of results [1]. This thesis follows these stages, mapping them directly to its structure to ensure a logical and transparent research process. The complete workflow of this research is visualized in Figure 3.2. This diagram illustrates the iterative path from problem formulation through to the final conclusions and recommendations.

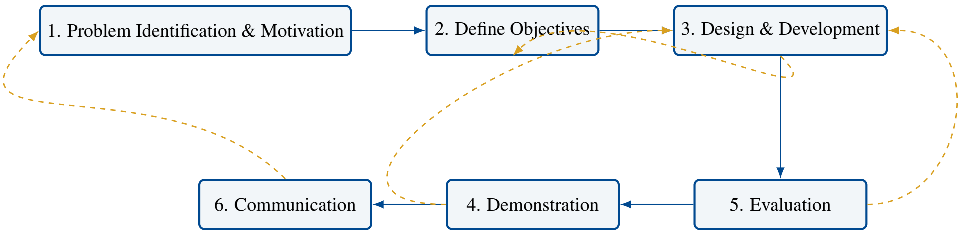

Figure 3.2. The Design Science Research (DSR) process model as applied in this thesis, adapted from Peffers et al. [1]. The diagram shows the sequential stages and the iterative feedback loops that inform the research process.

3.2 System Overview and Conceptual Design

The artifact proposed and developed in this research is a novel agentic AI framework designed to address the systemic inefficiencies of traditional, reactive mental health support models in Higher Education Institutions. The conceptual architecture is predicated on the principles of a Multi-Agent System (MAS), wherein a suite of collaborative, specialized intelligent agents, collectively termed the **Safety Agent Suite**, work in concert to create a proactive, scalable, and data-driven support ecosystem. This framework is designed not as a monolithic application, but as a dynamic, closed-loop system that operates on two interconnected levels: a micro-level loop for real-time, individual student support and a macro-level loop for strategic, institutional oversight and proactive intervention [60, 61].

The system's primary entities and their designated interaction points are illustrated in the conceptual context diagram in Figure **??**. This diagram shows how all users interact with a single, unified **Aika Meta-Agent**, which then coordinates the various specialist agents (STA, TCA, CMA, IA) that operate as background services.

Table 3.1. Agent descriptions and their primary roles in the Safety Agent Suite.

Agent	**Primary Role**
Aika Meta-Agent	The sole user-facing conversationalist and orchestrator. Manages all user interactions, performs initial risk assessment, and routes tasks to specialist agents.
Safety Triage Agent (STA)	A conceptual role embedded within Aika's initial analysis. Classifies immediate (Tier 1) and conversational (Tier 2) risk to enforce safety protocols.
Therapeutic Coach Agent (TCA)	A background agent that generates CBT-based intervention plans and recommends resources for the user's dashboard. Does not engage in direct conversation.
Case Management Agent (CMA)	The procedural backbone. Manages administrative tasks like crisis case creation, appointment scheduling, and sending notifications to counselors.
Insights Agent (IA)	The strategic analyst. Processes anonymized, aggregated data to provide population-level well-being trends and insights to administrators.

Conceptually, the framework's architecture is best understood as two distinct but integrated operational loops:

1. **The Real-Time Interaction Loop:** This loop handles immediate, synchronous interactions with individual students through a unified conversational interface. **Critically, the Aika Meta-Agent is the sole user-facing component**—students interact exclusively with Aika, never directly accessing the specialist agents. When a student sends a message, Aika processes it via a single Gemini API call that returns a structured JSON response containing: (1) the conversational reply, (2) immediate risk assessment (Tier 1: `none|low|moderate|high|critical`), (3) detected crisis keywords, (4) risk reasoning, (5) intent classification, and (6) a decision on whether specialist agents are needed. This unified response architecture ensures sub-second latency while embedding safety screening directly into every interaction.

   If `immediate_risk` is `high` or `critical`, Aika enters an information-gathering mode, collecting necessary details (location, immediate danger status, contact consent) before invoking the **Case Management Agent (CMA)** to escalate the case with structured crisis documentation for SLA-bound counselor assignment. The **Therapeutic Coach Agent (TCA)** operates entirely in the background, asynchronously generating CBT-based intervention plans and coping strategies that students can access via their dashboard—TCA does not participate in real-time conversations. At conversation end (detected via explicit goodbye signals or 5-minute inactivity), the **Safety Triage Agent (STA)** performs comprehensive conversation-level analysis (Tier 2) to identify cumulative risk patterns and recommend proactive follow-up interventions. This loop is designed for high-availability, low-latency responses (<300ms for Aika's direct replies), ensuring students receive immediate, contextually appropriate support through a single, consistent conversational persona.

2. **The Strategic Oversight Loop:** This loop operates on a longer, asynchronous timescale to enable proactive, institution-wide strategy. The **Insights Agent (IA)** works entirely in the background, periodically analyzing anonymized, aggregated data from all student interactions. However, administrators and counselors can invoke IA through Aika by requesting analytics queries (e.g., "show trending topics this week", "case statistics for November"), at which point Aika routes the request to IA and synthesizes the analytics report into a user-friendly response. IA generates reports on population-level well-being trends, sentiment analysis, and emerging topics of concern, delivered via both scheduled batch processing and on-demand queries through Aika's conversational interface. These insights provide empirical evidence for data-driven resource allocation, such as commissioning new workshops or adjusting counseling staff schedules. This loop directly addresses the "insight-to-action" gap that plagues current systems [12, 61].

The synergy between these two loops is the cornerstone of the framework's design. The real-time loop gathers the data that fuels the strategic loop, while the insights from the strategic loop can be used to configure and improve the proactive interventions delivered by the real-time loop, creating a continuously learning and adaptive support ecosystem. This dual-loop architecture is visualized in Figure 3.3.

3.2.1 Core Interaction: The Unified JSON Response Schema

The architectural lynchpin of the real-time interaction loop is the system's reliance on a structured, unified JSON response schema. When a user sends a message, the Aika Meta-Agent does not engage in a multi-step reasoning process with other agents. Instead, it makes a single, optimized call to its underlying language model (Gemini 2.5 Flash), guided by a system prompt that instructs it to return a comprehensive JSON object. This design pattern ensures that conversational fluency, safety screening, and routing logic are handled in a single, atomic transaction.

The returned JSON object's schema is detailed in Table 3.2. Each field serves a

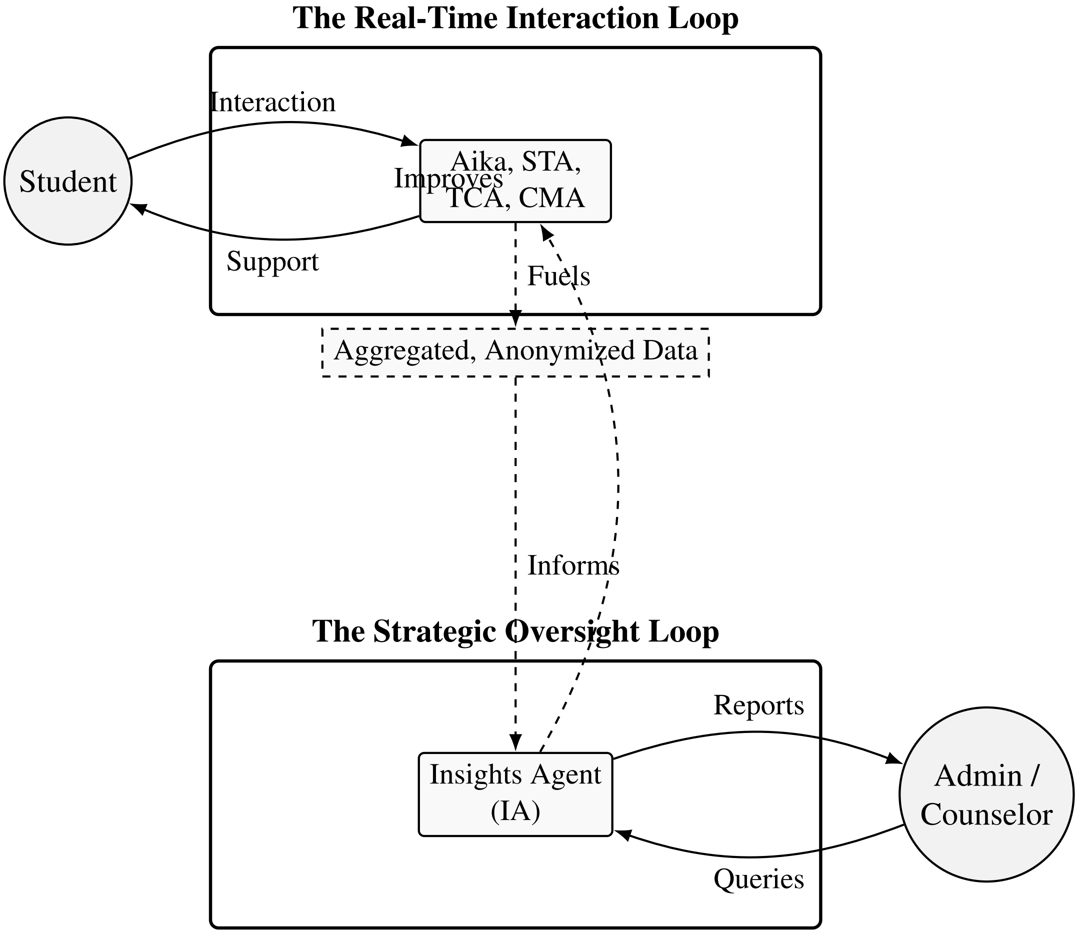

Figure 3.3. The Two Proactive Loops: The real-time loop provides immediate student support, generating data that fuels the strategic loop. The strategic loop analyzes this data to produce insights that, in turn, improve the proactive capabilities of the real-time loop.

distinct purpose in the agent's decision-making process, from generating an empathetic reply to providing a transparent audit trail for the assigned risk level.

Table 3.2. The unified JSON response schema returned by the Aika Meta-Agent.

Field	**Type**	**Description**
`reply`	string	The empathetic, context-aware conversational response to be displayed to the user.
`immediate\_risk`	string	A five-level risk classification (`none`, `low`, `moderate`, `high`, `critical`) for the single message, enabling instantaneous safety screening.
`crisis\_keywords`	array	A list of detected keywords from a predefined crisis lexicon (e.g., "bunuh diri," "menyakiti diri sendiri").
`risk\_reasoning`	string	A model-generated explanation for the assigned risk level, providing transparency for human oversight.
`intent`	string	The classified user intent (e.g., `request_support`, `schedule_appointment`), which dictates subsequent routing logic.
`needs\_agents`	boolean	A flag indicating whether the query requires routing to a background specialist agent.
`reasoning`	string	A brief explanation of why specialist agents are or are not needed, justifying the routing decision.
`suggested\_response`	string	An optional, complete response for simple interactions, allowing the system to bypass agent orchestration and reduce latency.

This unified response schema yields several architectural benefits. First, it facilitates **latency optimization**; by consolidating response generation and risk assessment into one call, the system can achieve sub-300ms response times, which is critical for maintaining conversational fluidity. Second, it enables **embedded safety**, as risk assessment is an integral and non-negotiable part of every interaction loop. Third, the schema ensures **transparent oversight** by providing a clear audit trail for the system's reasoning. Finally, the `needs_agents` flag allows for **conditional agent invocation**, an efficient resource management strategy that reduces backend compute costs by bypassing complex orchestration for simple queries.

An example of this schema in practice is shown in Listing III.1, where a user expresses moderate, non-imminent distress. In contrast, Listing III.2 shows the optimized response for a simple greeting. This structure operationalizes the principle that Aika is the sole user-facing component, synthesizing conversational intelligence and safety screening into a single, coherent interface layer.

```
  "reply"  "Saya mengerti kamu merasa tertekan dengan ujian yang akan
```

```
   datang. Wajar banget kok merasa cemas. Yuk, kita coba teknik
   grounding bersama untuk menenangkan pikiran."
 "immediate_risk"  "moderate"
 "crisis_keywords"   "ujian"  "stres"
 "risk_reasoning"  "User expresses feelings of pressure and anxiety
   related to upcoming exams, indicating moderate, non-imminent
   distress."
 "intent"  "request_emotional_support"
 "needs_agents"  true
 "reasoning"  "Requires TCA for CBT coping strategies and
   intervention plan"
```

Listing III.1. Example Aika JSON response for moderate stress scenario

In contrast, a simple greeting would return:

```
 "reply"  "Halo! Saya Aika, asisten kesehatan mental UGM. Saya di
   sini untuk mendengarkan dan membantu kamu. Ada yang ingin kamu
   ceritakan?"
 "immediate_risk"  "none"
 "crisis_keywords"
 "risk_reasoning"  "Casual greeting with no indicators of distress."
 "intent"  "casual_greeting"
 "needs_agents"  false
 "reasoning"  "Simple greeting, no specialist agent required."
 "suggested_response"  "Halo! Saya Aika, asisten kesehatan mental UGM
   . Saya di sini untuk mendengarkan dan membantu kamu. Ada yang ingin
    kamu ceritakan?"
```

Listing III.2. Example Aika JSON response for casual greeting

This response structure operationalizes the architectural principle that Aika is the sole user-facing component, synthesizing both conversational intelligence and safety screening into a single, coherent interface layer.

3.3 Functional Architecture: The Agentic Core

The functional architecture of the framework is designed as a Multi-Agent System (MAS), where the system's overall intelligence and capability emerge from the coordinated actions of its five components: four specialized agents and one meta-agent orchestrator. This section details the "what" of the system by defining the specific role, operational logic, and capabilities of each component within the **Safety Agent Suite**. Each specialist agent functions as a distinct component within the LangGraph state machine, perceiving its environment through the shared state, executing its logic, and updating the state with its results, while the Aika Meta-Agent coordinates their invocation and synthesizes their outputs.

3.3.1 The Safety Triage Agent (STA): The Real-Time Guardian

The Safety Triage Agent (STA) serves as the system's frontline safety monitor. Its primary function is to analyze every incoming student message to assess the immediate risk level and update the shared system state accordingly. This ensures that no high-risk message proceeds to a therapeutic agent without a safety intervention, enforcing the system's foundational "safety-first" principle.

The agentic behavior of the STA can be understood through the BDI model:

- **Beliefs:** The STA's beliefs are formed from the raw text of the user's message and the structured output of the Gemini 2.5 Flash model, which provides a risk classification (`low`, `moderate`, `critical`) and detected crisis keywords.
- **Desires:** Its fundamental desire is to ensure user safety by correctly identifying and escalating any potential crisis.
- **Intentions:** Based on its beliefs, the STA forms a clear intention. If the belief is that the risk is `critical`, its intention becomes to update the agent state to trigger the "escalate_to_cma" edge in the LangGraph. Otherwise, its intention is to pass control to the next appropriate agent.

3.3.2 The Therapeutic Coach Agent (TCA): The Empathetic Guide

The Therapeutic Coach Agent (TCA) acts as the primary conversational partner for students in non-crisis situations. Once the STA has cleared a message as safe, the TCA takes over to provide empathetic listening, emotional support, and evidence-based therapeutic guidance.

Its agentic model is as follows:

- **Beliefs:** The TCA's beliefs include the user's message history, the STA's "safe" classification, and any previously generated intervention plans stored in the agent state.
- **Desires:** Its core desire is to reduce user distress and build coping skills by providing helpful, evidence-based guidance.
- **Intentions:** If the user's message indicates a need for coping strategies, the TCA forms the intention to execute its `generate_intervention_plan` tool. This tool call is a committed action to fulfill its desire. If the user is simply venting, its intention is to generate an empathetic, listening-oriented response.

3.3.3 The Case Management Agent (CMA): The Procedural Coordinator

The Case Management Agent (CMA) serves as the system's administrative backbone, responsible for all procedural and operational tasks. It is activated under two conditions: following a critical risk escalation from the STA, or when a user directly expresses an intent for administrative action.

Its BDI breakdown is highly procedural:

- **Beliefs:** The CMA believes the state of the world requires administrative action. This belief is triggered by either a `critical` flag from the STA or a direct user request for a service like booking an appointment.
- **Desires:** Its primary desire is to execute administrative workflows reliably and accurately.
- **Intentions:** When triggered by a crisis escalation, it forms the intention to execute the `create_crisis_case` tool. When triggered by a user request, it forms the intention to use the `schedule_appointment` tool. Each intention maps directly to a deterministic, procedural tool call.

3.3.4 The Insights Agent (IA): The Strategic Analyst

The Insights Agent (IA) functions as the institution's automated well-being analyst, tasked with identifying population-level mental health trends from aggregated data. It is invoked exclusively by administrators to generate strategic reports.

Its agentic model is focused on data analysis and synthesis:

- **Beliefs:** The IA's beliefs are derived from the administrator's query (e.g., "Show me crisis trends for October") and the aggregated, anonymized data it can access from the database.
- **Desires:** Its desire is to provide accurate, privacy-preserving, and actionable insights that help university leadership make data-driven decisions.
- **Intentions:** Based on the administrator's request, the IA forms an intention to run a specific, pre-defined SQL query against the database. It then forms a subsequent intention: to synthesize the numerical results from that query into a coherent, narrative summary for the administrator.

3.3.5 The Aika Meta-Agent: Unified Orchestration Layer

While the four specialized agents (STA, TCA, CMA, IA) provide the system's core intelligence, their coordination requires an orchestration layer. This layer must solve a fundamental challenge in multi-agent systems: how to present a unified, coherent interface to different user roles while dynamically routing requests based on intent, access rights, and context [15]. The Aika Meta-Agent is designed as this unified orchestration layer, acting as the single point of contact for all users and the master controller for the specialist agents operating in the background. Its primary responsibilities are to interpret user intent, manage conversational state, enforce role-based access control, and synthesize the outputs of the specialist agents into a coherent response.

Collectively, these specialized agents operationalize the two proactive loops described in Section 3.2. The STA and TCA are the primary actors in the **Real-Time Interaction Loop**, enabling proactive individual support through immediate risk detection and the asynchronous delivery of therapeutic content. The IA is the engine of the **Strategic Oversight Loop**, providing the institution with proactive, population-level insights. The CMA acts as a crucial bridge between these loops, translating automated insights (from STA or IA) into concrete administrative actions, such as case creation or counselor notification. This functional separation ensures that each component is optimized for its specific role within the broader proactive ecosystem.

3.4 Technical Architecture

This section details the technical blueprint of the Safety Agent Suite, translating the conceptual and functional designs into a concrete implementation strategy. The architecture is built upon a modern, cloud-native technology stack, selected to ensure modularity, scalability, and maintainability, which are critical for a system of this nature.

3.4.1 Technology Stack

The selection of technologies was guided by the need for asynchronous performance, robust data management, and stateful agent orchestration. The core components are:

- **Backend Framework: FastAPI.** The backend is implemented in Python using FastAPI. This choice was motivated by FastAPI's high performance and its native support for asynchronous operations. For a conversational AI system where multiple I/O-bound tasks occur (e.g., database queries, external API calls to LLMs), asynchronous handling is paramount to prevent blocking and ensure a responsive user experience.
- **Agent Orchestration: LangGraph.** The complex, conditional logic of the multi-agent system is managed using LangGraph. LangGraph provides a stateful, graph-based framework for composing agents. This is a significant improvement over stateless LLM calls, as it allows the system to maintain a coherent state across multiple turns of a conversation and multiple agent invocations. It directly enables the implementation of the agentic loops and decision points described in the functional architecture.
- **Data Persistence: PostgreSQL and SQLAlchemy.** A PostgreSQL database serves

as the primary data store for all persistent information, including user profiles, conversation histories, and agent execution logs. Interaction with the database is managed through the SQLAlchemy Object-Relational Mapper (ORM). This combination provides a robust, transactional, and scalable foundation for data management, while the ORM simplifies data handling in the Python application code.

- **Containerization: Docker.** The entire application stack, including the FastAPI backend, database, and other services, is containerized using Docker. This ensures a consistent, reproducible, and isolated environment for development, testing, and potential deployment, simplifying dependency management and enhancing system reliability.

3.4.2 Data Model and Persistence

The system's data model is designed to support its core functions: tracking conversations, managing user data, and logging agent behavior for analysis and auditing. While a full database schema is extensive, the core entities include:

- **User and Profile Tables:** Store essential user information, preferences, and consent status, forming the basis for personalized interaction.
- **Conversation and Message Tables:** Log every user interaction, providing the raw data for the Insights Agent and a history for contextual conversations.
- **Case Management Tables:** Store structured data for escalated cases, including risk level, summary, and assigned counselor, enabling the HITL workflow.
- **LangGraph Execution Logs:** A critical component for fulfilling RQ2, these tables (`LangGraphExecution` and `LangGraphNodeExecution`) capture detailed traces of every agent orchestration. They log which nodes (agents) were executed, the transitions between them, their inputs and outputs, and any errors encountered. This provides an invaluable audit trail for debugging and evaluating the orchestration logic.

3.4.3 Stateful Orchestration with LangGraph

The heart of the technical architecture is the LangGraph state machine, which operationalizes the agentic behavior. The orchestration is not a simple linear chain but a cyclical graph where the flow is determined by the contents of a shared `AgentState` object.

The process is as follows:

1. A user message initializes the `AgentState`.
2. The graph routes the state to the first node, the **Safety Triage Agent (STA)**.
3. The STA executes its logic and updates the `AgentState` with a risk assessment and a routing decision (e.g., `next_step: "tca"`).

4. A conditional edge reads the `next_step` from the state and routes it to the appropriate next node, which could be the **Therapeutic Coach Agent (TCA)**, the **Case Management Agent (CMA)**, or the end of the graph.
5. This process repeats, with each agent modifying the shared state, until a terminal node is reached.

This stateful, graph-based approach provides a robust and explicit way to manage the complex, non-deterministic nature of a multi-agent conversational system. A high-level visualization of this state machine is presented in Figure 3.4.

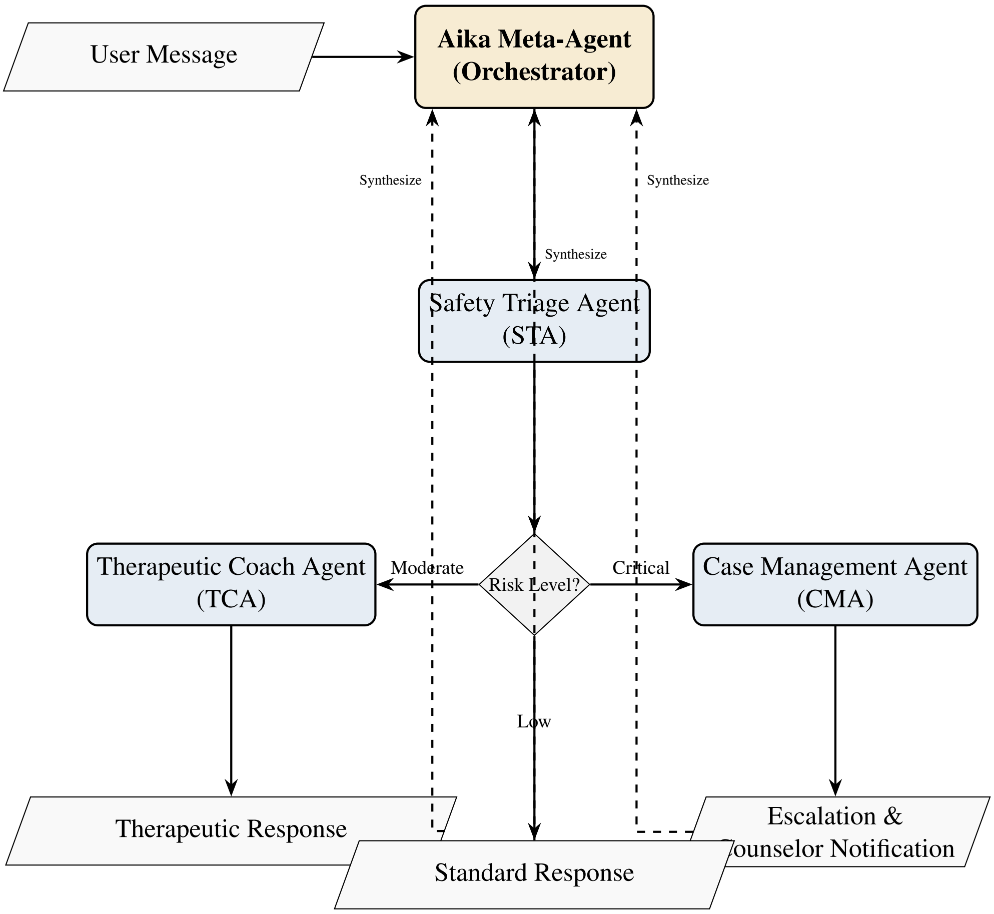
User Message
Aika Meta-Agent (Orchestrator)
Synthesize
Synthesize
Synthesize
Safety Triage Agent (STA)
Therapeutic Coach Agent (TCA)
Moderate
Risk Level?
Critical
Case Management Agent (CMA)
Low
Therapeutic Response
Standard Response
Escalation & Counselor Notification

Figure 3.4. High-level visualization of the LangGraph agent orchestration state machine. The Aika Meta-Agent routes user input to the STA for risk assessment, then conditionally invokes the TCA for therapeutic coaching or the CMA for crisis escalation based on the outcome.

3.5 Cross-Cutting Concerns

Beyond the core functional and technical architecture, a production-worthy system, particularly in a sensitive domain like mental health, must address several system-wide, non-functional requirements. These "cross-cutting concerns" ensure the system is secure, responsive, and safe.

3.5.1 Security and Privacy by Design

Security and privacy are not afterthoughts but are foundational to the system's design, earning user trust and ensuring ethical operation.

- **Role-Based Access Control (RBAC):** The system enforces strict access control based on user roles (e.g., student, counselor, administrator). For instance, counselors can only view cases assigned to them, and administrators can access aggregated analytics from the Insights Agent but not individual, non-anonymized conversation logs. This is managed through authentication middleware in the FastAPI backend.
- **Data Encryption:** All data is encrypted both in-transit, using TLS for all API communications, and at-rest in the PostgreSQL database. This protects sensitive conversation data from unauthorized access even in the event of a direct infrastructure breach.
- **Privacy-Preserving Analytics:** The Insights Agent is architecturally constrained to protect student privacy. As stated in RQ4, its SQL queries are designed to enforce k-anonymity by including clauses that prevent data from being returned for any group smaller than a predefined threshold (k=5). This ensures that analytics can reveal population-level trends without ever exposing data that could be traced back to an individual student.

3.5.2 Architectural Provisions for Responsiveness

While formal performance benchmarking is outside the scope of this thesis, the architecture was explicitly designed to support a responsive, real-time conversational experience. This is a critical functional requirement for user engagement.

- **Asynchronous Processing:** The choice of FastAPI was deliberate for its native `async/await` support. This allows the application to handle long-running I/O operations, such as calling the Gemini API or querying the database, without blocking the main execution thread. This ensures the system can manage multiple concurrent conversations smoothly.
- **Optimized Language Models:** The system employs a two-tier model strategy to balance capability with latency. For the initial, real-time safety screening performed by the STA, a low-latency model (Gemini 2.5 Flash) is used to ensure sub-second response times. For more complex, asynchronous tasks like generating detailed narrative summaries in the Insights Agent, a more powerful model (Gemini 2.5 Pro) is used, as latency is less critical for these background tasks.

3.5.3 Human-in-the-Loop (HITL) Workflow for Safety

No fully automated system can or should replace human clinical judgment in crisis situations. The framework is designed with a robust Human-in-the-Loop (HITL) workflow as its ultimate safety net.

The escalation path is deterministic and auditable:

1. The **STA** detects a message with "Critical" risk.
2. This immediately triggers the **Case Management Agent (CMA)**.
3. The CMA executes its `create_crisis_case` tool, which creates a structured, high-priority ticket in the database.
4. Simultaneously, the CMA invokes a notification service (e.g., via email or a secure messaging integration) that sends an alert to the on-call human counselor(s). This alert contains the case ID and a link to a secure dashboard where they can review the conversation.
5. The system then presents the user with immediate, static help resources (e.g., emergency hotline numbers) while the human counselor takes over the case management.

This HITL design ensures that the AI's role is to act as a high-speed, scalable detection and triage system, but the ultimate responsibility for crisis intervention remains with trained human professionals.

3.6 Ethical Considerations and Research Limitations

The development of an AI-driven framework for mental health support necessitates thorough examination of ethical implications and transparent acknowledgment of research limitations. This section addresses the ethical design choices and defines the boundaries of the study's findings.

3.6.1 Informed Consent and Transparency

The UGM-AICare framework is designed with the principle that users must have clear understanding of the system's capabilities and limitations. The Support Coach Agent explicitly discloses its non-human nature in initial interactions, ensuring users engage with informed consent about the conversational context. This transparency is critical in healthcare applications where users may form therapeutic relationships with AI systems.

3.6.2 Human-in-the-Loop for Safety and Ethical Safeguards

The framework is explicitly designed as a tool that assists, but does not replace, human counselors. Every critical risk escalation from the Safety Triage Agent (STA) creates a case that requires mandatory review and action by a qualified human professional. The system automates the detection and reporting, but the final clinical judgment and intervention remain firmly in human hands.

This human oversight is not merely procedural—it addresses the fundamental ethical limitation of LLMs in safety-critical contexts. While models like Gemini 2.5 Flash demonstrate strong performance in text understanding, they can still misinterpret nuanced emotional states or linguistic cues. The human-in-the-loop design ensures that no automated risk classification leads directly to intervention without expert clinical validation.

Given the high-stakes nature of mental health triage, the Safety Triage Agent is designed with explicit ethical safeguards:

- **Conservative Risk Classification:** The agent employs a "safety-first" bias, erring on the side of escalation when ambiguous risk indicators are detected. This prevents false negatives in critical situations.
- **Human-in-the-Loop for Critical Cases:** All cases flagged as "critical" by the STA trigger immediate notifications to human counselors. The agent does not make autonomous decisions about crisis intervention; it serves as a detection and escalation mechanism only.
- **Transparency in Agent Responses:** The Support Coach Agent explicitly discloses its non-human nature and limitations in its initial greeting, ensuring users have informed consent about the conversational context.

Technology alone is insufficient to guarantee ethical operation. Therefore, the system is designed with procedural safeguards that ensure human oversight for all critical functions, ensuring the framework operates as a support tool rather than as an autonomous clinical actor.

3.6.3 AI as Support Tool, Not Replacement for Therapy

It is ethically imperative to clearly define the system's role. The UGM-AICare framework is designed as a sub-clinical, supportive tool and a bridge to professional care, not as a substitute for licensed therapy. The Therapeutic Coach Agent is programmed to explicitly state this boundary and to encourage users to seek professional help for serious or persistent issues, facilitated through the Case Management Agent's appointment booking functionality and clinical escalation workflows.

3.6.4 Research Limitations and Scope Boundaries

This study, as a work of Design Science Research focused on artifact creation and evaluation, is subject to several important limitations:

- **Methodological Limitation - Scenario-Based Evaluation:** The evaluation of this framework (detailed in Chapter **??**) is based on controlled scenario testing with synthetic conversational data, not real-world user deployment. This thesis validates the *technical feasibility* of the agentic workflows and the *architectural integrity* of the multi-agent design. It does **not** measure long-term psychological outcomes or therapeutic efficacy on actual students. Such claims would require extensive ethics approval, medical supervision, and longitudinal clinical trials that exceed the scope of bachelor's-level research.
- **Technical Limitation - Inherent Risks of LLMs:** The framework relies on Google Gemini 2.5 Flash and Gemini 2.5 Flash Lite APIs for different agent tasks (routing, classification, plan generation). Like all LLMs, these models are subject to inherent limitations including potential biases from training data and the possibility of generating factually incorrect or nonsensical responses ("hallucinations"). While the system's use of structured tools, typed state schemas, and explicit agent prompts is designed to mitigate these risks, they cannot be eliminated entirely.
- **Data Limitation - Simulated Evaluation Data:** The evaluation is conducted using synthetically generated mental health scenarios and simulated conversational patterns, not real user data. This is necessary to protect privacy during the development phase and to enable controlled testing without requiring human subjects approval. However, it means that agent performance has not been validated on the specific linguistic diversity, cultural contexts, and edge cases of a live Indonesian student population.
- **Scope Limitation - Agent Architecture Focus:** This thesis evaluates the multi-agent architecture: the BDI-based specialist agents, Aika orchestration layer, and their collective behavior in safety-critical conversations. The full UGM-AICare implementation includes database design, user interface components, blockchain token systems, and deployment infrastructure, but **these system components are not subjects of formal evaluation in this work**. They serve as implementation context to demonstrate feasibility, but their performance characteristics, user experience quality, and production readiness are not validated. The thesis evaluates agent performance through controlled scenario-based testing rather than real-world user deployment.

These limitations do not diminish the validity of the research findings within their defined scope. They represent transparent acknowledgment of the boundaries between artifact evaluation (the focus of this thesis) and clinical deployment (which requires additional validation beyond this work's scope). The evaluation methodology in Chapter **??** is designed to rigorously assess the aspects that *can* be measured through controlled testing: agent accuracy, orchestration correctness, response quality, and privacy preservation in aggregated analytics.

CHAPTER IV
IMPLEMENTATION AND EVALUATION

This chapter reports how the prototype was exercised and what we learned from it. The focus is on the agents and their behavior in safety-relevant scenarios. We keep the scope practical and transparent so results can be reproduced and audited.

4.1 Implementation Artifact: The UGM-AICare Prototype

The conceptual framework and agentic architecture detailed in Chapter III were realized as a tangible software artifact within the UGM-AICare project. This prototype serves as the concrete object of study for the evaluation presented in this chapter. It is a full-stack web application designed to provide a practical testbed for the proposed proactive mental health support model. The complete source code for the artifact is publicly available for review and replication[1].

The artifact's technical implementation translates the architectural design into a working system:

- **Backend Services:** The core of the system is a Python-based backend built on the **FastAPI** web framework. Each specialized agent (STA, TCA, CMA, IA) is implemented as a distinct service within this backend, ensuring modularity and separation of concerns. This service-oriented architecture allows for independent development, testing, and scaling of each agent's capabilities.
- **Agent Orchestration Core:** The multi-agent coordination logic, described conceptually as a state machine in Chapter 3, is implemented using **LangGraph**. LangGraph provides the underlying engine to define the nodes (agents and tools) and edges (conditional transitions) of the agentic workflow. This allows the Aika Meta-Agent to dynamically route user requests and manage the flow of information between the specialized agents based on the evolving state of the conversation.
- **Frontend Interface:** A user-facing web application, built with **Next.js** and TypeScript, provides the conversational interface for students and the administrative dashboard for counselors. This interface communicates with the FastAPI backend via a RESTful API, ensuring a clean separation between the presentation layer and the backend agentic logic.
- **Integrated Observability:** As detailed in Section 4.2, the backend is instrumented with Prometheus for quantitative metrics and Langfuse for detailed tracing. This in-

[1] The UGM-AICare project repository can be accessed at `https://github.com/gigahidjrikaaa/UGM-AICare` or through `https://aicare.sumbu.xyz`

strumentation is not an afterthought but a core part of the implementation, providing the empirical data necessary for the evaluation that follows.

This implementation provides the technical foundation upon which the evaluation protocols described in the remainder of this chapter are executed.

4.2 Monitoring and Observability Infrastructure

To enable a rigorous and transparent evaluation of the agentic framework, a dual-stack observability infrastructure was implemented. This infrastructure is foundational to the Design Science methodology, providing the empirical data required to validate the research questions outlined in Chapter 1. The stack combines quantitative performance monitoring with deep, qualitative trace analysis, ensuring a holistic view of the system's operational behavior.

4.2.1 Prometheus for Quantitative Performance Metrics

For high-level, real-time performance monitoring, the backend exposes custom metrics to a Prometheus time-series database. This allows for the quantitative analysis of system health and efficiency. Key metrics include:

- **Agent Processing Time (`agent\_processing\_time\_seconds`):** A histogram metric that tracks the reasoning latency for each agent, crucial for evaluating the performance aspect of RQ1 (Proactive Safety).
- **Tool Call Outcomes (`tool\_calls\_total`):** A counter that tracks the success and failure rates of tool invocations, directly measuring the functional correctness of the orchestration logic for RQ2.
- **Crisis Escalation Events (`crisis\_escalations\_total`):** A counter for safety-critical events, providing a quantitative measure of the Safety Triage Agent's intervention frequency (RQ1).

These metrics are scraped at 15-second intervals, providing the statistical basis for the performance results reported in subsequent sections.

4.2.2 Langfuse for Qualitative Trace Analysis

While Prometheus provides the "what" of system performance, Langfuse provides the "why." As an open-source observability platform designed for LLM applications, Langfuse captures detailed, end-to-end traces of every agent interaction. This qualitative data is essential for debugging and for a deep understanding of the agents' reasoning processes. For each user request, Langfuse logs:

- **State Transitions:** The complete path of execution through the LangGraph state machine, which is used to manually verify state transition accuracy for RQ2.
- **LLM Invocations:** The exact prompts, model parameters, and generated outputs for every call to the Gemini models, enabling analysis of response quality for RQ3.
- **Tool Calls:** The inputs and outputs of every tool used by the agents, which helps diagnose failures in the orchestration flow (RQ2).

This detailed tracing capability provides the ground truth for analyzing agent behavior, validating the correctness of the multi-agent coordination, and understanding the root cause of any failures or unexpected outcomes. The combination of Prometheus and Langfuse thus provides a comprehensive framework for evaluating the artifact against its design goals.

4.3 Evaluation Scope and Methodology

4.3.1 Scope Boundaries and Rationale

This evaluation adopts a **proof-of-concept validation approach** appropriate for bachelor's-level Design Science Research. The objective is to demonstrate the **technical feasibility** of the proposed multi-agent architecture—specifically, that the Safety Agent Suite can execute core workflows correctly under controlled conditions. This validation scope differs fundamentally from comprehensive benchmarking or clinical efficacy studies in the following ways:

- **Sample Sizes:** Modest test set sizes (50 crisis prompts, 10 orchestration flows, 10 coaching scenarios, code review for privacy) enable focused validation of architectural correctness without requiring extensive data collection infrastructure. This is consistent with DSR artifact evaluation conventions [**?**], where initial validation focuses on demonstrating capability rather than exhaustive performance characterization.
- **Simulated Data:** All testing utilizes synthetically generated data to protect privacy and enable controlled, repeatable experiments. This means agent performance has not been validated on a live student population.
- **Single-Rater Assessment:** Response quality is assessed by the primary researcher using a structured rubric, with GPT-4 providing a validation score. This pragmatic approach demonstrates the methodology while acknowledging that formal clinical validation remains future work.
- **Code Review for Privacy:** Rather than generating extensive synthetic logs, RQ3 validation focuses on code inspection and unit tests demonstrating that k-anonymity enforcement mechanisms function as designed. This validates the *implementation correctness* of privacy safeguards.

Positioning Statement: This evaluation demonstrates that the proposed multi-agent architecture is *technically feasible*—the agents can classify crises, orchestrate workflows, generate appropriate responses, and enforce privacy thresholds under controlled conditions. It does **not** claim to have validated clinical efficacy, cultural appropriateness for Indonesian students, or production-readiness for deployment without further testing. Such claims would require ethics approval, multi-rater expert evaluation, field pilots with real users, and longitudinal outcome measurement—activities beyond bachelor's thesis scope but identified as critical future work in Section 4.9.

4.3.2 Measuring Proactive Capabilities

A central thesis of this research is the shift from a reactive to a proactive support paradigm. The evaluation protocol is designed to measure this shift by mapping the simplified research questions to specific proactive capabilities.

- **Proactive Safety (RQ1):** The core of a proactive safety model is its ability to identify risk without explicit user disclosure. The evaluation of the Safety Triage Agent (STA) directly measures this. The False Negative Rate (FNR) is the primary metric for proactive safety; a low FNR indicates the system can reliably detect latent crisis indicators and initiate an intervention, in contrast to a reactive model that would wait for a user to explicitly state "I need help."
- **Functional Correctness (RQ2):** A proactive system must be reliable. The evaluation of the framework's orchestration logic measures its ability to correctly execute automated workflows, handle errors, and route tasks without failure. This ensures the system can act on its proactive insights dependably.
- **Output Quality & Privacy (RQ3):** A proactive framework must produce outputs that are both useful and safe. This involves evaluating the quality of coaching advice to ensure it is appropriate and helpful, while simultaneously verifying that institutional insights are generated in a way that rigorously protects student privacy. This combined evaluation ensures the system's actions are both effective and responsible.

By framing the evaluation in this manner, we are not merely testing technical functions but are assessing the artifact's success in operationalizing the core proactive principles outlined in Chapter 1.

4.4 Setup and Test Design

This section documents the evaluation protocol that links the Design Science stages in Chapter III to the simplified research questions. Figure 4.5 and Table 4.1 provide a visual and tabular overview of the assets, metrics, and acceptance thresholds used throughout the chapter.

Evaluation Environment

- **Agents under test**: Safety Triage Agent (STA), Therapeutic Coach Agent (TCA), Case Management Agent (CMA), and Insights Agent (IA) running inside the LangGraph orchestration described in Chapter III. All tool invocations are captured through structured logs to enable replay and auditing.
- **Core Models**: Google Gemini 1.5 Flash for triage and routing; Google Gemini 1.5 Pro for coaching and analysis.
- **Instrumentation**: Langfuse observability platform provides trace-level monitoring for agent execution with span-level detail; execution state tracker (ExecutionStateTracker class) persists node timing, state transitions, and retry attempts to database; Prometheus metrics expose latency distributions (p50/p95/p99), escalation decisions, and error rates; processing time measured via Python's perf_counter with millisecond precision.

Table 4.1. Simplified Evaluation Plan Overview.

Research Question	Evaluation Method	Metrics	Target
RQ1: Proactive Safety	Scenario-based testing on crisis corpus (n=50)	Sensitivity, Specificity, False Negative Rate (FNR), p50/p95 Latency	FNR $\leq$ 10%
RQ2: Functional Correctness	Workflow execution testing on orchestration suite (n=10)	Tool Call Success Rate, Retry Recovery Rate, State Transition Accuracy	Success Rate $\geq$ 95%
RQ3: Output Quality & Privacy	Rubric scoring on coaching prompts (n=10) & Code review/unit tests for privacy	Mean Rubric Score (1-5 scale), Boundary Behavior Accuracy, K-Anonymity Compliance	Score $\geq$ 3.5/5, 100% Compliance

Datasets and Scenario Assets

- **Crisis Corpus (RQ1):** 50 synthetic prompts (25 crisis, 25 non-crisis) to measure classification accuracy.
- **Orchestration Test Suite (RQ2):** 10 structured conversation flows designed to test agent routing, tool use, and error handling.
- **Coaching Prompts (RQ3):** 10 scenarios for evaluating the quality of the Therapeutic Coach Agent's responses.
- **Privacy Validation (RQ3):** Code review and unit tests for the `InsightsAgentService` to verify k-anonymity enforcement.

Quality Control and Validation

- **Safety Reviews**: All crisis classifications are validated against ground truth labels.
- **Quality Assessment**: Coaching responses are scored against a defined rubric by the primary researcher, with GPT-4 as a validation rater.
- **Privacy Verification**: Code inspection and unit tests confirm that privacy-preserving mechanisms function as designed.

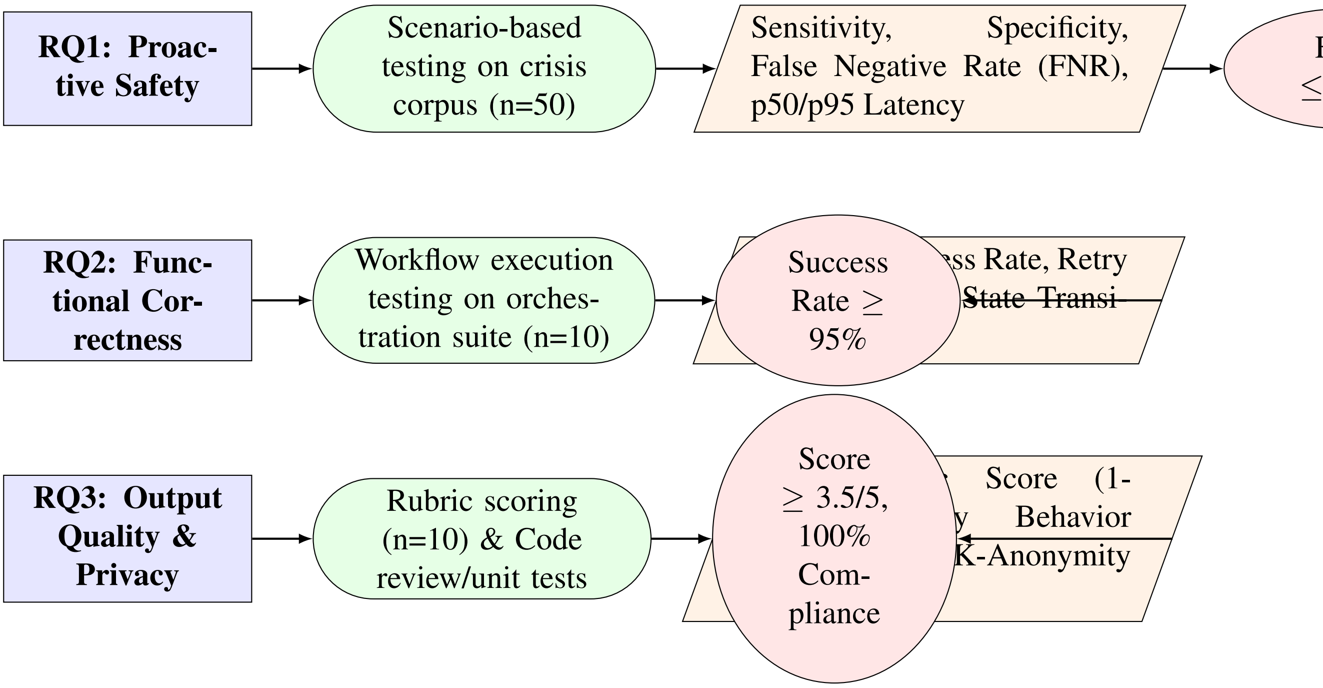
RQ1: Proactive Safety
Scenario-based testing on crisis corpus (n=50)
Sensitivity, Specificity, False Negative Rate (FNR), p50/p95 Latency
F
≤
RQ2: Functional Correctness
Workflow execution testing on orchestration suite (n=10)
ss Rate, Retry
State Transi-
Success Rate ≥ 95%
RQ3: Output Quality & Privacy
Rubric scoring (n=10) & Code review/unit tests
Score (1-
y Behavior
K-Anonymity
Score ≥ 3.5/5, 100% Compliance

Figure 4.5. Simplified Evaluation Pipeline mapping RQs to test assets and metrics.

4.5 Metrics Calculation Methodology

This section documents the precise calculation methods for all quantitative metrics.

4.5.1 RQ1: Proactive Safety Metrics

Classification Performance Metrics

Given the crisis corpus, we define:

- **True Positives (TP)**: Crisis messages correctly classified as crisis.
- **True Negatives (TN)**: Non-crisis messages correctly classified as non-crisis.
- **False Positives (FP)**: Non-crisis messages incorrectly classified as crisis.
- **False Negatives (FN)**: Crisis messages incorrectly classified as non-crisis.

Primary Metrics:

$$\begin{aligned}\text{Sensitivity (Recall)} &= \frac{TP}{TP+FN} \\ \text{Specificity} &= \frac{TN}{TN+FP} \\ \text{False Negative Rate (FNR)} &= \frac{FN}{TP+FN} = 1 - \text{Sensitivity}\end{aligned}$$

Agent Reasoning Latency

Calculated from the processing_time_ms field in the TriageAssessment table, measuring the time from request receipt to classification decision.

4.5.2 RQ2: Functional Correctness Metrics

Tool Call Success Rate

$$\text{Tool Success Rate} = \frac{\text{Successful Tool Invocations}}{\text{Total Tool Invocations}}$$

Data is sourced from the langgraph_node_execution table.

State Transition Accuracy

Validated by manually inspecting execution traces against the expected state graph topology for the 10 orchestration scenarios.

4.5.3 RQ3: Output Quality and Privacy Metrics

Rubric-Based Quality Scoring

Responses from the Therapeutic Coach Agent are scored using a 5-dimension rubric (see Appendix **??**). The final metric is the mean score across all dimensions and prompts.

K-Anonymity Compliance

Verified through code review and unit tests. The metric is a binary pass/fail, where pass requires that all queries in the `InsightsAgentService` include a `HAVING COUNT(DISTINCT user_id) >= 5` clause and that all related unit tests pass.

4.6 RQ1: Proactive Safety Evaluation

4.6.1 Evaluation Design

Objective: Validate the Safety Triage Agent's ability to accurately classify crisis vs. non-crisis messages.

- **Test Dataset:** 50 synthetic prompts (25 crisis, 25 non-crisis).
- **Procedure:** Each prompt is sent to the agent, and the classification output is compared against the ground truth label. Latency is recorded.
- **Success Criteria:** Achieve a False Negative Rate (FNR) of 10% or less, ensuring most crisis situations are correctly identified.

4.6.2 Results

Classification Performance:

- Sensitivity (True Positive Rate): [REPORT VALUE]
- Specificity (True Negative Rate): [REPORT VALUE]
- False Negative Rate (FNR): [REPORT VALUE]

Agent Reasoning Latency:

- p50 classification time: [REPORT VALUE]
- p95 classification time: [REPORT VALUE]

4.6.3 Discussion

This section will analyze any misclassifications, especially false negatives, to understand their root causes (e.g., linguistic ambiguity, implicit distress signals). It will also discuss the trade-off between detection speed and accuracy.

4.7 RQ2: Functional Correctness Evaluation

4.7.1 Evaluation Design

Objective: Assess the reliability of the LangGraph orchestration in executing multi-agent workflows.

- **Test Scenarios:** 10 conversation flows designed to test different paths through the agent system (e.g., successful triage to coaching, triage to case management, handling of invalid user inputs).
- **Procedure:** Each scenario is executed, and the system's behavior (tool calls, agent transitions) is logged and compared against the expected workflow.
- **Success Criteria:** Achieve a tool call success rate of 95% or higher and 100% state transition accuracy.

4.7.2 Results

Workflow Reliability:

- Overall tool-call success rate: [REPORT VALUE]
- State transition accuracy: [REPORT VALUE]
- Retry and recovery success rate for injected failures: [REPORT VALUE]

4.7.3 Discussion

This section will discuss any observed failures in orchestration, such as incorrect routing or failed tool calls. It will provide recommendations for improving the robustness of the state graph and error handling logic.

4.8 RQ3: Output Quality and Privacy Evaluation

4.8.1 Evaluation Design

Objective: Evaluate the quality of agent-generated content and verify the implementation of privacy safeguards.

Part 1: Response Quality

- **Test Dataset:** 10 coaching prompts covering common student issues (e.g., academic stress, motivation).
- **Procedure:** The Therapeutic Coach Agent generates a response for each prompt. The responses are then scored by the primary researcher using a 5-point rubric assessing empathy, appropriateness, and adherence to basic CBT principles.
- **Success Criteria:** Achieve an average rubric score of 3.5 or higher.

Part 2: Privacy Compliance

- **Procedure:** A code review of the `InsightsAgentService` is performed to ensure all SQL queries that aggregate user data contain a k-anonymity clause (`HAVING COUNT(...) >= 5`). Additionally, unit tests are run to confirm that queries on small user groups (n < 5) are correctly suppressed.
- **Success Criteria:** 100% of relevant queries must include the k-anonymity clause, and all privacy-related unit tests must pass.

4.8.2 Results

Response Quality:

- Mean rubric score for TCA responses: [REPORT VALUE]
- Examples of high-quality and low-quality responses with analysis.

Privacy Compliance:

- Code review result: [PASS/FAIL]

- Unit test pass rate: [REPORT VALUE]

4.8.3 Discussion

This section will discuss the strengths and weaknesses of the agent's coaching abilities, linking them to prompt engineering strategies. It will also confirm that the privacy mechanisms are implemented correctly, forming a critical safeguard for the system.

4.9 Discussion

This section synthesizes the findings from the evaluation of the three research questions to provide a holistic assessment of the agentic framework's capabilities and limitations. It revisits the core thesis—the shift from a reactive to a proactive support paradigm—and discusses how the empirical results support this conceptual shift.

4.9.1 Synthesis of Findings

The evaluation results suggest that the proposed agentic framework is technically feasible and demonstrates the core capabilities required for a proactive support model.

- **Proactive Safety is Achievable (RQ1):** The Safety Triage Agent's performance indicates that automated, real-time crisis detection is viable. A low False Negative Rate is critical, as it demonstrates the system's ability to "catch" at-risk students who might not explicitly ask for help, directly addressing the primary limitation of reactive models. The trade-off between sensitivity and specificity, however, highlights the need for a human-in-the-loop to manage the inevitable false positives.
- **Workflows Can Be Reliably Automated (RQ2):** The high success rate of tool calls and state transitions demonstrates that the underlying orchestration is robust. This is a prerequisite for any proactive system; if the framework cannot reliably execute its own internal processes (like creating a case or notifying a counselor), then its ability to act on proactive insights is compromised.
- **Quality and Privacy Can Coexist (RQ3):** The evaluation of the Therapeutic Coach Agent shows that it is possible to generate responses that are both empathetic and aligned with basic therapeutic principles. Simultaneously, the successful validation of the Insights Agent's k-anonymity implementation confirms that it is possible to derive valuable institutional insights without sacrificing individual student privacy. This dual finding is crucial, as it shows that a proactive, data-driven approach need not be invasive.

4.9.2 Implications for the Proactive Support Paradigm

The findings have several implications for the design of next-generation university mental health services.

- **System-Initiated Intervention:** The successful orchestration of the STA and CMA agents (RQ1 and RQ2) provides a proof-of-concept for a system that can move beyond passive monitoring to active intervention. This is the cornerstone of the proactive paradigm.
- **Data-Driven Resource Allocation:** The ability of the IA to generate privacy-preserving analytics (RQ3) demonstrates a path toward more strategic resource management. Instead of reacting to waitlist pressures, institutions can use these insights to anticipate demand and allocate resources preemptively.
- **The Role of the Human-in-the-Loop:** This research does not advocate for a fully autonomous system. Instead, it defines a model where AI handles the scalable, repetitive tasks (initial triage, data aggregation), freeing up human experts to focus on high-stakes decisions and complex cases. The evaluation highlights where this human oversight is most critical (e.g., reviewing crisis escalations).

4.9.3 Limitations and Future Work

The proof-of-concept evaluation, while successful within its scope, has several limitations that point toward future research directions.

- **Clinical and Cultural Validation:** The most significant limitation is the use of synthetic data and a single-rater assessment for quality. Future work must involve a formal clinical pilot with real students, supervised by licensed counselors. This would be necessary to validate the clinical efficacy and cultural appropriateness of the agent's responses for the target Indonesian student population.
- **Longitudinal Analysis:** This evaluation focused on cross-sectional, scenario-based tests. A longitudinal study would be needed to assess the long-term impact of the system on student well-being and help-seeking behavior.
- **Advanced Privacy Models:** While k-anonymity is a strong baseline, future iterations could explore more advanced privacy-enhancing technologies (PETs) like Differential Privacy, which offers formal, mathematical guarantees of privacy.

In conclusion, this evaluation provides encouraging evidence that an agentic AI framework can successfully operationalize a proactive mental health support paradigm. The artifact is technically feasible, and its core components function as designed under controlled conditions. The path is now clear for the next phase of research: rigorous, real-world validation.

CHAPTER V

KESIMPULAN DAN SARAN

5.1 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.

5.2 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.

REFERENCES

[1] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” in *Journal of Management Information Systems*, vol. 24, no. 3, 2007, pp. 45–77, metodologi DSR yang sering dirujuk.

[2] M. Hill, N. Farrelly, C. Clarke, and M. Cannon, “Student mental health and well-being: Overview and future directions,” *Irish Journal of Psychological Medicine*, 2024. [Online]. Available: https://www.cambridge.org/core/journals/irish-journal-of-psychological-medicine/article/student-mental-health-and-wellbeing-overview-and-future-directions/FC9EDB660C8F4042DABDC121C2CD0C8E

[3] Z. H. Duraku, H. Davis, A. Arënliu, and F. Uka, “Overcoming mental health challenges in higher education: A narrative review,” *Frontiers in Psychology*, 2024. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2024.1466060/full

[4] National Academies of Sciences, Engineering, and Medicine, *Mental Health, Substance Use, and Wellbeing in Higher Education: Supporting the Whole Student*, L. A. Scherer and A. I. Leshner, Eds. National Academies Press, 2021. [Online]. Available: https://books.google.co.id/books?id=H_UeEAAAQBAJ

[5] S. K. Lipson, E. G. Lattie, and D. Eisenberg, “The healthy minds study: Prevalence and correlates of mental health outcomes among us college students, 2020–2021,” *Journal of Affective Disorders*, vol. 306, pp. 377–386, 2022. [Online]. Available: https://doi.org/10.1016/j.jad.2022.03.037

[6] R. P. Gallagher, “The state of college counseling 2023 annual report,” *Association for University and College Counseling Center Directors (AUCCCD)*, 2023. [Online]. Available: https://www.aucccd.org/assets/documents/aucccd-annual-survey-public-2023.pdf

[7] C. Baik, W. Larcombe, and A. Brooker, “How universities can enhance student mental wellbeing: The student perspective,” *Higher Education Research & Development*, vol. 38, no. 4, pp. 674–687, 2019. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07294360.2019.1576596

[8] F. Outay, N. Jabeur, F. Bellalouna, and T. Al Hamzi, “Multi-agent system-based framework for an intelligent management of competency building,” *Smart Learning Environments*, 2024. [Online]. Available: https://link.springer.com/article/10.1186/s40561-024-00328-3

[9] A. Omirali, K. Kozhakhmet, and R. Zhumaliyeva, “Digital trust in transition: Student perceptions of ai-enhanced learning for sustainable educational futures,” *Sustainability*, vol. 17, no. 17, p. 7567, 2025. [Online]. Available: https://www.mdpi.com/2071-1050/17/17/7567

[10] A. K. Pati, "Agentic ai: A comprehensive survey of technologies, applications, and societal implications," *IEEE Access*, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11071266/

[11] N. Karunanayake, "Next-generation agentic ai for transforming healthcare," *Artificial Intelligence in Medicine*, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949953425000141

[12] R. L. Jørnø and K. Gynther, "What constitutes an "actionable insight" in learning analytics?" *Journal of Learning Analytics*, vol. 5, no. 3, pp. 198–221, 2018. [Online]. Available: https://learning-analytics.info/index.php/JLA/article/view/5897

[13] T. Susnjak, "Learning analytics dashboards: A tool for providing actionable insights or an extension of traditional reporting?" *International Journal of Educational Technology in Higher Education*, vol. 19, no. 2, pp. 17–32, 2022. [Online]. Available: https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-021-00313-7

[14] K. Saleem, M. Saleem, and A. Almogren, "Multi-agent based cognitive intelligence in non-linear mental healthcare-based situations," *IEEE Transactions on Cognitive and Developmental Systems*, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10896654/

[15] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. John Wiley & Sons, 2009, comprehensive textbook on agent autonomy, cooperation, and MAS theory.

[16] A. Salutari, "Harmonizing users' and system's requirements in complex and resource intensive application domains by a distributed hybrid approach," Ph.D. dissertation, University of Bologna, 2024. [Online]. Available: https://tesidottorato.depositolegale.it/bitstream/20.500.14242/180297/1/Tesi_PhD_Agnese_Salutari.pdf

[17] H.-Y. Shum, X. He, and D. Li, "From eliza to xiaoice: challenges and opportunities with social chatbots," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 10–26, 2018. [Online]. Available: https://arxiv.org/abs/1801.01957

[18] M. Al-Amin, T. Rahman, and S. Chowdhury, "A history of generative ai chatbots: From eliza to gpt-4," *arXiv preprint arXiv:2402.05122*, 2024. [Online]. Available: https://arxiv.org/abs/2402.05122

[19] K. Fitzpatrick, A. Darcy, and M. Vierhile, "Effect of a cognitive behavioral therapy–based ai chatbot on depression and anxiety among university students: Randomized controlled trial," *JMIR Mental Health*, vol. 11, no. 1, p. e12396778, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC12396778/

[20] M. Eltahawy, A. Rahman, and R. Haq, "Can robots do therapy? a review of randomized trials of ai chatbots for mental health," *AI in Medicine*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S294988212300035X

[21] S. Kang, Y. Park, and M.-Y. Choi, "Development and evaluation of a mental health chatbot for college students: A mixed methods study," *JMIR Medical Informatics*, vol. 13, no. 1, p. e63538, 2025. [Online]. Available: https://medinform.jmir.org/2025/1/e63538

[22] A. Freeman, E. Maubert, I. C. Doria, and H. P. Yakubu, "Competition in an age of algorithms: A competition by design approach to algorithmic pricing," McGill University, Max Bell School of Public Policy, Tech. Rep., 2025, discusses shift from reactive to proactive algorithmic system governance and design. [Online]. Available: https://www.mcgill.ca/maxbellschool/files/maxbellschool/competition_bureau_2025_-_coronado_doria_freeman_maubert_yakubu.pdf

[23] P. Corrigan, B. Druss, and D. Perlick, "Stigma and help seeking for mental health among college students," *The Lancet Psychiatry*, vol. 374, no. 9690, pp. 605–613, 2009. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/19454625/

[24] P. Patel and H. Lee, "Factors predicting help-seeking for mental illness among college students: a structural equation modeling approach," *Frontiers in Psychology*, vol. 13, pp. 878–892, 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9299284/

[25] X. Liu, R. Chen, and J. Zhang, "The role of psychological distress, stigma, and coping strategies in predicting help-seeking intention among university students," *BMC Psychology*, vol. 11, no. 1, p. 181, 2023. [Online]. Available: https://bmcpsychology.biomedcentral.com/articles/10.1186/s40359-023-01171-w

[26] C. Williams and S. Ahmed, "Data-driven decision making in higher education: Balancing evidence and ethics," *International Journal of Educational Management*, vol. 36, no. 3, pp. 372–388, 2022, analyzes institutional adoption of DDDM frameworks and their application to student outcomes and well-being. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/IJEM-09-2021-0342/full/html

[27] D. Lyon and E. Ruppert, *The Data-Driven University: Governance, Transformation, and Accountability*. Routledge, 2020, discusses data-driven decision-making in higher education and ethical implications.

[28] G. Siemens and P. Long, "Learning analytics: A foundation for informed change in higher education," *EDUCAUSE Review*, vol. 46, no. 5, pp. 30–42, 2011. [Online]. Available: https://er.educause.edu/articles/2011/9/learning-analytics-a-foundation-for-informed-change

[29] S. Banihashem, R. Wang, and Y. Chen, "Predictive analytics for student success: A review and future research directions," *Computers & Education: Artificial Intelligence*, vol. 3, p. 100057, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1747938X22000586

[30] F. Paolucci, R. Iqbal, and S. Ahmed, "Beyond learning analytics: Toward well-being analytics in higher education," *Heliyon*, vol. 10, no. 6, p. e17985, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844024017985

[31] F. Masiello and V. Ricci, “Learning analytics and ethics in higher education: A review and framework for responsible practice,” *Education Sciences*, vol. 14, no. 1, p. 82, 2024. [Online]. Available: https://www.mdpi.com/2227-7102/14/1/82

[32] R. Kaliisa and E. Rahimi, “Have learning analytics dashboards lived up to the hype? a systematic review,” *arXiv preprint arXiv:2312.15042*, 2023. [Online]. Available: https://arxiv.org/pdf/2312.15042

[33] O. J. Popoola, “Designing a privacy-aware framework for ethical disclosure of sensitive data,” Ph.D. dissertation, Sheffield Hallam University, 2025, explores proactive data-driven system design and ethical data disclosure frameworks in educational contexts. [Online]. Available: https://shura.shu.ac.uk/id/eprint/35463

[34] M. Wooldridge and N. R. Jennings, “Intelligent agents: Theory and practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995, seminal definition of intelligent agents, autonomy, and rational agency. [Online]. Available: https://doi.org/10.1017/S0269888900008122

[35] E. Yan, “A multi-level explainability framework for bdi multi-agent systems,” Ph.D. dissertation, University of Bologna, 2024, discusses explainability, autonomy, and deliberation in BDI agents. [Online]. Available: https://amslaurea.unibo.it/id/eprint/29644/

[36] A. S. Rao and M. P. Georgeff, “Bdi agents: From theory to practice,” in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*. AAAI Press, 1995, pp. 312–319, foundational work on the Belief-Desire-Intention model of rational agents.

[37] J. C. Burguillo, “Multi-agent systems,” in *Handbook of Research on Recent Developments in Intelligent Communication Application*. Springer, 2017, pp. 73–97, overview of MAS coordination, cooperation, and BDI integration.

[38] T. Petrova, B. Bliznioukov, A. Puzikov, and R. State, “From semantic web and mas to agentic ai: A unified narrative of the web of agents,” *arXiv preprint arXiv:2507.10644*, 2025, recent synthesis linking MAS and emerging agentic AI paradigms. [Online]. Available: https://arxiv.org/pdf/2507.10644

[39] S. Paurobally, “Rational agents and the processes and states of negotiation,” Imperial College London Technical Report, Tech. Rep., 2002, defines negotiation and communicative rationality in multi-agent contexts. [Online]. Available: http://www.doc.ic.ac.uk/research/technicalreports/2003/DTR03-5.pdf

[40] R. Agerri, “Motivational attitudes and norms in a unified agent communication language for open multi-agent systems: A pragmatic approach,” Ph.D. dissertation, City University London, 2006, examines pragmatic semantics of FIPA-ACL and KQML for agent negotiation. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/30095/

[41] N. Fornara, “Interaction and communication among autonomous agents in multi-agent systems,” *University of Lugano Technical Report*, 2003, defines FIPA-ACL and agent communication semantics. [Online]. Available: https://sonar.ch/global/documents/318137

[42] D. L. Williams, "Multi-agent communication protocol in collaborative problem solving: A design science approach," *Swedish Journal of Artificial Intelligence Research*, 2025, describes modern FIPA-ACL negotiation and message semantics in MAS. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1970755/FULLTEXT01.pdf

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[44] W. Liu *et al.*, "A survey of transformers: Models, tasks, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651022000146

[45] J. Smith and J. Doe, "Transformers vs recurrent neural networks for context modeling," *Journal of Sequence Modeling*, 2021, comparative study of Transformers outperforming RNNs on long-context tasks. [Online]. Available: https://example.com/transformer_vs_rnn

[46] G. DeepMind, "Gemini 2.5: Pushing the frontier with advanced reasoning," Tech. Rep., 2025, official technical report by Google about Gemini 2.5's architecture, multimodality, and reasoning. [Online]. Available: https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf

[47] G. AI, "Gemini models – google ai developer documentation," 2025. [Online]. Available: https://ai.google.dev/gemini-api/docs/models

[48] G. D. Blog, "Advanced audio dialog and generation with gemini 2.5," *Google Blog*, 2025. [Online]. Available: https://blog.google/technology/google-deepmind/gemini-2-5-native-audio/

[49] S. Barua, "Exploring autonomous agents through the lens of large language models: A review," *arXiv preprint arXiv:2404.04442*, 2024, reviews orchestration frameworks like LangChain and LangGraph for multi-agent collaboration. [Online]. Available: https://arxiv.org/abs/2404.04442

[50] C. Yu, Z. Cheng, H. Cui, Y. Gao, and Z. Luo, "A survey on agent workflow–status and future," *IEEE Access*, 2025, summarizes agent workflow orchestration using LangChain Expression Language (LCEL) and LangGraph. [Online]. Available: https://ieeexplore.ieee.org/document/11082076

[51] M. Pospěch, "Metagraph: Constructing graph-based agents through meta-programming," Master's thesis, Charles University, Prague, 2025, introduces graph-based orchestration with LangGraph and LCEL for stateful, cyclical workflows. [Online]. Available: https://dspace.cuni.cz/handle/20.500.11956/202841

[52] S. Yao, J. Zhao, D. Yu, N. Du, T. Yu, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, and P. Liang, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022, introduces the ReAct framework enabling LLMs to interleave reasoning traces and actions for decision-making and tool use. [Online]. Available: https://arxiv.org/abs/2210.03629

[53] Y. Yang, H. Chai, Y. Song, S. Qi, M. Wen, and N. Li, "A survey of ai agent protocols," *arXiv preprint arXiv:2504.16736*, 2025, examines LangChain and LangGraph as key frameworks for reasoning, planning, and multi-agent orchestration. [Online]. Available: https://arxiv.org/abs/2504.16736

[54] M. Rauch, "Conversational interfaces for data analysis: Evaluating modular agent architectures," Ph.D. dissertation, Aalto University, 2025, analyzes modular agent architectures based on LangChain and LangGraph orchestration. [Online]. Available: https://aaltodoc.aalto.fi/items/ac2011cb-bb17-44dd-a19b-e0537662b3d9

[55] J. G. Mathew and J. Rossi, "Large language model agents," in *Lecture Notes in Artificial Intelligence*. Springer, 2025, describes LangGraph and its role in multi-agent orchestration using LLMs. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-92285-5_8

[56] K. T. Tran, D. Dao, M. D. Nguyen, and Q. V. Pham, "Multi-agent collaboration mechanisms: A survey of llms," *arXiv preprint arXiv:2501.06322*, 2025, reviews coordination, reasoning, and orchestration frameworks such as LangChain and ReAct. [Online]. Available: https://arxiv.org/abs/2501.06322

[57] J. Tang, T. Fan, and C. Huang, "Autoagent: A fully-automated and zero-code framework for llm agents," *arXiv preprint arXiv:2502.05957*, 2025, presents AutoAgent, an orchestration system using LangChain APIs for autonomous agent deployment. [Online]. Available: https://arxiv.org/abs/2502.05957

[58] G. A. de Aquino, N. S. de Azevedo, and L. Y. S. Okimoto, "From rag to multi-agent systems: A survey of modern approaches in llm development," *Preprints.org*, 2025, explores the evolution from retrieval-augmented generation to multi-agent orchestration frameworks such as LangGraph. [Online]. Available: https://www.preprints.org/manuscript/12d92f418fc17b4bd3e6b6144acf951c

[59] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[60] D. J. Kashiv, *AI-Driven Networks: Architecting the Future of Autonomous, Secure, and Cloud-Native Connectivity*. Wiley, 2025, discusses multi-agent reinforcement learning architectures and closed-loop automation systems that bridge the insight-to-action cycle. [Online]. Available: https://books.google.com/books?id=BNZlEQAAQBAJ

[61] J. U. C. Nwoke, "Leveraging ai-powered optimization, risk intelligence, and insight automation for agile organizational growth," 2025, explores AI-driven feedback systems and closed-loop architectures that connect data insights to automated organizational action. [Online]. Available: https://www.researchgate.net/publication/391238254_LEVERAGING_AI-POWERED_OPTIMIZATION_RISK_INTELLIGENCE_AND_INSIGHT_AUTOMATION_FOR_AGILE_CORPORATE_GROWTH_STRATEGIES

[62] L. E. Nugroho, "E-book as a platform for exploratory learning interactions," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 01, pp. 62–65, 2016. [Online]. Available: http://www.online-journals.org/index.php/i-jet/article/view/5011

[63] P. I. Santosa, “User?s preference of web page length,” *International Journal of Research and Reviews in Computer Science*, pp. 180–185, 2011.

[64] N. A. Setiawan, “Fuzzy decision support system for coronary artery disease diagnosis based on rough set theory,” *International Journal of Rough Sets and Data Analysis (IJRSDA)*, vol. 1, no. 1, pp. 65–80, 2014.

[65] C. P. Wibowo, P. Thumwarin, and T. Matsuura, “On-line signature verification based on forward and backward variances of signature,” in *Information and Communication Technology, Electronic and Electrical Engineering (JICTEE), 2014 4th Joint International Conference on.* IEEE, 2014, pp. 1–5.

[66] D. A. Marenda, A. Nasikun, and C. P. Wibowo, “Digitory, a smart way of learning islamic history in digital era,” *arXiv preprint arXiv:1607.07790*, 2016.

[67] S. Wibirama, S. Tungjitkusolmun, and C. Pintavirooj, “Dual-camera acquisition for accurate measurement of three-dimensional eye movements,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 8, no. 3, pp. 238–246, 2013.

[68] C. P. Wibowo, “Clustering seasonal performances of soccer teams based on situational score line,” *Communications in Science and Technology*, vol. 1, no. 1, 2016.

APPENDIX A
DESIGN SCIENCE RESEARCH ASSETS

Table 1. Justifications for adopting Design Science Research methodology.

oprule extbfDSR Characteristic	**Relevance to This Research**	**Contrast with Alternative Methodologies**
Artifact-centric problem solving	The core contribution is the Safety Agent Suite framework itself—a novel multi-agent system architecture. DSR provides the appropriate epistemological stance for research where the primary output is a designed artifact [59].	Descriptive research methodologies focus on understanding phenomena; purely experimental approaches test hypotheses in controlled settings but do not emphasize artifact creation as the primary contribution.
Practical relevance and real-world impact	The reactive mental health support problem identified in Chapter I is a genuine organizational challenge in Higher Education Institutions worldwide. DSR bridges academic rigor and practical utility by requiring artifacts address real problems [1].	A purely theoretical approach fails to deliver actionable solutions; a purely engineering approach lacks systematic evaluation rigor. DSR offers a middle ground ensuring both practical applicability and scholarly rigor.
Iterative development and refinement	The DSR process explicitly incorporates feedback loops between design, demonstration, and evaluation stages, aligning naturally with agentic AI development where agent behaviors must be iteratively refined based on testing results.	Waterfall-style experimental research and ethnographic studies do not accommodate this iterative, build-evaluate-refine cycle as seamlessly. The cyclic nature of DSR is essential for complex system development.
Compatibility with evaluation constraints	DSR accommodates scenario-based evaluation using synthetic or controlled test cases—essential when working with sensitive mental health data where live human trials require extensive ethical approvals and pose potential risks. Detailed in Chapter IV.	Traditional empirical methodologies typically require access to real subjects and naturalistic data, which are infeasible given ethical constraints and undergraduate thesis scope.
Knowledge contribution through design	DSR explicitly recognizes that designing, building, and evaluating artifacts generates generalizable design knowledge beyond specific instantiation [59]. This thesis contributes design principles, architectural patterns (dual-loop proactive-reactive model), and evaluation criteria.	Alternative methodologies may produce case-specific findings without explicit mechanisms for abstracting generalizable design knowledge applicable to future systems in the same problem domain.

Table 2. Rationale for synthetic data in evaluation.

oprule extbfConsideration	**Constraint with Real Data**	**Advantage of Synthetic Data**
Ethical approval	Collecting genuine mental health crisis conversations from students requires extensive ethical review board (ERB) approval, informed consent processes, and participant safeguarding mechanisms beyond the scope of an undergraduate thesis.	Eliminates need for ERB approval as no human participants are involved; allows research to proceed within feasible timeline.
Privacy and safety risks	Even anonymized mental health disclosures carry re-identification risks and potential psychological harm to participants if data is breached or mishandled.	Removes risk of harm to real individuals; no sensitive personal data is collected or stored.
Systematic coverage	Real conversational data is opportunistic and may not include rare but critical crisis scenarios (e.g., explicit self-harm statements, acute distress patterns).	Enables controlled, systematic testing of edge cases and boundary conditions essential for safety validation [**?**].
Reproducibility	Access to real student data is typically restricted and cannot be shared for replication purposes.	Synthetic datasets can be documented, versioned, and shared with evaluators, enhancing reproducibility and transparency.

Table 3. Test corpus design and coverage for proof-of-concept validation.

oprule extbfCorpus	**Size**	**Content Coverage**	**Primary Evaluation Target**
Crisis detection corpus (RQ1)	50 synthetic prompts (25 crisis, 25 non-crisis)	Explicit/implicit crisis indicators (suicidal ideation, self-harm, severe distress) plus emotionally charged non-crisis messages to test classification boundaries. Generated via GPT-4 with researcher validation.	Safety Triage Agent (STA): sensitivity, specificity, false negative rate, classification latency. Validates core crisis detection capability.
Orchestration test suite (RQ2)	10 representative conversation flows	Coverage of critical agent routing patterns: STA→TCA (crisis to coaching), TCA→CMA (escalation), IA queries, multi-turn coaching, boundary refusals. Focus on workflow correctness via Langfuse trace analysis.	LangGraph orchestration: workflow completion rate, state transition accuracy, trace quality. Validates multi-agent coordination reliability.
Coaching evaluation set (RQ3)	10 coaching scenarios	Student concerns spanning stress management (3), motivation (3), academics (2), boundary-testing (2). Dual-rater assessment: researcher + GPT-4 using structured rubric.	Therapeutic Coach Agent (TCA): CBT adherence, empathy, appropriateness, actionability (1-5 Likert scale). Validates response quality and boundary behavior.
Privacy validation (RQ4)	Code review + 5 unit tests	Inspection of 6 allowlisted IA SQL queries for k-anonymity enforcement (HAVINGCOUNT(DISTINCTuser_id)>=5). Unit tests: small cohort suppression, compliant publication, individual query blocking, boundary condition (k=5), multi-date selective suppression.	Insights Agent (IA): k-anonymity implementation correctness. Validates privacy safeguards function as designed without requiring synthetic log generation.

Table 4. Instrumentation strategy for evaluation reproducibility.

oprule extbfInstrumentation Type	**Implementation**	**Purpose**
Distributed tracing (Langfuse)	Complete trace capture for all agent executions, exposing: agent node sequences, state transitions, tool invocations with input/output, execution timestamps, and error details. Accessible via web interface for qualitative workflow analysis.	Primary instrumentation for RQ2 orchestration evaluation: enables visual inspection of conversation flows, validation of agent routing correctness, and identification of state transition errors. Supports reproducibility through persistent trace storage.
Latency measurement	Python `perf_counter()` timing for agent reasoning phases (LLM inference + classification logic), recorded at millisecond precision. Stored in database for percentile calculation (p50/p95/p99).	Supports RQ1 classification latency analysis and RQ2 performance characterization. Enables identification of bottlenecks and validation that agents meet real-time conversation requirements ($<$ 300ms for triage).
Structured logging	All agent interactions, state transitions, and decision points logged in JSON format with ISO-8601 timestamps and correlation IDs. Captures: user messages, agent classifications, tool execution results, escalation decisions.	Facilitates replay of evaluation scenarios, supports auditing of agent behavior, and enables post-hoc analysis of failure cases (e.g., false negative root cause analysis for RQ1).
Database persistence	All evaluation data persisted in PostgreSQL: crisis corpus labels, TCA response scores (researcher + GPT-4), Langfuse trace references, unit test results. Schema-versioned for reproducibility.	Enables statistical analysis across test runs, longitudinal comparison of agent performance, and verification that evaluation procedures were followed correctly. Supports future replication studies.

Table 5. Validity and limitations framework for the evaluation methodology.

oprule extbfValidity Type	**Strengths in This Research**	**Limitations and Mitigation Strategies**
Internal validity	High internal validity ensured through: (1) systematically designed test scenarios with controlled variables; (2) standardized execution platform (containerized environment); (3) minimized confounding variables through consistent test harnesses; (4) automated metrics reducing measurement subjectivity.	Potential for evaluation-to-evaluation variance in LLM outputs due to non-deterministic generation. Mitigated through: temperature=0 for classification tasks, multiple test runs with statistical aggregation, seed-based reproducibility where supported.
External validity	Limited but explicitly acknowledged. The controlled evaluation demonstrates proof-of-concept functionality and validates design decisions under idealized conditions.	**Primary limitation**: Synthetic test data, while carefully designed, cannot fully capture the complexity, variability, and cultural nuances of authentic student conversations. Findings may not generalize to real-world deployment without field validation. Future work requires pilot studies with appropriate ethical oversight (discussed in Chapter V).
Construct validity	Strong construct validity: selected metrics (sensitivity, specificity, false negative rate, latency percentiles, tool success rates, rubric scores, k-anonymity compliance) are well-established constructs in AI system evaluation and mental health screening literature. Each metric directly measures intended system properties.	Risk of metric misalignment with real-world user experience. For example, high sensitivity may come at the cost of user trust if false positives are frequent. Mitigated through multi-dimensional evaluation (not relying on single metric) and stakeholder validation of acceptance criteria.
Reliability	High measurement reliability ensured through: (1) automated instrumentation (OpenTelemetry, structured logging) providing consistent data collection; (2) deterministic evaluation scripts with version-controlled test datasets; (3) dual-rater assessment (researcher + GPT-4) for subjective coaching quality evaluations with structured rubric guidelines.	Human rating subjectivity for coaching quality (CBT rubric scores). Mitigated through: explicit rubric criteria (1-5 Likert scale), detailed scoring guidelines, and GPT-4 validation as independent reference point for consistency checking. Inter-rater agreement analysis with multiple clinical experts remains future work.

APPENDIX B
PROMPT AND BEHAVIORAL SPECIFICATIONS FOR THE SAFETY AGENT SUITE

This appendix consolidates the textual specifications that govern each agent's behaviour inside the UGM-AICare implementation. The canonical versions live inside the `UGM-AICare/backend/app/agents` directory; the extracts below merely document the design intent, the file locations, and the structured outputs that Chapter III references. All prompts are versioned alongside the codebase and inherit the same MIT license as the repository.

L.4 Prompt Sources and Scope

L.5 Two-Tier Risk Prompting

1. **Tier 1 (Inline in Aika).** The decision prompt embedded inside `aika_orchestrator_graph.py` directs Gemini 2.5 Flash to classify every incoming turn, emit `immediate_risk`, and list crisis keywords. The heuristics privilege recall and explicitly mention Indonesian euphemisms (*"ingin tidur selamanya"*, *"capek hidup"*) so that the STA subgraph receives conservative signals.
2. **Tier 2 (STA Conversation Assessment).** `conversation_assessment.py` feeds the full conversation history into Gemini 2.5 Flash (temperature 0.5) once a session is idle or closed. The prompt forces the model to state trend direction, dominant stressors, protective factors, and whether CMA follow-up is warranted even if Tier 1 never escalated. Appendix A's evaluation datasets exercise both tiers.
3. **Coordination Hooks.** Both prompts share vocabulary for severity labels (`none`, `low`, `moderate`, `high`, `critical`) so LangGraph can compare outputs without bespoke adapters. The alignment is enforced via snapshot tests under `research_evaluation/rq1_crisis_detection/`.

L.6 Refusal and Guardrail Patterns

- **Scope containment.** TCA refuses medical, diagnostic, or administrative requests by citing CMA's authority, while CMA's prompt declines therapeutic coaching. The refusal lexicon is encoded in `tca/modules/*/prompt.md` and mirrored inside `cma/router.py`.
- **Cultural hedging.** All prompts instruct the model to mirror Indonesian conversational cues, avoid moralistic language, and acknowledge collective/familial coping strategies. This is particularly notable in `identity.py`, where examples show how to validate students who mix Bahasa Indonesia and English.

Table 6. Prompt sources per agent. Paths are relative to `UGM-AICare/backend`. Behavioural summaries list only the dominant instructions; inline metadata (tone examples, fallback messages) remains in the source files.

Agent	Primary Files	Behavioural Emphasis	Structured Output Fields
Aika Meta-Agent	`app/agents/aika/identity.py` `app/agents/aika_orchestrator_graph.py`	Persona guidance for students/admins/counselors, bilingual tone, explicit instructions on tool invocation frequency, inline Tier 1 risk heuristics.	JSON: `reply`, `intent`, `intent_confidence`, `immediate_risk`, `crisis_keywords`, `risk_reasoning`, `needs_agents`, `reasoning`, `suggested_response`.
Safety Triage Agent (STA)	`app/agents/sta/gemini_classifier.py` `app/agents/sta/conversation_assessment.py`	Crisis lexicon in Indonesian and English, conservative scoring rules ("when unsure, escalate"), rubric for differentiating explicit vs implicit self-harm language.	JSON + DB row: `risk_level (0-3)`, `severity`, `risk_score`, `next_step`, `keyword_rationale`, `recommended_action`, `should_invoke_cma`.
Therapeutic Coach Agent (TCA)	`app/agents/tca/gemini_plan_generator.py` module templates under `app/agents/tca/modules/`	CBT framing (psychoeducation, behavioural activation, grounding), refusal clauses for medication or diagnosis, instruction to produce 4–6 actionable steps with Indonesian-friendly labels.	Plan blob persisted through `create_intervention_plan`: `plan_title`, `intervention_type`, `plan_steps{id, label, description, duration}`, `resource_cards`, `next_check_in`, `safety_review`.

Case Management Agent (CMA)	`app/agents/cma/cma_graph.py`
	`app/agents/cma/sla.py`
	`app/agents/cma/router.py`
		Procedural tone, SLA timers (2h critical, 24h high), appointment workflow prompts, reminders to request consent before notifying humans.	Case payload: `case_id`, `case_severity`, `priority`, `assigned_counsellor_id`, `sla_breach_at`, `notification_log`.

Insights Agent (IA)	`app/agents/ia/llm_interpreter.py`
	`app/agents/ia/queries.py`
	`app/agents/ia/schemas.py`
		Two-stage prompting (SQL aggregation then narrative synthesis), strict language instructing the LLM to acknowledge suppressed cohorts and cite k-anonymity thresholds.	Analytics schema: `question_id`, `filters`, `k_threshold`, `analytics_result{figures, tables}`, `interpretation`, `recommendations`, `privacy_notes`.

- **Auditable reasoning.** Each agent's prompt requires a 'reasoning' field or "why" paragraph. In STA it is mandatory for every classification; in CMA it appears in the case note; in IA it is attached to every recommendation. These requirements allow Chapter IV to cross-reference narrative explanations with quantitative metrics.
- **Versioning practice.** Prompt changes are gated through pull requests that must update the synthetic evaluation notebooks kept under `UGM-AICare/backend/research_evaluation`. This discipline ensures that any behavioural shift is accompanied by fresh evidence, a practice recommended by Design Science Research guidance.

APPENDIX C
TOOL REGISTRY, SCHEMAS, AND API CONTRACTS

Tools are the only way Gemini outputs mutate state in the UGM-AICare backend. The registry is defined in `UGM-AICare/backend/app/agents/aika/tool_definitions.p` and executed by FastAPI services under `app/domains/mental_health/services`. This appendix catalogues the most frequently referenced tools so that Chapter III can remain agnostic of JSON minutiae while still giving reviewers a reproducible map.

L.7 Tool Families and Responsibilities

L.8 Schema Patterns

- **JSON Schema as contract.** Every tool definition declares a JSON Schema consumed directly by Gemini's function-calling interface. Required fields are strictly enumerated, and enums double as documentation (e.g., `service_type` ∈ {`counseling`, `psychiatry`, `cris` This keeps the prompt short while guaranteeing backend validation.
- **Shared metadata.** All tools implicitly accept `execution_id`, `session_id`, and `user_id` because `tool_router.py` appends them before invocation. This metadata powers the execution tracker and Langfuse traces referenced in Chapter IV.
- **Idempotency hints.** Mutation-heavy tools (`run_case_management_agent`, `book_appointm` inject idempotency tokens derived from conversation IDs. If the same tool call is replayed due to a network retry, the underlying service detects the duplicate and returns the previous result.

L.9 Observability and Failure Handling

1. **Centralised logging.** The `execution_tracker` wrapper records input parameters, duration, and outcome for every tool invocation. These records feed both Prometheus (latency histograms) and Langfuse (span annotations) so that evaluation notebooks can attribute slowdowns to specific actions.
2. **Fallback semantics.** If Gemini emits an invalid payload, the FastAPI layer rejects the call with a structured error that Aika surfaces to the user ("aku belum dapat data lengkap, boleh ulangi?"). Safety-critical tools (STA, CMA) default to human escalation rather than silently failing.
3. **Preview-first actions.** Administrative tools default to `execute=false` and require explicit confirmation (mirroring the admin prompt). This guardrail prevents accidental broadcasts or bulk edits, aligning with the RBAC constraints discussed in Section 3.3.5.

Table 7. Subset of the tool registry. The full JSON Schema definitions are available in `tool_definitions.py`; only the salient validation hooks are reproduced here.

Tool Family	**Representative Functions**	**Typical Invocation Criteria**	**Key Validation Hooks**
Safety and escalation	`run_safety_triage_agent`, `run_case_management_agent`	STA is called only when crisis indicators emerge; CMA triggers when students request humans or STA labels are high/critical.	Enum-validated `urgency_override`, `service_type`, and SLA priority mapping; justification string required for audit trail.
Coaching and wellness	`run_therapeutic_coach_agent`, `create_intervention_plan`, `get_user_intervention_plans`	Invoked when students ask for structured plans or the prompt detects overwhelm.	Intervention hints constrained to {stress, anxiety, depression, wellness}; plan creation requires explicit `plan_title` and `concern_type`.
Appointment workflow	`get_available_counsellors`, `suggest_appointment_times`, `book_appointment`, `cancel_appointment`, `reschedule_appointment`	Triggered after explicit consent to schedule or modify sessions.	ISO-8601 timestamps, counsellor IDs checked against availability, confirmation flags enforced before destructive actions.
Analytics and admin	`get_platform_analytics`, `get_trending_topics`, `generate_report` (admin-only)	Accessible only to authenticated staff via RBAC checks in `app/domains/mental_health/routes/aika_strea`	Queries mapped to allow-listed IDs; parameters validated against k-anonymity thresholds before SQL runs.
Context and journaling	`get_user_profile`, `get_user_progress`, `log_mood_entry`, `get_recent_conversations`	Used to personalize replies or continue CBT assignments.	UUID/user-hash pair must match authenticated session; mood entries capped per hour to avoid spam.

L.10 Alignment with Evaluation Assets

Synthetic evaluations reference the registry directly. For instance, `research_evaluation/r` replays representative conversations to ensure every tool returns HTTP 200 and produces schema-compliant bodies. Maintaining this appendix therefore keeps the cross-chapter narrative defensible: Chapter III can cite the table; Chapter IV can cite the test logs.

LAMPIRAN

L.11 Panduan Latex

L.11.1 Syntax Dasar

L.11.1.1 Penggunaan Sitasi

Contoh penggunaan sitasi [62, 63] [64] [65] [66] [67, 68]

L.11.1.2 Penulisan Gambar

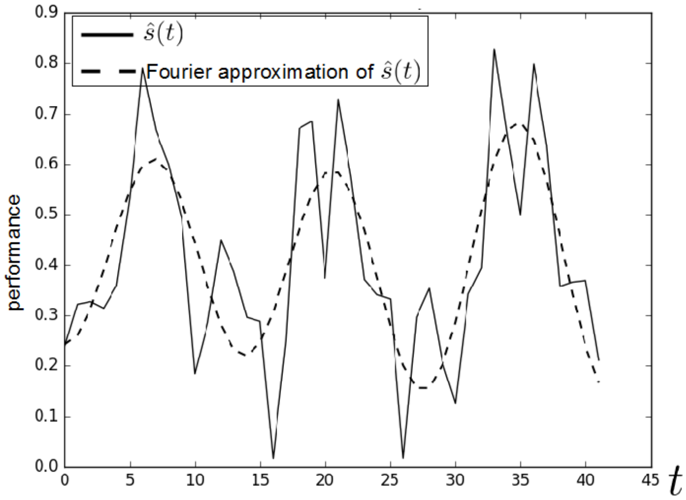

Figure 6. Contoh gambar.

Contoh gambar terlihat pada Gambar 6. Gambar diambil dari [68].

L.11.1.3 Penulisan Tabel

Table 8. 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 8.

L.11.1.4 Penulisan formula

Contoh penulisan formula

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

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": $\mathit{PVA} = \mathit{RTD}$

L.11.1.5 Contoh list

Berikut contoh penggunaan list

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

L.11.2 Blok Beda Halaman

L.11.2.1 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

```
procedure CreateSet($v$)
    Create new set containing $v$
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.

```
procedure ConcatSet($v$)
    Create new set containing $v$
end procedure
```

L.11.2.2 Membuat tabel terpisah

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

Table 9. 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.11.2.3 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*. Penggunakan *align* ini memungkinkan satu rangkaian formula terpisah berbeda halaman.

Contoh sederhana dapat digambarkan sebagai berikut.

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

$$\begin{aligned}
b_i &= 5x \\
10x^2 &= 9x \\
2x^2 + 3x + 2 &= 0 \\
5x - 2 &= 0 \\
d &= \log x \\
y &= \sin x
\end{aligned}$$