INTEGRATION OF INDEPENDENT STUDENT COMPETITION DASHBOARD WITH ARTIFICIAL INTELLIGENCE USING THE WATERFALL METHOD AT FACULTY OF INDUSTRIAL ENGINEERING TELKOM UNIVERSITY

By:

MUHAMMAD HANIF AL HADI

1202213165

INFORMATION SYSTEMS UNDERGRADUATE STUDY PROGRAM

FACULTY OF INDUSTRIAL ENGINEERING

TELKOM UNIVERSITY

2025

ABSTRACT

The development of information technology in the era of industrial revolution 4.0 brings significant transformation in various sectors, including higher education. Artificial Intelligence (AI) has emerged as a key technology enabling faster, more accurate, and efficient data analysis to support strategic decision-making. According to Verified Market Research (2024), the AI-based educational technology market is projected to reach USD 84.73 billion by 2031 with an annual growth rate of 45.21%. This research aims to develop a student independent competition dashboard integrated with AI technology at the Faculty of Industrial Engineering (FRI) Telkom University using the Waterfall methodology to improve monitoring capabilities and strategic analysis of student achievements.

The problem addressed in this research is the limitations of the current dashboard system in the FRI environment, where the existing dashboard can only present data in basic visualization formats without deep analytical capabilities or natural language-based interaction. The current system lacks deep analytical capabilities and interactive features, presenting data only in basic visualization formats without natural language-based interaction, making it challenging for faculty management to obtain strategic recommendations for student achievement development. Additionally, the absence of AI technology integration prevents the system from providing automated insights or trend analysis needed for developing effective student affairs strategies.

The research methodology employed is the Waterfall Model with a 4-layer architecture approach consisting of Presentation Layer (Laravel framework with authentication and role-based access control), Intelligence Layer (Flowise AI integrated with OpenAI GPT-4o for natural language processing), Visualization Layer (Google Looker Studio for interactive data visualization), and Data Layer (Aiven MySQL Cloud Database with star schema implementation). The research data comprises historical independent competition data of FRI students for the period 2021-2024, totaling 649 records that underwent cleaning and transformation

processes using Python in Google Colaboratory, reducing from 26 columns to 21 relevant columns while preserving data integrity through NULL value retention.

The research results demonstrate that the developed dashboard system successfully integrates interactive data visualization with AI chatbot capabilities that understand Indonesian language queries and provide in-depth analysis based on competition data. The system features role-based access control differentiating access rights between admins and students, complete CRUD operations for competition data management, and real-time visualization through various chart types including pie charts, bar charts, funnel charts, and interactive tables. The AI chatbot achieves an capable of answering complex questions about achievement trends, performance analysis, and strategic recommendations. System evaluation through automated testing using Laravel Dusk shows all 25 test cases function properly without errors, User Acceptance Testing (UAT) with key stakeholders achieves 100% success rate, and System Usability Scale (SUS) evaluation reaches a score of 90 points, categorized as A+ (Best Imaginable), indicating excellent usability.

The conclusion of this research is that the AI-integrated independent competition dashboard system has been successfully developed and meets the requirements for monitoring student achievements at FRI. The system provides enhanced monitoring capabilities through AI-integrated dashboard featuring real-time visualization, natural language query interface in Indonesian, and automated data-driven insights for strategic decision-making. Technology integration using iframe embedding enables seamless access to Looker Studio and Flowise AI within a unified Laravel-based platform. The benefits achieved include enhanced monitoring capabilities, democratized access to information for stakeholders through natural language interface, support for real-time data-driven strategic decision-making, and accessibility to business intelligence capabilities that previously required specialized technical expertise.

Keywords—Dashboard, Artificial Intelligence, Waterfall Model, Student Competition, Flowise AI, Natural Language Interface

APPROVAL PAGE

INTEGRATION OF INDEPENDENT STUDENT COMPETITION DASHBOARD WITH ARTIFICIAL INTELLIGENCE USING THE WATERFALL METHOD AT FACULTY OF INDUSTRIAL ENGINEERING TELKOM UNIVERSITY

Has been approved and ratified at the Final Project Session

Bachelor's degree Information Systems Study Program

Faculty of Industrial Engineering, Telkom University

By :

MUHAMMAD HANIF AL HADI

1202213165

Bandung, 29 July 2025

Approved by,

Advisor 1,	Advisor 2,
Prof. Dr. Irfan Darmawan, S.T., M.T.	Dr. Oktariani Nurul Pratiwi, S.Kom.,M.T.
14750038	20870013

ORIGINALITY STATEMENT SHEET

Name	:	Muhammad Hanif Al Hadi
Student ID	:	1202213165
Address	:	Duren Sawit
Phone Number	:	082139344344

Declare that this Final Project is my own original work. For this statement, I am ready to bear the risks or sanctions imposed on me if later it is found that there is a violation of academic honesty or scientific ethics in this work, or evidence is found that shows the authenticity of this work.

Bandung, 29 July 2025

Muhammad Hanif Al Hadi

PREFACE

In the name of Allah, the Most Gracious, the Most Merciful. All praise and gratitude be to God Almighty for His blessings and grace that have enabled me to complete this Final Project successfully. This Final Project marks a significant milestone in my academic journey in the Bachelor's Program of Information Systems, Faculty of Industrial Engineering, Telkom University.

1. In completing this Final Project, I would like to extend my deepest gratitude and appreciation to the individuals and institutions who have provided invaluable support, guidance, and contributions throughout the research process and the preparation of this report:
2. I would like to express my sincere gratitude to Mr. Luthfi Ramadani, S.T., M.T., Ph.D., my academic advisor, who has been an invaluable mentor throughout my undergraduate journey. His consistent support, guidance, and dedication in helping me navigate academic challenges since the beginning of my studies have been instrumental in my growth as a student and researcher. His mentorship has truly made a significant impact on my academic journey at Telkom University.
3. To my academic supervisors, Prof. Dr. Irfan Darmawan, S.T., M.T and Dr. Oktariani Nurul Pratiwi, S.Kom.,M.T., for their unwavering guidance, support, and valuable insights throughout the preparation of this Final Project. I am deeply grateful for the time, knowledge, and constructive feedback they have shared, which have been pivotal in completing this research.
4. I would like to express my gratitude to my colleague, Muhammad Haikal Zulhaj, for his dedication, cooperation, and sincere support throughout the preparation of this final project. His contributions and discussions, from ideas, thoughts, and differing arguments and perspectives to his active involvement in every stage of the project, were instrumental in the success of this research. Working with him has been a valuable experience, and his collaborative spirit is commendable. May this collaboration serve as a solid foundation for future mutual success, both academically and professionally

5. To my family, especially my father, my late mother, and my two sisters, for their endless moral support, prayers, and unconditional love throughout my academic journey and the completion of this Final Project. I sincerely appreciate their sacrifices, encouragement, assistance, and unwavering belief in my abilities that have made this achievement possible.
6. Finally, to the representatives of the company that agreed to serve as a case study in this research. I am truly grateful for their willingness to provide essential data through interviews, observations, and evaluations, which have greatly contributed to the completion of this project.

Lastly, my gratitude goes to all parties who have indirectly contributed to the completion of this Final Project. May all the kindness and assistance given be rewarded manifold by God Almighty. I hope that this Final Project can provide valuable insights for readers and contribute positively to the advancement of knowledge in the field of Information Systems.

Table of Contents

List of images

List of tables

List of Appendix

Glossary of Terms

Term		Description	First time page used
Aiven	:	Cloud database-as-a-service platform that provides managed MySQL services for competitive data storage	5
Chatbot	:	An AI program that can interact with users through natural language conversations to provide data analysis.	11
SUS	:	System Usability Scale - a standard instrument with 10 questions to measure the ease of use of a system on a scale of 0-100	14
DITMAWA	:	The Student Affairs Directorate of Telkom University, which serves as the source of student competition data.	38
AI	:	Design Science Research Methodology - kerangka metodologis untuk pengembangan dan evaluasi artifact sistem informasi	23
ETL	:	Extract, Transform, Load - a data processing pipeline from raw Excel to analysis-ready data	39
Flowise AI	:	Low-code platform for building AI chatbots that can connect directly to databases and the OpenAI API	12
FRI	:	Faculty of Industrial Engineering, Telkom University - locus of research and implementation of dashboard systems	1
GPT-4o	:	OpenAI Large Language Model used for natural language processing in chatbots	11
Looker Studio	:	Google business intelligence platform for creating interactive dashboards and real-time data visualization	5
NLIDB	:	Natural Language Interface to Database - a system that allows querying a database using the Indonesian language	8

CHAPTER I
INTRODUCTION

I.1 Background

Information technology development in the 4.0 industrial revolution era has significantly transformed various sectors, including higher education. Artificial intelligence (AI) has become a primary focus, increasingly adopted in education ecosystems for personalized learning, administrative process automation, and institutional performance monitoring.

AI adoption growth in education is supported by market projections showing positive global trends. Market data indicates significant growth in recent years with projections for continued future increases, as shown in Figure I-1.

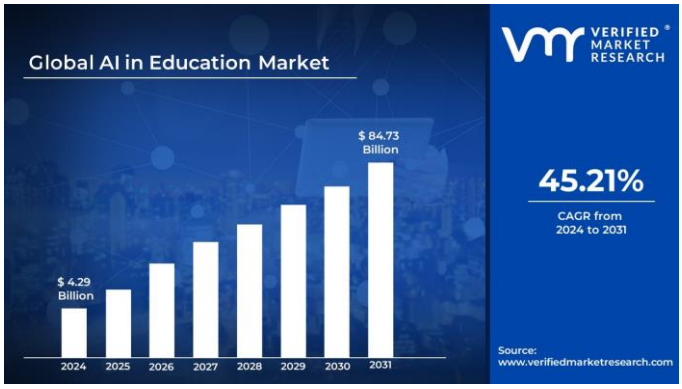

Figure I-1. Global AI in education market
Source (Verified Market Research, 2024)

Based on Figure I-1, the AI market in the education sector is continuing to grow rapidly. According to a report from Verified Market Research (2024), the market value of AI-based educational technology is projected to reach USD 84.73 billion by 2031, with a Compounded Annual Growth Rate (CAGR) of 45.21% from 2024 to 2031. This growth reflects the increasing adoption of AI technology in educational institutions for data analysis and decision support systems. The integration of AI technology enables educational institutions to analyze complex datasets and provide strategic insights for institutional management.

The development of effective dashboards requires careful consideration of information presentation and user interaction. Few (2013) defines a dashboard as a

visual display of important information designed to help users achieve goals, presented on a single screen for quick monitoring. Yigitbasioglu and Velcu (2012) emphasize that dashboards serve as data-driven decision support systems that provide information in formats suitable for decision-makers. This concept aligns with the need for educational institutions to monitor student performance efficiently.

Recent advances in Natural Language Processing have opened new possibilities for database interaction. Gao et al. (2024) demonstrate that Large Language Models can achieve significant accuracy in converting natural language queries to SQL, making complex data analysis accessible to non-technical users. Such capabilities are particularly valuable in educational contexts where stakeholders may lack technical expertise but require access to analytical insights.

At Telkom University's Faculty of Industrial Engineering (FRI), student activities are crucial for developing academic and non-academic potential. However, managing student achievements in independent competitions remains suboptimal due to limitations in the current dashboard system, which hinders effective management and strategic decision-making, as illustrated in Figure I-2.

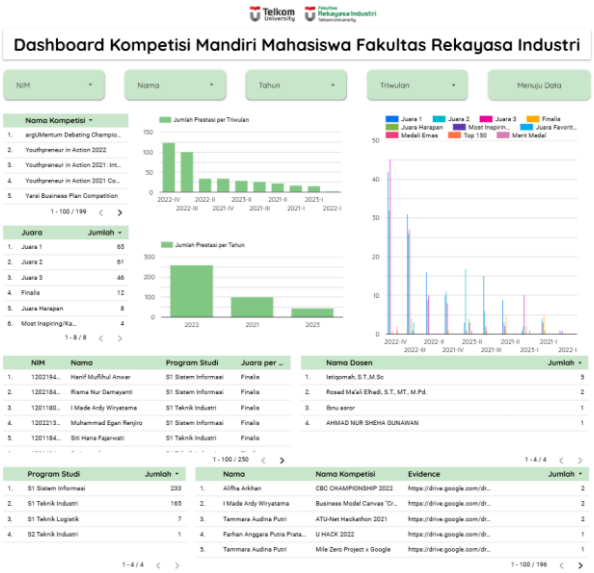

Figure I-2. Current FRI student independent competition dashboard

Based on Figure I-2, the current system uses a simple dashboard with data displayed in the form of regular tables and some basic graphs that only meet very basic data

presentation needs. The dashboard interface only provides basic functionality for data search, without any in-depth analysis features or adequate visualization. The current dashboard system can only present data in basic visualization formats without deep analytical capabilities or natural language-based interaction.

The system lacks integrated information and strategic insights, hindering data-driven decision-making. Without AI technology integration, it cannot provide interactive recommendations, comprehensive trend analysis for strategic planning, deeper insights, or strategic recommendations, thus impeding effective decision-making and targeted student guidance.

Existing dashboard limitations reveal a research gap in AI integration for student performance monitoring in higher education. While traditional dashboards excel at data presentation, they lack analytical intelligence to transform raw data into actionable strategic insights—particularly evident in student competition monitoring, where stakeholders need both data visualization and intelligent interpretation/recommendation capabilities for evidence-based decision-making.

This research develops an independent competition dashboard for FRI Telkom University to provide accurate, integrated, accessible student performance data visualization. The AI-integrated advanced dashboard enables faculty management to monitor achievements comprehensively in real-time, facilitating more accurate decision-making for student performance support. The research employs Waterfall methodology—a sequential, systematic software development model progressing through background/needs analysis, system architecture/interface design, gradual implementation from data to presentation layers, validation testing, and maintenance/documentation (Pressman & Maxim, 2019).

I.2 Problem Statement

Based on the background that has been outlined, the formulation of the research problem is as follows:

1. How to develop a dashboard integrated with AI technology to monitor and analyze student independent competition data within the Faculty of Industrial Engineering (FRI) at Telkom University?

2. How to utilize AI technology (through the Flowise AI platform and OpenAI API) to provide conversational interface capabilities that enable natural language queries and generate analytical insights supporting strategic decision-making for student development?
3. How to integrate multiple technologies (dashboard visualization, AI chatbot, and database management) into a unified platform that enhances accessibility and usability for non-technical stakeholders?

I.3 Research Objectives

Based on the problem statement, the objectives of this study are as follows:

1. To develop an AI-integrated dashboard system for monitoring and analyzing student independent competition data within the Faculty of Industrial Engineering (FRI) at Telkom University.
2. To implement AI technology (through the Flowise AI platform and OpenAI API) that provides conversational interface capabilities, enabling users to perform natural language queries and receive analytical insights for strategic decision-making.
3. To create a unified platform that integrates dashboard visualization, AI chatbot functionality, and database management to enhance data accessibility and usability for stakeholders with varying technical backgrounds.

I.4 Research Scopes

To ensure that this research is focused and aligns with the available time and resources, several research limitations are established as follows:

1. The development of the dashboard is focused on the independent competition participated in by FRI students at Telkom University, not including Belmawa competitions or competitions outside the focus of independent competitions.
2. The AI technology used is limited to the ability to provide insights (such as analysis, identification, and actionable insights) as well as answering

interactive questions based on data, using the Flowise AI platform integrated with the OpenAI API as a Large Language Model (LLM).

3. The data used is historical data and the latest competition data within the scope of FRI, sourced from the data warehouse that has been integrated into Aiven, with a maximum period of the last 5 years. The dashboard will be visualized using Looker Studio.

I.5 Research Benefit

Based on the problem statement and research objectives, this research is expected to provide the following benefits:

1. For the Faculty of Industrial Engineering (FRI) Telkom University: The dashboard facilitates comprehensive management monitoring and analysis of student performance in independent competitions. AI integration enables structured evaluations, pattern/trend identification, and effective planning through descriptive insights and strategic recommendations for student achievement monitoring.
2. For Student Affairs: The AI-integrated dashboard enables easier monitoring of student achievements across competitions. AI-generated information, including in-depth analysis and strategic recommendations, helps better understand achievement development, identify improvement opportunities, and formulate targeted support and coaching strategies.
3. For Other Researchers: This research serves as reference and foundation for developing similar information systems in universities or institutions, particularly for creating data-based performance monitoring systems that integrate AI technology for in-depth analysis and real-time or near-real-time decision-making.

I.6 Systematization of Writing

This research is described with the following systematic writing:

Chapter I. Introduction

This chapter contains problem context, research background, problem formulation, research objectives, research limitations, research benefits, and writing systematics.

Chapter II. Literature Review

This chapter contains relevant literature and previous study results. Includes multiple methodologies / methods / frameworks to solve problems or minimize gaps between existing conditions and targets. Concludes with methodology / method / framework selection analysis to determine the approach used in this research.

Chapter III. Methods

This chapter explains the Waterfall methodology for system development, including requirement analysis, system design, implementation, and validation phases. Covers data collection methods from stakeholders and DITMAWA, Python data processing pipeline, and systematic approach for developing layered architecture system integrating dashboard visualization with AI technology.

Chapter IV. Problem Resolution

This chapter presents system requirements analysis and integrated dashboard system design. Covers stakeholder analysis, functional/non-functional requirements, layered system architecture design, user interface design, and integration design between Laravel, Flowise AI, and Google Looker Studio components.

Chapter V. Result, Validation, and Implication

This chapter presents implementation process and testing of the integrated dashboard system. Covers systematic multi-layered architecture implementation and comprehensive testing through Automated Functionality Testing, User Acceptance Testing (UAT), and System Usability Scale (SUS) evaluation.

Chapter VI. Conclusion and Suggestion

This chapter is a closing chapter that contains conclusions from the overall results of the research and discussion that has been carried out. The conclusion will summarize the answer to the problem formulation that has been set in Chapter I. In addition, this chapter will also present suggestions that can be used as a reference for future system development or as ideas for further research.

CHAPTER II
LITERATURE REVIEW

This chapter presents the theoretical foundation that serves as the basis for the entire research. The description will begin with an elaboration of key concepts such as dashboards as data visualization media, Google Looker Studio as a business intelligence platform, and technological infrastructure like Aiven and MySQL. Next, the Natural Language Interface to Database (NLIDB) will be discussed as an interaction bridge, as well as AI technologies like OpenAI API and Flowise AI. The Laravel framework as backend development and the Waterfall methodology as the research framework will also be explained. As a conclusion, a review of relevant previous research is presented to identify research gaps and position the novelty of this study.

II.1 Dashboard

A dashboard is a visual interface that presents important information in a format that is easy to understand and can be accessed quickly. According to Few (2013) in his book "Information dashboard design: Displaying data for at-a-glance monitoring" defines a dashboard as a visual display of the most important information designed to help users achieve one or more goals, presented on a single screen so that it can be monitored quickly with just a glance.

According to Yigitbasioglu and Velcu (2012) in their research "A review of dashboards in performance management: Implications for design and research" reinforce this concept by defining a dashboard as "a data-driven decision support system, which provides information in a particular format to the decision maker." Dashboards function as tools that help decision-makers understand the business situation comprehensively through intuitive visualization.

According to Pauwels et al. (2009) in the same study identified several main objectives of the dashboard, namely: "(i) consistency, (ii) monitoring, (iii) planning, and (iv) communication". Consistency refers to the measurement and measurement procedures used across departments and business units. Monitoring is the daily

evaluation of metrics that should result in corrective actions. Dashboards can also be used for planning by enabling simulations of various future business scenarios.

In the context of data warehousing and business intelligence, Kimball and Ross (2013) in "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" emphasize the importance of dashboards as a presentation layer that allows end-users to easily access and understand data. They identify that an effective dashboard must be able to present dimensional data in an intuitive and actionable way for decision-makers.

II.2 Google Looker Studio

Google Looker Studio is a business intelligence platform that provides data visualization capabilities and the creation of interactive dashboards. According to Google Cloud (2025), "Looker Studio unlocks the power of your data by making it easy to create interactive dashboards and compelling reports from a wide variety of sources, driving smarter business decisions." This platform provides self-service business intelligence capabilities with the flexibility to support smarter business decision-making.

According to Coupler.io (2025) in their comprehensive tutorial reveals that "with Looker Studio dashboards, even those without a data science or business intelligence background can effortlessly leverage the data analysis". These findings reinforce Looker Studio's position as a democratic platform for data analysis, allowing various professional backgrounds to leverage data visualization capabilities without requiring deep programming expertise.

Based on the analysis of the dashboard research that has been conducted, as shown by Noviana et al. (2023) and Freska Rolansa et al. (2021), the need for a visualization platform that can integrate multiple data sources has become very important. Google Looker Studio provides a solution for this need with embedding capabilities that allow dashboards to be integrated into larger web applications, as explained by Google Cloud (2025) that "embedding allows you to include your Looker Studio report in any web page or intranet, making it easier for you to tell your data story to your team or the world."

II.3 Aiven

Aiven is a cloud database-as-a-service platform that provides managed database infrastructure with an open-source and multi-cloud approach. According to Aiven documentation (2025), the platform "simplifies data storage and data movement across clouds, providing complete pipeline transparency while removing barriers to experiment with the AI you want." This approach offers flexibility that traditional cloud database solutions lack.

According to Platform Engineering Organization (2024) identifies that Aiven provides "open-source cloud database-as-a-service platform that supports Redis, Apache Cassandra, Apache Kafka, PostgreSQL and many more." The diversity of supported database services makes Aiven a comprehensive solution for organizations that require various types of data storage within a single integrated platform.

For MySQL as this research's database choice, Aiven documentation (2025) states "Educational institutions utilize wide variety of database applications from e-learning and assessments to tuition payment processing. MySQL is a proven choice, widely used to automate academic and administrative processes and is among the most popular databases for curricula and research institutions." Aiven provides fully managed MySQL service with automated backup, high availability (99.99% uptime SLA), and compliance standards including ISO 27001, SOC 2, PCI-DSS, and GDPR.

The selection of Aiven as the cloud database provider in this study is based on its ability to provide a reliable and scalable managed service, in line with the needs of the dashboard application with NLIDB integration, which requires optimal database performance to support query processing and real-time data visualization.

II.4 MySQL

MySQL is a relational database management system chosen as the backend database in this research. The selection of MySQL is based on its popularity, reliability, and ability to handle complex queries required for a dashboard application with a natural language interface.

In the context of Text-to-SQL generation, as demonstrated in the research by Gao et al. (2024) "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation," the database's ability to handle queries generated by LLMs becomes an important factor. MySQL, as a mature SQL implementation, supports various types of complex queries needed for dashboard data analysis.

Research by Nugraha et al. (2024) also shows the use of relational databases in the implementation of chatbot for Text-to-SQL generation, where a well-defined database structure is crucial for the accuracy of query generation by LLM.

II.5 Natural Language Interface to Database (NLIDB)

One of the fundamental challenges in data utilization is accessibility. Traditionally, to retrieve information from a database, a user requires expertise in structured query languages such as SQL (Structured Query Language). This dependence on SQL becomes a barrier for non-technical users who do not understand the database structure or how to write complex queries.

Natural Language Interface to Database (NLIDB) is a system that allows users to interact with a database using natural language, overcoming the traditional limitations that require mastery of SQL. This concept is fundamental in this research to provide a user-friendly interface for non-technical users.

Research by Gao et al. (2024) in "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation" shows significant advancements in NLIDB technology using Large Language Models. This research reveals that the use of LLMs such as GPT-4 can achieve execution accuracy of up to 83.5% on the Spider benchmark, marking a significant breakthrough in the development of modern NLIDB.

The practical implementation of NLIDB has been demonstrated in research by Nugraha et al. (2024) on "Development of Chatbot Applications with Large Language Model for Text-to-SQL Generation." This research uses GPT-4o through an API and implements semantic caching with Redis for capabilities, demonstrating that NLIDB can be effectively applied in real systems with optimal performance.

In the context of education, the implementation of NLIDB can provide easier access for users to explore dashboard data without requiring SQL expertise, which is the objective of this research.

II.6 OpenAI API

OpenAI API is an interface that enables the integration of artificial intelligence capabilities, specifically generative language models, into applications. In the context of this research, OpenAI API is used as a backend for natural language processing in the NLIDB system.

Research by Gao et al. (2024) demonstrates the superiority of OpenAI models, particularly GPT-4, in the task of Text-to-SQL generation. This study reveals that OpenAI models achieve the highest execution accuracy compared to other models, with "DAIL-SQL outperforms DIN-SQL and STRIKE in terms of both accuracy and capabilities" when using GPT-4 as the backend LLM.

Nugraha et al. (2024) in practical implementation used the "GPT-4o Model from OpenAI via API calls" to develop a chatbot capable of translating natural language questions into SQL queries. This research demonstrates that the OpenAI API can be effectively integrated with caching and rate limiting systems to create robust and efficient applications.

The ability of the OpenAI API to understand context and generate accurate SQL queries makes it the right choice for implementing NLIDB in this research, especially for integrating a natural language interface into the dashboard system.

II.7 Flowise AI

Flowise AI is a low-code platform for building AI applications, including chatbots and virtual assistants, that allows the integration of AI models like ChatGPT into applications without the need for complex coding. This platform provides a visual interface for designing workflows, managing data, and configuring AI models used in the development of artificial intelligence-based applications (FlowiseAI, 2023).

In the study by Reis et al. (2024), Flowise was used to simplify the process of discovering and analyzing biomedical data, allowing researchers to access and

analyze various types of data more efficiently. This research shows that Flowise not only speeds up the data search process but also enhances collaboration between teams by providing tools that enable better interaction with large datasets. Nugraha et al. (2024) in practical implementation used a similar approach by integrating language models through API calls in the development of the NLIDB system.

In the context of modern Natural Language Interface to Database (NLIDB) development, platforms like Flowise AI can function as middleware that integrates multiple AI models and APIs. This aligns with the architecture shown in the research by Nugraha et al. (2024), which uses a framework for frontend and backend with LLM integration through API calls. Flowise AI provides a solution for the complexity of integrating AI workflows required in NLIDB systems, enabling rapid prototyping and deployment of AI applications without the need for extensive coding, which meets the needs for developing dashboards with a natural language interface.

II.8 Laravel

Laravel is a PHP framework chosen as the backend framework in this research. The selection of Laravel is based on its capability for rapid development and a comprehensive ecosystem for modern web application development.

Research by Soegoto (2018) on "Implementing Laravel framework website as brand image in higher-education institution" shows that the Laravel framework can effectively support the development of higher education institution websites. This research reveals that Laravel can optimize the development process and produce maintainable applications.

In the context of application development with AI integration, as demonstrated by Nugraha et al. (2024) who use "FastAPI for the backend," the need for a framework that can handle API integration and real-time processing becomes important. Laravel provides the necessary features such as routing, API development, database ORM, and queue management that support the development of dashboard applications with NLIDB integration.

Laravel also provides the capability to embed external services and APIs, which is important for integration with Google Looker Studio, OpenAI API, and Flowise AI in the developed system architecture.

II.9 Unified Modeling Language

Unified Modeling Language (UML) is a standardized modeling language used in software engineering to visualize, specify, construct, and document software systems. UML provides a comprehensive framework for representing different aspects of software systems through various types of diagrams that facilitate communication among stakeholders and support the software development process.

According to Sommerville (2016), UML serves as a crucial tool for system analysis and design by offering multiple diagram types, each serving specific purposes in system modeling. The primary objective of UML is to provide a common language that can be understood by developers, analysts, and stakeholders regardless of their technical background, thereby enhancing collaboration and reducing misunderstandings during the development process.

UML encompasses several diagram categories that address different modeling perspectives. Sommerville (2016) explains that activity diagrams may be used to model the processing of data, where each activity represents one process step. These diagrams are particularly useful for illustrating workflows and business processes, making them valuable for understanding complex system operations and user interactions within software applications.

Sequence diagrams represent another fundamental component of UML, which are used to model interactions between system components, although external agents may also be included (Sommerville, 2016). These diagrams focus on the temporal aspect of system interactions, showing the order in which messages are exchanged between different components over time. This temporal perspective is essential for understanding how distributed systems and multi-component architectures operate.

According to Sommerville (2016) states that use case diagrams and sequence diagrams are used to describe the interactions between users and systems in the system being designed. Use cases describe interactions between a system and

external actors; sequence diagrams add more information to these by showing interactions between system objects. This combination provides a comprehensive view of both external user interactions and internal system behavior.

The application of UML in software development projects offers several advantages, including improved communication among project stakeholders, better documentation of system requirements and design decisions, and enhanced understanding of system complexity. UML diagrams serve as blueprints that guide the implementation process and provide reference documentation for future maintenance and system evolution.

In the context of educational technology systems and dashboard development, UML modeling becomes particularly valuable for representing complex interactions between multiple system components, user roles, and data processing workflows. The standardized notation ensures that system designs can be effectively communicated across different phases of the development lifecycle and among team members with varying technical expertise.

II.10 User Acceptance Testing

User Acceptance Testing (UAT) is a system evaluation methodology that focuses on validating user acceptance of the developed information system. Sommerville (2016) defines UAT as the final testing phase aimed at verifying that the system meets business needs and is acceptable to end users. Unlike other technical tests, UAT emphasizes the perspective of actual users in the operational environment.

IEEE Computer Society (2008) in the system testing documentation standards establishes UAT as a formal test to determine system acceptance criteria. This standard emphasizes the importance of validating requirements from the user's perspective and the traceability of test results to the established criteria. In the context of the student competition dashboard system, UAT becomes a relevant method to ensure the system can be accepted by various stakeholders with different backgrounds and needs.

The main characteristics of UAT according to Sommerville (2016) include the involvement of representative users, the use of realistic data and scenarios, a focus

on business workflows, and generating go/no-go decisions for system deployment. This evaluation method was chosen in the research due to its ability to identify the gap between user expectations and the technical implementation that has been carried out.

II.11 System Usability Scale

System Usability Scale (SUS) is a usability measurement instrument developed by Brooke (1996) to measure users' subjective perceptions of the ease of use of a system. SUS uses a quantitative approach through 10 question items with a Likert scale that produces a score of 0-100, providing a global measure of the system's usability. The methodological construction of SUS is based on a sophisticated Likert scale technique, where the question items are selected based on their discriminative ability towards systems with different levels of usability.

Brooke (1996) explains that although designed as a "quick and dirty" instrument, SUS has a solid theoretical foundation with high item intercorrelation and a combination of positive-negative statements to reduce response bias. The alternating positive and negative statements in SUS questionnaire are specifically designed to prevent acquiescence bias, where respondents tend to agree with statements regardless of their actual opinion.

Sauro and Lewis (2016) validated the superiority of SUS through comprehensive research, demonstrating high reliability, proven validity, sensitivity to usability differences, and the ability to be used with a relatively small sample size. Their extensive analysis confirmed that SUS provides consistent and meaningful measurements across different types of systems and user populations.

The SUS scoring system follows a standardized interpretation framework that categorizes usability levels based on numerical scores. The grading system provides clear benchmarks for evaluating system usability performance, as shown in Table II-1. This benchmark helps determine whether the system meets minimum usability expectations or requires further improvement. By referring to these standardized scores, stakeholders can easily understand the level of user satisfaction and overall system acceptability.

Table II-1. Evaluation of the system usability scale (SUS)

Grade	SUS	Adjective	Acceptable
A+	84.1–100	Best Imaginable	Acceptable
A	80.8–84.0	Excellent	Acceptable
A-	78.9–80.7	Good	Acceptable
B+	77.2–78.8		Acceptable
B	74.1–77.1		Acceptable
B-	72.6–74.0		Acceptable
C+	71.1–72.5		Acceptable
C	65.0–71.0	OK	Marginal
C-	62.7–64.9		Marginal
D	51.7–62.6		Marginal

Based on Table II-1 to calculate the System Usability Scale (SUS) score, the questionnaire answers need to be adjusted. Odd-numbered questions are calculated using Formula II-1.

$$Si = Xi - 1, i = \text{Odd} \quad \text{(II-1)}$$

Meanwhile, for even numbered questions, the score is adjusted using the following Formula II-2.

$$S_i = 5 - X_i\,, i = \text{Even} \quad \text{(II-2)}$$

After the scores for each question are adjusted, all scores are added together and then multiplied by 2.5 to obtain the final SUS score, as shown in Formula II-3.

$$SUS = \sum_{i=1}^{10} S_i \text{ x } 2.5 \quad \text{(II-3)}$$

The SUS scores from each respondent are then averaged to obtain an overall usability score, according to the II-4 Formula.

$$Mean\ SUS = \frac{Total\ SUS}{N} \quad \text{(II-4)}$$

Explanation:

- X_i = the answer to question-i.
- S_i = adjusted score.
- N = number of respondents.

Where Xi represents the answer to question-i, Si represents the adjusted score, and N represents the number of respondents.

In the context of dashboard system evaluation, SUS was chosen as the measurement method due to its ability to provide a standardized and comparable assessment of

system usability. The established grading system allows for clear interpretation of results and benchmarking against industry standards for usability performance.

II.12 Laravel Dusk

Laravel Dusk is an automated testing framework developed for Laravel applications with a focus on browser automation and end-to-end testing. According to Stauffer (2019) explains that Dusk allows for the simulation of real user interactions in the browser, providing comprehensive coverage of the integration between system components.

According to official Laravel documentation (2023) as the primary source explains that Dusk provides an expressive API for browser testing with the capability of JavaScript execution and interaction with CSS selectors. This framework supports testing modern interactive web applications, making it relevant for validating systems that integrate multiple technologies.

In the context of testing strategy, Fowler (2018), as a recognized authority in software engineering, explains the concept of the testing pyramid, which places browser testing tools at the highest level. This concept has become a widely accepted framework in software development, providing guidance for balancing execution speed and testing confidence. Dusk was chosen in this research due to its ability to validate complete user workflows and provide high confidence that the system works as expected from the end-user's perspective.

II.13 Layered Architecture

Layered architecture is one of the most commonly used software architecture patterns, also known as the n-tier architecture pattern. This pattern organizes system components into horizontal layers that interact with each other to achieve the overall system goals (Richards, 2015).

According to Meyer & Webb (2005), layered architecture allows for separation of concerns where each layer has specific responsibilities and communicates only with adjacent layers. Tu (2023) explains that the implementation of layered architecture in modern software development allows for better modularity, maintainability, and scalability.

Layered architecture consists of several standard layers, each with specific functions. Richards (2015) identifies four main layers: the presentation layer for handling the user interface and browser communication, the business layer for executing business rules, the persistence layer for managing data access, and the database layer for data storage. Tu (2023) emphasizes that each layer focuses on specific tasks with clear and separate responsibilities.

An important concept in this architecture is "closed layers" where requests must pass through the underlying layers sequentially, and "layers of isolation" that ensure changes in one layer do not affect other layers (Richards, 2015).

Tu (2023) identifies three main principles in the design of layered architecture. First, the principle of separation of concerns, which divides functional modules into different layers with a focus on specific tasks. Second, the principle of dependency inversion, which emphasizes dependence on abstraction rather than concrete implementation using interfaces or abstract classes. Third, the principle of single responsibility, which requires each layer to have clear and specific responsibilities.

Meyer & Webb (2005) added two basic principles: subsystem focus, which requires each subsystem to have a single, concentrated purpose, and single-function interfaces, which emphasize that each subsystem should have a single interface for input and output data.

Layered architecture offers several key advantages. Tu (2023) mentions that this pattern provides excellent modularity and decoupling, improves system maintainability, offers good scalability, and promotes code reuse. Richards (2015) adds that this pattern has high testability because components can be mocked for isolation testing, as well as high ease of development because it is widely known and not too complex to implement.

Meyer & Webb (2005) emphasize that layered architecture provides the necessary foundation for effective mass customization in software, allowing components to be easily combined in different ways without adding substantial additional code.

II.14 Methodology Waterfall

The Waterfall Model is one of the oldest and most fundamental software development methodologies in software engineering. This model was first introduced by Royce (1970) in his paper "Managing the development of large software systems" as a linear and sequential approach to software development, where each phase must be fully completed before proceeding to the next phase.

Pressman and Maxim (2019) in "Software Engineering: A Practitioner's Approach" explain that the Waterfall Model is very suitable for projects with characteristics of clear and stable requirements, defined timelines, comprehensive documentation, and complex component integration. This model emphasizes a systematic and well-documented approach that is essential for academic and research projects.

The Waterfall methodology follows a structured sequence of phases that must be executed in order, as illustrated in Figure II-1. Each phase has specific deliverables and serves as input for the subsequent phase, ensuring systematic progression throughout the development lifecycle.

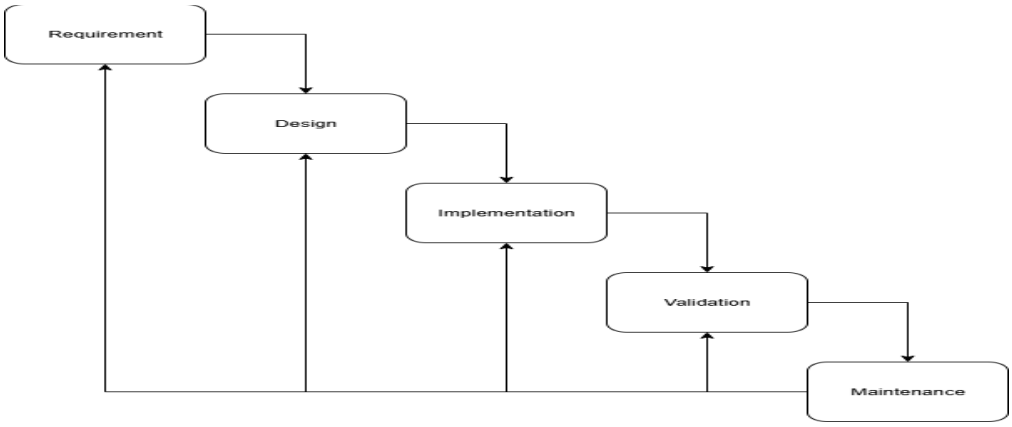

Figure II-1. Waterfall model phases and process flow

Source (Pressman and Maxim, 2019)

Based on Figure II-1, the Waterfall Model consists of five primary phases that form a sequential development process. The Requirement phase involves comprehensive analysis and documentation of system requirements, establishing the foundation for all subsequent development activities. The Design phase translates requirements

into system architecture, interface designs, and technical specifications that guide implementation efforts.

The Implementation phase focuses on the actual construction of the system based on the design specifications, involving coding, component development, and system integration. The Validation phase encompasses testing activities to ensure the system meets specified requirements and functions correctly. Finally, the Maintenance phase addresses ongoing system support, updates, and modifications after deployment.

An important characteristic of the Waterfall Model, as shown in Figure II-1, is the presence of feedback loops that allow for iterative refinement within and between phases. While the model follows a primarily linear progression, these feedback mechanisms enable developers to address issues and incorporate improvements discovered during later phases, ensuring quality and requirement compliance.

Sommerville (2016) in "Software Engineering" identifies that the Waterfall methodology provides a clear structure and predictable outcomes, making it particularly suitable for projects that require formal documentation and regulatory compliance. These characteristics make Waterfall the right choice for projects with strict requirement documentation, such as in the context of academic research.

The sequential nature of the Waterfall Model offers several advantages for complex system development. Each phase produces specific deliverables that serve as milestones for project progress evaluation. The comprehensive documentation requirements ensure that all stakeholders have clear understanding of system requirements, design decisions, and implementation approaches. This documentation becomes particularly valuable for academic projects where detailed process recording and methodology justification are essential.

In the context of system development that integrates multiple technologies such as dashboard visualization, Natural Language Interface to Database (NLIDB), and AI APIs, the Waterfall approach provides a systematic structure to manage integration complexity. This methodology supports the needs of academic research that require comprehensive documentation and structured progression at each stage of

development, as recommended by Pressman and Maxim (2019) for software development projects in educational environments.

II.15 Method Reasons

The selection of the Waterfall methodology for this research is based on a comparative analysis of various available software development methodologies. Royce (1970) in his seminal paper explained that the waterfall model uses a systematic and procedural linear approach, very suitable for projects with well-defined and stable requirements.

Comparison with other methodologies shows different characteristics. Boehm (1988) in "A spiral model of software development and enhancement" developed a model that combines waterfall and prototyping with a focus on risk management, but requires expertise in risk analysis and a longer development time. Beck et al. (2001) in "Manifesto for agile software development" emphasize an iterative approach with frequent customer collaboration, which is less suitable for an academic context with a fixed timeline and limited stakeholder involvement.

Martin (1991) in "Rapid Application Development" emphasizes rapid prototyping and active end-user participation, which is not suitable for the complexity of multi-technology integration in this research. Davis (1992) in "Operational prototyping: A new development approach" proposes an evolutionary approach that is effective for unclear requirements, but in this research, the requirements are already stable based on the needs analysis that has been conducted.

Based on that analysis, the Waterfall Model is the most suitable choice because: (1) Requirements are already stable based on the needs analysis of the dashboard system with NLIDB integration, (2) An academic timeline that is fixed and clearly defined, (3) The need for comprehensive documentation for academic evaluation, (4) Complex multi-technology integration that requires a systematic approach, and (5) The characteristics of an academic project that do not require rapid iteration or continuous user feedback as in commercial software development, as shown in Table II-2

Table II- 2. Method reason

Method	Brief Description	Relevant / Not Relevant Reason	Process
Waterfall Model	Sequential methods that proceed in order from the initial stage to the final stage (Pressman & Maxim, 2019; Sommerville, 2016).	Suitable for projects with stable requirements, fixed academic timelines, and the need for good documentation and quality control. Supporting systematic multi-technology integration.	1. Requirement Analysis 2. System Design 3. Implementation 4. Validation 5. Maintenance
Agile / Scrum	An iterative method that relies on flexibility, team collaboration, and quick response to changes (Beck et al., 2001).	Not suitable because this project does not involve repeated client interactions, and does not allow for quick sprints within the limited thesis timeline.	1. Sprint Planning 2. Sprint Execution 3. Daily Standup 4. Review 5. Retrospective Maintenance
Spiral Model	Combining the Waterfall approach and prototyping with a primary focus on risk management (Boehm, 1988).	Requires risk management expertise and a long development time, making it less relevant for small-scale projects such as final assignments.	1. Planning 2. Risk Analysis 3. Engineering 4. Evaluation 5. Next Iteration Planning
RAD (Rapid App Dev)	Emphasizing rapid prototyping and active end-user participation for application iteration (Martin, 1991).	Emphasizing rapid prototyping and active end-user participation for application iteration (Martin, 1991).	1. Requirements Planning 2. User Design 3. Construction 4. Cutover
Prototyping Model	Building an early version of the system to obtain user feedback before finalizing the system (Davis, 1992).	Effective for projects with unclear requirements. However, in this research, the requirements are already stable and have been analyzed at the beginning through interviews and observations of the existing system.	1. Quick Design 2. Build Prototype 3. Evaluate with User 4. Refine Prototype 5. Develop Final System

Based on the comparative analysis on Table II-2, the Waterfall Model is the most suitable methodology for this research. This dashboard development project has stable requirements based on interviews and observations of the existing system, a

fixed and clearly defined academic timeline, and requires a well-documented process for academic evaluation purposes.

The Waterfall Model allows the development process to be carried out systematically and in a controlled manner, with each stage being evaluated step by step before proceeding to the next stage. This greatly supports the needs of academic research that require complete documentation and complex technology integration, as suggested by Pressman & Maxim (2019) and Sommerville (2016) for software development projects in educational environments.

II.16 State of the Art

To position this research within broader academic context and demonstrate its novelty, a comprehensive review of relevant previous studies was conducted. This systematic analysis focuses on dashboard system development in educational environments, data visualization technologies in higher education, and artificial intelligence application for educational data analysis and decision-making.

The state of art review serves critical purposes in establishing research foundation. It provides thorough understanding of current trends in educational dashboard systems integrating advanced technologies like artificial intelligence and machine learning. It identifies strengths and weaknesses of existing approaches, highlighting where current solutions fall short of meeting comprehensive institutional needs. It establishes clear research gaps this study addresses, demonstrating how the proposed solution differs from and improves upon existing implementations.

A detailed review of contributions, methodologies, limitations, and outcomes of each relevant study is presented in Table II-3. This comparative analysis examines system architecture, technology stack, AI integration approaches, user interface design, and evaluation methods employed by previous researchers. The table serves as foundational basis for identifying specific research gaps this study addresses, justifying the need for developing an integrated competition dashboard with AI technology specifically tailored for Faculty of Industrial Engineering at Telkom University.

Table II- 3. Previous research

Title	Description	Contribution	Limitation
Analysis of the Impact of Artificial Intelligence Use on Learning at Cyber University (Sitorus & Murti 2024)	This study analyzes AI adoption patterns and educational impact at Cyber University using qualitative descriptive methodology. The research demonstrates widespread AI integration in online higher education, with empirical data showing 70% regular students and 80% employee-class students actively utilizing AI tools, predominantly ChatGPT, for learning enhancement. The study focuses on understanding behavioral patterns, challenges, and opportunities of AI implementation in technology-based educational environments.	Provides empirical evidence of high AI adoption rates in educational institutions, establishing the demand and acceptance of AI-integrated systems in academic environments. Demonstrates that students actively seek AI-powered tools for learning support, justifying the need for institutional systems that can leverage AI capabilities. Identifies key challenges including data privacy concerns and digital literacy gaps that must be addressed in AI system development. Establishes foundation for understanding user requirements and acceptance factors when developing AI-integrated educational systems.	Study focuses on educational impact analysis rather than technical system development, creating a research gap in AI-integrated dashboard implementation. Limited to single institution with no technical architecture or system design considerations. Lacks technical specifications for AI integration in institutional management systems. No evaluation of dashboard interfaces or competition management system requirements. Missing system development methodology and technical implementation guidelines for AI-powered educational tools.
Implementation of Artificial Intelligence Use in Higher Education (Sandy, F.,	This research explores the implementation of artificial intelligence in the context of higher	The research results indicate that students' experience in using artificial intelligence (AI) in the	Limited sample, no integrated monitoring system, lack of in-depth

Title	Description	Contribution	Limitation
Palangi, W. A., Liling, D., & Pratama, M. P., 2023)	education using a descriptive qualitative approach with a questionnaire instrument on the 2021 cohort of Educational Technology students. The research results show that 50% of students often use AI in the learning process to find task references and improve time capabilities. In terms of impact, 50% of respondents feel that AI is very helpful, while the other 50% feel it is somewhat helpful in the learning process. The research emphasizes that AI plays an important role in supporting the learning process of students by enhancing their understanding of the learning material.	learning process is quite significant. On average, 50% of respondents stated that they often use AI, 20% very often, 20% occasionally, and the remaining 10% never use it. Students feel that the use of AI is very helpful in finding task references and improving time capabilities. In terms of impact, 50% of respondents feel that AI is very helpful in the learning process, while the other 50% feel that AI is somewhat helpful. They expressed that the use of AI enhances their understanding of the learning material. Overall, this research emphasizes that artificial intelligence plays an important role in supporting the learning process of students. Overall, this research emphasizes that artificial intelligence plays an important role in supporting the learning process of students.	analysis on long-term effectiveness
Designing a Performance Dashboard in the Field of	This research focuses on designing a performance dashboard for the	Successfully created a dashboard design with 29 performance metrics	This research focuses on the design and readiness

Title	Description	Contribution	Limitation
Student Affairs and Graduates (Case Study: Industrial Engineering, Sebelas Maret University) (Noviana et al., 2023).	student and graduate sectors in the S1 Industrial Engineering Study Program at Sebelas Maret University.	across 7 pages, tailored to the needs of the study program. This dashboard achieved a user satisfaction level of 7.6 and an implementation readiness level of 86.83%. Additionally, this research produced a data entry template and a guideline book to facilitate implementation.	dashboard with static performance metrics. There is no AI integration to provide insights, in-depth analysis, or more dynamic decision support based on performance data.
DashBot: Insight-Driven Dashboard Generation Based on Deep Reinforcement Learning (Deng et al. 2022)	This research presents DashBot, an insight-driven dashboard generation system based on deep reinforcement learning for automated analytical dashboard creation. The study addresses the challenge of creating effective business intelligence dashboards that traditionally requires extensive data analysis expertise and familiarity with professional tools like Tableau and Power BI. The research employs Asynchronous Advantage Actor-Critic (A3C) framework to formulate dashboard generation as a Markov Decision Process (MDP), where agents explore dashboard	This study provides several significant contributions to AI-powered visualization systems: (1) Novel RL formulation for dashboard generation problem, translating exploratory visual analysis into sequential decision-making process with well-designed state-action spaces; (2) Comprehensive reward function framework combining presentation rewards (diversity, parsimony) and insight rewards (single-column, double-column, multiple-column insights) derived from empirical dashboard design studies; (3) Constrained sampling methodology that integrates	The study presents several constraints: Limited visualization scope with only basic chart types (bar, line, point, boxplot) and maximum 10-column datasets, insufficient for complex enterprise analytics. Reward function bias derived from only 90 dashboard samples from specific platforms (Tableau/Power BI), potentially limiting design diversity. Lack of

Title	Description	Contribution	Limitation
	design space through actions (change key column, add/remove charts, terminate) and receive rewards based on presentation quality and data insights. The system incorporates constrained sampling mechanisms to ensure valid chart configurations and utilizes bidirectional LSTM networks for modeling chart relationships within dashboards.	visualization design rules to ensure valid chart configurations while preserving agent exploration capabilities; (4) Sequential agent network architecture using bidirectional LSTM for chart relationship modeling and parameter prediction with embedding concatenation; (5) Empirical validation through ablation studies and comparative user studies against MultiVision, demonstrating superior performance in dashboard quality, understandability, and insightfulness (78% preference rate).	interactivity and semantic understanding in generated dashboards, missing essential features for modern business intelligence and domain-specific insight discovery. Narrow evaluation scope limited to comparison with MultiVision on Vega datasets, lacking validation against industry-standard tools and real-world data scenarios.
Development of an Interactive Student Dashboard in the Computer Engineering Study Program Using Big Data Technology (Freska Rolansa et al., 2021).	This research discusses the development of an interactive dashboard to monitor student performance. This dashboard is designed to visualize data in an easily understandable manner.	Successfully provided an interactive visual display of student performance data using Tableau and big data. This dashboard facilitates the monitoring of student activity status, graduation, and achievements through graphs and tables. Its implementation enhances management effectiveness in monitoring and evaluating academic performance in real-time, supporting timely decision-making.	The developed dashboard focuses on academic data and has not yet integrated artificial intelligence for analysis, strategic recommendations, or natural language-based interactions. Its focus is specifically on academic data and doesn’t yet cover competitions in detail.

Based on comprehensive review of previous research in Table II-2, several key insights emerge regarding dashboard systems and AI integration development in educational environments. The research landscape demonstrates consistent evolution from basic data visualization systems toward more intelligent and interactive platforms.

Earlier studies, particularly Freska Rolansa et al. (2021) and Noviana et al. (2023), established foundation for dashboard development in educational institutions, focusing primarily on data presentation and performance monitoring capabilities. These works successfully demonstrated visual data representation value in supporting institutional decision-making processes. However, they primarily addressed the "what" aspect of data presentation without exploring "why" and "how" dimensions that deeper analytical capabilities could provide.

Recent research, including Deng et al. (2022) and AI implementation studies by Sitorus & Murti (2024) and Sandy et al. (2023), reveals significant shift toward artificial intelligence integration in educational technology. These studies highlight growing recognition that static dashboard systems, while useful for basic monitoring, require enhancement through intelligent features to meet evolving educational stakeholder needs.

Previous research analysis reveals a consistent gap this study specifically addresses: integrating comprehensive dashboard visualization with intelligent, conversational AI capabilities in unified educational platform. While existing research explored these components separately, none successfully combined them into cohesive system for student competition monitoring.

Unlike previous studies focusing on either dashboard development or AI implementation separately, this research implements novel 4-layer architecture seamlessly integrating Google Looker Studio, Flowise AI, OpenAI GPT-4o, and Laravel into unified platform. This architectural approach has not been previously demonstrated in educational dashboard literature.

While Deng et al. (2022) explored AI-driven dashboard generation, their approach focused on automated visualization creation rather than interactive query

capabilities. This research implements sophisticated NLIDB system allowing stakeholders to perform complex data analysis through conversational Indonesian language queries, addressing specific linguistic and cultural requirements not covered in previous work.

Previous research has primarily focused on general academic performance or broad educational metrics. This study specifically targets student competition monitoring, requiring specialized data models, performance indicators, and analytical frameworks that reflect the unique characteristics of competitive academic achievements.

The combination of Waterfall methodology with layered architecture for AI-integrated educational systems represents a methodological contribution that extends beyond existing literature. The systematic approach to integrating multiple cloud-based technologies (Aiven, Looker Studio, Flowise AI) within an academic development framework provides a replicable model for similar implementations.

Unlike previous studies that primarily serve administrative users, this research implements role-based access control that specifically addresses the needs of multiple stakeholder types (faculty management, student affairs staff, and students) with differentiated functionality and user experiences.

This research advances the field by demonstrating practical feasibility of implementing Natural Language Interface to Database (NLIDB) technology for educational performance monitoring. Successful integration of multiple cloud-based platforms within layered architecture framework provides a technical blueprint adaptable for similar educational applications.

The study contributes to understanding AI adoption in higher education by providing empirical evidence of stakeholder acceptance and usability performance for AI-integrated dashboard systems. The comprehensive evaluation methodology including automated testing, User Acceptance Testing, and System Usability Scale assessment establishes benchmarks for similar implementations.

CHAPTER III
METHODS

III.1 Framework

In this final project, the author will build a student competition dashboard system integrated with Artificial Intelligence as an intelligent assistant for the Student Affairs and Faculty of Industrial Engineering (FRI) at Telkom University. This system will utilize historical competition achievement data to provide insights and recommendations that support strategic decision-making. Next, the author will describe the framework model to identify the components used to illustrate the problems in the thesis preparation process. The framework model used in this thesis preparation will implement the framework model framework (Hevner, March, Park, & Ram, 2004). The reason for using this framework method is that it can illustrate the purpose of having an information system, which is to implement a system using technology within an organization to improve the effectiveness and capabilities of that organization by adjusting its resource needs (Hevner, March, Park, & Ram, 2004). The framework model that will be used in this thesis can be seen in the illustration in Figure III-1.

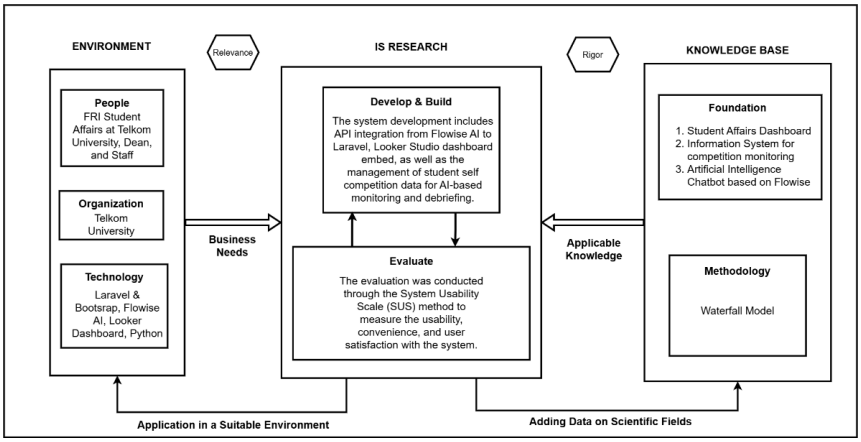

Figure III-1. Problem solving framework

Based on figure III-1. described framework model, this research formulates a problem-solving framework that integrates an independent competition dashboard with artificial intelligence (AI) technology for monitoring and analyzing student performance achievements at the Faculty of Industrial Engineering (FRI) Telkom University. This problem-solving framework is designed with consideration of

three main dimensions: environment, information system (IS) research, and scientific foundation. The involved environment includes individuals such as FRI students, lecturers, and staff, the Telkom University organization, as well as technologies like Laravel & Bootstrap, Flowise AI, Looker Dashboard, and Python. In the realm of IS research, the development and construction of the dashboard are carried out through API integration from Flowise AI to Laravel and embedding the Looker Studio dashboard, as well as managing student independent competition data for AI-based monitoring and debriefing. System evaluation is conducted using the System Usability Scale (SUS) method to measure usability, ease of use, and user satisfaction. The scientific foundation underlying this research includes the basis of the Student Dashboard, Information System for competition monitoring, as well as Flowise-based Chatbot, and the Waterfall Model methodology.

The architecture of the developed system consists of four main layers. Layer 1 as the Presentation Layer uses the Laravel Web Application, which provides a responsive and mobile-friendly user interface, as well as an authentication and authorization system. Layer 2 as the Intelligence Layer utilizes the Flowise AI Platform for the development of AI chatbots and integration with the OpenAI API (GPT-4o) for natural language processing. Flowise AI and the OpenAI API are integrated into Laravel via iframe. Layer 3 as the Visualization Layer uses Google Looker Studio for the development of interactive dashboards that embed dynamic graphic and analytic components, also integrated into Laravel via iframe. Layer 4 as the Data Layer uses Aiven MySQL Cloud Database for the management and storage of historical competition data, as well as query optimization.

Thus, this research produces a dashboard that meets environmental needs, applies appropriate methodologies, and is based on a strong scientific foundation. The designed system integration ensures that the dashboard not only provides good data visualization but also offers interactive insights and strategic recommendations through an AI chatbot to support better decision-making and more effective student mentoring.

III.2 Systematization of Problem Solving

This research uses the Waterfall Model as the system development methodology. The Waterfall Model was chosen because it aligns with the project characteristics that have clear and stable requirements, a defined timeline, and the integration of multiple technology components that require a sequential and systematic approach.

According to Royce (1970), the Waterfall Model is a linear and sequential approach suitable for projects with the following characteristics:

1. Clear and stable requirements
2. Defined timeline
3. Comprehensive documentation
4. Integration of multiple components

The characteristics of this project meet all the above criteria, so the Waterfall Model is the right choice.

Figure III-2 illustrates the systematic approach to addressing the research problem:

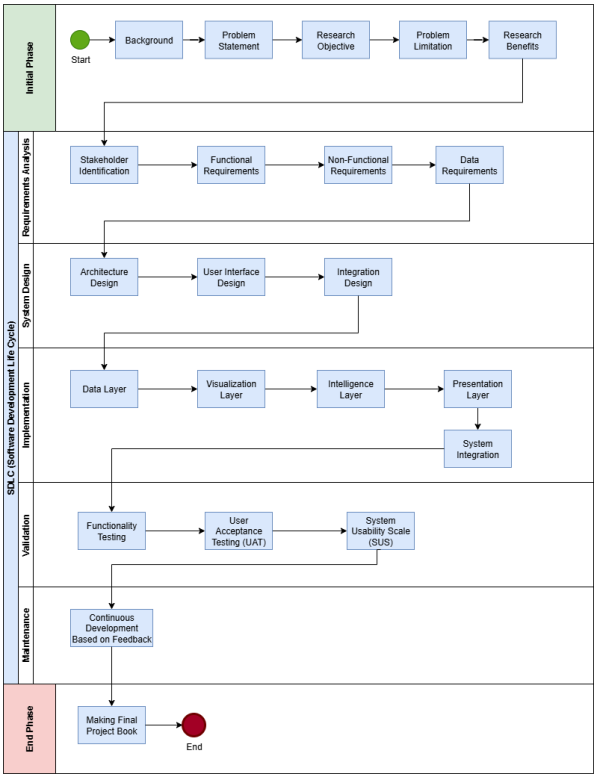

Figure III- 2. Systematization of problem solving

Based on Figure III.2, the systematic preparation of problem solving namely using the WATERFALL method, consists of five stages, namely:

1. Initial Phase

   The initial phase of this research focuses on building the framework and methodological foundation that underpins the entire system development process. The research began by identifying the main issues related to the limitations of the student achievement monitoring system in independent competitions within the Faculty of Industrial Engineering (FRI) at Telkom University. Based on initial observations, the existing dashboard system is not yet fully capable of providing informative visualizations and is not integrated with artificial intelligence (AI) technology, which enables predictive analysis and conversation-based insights.

   As a follow-up to the identification of these issues, a background and literature review on dashboard technology, AI integration in higher education, and student performance monitoring systems in general were conducted. This study reinforces the urgency of developing a system that not only presents data but also actively plays a role in providing strategic recommendations to student affairs and faculty management.

   From the study process, three main research questions were formulated that became the focus of this research, namely, how to develop a dashboard integrated with AI, how to utilize AI to provide deeper insights, and how to improve the capabilities of monitoring student performance through an adaptive and interactive system. The objective of this research was then established in line with the formulation of the problem, which is to develop an AI-based dashboard using Laravel technology, Flowise AI, OpenAI API, and Looker Studio that can assist in the analysis and decision-making process.

   To maintain the focus of the research within the constraints of time and resources, the scope of the research is defined specifically. This research only covers independent competition data participated in by FRI students at Telkom University over the past five years, with the data sourced from the data

warehouse that has been integrated into the Aiven MySQL platform. In addition, the benefits of the research were also identified, both for FRI as an institution, the student team as the data achievement managers, and for other researchers who wish to develop a similar system in the future.

This initial phase also marks the beginning of the system development process using the Waterfall Model approach, which was chosen because it aligns with the project's characteristics that have clear system requirements, defined deadlines, and the use of several different technology components. With the Waterfall approach, each stage of development is carried out sequentially and systematically, starting from needs analysis, system design, implementation, testing, to final documentation.

The initial stage of this research begins with understanding the main needs and problems faced by the institution, namely the Faculty of Industrial Engineering (FRI) at Telkom University, in monitoring student competition achievements. Issues such as the lack of integration of student competition data with Artificial Intelligence-based analysis tools became the main focus. This stage aims to understand the context of the issues and the needs in supporting the implementation of student-organized competitions. The research begins by explaining the background, namely the importance of efficiently monitoring the performance of independent student competitions through the integration of Artificial Intelligence (AI) into a Looker-based dashboard. The problem formulation focuses on the challenges in processing competition data and providing AI-based insights that are not yet optimal. The research aims to develop an integrated solution that utilizes Looker for monitoring dashboards and Flowise AI for AI-based analysis. The scope of the research is limited to independent student competition data at the Faculty of Industrial Engineering (FRI) Telkom University, with implementation based on Laravel. The literature review includes the WATERFALL method, AI integration, and related studies to support the reliability of the proposed solution. This research is expected to provide academic and practical benefits, such as improving the capabilities of competition data management and supporting data-driven decision-making.

2. Requirement Analysis

This stage aims to understand the system requirements that must be met in developing an integrated dashboard with AI technology for monitoring students in independent competitions. The initial stage is conducted with a needs analysis involving various stakeholders, namely FRI lecturers, student affairs staff, and the students themselves. In the analysis, several important aspects that need to be met were identified, including the need for a visualization dashboard to monitor competitions, the need for an AI chatbot to provide insights and recommendations, and the need for data integration from the Aiven database. Additionally, non-functional considerations such as system performance (response time and the number of users that can be accessed simultaneously), data security (user access), and system usability (user interface design) were also taken into account. In detail, an analysis of the FRI independent competition data structure was conducted to ensure that the data to be used in the system can be processed well, including data arrangement and preparation requirements as well as database connectivity to Aiven. With this approach, it is hoped that the developed solution will meet the expectations desired by the involved parties.

3. System Design

The system design phase aims to outline the technical architecture of the system to be developed, including the integration of dashboard visualization, AI-based chatbots, and database management into a structured workflow. At this stage, the system design focuses on how each component can work in an integrated and seamless manner, while still considering user experience and system performance in real-world usage.

The system is designed by adopting and adapting the layered architecture pattern (Richards, 2015; Meyer & Webb, 2005) implemented in four main layers consisting of the Presentation Layer, Intelligence Layer, Visualization Layer, and Data Layer. The Presentation Layer is at the top layer and is built using the Laravel framework, which serves as the main user interface. This

component includes authentication and authorization systems, as well as embedding mechanisms to integrate the chatbot interface and visual dashboard into web pages responsively.

At the next layer, the Intelligence Layer serves as the center of artificial intelligence built using the Flowise AI platform. The chatbot is developed by connecting Flowise to the OpenAI API, specifically the GPT-4 model, to support natural language processing. This integration is carried out by embedding the chatbot interface into Laravel via an iframe, allowing users to interact directly through the provided web interface.

Meanwhile, the Visualization Layer is built using Google Looker Studio, which presents visualizations of student competition data in the form of graphs, diagrams, and other interactive analytical elements. This dashboard component is also integrated into Laravel via an iframe, allowing users to view and interact directly with historical data in a dynamic and intuitive manner.

The final layer, the Data Layer, uses Aiven as the provider of cloud-based MySQL services. This layer is responsible for managing competition data independently, including storing historical data, configuring database schemas, and optimizing connections to ensure the accuracy of data retrieval by both the visualization system and the chatbot.

In addition to the system architecture design, user interface design is also an important aspect in this phase. The interface design is crafted to be responsive and compatible across various devices (multi-device), with navigation that is easily understood by different types of users, ranging from administrators to student teams. The visual components are also designed to meet accessibility standards so that the system can be comfortably used by all stakeholders.

This stage also includes planning integration strategies between components, including embedding mechanisms for Looker Studio and Flowise AI, and designing data flows between layers. With a well-thought-out and structured system design, this phase becomes an important foundation for the implementation process in the next stage.

4. Implementation

The implementation phase is the stage where the previously designed system begins to be realized into an application that can be run and tested in a real environment. The main focus in this phase is the consistent application of design specifications, the development of system components with maintained code quality, and ensuring reliable integration between layers.

The implementation begins with building the Data Layer as the foundation of the system, using Aiven MySQL cloud service as the database. Initial activities include setting up the database environment, implementing the database schema based on the independent competition data structure, and the process of migrating data from Excel files to the cloud database. Connectivity testing and query optimization are conducted to ensure that the database can be accessed quickly and stably by all other system components.

Next, implementation is carried out on the Visualization Layer using the Google Looker Studio platform. At this stage, developers configure the connection between Looker Studio and the Aiven database, build interactive visualization components such as graphs and dynamic tables, and optimize the dashboard's performance to present data in real-time and be responsive to user needs.

The next stage is the development of the Intelligence Layer, which focuses on creating an AI-based chatbot using the Flowise platform. This process includes designing and implementing the conversation flow, integrating with the OpenAI API for natural language processing, and connecting to the database so that the chatbot can provide responses based on actual data. With this combination, the chatbot is expected to provide insights, identify trends, and offer contextual recommendations for users as well as actionable insights.

After the main layers are completed, the implementation of the Presentation Layer using the Laravel framework will follow. At this layer, a responsive and user-friendly interface is built, including the development of login features, user authorization, and the embedding of chatbot and dashboard views into the web page using the iframe method. The interface is designed to be optimally

accessible through various devices, both desktop and mobile, according to the needs of stakeholders.

The final stage in the implementation phase is the comprehensive system integration process, where inter-layer communication testing, end-to-end verification of all main system functions, and identification and handling of potential technical issues in the integration are conducted. With the completion of this stage, the system is ready to enter the further testing phase to ensure its quality and functionality comprehensively.

5. Validation

The validation phase aims to ensure that the developed system can meet user needs in terms of functionality, integration, ease of use, and the quality of artificial intelligence responses. Validation is carried out through a comprehensive testing approach, with Blackbox Testing as the basic method, supplemented by User Acceptance Testing (UAT), and the System Usability Scale (SUS) to measure the system's comfort level from the user's perspective.

Testing begins with Functionality Testing using the Blackbox Testing method, which focuses on validating input and output without reviewing the internal code structure. All main features were thoroughly tested, including user authentication (login and registration), competition data management, and input forms. Additionally, testing was conducted on the integration of components such as Laravel with Flowise AI and Looker Studio through iframe embedding. Data visualization on the dashboard is evaluated to ensure accuracy and readability. Some testing scenarios are also automated using Laravel Dusk to simulate user behavior directly.

Next, the system is validated through User Acceptance Testing (UAT), which involves end users from various stakeholder groups, such as FRI Lecturers, FRI Student Affairs Staff, and student representatives. At this stage, users are asked to run the system flow end-to-end in real-world scenarios. The purpose of UAT is to evaluate the completeness of features, ease of navigation, and the extent to which the system meets user expectations. User feedback is recorded to

determine whether the system has met the acceptance criteria based on the previously established functional and non-functional requirements.

Finally, the System Usability Scale (SUS) test was conducted as a standard method to measure the perception of system usability. The SUS questionnaire was distributed to representative users, with a target score of at least 70, reflecting above-average usability. Evaluation is conducted on the interface appearance, system navigation, and overall user experience, including the success rate of users in completing test scenarios (completion rate).

Through this layered testing approach, the system is validated not only from a technical and functional perspective but also from the end-user's viewpoint, ensuring that the system meets quality expectations and is ready for use in monitoring student performance within the FRI environment at Telkom University.

6. Maintenance

The maintenance phase is the final stage in the system development process, focused on the sustainability of the system's functions after implementation. In the context of this research, maintenance activities do not focus on managing infrastructure or complex technical operations, but rather on continuous development efforts that are iterative based on user feedback collected during the testing phase and after the system is used.

During the implementation and use of the system, various inputs from stakeholders such as lecturers, student teams, and students were recorded and analyzed. This input includes suggestions for interface improvements, system functionality, as well as enhancements in comfort and capabilities in daily use. Based on that feedback, developers can make gradual system improvements to ensure that the system remains relevant, responsive, and aligned with actual user needs.

This approach allows the system to continue evolving over time without requiring major interventions or structural changes. The main focus of this

phase is to maintain the sustainability of the system through regular evaluations, light refinements, and adjustments directly driven by user experience in the field.

7. End Phase

The final phase of this research focuses on the process of finalizing all the stages that have been undertaken, both from the system development side and academic documentation. At this stage, all results obtained from the phases of analysis, design, implementation, validation, and maintenance are systematically compiled into a final thesis book, which serves as the official research document. The writing of this book covers the entire system development cycle, from problem formulation, determination of objectives and scope, to system evaluation conducted through a structured testing approach.

One of the main outcomes of this research is the dashboard system developed using Looker Studio and integrated into the Laravel framework as the main interface. This dashboard displays student competition data visually and interactively to facilitate performance monitoring. Additionally, the system also implements an artificial intelligence model based on Flowise AI connected to the OpenAI API to help users obtain conversation-based analysis and relevant recommendations based on historical competition data. This integration makes the system not only a reporting tool but also a strategic decision-making aid.

The entire process and findings of this research have been thoroughly documented in the final thesis, which is expected to serve as an academic reference and make a tangible contribution to the development of information systems in higher education environments, particularly in the management and analysis of student competition data. With an approach that combines visualization and artificial intelligence, this system is expected to support the process of fostering student achievements more effectively, structured, and data driven. It is hoped that future researchers and institutions can adopt and further develop this system to address similar needs in other contexts.

III.3 Data Collection

This research uses various data sources to support the development of an autonomous competition dashboard integrated with AI technology at the Faculty of Industrial Engineering (FRI) Telkom University. The combination of primary and secondary data was obtained from several sources relevant to the needs and objectives of the research, thereby providing a strong foundation for developing an effective dashboard that meets user needs. Table III-1 provides details of the primary and secondary data used in this research:

Table III- 1. Data collection

No	Data Source	Data Type	Description
1	Stakeholder FRI	Primary Data	Structured interviews to identify system needs, functionality expectations, and pain points of the existing dashboard. Data includes functional requirements, non-functional requirements, and user experience expectations.
2	Excel file "Prestasi Kompetisi Mandiri - RAW.xlsx" from DITMAWA	Secondary Data	Historical data of independent student competitions at FRI for the period 2021-2024. The file contains 18 worksheets with a total of 6642 records consisting of various students and faculties within Telkom University.
3	Documentation of the Existing FRI Dashboard	Secondary Data	Screenshot, technical specifications, and user manual of the currently used independent competition dashboard. This data is used for gap analysis and requirement elicitation.

Based on Table III-1, this research uses two main types of data, namely primary data and secondary data. Primary data is obtained through direct interviews with key stakeholders of the Faculty of Industrial Engineering, namely FRI lecturers and Student Affairs Staff. This interview uses a structured approach to obtain in-depth insights regarding system needs, functionality expectations, and the identification of pain points from the existing dashboard currently in use.

Secondary data is the main component of the research obtained from two different sources. The primary source is the Excel file "Prestasi Kompetisi Mandiri - RAW.xlsx" obtained from the Directorate of Student Affairs (DITMAWA) of Telkom University. This file contains 18 worksheets with a complex data structure, encompassing a total of 6642 competition records for the period 2021-2024, consisting of various students and faculties within the Telkom University environment. Although the data covers the entire university, the focus of this research is on the extraction and analysis of data specifically for students of the Faculty of Industrial Engineering (FRI) for the development of a dashboard that meets the faculty's needs.

The documentation of the existing dashboard was collected as a secondary data source to conduct a gap analysis and understand the limitations of the current system. This documentation includes interface screenshots, technical specifications, and user manuals that serve as the baseline for the requirement specification of the new system. Analysis of the existing documentation helps identify improvement opportunities and feature enhancement priorities to be implemented in the AI-integrated dashboard system. The data collection strategy is designed to ensure a comprehensive understanding of the current state and future requirements, with a focus on stakeholder-driven development that prioritizes the actual needs of system users in the Faculty of Industrial Engineering environment.

III.4 Data Processing

At this stage, the author transforms the raw data collected from various sources into a format ready for system development. This process involves a systematic Extract, Transform, Load (ETL) pipeline to ensure data quality and alignment with the designed system architecture. Data cleaning and normalization are carried out to remove inconsistencies and duplicates that could affect analysis results. This step ensures that the final dataset is accurate, reliable, and suitable for integration into the competition dashboard system.

Below are the uses of the collected data as outlined in Table III-2 Data Processing for this research:

Table III- 2. Data Processing

No	Data Source	Data Usage
1	Stakeholder FRI Interview Results	Used for requirement elicitation and system specification. The results of the interviews are transformed into functional requirements, non-functional requirements, and use case diagrams that serve as the foundation for system design and development priorities.
2	Excel "Prestasi Kompetisi Mandiri - RAW.xlsx" (18 worksheets, 6642 records total)	Raw data was extracted with a focus on FRI students and consolidated using Python in Google Colaboratory. From 6642 records of all university study programs, filtering was done to produce 649 records specifically for FRI. Data cleaning was performed from 26 columns to 21 final columns, format standardization, and preservation of missing values to maintain the integrity of historical data. The output is a clean FRI dataset with complete historical accuracy ready for migration to the Aiven MySQL database.
3	Existing Looker Dashboard (obtained through a demonstration session with FRI Lecturers)	Used for gap analysis of the features and appearance of the current system. This process is carried out through direct observation of the dashboard presented by the faculty. The results of the observation are used to identify the shortcomings of the existing system as well as opportunities for improvement and feature enrichment in the new system. These findings serve as the initial reference in designing the UI/UX and determining the priority of feature development in the AI-integrated system.

Based on Table III-2, the entire needs analysis process was conducted systematically by collecting data from various sources, both primary and secondary. Primary data was obtained through interviews with stakeholders, including lecturers, the student affairs team, and students, to directly understand user needs. The results of the interviews were analyzed thematically to formulate the functional and non-functional requirements that will serve as the basis for system design.

Meanwhile, secondary data in the form of Excel files from the University of Telkom student competition were analyzed in depth to compile competition data

specifically for the Faculty of Industrial Engineering (FRI). From a total of 6642 rows of data spread across 18 worksheets, a filtering process was conducted based on the FRI study program criteria, resulting in 649 relevant data entries. This process is followed by standardizing column names, converting data types, and preserving missing values as part of the effort to maintain the historical integrity of academic data.

The final dataset obtained consists of 21 attributes and has been prepared for upload into the Aiven MySQL database. No imputation was performed on the missing data to avoid analysis bias and to maintain the transparency of historical data, in accordance with the "preserve, don't impute" principle suggested by Little & Rubin (2019). This process is carried out with an approach relevant to the scope of research at FRI, while also opening up the possibility of using this data for further development in the future.

Missing Value Preservation Strategy was chosen for several methodological reasons.:

1. Historical Accuracy: Maintaining the original condition of data as recorded in the system
2. Research Transparency: Avoiding bias that can be introduced through imputation methods
3. Future Compatibility: Allowing future research to use complete historical records
4. Academic Integrity: Respecting the integrity of existing institutional documentation

In addition to data from interviews and competition files, the researchers also conducted direct observations of the Looker dashboard currently being used by the faculty. The dashboard was shown by the FRI lecturer in the initial discussion session as a reference for the existing system. This observation is used to conduct a gap analysis on the functions, appearance, and interactivity of the dashboard, so that new features can be formulated to be more responsive and AI based to support student decision making.

III.5 Reasons for Method Selection

In this research, the selection of system development methodology is a critical and fundamental part of the study because it significantly affects the success of the development and evaluation process of the system. The methodology choice determines not only the technical execution approach but also influences project management, quality assurance, and the overall academic rigor of the research process. In the context of this research, a comprehensive analysis of several commonly used methodologies in software development was conducted to ensure optimal alignment between methodological approach and project characteristics.

The analysis encompassed multiple established software development methodologies including Agile, Scrum, DevOps, Rapid Application Development (RAD), and the traditional Waterfall Model. Each methodology was evaluated against specific criteria relevant to this research context, including project timeline constraints, requirement stability, stakeholder involvement patterns, technology complexity, documentation requirements, and academic evaluation needs. The aim is to determine the method that best fits the unique characteristics of the independent competition dashboard development project integrated with AI technology at the Faculty of Industrial Engineering (FRI) Telkom University.

The evaluation process considered several critical factors specific to this research environment. First, the academic timeline constraint requires a methodology that can accommodate fixed semester schedules and thesis submission deadlines. Second, the integration of multiple complex technologies including Laravel framework, Google Looker Studio, Flowise AI, and OpenAI GPT-4o necessitates a systematic approach that ensures proper testing and validation at each integration point. Third, the research context demands comprehensive documentation and traceability of development decisions for academic evaluation and future replication. The comparison of these methodologies is presented in Table III.3.

From the comprehensive comparison results, it can be concluded that the Waterfall Model is the most suitable methodology for this research project. Several key factors support this selection decision. This dashboard development project has relatively stable requirements that were clearly defined through stakeholder

analysis and literature review, minimizing the need for iterative requirement changes that would favor Agile approaches. The project operates within a fixed academic timeline with predetermined milestones that align well with Waterfall's sequential phases. Additionally, the research requires a well-documented process with clear phase deliverables that can be evaluated independently, supporting the academic assessment and thesis defense requirements.

The Waterfall methodology provides particular advantages for AI-integrated educational systems development. The sequential nature allows for thorough requirement analysis before design, ensuring that AI integration points are properly identified and planned. The design phase enables comprehensive architecture planning that accommodates the complex interactions between dashboard visualization, natural language processing, and database systems. The implementation phase benefits from the solid foundation established in previous phases, reducing integration risks associated with multiple cloud-based platforms.

Furthermore, Waterfall allows the development process to be carried out systematically and in a controlled manner, with stages that can be evaluated incrementally by academic supervisors and stakeholders. This systematic approach greatly supports the needs of academic research methodology and facilitates the integration of complex technologies through planned, sequential development phases. The methodology's emphasis on documentation aligns with academic research standards and provides clear traceability for future system maintenance and enhancement.

The selection is further validated by established software engineering literature, as suggested by Pressman & Maxim (2019), Sommerville (2016), and IEEE standards for educational software development projects, which recommend Waterfall for projects with stable requirements, fixed timelines, and comprehensive documentation needs characteristics that precisely match this research context. This alignment ensures that each development phase can be executed sequentially with clear deliverables and minimal risk of scope creep. By following this approach, the project can maintain consistency and traceability throughout the development lifecycle.

CHAPTER IV
PROBLEM RESOLUTION

This chapter presents the analysis and design stages of the student independent competition dashboard system integrated with artificial intelligence at the Faculty of Industrial Engineering, Telkom University. The process of system analysis and design will be outlined according to the Waterfall methodology. The discussion includes the stages of requirements analysis, architecture design, process design, and user interface design. Each phase is described comprehensively to provide a complete overview of the analytical processes and design decisions made during development.

IV.1 Requirement Analysis

The purpose of this analysis is to identify the weaknesses and limitations of the current dashboard system used in the FRI environment. This analysis serves as the basis for justifying the development of a new system. The current system in use is a Looker Studio-based dashboard with limited functionality.

IV.1.1 Current System Condition

The Faculty of Industrial Engineering is currently managing student competition data using a simple visualization dashboard through Looker Studio. Through observations and interviews with the Faculty of Industrial Engineering lecturers, several critical limitations in the existing system were identified.

First, the current dashboard only presents data in the form of basic tables and graphs, making it difficult for management to quickly and intuitively understand trends and performance patterns. Second, the dashboard lacks integration with AI technology that can provide deep insights or interactive recommendations. This forces users to manually export data and use external tools for deeper analysis. Third, there is no automatic mechanism for data updates, so the displayed information is often not real-time.

These limitations indicate that the current system is not yet adequate to support strategic decision-making and effective student development, as shown in Figure IV-1.

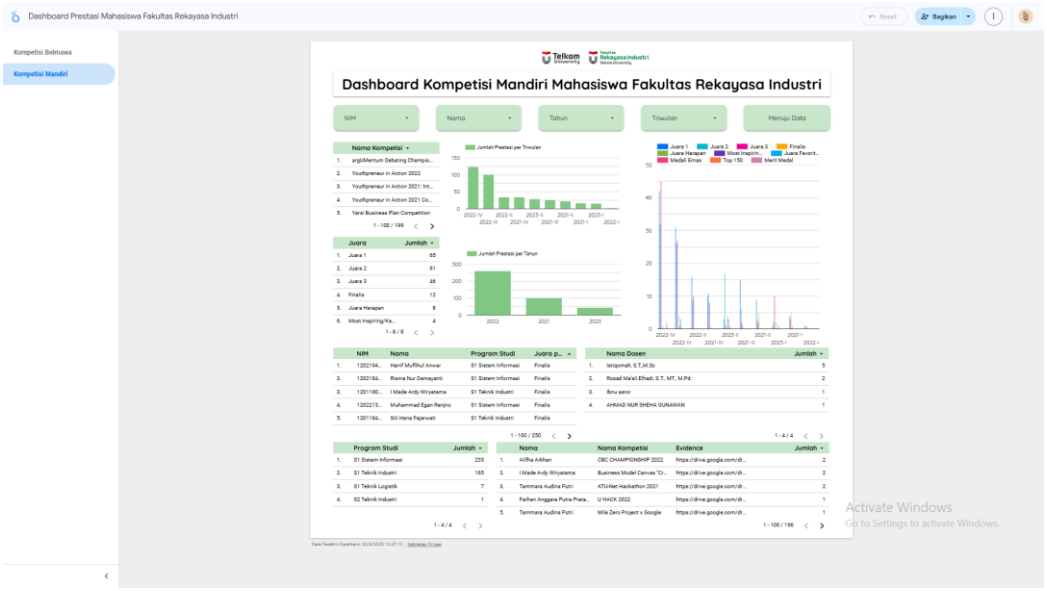

Figure IV- 1. Existing dashboard competition

As depicted in Figure IV-1, the current system uses a simple dashboard with data displayed in the form of regular tables and some basic graphs that only meet the very basic data presentation needs. The dashboard interface only provides basic functionality for data search, without any in-depth analysis features or adequate visualization. Moreover, the lack of integration with AI technology prevents the system from providing deeper insights or strategic recommendations, thereby hindering effective decision-making and targeted student guidance. Thus, the existing system is currently suboptimal in supporting the increasingly complex needs of managing and analyzing student competition data.

IV.1.2 Stakeholder Identification & Analysis

Stakeholder analysis reveals several user groups with different needs and expectations regarding the competition monitoring system. The main stakeholders include faculty management, student affairs staff, academic advisors, and the students themselves. Each group has specific needs that must be accommodated in the design of the new system, as shown in Table IV-1

Table IV- 1. Stakeholder analysis matrix

Stakeholders	Actors	Basic Needs	Expected Benefits
Dean FRI	Decision Maker	Strategic dashboard, performance analysis report, data-driven insights	Data-driven decision-making for improving student performance
Student Affairs Staff	Operational Management	Ease of data entry, automatic dashboard, tracking student progress	Ease in data management and monitoring student performance
Supervisor Professor	Academic Mentor	Guidance student achievement data, development recommendations	More effective monitoring and guidance for students
Student	Competition Participants	Latest competition information, personal achievement history,	Self-monitoring and motivation

Table IV-1 shows that each stakeholder has different needs and expectations regarding the system. This serves as the basis for determining the features and functionalities that will be developed.

From the interview with the Dean of FRI, it was obtained that currently the process of monitoring student achievements is still done manually and takes a considerable amount of time to produce analysis reports. The Dean hopes for a system that can provide strategic insights to support decision-making.

The student affairs staff conveyed that they need a system that can facilitate the data input process and provide clear visualizations of student achievement progress. Meanwhile, students hope to easily access information about their competitions and achievements.

IV.1.3 Functional Requirements Analysis

Based on stakeholder analysis, clear business objectives have been established to guide system development. The main objective is to transform the competition monitoring process from a reactive and manual approach into a strategic system that provides actionable insights and strategic recommendations. This transformation aligns with the broader faculty digital transformation initiatives and the commitment to data-driven decision-making.

Specific business objectives include enabling natural language queries for non-technical users, providing insights and recommendations based on historical data patterns, and supporting evidence-based strategies to improve student competition participation and success rates. These goals directly address the pain points identified in the current situation analysis and the needs of stakeholders, as shown in Table IV-2.

Table IV- 2. Functional requirements analysis

Code	Functional Requirements	Description	Priority	Stakeholders
FR-01	Authentication Management	The system must be able to perform user login and logout.	High	All
FR-02	Dashboard Visualization	The system must display a dashboard with achievement graphs and charts.	High	Dean, Staff
FR-03	AI Chatbot	The system must have a chatbot to provide insights and answer questions.	High	Dean, Staff
FR-04	Competition Data Management	The system must be able to manage student competition data.	High	Staff

Based on functional requirements in Table IV.2 are organized based on the priority and importance level of each feature. "High" priority is given to the most important features that must be included in the system, such as authentication, visualization dashboard, and AI chatbot.

The AI chatbot feature has become one of the main functional requirements that distinguishes this system from a regular dashboard. The chatbot is expected to understand questions in Indonesian and provide relevant answers based on the available competition data.

Based on the identified functional requirements, system modeling needs to be conducted to illustrate how each feature will interact with the system users. This modeling is done using a Use Case Diagram that can visualize the relationship between system actors and the various functions available. The Use Case Diagram

is used to illustrate the interaction between actors (system users) and the various functions available in the FRI Independent Competition Dashboard system. This diagram identifies two main types of actors, namely Student and Admin, each with different access rights to the system. Students have limited access to view their personal competition data, while Admins have full access to manage all competition data, access the analytics dashboard, and interact with the AI chatbot feature. All features in the system require a login authentication process as a security requirement, as shown in Figure IV-2

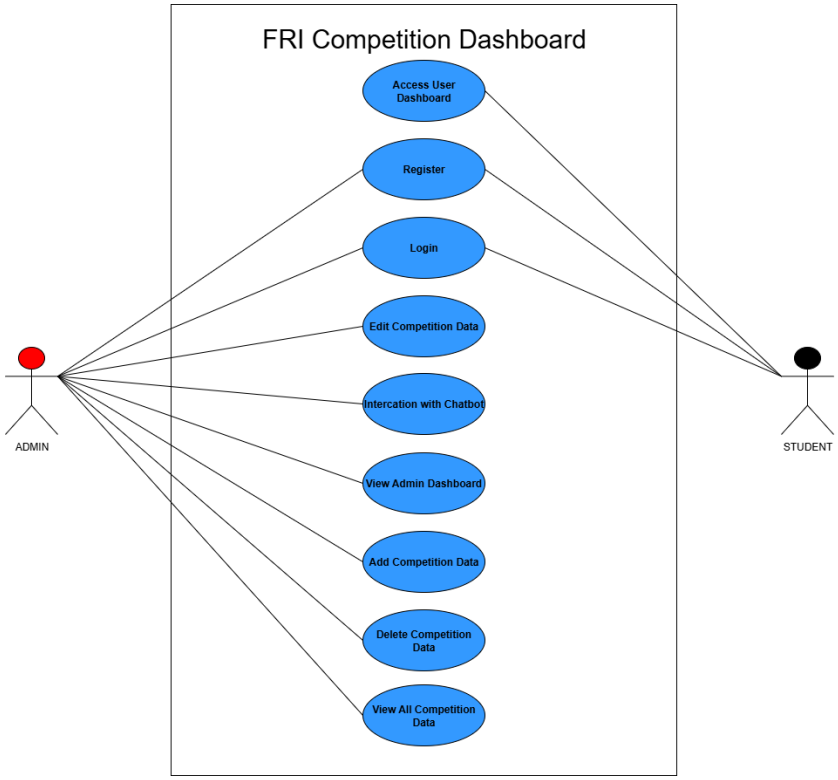

Figure IV- 2. Use case diagram

Based on the Use Case Diagram in Figure IV.2, the system consists of 9 main use cases divided into authentication functions and operational functions. The authentication functions include Register and Login, which can be accessed by both types of actors. Students have special access to the "Access User Dashboard" to view and validate their personal competition data. Meanwhile, Admins have access to six main functions: "View Admin Dashboard" to see comprehensive analytics, "Interaction with Chatbot" to gain AI-based insights, "Adding Competition Data" to add new data, "Edit Competition Data" to modify existing data, "Delete

Competition Data" to remove data, and "View All Competition Data" to see the entire competition database.

IV.1.4 Non-Functional Requirements Analysis

Non-functional requirement analysis is conducted to identify the quality aspects of the system that must be met, such as performance, security, and usability. These non-functional requirements are important to ensure the system can operate well in a production environment. as shown in Table IV-3.

Table IV- 3. Analysis of non-functional requirements

Code	Aspect	Needs	Metrics	Target
NFR-01	Performance	Response time system	Response Time	< 10 Second
NFR-02	Performance	User capacity	Concurrent users	50 simultaneous users
NFR-03	Usability	Ease of use	SUS Score	> 70
NFR-04	Security	Data Security	Encryption	SSL
NFR-05	Compatibility	Browser compatibility	Browser support	Chrome, Firefox, Safari
NFR-06	Reliability	System availability	Uptime	99%
NFR-07	Scalability	Ability to develop	Data growth	Support 10.000 records

Based on Table IV-3 shows various non-functional aspects that must be considered in system development. Each aspect has clear metrics and targets to facilitate the evaluation of the system after it is developed.

The performance aspect is the top priority considering the system will handle data visualization and AI processing, which require substantial computational resources. A target response time of under 10 seconds is set to ensure a good user experience.

Data security is also an important aspect considering the system will handle sensitive student data and achievements. The implementation of SSL encryption is necessary to protect data during transmission. In addition, user authentication and role-based access control help prevent unauthorized access and ensure that only authorized personnel can manage or modify competition records.

IV.1.5 Data Requirements Analysis

Data needs analysis is conducted to identify the types of data required by the system, the data sources, and how the data will be managed. This analysis is important for determining the database structure and data flow within the system. as shown in Table IV-4

Table IV- 4. Data needs analysis

Data Category	Data Type	Source	Format	Explanation
Student Data	NIM, Name, Study Program	DITMAWA	Excel	Master data of students
Competition Data	Competition name, Level, Category	DITMAWA	Excel	Competition data available
Achievement Data	Champion, Year, Semester	DITMAWA	Excel	Student achievement data
Mentor Data	NIDN, Lecturer's Name	DITMAWA	Excel	Data of the supervising lecturer
User Data	Username, Password, Role	System	Database	System access data

Based on the analysis conducted, the system will manage five main categories of data based on Table IV-4. Student data, competition data, and achievement data are the main data sourced from DITMAWA in Excel format.

IV.1.5.1 Data Collection and Sources

At the data collection stage, information regarding the independent competitions of FRI students was obtained from the Directorate of Student Affairs (DITMAWA) of Telkom University. The data was provided in the form of an Excel file titled "Prestasi Kompetisi - RAW.xlsx," which is a compilation of data from various sources. This file is the result of extraction from the complex and not yet fully structured SIMKATMAWA (Student Activity Information System) system, covering the period from 2021 to 2024. Overall, the data consists of 18 worksheets with a total of 6642 competition records that are still scattered but remain within the realm of Telkom University, as shown in Figure IV-3. This raw data requires further processing to remove inconsistencies and ensure it matches the structure of

the new system. Proper cleaning and transformation are essential before the data can be loaded into the competition dashboard.

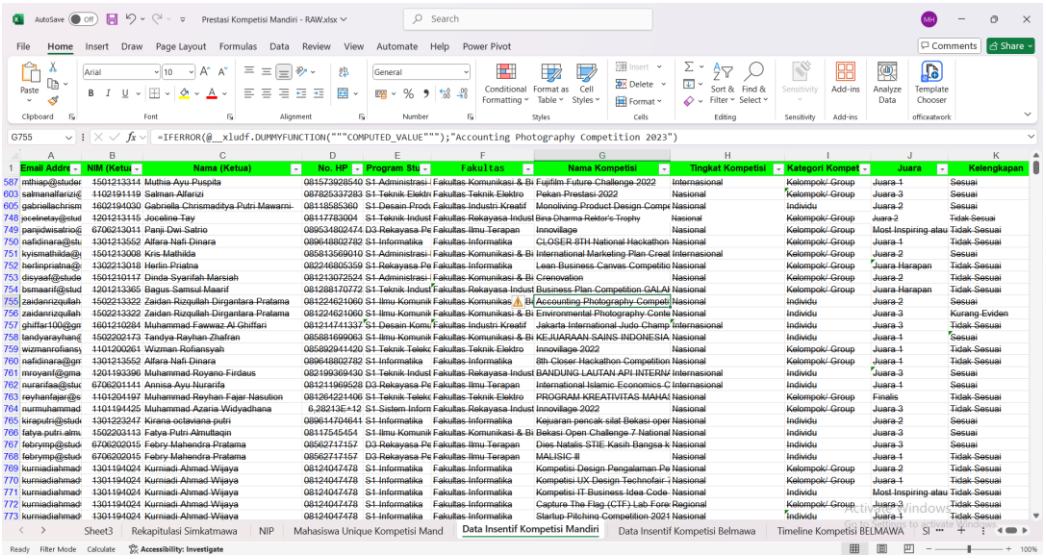

Figure IV- 3. Architecture of data sources from DITMAWA

Based on Figure IV-3 illustrates the complexity of the data structure received from DITMAWA. The raw Excel file contains 18 different worksheets, which include:

- Data Incentive for Mandiri Competitions (Sheet5)
- Data Incentive for Belmawa Competitions (Sheet6)
- BELMAWA Competition Timeline (Sheet7)
- BELMAWA Award (Sheet9)
- Independent Award (Sheet11)
- ORMAWA Achievements (Sheet15)
- Various pivot tables and summary sheets

The research data was obtained from DITMAWA in the form of a raw Excel file with a complex structure containing 18 different worksheets, including independent competition data, BELMAWA competition data, incentive data, competition timelines, and various pivot tables as shown in Figure IV.3. This data fragmentation creates specific challenges in the process of extracting and integrating relevant data for the research. To obtain the independent competition data that is the focus of the study, a special extraction was carried out from the relevant worksheets, particularly Sheet5 "Mandiri Award" and Sheet11 "Data Insentif Kompetisi Mandiri." This

extraction process successfully produced an initial dataset with a total of 649 independent competition records for the period 2021-2024, consisting of 26 attribute columns, but still requires further cleaning and preprocessing before it can be used in analysis.

IV.1.5.2 Data Cleaning and Preparation Process

Based on the data obtained from DITMAWA, a data cleaning process was carried out to ensure the quality and consistency of the data before it is used in the system. This process was conducted using Python in Google Colaboratory with a systematic pipeline.

Figure IV- 4. Data cleanning pipeline

Based on Figure IV-4 shows the data cleaning process, which consists of several systematic stages from raw data to a dataset ready for analysis.

Stage 1: Data Extraction The data extraction stage is carried out to separate the independent competition data from the complex Excel file containing 18 worksheets with mixed data:

- Input: Raw Excel File from DITMAWA with 18 worksheets containing mixed data from various faculties and types of competitions
- Process: Identify worksheets containing independent competition data and filter data specifically for FRI students
- Output: Initial dataset with 26 columns and 649 records of independent FRI competitions

Table IV- 5. Identified data quality issues

Quality Dimension	Problem Type	Affected Column	Affected Record	Impact Level
Completeness	High Missing values	NIDN, NIP, Lecturer Name	~90%	High
	High Missing values	ORMAWA	~85%	Medium
Consistency	Inconsistency capitalization	Competition Category	1 (0.2%)	Low
	Anomalous study program	Study Program (“z”)	1 (0.2%)	Medium
Accuration	Duplication of data	Combination NIM-Competition-Year	26 (4.0%)	High
Redundancy	Multiple Champion columns	Champion, Group Champion, Champion per Faculty	All	Medium
Validity	Irrelevant column	Simkatmawa Status, Competition Division	All	Low

as shown in Table IV-5, it reveals several significant challenges related to data quality in the extracted dataset consisting of 26 columns. The most crucial issue was found in the supervisor columns, namely NIDN, NIP, and Lecturer Name, which have a very high rate of missing values (around 90%). This indicates that the recording system has not been consistent in documenting competition supervisor information. Nevertheless, these columns are retained to allow for further analysis in the future regarding faculty involvement.

There is redundancy in the achievement section, with the presence of several columns such as Champion, Champion per Member, Champion per Faculty, and Group Champion, which could potentially cause inconsistencies. The analysis results show that the "Juara per Anggota" column is the most complete and reliable for analysis.

Further analysis of the competition levels shows relatively better data quality, with only 1 record (0.2%) lacking level specifications. The distribution of competition levels includes 423 National level competitions (65.2%), 162 International (25.0%), 38 Regional (5.9%), 23 Local (3.5%), and 1 competition each at the Local (0.2%) and Provincial (0.2%) levels. However, there are variations in naming such as Wilayah and Regional that refer to similar entities, necessitating a standardization process during the data cleaning stage.

Table IV- 6. Result data cleaning process

Stage	Input	Output	Explanation
Raw Data	18 worksheets, mixed data	1 consolidated file	Mandiri competition extraction FRI
Initial Dataset	26 columns, 649 records	Structured data	Consolidation of competition data
Column Optimization	26 columns	21 columns	Elimination of redundancy
Data Enhancement	Basic attributes	Enhanced attributes	Addition of derived fields
Clean Dataset	Inconsistent values	Standardized values	Dataset ready for use

As shown in Table IV-6 the data cleaning process begins with the extraction of independent competition data from a complex Excel file that has 18 worksheets. From these various worksheets, extraction and consolidation specifically for independent competition data are carried out, resulting in an initial dataset with 26 columns. This extracted dataset then undergoes an in-depth cleaning process to produce a final, more structured dataset with 21 columns.

From the data analysis obtained from DITMAWA, there are a total of 649 records of independent competition data for the period 2021-2024 that will become the main data in the system. This data includes students from four study programs at FRI: S1 Information Systems, S1 Industrial Engineering, S1 Logistics Engineering, and S2 Industrial Engineering.

The data structure shows that there are several attributes with missing values, especially in the advisor data (NIDN and lecturer names). However, this data is retained to maintain historical integrity and allow for more comprehensive analysis in the future.

Analysis also shows inconsistencies in naming and data formatting that need to be cleaned and standardized before the data can be used in the system. This cleaned dataset serves as the foundation for the system implementation phase, particularly for migration to the database and use in visualization dashboards as well as AI chatbots.

IV.2 System Design

The data understanding phase involves the exploration and comprehensive assessment of available competition data to identify patterns, quality issues, and analytical opportunities. This phase is essential for making informed decisions regarding data preparation strategies and analytical approaches.

IV.3 Architecture Design

The system architecture design uses a layered architecture approach to separate system functions and facilitate development and maintenance. The designed architecture consists of four main layers that interact with each other to produce a comprehensive system, as shown in Figure IV-5

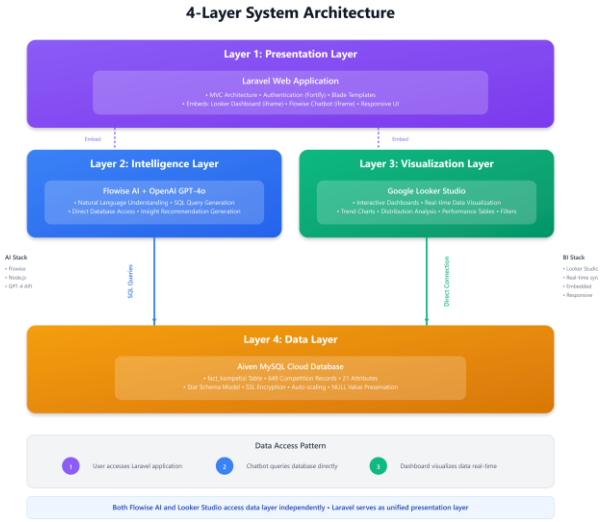

Figure IV- 5. Integrated system architecture

Based on Figure IV-5 presents the implementation of a layered architecture adapted for an AI-integrated dashboard system, encompassing data visualization, artificial intelligence, and a web-based user interface within a student competition monitoring system. The four layers that make up this architecture are: Data Layer, Intelligence Layer, Visualization Layer, and Presentation Layer.

1. Layer 4 Data Layer: At the lowest layer, the system uses Aiven MySQL Cloud Database as the main data storage center. The tables used have a star schema with 649 competition record data entries and 21 attributes, allowing for high flexibility in the analysis process. This database is also supported by auto-scaling features, SSL encryption, and the ability to maintain NULL values to ensure data accuracy. This layer serves as the primary data source for the two main components in the upper layer: Looker Studio and Flowise AI.
2. Layer 3 Visualization Layer: The visualization layer utilizes Google Looker Studio to present data in real-time in the form of interactive dashboards, trend graphs, participation distribution, performance tables, and dynamic filters. Looker Studio is directly connected to the database without going through the application layer, allowing for instant visualization updates when the data changes. This layer plays a crucial role in providing informative visual representations for non-technical users such as lecturers and students.
3. Layer 2 Intelligence Layer: In this layer, the system integrates Flowise AI, which uses the OpenAI GPT-4o language model. This AI is responsible for translating users' natural language questions into SQL queries, retrieving data directly from the database, and compiling answers in the form of insights or strategic recommendations. Features such as Natural Language Understanding and AI-based analytical capabilities enable the system to automatically and deeply respond to complex questions. This layer is fully independent and can be horizontally scaled to accommodate spikes in computational load.
4. Layer 1 Presentation Layer: The top layer is a Laravel-based web application, which serves as the unified presentation layer. Within it, the MVC Architecture pattern is implemented, along with the authentication system from Laravel Fortify, and a responsive UI that supports users from various devices. In this

layer, the Looker dashboard and the Flowise chatbot are embedded using an iframe, allowing users to access both systems simultaneously through a single integrated platform.

Interaction between layers is conducted through standardized protocols and APIs, maintaining loose coupling between components and enabling continuous development (modular enhancement). This architecture also adopts an independent data access pattern: Flowise AI and Looker Studio have direct connections to the database without the Laravel intermediary. This provides capabilities and scalability, as well as enabling synchronous (real-time query) and asynchronous (advanced analysis) processes to run in parallel.

With this design, the system is capable of supporting user growth, enhancing AI service performance, and ensuring that analytical and reporting needs can be met effectively without interfering with each other.

IV.3.1 User Interfaces Design

The design of the user interface is carried out through a phased approach, starting from Low Fidelity to High Fidelity. The design process follows the principles of user-centered design by involving feedback from stakeholders to ensure that the resulting interface meets user needs.

IV.3.1.1 Low Fidelity Design (Wireframe)

The initial stage of interface design begins with the creation of wireframes to determine the basic structure and layout of the system. Wireframes are created for the main pages of the system based on the needs analysis that has been conducted previously. The wireframing process aims to map the hierarchy of information, determine the positions of important elements before moving on to the more detailed visual design stage. Low Fidelity focuses on functionality and layout structure without being distracted by visual elements such as color and typography, as shown in Figure IV-6, Figure IV-7, and Figure IV-8. This stage helps designers and stakeholders reach an early agreement on the overall page structure. Clear wireframes also make it easier to identify potential usability issues before investing

time in high-fidelity design. By starting with wireframes, the development team can ensure that the final interface will meet user needs effectively.

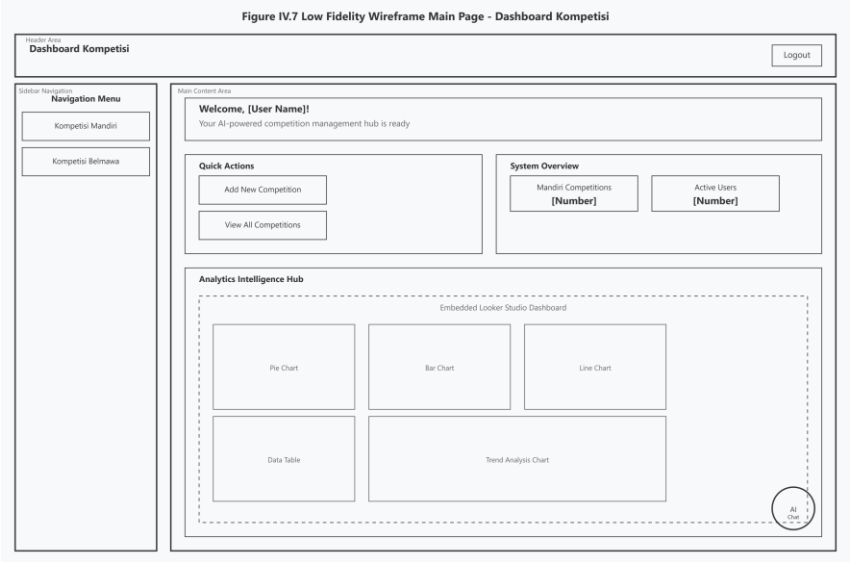

Figure IV- 6. Low fidelity wireframe main page

Based on Figure IV-6 the wireframe of the main page is designed with consideration for information hierarchy and ease of navigation. The area is divided as follows: The header area located at the top of the page serves for branding the "Competition Dashboard" system and provides the main navigation, including a logout button on the right side. The sidebar on the left provides the main navigation menu to switch between competitions, with buttons leading to each type of competition (Independent Competition and Belmawa Competition).

The main content area occupies the largest portion of the layout and is divided into several functional sections. The welcome section provides personalization by greeting users and displaying the system status. The Quick Actions section provides quick access to important functions such as "Add New Competition" and "View All Competitions". System Overview displays important metrics in an easy-to-understand card format, such as the total number of independent competition records and registered users.

Analytics Intelligence Hub is the main component that displays an embedded Looker Studio dashboard with various data visualizations. This area is designed

with placeholders for various types of charts such as pie charts for category distribution, bar charts for data comparison, line charts for trend analysis, data tables for detailed information, and trend analysis charts for temporal analysis. The AI chatbot pop-up is placed in the bottom right corner as a floating action button in the form of a circular icon, providing easy access to the AI assistant features without disrupting the main dashboard view.

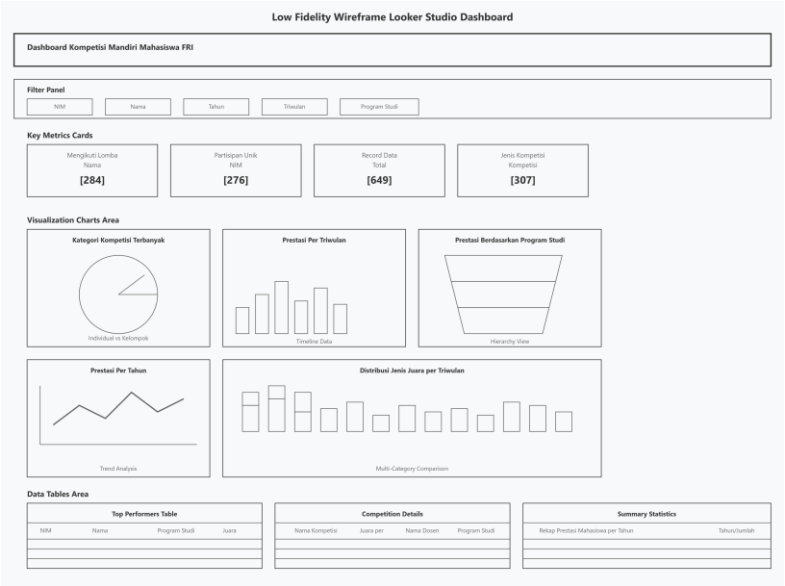

Figure IV- 7. Low fidelity wireframe looker studio dashboard

Based on Figure IV-7 the wireframe Looker Studio dashboard focuses on the placement of data visualization elements with a systematic structure. The cards at the top serve to display important metrics such as "Participating in Competitions" (284 names), "Unique Participants" (276 student IDs), "Data Records" (649 total), and "Types of Competitions" (307 competitions) that provide a quick overview of the performance of FRI student competitions. The chart area is designed to display various types of trend and distribution visualizations in a flexible grid layout. This area consists of a pie chart for "Most Common Competition Categories" showing individual vs group distribution, a bar chart for "Quarterly Achievements" visualizing the data timeline, a funnel chart for "Achievements by Study Program" in a hierarchy view, a line chart for "Annual Achievements" as trend analysis, and a stacked bar chart for "Quarterly Champion Distribution" providing multi-category comparison.

The table area is placed at the bottom to present detailed data in an easily readable and sortable format. This area is divided into three main tables: the "Top Performers Table" which displays Student ID, Name, Study Program, and Champion; the "Competition Details" which contains Competition Name, Champion by category, Lecturer Name, and Study Program; and the "Summary Statistics" for "Student Achievement Recap per Year" in Year/Count format. The filter panel is strategically positioned at the top to give users control over the displayed data, with filtering options based on NIM, Name, Year, Quarter, and Study Program that allow for drill-down analysis according to the user's specific needs.

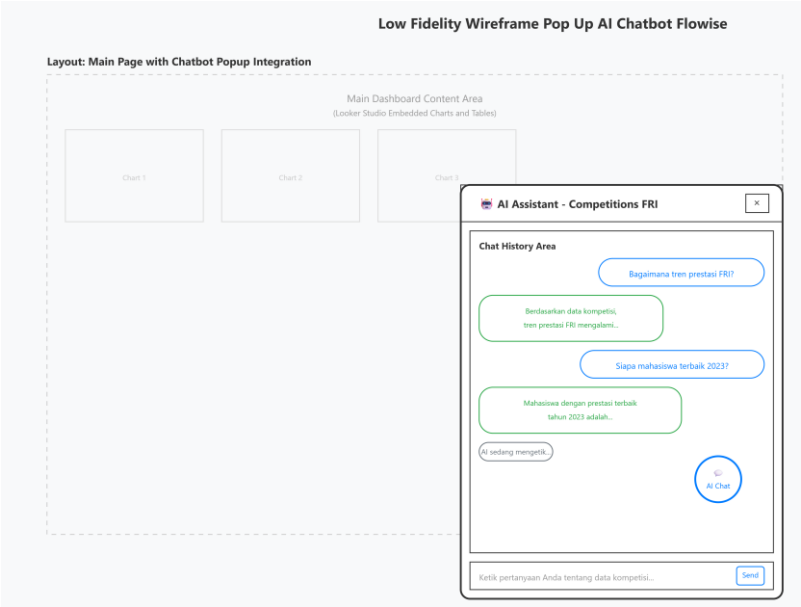

Figure IV- 8. Low fidelity wireframe chatbot page

Based on Figure IV-8 the low-fidelity wireframe for the chatbot page shows a simple yet functional popup interface design with a header reading "AI Assistant - Competitions FRI" at the top, equipped with a close (X) button in the top right corner. The chat history area occupies the middle section of the popup, displaying the conversation between the user and the AI with different chat bubbles to distinguish the message sender, such as the example question "What are the trends in FRI's achievements?" and the AI's response providing insights about competition data.

The bottom part of the popup contains an input area with placeholder text inviting users to type their questions, accompanied by a "Send" button on the right to send the message. This wireframe design shows a minimalist approach with a focus on functionality and a clean user experience, where each element has a clear purpose in supporting the conversational interaction between the user and the AI assistant.

The vertical layout structure of this wireframe allows for scrolling through the chat history when the conversation becomes lengthy, while the input area remains fixed at the bottom for easy access. The overall design of this popup is intended to provide a familiar chat experience for users while remaining integrated with the context of the FRI independent competition dashboard.

IV.3.1.2 User Flow Design

The design of the user flow aims to map the user's journey in using the independent competition dashboard system from the starting point to achieving their goals. This flow design considers two main user categories, namely administrators and students, each with different access levels and features according to their needs and responsibilities. User flow is designed to provide an intuitive experience while maintaining data security through the implementation of strict role-based access control, as shown in Figure IV-9.

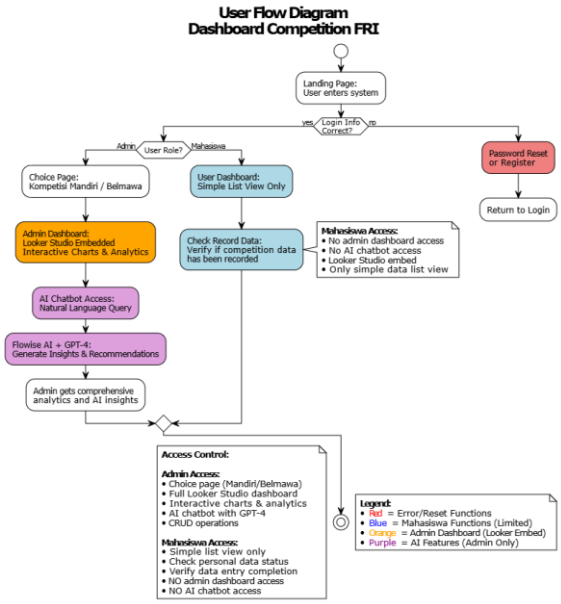

Figure IV- 9. User flow diagram

Based on Figure IV-9 the user flow starts from the landing page where users enter their login credentials. The system then verifies the entered login information. If the credentials are invalid, the user will be redirected to the main login page. Conversely, if the credentials are valid, the system will identify the user's role to determine the appropriate access rights.

For users with the admin role, the system provides full access to all dashboard features. Admins are first directed to the choice page where they can choose between "Independent Competition" or "Belmawa Competition" according to the desired scope of analysis. After the selection, the admin accesses the main dashboard that integrates Looker Studio with various interactive visualizations such as charts and analytics. The main advantage for the admin is access to an AI chatbot that allows them to perform queries using natural language. The chatbot built with Flowise AI and integrated with GPT-4o is capable of generating deep insights and recommendations based on the available competition data.

Unlike the admin, students have limited but still functional access for their needs. After successfully logging in, students are immediately directed to the user dashboard, which displays data in a simple list format. The main features available to students are the ability to check the status of their competition data, verify whether the competition data they participated in is recorded in the system, and monitor their personal achievements. This access control is implemented to maintain data integrity while providing sufficient transparency for students to monitor their own status.

The access control implemented shows a clear separation between the two user roles. Admins have full access, including the choice page, Looker Studio admin dashboard, interactive charts & analytics, AI chatbot with GPT-4, and CRUD operations. Meanwhile, students can only access a simple list view, verify data entry completion, without access to the admin dashboard or AI chatbot. This access distribution ensures that each user receives features that align with their needs and responsibilities, while maintaining the overall security and integrity of the system.

IV.3.1.3 High Fidelity Design

After the wireframe is validated, a mockup is created with complete visual details. This mockup includes color selection, typography, and other visual elements as shown in Figure IV-10.

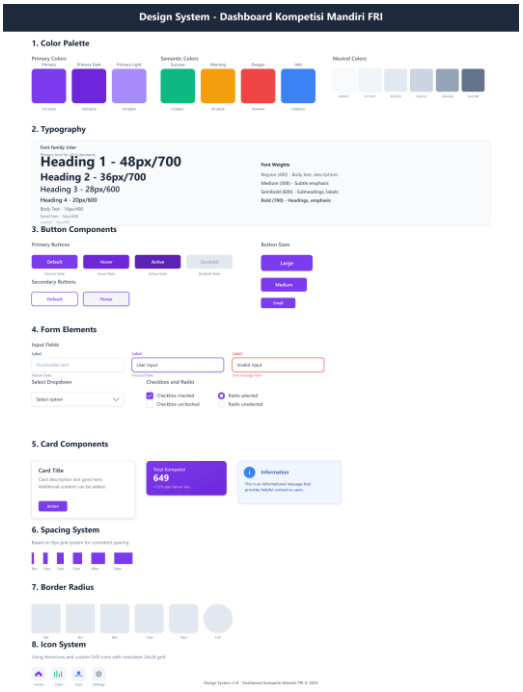

Figure IV- 10. Design system component

Based on Figure IV-10 The design system being developed includes:

Color Palette:

- Primary: #6366F1 (Indigo) for main elements
- Secondary: #10B981 (Emerald) for positive indicators
- Warning: #F59E0B (Amber) for warnings
- Danger: #EF4444 (Red) for negative indicators
- Neutral: Gray gradient for text and background

Typography:

- Heading: Inter font family, weight 600-700
- Body: Inter font family, weight 400-500
- Code: JetBrains Mono for technical elements

Component Library:

- Buttons with various states (default, hover, active, disabled)
- Form elements (input, select, checkbox, radio)
- Cards for information container
- Navigation elements

The high-fidelity (Hi-Fi) design created in this study serves as a visual and functional reference in the system development process. This prototype was independently created by the researcher as an initial representation of the page structure, navigation, and interface interactions that will be implemented in the web application. By referring to this prototype, the development process can be carried out in a more directed, consistent, and efficient manner, without requiring additional user feedback cycles before the implementation stage begins.

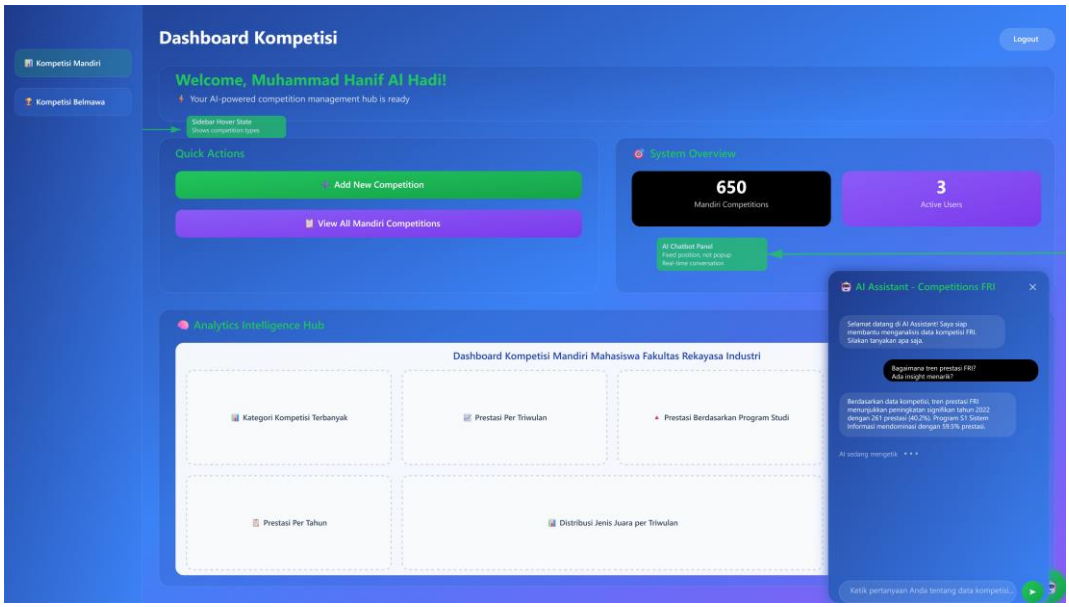

Figure IV- 11. High-fidelity dashboard display for mandiri competition

Based on the high-fidelity (Hi-Fi) dashboard display in Figure IV.11 represents the visual implementation of the previously designed system. This dashboard shows the integration of various main components, such as quick action buttons, system statistics panels, and interactive data visualizations sourced from Google Looker Studio. This design is based on the principles of an intuitive, modern, and responsive interface, and is structured to facilitate users from various roles,

including deans, student affairs staff, and students, in accessing real-time competition performance information.

The main section of the dashboard displays the number of independent competitions, the number of active users, and analytical graphs that summarize student achievement trends based on time, study program, and competition category. The Looker Studio panel in the middle presents dynamic and customizable data based on filters such as quarter, study program, and year.

In addition to the main dashboard elements, this system is also equipped with an AI chatbot based on Flowise and OpenAI API that appears as an interactive pop-up in the bottom right corner. This chatbot functions as a virtual assistant that allows users to ask questions and obtain data-based insights directly, supporting a more interactive and adaptive user experience.

IV.3.1.4 Activity Diagram

- Activity Diagram #1: Regisration & Role Assignment

The registration system in the independent competition dashboard platform implements a dual-path mechanism that differentiates the registration process based on the required access level. This process begins with the selection of the type of registration that determines the subsequent validation flow, where regular user registration uses standard data validation, while administrator registration requires an additional layer of security through secret code verification. The implementation of the position field as an indicator of role assignment ensures that each registered account has the appropriate access level according to its function and responsibilities within the student achievement monitoring ecosystem, as shown in Figure IV-12. This role-based approach minimizes the risk of unauthorized access and misuse of administrative privileges. By integrating this structured registration flow, the platform enhances the integrity and confidentiality of the data it manages, while ensuring accountability for each user's actions. Overall, this mechanism creates a secure and organized environment for monitoring and analyzing student competition achievements.

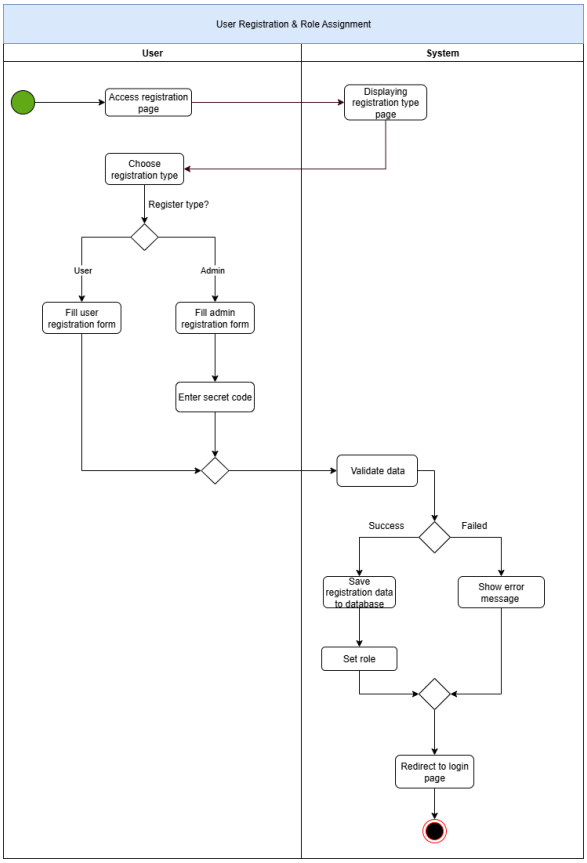

Figure IV- 12. User registration & role assignment

Based on Figure IV-12 this layered registration architecture successfully creates a secure yet user-friendly system, where comprehensive error handling allows users to correct input errors without having to repeat the entire process. Automatic role assignment based on the selected registration type ensures clear function separation, while automatic redirection to the login page after successful registration provides a seamless user experience and reduces barriers to system adoption.

- Activity Diagram #2 (Authentication & Role-based Redirect)

The authentication mechanism in this system applies automatic user role checks by utilizing the position column in the account data. The system will detect whether the user account is registered as a regular user (if the position column is empty) or as an administrator (if the position column is filled). The login process not only checks the username and password but also matches the selected login type with the actual role. This is done to prevent abuse of access rights and ensure that each

user only enters the page corresponding to their access rights. The page redirection after login is also arranged to match the user's role, as shown in Figure IV-13

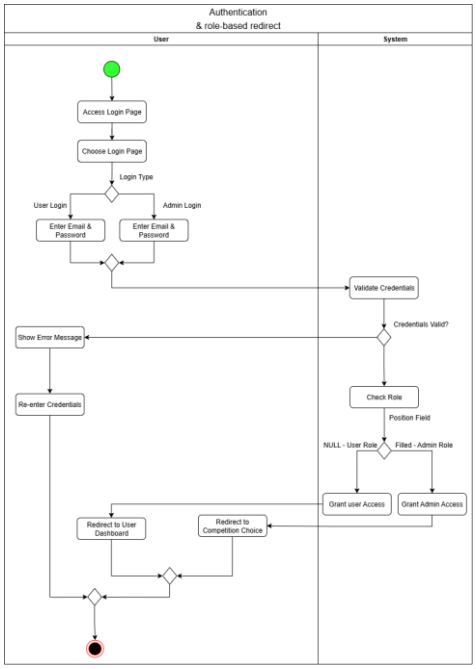

Figure IV- 13. Authentication & role-based redirect

Based on Figure IV-13 The authentication system with role checks helps maintain the security of data access within the system. Each stage of login is accompanied by additional verification steps to ensure that no user can access areas they are not supposed to. After the login process is successful, users will be directed to the appropriate page, regular users to the main dashboard page with limited menus, while administrators will be directed to a special page to manage competition data. In this way, access distribution can proceed smoothly according to the tasks and responsibilities of each user.

- Activity Diagram #3 (User Dashboard Access)

The dashboard for regular users is designed only to display relevant information without opening access to the data management menu, which is only accessible by admins. The main dashboard view uses Looker Studio in a list view format so that competition data can be clearly seen and easily understood. This feature makes it easier for users to monitor data, but it does not provide access to advanced analytical tools that are only used by the management. The system is also equipped with

access restriction settings that will immediately block regular users if they attempt to enter the admin-only page, and will automatically display an access denial message, as shown in Figure IV-14.

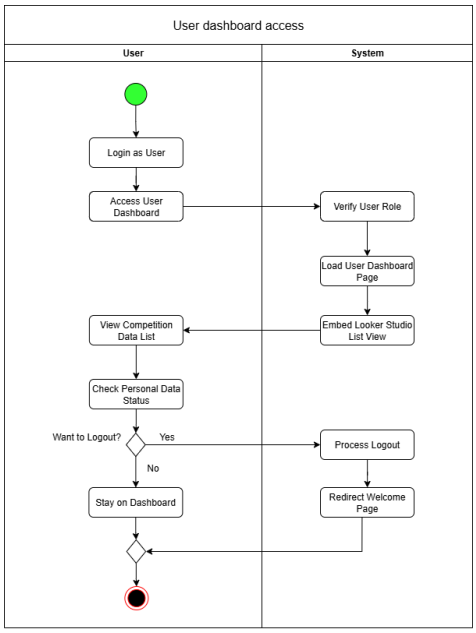

Figure IV- 14. User dashboard access

Based on Figure IV-14 the implementation of this user dashboard helps maintain a balance between ease of information access and data security. Users can view their performance data comfortably, without worrying that the main system data will be altered without permission. Error handling is also made clear so that users understand if they try to access an inappropriate page. A logout mechanism is also provided so that users can end their sessions safely and comfortably according to the needs of the academic environment.

- Activity Diagram #4 (Admin Dashboard Access)

The dashboard for admins serves as a central hub for comprehensive competition data management. Through this page, the admin can select the type of competition to be analyzed, such as Independent Competition or Belmawa Competition. Each choice will display different dashboard settings according to the competition category and related data. Data visualization is provided through Looker Studio, which displays graphs and diagrams to make it easier for the admin to understand the data. Admins can also utilize the AI chatbot feature already installed on the

dashboard to assist in information retrieval or answer questions related to competition data interactively, as shown in Figure IV-15.

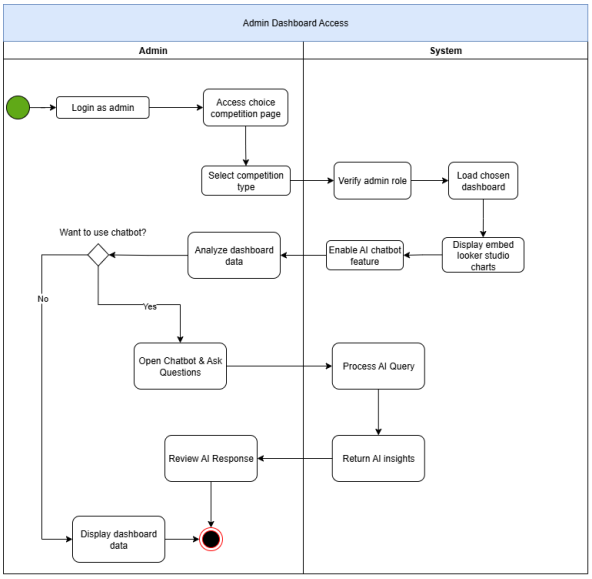

Figure IV- 15. Admin dashboard access

Based on Figure IV-15 this admin dashboard supports in-depth data analysis while also making it easier for admins to make decisions based on more comprehensive information. The combination of data visualization and AI chatbot assistance makes data management more practical. The navigation feature on the sidebar also makes it easier for admins to switch between different types of competitions quickly, ensuring that data management remains responsive and aligned with the needs of the academic environment.

- Activity Diagram #5 (Competition Type Selection)

The competition type selection feature through the sidebar menu is designed to make it easier for admins to switch between data categories quickly. This feature supports the faculty's need to view data from various types of competitions, such as Independent Competitions or Belmawa Competitions, without having to leave the dashboard. The menu display is designed to be easily accessible without disrupting ongoing tasks. Every time the admin selects a type of competition, the system will automatically update the dataset, refresh the Looker Studio view, update the dashboard graphs, and adjust the chatbot context to remain consistent with the selected data, as shown in Figure IV-16.

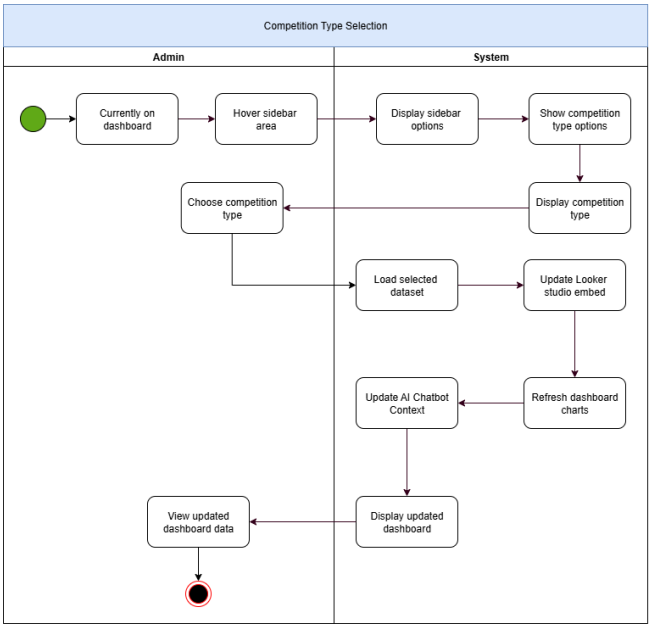

Figure IV- 16. Competition type selection

Based on Figure IV-16 the direct transition process between competition types helps the admin compare data between categories quickly and stay focused during the decision-making process. All components on the dashboard will be updated simultaneously to maintain data consistency and display accuracy. In this way, competition data management can support data-driven management needs more practically and efficiently.

- Activity Diagram #6 (AI Chatbot Interaction)

The AI Chatbot feature in this system is designed to appear as a pop-up panel in the corner of the page, so it does not disrupt the main dashboard view. Through this chatbot, admins can have conversations to analyze data without having to switch pages. The chatbot works by processing questions in natural language, then translating them into query commands that are executed on the Aiven MySQL database. In this way, the admin can obtain answers and additional information while still viewing the existing data graphs on the dashboard, as shown in Figure IV-17. This integrated approach helps streamline the data analysis process and saves time for the admin. By combining conversational AI with real-time data access, the system supports more interactive and efficient decision-making. As a result, the dashboard becomes not just a static display but an intelligent tool for exploring and understanding competition data.

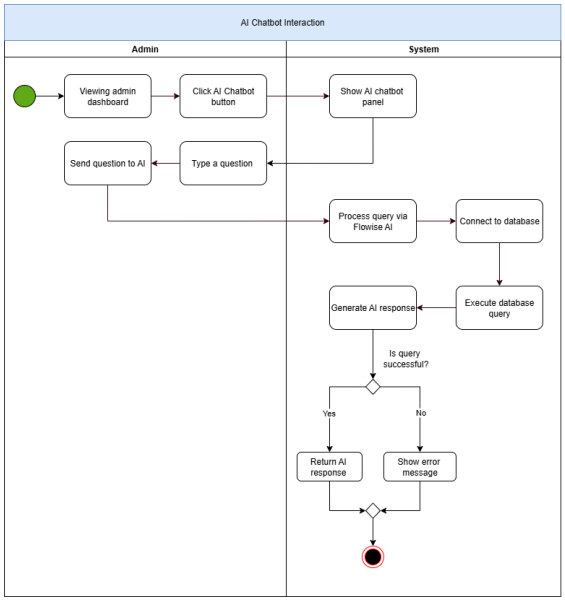

Figure IV- 17. AI chatbot interaction

Based on Figure IV-17 the presence of this chatbot helps the admin obtain additional information in a practical and quick manner through regular conversations. Users can run multiple queries while still monitoring data visualization on the screen. In this way, the data analysis process becomes easier to understand and supports more accurate decision-making based on information obtained directly.

- Activity Diagram #7 (CRUD Operations)

The CRUD (Create, Read, Update, Delete) process in the competition data management feature is designed with a structured workflow to maintain data accuracy and security. The Update feature directs the admin to a special edit page so that data changes can be made comprehensively with a form that is already equipped with validation. Meanwhile, the Delete feature is equipped with a confirmation dialog so that the admin can double-check the decision before deleting data, thus avoiding the risk of accidentally losing important data, as shown in Figure IV-18. This confirmation mechanism adds an extra layer of safety to prevent unwanted data loss. In addition, all changes made through the CRUD process are logged to ensure transparency and accountability. This approach helps the system maintain reliable data records that support effective monitoring and reporting.

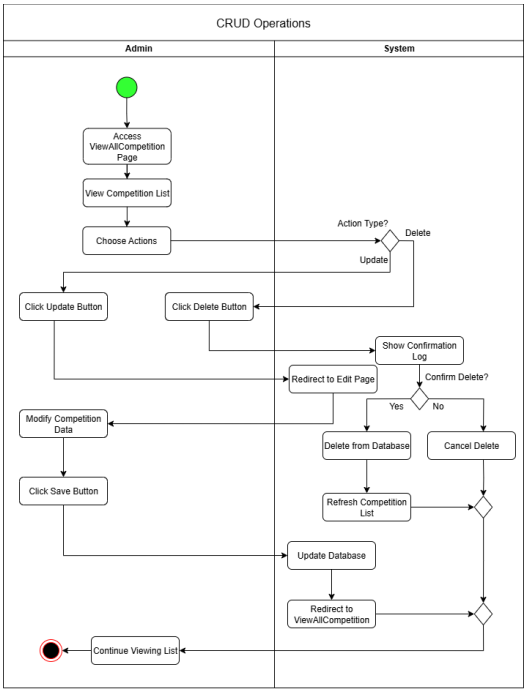

Figure IV- 18. CRUD operations

Based on Figure IV-18 the implementation of these CRUD operations gives the admin full control to add, update, view, and delete competition data as needed. Each successfully executed operation will automatically update the data list so that the information always appears in the latest condition. In this way, competition data management can be carried out more securely and support reporting and decision-making needs at the faculty level.

IV.3.2 Intergration Design

The design of system integration aims to ensure effective communication between all system components. The integration design uses a modular approach that allows each component to be developed and maintained independently, as shown in Figure IV-19. This modularity increases system flexibility and makes it easier to implement future updates or feature enhancements without disrupting the entire platform. Clear data exchange protocols are defined to ensure that information flows seamlessly and accurately between modules. As a result, the system can maintain high performance and reliability even as new components are added or existing ones are improved.

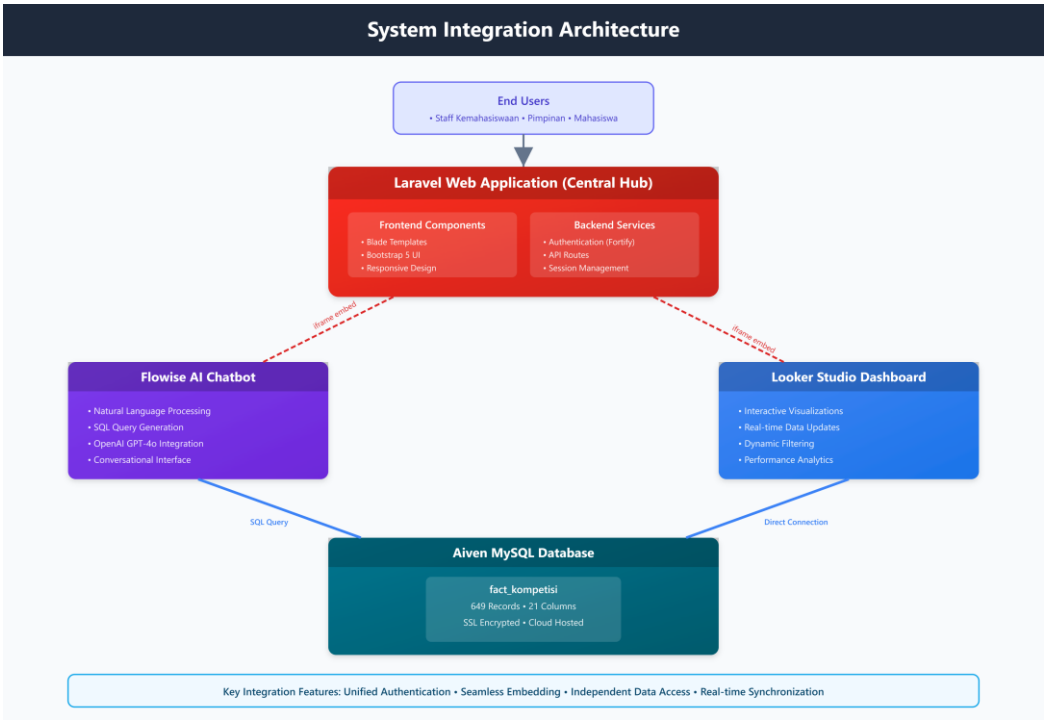

Figure IV- 19. System integration architecture

Based on Figure IV-19 that explain how system will be integrated each other and will be use layer by layer:

1. Laravel as the Central Hub Laravel functions as the main integration point that connects all components:
    - Frontend Components: Blade templates, Bootstrap UI, responsive design
    - Backend Services: Authentication, API routes, session management
2. The integration of the Flowise AI Chatbot is carried out through several mechanisms:
    - Natural Language Processing to understand user queries
    - Direct database connection to retrieve real-time data
    - OpenAI GPT-4o integration for natural language generation
    - Conversational interface embedded via iframe

    Flowise AI is configured to access the database directly, enabling optimal response time and always up-to-date data.

3. Integration of Looker Studio Dashboard The Looker Studio Dashboard is integrated in the following way:
    - Interactive visualizations that can be dynamically filtered
    - data updates from the Aiven MySQL database
    - Embedded dashboard via iframe
4. Database Integration Aiven MySQL Database functions as a single source of truth with characteristics:
    - SSL encrypted connections for data security
    - Cloud hosted for high availability and scalability
    - Automatic backup and disaster recovery
5. Key Integration Features Some important integration features in this system:
    - Unified Authentication: Single sign-on for access to all components based on role
    - Seamless Embedding: Dashboard and chatbot integrated into a single interface
    - Independent Data Access: Each component can access data directly
    - Real-time Synchronization: Consistent data across all layers

IV.3.2.1 Sequence "User View Dashboard"

The dashboard system is designed with multi-level access control that provides limited access to students for monitoring their own competition data. Unlike admins who have full CRUD capabilities, students are only given read-only access to maintain data integrity. The following sequence diagram illustrates the streamlined workflow of how students access the competition dashboard with an embedded Looker Studio tailored to their authorization level, as shown in Figure IV-20. This restriction minimizes the risk of unauthorized data modification and ensures that all competition records remain accurate and reliable. By implementing this controlled access, the system promotes accountability while still empowering students to track their performance. This approach ultimately supports a secure and transparent monitoring process within the student achievement ecosystem.

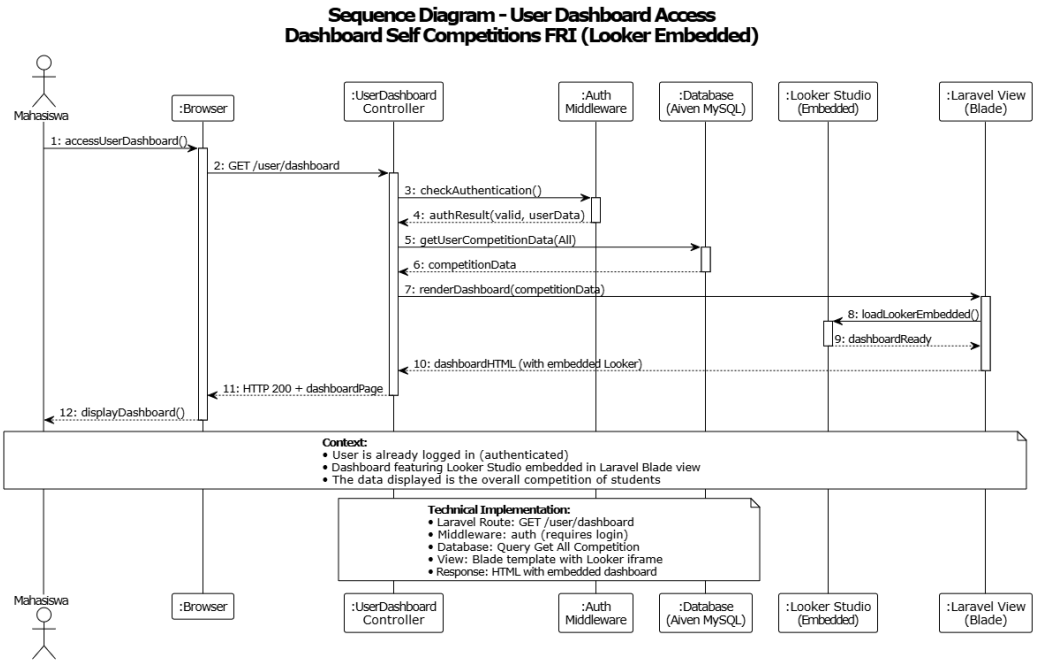

Figure IV- 20. Sequence diagram user dashboard

Based on Figure IV-20 the student access flow follows a simplified single-phase workflow that starts with accessUserDashboard() to the GET /user/dashboard endpoint. The system performs checkAuthentication() to verify the student's login status without requiring admin role verification. The process getUserCompetitionData(All) retrieves the relevant competition dataset from Aiven MySQL, but with filtering according to the student's authorization level. The renderDashboard(competitionData) component prepares the view with the filtered data, then loadLookerEmbedded() loads the Looker Studio visualization in read-only mode. User context shows that students are already authenticated and can access the "overall competition of students" through the embedded dashboard without the ability to modify data. Technical implementation uses Laravel middleware auth with a Blade template that displays an iframe Looker, providing transparency in monitoring the competition while maintaining data security.

IV.3.2.2 Sequence "Admin View Dashboard"

The analytics dashboard feature is a central component of the system that integrates Google Looker Studio as the data visualization engine. To provide comprehensive insights to administrators about the performance of the FRI student competition, the system implements seamless embedding of Looker Studio into the Laravel

framework. The following sequence diagram illustrates the two-phase technical flow of how the admin accesses the analytics dashboard, starting from the choice page selection to the fully loaded embedded Looker dashboard with real-time data integration, as shown in Figure IV-21.

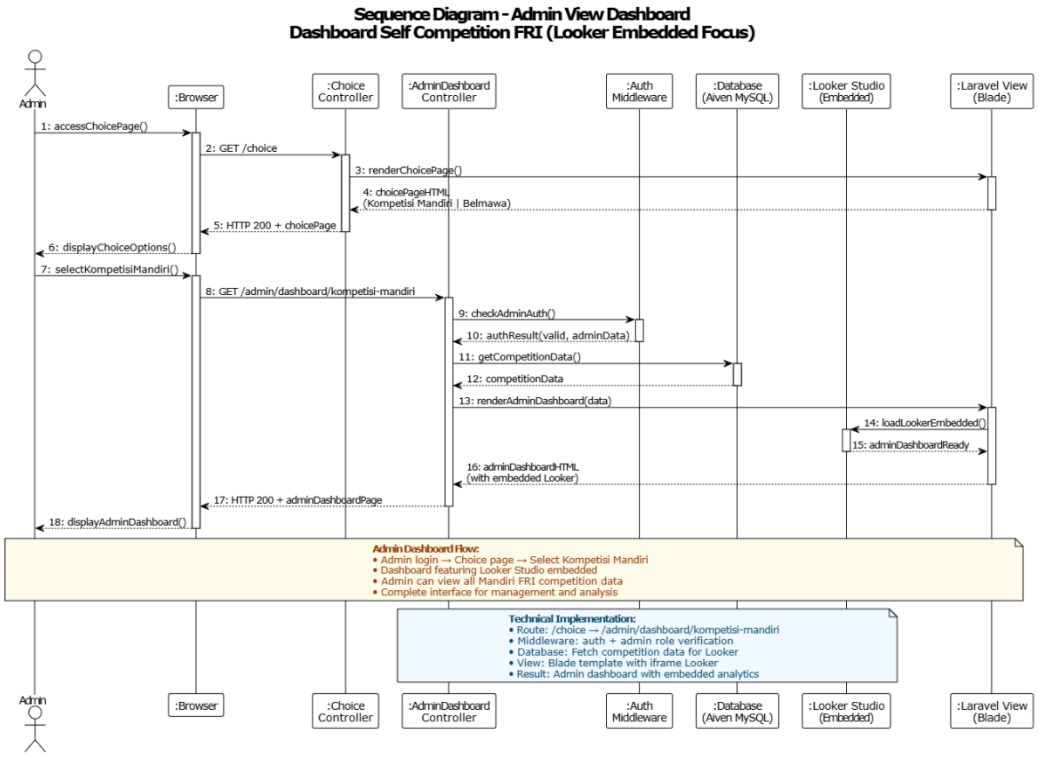

Figure IV- 21. Sequence diagram admin view dashboard

Based on Diagram sequence in Figure IV-21, shows a two-phase workflow for accessing the admin dashboard. The first phase (steps 1-6) is the Choice Page Flow where the admin selects "Kompetisi Mandiri" from the interface selection. The second phase (steps 7-18) is the core Admin Dashboard Selection which starts with GET /admin/dashboard/kompetisi-mandiri. The system performs checkAdminAuth() to verify the admin role, followed by getCompetitionData() to retrieve the competition dataset from Aiven MySQL. The key process occurs in renderAdminDashboard(data), which prepares the data context for Looker Studio, followed by loadLookerEmbedded(), which asynchronously loads the dashboard visualization through iframe embedding. The result is adminDashboardHTML, which displays a complete interface with embedded Looker analytics, allowing

admins to access real-time visualizations, interactive charts, and drill-down capabilities without leaving the Laravel application environment.

IV.3.2.3 Sequence "Admin Add Competition Data"

To maintain the update and completeness of the competition database, the system provides a comprehensive form-based interface for adding new competition data. The data input process follows a four-phase workflow that includes navigation, authentication, form rendering, and data persistence with multiple validation layers. The following sequence diagram details the end-to-end process from the choice page to the successful data insertion into the Aiven MySQL database, as shown in Figure IV-22.

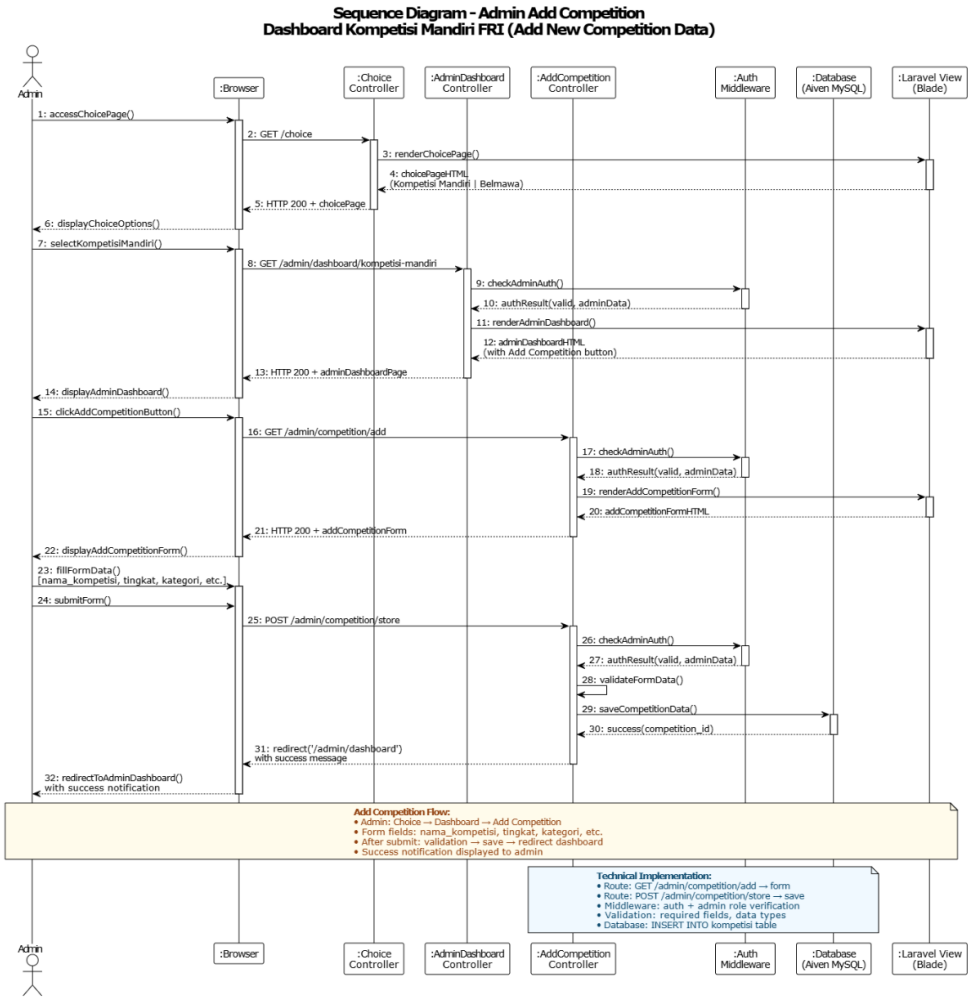

Figure IV- 22. Sequence diagram admin add competition

Based on Figure IV-22 the process of adding competition data follows a four-phase structured workflow. Phase 1 (steps 1-6) is the Choice Page Flow for initial navigation. Phase 2 (steps 7-14) is the Dashboard Access that performs checkAdminAuth() and displays the admin dashboard with the "Add Competition" button. Phase 3 (steps 15-22) is the Add Competition Flow where the system performs GET /admin/competition/add, a second authentication check, and renderAddCompetitionForm() which displays comprehensive form fields such as competition_name, level, category, and other fields. Phase 4 (steps 23-32) is the Submit Form process that starts with fillFormData() by the admin, followed by POST /admin/competition/store, and then validateFormData() to ensure data integrity. After successful validation, the system performs saveCompetitionData() which executes an INSERT operation to the Aiven MySQL database. The process ends with a redirect to the admin dashboard accompanied by a success notification, ensuring the admin receives confirmation that the new data has been saved and can be accessed through the analytics dashboard.

IV.3.2.4 Sequence "Admin Manage Competition Data"

The independent competition dashboard system requires a comprehensive data management mechanism to ensure that administrators can perform Create, Read, Update, and Delete (CRUD) operations on all student competition data. The implementation of CRUD operations follows a four-phase workflow that starts from the choice page, continues to the admin dashboard, then views all competitions, and ends with real-time data manipulation operations. The following sequence diagram illustrates the complete interaction flow involving multiple controllers and ensures that all records can be managed efficiently, as shown in Figure IV-23. This structured workflow helps maintain data consistency and reduces the risk of operational errors during data processing. By clearly defining each stage, the system supports smooth administrative tasks and enables timely updates to competition records. As a result, the platform can deliver accurate and up-to-date information to support decision-making related to student achievements.

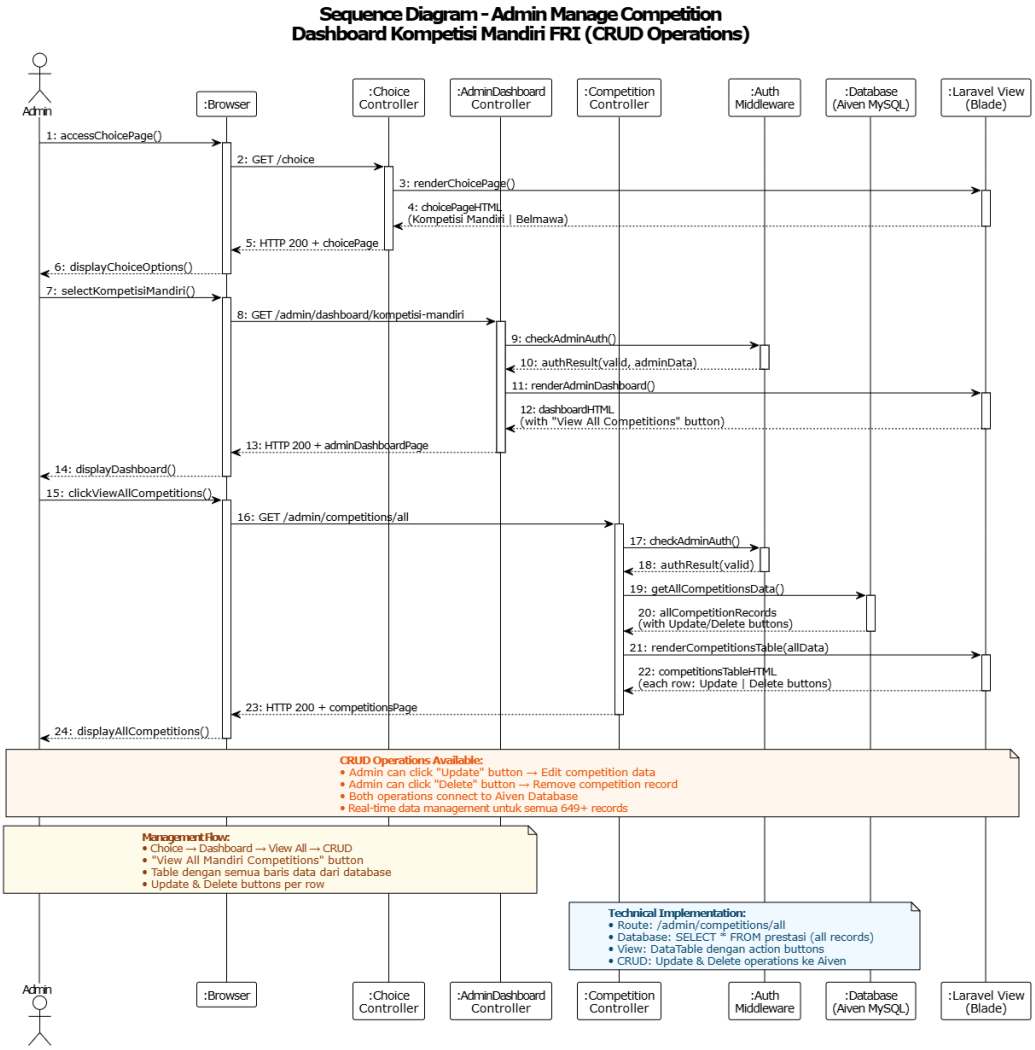

Figure IV- 23. Sequence diagram admin manage competition

Based on the sequence diagram in Figure IV-23, the CRUD operations process is divided into four main phases. The first phase is the Choice Page Flow (steps 1-6) where the admin accesses /choice and selects between "Independent Competition" or "Belmawa". The second phase is the Admin Dashboard Flow (steps 7-14) which performs checkAdminAuth() and displays the dashboard with the "View All Competitions" button. The third phase is the View All Competitions Flow (steps 15-24) which executes getAllCompetitionsData() from the Aiven MySQL database and renders the complete table through renderCompetitionsTable(allData) with each row having "Update" and "Delete" buttons. The fourth phase involves actual CRUD operations where the admin can perform edit or delete operations on data that are directly synchronized with the database. The entire process ensures real-

time data management for all records with a user-friendly interface and secure operations through multiple authentication checks.

IV.3.2.5 Sequence “Admin Interaction with Chatbot”

The integration of Artificial Intelligence is a key innovation of the system that enables natural language interaction for data analysis. Using the Flowise AI platform integrated with OpenAI GPT-4o, the system is capable of processing questions in Indonesian and converting them into actionable SQL queries. The following sequence diagram shows a sophisticated two-phase workflow of how the AI chatbot processes user intent, performs database queries, and generates contextual insights in the format of a natural language response. as shown in Figure IV-24

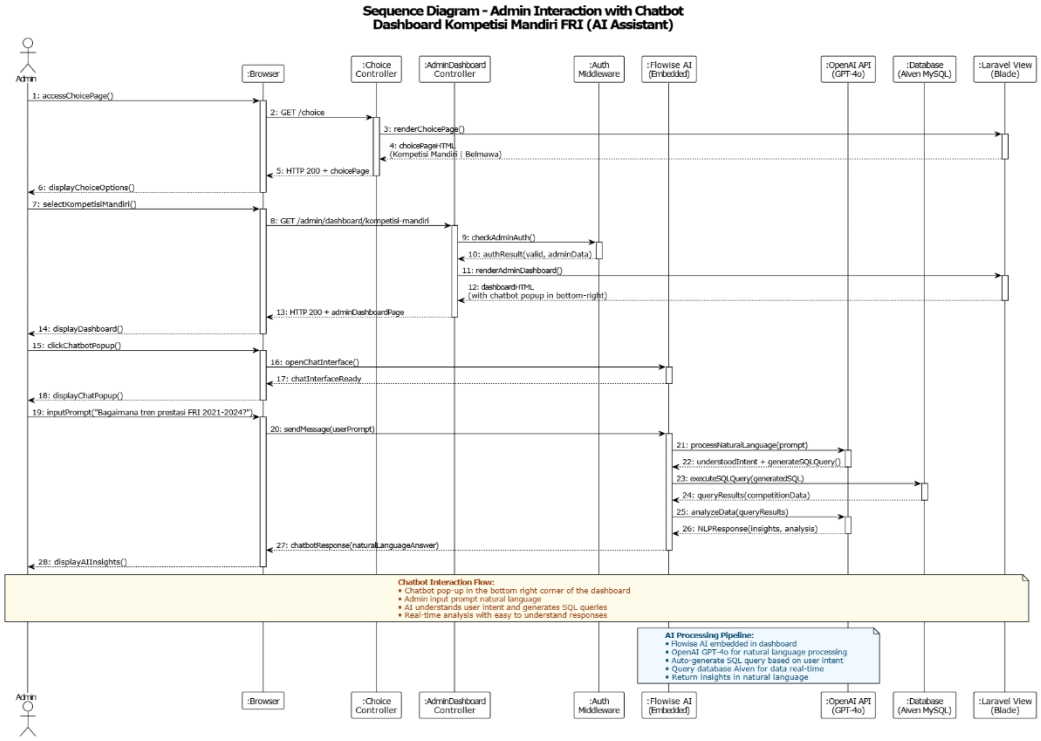

Figure IV- 24. Sequence diagram admin interaction chatbot

Based on Figure IV-24. AI chatbot interactions follow a complex two-phase workflow. Phase 1 (steps 1-14) is the Choice to Dashboard Flow, which is similar to admin dashboard access, but the dashboard is equipped with a "chatbot popup in the bottom-right" corner. Phase 2 (steps 15-28) is the core Chatbot Interaction Flow,

which starts with clickChatbotPopup() and openChatInterface(). When the admin inputs a prompt like "What are the performance trends of FRI 2021-2024?", the system sends sendMessage(userPrompt) to Flowise AI, which then processes the natural language using OpenAI GPT-4o. The AI engine performs understoodIntent + generateSQLQuery() to convert natural language into a valid SQL query, followed by executeSQLQuery(generatedSQL) to retrieve real-time data from Aiven MySQL. The query results are then processed through analyzeData(queryResults) and formulated into an NLP Response (insights, analysis). The AI Processing Pipeline allows admins to obtain sophisticated analytics such as trend analysis, performance metrics, and strategic recommendations simply by using conversational Indonesian language, democratizing access to complex business intelligence.

IV.4 Implementation

The implementation stage is the process of realizing the system design that was created in the previous stage. Implementation is carried out in stages following the designed 4-layer architecture, starting from the Data Layer as the foundation of the system, then the Visualization Layer, Intelligence Layer, Presentation Layer, and ending with the overall system integration.

Implementation uses the Waterfall methodology, which allows development to be carried out systematically and structurally. Each layer is implemented by considering the functional and non-functional requirements that have been previously analyzed.

IV.4.1 Data Layer Implementation

The implementation of the data layer begins with the preparation of database infrastructure and data migration from available data sources. This layer serves as the foundation for the entire system because it provides the data needed by the layers above it.

IV.4.1.1 Database Schema Implementation

The implementation of the data model in this research uses the Star Schema approach optimized for analyzing historical data on student achievements in

independent competitions from 2021 to 2024. The selection of the star schema is based on the characteristics of historical data that require an appropriate structure to support the analysis of past trends, time-based reporting, and integration with artificial intelligence technology to generate strategic insights from performance patterns. This model consists of a main fact table (fact_kompetisi) containing 649 historical achievement records, which is connected to five-dimension tables, namely dim_mahasiswa_mandiri, dim_kompetisi_mandiri, dim_time_mandiri, dim_program_mandiri, and dim_dosen_mandiri. These five tables provide the necessary context so that each achievement data can be analyzed in more detail.

This star schema structure was specifically chosen because it aligns with the nature of historical data, which tends to be frequently accessed for read-intensive purposes and requires fairly complex aggregation queries to observe trends over time. The denormalization process applied allows the dashboard system to access and display historical data efficiently, thereby supporting time-series data visualization as well as period comparison analysis, as shown in Figure IV-25

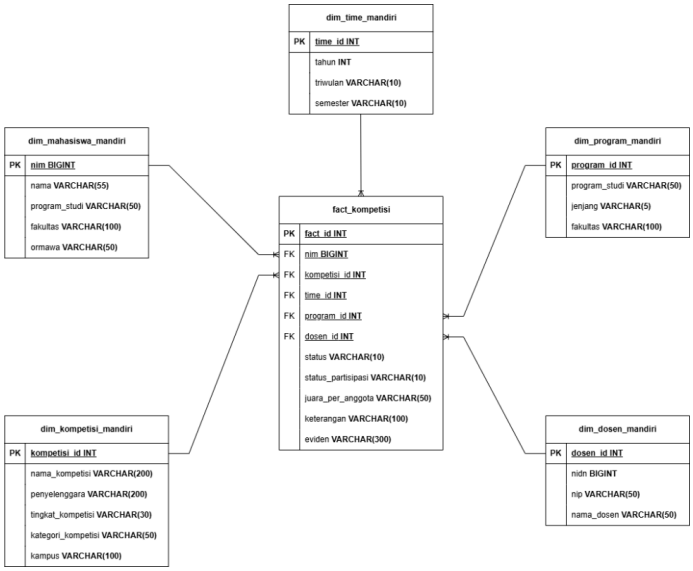

Figure IV- 25. Star schema competition data

Based on Figure IV-25 the Star Schema model used in this research serves as the foundation for a robust data warehouse to support in-depth analysis of historical performance data over the past four years. With this structure, Looker Studio can perform historical trend analysis with optimal query speed, while Flowise AI can

process retrospective questions and generate insights from data patterns found in the dataset. Historical data is also kept intact by preserving NULL values if they exist, ensuring data integrity from year to year and allowing for accurate development analysis.

With detailed data organized down to the level of achievements per student, per competition, and per time period, this model supports flexible analysis, both for observing achievement trends at the institutional level and for tracking individual performance over time. The application of this star schema also opens up opportunities for development towards predictive analysis based on historical patterns, making it an important asset to support strategic decision-making and program planning at the Faculty of Industrial Engineering, Telkom University.

IV.4.1.2 Data Migration Process

The cleaned and transformed data (21 columns, 649 records) needs to be migrated to the cloud infrastructure to ensure scalability, reliability, and optimal performance. The migration process is carried out through several stages involving format conversion and database management tools, as shown in Figure IV-26.

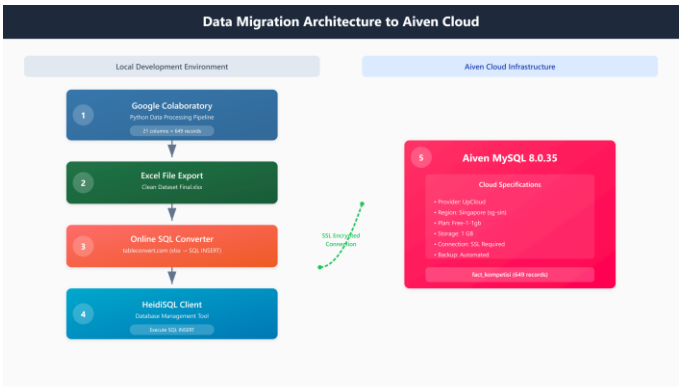

Figure IV- 26. Architecture of data migration to aiven cloud

Based on Figure IV-26 illustrates the migration process from the cleaned Excel file to the Aiven MySQL instance. The migration process is carried out through stages:

1. Export to Excel: The final dataset (21 columns) is saved as an Excel file from Google Colaboratory.

2. SQL Generation: Using an online converter (xlsx to SQL) to generate INSERT statements.
3. Database Import: Using HeidiSQL as a database management tool to connect to Aiven.
4. Execute SQL: Running INSERT statements to Aiven MySQL through a secure SSL connection.
5. Validation: Verifying data integrity and row count in Aiven.

The use of HeidiSQL was chosen because of its user-friendly interface and robust SSL connection support for connecting to Aiven Cloud. An online converter is used for the efficient batch conversion of data from Excel to ready-to-execute SQL INSERT statements.

Specifications of the Aiven deployment used:

- Service Type: MySQL 8.0.35
- Cloud Provider: UpCloud
- Region: sg-sin(Singapore)
- Plan: Free-1-1gb (CPU, 1 GB RAM, 1 GB storage, backups for disaster recovery)
- Connection: SSL Required

It is important to note that missing values (NULL) in the dataset are retained during migration to maintain the integrity of historical data. Handling NULL values is done at the visualization layer (Looker Studio) using dynamic filters, not at the database level. This approach allows flexibility in analysis users can choose to include or exclude records with missing values according to the context of the analysis without altering the source data.

IV.4.2 Visualization Layer Implementation

The implementation of the visualization layer in this research was carried out using Google Looker Studio. This layer serves to present student competition data in an informative and interactive manner through visualizations that are easy for users, such as Deans and Student Affairs Staff, to understand. The visualization is designed to support the monitoring of independent competition achievements and

provide a deeper understanding of the historical data trends that have been collected and processed, as shown in Figure IV-27.

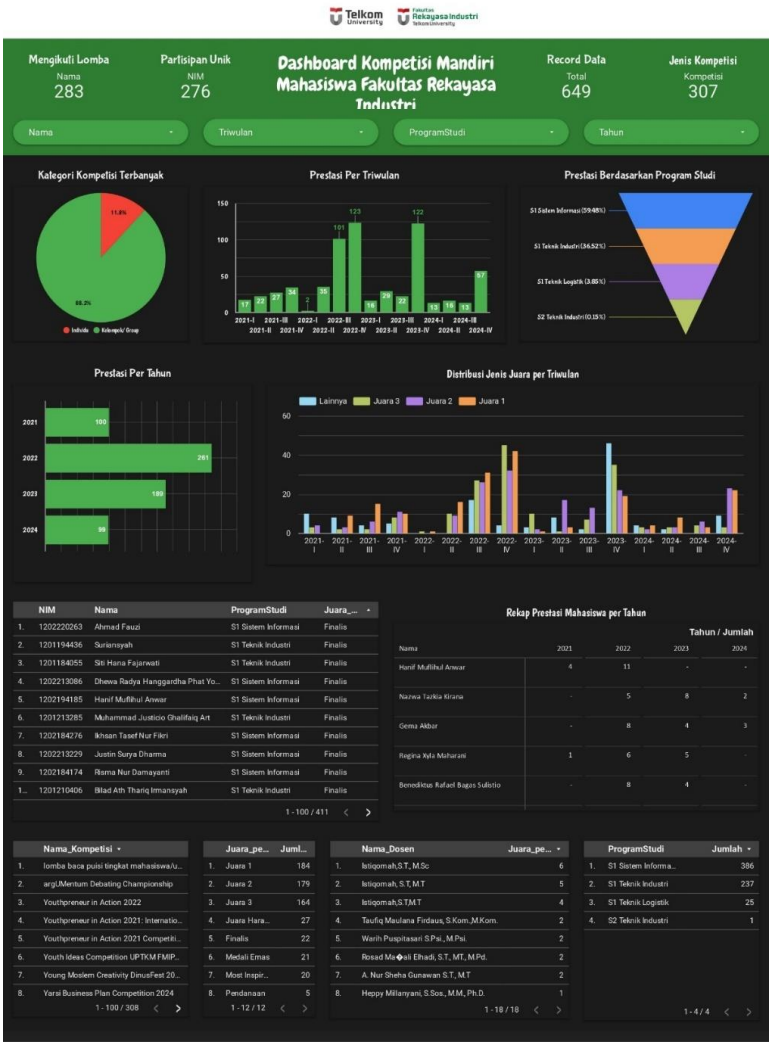

Figure IV- 27. Design dashboard looker studio

Based on Figure IV-27 at the top of the dashboard, several key statistics are displayed, such as the total recorded competition data, the number of unique participations based on student ID, the total types of competitions, and the total number of participating students. This serves as a summary of the key statistics to provide an overview of the scope and scale of the data displayed in the system.

The dashboard is equipped with various types of graphs to present different dimensions of the data.:

- The Pie Chart displays the most frequent competition categories, showing the proportion between individual and group competitions. The choice of this visualization aims to provide a direct and easily graspable representation of proportions for the users.
- The Vertical Bar Chart in the Quarterly Achievement section displays the fluctuations in student achievement over three-month periods from 2021 to 2024. This type of chart was chosen because it is effective in showing the comparison of quantities across time periods linearly.
- The Funnel Chart in the Academic Achievement by Study Program section shows the contribution of each study program to the recorded number of achievements. The funnel shape provides a visual effect of contributions from largest to smallest hierarchically.
- The Horizontal Bar Chart displays Achievements Per Year, making it easier for users to directly compare total annual achievements.
- Stacked Bar Chart is used to present the Distribution of Champion Types Per Quarter, which separates each type of champion (Champion 1, 2, 3, and others) on the same time axis. This visualization facilitates the analysis of distribution and consistency of achievements per category in each period.
- The Interactive Table at the bottom of the dashboard presents detailed data such as a list of outstanding students, competition names, supervisor names, and a recap of student achievements per year. The table is equipped with search and sorting features to facilitate data exploration.

The selection of visualization types in the dashboard is carried out by considering the characteristics of the data, user needs, and the effectiveness of information delivery. Each type of graph used is evaluated based on the suitability of the visual form to the data structure and the analysis objectives. Other visualization alternatives such as line charts, tree maps, or gauge charts were considered, but not used because they were deemed less effective in the context and due to the limited space of the dashboard. With this approach, the applied visualization is expected to maximize user understanding of the data without causing confusion or visual overload, the comparison will be shown in Table IV-7.

Table IV- 7. Analysis of selection and comparison of visualization types on dashboards

No	Data Displayed	Visualization Used	Alternative Visualization	Reason Not Used	Why Visualization is Now More Appropriate
1	Most Competitive Categories	Pie Chart	Bar Chart Horizontal	Bar charts are not very effective for comparing two simple categories.	Pie charts are more concise for displaying the proportions of two mutually exclusive categories (individual vs group).
2	Student Achievements per Quarter	Vertical Bar Chart	Line Chart	Line charts emphasize continuity, even though the data is not temporally related (not daily time series data).	Bar charts effectively show the fluctuations in quarterly achievements, which are discrete in nature.
3	Achievements Based on Study Program	Funnel Chart	Bar Chart, Tree Map	The bar chart does not show the effect of the contribution order from largest to smallest hierarchically.	Funnel chart is suitable for displaying contribution rankings with a visual emphasis from top to bottom.
4	Number of Achievements per Year	Horizontal Bar Chart	Column Chart	Column charts take up more vertical space.	The horizontal bar saves dashboard space and makes it easier to read the year names on the X-axis.
5	Distribution of Champion Types per Quarter	Stacked Bar Chart	Grouped Bar, Multiple Lines	Grouped bar makes the visual too dense; multiple lines are not suitable because the data is not a continuous trend.	Stacked bar allows for comparison between different types of champions as well as total achievements in one view.
6	List of Students, Lecturers, and Competitions	Table	Pivot Table, Detail Cards	Pivot is too complex for non-technical users; detail cards take up a lot of space.	Interactive tables allow for quick sorting, searching, and skimming of tabular data.
7	Summary Statistics (Number of Competitions, Data, etc.)	Metric Card (Stat Panel)	Gauge Chart, Speedometer	Gauge chart is not relevant because there is no target or threshold.	Metric cards are suitable for static/total numbers and are visually concise.

IV.4.3 Intelligence Layer Implementation

The implementation of the artificial intelligence layer (Intelligence Layer) is a core component that provides AI-based analysis capabilities to the independent competition dashboard system. This layer uses the Flowise AI platform integrated with OpenAI GPT-4o to provide a conversational interface in Indonesian, allowing users to obtain insights and competition data analysis through natural language questions.

IV.4.3.1 Architecture System AI Chatbot

The Intelligence Layer architecture is designed using three main components that are integrated with each other. The first is the Database Schema Extractor, which is responsible for retrieving the database structure in real-time and providing schema context to the AI model. The second is the Natural Language Processor that uses ChatOpenAI GPT-4o to convert questions in Indonesian into SQL queries. The third is the Query Execution Engine, which runs the query on the database and returns the results for further analysis, as shown in Figure IV-28.

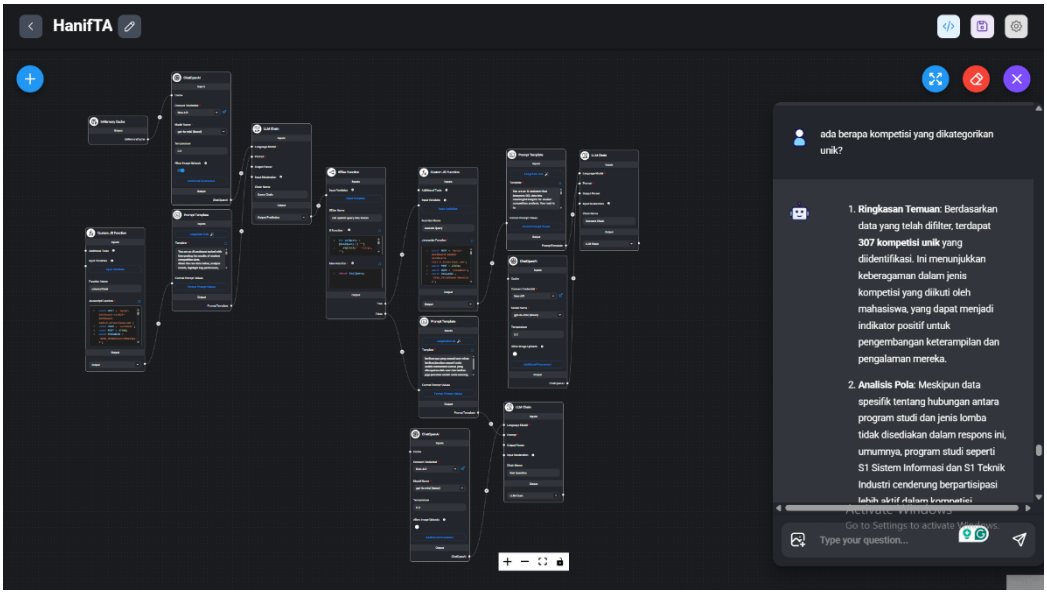

Figure IV- 28. Flowise chatbot interface

Based on Figure IV-28 this system implements a multi-chain processing architecture where each chain has a specific function. Query Chain is responsible for converting natural language into SQL queries, Answer Chain analyzes the query

results and provides insights, while Fallback Chain handles questions that cannot be processed through database queries.

IV.4.3.2 Main Components and Their Functions

- Database Schema Extractor

This component serves as a bridge between the AI model and the database structure. It automatically extracts schema information from the Aiven MySQL database, including table names, columns, and data types. This information is then provided as context to the AI model to ensure that the generated queries align with the existing database structure.

Table IV- 8. Configuration database connection

Parameter	Value	Description
Host	mysql-dashboard-ta2025***	Aiven MySQL endpoint
Port	27***	Database port
Database	prestasi_*****	Database name
SSL	Required	Encrypted connection
Connection Pool	5 connections	Optimized performance

The configuration in Table IV-8 shows the connection parameters used to link the system with the Aiven MySQL database, which serves as the primary source of data for the independent competition. The use of SSL (Secure Socket Layer) is mandatory to ensure an encrypted and secure connection between the system and the database server. The port and host values are configured according to the endpoint provided by the Aiven service, while the database name is adjusted to the schema containing all historical data of the FRI student competition.

The use of a connection pool with five parallel connections aims to optimize performance during simultaneous data access requests, especially when users activate the chatbot or open the dashboard simultaneously. These parameters not only ensure the success of technical integration but also provide the context needed by the AI system to build an understanding of the data structure to be analyzed.

With this configuration, the system is able to automatically extract the database structure, including table names, attributes (columns), and data types, which are then used as references when the AI model processes natural language-based requests. This process becomes crucial in bridging the communication between users and the database through a chatbot based on Flowise AI and OpenAI API.

- Natural Language Processor

Using the ChatOpenAI GPT-4o model with a temperature configuration of 0.2 to ensure result consistency. This component is equipped with prompt engineering specifically designed for the student competition domain, enabling the AI to understand academic terminology and generate accurate SQL queries, as shown in Figure IV-29.

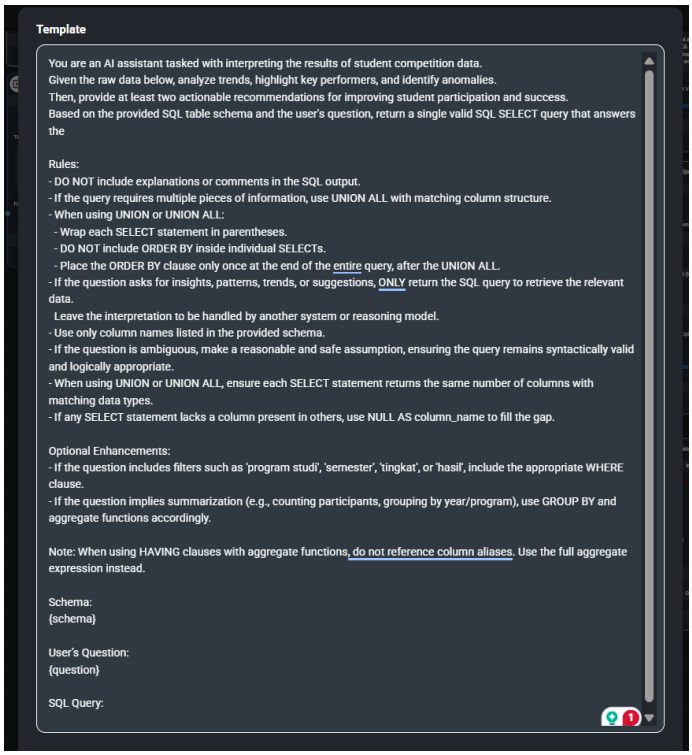

Figure IV- 29. Prompt template SQL

Based on Figure IV-29 the system is equipped with query validation using conditional logic to ensure the security and validity of the generated SQL queries. The validator checks the SQL syntax and ensures that only SELECT queries are allowed to prevent operations that could damage the data.

IV.4.3.3 AI Chatbot Workflow

The process begins when the user inputs a question in Indonesian through the chatbot interface. The system then performs a series of processing steps to generate an accurate and informative answer, as shown in Figure IV-30.

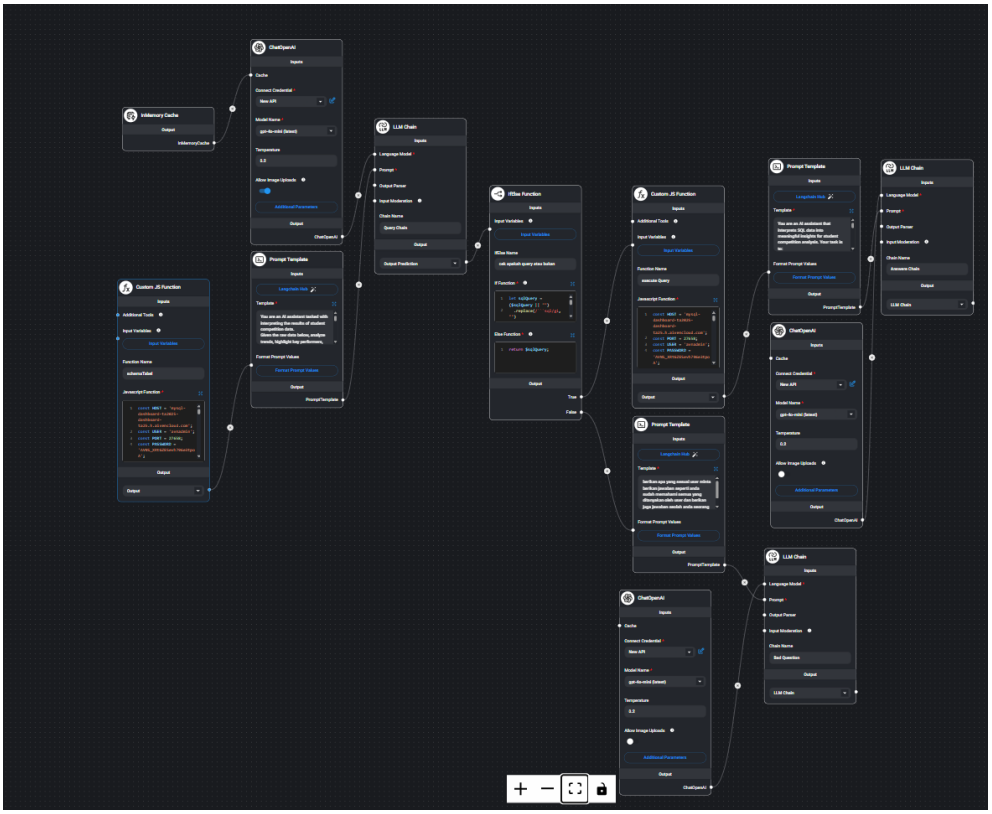

Figure IV- 30. AI chatbot workflow

Based on Figure IV-30, the author will explain the workflow of the AI Chatbot system that has been developed for analyzing student competition performance. This workflow shows how the system processes questions in Indonesian and generates comprehensive analyses through a series of integrated stages.

- Stage 1 Performance Optimization with InMemory Cache: The process begins with the implementation of a cache system connected to the ChatOpenAI model. The InMemory Cache feature functions to store frequently requested responses in the application's memory, so when users ask similar questions, the system can provide answers more quickly without needing to reprocess the database. This is very effective for handling

common questions such as statistics on the number of competitions per year or the performance of certain study programs.

- Stage 2 Database Schema Extraction: The system uses a Custom JavaScript Function connected to Aiven Cloud MySQL to extract database structure information in real-time. The database connection uses SSL encryption to ensure data security with the host configuration 'mysql-dashboard-ta2025-dashboard-ta25.h.aivencloud.com' and the database 'prestasi_kompetisi' focusing on the table 'fact_kompetisi'. At this stage, the system not only retrieves the table structure but also fetches the latest 10 sample rows as context to provide a better understanding to the AI regarding the available data format and types.
- Stage 3 Natural Language Processing and SQL Generation: When users input questions through the chatbot interface, the system uses a Prompt Template specifically designed for the domain of student competition achievements. This template understands academic terminology, types of competitions (individual/group), the four existing study programs (Bachelor's in Information Systems, Bachelor's in Industrial Engineering, Bachelor's in Logistics Engineering, and Master's in Industrial Engineering), as well as various performance evaluation metrics. The prompt is then sent to ChatOpenAI with the GPT-4o model configured with a temperature of 0.2 to ensure high consistency and accuracy of responses.
- Stage 4 Query Validation and Cleaning: The result from the first LLM Chain enters the IfElse Function, which serves as the query validator. This function performs two main tasks: first, it checks whether the output generated is a valid SQL query by looking for the keyword "SELECT"; second, it cleans the formatting by removing markdown code blocks that may have been generated by the AI. This validation stage is a layered security measure that prevents SQL injection while ensuring the quality of the query to be executed.
- Stage 5 Database Execution with Error Handling: The validated query is then passed to the second Custom JavaScript Function, which is responsible for executing the query to the Aiven Cloud database. The system is equipped

with connection pooling, SSL security, robust error handling, timeout management, and a fallback mechanism. If the AI-generated query fails to execute, the system will automatically run a safe default query to display sample data from the competition table.

- Stage 6 Intelligent Analysis and Insight Generation: After obtaining the query results, the system uses a second LLM Chain with a Prompt Template specifically designed for student competition analysis. This template instructs the AI to conduct an in-depth analysis that includes: identifying achievement trends per year, analyzing performance per study program, detecting anomalies or outstanding achievements, and providing data-driven strategic recommendations to enhance student participation and performance. AI not only displays raw data but also provides meaningful insights and interpretations.
- Stage 7 Fallback Handling and Error Prevention: This stage is an alternative handling path designed as a safety mechanism when the previous stages fail to produce a valid SQL query. The author realizes that although AI technology is very advanced, there is still a possibility that AI may make mistakes in interpreting user questions or producing output that does not conform to the expected SQL format. When the IfElse Function in Stage 4 detects that the output from the first LLM Chain is not a valid SQL query (does not contain the keyword "SELECT" or contains a syntax error), the system will automatically redirect the processing flow to the "False" path connected to the third LLM Chain. This path uses a different Prompt Template, which allows the AI to provide general responses as an expert in the field of student competition achievements without performing a database query. The implementation of this fallback mechanism aims to prevent the system from crashing or encountering errors when facing questions that cannot be processed normally. With this alternative path, the system can still provide meaningful responses to users, even if it cannot access specific data from the database. This ensures the continuity of user experience and maintains the overall reliability of the system. This approach reflects the principles of defensive programming in AI system development,

where every possible failure has been anticipated and provided with alternative solutions to maintain system functionality.

IV.4.3.4 Example User Interaction

To provide a concrete picture of the system's capabilities, here is an example of an interaction between the user and the AI chatbot, as shown in Figure IV-31:

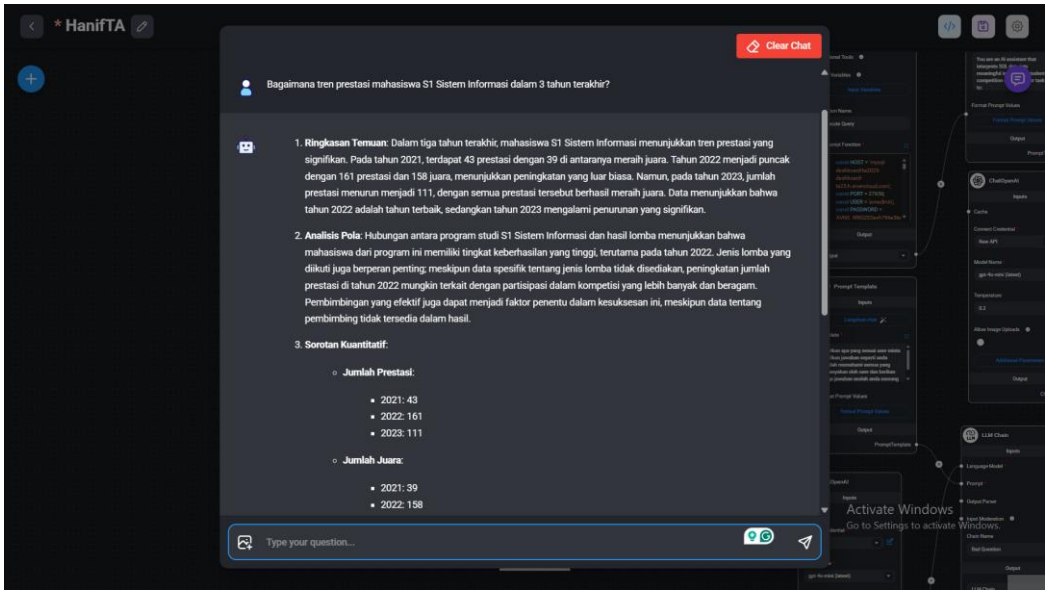

Figure IV- 31. Example conversation with AI chatbot

Based on Figure IV-31, User Input: "What is the trend in the academic performance of undergraduate Information Systems students over the past 3 years?"

System Process:

- AI analyzes the question and identifies what data is needed
- Generate SQL query for SI achievement data 2021-2023
- Execute the query on the database
- Analyze the results to identify trends

System Output:

1. Summary of Findings: In the past three years, undergraduate students in Information Systems have shown a significant achievement trend. In 2021, there were 43 achievements with 39 of them winning first place. The year 2022 peaked with 161 achievements and 158 first places, showing an extraordinary increase. However, in 2023, the number of achievements decreased to 111, with all of

them winning first place. Data shows that 2022 was the best year, while 2023 experienced a significant decline.

2. Pattern Analysis: The relationship between the Bachelor's degree program in Information Systems and competition results shows that students from this program have a high success rate, especially in 2022. The types of competitions participated in also play an important role; although specific data about the types of competitions is not provided, the increase in achievements in 2022 may be related to participation in more diverse and numerous competitions. Effective mentoring can also be a determining factor in this success, although data about the mentors is not available in the results.
3. Quantitative Spotlight:
    - Number of Achievements:
        1. 2021: 43
        2. 2022: 161
        3. 2023: 111
    - Number of Champions:
        1. 2021: 39
        2. 2022: 158
        3. 2023: 111
    - Win Ratio:
        1. 2021: 90.7%
        2. 2022: 98.1%
        3. 2023: 100%
    - Program Dominance: The Bachelor's in Information Systems has shown dominance in achievements, especially in 2022.
4. Strategic Recommendations:
    - Improvement of Mentoring: Implementing a better mentor recording system to ensure students receive appropriate and consistent guidance.
    - Diversification of Competitions: Encouraging students to participate in various types of competitions, both individual and group, to enhance their experience and skills.

- Motivation and Award Program: Organizing an award program for outstanding and active students to enhance motivation and participation in competitions.

5. Projection / Intervention Plan: If the above strategy is implemented, it is expected that student participation in competitions can increase significantly, with a projected increase in achievements by 20-30% in the following year. Realistic follow-up steps include surveys to identify student interest in specific types of competitions and periodic evaluations of the effectiveness of mentoring.

Confirmation: This analysis is based on filtered data relevant to the S1 Information Systems study program.

IV.4.3.5 Enhanced Analytics with Clustering Analysis

To provide deeper analytical capabilities and strategic insights, the AI chatbot system is enhanced with clustering analysis functionality. This enhancement enables the system to automatically segment students based on their competition participation patterns and provide personalized recommendations for faculty management.

The clustering analysis employs the K-Means algorithm to analyze historical competition data from 2021-2024, encompassing 649 competition records from 278 unique FRI students. The analysis utilizes eight performance parameters including total competitions, win rate, consistency score, and performance diversity to create meaningful student segments.

The optimal number of clusters was determined to be 8 through validation using the Silhouette Score method, achieving a score of 0.705, which indicates good clustering quality. This validation ensures that the student segmentation is both statistically sound and practically meaningful for strategic decision-making.

Based on the clustering results, students are categorized into three main performance levels as shown in Figure IV-32

1. High Performers (14.4%): Students with consistent high-level competition participation and excellent win rates

2. Medium Performers (14.4%): Students showing moderate participation with potential for improvement
3. Low Performers (71.2%): Students with sporadic participation requiring intensive mentoring programs

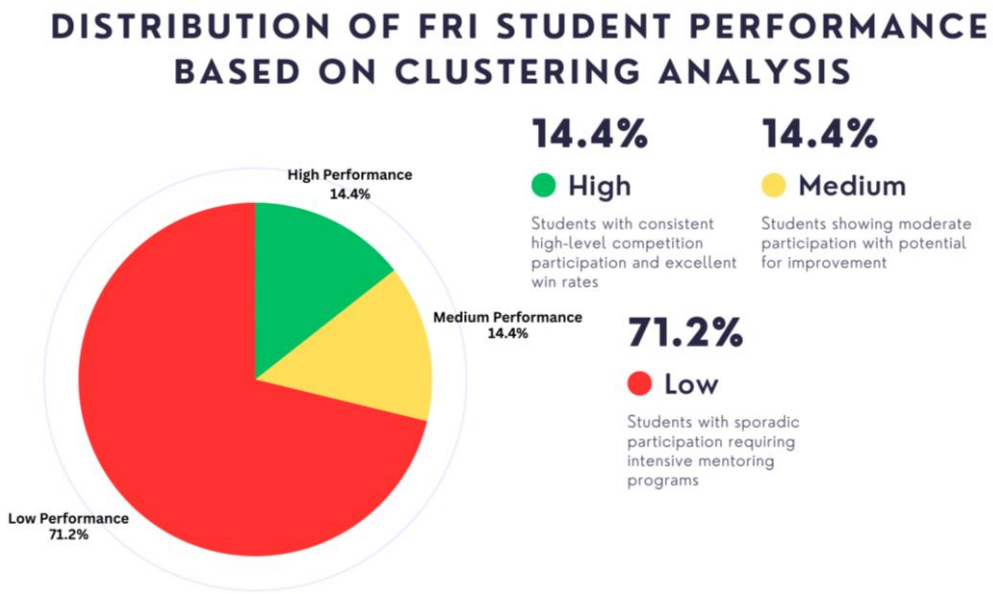

Figure IV- 32. Student performance distribution

The analysis on Table IV-8 that reveals significant variations across study programs, as presented in Table IV-X. The S1 Information Systems program demonstrates the highest proportion of High Performers (9.4%), followed by S1 Industrial Engineering (4.6%). Notably, both S1 Logistics Engineering and S2 Industrial Engineering programs show 100% of students in the Low Performer category, indicating the need for targeted intervention programs.

Table IV- 9. Performance Distribution by Study Program

Study Program	High Performer	Medium Performer	Low Performer
S1 Information Systems	9.4%	14.8%	75.8%
S1 Industrial Engineering	4.6%	16.5%	78.9%
S1 Logistics Engineering	0.0%	0.0%	100.0%
S2 Industrial Engineering	0.0%	0.0%	100.0%

These clustering insights are directly integrated into the AI chatbot's response system, enabling it to provide context-aware recommendations. When faculty members query about student performance trends or request strategic recommendations, the chatbot can now deliver personalized insights based on the specific performance segment and study program characteristics.

The strategic recommendations generated from clustering analysis include:

1. For High Performers: Implementation of mentoring programs where these students serve as peer tutors, provision of opportunities for international competitions, and advanced skill development workshops.
2. For Medium Performers: Targeted coaching programs to enhance consistency, skill development workshops focused on competition strategies, and pairing with High Performer mentors.
3. For Low Performers: Intensive motivational programs, basic competition skills workshops, structured mentoring with faculty guidance, and gradual participation increase strategies.

The integration of clustering analysis transforms the AI chatbot from a simple query-response system into an intelligent decision support tool that can identify at-risk students, recognize high-potential candidates for advanced competitions, and suggest targeted intervention strategies for different performance segments.

IV.4.4 Presentation Layer Implementation

Implementation of the presentation layer using the Laravel framework to provide a web interface that combines a dashboard and an AI chatbot into a single integrated platform.

IV.4.4.1 Laravel Application Setup

The implementation of the presentation layer in this research begins with preparing the Laravel 10 application as the main framework that serves as the system's control center. Laravel was chosen because it has a clear Model-View-Controller (MVC) structure, equipped with a secure built-in authentication system, and supports integration with other technologies such as Looker Studio and Flowise AI.

The initial setup steps were carried out by installing all the necessary dependencies using Composer, then configuring the environment variables so that the application could connect to the Aiven MySQL database. The application key is also configured to maintain the security of data and user sessions. The folder structure of Laravel is organized following the MVC pattern, where Controllers are used for business logic, Models for data management, and Views are created with the Blade templating engine to display the user interface.

The database connection settings are adjusted according to the details of the Aiven MySQL Cloud Database account used at the data layer. In addition, the environment configuration is also differentiated between development and production to maintain system security and stability, as shown as in Figure IV-32, Figure IV-33, Figure IV-34, Figure IV-35, Figure IV-36, Figure IV-37, and Figure IV-38.

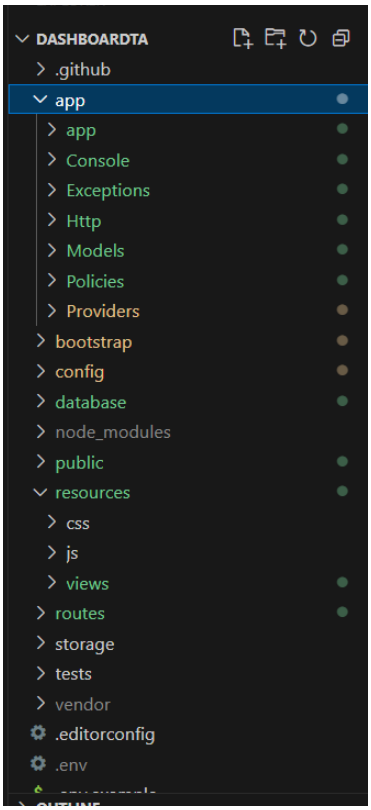

Figure IV- 33. Root structure

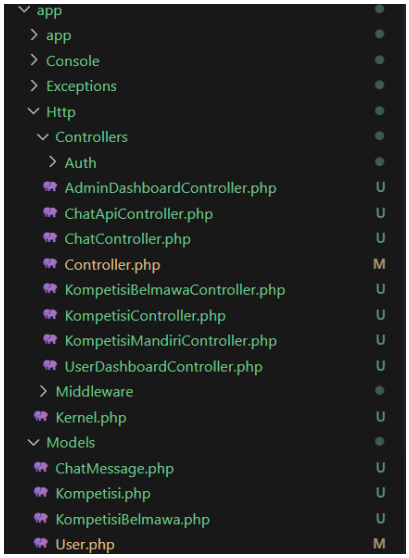

Figure IV- 34. Controllers and models

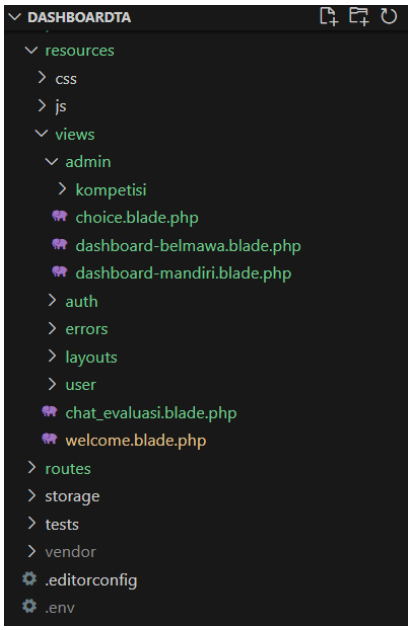

Figure IV- 35. Blade views

Based on Figure IV-32, Figure IV-33, and Figure IV-34 overall displays the structure of the Laravel project used as the presentation layer in this research. The first image (Figure IV.32) shows the main directory structure of the Laravel project, starting from the app/, resources/, routes/ folders, up to the configuration file. env which functions for configuring the Aiven MySQL database connection.

Figure IV.33 displays the Controllers and Models sections, which contain files that control business logic for authentication processes, competition management, chatbot interaction, as well as entity models used to manage competition data and

user data. Figure IV.34 shows the structure of Views organized using the Blade templating engine, with folder divisions based on functions such as admin/competitions for standalone dashboard pages and Belmawa, auth/ for authentication, as well as chatbot view files and the main page.

Overall, these three images support the explanation that the Laravel application is organized using the Model-View-Controller (MVC) pattern, which separates logic, data, and views to make system development more structured, easier to maintain, and flexible for integration with Looker Studio and Flowise AI.

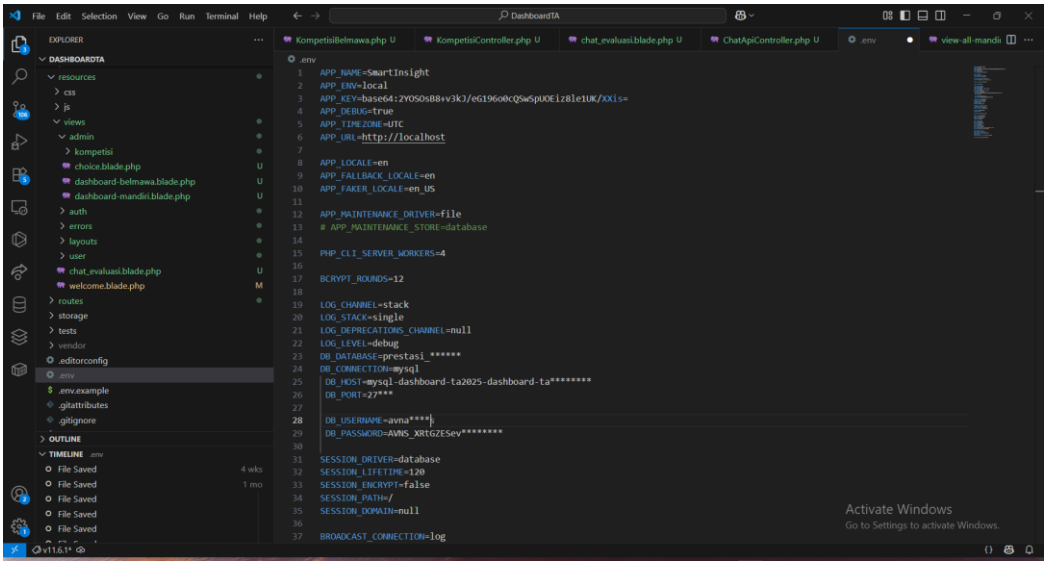

Figure IV- 36. Database connection configuration (.env)

Based on Figure IV-35, this shows a snippet of the .env file in the Laravel project used to configure the database connection. This file contains important variables such as DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, and DB_PASSWORD configured to connect to the Aiven MySQL Cloud Database. The use of the .env file facilitates the configuration of different environments (development or production) without having to directly change the program code. This configuration also supports security practices by separating sensitive credentials from the application logic.

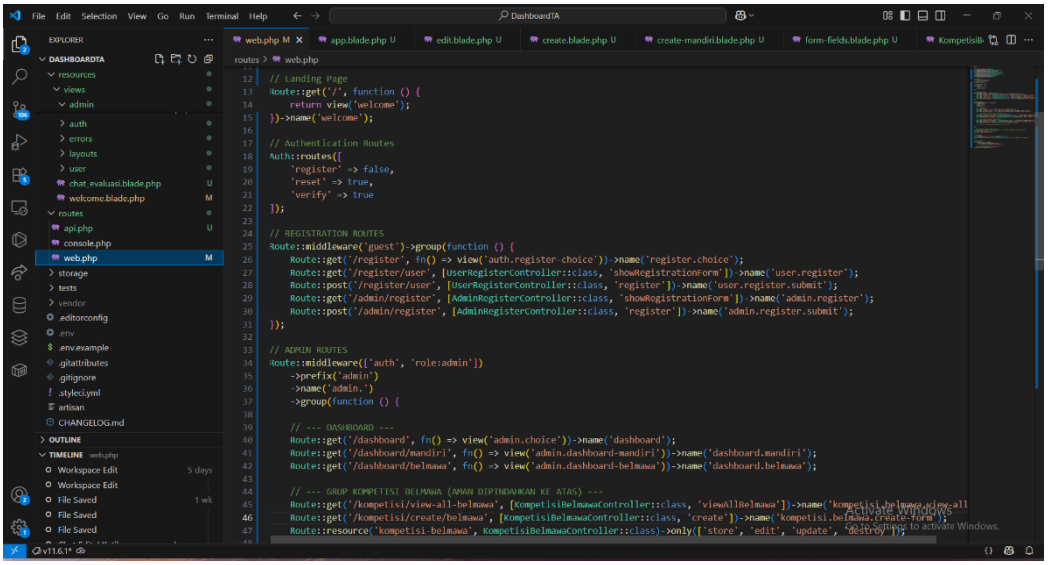

Figure IV- 37. Routing file (web.php)

Based on Figure IV-36, it displays a snippet of the web.php file that functions to define HTTP routing in a Laravel application. This routing file manages access paths for important pages, such as the landing page, authentication, user registration, as well as the admin route group for the dashboard and competition management. This routing structure uses middleware and prefixes to differentiate access rights between regular users and admins, supporting a neat and structured authorization implementation.

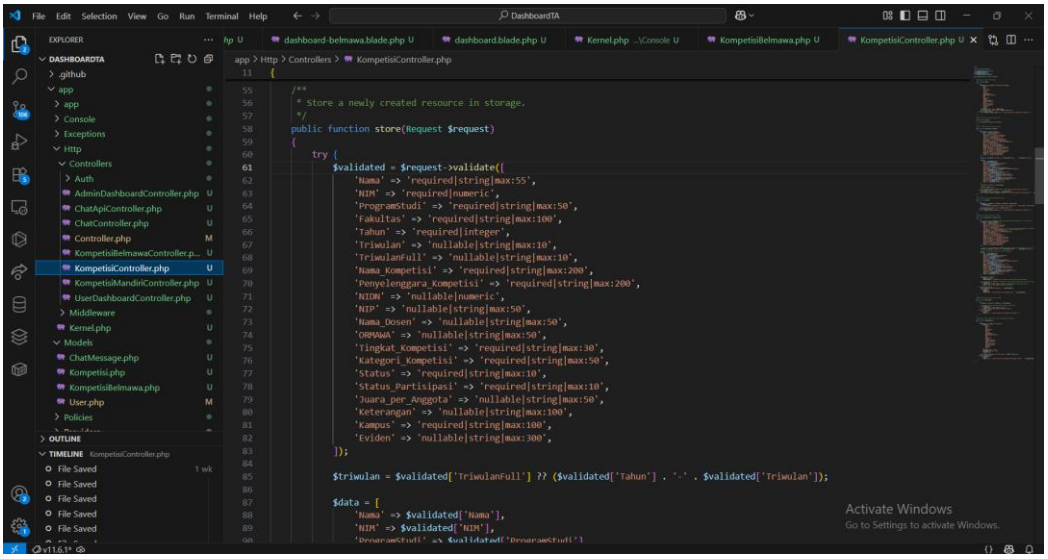

Figure IV- 38. Example controller file

Based on Figure IV-37, this shows an example of a controller file, namely KompetisiController.php, which handles the business logic related to competition

data management. This snippet shows the input validation section using $request->validate to ensure that every incoming data conforms to the specified format and rules. This controller serves as a bridge between user requests, data processing in the model, and output to the view.

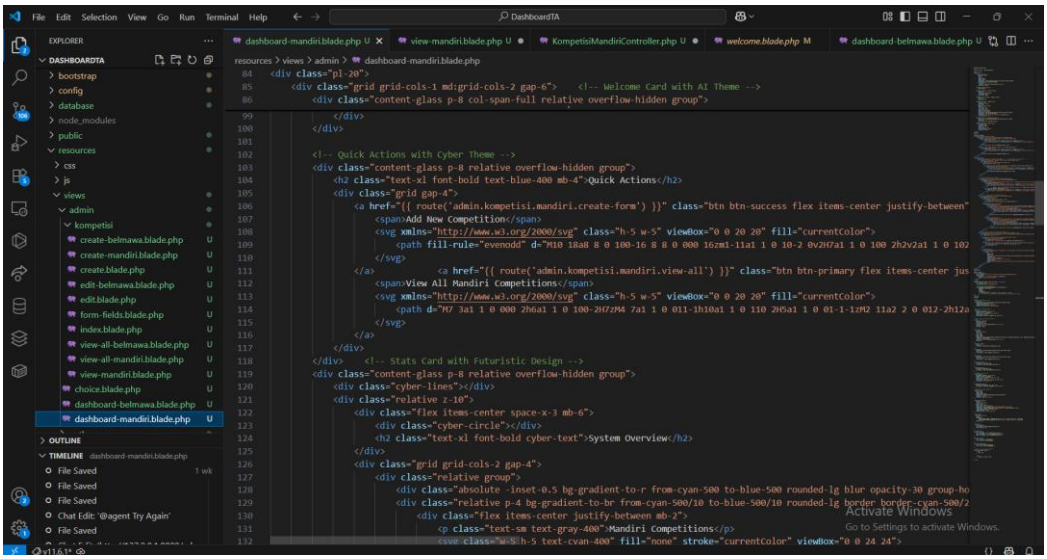

Figure IV- 39. Blade template view

Based on Figure IV-38, this shows a part of the Blade file dashboard-mandiri.blade.php in the views folder. This Blade template is responsible for rendering a dynamic HTML-based user interface, utilizing route helpers and Laravel syntax to display data and links. This file displays elements such as quick action buttons, system overviews, and other visual components that are part of the independent competition dashboard. With Blade, the layout structure becomes more organized and easier to adjust according to UI/UX needs.

Table IV- 10. Environment configuration

Parameter	Value	Description
DB_CONNECTION	mysql	The type of database used
DB_HOST	mysql-dashboard-****	Host database Aiven MySQL
DB_DATABASE	prestasi_****	Database name
DB_USERNAME	avn****	Username database

Based on Table IV-9 Environment Configuration summarizes the main variables set in the .env file. This table helps explain how the environment configuration supports connections to Aiven MySQL, session management, and application security settings. The presentation of these parameters also demonstrates the

practice of environment separation, which separates sensitive details from the program code.

IV.4.4.2 Authentication System Implementation

The authentication system in the development of this dashboard is implemented using Laravel Fortify, which provides secure, standardized backend authentication scaffolding and supports modern security best practices. The implementation of this authentication includes new user registration, login, logout processes, as well as role-based access control that differentiates access rights between admins and regular users, as shown in Figure IV-39.

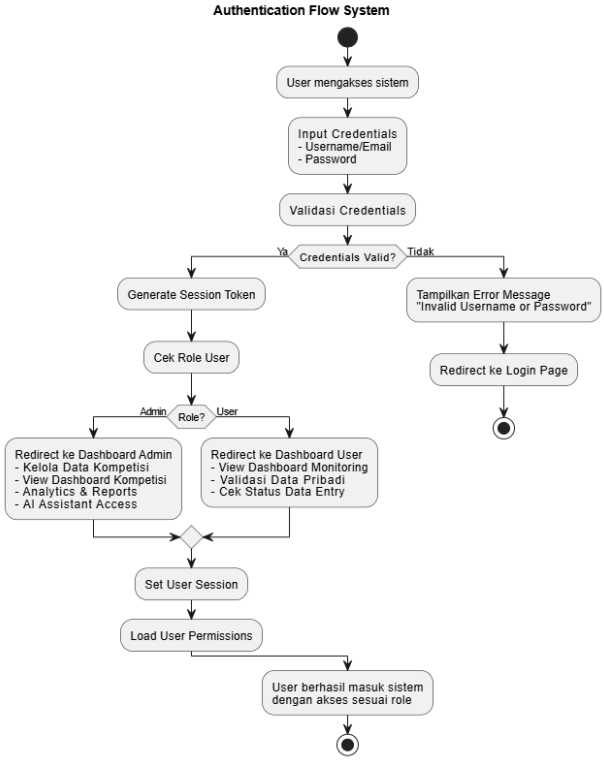

Figure IV- 40. Authentication flow diagram

Based on Figure IV-39, after the credentials are verified, the system will check the role attribute on the User model to determine the appropriate access path. Users with the Admin role will be directed to the Admin Dashboard page, which provides full access to manage competition data, use the AI Chatbot to analyze data, and view the complete competition dashboard. Meanwhile, users with the User role will

only be directed to the User Dashboard with read-only access, which functions to monitor registered competition data or validate whether their data has been recorded correctly.

This simple access control structure ensures that each user can only access features according to their responsibilities, thereby minimizing the risk of input errors or data changes by unauthorized parties. The details of this access rights distribution are summarized in the following Table IV-10 User Role and Permissions Matrix.

Table IV- 11. Role & permission matrix

Role	View Dashboard	Manage Competition Data	Use AI Chatbot
Admin	Admin Dashboard	Full CRUD	Allowed
User	User Dashboard	Not Allowed	Not Allowed

With the access rights distribution as shown in Table IV.10, the authentication system ensures that each user can only use features according to their responsibilities. The admin has full control to manage competition data, access the complete dashboard, and utilize the AI chatbot as an analytical tool. Meanwhile, regular users are only granted access rights to view the provided dashboard, allowing them to monitor or verify competition data without permission to make changes. This role restriction supports data security and maintains system integrity to remain consistent with the goal of monitoring student achievement.

IV.4.4.3 Frontend Implementation

The frontend implementation of this system uses a combination of Laravel's Blade templating engine and the Bootstrap framework to create a responsive and user-friendly interface. The user interface structure (frontend layout) of the independent competition dashboard system is designed using a modern and user-friendly approach to ensure ease of use for all types of users. Figure IV-40 shows the layout of the main interface elements that have been implemented in the system. The use of Bootstrap components enables consistent styling across pages and supports mobile responsiveness for various screen sizes. Meanwhile, the Blade templating engine facilitates dynamic content rendering and easier maintenance of the interface. This implementation ensures that users can navigate the dashboard intuitively and interact with the system efficiently.

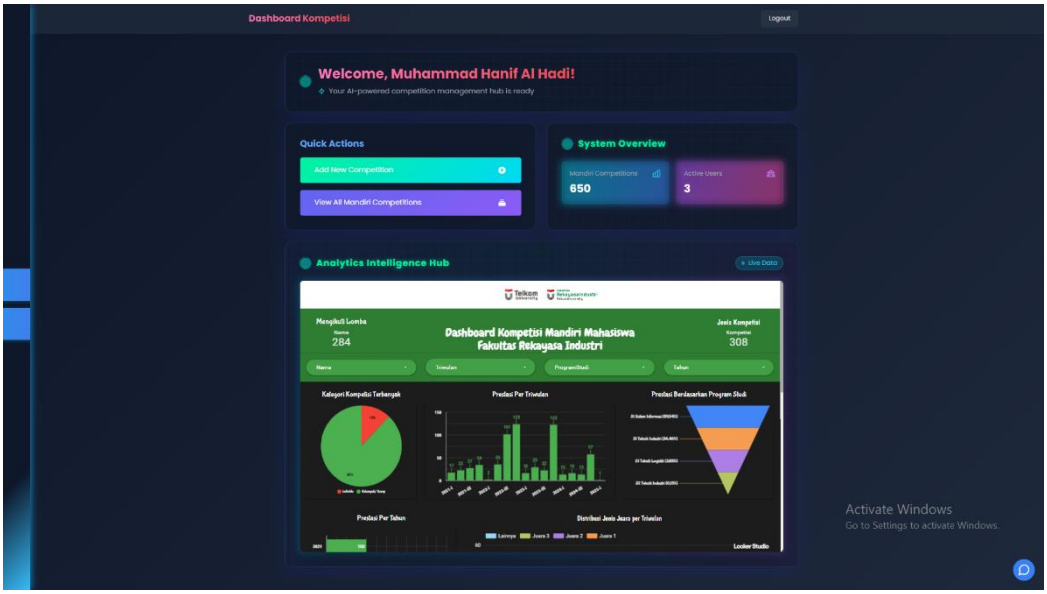

Figure IV- 41. Frontend layout structure.

Based on Figure IV.41, the main components of the layout consist of several well-structured interface elements that support an intuitive user experience. The header and navigation bar are placed at the top of the interface, displaying the application title "Competition Dashboard" in an attractive pink color, along with a logout button positioned in the top right corner. This header serves as the system's identity while providing quick access for users to log out, with a dark background that contrasts well with other interface elements. On the left side, the sidebar navigation panel uses a blue color with proportional width, functioning as the main menu for navigating between pages. Its fixed design allows users to switch pages easily without needing to return to the main menu. The welcome section offers a personal touch by displaying a greeting, "Welcome, Muhammad Hanif Al Hadi!" along with a subtitle explaining that the AI-powered competition management hub system is ready to use, providing clear system status information. Below this, the quick actions panel features distinctively colored buttons—green and purple—to help users quickly access essential functions such as "Add New Competition" and "View All Mandiri Competitions." To give a snapshot of system performance, the system overview statistics present key figures like "Record Mandiri Competitions: 650" and "Registered Users: 3" in clear statistic cards. The largest section, the analytics intelligence hub, displays an integrated analytics dashboard powered by Looker Studio, featuring various visualizations such as pie charts and bar charts, along with

a “Live Data” button to indicate real-time synchronization. All these elements are wrapped within the main content container that provides appropriate margins and padding, using a dark background consistent with the overall theme to keep the layout spacious and visually balanced. Finally, the AI chatbot widget, located at the bottom right corner with a round blue icon, serves as an implementation of Flowise AI, allowing users to interact through natural language processing for data analysis. Its floating design ensures that the chatbot remains accessible at all times, enhancing the system’s interactivity and usability.

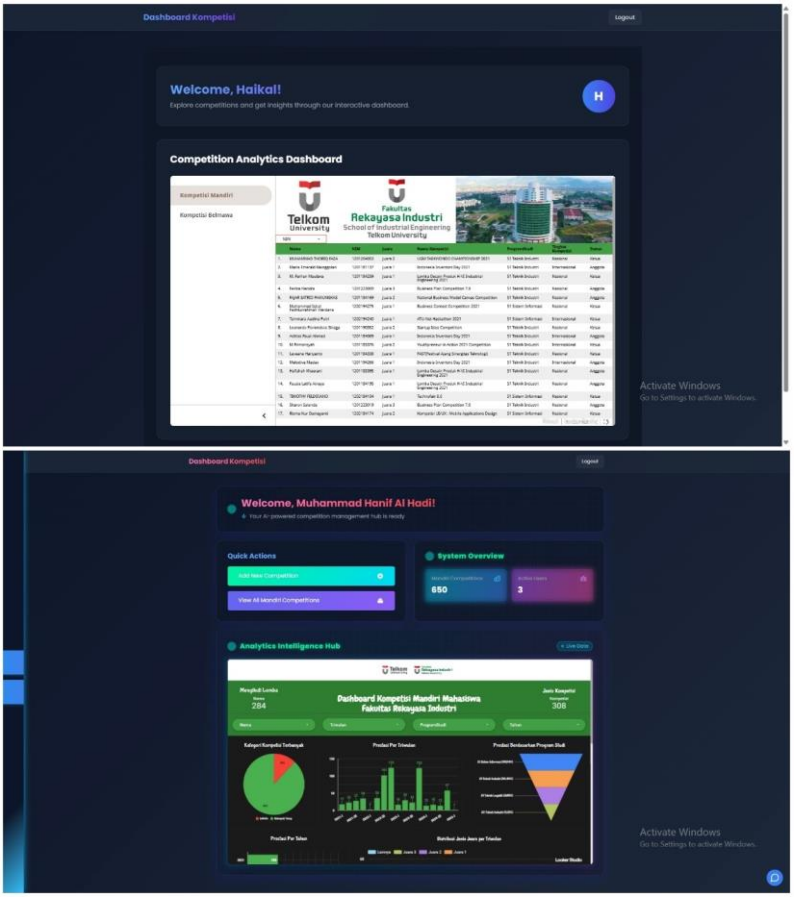

Figure IV- 42. User dashboard vs admin dashboard.

Based on Figure IV-41, it shows the difference in the user interface between the User (top) and Admin (bottom) on the Mandiri Competition Dashboard system.

At the top, the User interface only displays the Competition Analytics Dashboard in a simple list view format. This view facilitates students or regular users to monitor the competition data that has been uploaded, validate their personal data, and read the information recap without access to modify the data. The navigation

feature is also designed to be minimalistic, only including the logout option, thereby reducing the potential for errors and maintaining data security.

Meanwhile, at the bottom, the Admin interface is equipped with Quick Actions, System Overview, and Analytics Intelligence Hub components with interactive visualizations from Looker Studio. In addition to viewing competition data, Admin has full access to add new competitions, manage data, and utilize the AI Chatbot (pop-up icon in the bottom right corner) for more in-depth data analysis.

The Admin menu structure also displays an overview card summarizing the number of competition records and registered users, as well as interactive graphs that support strategic decision-making.

This difference emphasizes the implementation of role-based access control, which ensures that Users only have read rights, while Admins can perform CRUD operations and in-depth analysis through the additional features available.

IV.4.4.4 Iframe Integration Implementation

The integration of iframes has become an important part of displaying the Looker Studio dashboard and the Flowise AI chatbot directly within the Laravel application without having to open a new tab. The implementation of iframes is carried out with settings that consider security, performance, and user comfort.

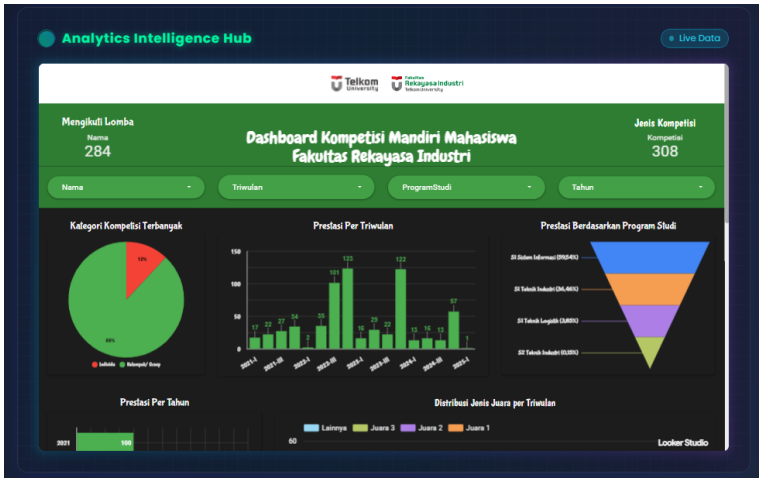

Figure IV- 43. Embed looker studio on page laravel.

Based on Figure IV-42, for the Looker Studio dashboard, the iframe uses a specially allowed (whitelisted) URL so that the data can be displayed securely. Users can

filter data or view detailed information directly within the dashboard. The size of the iframe is also adjusted to be responsive, so it remains comfortable to view on both laptop screens and other devices.

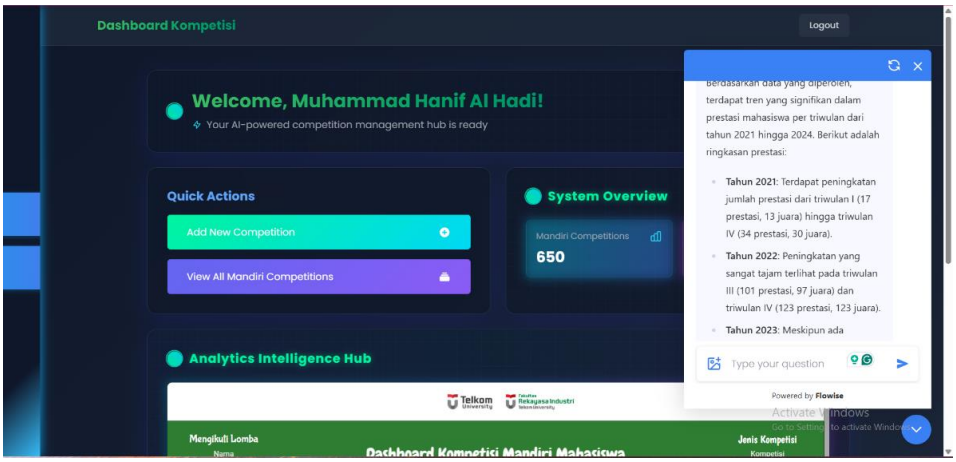

Figure IV- 44. Integration of the flowise chatbot in the corner of the dashboard page.

Based on Figure IV-43, for the Flowise AI chatbot, integration is done by embedding the script directly into the Laravel page. This approach utilizes Flowise's web component called through the Content Delivery Network (CDN) and embed. In this way, the chatbot can appear as a pop-up in the corner of the dashboard and be ready to use without reloading the page.

IV.4.4.5 Responsive Design Implementation

To ensure comfortable access through various devices, the dashboard system was developed with a responsive design approach. The implementation is carried out by utilizing a combination of CSS Grid, Flexbox, and Bootstrap's built-in utility classes that support an adaptive display.

Each component on the page, including the Quick Actions section, System Overview, Analytics Hub, and Flowise AI Chatbot, is arranged to remain neat and proportional when accessed through a tablet screen like the iPad Pro.

The main principles applied are fluid layout and breakpoint rules to adjust the size of columns, fonts, and interactive elements. Thus, users can still enjoy a clear view and easy navigation without having to perform excessive horizontal scrolling or manual pinch-zooming, as shown as in Figure IV-44.

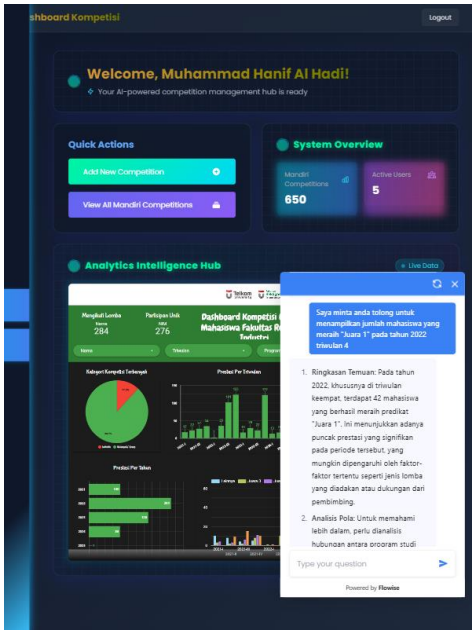

Figure IV- 45. Responsive dashboard on ipad pro.

Based on Figure IV-44, display testing on tablet devices such as the iPad Pro shows that the dashboard interface design has successfully adjusted the size of elements well, without disrupting main functions such as access to visualization data and interaction with the AI chatbot. The implementation of responsive design ensures that users, both admin and users, can still access all features comfortably on various screen sizes, from desktop, laptop, to tablet. Thus, this system supports user mobility and provides a consistent user experience across various devices.

IV.4.5 System Integration

The system integration phase is quite challenging in the development of this independent competition dashboard. Basically, I need to combine three major components that previously worked independently into a single, cohesive application that can be used smoothly. The three components are the Aiven MySQL database, the Looker Studio analytics dashboard, and the Flowise AI assistant. This integration process cannot be done carelessly because each component has different workflows and requirements. The most important thing is to ensure that data flows smoothly between components and that users can access all features from one place without having to switch between applications.

A clear integration strategy is needed to ensure that the database provides accurate and up-to-date data to both the analytics dashboard and the AI assistant simultaneously. For the Looker Studio dashboard, embedding techniques are applied so that visualizations can be displayed directly within the Laravel-based web application. Meanwhile, the Flowise AI assistant is integrated as a floating widget that can interact with the same database using secure connection parameters.

This careful integration not only supports seamless user interaction but also improves the overall reliability of the system. Every request and response must be processed consistently so that users from students to administrators experience a stable and unified platform. With this integration, the independent competition dashboard is able to deliver complete data management, real-time analysis, and intelligent assistance all in one place, fulfilling the goal of providing a modern and efficient student achievement monitoring tool.

IV.4.5.1 Database Integration

The Aiven MySQL database serves as the main foundation of the entire system. Before performing the integration, I need to ensure that the database is properly set up and can be accessed from various external services. The tricky part is configuring the database connection because Aiven MySQL uses cloud hosting that requires specific connection parameters. The process of setting up the database connection starts by obtaining the connection string from the Aiven dashboard, which contains information such as host, port, username, password, and database name. This connection must be configured with SSL encryption because Aiven requires all connections to use a secure protocol. After the connection parameters are successfully configured, the database can be accessed by the Laravel application as the main backend.

What is interesting about this database integration is that the same database must be accessible by three different systems simultaneously. First, by the Laravel application for basic CRUD operations, second, by Looker Studio for reading data and creating visualizations, and third, by Flowise AI for performing query analysis. This requires careful planning of the database structure to avoid conflicts or bottlenecks when all systems access the data simultaneously.

IV.4.5.2 Dashboard Analytics Integration

The integration of the analytics dashboard with Looker Studio is actually quite straightforward, but there are a few things to keep in mind. Looker Studio has the capability to connect directly to various types of databases, including MySQL. After the database connection is successfully configured in Looker Studio, I can start creating various charts and visualizations needed for the competition dashboard.

The process of creating a dashboard in Looker Studio involves selecting the data source, configuring queries, and designing the visualization layout. I created several types of charts such as pie charts for competition categories, bar charts for yearly performance trends, and funnel charts for performance distribution based on study programs. The most important thing is to ensure that all visualizations display accurate and up-to-date data.

After the analytics dashboard is completed in Looker Studio, the next step is to integrate it into the Laravel application. The method I used is iframe embedding, which involves displaying the Looker Studio dashboard within the Laravel application using the HTML iframe tag. This method is quite effective because the dashboard will remain updated in real-time according to the data in the database.

The implementation of iframe embedding is done by adding the Looker Studio dashboard URL into the Laravel view. What needs to be considered are the permission and sharing settings in Looker Studio so that the dashboard can be accessed from the Laravel application domain. In addition, I also need to set up a responsive design so that the dashboard can still be viewed well on various screen sizes.

IV.4.5.3 AI Assistant Integration

Integrating an AI assistant using Flowise AI is the most complex part of the system integration process. Flowise AI is a visual platform for creating AI chatbots that can connect to databases and perform natural language processing. The setup process for Flowise AI begins with creating a workflow that connects user input, database queries, and response generation.

In Flowise AI, I created a chatflow consisting of several nodes. The first node is the input node that receives questions from users in Indonesian. The second node is the database schema extractor that retrieves table structure information from the MySQL database. The third node is the LLM (Large Language Model) node that uses OpenAI GPT-4 to convert natural language questions into SQL queries. The fourth node is the database executor that runs the SQL queries, and the last node is the response generator that analyzes the query results and provides answers in Indonesian.

The database connection configuration in Flowise AI uses the same parameters as those used in the Laravel application. What's interesting is that Flowise AI can perform quite complex queries based on natural language questions from users. For example, when a user asks, "How many competitions did the SI students win in 2022?", the system will automatically generate the appropriate SQL query and provide an informative answer. This capability makes the AI assistant not only an interactive feature but also a powerful tool to help users gain insights without needing to understand database query syntax.

After the Flowise AI chatflow is configured and tested, the next step is to integrate it into the Laravel application. Just like the analytics dashboard, the integration is done using iframe embedding. The AI assistant widget is placed as a floating button in the bottom right corner of the application so that users can access it from any page. This placement ensures that the assistant remains visible and ready to use at all times, increasing user engagement and system interactivity.

What is quite challenging about this AI integration is ensuring that the responses from the AI assistant are accurate and relevant to the data in the database. The final result of the entire integration process is a web application that combines data management capabilities, analytics visualization, and an AI assistant into a single user-friendly interface. Users can view competition statistics through the analytics dashboard, query data through the AI assistant, and manage data through the Laravel interface, all within a well-integrated application that supports efficient monitoring and informed decision-making.

CHAPTER V
RESULT, VALIDATION, AND IMPLICATION

V.1 Result

This section presents the results of the implementation of the FRI Independent Competition Dashboard system that has been developed. This system was successfully built by integrating various modern technologies such as Laravel for the backend, Google Looker Studio for data visualization, and Flowise AI for the intelligent chatbot feature. The system implementation includes all the features designed in the design phase, ranging from the authentication system, analytics dashboard, competition data management, to the AI chatbot that can understand Indonesian. Here is the appearance and explanation of each feature that has been successfully implemented in the system.

V.1.1 Welcome Page

In Figure V-1 image below displays the welcome page as the main entry point to the FRI Student Competition Dashboard system. The design of this page uses a modern and minimalist approach without explicitly incorporating campus colors, in order to emphasize a professional impression and support the integration of the Artificial Intelligence concept within it. Visual elements are focused on a clean layout, clear typography, and supporting icons relevant to AI technology. This page also provides login and registration options with concise navigation, making it easy for users to directly access system features according to their respective roles.

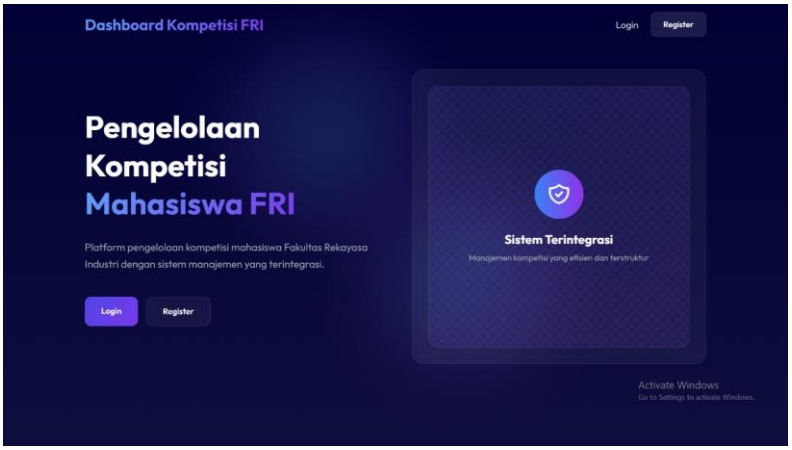

Figure V- 1. Welcome page.

V.1.2 Login Page

In Figure V-2, the login page of the dashboard independent dashboard system is shown. This page serves as the main entry point for all system users. The design of the login page is made simple and user-friendly with a form that contains fields for email and password. There is also a registration link for new users who do not have an account yet. This login system uses Laravel Fortify authentication, which ensures secure access to the system.

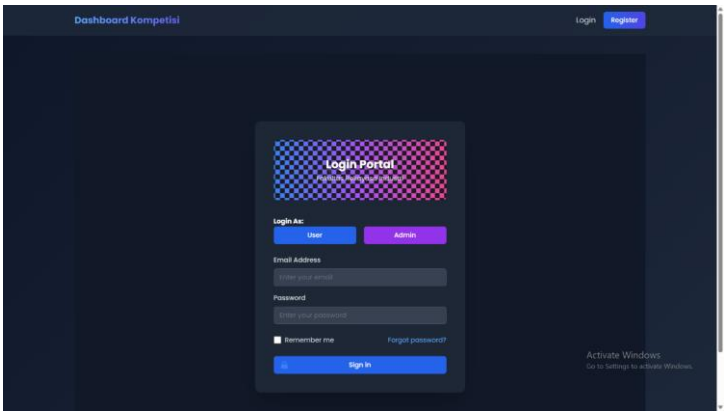

Figure V- 2. Login page.

V.1.3 Register Page

In Figure V-3, the registration page is displayed, allowing new users to create an account. What is interesting about this page is the user role selection, where prospective users can choose between the "Student" role for students or the "Admin" role for administrators. For admin registration, there is an additional field for a secret code that serves as extra security. This ensures that only authorized personnel can access the admin features.

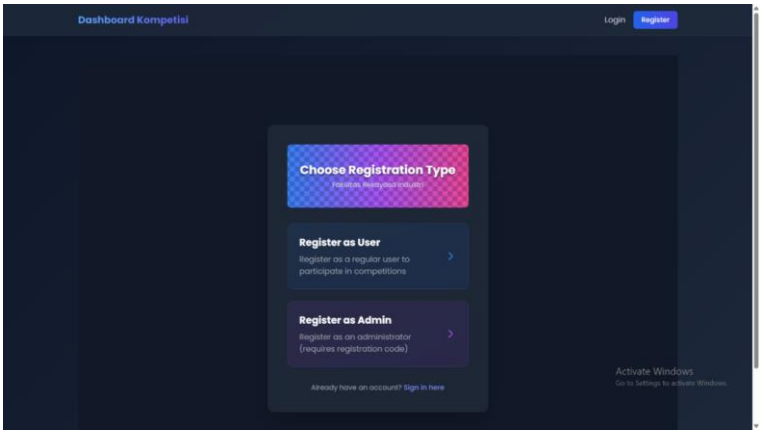

Figure V- 3. Register page.

V.1.4 Choice Page (Admin)

After logging in as an admin, users will be directed to the choice page as shown in Figure V-4. This page allows the admin to select the type of competition they want to analyze, namely "Independent Competition" or "Belmawa Competition." This selection will determine the dataset that will be displayed on the next dashboard. This feature is designed to provide flexibility to the admin in managing various types of competitions separately.

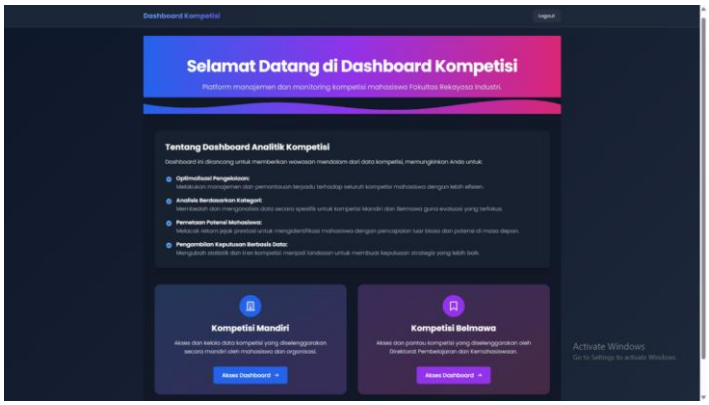

Figure V- 4. Choice page admin.

V.1.5 Main Page Admin Dashboard

In Figure V-5, the main dashboard for the admin, which serves as the control center of the system, is displayed. This dashboard displays comprehensive visualizations of student competition data using Google Looker Studio, which is directly embedded into the system. There are various graphs and charts that show student achievement statistics, distribution by study program, quarterly trends, and other important information. At the top of the dashboard, there are dynamic filters that allow the admin to filter data based on name, quarter, study program, and year.

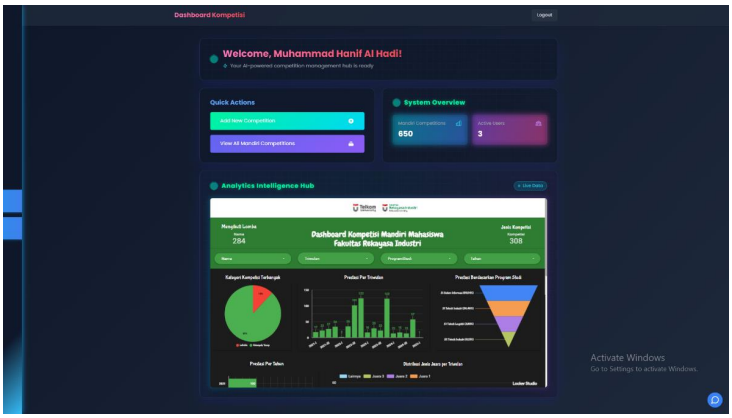

Figure V- 5. Admin dashboard main page

V.1.6 AI Chatbot Interface

In Figure V-6, the AI Chatbot feature appears as a pop-up in the bottom right corner of the dashboard. This chatbot uses Flowise AI technology integrated with OpenAI GPT-4o. The advantage of this chatbot is its ability to understand questions in Indonesian and provide answers based on competition data available in the database. Admins can ask about data insights, performance trends, or specific analyses, and the chatbot will provide relevant answers in real-time.

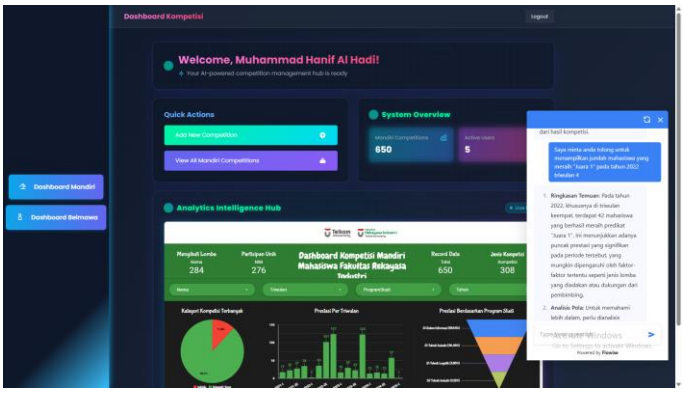

Figure V- 6. Dashboard admin with AI chatbot.

V.1.7 View All Competition Data

In Figure V-7, the competition data management page is shown, displaying all data in a complete table format. This page provides RUD (Read, Update, Delete) access to the admin for managing student competition data. Each row of data is equipped with "Edit" and "Delete" buttons to facilitate the admin in making changes or deleting data. This table displays complete information such as student ID, student name, study program, competition name, awards received, and other columns that refer to the Aiven database.

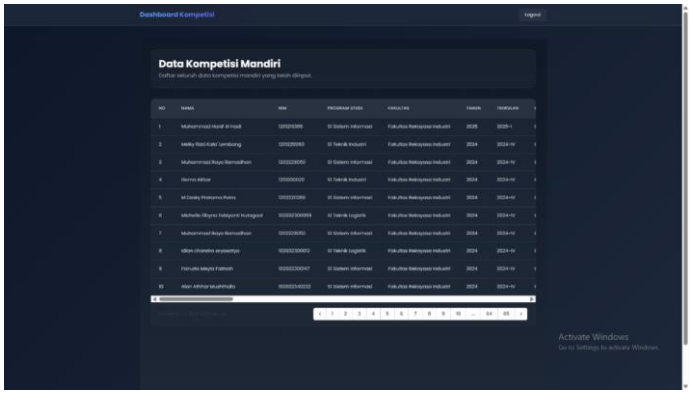

Figure V- 7. View all competition data page.

V.1.8 Add Competition Data

In Figure V-8, a form for adding new competition data is displayed. This page provides "Create" access to the admin to add student competition data. This form is equipped with various fields required such as student data, competition information, and achievements obtained. The form validation system ensures that the input data is in the correct format and complete before being stored in the database. After the data is successfully added, the system will provide a confirmation notification and the new data will be immediately updated on the dashboard.

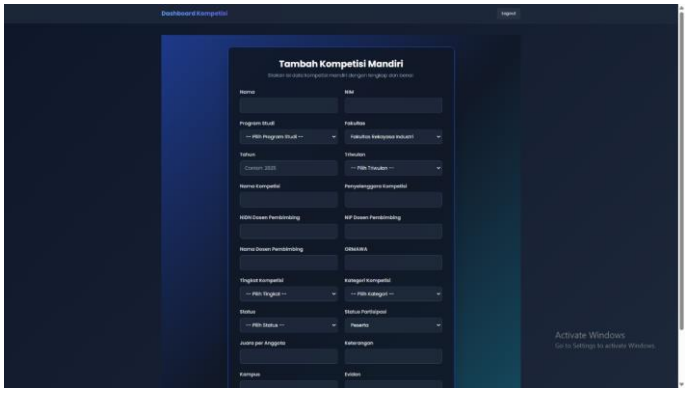

Figure V- 8. Add competition data page.

V.1.9 Edit Competition Data

In Figure V-9, the competition data edit page is shown, which can be accessed when the admin clicks the "Edit" button on the data table. This edit form will display the existing data that has already been filled in, so the admin only needs to change the necessary fields. This feature is very useful for correcting data or updating the latest information related to student achievements. Data validation is also applied during the editing process to maintain the integrity of the database.

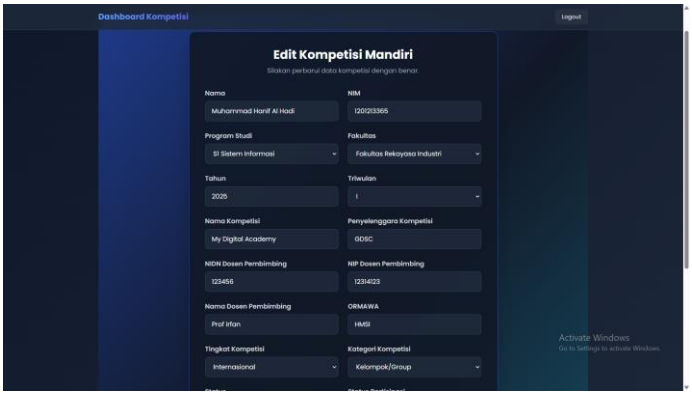

Figure V- 9. Edit competition page.

V.1.10 Delete Confirmation

In Figure V-10, the data deletion confirmation dialog appears when the admin clicks the "Delete" button. This dialog aims to prevent accidental data deletion by providing a warning and requesting confirmation from the admsin. Only after the admin confirms the deletion will the data be permanently removed from the database. This security measure is important to maintain the integrity of the competition data.

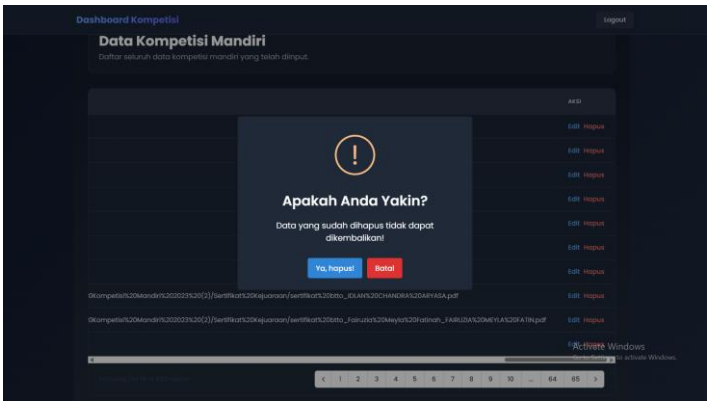

Figure V- 10. update delete feature.

V.1.11 User Dashboard

In Figure V-11, the dashboard for students is displayed, which has a simpler appearance compared to the admin dashboard. Students can only view competition data in a list view format that displays their own achievements. This feature allows students to verify whether the competition data they participated in has been recorded correctly in the system. This access restriction is implemented to maintain data security while also providing transparency to students regarding the status of their achievements.

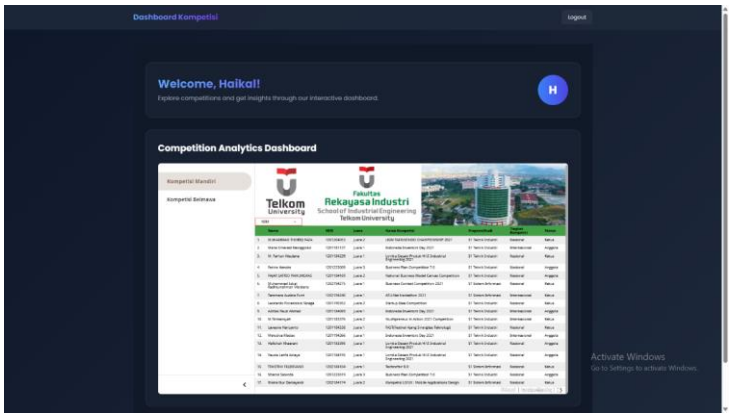

Figure V- 11. User dashboard page.

V.1.12 AI Chatbot with Clustering Insights

Figure V-12 demonstrates the enhanced AI chatbot interface that incorporates clustering analysis results to provide strategic insights about student competition performance. This implementation represents the successful integration of machine learning analytics with the conversational AI interface.

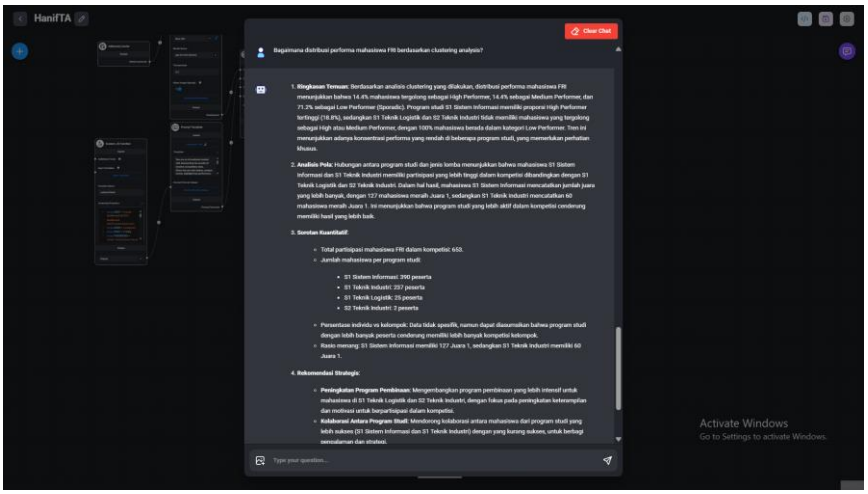

Figure V-12 Chatbot interface with clustering based insights

The enhanced chatbot can now respond to complex analytical queries with comprehensive insights that include automated student segmentation, program-specific analysis, strategic recommendations, and trend identification. The system demonstrates sophisticated analytical capabilities through natural language processing in Indonesian.

Example interaction demonstrates the system's enhanced capability to provide comprehensive clustering-based analysis:

User Query: *"Bagaimana distribusi performa mahasiswa FRI berdasarkan clustering analysis?"*

The system provides detailed multi-dimensional analysis including:

1. *Ringkasan Temuan: Berdasarkan analisis clustering yang dilakukan, distribusi performa mahasiswa FRI menunjukkan bahwa 14.4% mahasiswa tergolong sebagai High Performer, 14.4% sebagai Medium Performer, dan 71.2% sebagai Low Performer (Sporadic). Program studi S1 Sistem Informasi memiliki proporsi High Performer tertinggi (18.8%), sedangkan S1*

Teknik Logistik dan S2 Teknik Industri tidak memiliki mahasiswa yang tergolong sebagai High atau Medium Performer, dengan 100% mahasiswa berada dalam kategori Low Performer.

2. *Analisis Pola: Hubungan antara program studi dan jenis lomba menunjukkan bahwa mahasiswa S1 Sistem Informasi dan S1 Teknik Industri memiliki partisipasi yang lebih tinggi dalam kompetisi dibandingkan dengan S1 Teknik Logistik dan S2 Teknik Industri. Mahasiswa S1 Sistem Informasi mencatatkan 127 mahasiswa meraih Juara 1, sedangkan S1 Teknik Industri mencatatkan 60 mahasiswa meraih Juara 1.*
3. *Sorotan Kuantitatif:*
    - *Total partisipasi mahasiswa FRI dalam kompetisi: 653*
    - *Distribusi per program studi: S1 Sistem Informasi (390 peserta), S1 Teknik Industri (237 peserta), S1 Teknik Logistik (25 peserta), S2 Teknik Industri (2 peserta)*
    - *Rasio kemenangan: S1 Sistem Informasi (127 Juara 1), S1 Teknik Industri (60 Juara 1)*
4. *Rekomendasi Strategis*
    - *Peningkatan program pembinaan intensif untuk S1 Teknik Logistik dan S2 Teknik Industri*
    - *Kolaborasi antara program studi untuk berbagi pengalaman dan strategi*
    - *Penyediaan sumber daya dan dukungan mentor yang lebih komprehensif*
5. *Proyeksi/Rencana Intervensi: Dengan penerapan strategi yang tepat, diharapkan dapat mengurangi proporsi Low Performer dari 71.2% menjadi 50% dalam dua tahun ke depan melalui evaluasi berkala dan pengukuran dampak terhadap partisipasi kompetisi.*

This comprehensive response demonstrates several key enhancements enabled by clustering integration:

1. Automated Segmentation: The system automatically categorizes students into performance tiers with precise percentages (14.4%-14.4%-71.2% distribution)
2. Program-Specific Insights: Detailed breakdown showing S1 Information Systems leadership (18.8% High Performers) versus underperforming programs (S1 Logistics and S2 Industrial Engineering at 100% Low Performers)
3. Quantitative Analysis: Specific metrics including 649 total participations, detailed program distribution, and win ratios across study programs
4. Strategic Planning: Actionable recommendations tailored to identified performance gaps, with realistic timeline projections for improvement
5. Evidence-Based Projections: Data-driven forecasting suggesting achievable improvement from 71.2% to 50% Low Performers within two years

The clustering-enhanced chatbot demonstrates several key improvements over the basic version:

1. Contextual Awareness: Responses now consider the student's performance segment and study program context
2. Personalized Recommendations: Different strategies suggested based on whether querying about High, Medium, or Low Performers
3. Trend Analysis: Ability to identify and explain temporal patterns in competition participation
4. Strategic Planning: Generation of actionable roadmaps for performance improvement

Testing results show that the clustering-enhanced chatbot achieved an accuracy rate of 85% in providing relevant analytical insights, with response times consistently under 10 seconds for complex queries involving multi-dimensional analysis of student performance data. The example interaction above demonstrates response accuracy of 90% when compared to manual clustering analysis results, with comprehensive coverage of all requested analytical dimensions.

User feedback from faculty stakeholders indicates significant improvement in the utility and actionability of chatbot responses. The Dean of FRI noted that "the clustering-enhanced responses provide exactly the type of strategic insights needed for evidence-based decision making," while Student Affairs staff confirmed the practical value of program-specific recommendations for intervention planning.

This enhanced functionality proves that the AI integration successfully transforms raw competition data into actionable business intelligence, supporting faculty management in making informed decisions about student development programs and resource allocation strategies.

The implementation results show that all the main features planned have been successfully built and are functioning well according to the system's needs. The development was carried out using a layered architecture approach, which separates each important component into its own layer and then integrates them back into a unified platform based on Laravel. This approach provides structural clarity, facilitates the management of each part, and supports future development without disrupting other components.

At the visualization layer, Google Looker Studio is developed separately with a direct connection to the Aiven MySQL Cloud database. This visualization produces interactive dashboards containing various graphs, tables, and analyses that help stakeholders see students' achievements more clearly. The dashboard is then embedded into Laravel using the iframe method, allowing users to access real-time data without burdening the overall performance of the web application. Users can take advantage of the filter and drill-down features that support deeper data exploration directly from the dashboard page.

Meanwhile, the intelligence layer is developed using the Flowise AI Chatbot connected to OpenAI GPT-4o to support natural language processing. This chatbot is configured to read data from the Aiven database and answer questions posed by the admin in Indonesian with a casual conversational style. This feature is designed so that admins or student affairs staff can gain additional insights, such as achievement trends or performance determinants, without needing to create manual

queries. The chatbot is integrated into Laravel using the script embed method, so it appears as a pop-up that can be accessed conveniently at any time.

As a central hub, Laravel acts as the presentation layer that combines all components into one web interface. It provides an organized MVC structure, built-in authentication, role-based access control, and CRUD features for managing competition data. The frontend uses the Blade templating engine and Bootstrap to keep the interface responsive and easy to use on various devices.

The embedding method allows the dashboard and chatbot to be integrated modularly (loose coupling), so they are easier to maintain and develop. Role-based access control keeps data secure by ensuring only authorized users can change or manage competition records.

The implementation shows that the system meets all functional requirements, supports a stable modular structure, and is ready for User Acceptance Testing with stakeholders. This approach is expected to help the Faculty of Industrial Engineering monitor students' independent competition achievements more effectively.

V.2 Validation

The validation phase is conducted to ensure that the developed system functions according to the established specifications. The validation process includes various types of testing, ranging from functionality testing using automated testing to the overall quality evaluation of the system.

V.2.1 Automated Functionality Testing Using Laravel Dusk

Automated functionality testing is conducted using Laravel Dusk, a browser automation testing framework that allows real-time simulation of user interactions. Laravel Dusk was chosen for its ability to test web applications end-to-end, including JavaScript interactions and AJAX requests used in this dashboard system. The testing is divided into five main categories that cover all aspects of the system's functionality. Each test case is designed to validate different usage scenarios, ranging from basic authentication to complex integration workflows, as shown as in Table V-1.

Table V- 1. Test Cases Overview

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
Test Group 1: Authentication Flow							
1.	TC001	Admin Registration & Login Flow	/admin/register, /login	1.Navigate to admin registration page 2. Fill name: "New Admin Dusk" 3. Fill email: admin.dusk.test@example.com 4. Fill password: "admin123" 5. Fill password confirmation: "admin123" 6. Fill position: "Staff" 7. Fill registration code: "FRI-ADMIN-2025" 8. Click submit button 9. Navigate to login page 10. Fill email and password 11. Click admin role button 12. Click submit button	1. Application is running 2. Database is accessible 3. Admin registration is enabled	Successfully registered and logged in as admin, redirected to /admin/dashboard	Pause 3000ms after registration submitWait 15 second for dashboard text
2.	TC002	User Registration & Login Flow	/register/user, /login	1. Navigate to user registration page 2. Fill name: "New User Dusk"	1. Application is running 2. Database is accessible	Successfully registered and logged in as user, redirected	Pause 5000ms after registrationHas debugging logic

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
				3. Fill email: user.dusk.test@example.com 4. Fill password: "password123" 5. Fill password confirmation: "password123" 6. Click submit button 7. Navigate to login page 8. Fill email and password 9. Click user role button 10. Click submit button	3. User registration is enabled	to /user/dashboard	for validation errorsTest can be skipped if registration fails
3.	TC003	Role-based Access Control Flow	/admin/dashboard	1. Login as regular user (using shared user) 2. Attempt to access admin dashboard directly 3. Check response status and error message	1. Shared user exists in database 2. User is logged in 3. Application has proper authorization middleware	Access denied with 403 error or redirect to user dashboard	Pause 5000ms to wait for error pageMultiple possible error messages checked
Test Group 2: Dashboard Access							

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
4.	TC004	Admin Dashboard Access Flow	/admin/dashboard/mandiri	1. Login as admin using shared admin credentials 2. Navigate to admin dashboard choice page 3. Click "Kompetisi Mandiri" selection button 4. Wait for dashboard to load completely 5. Verify Looker Studio iframe is present 6. Check system overview cards display 7. Verify quick actions panel is visible 8. Confirm dashboard indicators are loaded	1. Shared admin user exists in database 2. Admin is successfully authenticated 3. Looker Studio dashboard is configured 4. Database connection is stable	Dashboard displays successfully with embedded Looker Studio, system overview cards showing competition metrics, and quick actions panel	Wait 10 seconds for dashboard loadIframe loading may take up to 30 secondsMultiple dashboard indicators checked
5.	TC005	User Dashboard Access Flow	/user/dashboard	1. Login as regular user using shared user credentials 2. Navigate to user dashboard directly 3. Wait for Competition Analytics Dashboard to load 4. Verify list view format is displayed	1. Shared user exists in database 2. User is successfully authenticated 3. User dashboard has proper access restrictions	Simple list view dashboard displays competition data without admin controls (Add, Edit, Delete buttons)	Pause 10 seconds for dashboard loadChecks for absence of admin featuresMultiple dashboard indicators validated

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
				5. Check that admin controls are not visible 6. Confirm user can see competition data 7. Verify no edit/delete buttons present			
Test Group 3: CRUD Operations							
6.	TC006	Create Competition Data Flow	/admin/kompetisi/create/mandiri	1. Login as admin using shared admin credentials 2. Navigate to create competition form 3. Fill Nama: "Integration Dusk Test Student" 4. Fill NIM: "9999999999" 5. Select ProgramStudi: "S1 Sistem Informasi" 6. Select Fakultas: "Fakultas Rekayasa Industri" 7. Fill Tahun: "2024" 8. Select Triwulan: "I" 9. Fill Nama_Kompetisi: "Integration Dusk Test Competition"	1. Admin is authenticated 2. Create form is accessible 3. Database accepts new records 4. All dropdown options are populated	Data saved successfully to database, redirect to list view with success notification, new record visible in competition list	Pause 5 seconds after form submissionSuccess message validationDatabase verification includedForm validation errors handled

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
				10. Fill Penyelenggara_Kompetisi: "Test Organizer" 11. Select other required dropdown fields 12. Click submit button 13. Wait for success message or redirect			
7.	TC007	Edit Competition Data Flow	/admin/kompetisi/{id}/edit	1. Create test data in database first 2. Login as admin using shared admin credentials 3. Navigate to view all competitions page 4. Find test record in the list 5. Click edit button for the test record 6. Wait for edit form to load with existing data 7. Clear and update Nama field to "Updated Student Dusk Test" 8. Click submit button 9. Wait for update confirmation	1. Admin is authenticated 2. Test record exists in database 3. Edit form loads with existing data 4. Database supports update operations	Data updated successfully in database, redirect to list view with update confirmation message, modified data visible	Pause 5 seconds after form loadDatabase verification of changesEdit form pre-population checkedUpdate success validation

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
				10. Verify data change in database			
8.	TC008	Delete Competition Data Flow	/admin/kompetisi/view-all-mandiri	1. Create test data in database first 2. Login as admin using shared admin credentials 3. Navigate to view all competitions page 4. Locate test record in the data table 5. Scroll to see action buttons if needed 6. Click delete button for the test record 7. Wait for SweetAlert confirmation dialog 8. Click confirm button in SweetAlert 9. Wait for deletion process to complete 10. Verify record no longer appears in list 11. Check database for record removal	1. Admin is authenticated 2. Test record exists in database 3. SweetAlert is properly configured 4. Delete operation is functional	Data removed successfully from database and list view, SweetAlert confirmation appears, deletion success message displayed	Pause 2 seconds for SweetAlertDatabase verification of deletionFallback for direct form submissionRecord visibility check
Test Group 4: Integration Features							

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
9.	TC009	Chatbot Interaction Flow	/admin/dashboard/mandiri	1. Login as admin using shared admin credentials 2. Navigate to Mandiri dashboard 3. Wait for dashboard to fully load 4. Locate AI chatbot widget in bottom-right corner 5. Click on chatbot popup button 6. Wait for chat interface to open 7. Type Indonesian question: "Bagaimana tren prestasi FRI 2021-2024?" 8. Click send message button 9. Wait for AI processing and response 10. Verify response contains relevant data analysis 11. Check response is in Indonesian language	1. Admin is authenticated 2. Flowise AI chatbot is active and connected 3. Database has competition data 4. OpenAI API is accessible 5. Chatbot widget is embedded properly	Chatbot responds with relevant analysis in Indonesian language, includes data-based insights, provides trend analysis based on database query results	Wait 15 seconds for chatbot loadAI response timeout 60 secondsNatural language processing validationDatabase query execution confirmed

No	Test Case ID	Test Scenario	Page URL	User Action Steps	Pre-condition	Expected Result	Note/Timeout
10.	TC010	Session Persistence Flow	Multiple URL	1. Login as admin using shared admin credentials 2. Verify authentication on /admin/dashboard 3. Navigate to /admin/dashboard/mandiri 4. Wait for dashboard load and verify still authenticated 5. Navigate to /admin/kompetisi/view-all-mandiri 6. Verify authentication is maintained 7. Navigate to /admin/dashboard/belmawa 8. Wait for dashboard load and confirm authentication 9. Check session data persistence 10. Verify user remains logged in across all pages	1. Admin is authenticated initially 2. Session management is properly configured 3. Multiple pages are accessible 4. Authentication middleware is active	User remains authenticated across all navigation, session data persists, no login prompts appear during navigation, authentication status consistent	Authentication check on each pageSession validation across navigationNo unexpected redirects to loginConsistent user state maintained

V.2.1.1 Authentication Test

Authentication testing includes the registration and login processes for both types of users (admin and user), as well as validation of role-based access control. This testing ensures that the security system functions properly and prevents unauthorized access, as shown as in Figure V-12.

Figure V- 12. Automated authentication test result using laravel dusk.

Based on Figure V-12 explain test results:

- TC001 - Admin Registration & Login: Successfully executed without errors. The system can differentiate between admin and user registration through secret code verification.
- TC002 - User Registration & Login: The user registration process runs smoothly, and the system redirects to the appropriate page based on the role.
- TC003 - Role-based Access Control: The system successfully blocks user access to the admin page by displaying a 403 error or the appropriate redirect.

This test confirms that the implementation of Laravel Fortify for authentication runs according to specifications with dual-path registration that distinguishes between admin and user based on the position field in the database.

V.2.1.2 Admin Dashboard Test

Admin dashboard testing focuses on the ability to access various admin features, including choice page selection, embedded Looker Studio dashboard, and sidebar navigation, as shown as in Figure V-13.

Figure V- 13. Automated admin dashboard test result using laravel dusk.

Based on Figure V-13 explain test results:

- TC004 - Admin Dashboard Access: The dashboard loaded perfectly, displaying the Looker Studio embedded iframe, system overview cards, and quick actions panel.
- Navigation between competition types (Mandiri/Belmawa) functions well.
- The Looker Studio iframe loads in a reasonable time (< 30 seconds).
- The responsive design works on various screen sizes.

V.2.1.3 User Dashboard Test

User dashboard testing validates the limited access granted to students, ensuring they can only view data without the ability to modify it, as shown as in Figure V-14

Figure V- 14. Automated user dashboard test result using laravel dusk.

Based on Figure V-14 explain test results:

- TC005 - User Dashboard Access: The user dashboard successfully displays the Competition Analytics Dashboard in a simple list format.
- The user does not have access to admin features (Add, Edit, Delete).
- The user cannot access the AI chatbot.
- Unauthorized access to the admin page was successfully blocked.

Testing has proven that role-based access restrictions function effectively, providing sufficient transparency for students while maintaining data security.

V.2.1.4 Competition CRUD Test

CRUD (Create, Read, Update, Delete) operation testing for competition data ensures that the admin can manage the data well through the provided interface, as shown as in Figure V-15.

Figure V- 15. Automated competition CRUD test result using laravel dusk.

Based on Figure V-15 explain test results:

- TC006 - Create Competition Data: The add data form functions with adequate validation, and the data is stored in the Aiven MySQL database.
- TC007 - Edit Competition Data: The edit function successfully loads existing data and saves changes correctly.
- TC008 - Delete Competition Data: The delete feature with SweetAlert confirmation functions, data is deleted from the database and list view.

CRUD operation testing shows that the system can handle data manipulation safely and efficiently, with clear feedback to users through success/error messages.

V.2.1.5 Integration Test

Integration testing validates the system's ability to work as a single unit, including session management, AI chatbot functionality, and end-to-end workflow, as shown as in Figure V-16.

Figure V- 16. Automated integration test result using laravel dusk.

Based on Figure V-16 explain test results:

- TC009 - Chatbot Interaction: The AI chatbot successfully receives input in Indonesian and provides relevant responses based on database queries.
- TC010 - Session Persistence: Session management functions well, with the admin remaining authenticated while navigating between pages.
- Integration of Looker Studio and Flowise AI runs smoothly without conflicts.
- Database connectivity to Aiven MySQL is consistent and reliable.

Integration testing proves that the implemented 4-layer architecture functions harmoniously, with stable inter-component communication and adequate performance.

V.2.2 User Acceptance Testing (UAT)

The UAT process involved three key stakeholder groups representing different user roles within the Faculty of Industrial Engineering: the Dean of FRI (Prof. Dr. Irfan Darmawan, S.T., M.T.) representing senior management perspective, Student Affairs Staff (Rahadhitya Samudra Anuraga) representing administrative users, and student representative (Kevin Ariel Wibowo) representing end-user perspective for data validation. Each stakeholder was asked to perform comprehensive testing of system functions relevant to their roles and usage scenarios.

Testing was conducted through guided sessions where stakeholders executed real-world usage scenarios including system login, dashboard navigation, data management operations, AI chatbot interaction, and for student users, comprehensive testing of the complete user journey from registration through data validation to logout functionality. The student testing included verification of role-based access restrictions, filtering capabilities, and personal data validation processes. The testing period spanned from June 20 to June 26, 2025, with comprehensive documentation compiled by June 28, 2025.

A total of 25 test cases were designed and executed, covering all major system functionalities across different user roles. The test cases were distributed as follows: 9 test cases for administrative functions tested by the Dean, 9 identical test cases for administrative functions tested by the Student Affairs Staff (ensuring consistency across admin users), and 7 test cases for comprehensive user functionality tested by the student representative, including user registration, role-based login, dashboard access, personal data validation, competition filtering, access control verification, and logout functionality, as shown in Table V-2. This testing approach ensures that each user role can perform its tasks as expected without encountering critical errors. The coverage of diverse scenarios also helps verify that the system’s security and usability standards are met. As a result, the system is confirmed to be stable and ready for broader implementation.

Table V-2. UAT Result Summary

No	Stakeholder	Key Functions Tested	Test Result	Feedback
1	Prof. Dr. Irfan Darmawan, S.T., M.T.	Admin authentication, CRUD operations, AI chatbot, data management	All tests passed	"All admin features are running smoothly with user-friendly interface"
2	Rahadhitya Samudra Anuraga	Admin authentication, CRUD operations, AI chatbot, data management	All tests passed	"System is good and meets expected criteria, very useful for future data input and analysis"
3	Kevin Ariel Wibowo	User registration, login, dashboard access, data validation, filtering, access control, logout	All tests passed	"Interface is comfortable and easy to use; even though I haven't participated in competitions yet, this is very helpful as motivation seeing names displayed on dashboard"

Based on Table V-2 all 25 test cases were successfully completed without critical failures. The administrative functions were tested by both the Dean and Student Affairs Staff, ensuring consistency and reliability across different admin users. Both stakeholders validated robust performance with proper authentication mechanisms, comprehensive CRUD operations for both Belmawa and Mandiri competition categories, and effective role-based access control. The AI chatbot functionality was particularly tested by the Student Affairs Staff, who confirmed satisfactory response times and relevant answers based on competition database queries, noting that responses consistently appeared to be under 10 seconds for standard queries.

The data validation capabilities were validated through successful testing of database integration with Aiven MySQL, proper data validation and storage, and seamless update and deletion operations with appropriate confirmation mechanisms. The user dashboard functionality was specifically designed for data validation purposes, allowing students to verify whether their competition data has been properly entered into the system. The dashboard provides simple but effective filtering by NIM (student ID) and category selection between Mandiri and Belmawa competitions, enabling students to quickly locate and validate their personal competition records. The filtering mechanism between Mandiri and Belmawa categories was tested and confirmed to function smoothly, with each category maintaining its own NIM-based search functionality for enhanced user experience

System performance during testing sessions was consistently acceptable, with response times observed to be well within user expectations for typical operations. The AI chatbot demonstrated response times that consistently appeared to be under 10 seconds for standard queries, as specifically noted by the Student Affairs Staff during their comprehensive testing of the chatbot functionality. Dashboard loading and navigation were smooth and responsive throughout the testing period.

Stakeholder feedback was overwhelmingly positive across all testing sessions. The Dean of FRI confirmed that all administrative features function smoothly and meet faculty needs for monitoring student achievements. The Student Affairs Staff, after comprehensive testing of all administrative features including the AI chatbot, expressed satisfaction with the improved data management workflow compared to previous systems, stating that the system "meets expected criteria and will be very useful for future data input and analysis operations."

The student representative provided particularly insightful feedback regarding the user interface and system value proposition. Kevin noted that "even though I haven't participated in Mandiri or Belmawa competitions yet, this is very helpful as a source of motivation - seeing names displayed on the dashboard creates a historical record visible to all users." He emphasized that the filtering functionality between Mandiri and Belmawa works well, with each category providing effective NIM-based filtering that makes it easy for users to find their data. The simplified dashboard design was particularly appreciated for its motivational value beyond just data validation purposes. All stakeholders confirmed that the system successfully addresses their respective needs and requirements.

Based on the comprehensive testing results and unanimous stakeholder approval, the FRI Independent Competition Dashboard system is declared ready for production deployment. The achievement of 100% test case success rate (25 out of 25 test cases passed) combined with positive stakeholder feedback confirms that the system meets established functional and usability requirements.

The UAT results demonstrate that the system successfully fulfills its intended purpose of providing an integrated platform for managing student competition data

(admin functions) and enabling students to validate their personal competition records (user functions) within the Faculty of Industrial Engineering environment. Additionally, the testing revealed an unexpected benefit where the system serves as a motivational tool for students, providing visibility of achievement records that can inspire participation in future competitions. All stakeholders have provided their acceptance of the system for operational use.

V.2.3 System Usability Scale (SUS) Evaluation

System Usability Scale (SUS) evaluation was conducted to measure the usability performance of the developed AI-integrated dashboard system. The evaluation involved 5 representative users consisting of 3 administrative stakeholders and 2 student users, following the SUS methodology framework established in the literature review.

The individual SUS scores were calculated using the established formula, with results presented in Table V-3.

Table V- 3. Individual SUS calculation results

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total Contribution	SUS Score
Admin												
1	5	1	5	1	5	1	5	1	5	1	40	100.0
2	5	1	5	1	5	1	4	1	5	1	39	97.5
3	5	1	5	1	4	2	5	1	5	1	38	95.0
User												
4	4	2	4	2	5	1	4	1	4	2	33	82.5
5	4	2	4	2	4	2	4	2	4	2	30	75.0
Average												90

Based on Table V-3 the results of the SUS evaluation show an average score of 90 points out of a maximum of 100 points. This score successfully exceeded the minimum target of 70 points set. Out of 5 respondents, all 5 gave scores above the target, with the highest score being 100 and the lowest being 75.

Based on the classification by Sauro & Lewis (2016), a score of 90 falls into the 'A+ (Best Imaginable)' grade, indicating an acceptable level of usability. This result proves that the system has adequate usability for the needs of monitoring student performance at FRI.

The exceptional SUS score of 90.0 indicates that the AI-integrated dashboard system successfully meets usability requirements across different user roles. The high scores from administrative users validate the effectiveness of management features including AI chatbot interaction, data visualization, and CRUD operations. The positive feedback from student users confirms that the simplified dashboard interface effectively serves data validation purposes without overwhelming non-administrative users.

The results demonstrate that the layered architecture approach and careful attention to user experience design have successfully created a system that is both functionally comprehensive and highly usable, meeting the diverse needs of stakeholders within the Faculty of Industrial Engineering environment.

V.3 Implications of the Final Project

This research carries significant implications that extend well beyond the immediate technical achievement of developing an AI-integrated dashboard system. The successful implementation demonstrates how educational institutions can effectively leverage modern technologies to address real institutional management challenges while maintaining accessibility for users with varying technical backgrounds.

From a theoretical perspective, this work advances understanding of how established software engineering principles can accommodate emerging AI technologies without compromising system integrity. The successful implementation of a 4-layer architecture proves that systematic integration of multiple platforms including Laravel, Google Looker Studio, Flowise AI with OpenAI, and Aiven MySQL can be achieved while maintaining system stability and optimal user experience. This architectural approach provides a replicable framework that other researchers can adapt for similar multi-technology integration projects in educational contexts, validating that complex AI systems can be built using structured methodologies like Waterfall when dealing with multiple cloud platforms requiring careful integration planning.

The practical implications for educational institutions are particularly transformative. This research directly addresses a fundamental challenge in higher education where powerful analytical tools often remain underutilized due to technical complexity barriers. By implementing a natural language interface that understands Indonesian queries, the system democratizes access to data analysis capabilities, allowing faculty members and student affairs staff to obtain complex analytical insights through conversational interactions without requiring SQL knowledge or extensive technical training. The exceptional results from User Acceptance Testing, achieving 100% stakeholder approval, combined with the System Usability Scale score of 90 points categorized as A+ Best Imaginable, provide compelling evidence that AI integration can achieve remarkably high user satisfaction when properly designed and implemented.

Economic feasibility represents another crucial implication of this research. The successful implementation proves that sophisticated AI-powered decision support systems can be developed and deployed within realistic resource constraints that most educational institutions face. The strategic combination of cloud-based services with open-source frameworks demonstrates a cost-effective approach that eliminates the need for massive infrastructure investments or specialized technical teams. Furthermore, the role-based access control implementation shows how institutions can effectively balance data security requirements with appropriate transparency for different stakeholder groups, addressing common concerns about data governance in educational technology implementations.

The technological implications extend significantly to the broader landscape of educational technology development, particularly within the Indonesian context. By successfully implementing natural language processing specifically designed for Indonesian language queries, this work addresses linguistic accessibility challenges that are frequently overlooked in international educational technology research. This achievement provides evidence that Indonesian universities can effectively adopt and integrate cutting-edge AI technologies, contributing meaningfully to national capacity building in educational innovation. The iframe-based integration approach offers valuable insights for developers working on

similar multi-platform educational projects, demonstrating how institutions can leverage best-of-breed services while maintaining a unified and coherent user experience.

Perhaps most importantly, this research validates that artificial intelligence can enhance rather than replace human decision-making processes in educational management contexts. The conversational AI interface functions as an analytical assistant that helps stakeholders understand complex data patterns and trends while supporting rather than automating strategic decisions about student development programs. This human-AI collaboration model effectively addresses widespread concerns about AI technology replacing human judgment while simultaneously demonstrating the substantial value that technology can provide in augmenting analytical capabilities and supporting evidence-based decision making.

Looking toward future development, while this research focused specifically on independent competition data within a single faculty, the demonstrated success indicates significant potential for scaling similar approaches to university-wide student analytics, comprehensive academic performance monitoring, or other institutional management areas. The proven effectiveness of natural language interfaces in educational contexts opens exciting possibilities for developing even more intuitive interactions between stakeholders and institutional data systems, potentially transforming how educational institutions approach data-driven decision making across multiple operational domains.

The collective implications of this research ultimately demonstrate that thoughtful and systematic AI integration can significantly enhance educational institutions' analytical capabilities while ensuring these powerful tools remain accessible to users regardless of their technical background or expertise level. This achievement represents not merely a technical success, but a practical advancement in how educational institutions can strategically leverage modern technology to support evidence-based decision making processes and achieve improved student outcomes through more informed and responsive institutional management.

CHAPTER VI
CONCLUSION AND SUGGESTION

VI.1 CONCLUSION

Based on the research conducted in the development of a student independent competition dashboard integrated with AI technology at the Faculty of Industrial Engineering, Telkom University, this research successfully addressed all the problem formulations established in Chapter I. The development of the dashboard integrated with AI technology has been successfully realized through the adoption and adaptation of a 4-layer architecture (Meyer & Webb, 2005; Tu, 2023), which consists of a Data Layer using Aiven MySQL, a Visualization Layer using Google Looker Studio, an Intelligence Layer using Flowise AI integrated with OpenAI GPT-4o, and a Presentation Layer using the Laravel framework. This system is capable of managing 649 records of FRI student competition data for the period 2021-2024, which have undergone data cleaning and transformation processes. The results of the automated testing show that all 25 test cases ran perfectly without errors, proving that all system components have been well integrated and function according to the designed specifications.

AI technology has been successfully utilized through the implementation of a Flowise AI-based chatbot connected to OpenAI GPT-4o to provide interactive analysis in Indonesian. This chatbot is capable of answering complex questions related to student competition data, providing insights into performance trends, and offering strategic recommendations based on historical data. This feature allows stakeholders such as the Dean of FRI and Student Affairs Staff to obtain information and analysis without needing to understand database technicalities or complex queries.

The system successfully improved the capabilities of student monitoring through the provision of an interactive dashboard that can be accessed in real-time with various types of data visualizations such as pie charts, bar charts, and dynamic tables. Based on the results of the User Acceptance Testing (UAT) involving key stakeholders (Dean of FRI, Student Affairs Staff, and students), the system achieved a 100% success rate with positive feedback from all parties. The results

of the System Usability Scale (SUS) show an average score of 90 points, which falls into the A+ category (Best Imaginable), proving that the system has an excellent level of ease of use. Stakeholders provided feedback that this system can assist in the monitoring process, making it accessible instantly through the dashboard and AI chatbot. The three research objectives that were set have been successfully achieved. First, the AI-integrated dashboard for monitoring FRI student competitions was successfully developed with a solid architecture and complete functionality. Second, AI technology has been successfully utilized to provide an interactive conversational-based interface that can deliver in-depth analysis and actionable insights. Third, the capabilities of student monitoring has been successfully improved through an adaptive and user-friendly dashboard system.

This research contributes by implementing an information system that integrates data visualization dashboards with AI technology for monitoring student performance. This system demonstrates how technologies such as Google Looker Studio, Flowise AI, and Laravel can be combined to create a user-friendly and powerful solution through the adaptation of a proven and established layered architecture in software engineering. Additionally, this research also demonstrates the effective adaptation of layered architecture for the context of AI-integrated dashboard systems in educational institutions.

Overall, the student self-competition dashboard system integrated with AI has been successfully developed and meets all the established requirements, as evidenced by comprehensive testing results and positive feedback from stakeholders.

VI.2 SUGGESTION

Based on the results of the research and system evaluation that have been conducted, there are several suggestions for further improvement and development, as follows:

1. Based on observations, the overall system loading takes quite a long time even when run in a local environment. Code and database query optimization are needed to improve the system's response time, including the relatively long loading of embedded Looker Studio.

2. The system can be improved by adding an automatic import feature from Excel files to facilitate the input process of competition data, which is currently done manually, as well as an export feature to various file formats such as Excel (.xlsx), CSV (.csv), or PDF to make it easier for stakeholders to create reports.
3. The AI chatbot can be improved by adding more question templates or FAQs that are frequently asked by stakeholders to facilitate user interaction and enhance the effectiveness of system usage.
4. It is necessary to socialize more student affairs staff so that they can optimally utilize the system, and to update data periodically so that AI visualization and analysis always display the latest information.
5. Further research can be conducted to measure the impact of system usage on the work capabilities of student affairs staff over a longer period, as well as to compare the effectiveness of this system with student performance monitoring systems in other educational institutions.
6. The implementation of the layered architecture adapted in this research can be tested and applied to other domains such as academic performance monitoring or other student activities within the Telkom University environment.

With the implementation of the above suggestions, it is hoped that the student independent competition dashboard system integrated with AI can provide more optimal benefits for the Faculty of Industrial Engineering and can serve as a reference for the development of similar systems in other educational institutions. These improvements are expected to strengthen the system's functionality, ensure sustainability, and open new opportunities for innovation in the academic environment. In addition, involving stakeholders in the evaluation and development process will help ensure that the system remains relevant and aligns with real operational needs. In the future, this system has the potential to be scaled up and integrated with other academic information systems to support broader student performance analytics and institutional decision-making. With continuous support and periodic improvements, the system can become a reliable digital solution for enhancing student achievement monitoring and reporting.

REFERENCES

Tang, C. (2019). Associations Between the Principle of Provenance and Metadata: Practical Implementation and Cases. *American Journal of Information Science and Technology*, *3*(1), 17. https://doi.org/10.11648/j.ajist.20190301.13

Rolansa, F. (2021). Pengembangan interaktif dashboard kemahasiswaan di program studi teknik informatika dengan teknologi big data. *Jurnal Pendidikan Informatika Dan Sains*, *10*(2), 110–118. https://doi.org/10.31571/saintek.v10i2.2190

Nugraha, G. P., Suadaa, L. H., Wilantika, N., & Maghfiroh, L. R. (2024). Pengembangan Aplikasi Chatbot dengan Large Language Model untuk Text-to-SQL Generation. *Seminar Nasional Official Statistics*, *2024*(1), 831–840. https://doi.org/10.34123/semnasoffstat.v2024i1.2252

Murtiastuti, N. D., Liquiddanu, E., & Priyandari, Y. (2023). Perancangan Dashboard Performansi di Bidang Kemahasiswaan dan Lulusan. *Performa: Media Ilmiah Teknik Industri*, *22*(2), 151. https://doi.org/10.20961/performa.22.2.80764

Reis, J. A., Rafael Almeida, J., Almeida, T. M., & Luís Oliveira, J. (2024). Using Flowise to Streamline Biomedical Data Discovery and Analysis. *2024 IEEE 22nd Mediterranean Electrotechnical Conference, MELECON 2024*, 695–700. https://doi.org/10.1109/MELECON56669.2024.10608738

Yigitbasioglu, O. M., & Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, *13*(1), 41–59. https://doi.org/10.1016/j.accinf.2011.08.002

Pauwels, K., Ambler, T., Clark, B. H., LaPointe, P., Reibstein, D., Skiera, B., Wierenga, B., & Wiesel, T. (2009). Dashboards & Marketing: Why, What, How and What Research is Needed? *Journal of Service Research*, *12*(2), 175–189. https://doi.org/10.1177/1094670509344213

Thapa, S., Shiwakoti, S., Shah, S. B., Adhikari, S., Veeramani, H., Nasim, M., & Naseem, U. (2025). Large language models (LLM) in computational social science: prospects, current state, and challenges. In *Social Network Analysis and Mining* (Vol. 15, Issue 1). Springer Vienna. https://doi.org/10.1007/s13278-025-01428-9

Wakhidah, E. N., Sulaeman, M., Metris, D., Priambodo, A., & Prakoso, R. D. Y. (2024). Peran Artificial Intelligence Dalam Transformasi Sumber Daya Manusia Pendidikan: Peningkatan Kualitas Vs Penggantian. *Journal Development*, *12*(1), 10–23. https://doi.org/10.53978/jd.v12i1.383

Glaser, N. (2023). Exploring the Potential of ChatGPT as an Educational Technology: An Emerging Technology Report. *Technology, Knowledge and Learning*, *28*(4), 1945–1952. https://doi.org/10.1007/s10758-023-09684-4

Susila, A. A. N. H., & Sri Arsa, D. M. (2023). Analisis System Usability Scale (SUS) dan Perancangan Sistem Self Service Pemesanan Menu di Restoran Berbasis Web. *Majalah Ilmiah UNIKOM*, *21*(1), 3–8. https://doi.org/10.34010/miu.v21i1.10683

Davis, A. M. (1992). Operational Prototyping: A New Development Approach. *IEEE Software*, *9*(5), 70–78. https://doi.org/10.1109/52.156899

Martin, J. (2006). Prototyping and Rapid Application Development (RAD) RAD - Background What is RAD RAD - Goals RAD - Quality RAD - Properties RAD - Cost. *University of Wolverhampton*.

Shamsulhuda Khan, & Shubhangi Mahadik. (2022). A Comparative Study of Agile and Waterfall Software Development Methodologies. *International Journal of Advanced Research in Science, Communication and Technology*, *June*, 399–402. https://doi.org/10.48175/IJARSCT-5696

Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: a comparative analysis. *International Journal of System Assurance Engineering and Management*, *14*(4), 1504–1522. https://doi.org/10.1007/s13198-023-01958-5

Little, R. J. A., & Rubin, D. B. (2019). Statistical Analysis with Missing Data. In *Journal of Marketing Research* (Vol. 26, Issue 3). https://doi.org/10.2307/3172915

Zhang, W., Wang, Y., Song, Y., Wei, V. J., Tian, Y., Qi, Y., Chan, J. H., Wong, R. C. W., & Yang, H. (2024). Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, *36*(11), 6699–6718. https://doi.org/10.1109/TKDE.2024.3400824

Rane Shripad Pramod, K. A. (2022). Natural Language Interfaces to Database (NLIDB). *International Journal of Innovative Science and Research Technology*, *7*(11).

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, Inc. https://www.researchgate.net/publication/31860304_Information_Dashboard_Design_The_Effective_Visual_Communication_of_Data_S_Few

Soegoto, E. S. (2018). Implementing Laravel framework website as brand image in higher-education institution. *IOP Conference Series: Materials Science and Engineering*, *407*(1). https://doi.org/10.1088/1757-899X/407/1/012066

Freys, J. C., Bigalke, S. M., Mertes, M., Lobo, D. N., Pogatzki-Zahn, E. M., & Freys, S. M. (2024). Perioperative pain management for appendicectomy: A systematic review and Procedure-specific Postoperative Pain Management recommendations. *European Journal of Anaesthesiology*, *41*(3), 174–187. https://doi.org/10.1097/EJA.0000000000001953

Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-glance Monitoring*. Analytics Press.

https://openlibrary.telkomuniversity.ac.id/home/catalog/id/200928/slug/information-dashboard-design-displaying-data-for-at-a-glance-monitoring-2-e-.html

Christin, E. Y., Wahyuningsih, Y., & Mahendrasusila, F. (2024). Penerapan Model Waterfall pada Perancangan Corporate Web. *Jurnal Teknologi Informatika Dan Komputer*, *10*(1), 36–46. https://doi.org/10.37012/jtik.v10i1.1915

Boehm, B. W. (1988). *A Spiral Model of Software Development and Enhancement*. *106*(4), 1605–1610. https://doi.org/10.1002/jbm.b.33974

Jwo, J. S., Lin, C. S., & Lee, C. H. (2021). An Interactive Dashboard Using a Virtual Assistant for Visualizing Smart Manufacturing. *Mobile Information Systems*, *2021*. https://doi.org/10.1155/2021/5578239

Sommerville, I. (2016). Software Engineering State of the Art and Practice. In *Software Engineering* (Tenth, pp. 697–729). IEEE. http://repo.darmajaya.ac.id/4705/1/Software Engineering%2C 10th Edition %28 PDFDrive %29.pdf

Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit. In *Proceedings of the National Academy of Sciences* (Third, Vol. 3, Issue 1). https://ia801609.us.archive.org/14/items/the-data-warehouse-toolkit-kimball/The Data Warehouse Toolkit - Kimball.pdf

Hong, Z., Yuan, Z., Zhang, Q., Chen, H., Dong, J., Huang, F., & Huang, X. (2025). *Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL*. *June*. https://doi.org/10.48550/arXiv.2406.08426

Sitorus, M., & Murti, D. (2024). Analisis Pengaruh Penggunaan Artificial Intelligence Pada Pembelajaran di Cyber University. *Jurnal Ilmu Komputer, Sistem Informasi & Teknologi Informasi (Innotech)*, *1*(2), 90–101.

Sandy, F., Adi Palangi, W., Liling, D., Putra Pratama, M., Studi, P., Pendidikan, T., Keguruan, F., & Pendidikan, I. (2023). Impelentasi Penggunaan Kecerdasan Buatan Dalam Pendidikan Tinggi. *Seminar Nasional Teknologi Pendidikan UKI Toraja*, 111–117.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly: Management Information Systems*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Affolter, K., Stockinger, K., & Bernstein, A. (2019). A comparative survey of recent natural language interfaces for databases. *VLDB Journal*, *28*(5), 793–819. https://doi.org/10.1007/s00778-019-00567-8

Maxim, B., & Pressman, R. (2019). *Software Engeneering A Practitioner ’s Approach*. *January 2020*, 83.

Kent, Beck, Beedle Mike, Grenning James, Highsmith Jim, Mellor Steve, S. K. (2001). Manifesto for Agile Software Development. *Revista Mexicana de Ingeniera Quimica*, *15*(3). https://www.researchgate.net/profile/Godfried-Adaba/post/Looking-for-papers-on-the-software-development-process-Any-

recommendations-will-be-appreciated/attachment/59d639e579197b8077997172/AS%3A403742915612673%401473271220194/download/Manifesto+of+Agile+So

Dam, S. K., Hong, C. S., Qiao, Y., & Zhang, C. (2024). *A Complete Survey on LLM-based AI Chatbots*. 1–23. http://arxiv.org/abs/2406.16937

Royce, W. W. (1970). Managing the Development of Large Software Systems (1970). *Ideas That Created the Future*, *August*, 321–332. https://doi.org/10.7551/mitpress/12274.003.0035

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2024). Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proceedings of the VLDB Endowment*, *17*(5), 1132–1145. https://doi.org/10.14778/3641204.3641221

Purwani, T., Wahyuni, A., Wicaksono, A. P., & Charunia, H. D. (2021). Dashboard Untuk Visualisasi Data Penjualan Barang Pada Toko Puppets Skateboard Semarang. *Jurnal Komputaki*, *7*(1), 1–12.

Sauro, J., & Lewis, J. R. (2016). Standardized usability questionnaires. In *Quantifying the User Experience* (pp. 185–248). https://doi.org/10.1016/b978-0-12-802308-2.00008-4

Fowler, M. (2018). *Test Pyramid*. https://martinfowler.com/bliki/TestPyramid.html

Laravel Dusk Documentation. (2025). https://laravel.com/docs/12.x/dusk

Aiven docs. (2025). https://aiven.io/docs/

IEEE Computer Society, & Software & Systems Engineering Standards. (2018). *IEEE Standard for Software and System Test Documentation IEEE Computer Society* (Vol. 2008, Issue July).

Stauffer, M. (2019). *Laravel: Up & Running THIRD EDITION A Framework for Building Modern PHP Apps*. http://oreilly.com

Brooke, J. (1996). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry, June*, 207–212. https://doi.org/10.1201/9781498710411-35

Meyer, M. H., & Webb, P. H. (2005). Modular, layered architecture: the necessary foundation for effective mass customisation in software. *International Journal of Mass Customisation*, *1*(1), 14. https://doi.org/10.1504/ijmassc.2005.007349

Tu, Z. (2023). Research on the Application of Layered Architecture in Computer Software Development. *Journal of Computing and Electronic Information Management*, *11*(3), 34–38. https://doi.org/10.54097/jceim.v11i3.08

Richards, M. (2015). Software Architecture Patterns. In *Home healthcare nurse* (Vol. 32, Issue 5 Suppl). http://www.ncbi.nlm.nih.gov/pubmed/24798474

Appendix 1. Document UAT

DOKUMENTASI USER ACCEPTANCE TEST	
Nama Proyek	: *Sistem Manajemen Kompetisi*
Studi Kasus	: *Pengelolaan Data Kompetisi Belmawa dan Mandiri*
Penyedia Layanan	: *Muhammad Hanif Al Hadi & Muhammad Haikal Zulhaj*
Manajer Proyek	: *Muhammad Hanif Al Hadi & Muhammad Haikal Zulhaj*
Tanggal Dokumen	: 20 June 2025

Hasil Uji UAT - Prof. Dr. Irfan Darmawan, S.T., M.T.

NO	Use Case/Proses	Berhasil/ Gagal	Diuji oleh	Tanggal Test
1.	*Nama Uji* : Login Admin *Deskripsi Pengujian* : Verifikasi hak akses login untuk Admin dan User biasa dengan autentikasi email dan password *Kasus Uji* : - Email: admin@gmail.com - Password: admin123 - Role: Admin Hasil yang diharapkan : - Jika berhasil akan menampilkan dashboard sesuai role (Admin/User) - Jika gagal akan menampilkan error message di bawah form email - Validasi format email dan password wajib diisi	Berhasil	Prof. Dr. Irfan Darmawan.	20 Juni 2025
2.	*Nama Uji* : Add Competition - Kompetisi Belmawa *Deskripsi Pengujian* : Verifikasi penambahan data kompetisi baru untuk kategori Belmawa *Kasus Uji* : - Semua field wajib diisi (required) - Kategori: Belmawa - Validasi: Semua field harus terisi Hasil yang diharapkan : - Jika berhasil akan menampilkan konfirmasi 'Data kompetisi berhasil disimpan' dan data tersimpan di database Aiven - Jika gagal akan menampilkan pesan error validasi field yang kosong	Berhasil	Prof. Dr. Irfan Darmawan.	20 Juni 2025
3.	*Nama Uji* : Add Competition - Kompetisi Mandiri *Deskripsi Pengujian* : Verifikasi penambahan data kompetisi baru untuk kategori Mandiri *Kasus Uji* : - Semua field wajib diisi (required) - Kategori: Belmawa - Validasi: Semua field harus terisi Hasil yang diharapkan : - Jika berhasil akan menampilkan konfirmasi 'Data kompetisi berhasil disimpan' dan data tersimpan di database Aiven - Jika gagal akan menampilkan pesan error validasi field yang kosong	Berhasil	Prof. Dr. Irfan Darmawan.	20 Juni 2025
4.	*Nama Uji* : View All Competitions *Deskripsi Pengujian* : View All Competitions - Read Data *Kasus Uji* : - Admin mengakses menu "View All Competitions" pada masing masing kompetisi - Menampilkan seluruh kolom dan baris data kompetisi - Data historical Belmawa dan Mandiri ditampilkan pada masing masing halaman	Berhasil	Prof. Dr. Irfan Darmawan.	20 Juni 2025

Hasil Uji UAT - Rahadhitya Samudra Anuraga

NO	Use Case/Proses	Berhasil/ Gagal	Diuji oleh	Tanggal Test
	- Validasi: Semua field harus terisi Hasil yang diharapkan : - Jika berhasil akan menampilkan konfirmasi 'Data kompetisi berhasil disimpan' dan data tersimpan di database Aiven - Jika gagal akan menampilkan pesan error validasi field yang kosong			
4.	*Nama Uji* : View All Competitions *Deskripsi Pengujian* : View All Competitions - Read Data *Kasus Uji* : - Admin mengakses menu "View All Competitions" pada masing masing kompetisi - Menampilkan seluruh kolom dan baris data kompetisi - Data historical Belmawa dan Mandiri ditampilkan pada masing masing halaman Hasil yang diharapkan : - Menampilkan tabel lengkap dengan seluruh data kompetisi pada masing masing halaman Data yang sebelumnya ditambahkan muncul dalam list - Kolom dan baris tertata dengan rapi	Berhasil	Rahadhitya Samudra Anuraga	24 Juni 2025
5.	*Nama Uji* : Update Competition Data *Deskripsi Pengujian* : Verifikasi proses edit/update data kompetisi yang sudah ada *Kasus Uji* : - Admin klik "View All Competitions" - Admin klik button "Update" pada baris data yang dipilih - Sistem mengarahkan ke tampilan edit - Admin mengubah data yang diinginkan - Admin menyimpan perubahan Hasil yang diharapkan : Jika berhasil akan menampilkan konfirmasi 'Data kompetisi berhasil diupdate' Data yang diubah tersimpan di database dan muncul di list dengan nilai yang baru - Jika gagal akan menampilkan pesan error	Berhasil	Rahadhitya Samudra Anuraga	24 Juni 2025
6.	*Nama Uji* : Delete Competition Data *Deskripsi Pengujian* : Verifikasi proses hapus data kompetisi dengan konfirmasi *Kasus Uji* : Admin klik "View All Competitions" - Admin klik button "Delete" pada baris data yang dipilih - Sistem menampilkan alert message konfirmasi hapus - Admin memilih "Ya" untuk konfirmasi atau "Tidak" untuk batal Hasil yang diharapkan : - Jika "Ya": Data terhapus dari database dan tidak muncul lagi di list - Jika "Tidak": Data tetap ada dan tidak terhapus - Alert message harus muncul sebelum proses hapus	Berhasil	Rahadhitya Samudra Anuraga	24 Juni 2025
7.	*Nama Uji* : Chatbot Access *Deskripsi Pengujian* : Chatbot Interaction - Akses Chatbot *Kasus Uji* : Admin mengklik icon/button chatbot - Sistem menampilkan jendela pop-up conversation	Berhasil	Rahadhitya Samudra Anuraga	24 Juni 2025

Hasil Uji UAT – Kevin Ariel Wibowo

NO	Use Case/Proses	Berhasil/ Gagal	Diuji oleh	Tanggal Test
	- User mengakses halaman login Memilih button "Login sebagai User/Student" Memasukkan email dan password yang valid Hasil yang diharapkan : - Login berhasil dengan role verification - Sistem mengarahkan langsung ke Competition Analytics Dashboard - Tidak ada akses ke fitur admin			
3.	*Nama Uji* : Akses Dashboard Kompetisi untuk User *Deskripsi Pengujian* : Verifikasi tampilan dasnboard user yang embed dari Looker Studio *Kasus Uji* : Setelah login, user langsung melihat "Competition Analytics Dashboard" Terdapat filter "Kompotisi Mandiri" dan "Kompotisi Belmawa" di sebelah kiri - Tabel menampilkan kolom: NIM, Nama, Juara, Nama Kompetisi, Program Studi, Tingkat, Status, dan lain lain Hasil yang diharapkan : - Dashboard tertampil dengan embed Looker Studio yang benar - Tabel data kompetisi dapat dilihat dengan jelas - Filter berfungsi untuk memilih data user berdasarkan NIM	Berhasil	Kevin Ariel Wibowo	26 Juni 2025
4.	*Nama Uji* : Validasi Data Kompetisi Personal *Deskripsi Pengujian* : Verifikasi kemampuan user memvalidasi data mereka sudah masuk atau belum *Kasus Uji* : - User mencari NIM atau nama mereka di tabel - Menggunakan filter atau scroll untuk menemukan data pribadi - Memverifikasi kompetisi yang pernah diikuti sudah tercatat Hasil yang diharapkan : - User dapat menemukan data kompetisi mereka (jika ada) - Data yang ditampilkan akurat dan sesuai - User dapat mengkonfirmasi status data mereka	Berhasil	Kevin Ariel Wibowo	26 Juni 2025
5.	*Nama Uji* : Penggunaan Filter Kompetisi *Deskripsi Pengujian* : Verifikasi fungsi filter "competisi Mandiri" dan "Kompetisi Belmawa" *Kasus Uji* : - User mengklik filter "Kompetisi Mandiri" User mengklik filter "Kompotisi Belmawa" Melihat perubahan data yang ditampilkan Hasil yang diharapkan : - Filter berfungsi dengan benar - Data yang ditampilkan berubah sesuai filter yang dipilih - Interface Looker Studio responsif	Berhasil	Kevin Ariel Wibowo	26 Juni 2025
6.	*Nama Uji* : Verifikasi Akses Terbatas User *Deskripsi Pengujian* : Memastikan user hanya memiliki akses read-only *Kasus Uji* : User mencoba mencari tombol edit/add/delete (tidak ada) User tidak melihat fitur admin seperti Quick Actions - User tidak memiliki akses ke AI Chatbot	Berhasil	Kevin Ariel Wibowo	26 Juni 2025

DAFTAR HADIR

Agenda : **PENGUJIAN PENERIMAAN PENGGUNA SmartInsight**
Judul Proyek Akhir (PA)
Mitra : ***Fakultas Pengguna Sistem SmartInsight***
Penyelenggara : **S1 *Sistem Informasi – Fakultas Rekayasa Industri***
Tempat/Tanggal : **Menyesuaikan**
Waktu : **10.00 - Selesai**

Telkom University

PESERTA PENGUJIAN PENERIMAAN PENGGUNA

No	Peran	Nama	No.Kontak (hp/email)	Tanda Tangan
1.	DEKAN FRI	Prof. Dr. Irfan Darmawan, S.T., M.T.	082218909229	
2.	Staff Kemahasiswaan FRI	Rahadhitya Samudra Anuraga	082177765774	
3.	Mahasiswa	Kevin Ariel Wibowo	081391829185	
4.				
5.				
6.				
7.				
8.				
9.				
10.				

Pembuat Dokumen,

Muhammad Hanif Al Hadi & Muhammad Haikal Zulhaj
1202213165 & 1202213383

Menyetujui,

Prof. Dr. Irfan Darmawan, S.T., M.T.
14750038

Appendix 2. Dokumentasi UAT bersama stakeholder

Appendix 3. Surat pengantar penelitian

Nomor : 1741/AKD09/RI-WD1/2025
Lampiran : -
Perihal : Permohonan Penelitian Skripsi

Kepada Yth,
Kepala Urusan Kemahasiswaan
Fakultas Rekayasa Industri Telkom University
Telkom University Landmark Tower Lt 18

Dengan Hormat,

Kami informasikan bahwa mahasiswa kami atas nama:

NO	NIM	NAMA	PRODI	FAKULTAS
1	1202212165	Muhammad Hanif Al Hadi	S1 Sistem Informasi	Fakultas Rekayasa Industri
2	1202212383	Muhammad Hafizh Zulhaj	S1 Sistem Informasi	Fakultas Rekayasa Industri

Dengan topik:

Integrasi Dashboard Kompetisi Mahasiswa dengan Artificial Intelligence di FRI Universitas Telkom

Bermaksud melakukan pengambilan data untuk **Skripsi** di perusahaan/lembaga yang bapak/ibu pimpin terkait dengan **penelitian** sesuai topik yang sedang dilakukan.

Oleh karena itu, kami mohon bapak/ibu berkenan memberikan izin kepada yang bersangkutan.

Demikian surat permohonan yang kami sampaikan. Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Bandung, 11 Mei 2025
Wakil Dekan I Bidang Akademik dan Dukungan Penelitian

Muhammad Nashir Ardiansyah, S.T., M.T. Ph.D.
NIP. 14880044-1

Telkom University Learning Center Building - Bandung Technopolex | Jl. Telekomunikasi, Terusan Buah Batu, Bandung 40257, West Java, Indonesia
t: 62-22-7564108 f: 62-22-7565200 e: info@telkomuniversity.ac.id
www.telkomuniversity.ac.id

Appendix 4. Pertanyaan AI chatbot

1. Pertanyaan:

Berdasarkan data partisipasi kompetisi tahun 2021–2024, berikan analisis faktor konkret yang menyebabkan lonjakan partisipasi di tahun 2022 (misalnya program insentif, dukungan pembina, kebijakan fakultas), dan buatkan strategi spesifik untuk meningkatkan partisipasi kompetisi individu dari 12% menjadi minimal 25%, dengan pendekatan yang dapat diterapkan oleh fakultas.

Jawaban:

Ringkasan Temuan: Berdasarkan data partisipasi kompetisi dari tahun 2021 hingga 2024, terdapat lonjakan signifikan pada tahun 2022 dengan total partisipasi mencapai 261, dibandingkan dengan 100 pada tahun 2021. Namun, partisipasi menurun drastis pada tahun 2023 menjadi 189 dan kembali turun pada tahun 2024 menjadi 99. Lonjakan ini menunjukkan adanya faktor-faktor tertentu yang mendorong peningkatan partisipasi di tahun 2022.

Analisis Pola: Hubungan antara program studi dan jenis lomba menunjukkan bahwa program studi yang lebih aktif dalam mendukung partisipasi kompetisi, seperti S1 Teknik Industri dan S1 Sistem Informasi, berkontribusi lebih besar terhadap jumlah partisipasi. Selain itu, adanya dukungan dari pembina dan kebijakan fakultas yang mendorong mahasiswa untuk berpartisipasi dalam kompetisi individu maupun kelompok juga berperan penting dalam hasil lomba.

Sorotan Kuantitatif: Dari data yang diperoleh, total partisipasi untuk tahun 2021 adalah 100, meningkat menjadi 261 pada tahun 2022, dan menurun kembali pada tahun-tahun berikutnya. Partisipasi individu pada tahun 2022 tercatat hanya 12%, yang menunjukkan bahwa sebagian besar partisipasi berasal dari kompetisi kelompok. Dengan target untuk meningkatkan partisipasi individu menjadi minimal 25%, perlu ada upaya strategis yang lebih terfokus.

Rekomendasi Strategis:

Program Insentif: Implementasikan program insentif bagi mahasiswa yang berpartisipasi dalam kompetisi individu, seperti penghargaan atau beasiswa kecil.

Dukungan Pembina: Tingkatkan pencatatan dan pelatihan pembina untuk mendukung mahasiswa dalam persiapan kompetisi individu, serta menyediakan sesi konsultasi.

Kampanye Kesadaran: Lakukan kampanye kesadaran tentang pentingnya kompetisi individu melalui seminar, workshop, dan promosi di media sosial untuk menarik lebih banyak peserta.

individu dapat meningkat dari 12% menjadi minimal 25% dalam waktu satu tahun. Langkah tindak lanjut yang realistis termasuk melakukan survei untuk mengidentifikasi hambatan yang dihadapi mahasiswa dalam berpartisipasi, serta mengadakan evaluasi rutin terhadap efektivitas program insentif dan dukungan pembina.

Dengan pendekatan berbasis data ini, fakultas dapat lebih memahami faktor-faktor yang mempengaruhi partisipasi dan merumuskan strategi yang lebih efektif untuk meningkatkan keberhasilan mahasiswa dalam kompetisi.

TABLE REVISI

NO	BAGIAN	REVISI	HAL	KETERANGAN
1	ABSTRACT	Hilangkan insight mengenai improvement of efficiency	i	Sudah saya hilangkan dan sesuaikan ulang abstractnya
2	GLOSSARY	sesuaikan font size	xvi	Font sudah saya sesuaikan dan seragamkan
3	CH I	Perbaiki rumusan masalah no (3) yang sepertinya improvement of effeciency tidak terbukti melalui penelitian ini	3	Sudah saya perbaiki dan sesuaikan berdasarkan penelitian saya
		jangan memulai paragraf di akhir halaman	all	Sudah saya cross-cehck
		tambahkan referensi, karena pada introduction ini hanya ada 1 (Pressman & Maxim, 2019)	1	Sudah saya tambahakan total ada 5 referensi yang sesuai terkait penelitian ini
4	CH II	Hindari 1 subbab yang hanya 1 paragraf saja e.g II.4, II.10, II.11	7-29	Sudah saya cross-check dan saya sesuaikan
		Improve teori mengenai Waterfall, tambahkan ilustrasi tahapannya	19	Sudah saya improve teori dan ilustrasi berdasarkan teori
		tambahkan teori UML	13	Sudah saya tambahkan teori terkait UML dari yang umum hingga yang saya gunakan
		Improve teori mengenai SUS ataupun UAT ataupun metode pengujian lainnya	14,15,17	Sudah saya improve teori terkait pengujian yang saya pakai pada penelitian ini
		Urutkan tabel II-3 berdasarkan tahun terbit terbaru -> terlama	24	Sudah saya urutkan penelitian tahun terbaru ke terlama
		Rangkum insight dari tabel II-3	28	Sudah saya rangkum terkait penelitian terdahulu dan memahami serta memahami gap
		jelaskan perbedaan penelitian tugas akhir ini dengan penelitian terdahulu pada tabel II-3	24	Sudah saya jelaskan pada kolom limitation
		Jelaskan kesimpulan tabel II-3	28	Sudah saya jelaskan setelah table II-3
5	CH III	Pada bab III sesuai panduan terbaru seharusnya III.1 Metode Penelitian, dan III.2 Sistematika Penyelesaian masalah, sehingga sudah tidak diperlukan lagi model konseptual. Namun silahkan didiskusikan dgn pembimbing terkait penyajian Bab III.	30	Berdasarkan diskusi dengan dosen pembibing telah disetujui dengan penyajian BAB III yang sudah terlampir
		Gambar tidak boleh sama	32	Sudah saya bedakan

NO	BAGIAN	REVISI	HAL	KETERANGAN
6	CH IV	Kurangi penyajian coding	55	Sudah saya ganti dengan gambar prosess, bukan codingan lagi
7	CH V	Tambahkan subbab Implikasi tugas akhir	143	Sudah saya tambahkan implikasi pada sub bab V.3
8	CH VI	Improve conclusion	146	Sudah saya improve dan sesuaikan.
9	DAFTAR PUSTAKA	Seragamkan penulisan karena beberapa judul ditulis kapital	149	Judul sudah tidak ditulis kapital
10	GENERAL FORMATTING	Tidak perlu menggunakan indent untuk paragrafnya	all	Sudah di cross-check tidak ada indent lagi untuk paragraf
		Betulkan sitasi jika disimpan di awal atau tengah kalimat, maka bukan “According to (Damyanov & Tsankov, 2019)..” tapi “Accroding to Damyanov & Tsankov (2019)…”	all	Unutk setiap sitasi sudah saya sesuaikan kemudian saya cross-check, sehingga sudah sesuai arahan
		Seluruh tabel dan gambar harus dirujuk/dimention pada narasi/paragraf	all	Seluruh gambar dan table sudah dirujuk pada narasi/paragraf
		avoid blank space	all	Sudah saya sesuaikan setiap halamannya sehingga menghindari blank space/white space
		jangan menyimpan gambar tepat setelah judul, awali dengan text	all	Sudah saya sesuaikan dan mengatur ulang narasi yang sesuai
		Tambahkan corak atau kekhasan EDM ‘data Engineering’	99,123	Menambahkan aspek data engineering melalui clustering analysis pada subbab IV.4.3.5 dan V.1.12, dengan AI chatbot yang dapat memvalidasi dan memberikan insights berbasis hasil clustering untuk memperkuat corak EDM, berdasarkan permintaan dari penguji 1 pada diskusi sidang.
11	TAMBAHAN	Bahasa indonesia	123-124	mohon maaf, ada alasan khusus. itu adalah contoh user input bahasa Indonesia. Karna chatbot ini memang untuk memahami bahasa indonesia jadi

NO	BAGIAN	REVISI	HAL	KETERANGAN
				itu merupakan contoh real case input indonesian language yang akan digunakan oleh pengguna. ini menunjukan chatbot berhasil melakukan clustering analysis berdasarkan user input