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

SKRIPSI

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

Written by:

GIGA HIDJRIKA AURA ADKHY
21/479228/TK/52833

BACHELOR'S DEGREE IN INFORMATION ENGINEERING

PROGRAM
DEPARTMENT OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY
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 Engineering and Information Technology
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

Tugas akhir ini kupersembahkan kepada kedua orang tuaku. Kupersembahkan pula kepada keluarga dan teman-teman semua, serta untuk bangsa, negara, dan agamaku.

[contoh]

PREFACE

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

1. <isi dengan nama Kadep>
2. <isi dengan nama Sekdep>
3. <isi dengan nama Dosen Pembimbing>
4. Kedua Orang Tua, kakak, dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
5. <isi dengan nama orang lainnya>

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

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

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

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

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

Contoh Abstrak Teknik Elektro:

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

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

Contoh Abstrak Teknik Biomedis:

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

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

Contoh Abstrak Teknologi Informasi:

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

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

ABSTRACT

The provision of mental health support in Higher Education Institutions (HEIs) is often hampered by a reactive model, limited scalability, and inefficient resource allocation. This research addresses these challenges by proposing a novel agentic AI framework designed to enable a proactive, data-driven approach to student well-being. The framework is comprised of three collaborative agents: an ***Analytics Agent*** *for trend identification, an* ***Intervention Agent*** *for automated outreach, and a* ***Triage Agent*** *for efficient resource routing.*

We designed and implemented a functional prototype of this framework utilizing a hybrid architecture that combines a locally-hosted Large Language Model (LLM) managed by LangChain within a FastAPI backend, orchestrated by the n8n workflow automation platform. The prototype's feasibility was validated through three testing scenarios demonstrating its capability to autonomously generate insight reports, manage intervention workflows, and perform initial triage. The results indicate that this agentic framework presents a viable pathway for transforming university mental health services, offering significant potential to improve operational efficiency and enable proactive, evidence-based decision-making.

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

CHAPTER I
INTRODUCTION

1.1 Background

Higher Education Institutions (HEIs) are facing a critical and growing challenge in supporting student well-being [1]. 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 [1]. 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 [1].

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 [1]. Consequently, a critical gap persists between the need for mental health services and their actual provision, leaving many students without timely support [1].

To bridge this gap, a paradigm shift from a reactive to a **proactive** support model is imperative [1]. The engine for this evolution is **Digital Transformation**, a process that leverages technology to fundamentally reshape organizational processes and enhance value delivery within HEIs [1]. 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 [1].

This research 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 [1], representing a new frontier in educational technology [1]. We propose that a framework built upon a system of collaborative intelligent agents, a Multi-Agent System (MAS), a concept already explored for smart campus management [1], 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 thesis details the design, development, and evaluation of such a framework, prototyped within the UGM-AICare project.

1.2 Problem Formulation

The inefficiency and reactive nature of current university mental health support systems present a complex problem. This research addresses the following core challenges:

1. The primary challenge is the **design of an agentic AI framework** capable of automating key institutional processes, specifically in the areas of trend analysis, proactive intervention, and initial user triage.
2. A significant technical challenge lies in the **design and implementation of a robust, modular, and hybrid architecture** to realize this framework, requiring the integration of local Large Language Models (e.g., self-hosted Gemma 3), cloud-based models (e.g., Gemini through AI Studio), and workflow automation platforms (e.g., n8n).
3. Finally, there is a need to **evaluate the potential impact** of such a framework on institutional operational efficiency and data-driven decision-making, which will be validated through a proof-of-concept prototype and scenario-based testing.

To address these challenges, this thesis proposes and details a framework comprised of three specialized, collaborative intelligent agents: an **Analytics Agent**, an **Intervention Agent**, and a **Triage Agent**.

1.3 Objectives

The primary objectives of this thesis are:

1. To design the conceptual and technical framework for the agentic AI system.
2. To implement a functional proof-of-concept prototype.
3. To evaluate the prototype's capabilities against predefined functional scenarios.

1.4 Scope and Limitations

To ensure the feasibility and focus of this research, the following boundaries are established:

1. This research is focused on the **design and prototype implementation** of the agentic AI framework, not a full-scale, university-wide deployment.
2. The evaluation of the framework is based on **functional, scenario-based testing** of the prototype's capabilities. It does not measure the long-term psychological impact on students or the real-world operational savings for the institution.
3. The data utilized for testing the analytics agent will consist of anonymized, pre-

existing chat logs or simulated data to ensure user privacy and controlled testing conditions.

4. The research will not address the ethical implications of AI in education, such as bias in AI algorithms or the potential for misuse of student data. These are acknowledged as important issues but are outside the scope of this thesis.

1.5 Contributions

1. Academic: A novel framework for applying agentic AI in an institutional (higher education) context.
2. Practical: A blueprint for UGM to develop a more proactive, data-driven, and efficient mental health support system.

1.6 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 presents a comprehensive review of relevant prior research in the fields of conversational AI, the application of gamification in educational and well-being contexts, related blockchain applications, and studies on user engagement and student welfare. Furthermore, this chapter 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. This chapter explains the adoption of Design Science Research and presents the system's high-level conceptual architecture, including its core functional components. It details the underlying technical architecture, justifying the chosen technology stack, and describes the database structure. The chapter also covers the user interface design for system administration and specifies the integrated security and privacy measures.

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 provides the theoretical foundations and academic context for the research. The first section, Theoretical Background, explains the core concepts and technologies that constitute the proposed framework. The second section, Literature Review, surveys existing work in related fields to identify the research gap that this thesis aims to address.

2.1 Theoretical Background

This section describes the foundational principles and technologies upon which the agentic AI framework is built, including Agentic AI, Large Language Models (LLMs), LLM orchestration, and workflow automation platforms.

2.1.1 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 [1]. 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 [1].

To operationalize this concept, this thesis formally introduces a framework built upon three distinct, specialized intelligent agents. Each agent is designed to address a specific challenge outlined in Chapter 1, and together they form the core of the proposed proactive support system. These agents are:

- The **Analytics Agent**, responsible for data-driven trend identification.
- The **Intervention Agent**, responsible for automating proactive outreach.
- The **Triage Agent**, responsible for real-time user support and resource routing.

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) [1]. A prominent and highly relevant architecture for designing such goal-oriented agents is the **Belief-Desire-Intention (BDI)** model [1]. 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 **Analytics 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 **Intervention Agent's** overarching goal to "automate proactive outreach."
- **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 **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 [1].

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 [1]. 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 **Analytics**, **Intervention**, and **Triage** agents must cooperate: the Analytics Agent provides the data-driven insights (beliefs) that the Intervention Agent uses to form its outreach plans (intentions), while the Triage Agent handles immediate, real-time needs that may fall outside the other agents' scopes.
- **Negotiation:** When agents have conflicting goals or must compete for limited resources, they must be able to negotiate to find a mutually acceptable compromise [1].

- **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 [1].

Therefore, this thesis leverages the MAS paradigm by designing a framework composed of three specialized, collaborative agents. Their individual, goal-directed behaviors, orchestrated within a hybrid 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.

2.1.2 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 Gemma and Gemini models used in this research, is the **Transformer architecture**, first introduced by Vaswani et al. [1]. 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) [1].

The core operation of a Transformer-based model involves processing input text through a series of encoding and/or decoding layers. In a generative, decoder-only model like Gemma, 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 [1].

Placeholder for Diagram: Simplified Transformer Architecture for Generative LLMs

This diagram should illustrate the flow of information:
1. Input Text -> Tokenizer
2. Tokens -> Embedding Layer + Positional Encoding
3. Embedded Tokens -> A stack of 'N' Decoder Blocks
4. Inside a Decoder Block: Multi-Head Self-Attention -> Feed-Forward Network
5. Final Output -> Linear Layer -> Softmax -> Probability of Next Token

Figure 2.1. A simplified view of the decoder-only Transformer architecture used in generative LLMs like Gemma. The model processes input embeddings through multiple layers of self-attention and feed-forward networks to predict the next token in a sequence.

This research utilizes a **hybrid LLM strategy** that leverages two distinct families of models based on this architecture to balance performance, privacy, and capability:

- **Locally-Hosted Open Models (Gemma):** The Gemma models are a family of lightweight, open-weight models developed by Google, built from the same research and technology used to create the Gemini models [1]. As decoder-only Transformers, they are optimized for generative text tasks. Being "open-weight" means their parameters are publicly available, allowing them to be deployed on institutional hardware. This approach is critical for this project as it guarantees data privacy and security—sensitive student conversations processed by the Triage Agent never leave the university's secure servers. Furthermore, local hosting provides low latency and eliminates per-token API costs, making it a sustainable choice for real-time, high-volume interactions.
- **Cloud-Based API Models (Gemini):** 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 [1]. In this framework, a powerful model like Gemini Pro is accessed via a secure API for complex, non-sensitive tasks, such as the weekly trend analysis performed by the Analytics Agent. It also serves as a robust fallback mechanism, ensuring service continuity and reliability should the local model encounter issues.

2.1.3 LLM Orchestration Frameworks (LangChain)

An LLM, in isolation, is a powerful text processor but lacks the ability to perform complex, multi-step tasks or interact with external systems. This limitation is addressed by LLM orchestration frameworks like **LangChain** [1]. LangChain is a software development framework that provides modular components for building applications powered by LLMs.

Key components of LangChain utilized in this research include:

- **Chains:** Sequences of calls, either to an LLM or another utility, that allow for complex, multi-step logic.
- **Agents and Tools:** LangChain allows the creation of agents that use an LLM as a reasoning engine to decide which "tools" to use. A tool can be any function, such as a database query, a calculation, or an API call.
- **Retrieval-Augmented Generation (RAG):** This technique allows an LLM to access and incorporate information from external knowledge bases before generating a response, which is crucial for providing context-aware answers.

In this thesis, LangChain serves as the core of the "Brain" within the FastAPI backend, enabling the development of the sophisticated logic required by the three agents.

2.1.4 Workflow Automation Platforms (n8n)

Modern software systems rarely exist in isolation. The ability to integrate disparate systems and automate data flows is critical for operational efficiency. **Workflow Automation Platforms** are designed for this purpose [1]. These platforms provide a visual interface to connect various applications, databases, and APIs into automated sequences, or "workflows."

This project utilizes **n8n**, an open-source workflow automation tool. Unlike a general-purpose programming language, n8n excels at tasks involving scheduling, event triggers, and system-to-system communication. In our architecture, n8n functions as the "Nervous System," responsible for:

- **Scheduled Triggers:** Running tasks at specific times (e.g., triggering the Analytics Agent every Sunday via a Cron job).

- **API Integration:** Calling the API endpoints exposed by our FastAPI backend and communicating with external services (e.g., an email server).
- **Visual Workflow Management:** Providing a clear, visual representation of the automated processes, which simplifies debugging and modification.

2.2 Literature Review

Bagian ini memaparkan landasan konseptual dan teoritis yang relevan dengan komponen-komponen utama platform yang diusulkan. Pemahaman mendalam terhadap teori ini esensial untuk perancangan sistem yang efektif dan evaluasi yang valid. Sumber utama bagian ini adalah buku referensi, artikel tinjauan (*review articles*), dan publikasi ilmiah fundamental di bidang terkait.

2.3 Analisis Perbandingan Metode

Di dalam tinjauan pustaka hasil akhirnya adalah analisis secara kualitatif atau pun secara kuantitatif kelebihan dan kekurangan metode jika dikaitkan dengan masalah, batasan-batasan masalah dan solusi yang dinginkan. Analisis kuantitatif tidak wajib teapi mempunyai nilai tambah di dalam tugas akhir saudara. Bagian ini menjelaskan kenapa metode tersebut dipilih dan uraikan dengan lebih jelas metode pelaksanaan tugas akhir yang ingin Anda lakukan.

2.4 Pertanyaan Tugas Akhir (Jika Perlu)

Pertanyaan tugas akhir bersifat opsional dan dapat ditambahkan untuk menekankan hal-hal yang hendak diketahui dari tugas akhir berdasar pada tujuan tugas akhir. Pertanyaan tugas akhir dikenal dengan RQ (*Research Question*) dan harus memiliki keterkaitan dengan RO (*Research Objective*). Satu RO dapat memiliki satu atau lebih dari satu RQ.

CHAPTER III

SYSTEM DESIGN AND ARCHITECTURE

3.1 Research Methodology: Design Science Research (DSR)

3.2 System Overview and Conceptual Design

Purpose: To provide a high-level, 10,000-foot view of the system's purpose and its main components, making it understandable before diving into technical specifics.

Elaboration Points:

Present a concise paragraph describing the conceptual model: "The proposed framework is an ecosystem of three collaborative, intelligent agents that work in concert to transform an institution's mental health support from a reactive to a proactive model..."

Include a high-level Context Diagram showing the main entities (Students, University Staff/Counselors) and systems (UGM-AICare User App, Agentic AI Backend, Admin Dashboard) and how they interact.

3.3 Functional Architecture: The Agentic Core

Purpose: To detail the "what" of the system. This section explains the specific roles and functions of each agent. This is the heart of your functional design.

Elaboration Points: Create a subsection for each of the three agents. For each, define its:

3.3.1 The Analytics Agent

Goal: To autonomously identify mental health trends from anonymized user data.

Perception (Inputs): Anonymized conversation logs from the PostgreSQL database.

Processing Logic: Describe the NLP tasks it performs: topic modeling, sentiment analysis, and summarization.

Action (Outputs): A structured weekly report (in JSON format) containing key insights (e.g., top 5 trending stress topics, overall sentiment score).

3.3.2 The Intervention Agent

Goal: To automate targeted, proactive outreach campaigns.

Perception (Inputs): The structured report from the Analytics Agent and a set of predefined "campaign rules."

Processing Logic: A rule-based engine that maps insights to actions (e.g., IF 'exam stress' > threshold THEN trigger 'time-management-workshop' campaign).

Action (Outputs): A signal sent to the orchestration layer (n8n) with target audience segment and message content.

3.3.3 The Triage Agent

Goal: To efficiently route a student to the most appropriate level of support in real-time.

Perception (Inputs): A user's live conversation with the chatbot.

Processing Logic: A real-time text classification model to determine the conversation's severity level (e.g., Level 1: Casual, Level 2: Moderate, Level 3: Red Flag).

Action (Outputs): A structured recommendation (e.g., suggest a self-help module, suggest booking a counselor, provide an emergency hotline).

Include a detailed Data Flow Diagram (DFD) showing how data moves between these three agents and the database.

3.4 Technical Architecture: The Hybrid System

Purpose: To detail the "how" of the system—the engineering blueprint, including justification for key technology choices.

Elaboration Points:

3.4.1 Overall System Architecture Diagram

A detailed diagram showing the interplay between all technologies: Next.js (Admin Dashboard), FastAPI (Backend), PostgreSQL, LangChain (as a library), and n8n (as a separate, orchestrated service). Show the communication protocols (e.g., REST API, direct DB connection).

3.4.2 The "Brain": FastAPI + LangChain Service

Justify the choice of FastAPI (for performance, async support) and LangChain (for LLM orchestration). Detail the API design, defining key endpoints (e.g., /api/agents/generate-report) and their request/response schemas.

3.4.3 The "Nervous System": n8n Workflows

Justify the choice of n8n for robust workflow automation. Provide screenshots or diagrams of the primary n8n workflows (e.g., the "Weekly Report Generation" workflow triggered by a Cron job).

3.5 Database Design

Purpose: To define the data persistence layer of the system.

Elaboration Points:

Present a clean Entity-Relationship Diagram (ERD).

Table 3.1. Key Columns and Data Types for `conversation_logs` Table

Column Name	Data Type	Description
id	SERIAL PRIMARY KEY	Unique identifier
user_id	UUID	Reference to user (anonymized)
timestamp	TIMESTAMP WITH TIME ZONE	Time of message
message	TEXT	User or agent message content
sender	VARCHAR(16)	'user' or 'agent'
sentiment_score	FLOAT	Sentiment analysis result
topic	VARCHAR(64)	NLP-inferred topic label

3.6 User Interface (UI) Design

Purpose: To show the design of the human interface for the system's administrative users.

Elaboration Points:

Define the primary user persona for the dashboard (e.g., "Dr. Astuti, Head of Counseling Services").

Present wireframes or high-fidelity mockups for the key screens of the Admin Dashboard (e.g., the main analytics view, the report history page).

3.7 Security and Privacy by Design

Purpose: To demonstrate that critical security and privacy considerations are integral to the architecture.

Elaboration Points:

Detail the Data Anonymization Pipeline: How is Personally Identifiable Information (PII) identified and redacted from chat logs before they are stored for analysis?

Describe the Role-Based Access Control (RBAC) mechanism for the admin dashboard.

Mention standard security practices like data encryption in transit (TLS) and at rest.

3.8 Alur Tugas Akhir

Menguraikan prosedur yang akan digunakan dan jadwal atau alur penyelesaian setiap tahap. Alur penelian ini dapat disajikan dalam bentuk diagram. Diagram dapat disusun dengan aturan yang baik semisal menggunakan *flowchart*. Aturan dan tutorial pembuatan *flowchart* dapat dilihat di http://ugm.id/flowcharttutorial. Setelah menggambarkannya, penulis wajib menjelaskan langkah-langkah setiap alur tugas akhir dalam sub bab tersendiri sesuai dengan kebutuhan.

3.9 Etika, Masalah, dan Keterbatasan Penelitian (Opsional))

Bagian ini membahas pertimbangan etis penelitian dan [potensi] masalah serta keterbatasannya. Jika menyangkut penelitian dengan makhluk hidup, maka dibutuhkan adanya *ethical clearance*, di bagian ini hal itu akan dibahas. Demikian juga tentang keterbatasan ataupun masalah yang akan timbul.

CHAPTER IV
HASIL DAN PEMBAHASAN

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

Poin pertama adalah membahas tujuan penelitian pertama. Dapat ditambahkan beberapa sub bab jika diperlukan.

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

Poin kedua adalah membahas tujuan penelitian kedua. Dapat ditambahkan beberapa sub bab jika diperlukan. Dapat juga diteruskan ke Sub Bab Pembahasan hasil 3 dan seterusnya, jika ada tiga atau lebih tujuan penelitian.

4.3 Perbandingan Hasil Penelitian dengan Hasil Terdahulu

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

CHAPTER V

TAMBAHAN (OPSIONAL)

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

CHAPTER VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

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

6.2 Saran

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

REFERENCES

[1] Anonymous, "Citation needed," 2025, placeholder reference. Please replace with actual citation.

[2] 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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

LAMPIRAN

L.1 Isi Lampiran

Lampiran bersifat opsional bergantung hasil kesepakatan dengan pembimbing dapat berupa:

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

LAMPIRAN

L.2 Panduan Latex

L.2.1 Syntax Dasar

L.2.1.1 Penggunaan Sitasi

Contoh penggunaan sitasi [2, 3] [4] [5] [6] [7, 8]

L.2.1.2 Penulisan Gambar

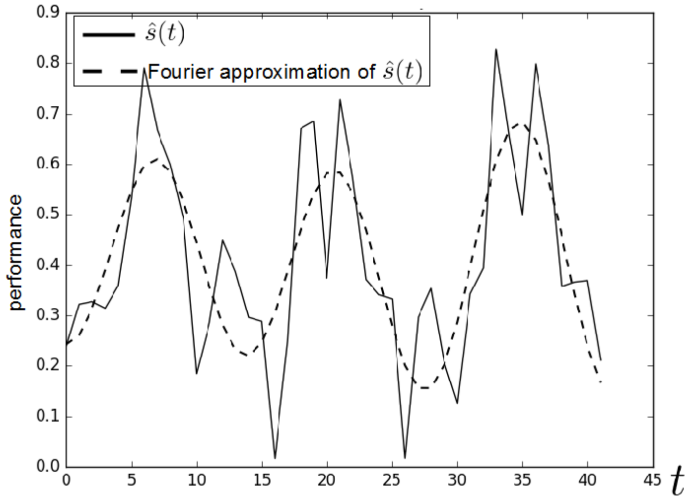

Figure 1. Contoh gambar.

Contoh gambar terlihat pada Gambar 1. Gambar diambil dari [8].

L.2.1.3 Penulisan Tabel

Table 1. Tabel ini

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

Contoh penulisan tabel bisa dilihat pada Tabel 1.

L.2.1.4 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.2.1.5 Contoh list

Berikut contoh penggunaan list

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

L.2.2 Blok Beda Halaman

L.2.2.1 Membuat algoritma terpisah

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

Algorithm 1 Contoh algorima

```
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.2.2.2 Membuat tabel terpisah

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

Table 2. Contoh tabel panjang

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

L.2.2.3 Menulis formula terpisah halaman

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

Contoh sederhana dapat digambarkan sebagai berikut.

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

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

LAMPIRAN

L.3 Format Penulisan Referensi

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

Berikut ini adalah sampel penulisan dalam format IEEE:

L.3.1 Book

Basic Format:

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

Examples:

[1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

[2] L. Stein, "Random patterns," in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

[3] R. L. Myer, "Parametric oscillators and nonlinear materials," in Nonlinear Optics, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.

[4] M. Abramowitz and I. A. Stegun, Eds., Handbook of Mathematical Functions (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.

[5] E. F. Moore, "Gedanken-experiments on sequential machines," in Automata Studies (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.

[6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), Integrated Electronic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1970.

[7] M. Gorkii, "Optimal design," Dokl. Akad. Nauk SSSR, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., The Mathematical Theory of Optimal Processes. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).

[8] G. O. Young, "Synthetic structure of industrial plastics," in Plastics, vol. 3,

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

L.3.2 Handbook

Basic Format:

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

Examples:

[1] Transmission Systems for Communications, 3rd ed., Western Electric Co., Winston Salem, NC, 1985, pp. 44-60.

[2] Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[3] RCA Receiving Tube Manual, Radio Corp. of America, Electronic Components and Devices, Harrison, NJ, Tech. Ser. RC-23, 1992.

Conference/Prosiding

Basic Format:

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

Examples:

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

Sumber Online/Internet

Basic Format:

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

Examples:

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

Skripsi, Tesis dan Disertasi

Basic Format:

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

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

Examples:

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

LAMPIRAN

L.4 Contoh Source Code

L.4.1 Sample algorithm

Algorithm 2 Kruskal's Algorithm

```
procedure MAKESET(v)
    Create new set containing v
end procedure
function FINDSET(v)
    return a set containing v
end function
procedure UNION(u,v)
    Unites the set that contain u and v into a new set
end procedure
function KRUSKAL(V, E, w)
    A ← {}
    for each vertex v in V do
        MakeSet(v)
    end for
    Arrange E in increasing costs, ordered by w
    for each (u,v) taken from the sorted list do
        if FindSet(u) ≠ FindSet(v) then
            A ← A ∪ {(u, v)}
            Union(u, v)
        end if
    end for
    return A
end function
```

L.4.2 Sample Python code

```
import numpy as np

def incmatrix(genl1,genl2):
  m = len(genl1)
  n = len(genl2)
  M = None #to become the incidence matrix
  VT = np.zeros((n*m,1), int)  #dummy variable

  #compute the bitwise xor matrix
  M1 = bitxormatrix(genl1)
  M2 = np.triu(bitxormatrix(genl2),1)

  for i in range(m-1):
    for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
      for k in range(len(r)):
        VT[(i)*n + r[k]] = 1;
        VT[(i)*n + c[k]] = 1;
        VT[(j)*n + r[k]] = 1;
        VT[(j)*n + c[k]] = 1;

  if M is None:
    M = np.copy(VT)
  else:
    M = np.concatenate((M, VT), 1)

  VT = np.zeros((n*m,1), int)

  return M
```

L.4.3 Sample Matlab code

```
function X = BitXorMatrix(A,B)
%function to compute the sum without charge of two vectors

  %convert elements into usigned integers
  A = uint8(A);
  B = uint8(B);

  m1 = length(A);
  m2 = length(B);
  X = uint8(zeros(m1, m2));
  for n1=1:m1
     for n2=1:m2
        X(n1, n2) = bitxor(A(n1), B(n2));
     end
  end
```