CHAPTER 1
INTRODUCTION

1.1 Background

Ensuring safe air travel depends on the effective delivery of operational updates to flight crews. One of the primary tools used for this purpose is the Notice to Airmen (NOTAM), which communicates time-sensitive alerts about issues such as airspace closures, changes in runway availability, or disruptions in navigation systems [1]. These notices are written in a fixed format, using uppercase letters and aviation-specific abbreviations that are not immediately intuitive [1]. Their purpose is to inform pilots and air traffic personnel of real-time conditions that cannot be published through standard procedures [1]. To help categorize these messages, NOTAMs are labeled using Q-codes, which indicate the general type of alert, such as airspace activity or airport facility updates [2]. Despite their safety-critical nature, the volume and complexity of these notices often make them difficult to process. Pilots preparing for a flight may need to review lengthy documents filled with technical shorthand and irregular phrasing [3]. This increases the risk that important alerts may be buried under less relevant content. Improving how NOTAMs are structured and delivered is essential to ensure that the most critical information is accessible and clearly presented.

In response to these challenges, research has introduced machine learning approaches to enhance how NOTAMs are managed. One area of focus has been on classification. Historically, Q-codes were assigned manually by experts after reviewing the content of each notice. Recent developments in language processing, however, have enabled automated systems to predict these labels based on the text itself. One such approach involved building a classifier that analyzes the free-text section of a NOTAM and generates the corresponding Q-code along with an explanation of its decision [2]. This assists in sorting NOTAMs by relevance, helping crews focus on what directly affects their operations. Studies have shown that natural language processing tools can effectively reorganize these notices into a structured and more user-friendly format [1]. In particular, a project supported by NASA developed a customized model to categorize NOTAMs with a high degree of reliability [1]. This model has been integrated into tools that label new NOTAMs as they arrive, improving how pilots and dispatchers receive targeted alerts in real time

[1]. These innovations demonstrate how intelligent systems can reduce the burden of excessive information by emphasizing the most relevant updates.

Beyond sorting notices, other efforts have focused on pulling structured information from the unstructured content of NOTAMs. Important elements such as locations, altitudes, times, and hazards are often embedded in free-form language and technical codes. Researchers have addressed this by training models on large collections of NOTAMs to identify and extract these elements accurately. One approach involved adapting a language model to recognize entities like aerodrome codes, coordinates, and temporal markers [3]. The model was refined for tasks such as named entity recognition and showed strong results in extracting meaningful data from these complex texts [3]. This suggests a path forward for tools that not only interpret or classify notices but also generate clearer and more tailored versions of the messages themselves. Such tools could play a major role in improving how vital operational updates are delivered and understood in aviation contexts.

One key limitation is the lack of language models that are specifically trained on aviation-related material. Although the aviation industry produces a significant amount of technical writing, including maintenance records, flight safety reports, weather forecasts, and NOTAMs themselves, much of this content has not been annotated or systematically used for training AI models [4]. As a result, general-purpose language models may not fully understand the terminology, structure, or factual expectations required in aviation contexts [4]. This limits their usefulness when applied to tasks like NOTAM generation, where both format and accuracy are critical.

Recent work has begun to address this issue through the development of domain-focused language models. AviationGPT, for instance, was created by continuously training on various aviation documents to capture the specific language and knowledge patterns found in the field [4]. By doing so, it is better suited for tasks such as summarization, document interpretation, and question answering in aviation settings [4]. These improvements suggest that tailoring language models to the domain can lead to better performance and more reliable outputs [4].

Even with these advances, generating complete NOTAM messages remains a difficult challenge. A model must produce text that follows the strict formatting rules of NOTAMs, use the correct codes and abbreviations, and most importantly, remain factually grounded in the current operational environment. A generic language model, when applied without proper controls, may produce outputs that look correct but contain subtle errors or outdated information. These risks highlight the importance of connecting any language generation process to trustworthy and

up-to-date aviation data. Without that grounding, even a technically impressive model could produce output that is misleading or incomplete. Bridging this final gap will require approaches that ensure every generated NOTAM is accurate, aligned with the latest airspace conditions, and presented in a form that flight crews can rely on.

1.2 Problem Identification

Despite recent progress in organizing and interpreting NOTAMs using machine learning, the ability to automatically generate high-quality NOTAM messages remains out of reach. Most research has focused on well-defined tasks such as tagging, classification, or converting unstructured text into formal representations. While these steps are important, they fall short of producing finalized notices that are ready for operational use by pilots and flight planners. There is still no widely adopted AI system that can take complex input such as situational data, early draft text, or informal notes and turn it into a well-formed NOTAM that meets regulatory and communication standards.

1.3 Objectives

The objectives of this thesis are as follows:

1. To create a framework to generate a highly regulated NOTAM by combining LLM's great natural language knowledge and retrieval augmented generation.
2. To compare several LLM using the same evaluation method to assess our proposed framework impact and performance
3. To compare several embedding model using the same evaluation method to gain insight from the retrieval augmented generation system.

1.4 Scope of Work

The assumptions and limitations of the problem in this thesis are:

1. This thesis focuses on NOTAMs published in Indonesia.
2. The input data used in this thesis is in Indonesian.
3. The dataset used consists of 900 free-form natural language and NOTAM pair.

4. The proposed framework is evaluated using the accuracy metric by comparing the generated NOTAM with the already published NOTAM.

5. The large language models used in this study include several GPT variants from OpenAI, namely gpt-3.5, gpt-4o, gpt-4o-mini, gpt-4.1-nano, gpt-4.1-mini, and gpt-4.1. Models with dedicated reasoning capabilities were intentionally excluded from the evaluation.

1.5 Expected Results

The expected result of this thesis is that the proposed framework will be able to generate structured NOTAMs from free-text input (natural language) written in Indonesian. The generated NOTAMs should follow the formatting and content rules set by the Indonesian aviation authority. The evaluation results are expected to show how well the model performs when using the proposed framework compared to when it does not. This research also aims to contribute to the advancement of knowledge in the field of artificial intelligence and its application in the aviation industry.

1.6 Research Methodology

The methodology used for the process of completing this research consists of several stages such as:

1. Design and implement the overall workflow of the framework using several techniques such as LLM as the main NOTAM generator, embedding model to convert string into vector, vector similarity search to find the most similar vector to the input, and prompting to put all of the information together.

2. Run experiments by iterating through the test data to get the generated NOTAM and carry out evaluations using evaluation metrics such as accuracy, normalized discounted cumulative gain (NDCG), mean reciprocal rank (MRR), and recall while analyzing the results.

The detailed explanation about research methodology used in this study will be discussed in Section 3.4.

1.7 Structure of Thesis

The structure of this thesis writing can be seen as follows:

- **CHAPTER II: RELATED WORKS**

  This chapter presents a review of related works concerning the algorithms employed in this thesis. It provides an overview of NOTAM, Retrieval-Augmented Generation (RAG), and text embedding models. Additionally, this chapter discusses several previous studies that have explored these topics.

- **CHAPTER III: THE PROPOSED METHOD**

  This chapter outlines the system design and experimental procedures implemented in this thesis. It describes the main contributions of the research and provides a detailed explanation of the functionality of each component within the proposed framework.

- **CHAPTER IV: PERFORMANCE EVALUATIONS**

  This chapter presents the results and analysis of the conducted experiments. It compares the performance of NOTAM generation between the baseline model and the proposed framework. Additionally, the retrieval performance of various embedding models is examined and discussed in detail.

- **CHAPTER V: CONCLUSION**

  This chapter presents the conclusions drawn from the research findings. It outlines the limitations encountered during the study and discusses potential directions for future work.